



International Technical Support Organization

Advanced POWER Virtualization on IBM System p5

Annika Blank
Paul Kiefer
Carlos Sallave Jr.
Gerardo Valencia
Jez Wain
Armin M. Warda

December 2005

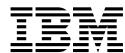
Note: Before using this information and the product it supports, read the information in "Notices".

Second Edition (December 2005)

This edition applies to IBM AIX 5L Version 5.3, HMC Version 5 Release 1.0, Virtual I/O Server Version 1.2 running on IBM System p5 and IBM eServer p5 systems.

© Copyright International Business Machines Corporation 2004, 2005. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADPSchedule Contract with IBM Corp.



Международная организация технической поддержки

Технология Advanced POWER Virtualization в IBM System p5

Анника Бланк
Пол Кифер
Карлос Сальяве, мл.
Герардо Валенсия
Джез Вейн
Армин М. Варда

Перед использованием данного руководства и продукта ознакомьтесь с информацией в разделе «Примечания».

Перевод
А. Казаков, И. Легостаев, Д. Миронов

Научный редактор
IBM Certified Advanced Technical Expert – IBM System p5
Д. Миронов

Первое издание на русском языке (2007 г.)

Издание охватывает IBM AIX 5L (версия 5.3), HMC (версия 5, релиз 1.0), Virtual I/O Server (версия 1.2) на IBM System p5 и IBM eServer p5 systems.

© Copyright International Business Machines Corporation 2004, 2005. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Second Edition (December 2005)

© IBM Corporation (Корпорация International Business Machines Corporation), 2004, 2005 г.
Все права защищены.

Ограничение прав пользователей правительством США: использование, копирование и распространение сведений настоящего руководства ограничено условиями контракта GSA ADP Schedule с корпорацией IBM. Второе издание – декабрь 2005 г.

Оглавление

Рисунки	X
Таблицы	XV
Примечания	XVI
Предисловие	XIX
Глава 1. Введение	1
1.1. Решение Virtualization Engine компании IBM и модель предоставления ресурсов «по требованию»	2
1.2. Решение Virtualization Engine в системе IBM System p5	4
1.2.1. POWER Hypervisor	4
1.2.2. Технология параллельной многопоточной обработки (SMT)	4
1.2.3. LPAR и разделы общего процессорного пула	5
1.2.4. Динамическая реконфигурация	5
1.2.5. Virtual LAN	5
1.2.6. Virtual I/O	5
1.2.7. Нарашивание ресурсов «по требованию» (CUoD)	5
1.2.8. Поддержка нескольких операционных систем	6
1.2.9. Integrated Virtualization Manager	6
1.3. Характеристики RAS виртуализованных систем	6
1.3.1. Надежность, доступность и обслуживаемость	6
1.3.2. Доступность и обслуживаемость в виртуализованных средах	8
1.4. Безопасность в виртуализованной среде	10
1.5. Поддержка операционных систем	10
1.5.1. IBM AIX 5L для систем System p5	11
1.5.2. Linux для систем System p5	11
1.5.3. IBM i5/OS для систем System p5	12
1.5.4. Итоги	15
1.6. Сравнение двух технологий виртуализации компании IBM	16
Глава 2. Ценность технологии Advanced POWER Virtualization	19
2.1. Упрощение ИТ-систем и оптимизация ТCO	20
2.2. Доступность бизнес-приложений	23
2.2.1. Улучшенные средства перемещения приложений	24
2.2.2. Решения с высокой доступностью с НАСМР и APV	27
2.3. Улучшение решений обеспечения непрерывности бизнеса	29
Глава 3. Технологии Virtualization Engine в серверах System p5	33
3.1. Новые функции в версии 1.2 сервера Virtual I/O Server	34
3.1.1. Виртуальные DVD-RAM, DVD-ROM и CD-ROM	34
3.1.2. Переход на резерв с помощью общего Ethernet-адаптера	34

3.1.3.	Integrated Virtualization Manager	35
3.1.4.	Новые команды пула хранения	36
3.1.5.	Улучшения НМС	36
3.2.	Функция Advanced POWER Virtualization	37
3.3.	Введение в микроразделы	40
3.3.1.	Разделы с общими процессорами	40
3.3.2.	Обзор общего процессорного пула	45
3.3.3.	Наращивание ресурсов «по требованию» (CUoD)	48
3.3.4.	Динамическое освобождение процессоров и резервирование процессоров	48
3.3.5.	Динамические разделы	49
3.3.6.	Учитываемые факторы	49
3.4.	Ознакомление с процессором POWER5	52
3.5.	Начальные сведения об одновременной многопоточной обработке (SMT)	53
3.5.1.	Режим SMT процессора POWER5	54
3.5.2.	SMT и AIX 5L	54
3.5.3.	Управление SMT в Linux	57
3.6.	Начальные сведения о POWER Hypervisor	57
3.6.1.	Диспетчеризация гипервизором POWER виртуальных процессоров	58
3.6.2.	Гипервизор POWER и виртуальный ввод-вывод	61
3.6.3.	Системный порт (поддержка виртуального TTY/консоли)	62
3.7.	Лицензирование ПО в виртуализованной среде	62
3.7.1.	Лицензирование IBM i5/OS	62
3.7.2.	Методы лицензирования ПО для операционных систем UNIX	63
3.7.3.	Факторы лицензирования в виртуализированной системе	63
3.7.4.	Планирование и обеспечение лицензий программного обеспечения IBM	66
3.7.5.	Лицензирование Sub-capacity для программного обеспечения IBM	69
3.7.6.	Лицензирование программного обеспечения IBM	71
3.7.7.	Лицензирование операционной системы Linux	75
3.8.	Ознакомление с виртуальным и разделяемым Ethernet	76
3.8.1.	Виртуальная сеть	77
3.8.2.	Построение сетей между разделами с помощью виртуального Ethernet	84
3.8.3.	Совместное использование физических Ethernet-адаптеров ..	85
3.8.4.	Пример конфигурации виртуального и общего Ethernet	89
3.8.5.	Ограничения и учитываемые факторы	93
3.9.	Ознакомление с виртуальным SCSI	93
3.9.1.	Доступ разделов к виртуальным SCSI-устройствам	94
3.9.2.	Основные учитываемые факторы	99
3.10.	Ознакомление с Partition Load Manager	101
3.11.	Integrated Virtualization Manager	102
3.11.1.	Основные правила установки IVM	103
3.11.2.	Конфигурирование разделов с помощью IVM	105
3.12.	Динамические операции с LPAR	106

3.13.	Концепции виртуального ввода-вывода в Linux	107
3.13.1.	Драйверы устройств Linux для виртуальных устройств	
	IBM System p5	108
3.13.2.	Linux как VIO-клиент	108
3.13.3.	Linux как VIO-сервер	111
3.13.4.	Учитываемые факторы	112
3.13.5.	Что читать дальше	113
Глава 4. Установка Virtual I/O Server: базовые настройки	115	
4.1.	Начальные сведения	116
4.1.1.	Интерфейс командной строки	116
4.1.2.	Управляемые аппаратные ресурсы	119
4.1.3.	Структура пакета ПО и поддержка	120
4.2.	Создание раздела с Virtual I/O Server	121
4.2.1.	Определение раздела с Virtual I/O Server	121
4.3.	Установка ПО Virtual I/O Server	130
4.4.	Базовый сценарий для VIOS	133
4.4.1.	Создание виртуального Ethernet-адаптера для VIOS	135
4.4.2.	Создание серверных виртуальных SCSI-адаптеров	137
4.4.3.	Создание клиентских разделов	140
4.4.4.	Создание виртуальных Ethernet-адаптеров для клиентских	
	разделов	142
4.4.5.	Создание виртуального SCSI-адаптера для клиентских	
	разделов	143
4.4.6.	Определение групп томов и логических томов	143
4.4.7.	Создание общего Ethernet-адаптера (SEA)	146
4.4.8.	Установка AIX 5L в клиентском разделе	148
4.4.9.	Зеркалирование rootvg сервера VIOS	150
4.5.	Взаимодействие с клиентскими UNIX-разделами	151
4.5.1.	Виртуальные SCSI-сервисы	152
4.5.2.	Виртуальные Ethernet-ресурсы	155
Глава 5. Установка Virtual I/O: расширенная конфигурация	157	
5.1.	Обеспечение повышения доступности VIOS	158
5.1.1.	Обеспечение повышения доступности путем увеличения	
	количества Virtual I/O Server	158
5.1.2.	Использование агрегирования каналов или EthernetChannel	
	для внешних сетей	161
5.1.3.	Обеспечение повышения доступности для связи	
	с внешними сетями	163
5.1.4.	Управление системой на Virtual I/O Server	172
5.1.5.	Реализация виртуального Ethernet в гипервизоре POWER	174
5.1.6.	Замечания о производительности Virtual I/O Server	175
5.1.7.	Достионства виртуального Ethernet и общих	
	Ethernet-адаптеров	176
5.1.8.	Ограничения и соглашения	178
5.2.	Сценарий 1: Зеркалирование логического тома	179
5.3.	Сценарий 2: Перехват SEA	182
5.4.	Сценарий 3: MPIO на клиенте с SAN в VIOS	187

5.4.1.	Настройка HMC	189
5.4.2.	Настройка Virtual I/O Server	195
5.4.3.	Работа с MPIO на клиентских разделах	199
5.5.	Запуск установки Linux на клиенте VIO	202
5.6.	Поддерживаемые конфигурации	202
5.6.1.	Поддерживаемые конфигурации VSCSI	203
5.6.3.	HACMP для клиентов виртуального ввода-вывода	212
5.6.4.	General Parallel Filesystem (GPFS)	217
Глава 6. Управление системой	219	
6.1.	Динамические операции с LPAR	220
6.1.1.	Динамическое удаление памяти	220
6.1.2.	Динамическое удаление виртуальных адаптеров	222
6.1.3.	Динамическое удаление процессоров	222
6.1.4.	Динамическое добавление адаптеров	224
6.1.5.	Динамическое добавление памяти	225
6.1.6.	Просмотр топологии на HMC	228
6.2.	Резервное копирование и восстановление сервера Virtual I/O Server	229
6.2.1.	Резервное копирование Virtual I/O Server	229
6.2.2.	Резервное копирование на ленту	230
6.2.3.	Резервное копирование на DVD	230
6.2.4.	Резервное копирование на файловую систему	231
6.2.5.	Восстановление сервера Virtual I/O Server	231
6.3.	Пересоздание сервера Virtual I/O Server	234
6.3.1.	Пересоздание конфигурации SCSI	235
6.3.2.	Пересоздание конфигурации сети	236
6.4.	Обслуживание сервера Virtual I/O Server	237
6.4.1.	Параллельные обновления ПО VIOS	237
6.4.2.	Устройства с горячей заменой	245
6.4.3.	Восстановление после сбоя диска на VIOS	248
6.4.4.	Дополнительные соображения и рекомендации по обслуживанию	250
6.5.	Мониторинг виртуализованной среды	252
6.5.1.	Process Utilization Resource Register (PURR)	253
6.5.2.	Общесистемные инструменты, модифицированные для виртуализации	256
6.5.3.	Команда topas	257
6.5.4.	Новые команды мониторинга	260
6.5.5.	Команда mpstat	264
6.5.6.	Мониторинг с помощью PLM	267
6.5.7.	Performance Workbench	268
6.5.8.	Команда nmon	268
6.5.9.	AIX Performance Toolbox	270
6.5.10.	Осведомленность о динамической реконфигурации	270
6.6.	Соображения по подбору количества ресурсов	271
6.6.1.	Соображения по конфигурации разделов	272
6.6.2.	Виртуализация и приложения	273
6.6.3.	Управление ресурсами	273

Глава 7. Partition Load Manager	275
7.1. Введение в Partition Load Manager	276
7.1.1. Режимы работы PLM	276
7.1.2. Модель управления	276
7.1.3. Политики управления ресурсами	278
7.1.4. Управление памятью	280
7.1.5. Управление процессором	281
7.1.6. Resource Monitoring and Control (RMC)	281
7.2. Установка и настройка Partition Load Manager	282
7.2.1. Подготовка AIX 5L для PLM	283
7.2.2. Установка и настройка SSL и SSH	284
7.2.3. Настройка RMC для PLM	288
7.2.4. Установка Partition Load Manager	289
7.2.5. Определение групп разделов и политик	289
7.2.6. Базовая настройка PLM	294
7.2.7. Интерфейс командной строки Partition Load Manager	307
7.3. Рекон��фигурация по расписанию	311
7.3.1. Рекон��фигурация раздела	311
7.3.2. Рекон��фигурация политики PLM	314
7.4. Советы и поиск неисправностей PLM	314
7.4.1. Поиск неисправностей соединения SSH	314
7.4.2. Поиск неисправностей соединения RMC	316
7.4.3. Поиск неисправностей на сервере PLM	320
7.5. Соглашения и ограничения PLM	322
7.6. Управление ресурсами	322
7.6.1. Управление ресурсами и рабочей нагрузкой	323
7.6.2. Как оценивается загрузка	325
7.6.3. Управление ресурсами ЦП	327
7.6.4. Управление ресурсами памяти	328
7.6.5. Какой инструмент управления ресурсами использовать?	328
Приложение А.Характеристики надежности, готовности и ремонтопригодности (RAS) System p5	329
Приложение В.Системные вызовы confer и cede гипервизора POWER	333
Аббревиатуры и акронимы	335

Рисунки

1-1.	Общий вид платформы IBM Virtualization Engine	3
1-2.	Технологии виртуализации, реализованные в серверах System p5 ...	4
1-3.	Компоненты резервирования в виртуализированной системе	9
1-4.	Уровни программного обеспечения между аппаратными ресурсами и операционными системами	13
1-5.	Консолидированные разделы i5/OS и UNIX в системе p5	14
2-1.	Примерный сценарий с обычным подходом	20
2-2.	Консолидация бизнес-приложений в системе p5	22
2-3.	Технические подробности консолидации в системе p5	23
2-4.	Сводка дополнительных преимуществ решения на базе системы с p5	24
2-5.	Динамическая реконфигурация системных ресурсов в системе А с процессорами p5	25
2-6.	Сценарий перемещения или восстановления приложения	25
2-7.	Система с новыми разделами из другой системы	26
2-8.	Пример восстановления после отказа в предложенной конфигурации	28
2-9.	Конфигурация двух систем с совместными кластерами	28
2-10.	Пример восстановления с перемещением всей системы на другую систему	29
2-11.	Сценарий с системой для восстановления бизнес-операций	30
2-12.	Восстановление системы А перемещением в систему В	30
3-1.	Окно HMC для активации Virtualization Engine Technologies	38
3-2.	Меню ASMI для активации Virtualization Engine Technologies	39
3-3.	Распределение выделенной мощности между виртуальными процессорами	46
3-4.	Разделы с общим процессором и верхним пределом	47
3-5.	Раздел с общим процессором и без верхнего предела	47
3-6.	Физические, виртуальные и логические процессоры	55
3-7.	Панель SMIT SMT с опциями	56
3-8.	Абстрагирование гипервизором POWER физического серверного оборудования	58
3-9.	Установление связей виртуальных и физических процессоров: проход 1 и проход 2	59
3-10.	Диспетчеризация процессора с микроразделами	60
3-11.	Границы лицензирования ПО по количеству процессоров	66
3-12.	Пример первоначального планирования лицензирования	68

3-13.	Роль IBM Tivoli License Manager в установлении соответствия мощности	70
3-14.	Пример VLAN	78
3-15.	Метка VID добавляется в расширенный Ethernet-заголовок	80
3-16.	АдAPTERы и интерфейсы с сетями VLAN (слева) и LA (справа)	83
3-17.	Соединение с внешней сетью с помощью маршрутизации	85
3-18.	Общий Ethernet-адаптер	87
3-19.	Пример конфигурации VLAN	89
3-20.	Добавление виртуальных Ethernet-адаптеров в VIOS для сетей VLAN	91
3-21.	Обзор архитектуры виртуального SCSI	95
3-22.	Логический удаленный прямой доступ к памяти	96
3-23.	Взаимоотношения устройств виртуального SCSI в VIOS	97
3-24.	Взаимоотношения устройств виртуального SCSI в клиентском разделе AIX 5L	98
3-25.	Общий вид Partition Load Manager	102
3-26.	Конфигурирование Integrated Virtualization Manager	104
3-27.	Реализация MPIO в VIO-клиенте и в VIO-сервере	110
3-28.	Реализация зеркалирования в VIO-клиенте и в VIO-сервере	111
3-29.	Связывание мостом виртуального и физического Ethernet-адаптеров с Linux	112
4-1.	Вид аппаратной консоли (Hardware Management Console)	122
4-2.	Запуск мастера Create Logical Partition Wizard	122
4-3.	Определение идентификатора и имени раздела	123
4-4.	Пропуск группы управления нагрузкой	124
4-5.	Определение имени профиля раздела	125
4-6.	Настройки памяти разделов	125
4-7.	Использование общего процессора	126
4-8.	Настройки общего процессора	126
4-9.	Режим общего процессора и настройки виртуальных процессоров	127
4-10.	Выбор физических компонентов ввода-вывода	128
4-11.	Настройки пула ввода-вывода	128
4-12.	Пропуск определения виртуальных адаптеров ввода-вывода	129
4-13.	Пропуск настроек для разделов управления питанием	129
4-14.	Выбор настройки режима загрузки	130
4-15.	Общий вид настроек раздела	131
4-16.	Окно состояния	131
4-17.	Теперь в окне показан новый созданный раздел VIO_Server1	132
4-18.	Активация раздела VIO_Server1	132
4-19.	Выбор профиля	133
4-20.	Выбор режима загрузки SMS	133
4-21.	Меню SMS	134
4-22.	Сценарий создания базового VIOS	134

4-23.	Добавление виртуального Ethernet	135
4-24.	Вкладка виртуального Ethernet	136
4-25.	Свойства виртуального Ethernet	136
4-26.	Вкладка нового виртуального Ethernet-адаптера	137
4-27.	Вкладка свойств SCSI	138
4-28.	Вкладка свойств VIOS SCSI	139
4-29.	Свойства слота серверного адаптера	139
4-30.	Создание раздела NIM_server	140
4-31.	Вид НМС с созданными новыми разделами	141
4-32.	Свойства виртуального Ethernet	142
4-33.	Задание свойств клиентского виртуального SCSI-адаптера	143
4-34.	Активация раздела DB_server	149
4-35.	Базовая последовательность конфигурирования виртуальных SCSI-ресурсов	152
4-36.	Базовая последовательность конфигурирования виртуальных SCSI-ресурсов	153
4-37.	Шаги по активации виртуального SCSI-сервиса для клиентского раздела с AIX 5L	154
4-38.	Последовательность шагов, необходимых для активации подключения к виртуальному Ethernet	155
5-1.	MPIO и зеркалирование для двух VIOS	160
5-2.	Дублирование общего Ethernet-адаптера	161
5-3.	Агрегирование каналов (EtherChannel) на AIX 5L	163
5-4.	Резервирование сетевого интерфейса (NIB) с двумя VIOS	165
5-5.	Многопутевой IP в клиенте, использующий два SEA различных VIOS	166
5-6.	Перехват маршрутизатора	167
5-7.	Базовая настройка SEA	168
5-8.	Альтернативная настройка восстановления SEA после ошибок	170
5-9.	Резервирование сетевого интерфейса (NIB) для нескольких клиентов	171
5-10.	Избыточные виртуальные серверы ввода-вывода во время проведения обслуживания	173
5-11.	Логический вид VLAN между разделами	174
5-12.	Отдельные серверы виртуального ввода-вывода для каждого ресурса	177
5-13.	Сценарий зеркалирования LVM	179
5-14.	Выбор физических компонент для VIO_Server2	180
5-15.	Свойства Virtual SCSI-раздела VIO_Server2	181
5-16.	Настройка повышения доступности адаптера SEA	183
5-17.	Динамическая операция с LPAR по добавлению виртуального Ethernet на VIO_Server1	184
5-18.	Слоты виртуального Ethernet и знание ID для Virtual LAN (PVID)	185
5-19.	Присоединение SAN к нескольким Virtual I/O Server	188

5-20.	Общий вид настройки DS4200	189
5-21.	Начальные настройки сценария	190
5-22.	Окно свойств профиля локального раздела	191
5-23.	Virtual SCSI: окно свойств адаптера сервера	191
5-24.	Общий вид SCSI-адаптера сервера на VIO_Server_SAN1	192
5-25.	Внешний вид SCSI-адаптера сервера на VIO_Server_SAN2	192
5-26.	Virtual SCSI: окно свойств клиентского адаптера	193
5-27.	Внешний вид адаптера SCSI клиента на разделе APP_server	193
5-28.	Внешний вид SCSI-адаптера клиента на разделе DB_server	194
5-29.	Установленный SUSE-раздел	203
5-30.	Поддерживаемый и рекомендуемый способы зеркалирования виртуальных дисков	204
5-31.	Конфигурация RAID5, использующая RAID-адаптер на Virtual I/O Server	205
5-32.	Рекомендуемый способ зеркалирования виртуальных дисков для двух VIOS	206
5-33.	Использование MPIO на Virtual I/O Server с IBM TotalStorage	208
5-34.	Использование RDAC на Virtual I/O Server с IBM TotalStorage	209
5-35.	Конфигурация для нескольких Virtual I/O Server и IBM ESS	210
5-36.	Конфигурация для нескольких Virtual I/O Server и IBM FASST	211
5-37.	Настройка резервного сетевого интерфейса	212
5-38.	Конфигурация перехвата SEA	213
5-39.	Базовые задачи хранения на клиентских разделах AIX 5L и HACMP	214
5-40.	Пример кластера HACMP между двумя клиентскими разделами AIX 5L	216
5-41.	Пример клиентского раздела AIX 5L с HACMP, использующего два VIOS	217
6-1.	Начальное окно динамической операции с LPAR	220
6-2.	Динамическое удаление 256 МБ памяти	221
6-3.	Окно статуса	221
6-4.	Окно динамических операций с виртуальными адаптерами	222
6-5.	Динамическая операция с вычислительными единицами ЦП	223
6-6.	Динамическая операция над LPAR для удаления 0.1 вычислительной единицы	223
6-7.	Окно динамического добавления виртуальных адаптеров	224
6-8.	Окно Virtual SCSI client adapter properties	225
6-9.	Окно динамического создания серверного адаптера	226
6-10.	Окно после создания серверного адаптера virtual SCSI	226
6-11.	Динамическая операция с памятью LPAR	227
6-12.	Дополнительные 256МБ памяти будут добавлены динамически	227
6-13.	Выбор просмотра топологии Virtual I/O	228
6-14.	Идет динамическая операция с LPAR	228
6-15.	Топология виртуальных SCSI-адаптеров на VIOS	229

6-16.	Выбор ленточного устройства для восстановления Virtual I/O Server	232
6-17.	Конфигурация для параллельного обновления ПО	238
6-18.	Свойства профайла на HMC	240
6-19.	Окно свойств LPAR	241
6-20.	PURR на уровне нитей	254
6-21.	Performance Workbench: окно Procmon	268
6-22.	Экран LPAR команды nmon	269
7-1.	Архитектура PLM	277
7-2.	Пороговые значения загрузки ресурсов	278
7-3.	Распределение ресурсов PLM	280
7-4.	Сервер управления RMC и управляемые разделы	283
7-5.	Шаги, необходимые для настройки PLM	294
7-6.	Стартовое окно Partition Load Manager	295
7-7.	Закладка General окна Create Policy File мастера PLM	296
7-8.	Закладка Globals окна Create Policy File мастера PLM	297
7-9.	Окно Group Definitions мастера PLM	297
7-10.	Закладка tunables окна Add Group of Partitions мастера PLM	298
7-11.	Окно wizard group definition summary мастера PLM	299
7-12.	Окно add managed partition мастера PLM	299
7-13.	Окно partition resource entitlement мастера PLM	300
7-14.	Окно мастера PLM с заполненной информацией о разделах	301
7-15.	Окно настройки коммуникаций PLM с управляемыми разделами ...	302
7-16.	Запуск сервера PLM	303
7-17.	Мастер PLM: окно Edit Policy File	304
7-18.	Диалог PLM для добавления группы разделов в существующий файл политики	305
7-19.	Диалог Add Managed Partition мастера PLM	305
7-20.	Диалог Resource Entitlements мастера PLM.	306
7-21.	Итоговая информация о разделах в окне Edit Policy File	306
7-22.	Мастер PLM: установка настроек группы	307
7-23.	Окна HMC Configuration и Schedule Operations	311
7-24.	Окно Customize Scheduled Operations	312
7-25.	Окно dd a Scheduled Operation	312
7-26.	Закладка Date and Time окна Setup a Scheduled Operation	313
7-27.	Закладка Repeat окна Set up a Scheduled Operation	313
7-28.	Закладка Options окна Set up a Scheduled Operation	314
7-29.	Меню Customize Network Setting: выбор закладки LAN Adapters	315
7-30.	Окно Firewall Settings	315
7-31.	Пример конфигурации для аутентификации PLM RMC	320
7-32.	Механизмы управления ресурсами и рабочей нагрузкой	323

Таблицы

1-1.	Операционные системы, поддерживаемые виртуализованной системой System p5	10
1-2.	Операционные системы, поддерживаемые для функций APV	15
1-3.	Сравнение возможностей виртуализации систем IBM System z9 и System p5	16
2-1.	Преимущества систем с APV для восстановления приложений	27
3-1.	Обзор кодов функции APV	38
3-2.	Обзор Micro-Partitioning	41
3-3.	Разумные настройки для разделов с разделяемыми процессорами ..	51
3-4.	Характеристики лицензирования отдельного ПО IBM	70
3-5.	Оценка лицензирования для первоначального приобретения процессорных лицензий	72
3-6.	Пример лицензирования для установленной системы	74
3-7.	Обмен между разделами VLAN	90
3-8.	Обмен VLAN с внешней сетью	92
3-9.	Модули ядра для виртуальных устройств IBM System p5	108
4-1.	Сетевые настройки	147
5-1.	Основные различия между EC- и LA-агрегированием	162
5-2.	Резюме касательно способов повышения доступности для доступа к внешним сетям	172
5-3.	Указание SCSI-адаптера клиента для раздела SB_server	194
5-4.	Сетевые настройки	195
5-5.	Минимальные уровни программного обеспечения для настройки НАСМР с APV	214
6-1.	Опции команды mpstat	264
6-2.	Интерпретация вывода команды mpstat	265
7-1.	Конфигурационные параметры PLM	290
7-2.	Настройки, связанные с ЦП	292
7-3.	Настройки, связанные с виртуальными процессорами	292
7-4.	Настройки, связанные с памятью	293
7-5.	Начальная конфигурация разделов в пуле	294
7-6.	Начальная конфигурация разделов с выделенными разделами	304
7-7.	Сводные данные функций RAS для различных моделей System p5 ...	330
7-8.	Поддержка операционными системами отдельных функций RAS ...	330

Примечания

Информация в этой книге охватывает продукцию и услуги, предлагаемые в США. Предложения IBM по услугам, товарам и их возможностям, описанным в данной книге, могут не действовать в других странах. За информацией о текущем ассортименте доступных продуктов и услуг обращайтесь в местные представительства IBM. Явные и неявные упоминания услуг, продуктов и их возможностей не означают необходимости их применения. Допускается их замена любыми функционально эквивалентными продуктами и службами сторонних производителей, не нарушающими права на интеллектуальную собственность IBM, при этом ответственность за проверку совместимости и продуктивности решений сторонних производителей принимает на себя пользователь.

IBM может обладать патентами или патентными заявками на технологии, описанные в настоящей книге, предоставление которых не означает наличия лицензии на технологии. Письменные запросы лицензий следует направлять по адресу *IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 USA*.

Приведенный ниже абзац не относится к Соединенному Королевству и иным странам, законодательству которых противоречит данное положение: КОРПОРАЦИЯ IBM ПРЕДОСТАВЛЯЕТ ДАННУЮ КНИГУ «КАК ЕСТЬ» И НЕ ДАЕТ НИКАКИХ ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ (НО НЕ ОГРАНИЧИВАЯСЬ) ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ ПАТЕНТНОЙ ЧИСТОТЫ, КОММЕРЧЕСКОЙ ПРИГОДНОСТИ И СООТВЕТСТВИЯ КОНКРЕТНОМУ НАЗНАЧЕНИЮ. В отдельных государствах отказ от явных и подразумеваемых гарантий по ряду сделок запрещен, так что указанное ограничение может вас не коснуться. В книге возможны опечатки и технические неточности. Приводимые в ней сведения регулярно обновляются, соответствующие изменения будут внесены в новую редакцию книги. Корпорация IBM оставляет за собой право в любое время без уведомления модифицировать описанные в этой книге продукты и программные средства.

Любые ссылки на сайты сторонних компаний в этой книге носят исключительно информационный характер и не свидетельствуют об их поддержке корпорацией IBM; риск, связанный с применением ресурсов этих сайтов, принимает на себя пользователь.

По своему усмотрению и без каких-либо обязательств IBM может использовать и распространять любые предоставленные сведения.

Все содержащиеся здесь рабочие параметры были определены в контролируемой среде. Таким образом, получаемые в других средах результаты могут варьироваться в значительной степени. Некоторые измерения могли выполняться в системах на стадии разработки, и поэтому получение результатов, одинаковых с этими измерениями, не гарантируется для серийно выпускаемых систем. Более того, некоторые измерения могли быть получены посредством экстраполяции.

Действительные результаты могут от них отличаться. Пользователям данной книги следует проверять пригодность информации для их конкретной среды.

Информация о сторонних продуктах получена от соответствующих поставщиков, из их рекламных публикаций и других открытых источников. IBM не тестировала эти продукты и не может подтвердить точность заявленных рабочих параметров, совместимости и других характеристик. Вопросы о возможностях сторонних продуктов следует направлять соответствующим поставщикам.

Настоящая книга содержит в качестве примеров данные и отчеты, используемые в повседневной практике предприятий. Для наиболее полной иллюстрации примеров в книге встречаются имена лиц, названия компаний, торговых марок, товаров. Все они вымышлены, и любые совпадения с именами и данными реально существующих компаний случайны.

Авторское право

Данная книга содержит примеры исходного кода прикладных программ, которые иллюстрируют приемы программирования на различных платформах. Вы можете копировать, изменять и распространять их в любом виде без отчислений в пользу IBM, руководствуясь целями разработки прикладных программ, включая коммерческие, в соответствии с интерфейсом прикладного программирования платформы, для которой предназначены эти программы. Примеры не подвергались всеобъемлющему тестированию, поэтому IBM не может явно или неявно гарантировать надежность, удобство в обслуживании и работоспособность приведенных примеров. Вам разрешено копировать, изменять и распространять программы-примеры, содержащиеся в данной книге, в любом виде без отчислений в пользу IBM с целью разработки, применения, в том числе коммерческого, и дистрибуции прикладных программ, сообразуясь с интерфейсами прикладного программирования корпорации IBM.

Товарные знаки

Ниже приведены товарные знаки корпорации IBM в США и (или) других странах:

AIX 5L™	IBM®	pSeries®
AIX®	iSeries™	PTX®
AS/400®	LoadLeveler®	Redbooks™
Chipkill™	Lotus®	Redbooks (logo)™
Domino®	Micro-Partitioning™	RS/6000®
DB2®	MQSeries®	System p5™
Enterprise Storage Server®	OpenPower™	System z9™
@server®	Parallel Sysplex®	Tivoli®
@server®	Passport Advantage®	TotalStorage®
Everyplace®	PowerPC®	TXSeries®
Geographically Dispersed	POWER™	Virtualization Engine™
Parallel Sysplex™	POWER4™	WebSphere®
GDPS®	POWER4+™	z/OS®
HiperSockets™	POWER5™	z/VM®
HACMP™	POWER5+™	z9™
i5/OS®	PR/SM™	zSeries®

Другим компаниям принадлежат следующие товарные знаки:

Java, SunOS и все товарные знаки, связанные с Java-приложениями, являются товарными знаками Sun Microsystems, Inc. в США и (или) других странах.

Excel, Microsoft, Windows, логотип Windows – товарные знаки Microsoft Corporation в США и (или) других странах.

UNIX – зарегистрированный товарный знак Open Group в США и (или) других странах.

Linux – товарный знак, принадлежащий Линусу Торвальдсу (Linus Torvalds) в США и (или) других странах.

Названия других компаний, товаров и услуг могут быть товарными знаками соответствующих владельцев.

Предисловие

Эта книга серии IBM® Redbook знакомит с технологией Advanced POWER™ Virtualization в системах IBM System p5™ и IBM @server p5.

Advanced POWER Virtualization (средства расширенной виртуализации) – это комплекс аппаратных и программных ресурсов, который обеспечивает поддержку и управление виртуальной средой в системах на базе процессоров POWER5™ и POWER5+™. Данное решение включает в себя следующие основные технологии:

- ▶ Виртуальные сетевые интерфейсы (Virtual Ethernet).
- ▶ Разделяемые сетевые адаптеры (Shared Ethernet Adapter).
- ▶ Сервер виртуальных дисков (Virtual SCSI Server).
- ▶ Технология микроразделов (Micro-Partitioning™).
- ▶ Средства управления нагрузкой (Partition Load Manager).

Главным преимуществом Advanced POWER Virtualization является увеличение общей утилизации системных ресурсов благодаря выделению только необходимых ресурсов ввода-вывода и процессоров каждому разделу.

Эта книга также может использоваться в качестве справочника системными администраторами серверов. В ней содержатся подробные инструкции для:

- ▶ Конфигурирования и создания разделов с помощью аппаратной консоли (HMC).
- ▶ Установки и конфигурирования сервера виртуального ввода-вывода (Virtual I/O Server).
- ▶ Создания виртуальных ресурсов для разделов.
- ▶ Установки разделов с виртуальными ресурсами.

Хотя материал этой книги сосредоточен на оборудовании систем IBM System p5 и операционной системе AIX® 5L™ (и применим также к системам IBM @server p5), ее основные концепции могут быть распространены на операционные системы i5/OS® и Linux®, а также на IBM @server i5 и платформу и OpenPower™.

Необходимы базовые знания в области логических разделов.

Данная публикация является полной переработкой книги *Advanced POWER Virtualization on IBM ^ p5 Servers: Introduction and Basic Configuration*, SG24-7940.

Группа, написавшая эту книгу

Данная книга серии Redbook была создана группой специалистов со всего мира, работающих в организации International Technical Support Organization в Austin Center.

Анника Бланк (Annika Blank) является ИТ-специалистом предпродажной технической поддержки в IBM System Sales в Гамбурге, Германия. У нее имеется шестилетний опыт работы с AIX, RS/6000® и IBM **@server** pSeries®. Она работает в IBM двенадцать лет. Ее опыт охватывает веб-технологии в системах pSeries и серверах уровня предприятия, поддержку продаж продукции IBM, бизнес-партнеров и клиентов IBM предпродажными консультациями и реализацию клиент-серверных сред.

Пол Кифер (Paul Kiefer) является менеджером по решениям Account Technical Solution в группе IBM Engagement Services в Сиднее, Австралия. У него восьмилетний опыт работы с AIX, RS/6000 и IBM **@server** pSeries. Он работает в IBM два года. Его опыт охватывает продажи решений IBM Global Service Solution с IBM **@server** pSeries и технологии IBM TotalStorage®.

Карлос Сальяве-мл. (Carlos Sallave Jr.) является администратором систем UNIX® компании IBM в Денвере, Колорадо, и в настоящее время работает в учетной системе Nissan North America. У него более 10 лет опыта работы с RS/6000, AIX, PSSP и HACMP™. У него диплом разработчика компьютерных систем (Computer Engineering), полученный в техническом колледже Don Bosco, Филиппины. Его опыт охватывает реализацию, установку и администрирование IBM **@server** pSeries и IBM System p5.

Герардо Валенсия (Gerardo Valencia) является ИТ-специалистом из Колумбии. У него 12-летний опыт работы с AIX и IBM **@server** pSeries. У него диплом разработчика компьютерных систем. Его опыт охватывает ИТ-архитектуру для бизнес-приложений, планирование объема ресурсов, обеспечение высокой доступности, предпродажную поддержку специалистов по продажам IBM и бизнес-партнеров IBM, а также консультации клиентов IBM.

Джез Вейн (Jez Wain) является старшим системным архитектором в группе Group Bull в Гренобле, Франция. Он работал с бортовыми авиационными системами реального времени, системами телеметрии и сбора данных, ядром AIX и J2EE. У него диплом инженера по производству (Production Engineering), полученный в университете Лафборо, Великобритания, и он имеет знания в области ИТ-инфраструктуры.

Армин М. Варда (Armin M. Warda) является ИТ-архитектором в компании Postbank Systems AG, являющейся ИТ-провайдером Deutsche Postbank AG в Германии. Более десяти лет назад он начал работать с SunOS™ и Linux. В настоящее время у него восьмилетний опыт работы в информационном центре с IBM RS/6000, IBM **@server** pSeries и IBM System p5, AIX, HACMP, Storage Systems, SAP R/3, Oracle, DB2® и с сетевыми системами. В последние несколько лет он сосредоточился на обеспечении высокой доступности, анализе производительности, безопасности UNIX, интеграции UNIX и майнфреймов, работе с разделами серверов и виртуализации. У него диплом по информатике (Computer Science), полученный в Дортмундском университете, Германия.

Проектом создания этой книги руководил:

Скотт Веттер (Scott Vetter)
IBM U.S.

Выражаем благодарность следующим людям за вклад в данный проект:

Боб Ковач, Джим Пафьюми, Джим Партидж, Пол Финли, Эдуардо Рейес, Джулия Крафт, Рей Андерсон, Стивен Ти, Скот Трен, Санкет Ратхи, Джая Срикришнан, Кейси Маккрири, Маркос Вильяреаль, Майк Молл, Дин Бердик, Люк Смолдерс, Серж Дюваль, Патрик Во, Васу Валлабханени, Ромни Уайт, Хорхе Р. Ногерас, Винит Джайн, Шон Бодили, Джим Митчелл, Дэниэл Хендерсон, Джордж Аренс, Томас Уайт, Дэвид Чейз, Дэйв Льюис, Вани Д. Рамагири, Анил К.

IBM U.S.

Найджел Гриффитс
IBM UK

Тьерри Юш, Николя Герен
IBM France

Федерико Ваньини
IBM Italy

Фолькер Хауг
IBM Germany

Станьте публикуемым автором

Примите участие в нашей программе с проживанием в течение двух–шести недель! Окажите помощь в написании очередной книги IBM серии Redbook, касающейся конкретных продуктов или решений, и получите практические знания в области передовых технологий. Вы окажетесь в группе технических профессионалов, бизнес-партнеров или клиентов IBM.

Ваша работа поможет повысить признание продукции и удовлетворение запросов клиентов. За это вы сможете установить сеть контактов с лабораториями по разработкам IBM и улучшить вашу продуктивность и положение на рынке.

Более подробно о программе с проживанием вы можете узнать, просмотрев индекс с предложениями условий проживания, и подать заявку через Интернет по адресу:

ibm.com/redbooks/residencies.html

Комментарии приветствуются

Ваши комментарии важны для нас!

Мы хотим, чтобы наши книги Redbook™ были максимально полезными. Присылайте нам ваши комментарии об этой и других книгах Redbook™ одним из следующих способов:

- ▶ С помощью онлайновой формы пересмотра **Contact us**, размещенной по адресу:
ibm.com/redbooks
- ▶ С помощью электронной почты по адресу:
redbook@us.ibm.com
- ▶ Почтой по адресу:
IBM Corporation, International Technical Support Organization
Dept. JN9B Building 905
11501 Burnet Road
Austin, Texas 78758-3493



1

Введение

Первое издание книги *Advanced POWER Virtualization on IBM @server p5 Servers: Introduction and Basic Configuration*, SG24-7940 (*Применение технологии Advanced POWER Virtualization в серверах IBM @server p5: введение и базовая конфигурация*) было опубликовано лишь год назад. С того времени компания IBM и ее партнеры накопили значительный опыт работы с этой технологией и получили отзывы от клиентов по использованию виртуализации в своих областях. В настоящем, втором издании книги учтены извлеченные уроки и «надстроен» основополагающий материал первого выпуска данной книги. В результате вам предоставляется новая книга серии Redbook с полной переработкой первоначальной ее версии.

Эта книга предназначена как для новичков в области виртуализации, так и для тех, кто уже проводил эксперименты с данной технологией. Она написана в практическом стиле и подтолкнет вас к немедленной работе с технологией. В ней расширен приведенный в первом издании набор базовых сценариев виртуализации, и, включены некоторые из более совершенных конфигураций, в частности предназначенных для улучшения доступности, использования диспетчера управления загрузкой (Partition Load Manager), выполнения задач администрирования и отслеживания системных ресурсов.

В остальной части этой главы приводится краткий обзор ключевых технологий виртуализации. В главе 2, «Ценность технологии Advanced POWER Virtualization» обсуждаются выгоды, получаемые от виртуализации. В главе 3, «Технологии Virtualization Engine в серверах System p5», подробно обсуждаются основные технологии, а также новые особенности сервера виртуального ввода-вывода Virtual I/O Server V1.2. Усвоение материала этой главы необходимо для понимания остальной части книги. В главе 4, «Установка Virtual I/O Server: базовая конфигурация», рассматриваются первоначальные конфигурации разделов, а в главе 5 «Установка Virtual I/O: расширенная конфигурация», рассказывается о таких более сложных системах, как подключенное к сети SAN хранилище данных и система серверов Virtual I/O Server. В главе 6, «Администрирование системы» описываются системное администрирование и новые средства мониторинга, а в последней главе 7, «Partition Load Manager» показано, как автоматизировать управление ресурсами с помощью PLM. Итак, читайте...

1.1. Решение Virtualization Engine компании IBM и модель предоставления ресурсов «по требованию»

Решение IBM Virtualization Engine™ (VE) обеспечивает как системные сервисы, так и системные технологии, помогающие достигать виртуализации в разнородной ИТ-среде.

Оно позволяет вам:

- ▶ Создать гибкую систему управляющих связей для виртуализации ресурсов по всей вашей ИТ-инфраструктуре с сервисами предоставления ресурсов и управления нагрузкой.
- ▶ Оптимизировать использование ресурсов с помощью таких технологий, как динамические логические разделы.
- ▶ Создать общее, согласованное и межплатформенное управление системами для серверов IBM, хранилища данных и операционных систем.
- ▶ Лучше оптимизировать использование ресурсов, снизить стоимость и сложность и вместе с тем обеспечить экономичные ИТ-решения для разнородных сред.
- ▶ Углубить интеграцию информационных технологий с деловой средой с помощью усовершенствованных технологий виртуализации и управляющих сервисов.

Модель вычислений «по требованию» (on-demand) компании IBM применима ко всем уровням стека информационных технологий предприятия. На уровне системы компонентами являются системные объекты (например, вычислительные мощности, хранилище данных и файлы). На уровне приложений компонентами являются динамически объединяемые прикладные модули, образующие хотя более сложные, но и более гибкие приложения. На уровне предприятия компонентами являются объекты деловой деятельности, определяемые вертикально для конкретной отрасли или, еще более широко, с учетом их горизонтального применения между отраслями.

Решение IBM Virtualization Engine, показанное на рисунке 1-1, состоит из комплекса системных сервисов и технологий, образующих ключевые элементы модели IBM вычислений «по требованию». В нем ресурсы отдельных серверов, хранилищ данных и сетевых продуктов считаются принадлежащими единому пулу, что позволяет более эффективно обеспечивать доступ к ресурсам и их управление в масштабе всего предприятия. Виртуализация является наиболее важным компонентом в операционной среде «по требованию», и системные технологии, реализованные в серверах IBM System p5 с процессорами POWER5, обеспечивают значительные преимущества при осуществлении функций, необходимых для работы в такой среде.

Нижеперечисленные технологии являются компонентами IBM Virtualization Engine, интегрированными в серверах System p5:

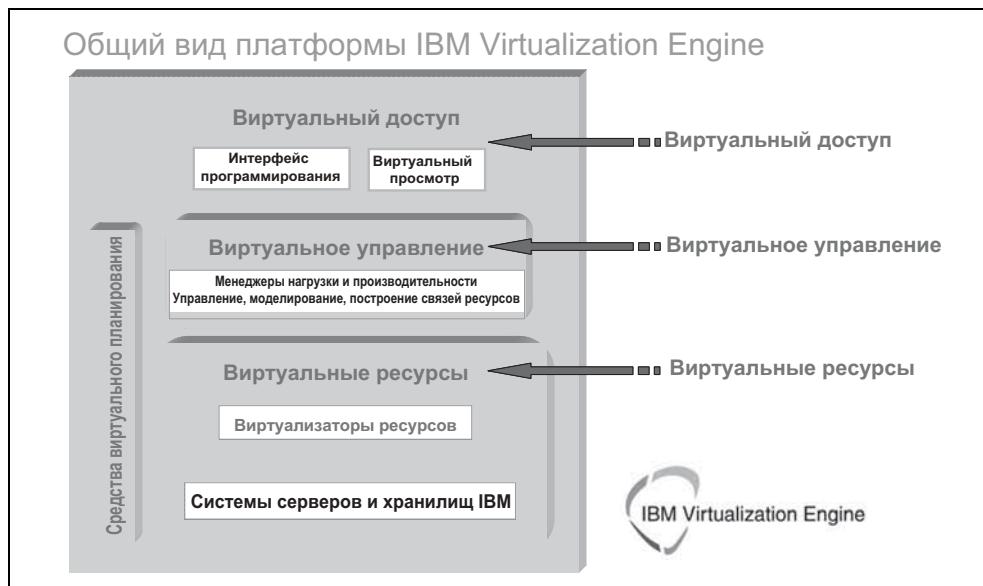


Рис. 1-1. Общий вид платформы IBM Virtualization Engine

Гипервизор POWER (POWER Hypervisor) – Поддерживает работу с разделами и динамическое перемещение ресурсов в средах с различными операционными системами.

Микроразделы (Micro-Partitioning) – Обеспечивают выделение логическому разделу доли в процентах от ресурса процессора вместо полного физического процессора.

Виртуальная сеть (Virtual LAN) – Предоставляет функции сетевой виртуализации.

Виртуальный ввод-вывод (Virtual I/O) – Позволяет совместно использовать адаптеры и устройства ввода-вывода различными разделами.

Наращивание ресурсов по требованию (Capacity Upgrade on Demand, CUoD) – Позволяет активировать такие системные ресурсы, как процессоры и память, по мере необходимости.

Параллельная многопоточная обработка (Simultaneous multithreading) – Реализованные на аппаратном уровне потоки, улучшающие использование ресурсов.

Поддержка нескольких операционных систем – Логические разделы позволяют одному серверу параллельно работать с образами нескольких операционных систем.

На рисунке 1-2 показано, как комбинация нескольких этих технологий может гибко обеспечивать ваши вычислительные потребности.

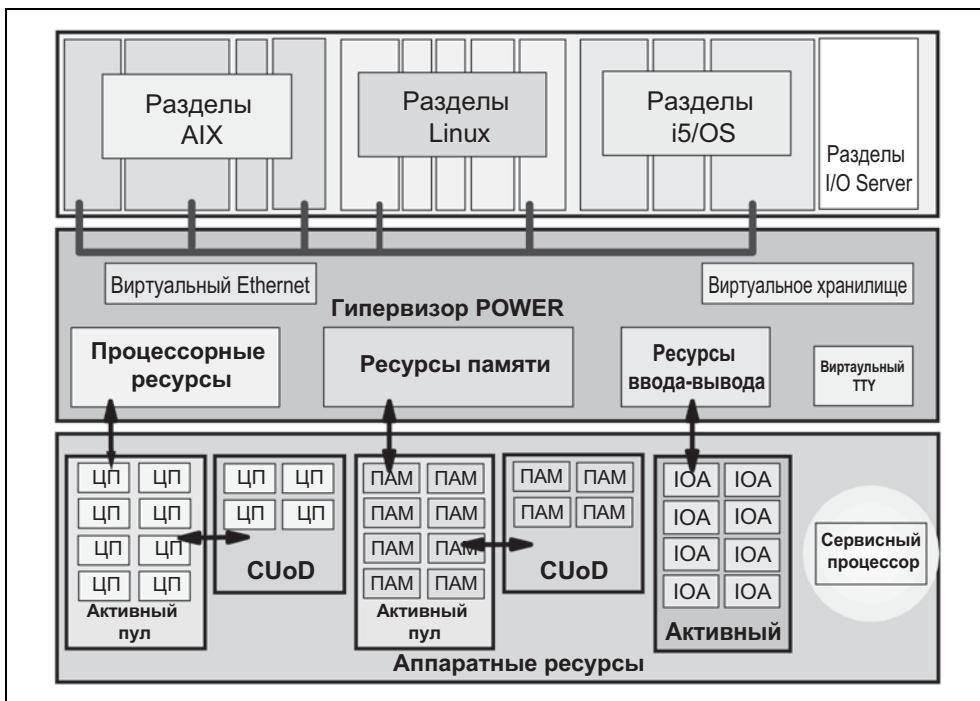


Рис. 1-2. Технологии виртуализации, реализованные в серверах System p5

1.2. Решение Virtualization Engine в системе IBM System p5

В этом разделе описываются системные технологии решения IBM System p5 Virtualization Engine, имеющиеся в серверах System p5 с процессорами POWER5.

1.2.1. POWER Hypervisor

Гипервизор POWER (POWER Hypervisor) является основой для виртуализации в сервере System p5. Этот компонент позволяет разделять аппаратные ресурсы на несколько разделов и обеспечивает их надежную взаимную изоляцию.

Находясь в постоянно активном состоянии на серверах с процессорами POWER5, гипервизор POWER отвечает за диспетчеризацию нагрузки логических разделов между физическими процессорами. POWER Hypervisor также обеспечивает безопасность разделов и может обеспечивать обмен между разделами, активируя функции виртуального SCSI и виртуального Ethernet-сервера Virtual I/O Server.

1.2.2. Технология параллельной многопоточной обработки (SMT)

Усовершенствования в дизайне процессора POWER5 повышают степень общего использования аппаратных ресурсов. В технологии SMT два отдельных потока инструкций (threads) параллельно выполняются на одном физическом процессоре, что увеличивает общую пропускную способность.

1.2.3. LPAR и разделы общего процессорного пул

Логический раздел LPAR (Logical Partition) не ограничивается границами физического процессора и может использовать процессорные ресурсы из общего пула процессоров (shared processor pool). Раздел LPAR, использующий процессорные ресурсы из общего пула процессоров, называется микроразделом (Micro-Partition LPAR)¹.

Ресурсы физического процессора выделяются в виде процентной доли от его мощности. Эта процентная доля называется «выделенная вычислительная мощность» (processor entitlement). Такая доля может составлять от десяти процентов мощности одного физического процессора до полной мощности всех процессоров, установленных в системе IBM System p5. Дополнительное выделение доли вычислительной мощности может производиться с точностью до одного процента мощности физического процессора.

1.2.4. Динамическая реконфигурация

Имеется возможность динамического перемещения системных ресурсов, физических процессоров, виртуальных процессоров, памяти и слотов ввода-вывода между разделами без перезагрузки. Это называется динамической реконфигурацией (dynamic reconfiguration, DR) или динамическим логическим разделом (DLPAR).

1.2.5. Virtual LAN

Одна из функций гипервизора POWER – виртуальная локальная сеть Virtual LAN обеспечивает безопасный обмен между логическими разделами без участия физического адаптера ввода-вывода. Возможность безопасно использовать пропускную способность Ethernet несколькими разделами повышает степень использования аппаратных ресурсов.

1.2.6. Virtual I/O

Функция виртуального ввода-вывода Virtual I/O обеспечивает возможность использования одного физического адаптера ввода-вывода и диска несколькими логическими разделами одного и того же сервера, позволяет консолидировать ресурсы ввода-вывода и минимизировать число необходимых адаптеров ввода-вывода.

1.2.7. Наращивание ресурсов «по требованию» (CUoD)

Есть несколько вариантов наращивания ресурсов «по требованию» CUoD (Capacity Upgrade on Demand), а именно:

- ▶ Permanent Capacity Upgrade – наращивание ресурсов на постоянной основе. Позволяет наращивать ресурсы системы без прерывания текущей работы посредством активации процессоров или памяти.
- ▶ On/Off Capacity Upgrade on Demand – включение-выключение ресурсов по требованию. Предусматривает оплату только используемых ресурсов по мере активации необходимых процессоров и памяти.

¹ Используется также термин «SPLPAR» – Shared Processor LPAR. Прим. науч. ред.

- ▶ Reserve Capacity Upgrade on Demand – подключение резервных ресурсов по требованию. Соглашение с предварительной оплатой, предусматривающее добавление резервного процессора к общему пулу процессоров, используемое в ситуациях, когда недостаточно вычислительной мощности общего пула.
- ▶ Trial Capacity Upgrade on Demand – пробное подключение ресурсов по требованию. Частичная или полная активация установленных процессоров или памяти на фиксированный период времени.

1.2.8. Поддержка нескольких операционных систем

Системы System p5 с процессорами POWER5 поддерживают IBM AIX 5L Version 5.2 ML2, IBM AIX 5L Version 5.3, i5/OS и дистрибутивы Linux компании SUSE и Red Hat.

1.2.9. Integrated Virtualization Manager

Integrated Virtualization Manager (IVM) является решением по управлению аппаратными ресурсами, унаследовавшим базовые функции аппаратной консоли Hardware Management Console (HMC) и избавленным от необходимости внешней HMC. Его возможности ограничены управлением одним сервером System p5. IVM работает на сервере Virtual I/O Server Version 1.2.

1.3. Характеристики RAS виртуализованных систем

Отдельные системы System p5 обладают возможностью содержать большее количество образов систем, и становится более важной изоляция и обработка вероятных отказов. Функции аппаратных ресурсов и операционных систем встроены в саму систему для того, чтобы отслеживать ее работу, предсказывать, где действительно могут произойти отказы, затем обрабатывать условия их возникновения и, по возможности, продолжать работу. Инженеры компании IBM, занимающиеся вопросами надежности, доступности и обслуживаемости (reliability, availability, serviceability – RAS), постоянно совершенствуют конструкцию серверов, чтобы серверы System p5 поддерживали высокий уровень обнаружения одновременных сбоев, изоляции неисправностей, восстановления и доступности.

1.3.1. Надежность, доступность и обслуживаемость

Целью RAS системы IBM является минимизация простоев. В этом разделе объясняются конкретные возможности RAS, как новые, так и улучшенные, в продуктах System p5.

Уменьшение и исключение простоев

Базовая надежность вычислительной системы, на самом нижнем уровне, зависит от частоты отказов ее внутренних компонентов. Наиболее надежные серверы создаются из самых надежных компонентов. В серверах System p5 предусмотрено резервирование с помощью нескольких системных компонентов и механизмов диагностики и обработки конкретных ситуаций, ошибок или отказов на уровне компонентов. Подробнее об этом см. Приложение А «Характеристики RAS систем System p5».

Наблюдение за системой

Обнаружение и изоляция отказов являются ключевыми элементами сервера с высокой доступностью и обслуживаемостью. Совместная работа трех элементов систем System p5 – гипервизора POWER, сервисного процессора (service processor, SP) и консоли Hardware Management Console (HMC) – обеспечивает обнаружение сбоев аппаратных ресурсов, уведомление о них и управление ими.

Сервисный процессор

Сервисный процессор активирует наблюдение с помощью гипервизора POWER и аппаратной консоли, дистанционное управление питанием, отслеживание параметров окружающей среды, функции сброса и загрузки, дистанционное обслуживание и диагностику, включая зеркалирование консоли. В системах без HMC на сервисном процессоре могут размещаться запросы для уведомления об отказах от системы наблюдения с гипервизора POWER, критические состояния среды и критические отказы обработки.

Сервисный процессор обеспечивает следующие сервисы:

- ▶ Мониторинг среды.
- ▶ Общее наблюдение с помощью гипервизора POWER.
- ▶ Самозащиту системы с перезапуском при неисправимом сбое в микропрограммном коде (firmware), зависании микропрограмм, аппаратных отказах или отказах, вызванных окружающей средой.
- ▶ Мониторинг отказов и уведомление операционной системы при загрузке системы.

Характеристики RAS гипервизора POWER

Элементы гипервизора POWER используются для управления обнаружением и устранением некоторых видов ошибок. Гипервизор POWER обменивается данными с сервисным процессором и консолью Hardware Management Console.

Предсказание отказов

Компания IBM создала серверы, работающие с «диагностикой по информации первого отказа» First Failure Data Capture (FFDC), в которые встроены тысячи элементов диагностики аппаратных сбоев, фиксирующих условия отказов и помогающих их идентифицировать в самом сервере. Это позволяет системам System p5 выполнять самодиагностику и самовосстановление. Каждое из этих средств контроля представляет собой расположенный в сервере диагностический датчик, который в сочетании с хорошо разработанными диагностическими микропрограммами позволяет оценивать условия аппаратного сбоя в динамическом режиме.

Благодаря использованию в системах System p5 технологии «диагностики по информации первого отказа» значительно снизилась необходимость воссоздания условий отказов центрального процессорного комплекса (central electronic complex, CEC) при их диагностике. Сервисный процессор, работающий совместно с технологией FFDC, обеспечивает автоматическое обнаружение и изоляцию отказов без необходимости воссоздавать условия их возникновения. Это означает, что отказы будут правильно обнаруживаться и изолироваться в момент их возникновения.

Улучшение сервиса

Программное обеспечение НМС содержит возможность улучшенного сервиса и поддержки, включая автоматическую установку и модернизацию, а также параллельное обслуживание и модернизацию аппаратных ресурсов и микропрограммных прошивок. НМС также обеспечивает «фокусную сервисную точку» (Service Focal Point) для получения сервиса, регистрации, отслеживания системных сбоев и, при наличии возможности, для ретрансляции отчетов о проблеме в сервисную службу IBM.

Для систем System p5 проблемы с такими компонентами, как источники питания, вентиляторы, диски, НМС-консоли, PCI-адAPTERЫ и другие устройства, могут быть устранены без выключения системы. НМС поддерживает много новых функций параллельного обслуживания в системах System p5.

Планируется выпуск микрокода для систем System p5 в кумулятивном инкрементальном формате обновлений для применения и активации в параллельном режиме. Целью является возможность установки и активации большинства обновлений микрокода без отключения питания или перезагрузки системы.

Более подробную информацию о RAS систем System p5 можно получить в официальном документе *IBM System p5: a Highly Available Design for Business-Critical Applications*, находящемся по адресу:

http://www.ibm.com/servers/@server/pseries/library/wp_lit.html

За информацией об инструментах сервиса и повышения продуктивности для Linux на POWER обращайтесь по адресу:

<http://techsupport.services.ibm.com/server/lopdiags>

1.3.2. Доступность и обслуживаемость в виртуализованных средах

В средах с разделами, где все больше критически важных для бизнеса приложений консолидируются с различными операционными системами на одних и тех же аппаратных ресурсах, требуется повышенная доступность и обслуживаемость. Это должно обеспечить «гладкое» восстановление после единичных отказов и позволить большинству приложений сохранять работоспособность, когда одна из операционных систем выходит из строя. Более того, функции высокой доступности на уровнях операционных систем и приложений необходимы для быстрого восстановления сервиса для конечных пользователей.

В системах System p5 имеется ряд механизмов, которые повышают общую доступность системы и приложений с помощью комбинирования аппаратных ресурсов, построения самой системы и кластеризации.

Динамическое высвобождение и резервирование процессоров

Гипервизор POWER будет высвобождать отказавшие процессоры и заменять их процессорами из неиспользуемых ресурсов CUoD при их наличии. Эта технология называется Dynamic processor deallocation and sparing.

Резервирование памяти

Если при подаче питания сервисный процессор обнаружит отказавшую память в сервере, содержащем память CUoD, то гипервизор POWER заменит эту память неиспользуемой памятью из пула CUoD. Если резервной памяти нет, то гипервизор POWER уменьшит ресурсы одного или нескольких разделов. В IBM @server p5 моделей 590 и 595 механизм резервирования памяти будет применять неис-

пользуемую память CUoD для замены в случае всех общих отказов карт памяти. В IBM **@server** p5 модели 570 механизм резервирования памяти будет применять память CUoD для замены в случае отказа не более одной карты памяти.

Узнавайте у вашего представителя IBM о наличии данной функции в серверах IBM System p5.

Резервирование адаптеров

Резервирование адаптеров может осуществляться с помощью сохранения комплекса PCI-адаптеров восстановления в качестве глобального резерва для операций динамического реконфигурирования в случае отказов адаптеров. Включайте в конфигурацию резервные PCI-адAPTERЫ в разных разделах системы, в том числе и в серверах Virtual I/O Server, чтобы они были настроены и готовы к использованию.

Резервные серверы Virtual I/O

Так как раздел AIX 5L или Linux может быть одновременно клиентом одного и более серверов виртуального ввода-вывода Virtual I/O Server, то правильной стратегией улучшения доступности наборов клиентских разделов AIX 5L или Linux является подключение их к двум серверам Virtual I/O Server. Этот метод обеспечивает избыточную конфигурацию для каждого соединения клиентского раздела с внешней сетью Ethernet или ресурсами хранилища данных. В этой книге обсуждаются такие конфигурации.

На рисунке 1-3 показано несколько механизмов, используемых для улучшения доступности и обслуживаемости в сервере System p5 с разделами.

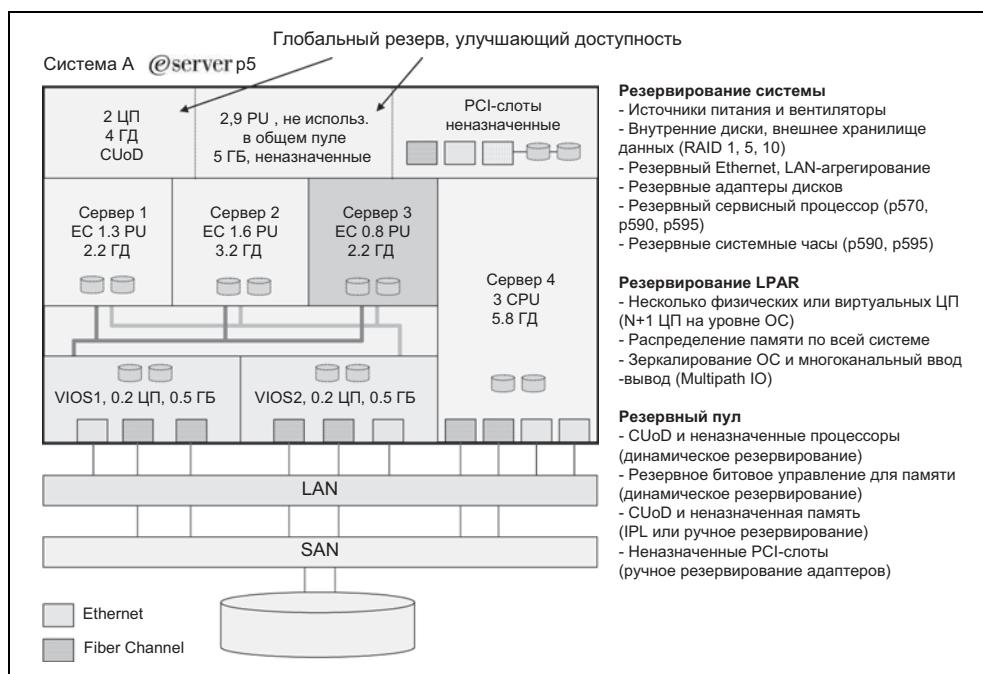


Рис. 1-3. Компоненты резервирования в виртуализованной системе

1.4. Безопасность в виртуализованной среде

Характеристики логических разделов, реализованных в системах IBM *server pSeries* с процессорами POWER4™ или POWER4+™, основаны на поддержании самых высоких уровней безопасности. Ресурсы разделов изолированы друг от друга, и между разделами почти нет совместного использования ресурсов или взаимодействия (клиенты могут конфигурировать разделы без обмена данными между разделами).

Архитектурные усовершенствования при создании платформы POWER5 делают возможным общее использование ресурсов разделами и взаимодействие между ними. Новые функции виртуализации поддерживают требования системной безопасности. Возможности взаимодействия между разделами не уменьшают безопасность ниже того уровня, какой подразумевает данная функция. Например, виртуальное сетевое соединение будет иметь такую же безопасность, как и физическое сетевое соединение.

1.5. Поддержка операционных систем

Системы System p5 предназначены для поддержки самых различных типов и версий операционных систем и позволяют клиентам консолидировать разнородные среды. В таблице 1-1 перечислены операционные системы, поддерживаемые решением Advanced POWER Virtualization (APV).

Таблица 1-1. Операционные системы, поддерживаемые виртуализованной системой System p5

Операционная система	Клиент функций APV	Сервер для функций APV
IBM AIX 5L V5.2 ML2 или более поздней версии		
IBM AIX 5L V5.3	X	
VIOS V1.1 и V1.2 X		X
IBM i5/OS	X	
RHEL AS V3 update 3 или более позднее	X	X ^{a,b}
RHEL AS V4	X	X ^{a,b}
Novell SLES V9 SP1	X	X ^b

a. Не для виртуального SCSI.
b. Не для клиентов с AIX 5L.

Важно. Сервер виртуального ввода-вывода Virtual I/O Server не выполняет приложений конечного пользователя.

1.5.1. IBM AIX 5L для систем System p5

AIX 5L поддерживается серверами System p5 в разделах с выделенными процессорами подобно системам IBM @server pSeries с процессорами POWER4, но серверы System p5 не поддерживают технологию «affinity partitioning» (деление по аппаратным границам). Для систем System p5, сконфигурированных с функцией Advanced POWER Virtualization (APV), требуется AIX 5L V5.3 или более поздней версии для разделов, работающих в общем процессорном пуле, для виртуального ввода-вывода и виртуального Ethernet.

В системе System p5 поддерживаются смешанные среды с разделами AIX 5L V5.2 ML2 и AIX 5L V5.3 с выделенными процессорами и адаптерами и с разделами AIX 5L V5.3, использующими микроразделы и виртуальные устройства. Разделы AIX 5L V5.3 могут использовать одновременно физические и виртуальные ресурсы.

1.5.2. Linux для систем System p5

Linux является операционной системой с открытым исходным кодом, работающей на многочисленных платформах от встроенных систем до компьютеров-мэнфреймов. Она обеспечивает UNIX-подобную реализацию на многих компьютерных архитектурах.

В этом разделе обсуждаются две версии Linux, выполняющиеся в логических разделах; здесь не затрагивается базирующийся на Linux сервер Virtual I/O Server. Версиями Linux, поддерживаемыми серверами System p5, являются:

- ▶ Novell SUSE Linux Enterprise Server V9
- ▶ Red Hat Enterprise Linux Advanced Server V3 и V4

Функции виртуализации APV, кроме предназначенных для виртуального SCSI-сервера, поддерживаются версией 2.6.9 ядра Linux. Серийно выпускаемые последние дистрибутивы фирмы Red Hat, Inc. (RHEL AS 4) и фирмы Novell SUSE LINUX (SLES 9) поддерживают архитектуры IBM POWER4, POWER5 и 64-разрядную архитектуру PowerPC® 970 и ориентированы на его ядро серии 2.6. Также поддерживается модифицированная версия 2.4 ядра сервера Red Hat Enterprise Server AS 3 (RHEL AS 3) с обновлением 3. Ядро Linux, поставляемое с SLES9 SP1, также поддерживает виртуальный SCSI-сервер.

Клиенты, желающие конфигурировать разделы Linux в виртуализованных системах System p5, должны учитывать следующее:

- ▶ Не все устройства и функции, поддерживаемые операционной системой AIX 5L, поддерживаются в логических разделах с операционной системой Linux.
- ▶ Лицензии на операционную систему Linux закзываются отдельно от аппаратных ресурсов. Клиенты могут приобрести лицензии на операционную систему Linux у IBM, и они будут включаться в поставку систем System p5, или приобрести их у других дистрибуторов Linux.
- ▶ Клиенты или авторизованные бизнес-партнеры отвечают за установку и поддержку операционной системы Linux в системах System p5.
- ▶ Независимо от способа заказа дистрибутива Linux дистрибуторы предлагают обслуживание и поддержку. У компании IBM также есть предложения по поддержке этих дистрибутивов, реализуемые службой IBM Global Services.

- ▶ Хотя Linux может успешно выполняться в разделах с более чем восемью процессорами (ядро 2.6 Linux может масштабироваться до 16 или даже 24 процессоров для определенных нагрузок), типичные нагрузки Linux будут эффективно использовать только до 4 или 8 процессоров.

Поддерживаемые функции виртуализации SLES9 и RHEL AS4 поддерживают следующие функции виртуализации:

- ▶ Virtual SCSI, включая устройство загрузки
- ▶ Разделы, работающие в общем процессорном пуле, и виртуальные процессоры с верхним пределом (capped) и без него (uncapped)
- ▶ Разделы с выделенными процессорами
- ▶ Динамическую реконфигурацию процессоров
- ▶ Virtual Ethernet, включая соединения через Shared Ethernet Adapter в сервере Virtual I/O Server с физическим соединением Ethernet
- ▶ Одновременную многопоточную обработку (SMT)

Как SLES9, так и RHAS4 не поддерживают следующих функций:

- ▶ Динамическую реконфигурацию памяти
- ▶ Динамическую реконфигурацию слотов ввода-вывода
- ▶ Partition Load Manager (PLM)

1.5.3. IBM i5/OS для систем System p5

Операционная система IBM i5 (i5/OS) обеспечивает гибкие опции управления нагрузкой, позволяющие быстро развертывать высокопроизводительные приложения в масштабе предприятия. С помощью своего базового набора функций и бесплатных опций i5/OS обеспечивает легкую реализацию, управление и эксплуатацию с помощью интеграции базового программного обеспечения, необходимого большинству компаний.

Позиционирование операционной системы в системах System p5

i5/OS, работающая в системе System p5, предназначена для клиентов, имеющих ограниченный объем нагрузки на i5/OS, ограниченный ожидаемый рост такой нагрузки и желающих консолидироваться на едином сервере, в котором основная часть нагрузки будет представлена AIX 5L или Linux. Клиенты с новыми средними или малыми нагрузками i5/OS или с более старыми моделями AS/400® или iSeries™ могут модернизировать существующие системы System p5 для включения в них разделов i5/OS с выделенными процессорами или разделами, совместно использующими процессоры.

Архитектура i5/OS

i5/OS в серверах System p5 немного отличается от i5/OS в сервере IBM @server i5. Вам следует понять эти различия перед работой с разделами i5/OS совместно с разделами AIX 5L и Linux. Различные уровни системного программного обеспечения для каждой ОС показаны на рисунке 1-4, а уровни i5/OS обсуждаются в следующих разделах книги.

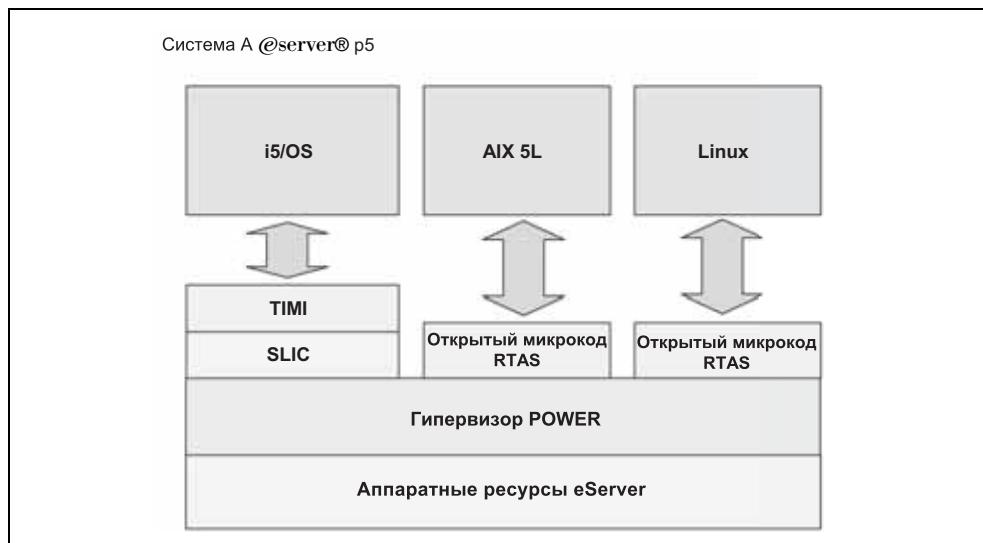


Рис. 1-4. Уровни программного обеспечения между аппаратными ресурсами и операционными системами

Примечание. Разделы i5/OS поддерживаются только на серверах p5-570, p5-590 и p5-595 и требуют подсистемы ввода-вывода для i5/OS.

Лицензируемый внутренний код системы

В i5/OS имеются два компонента: Лицензируемый внутренний код системы – System Licensed Internal Code (SLIC) и i5/OS. SLIC – это уровень программного обеспечения, размещающийся над гипервизором POWER.

Не зависящий от технологии машинный интерфейс

Не зависящий от технологии машинный интерфейс – Technology Independent Machine Interface (TIMI) контролирует осуществление доступа операционной системы к аппаратным ресурсам, абстрагируя управление устройством от операционной системы.

Поддержка виртуального ввода-вывода

Хотя возможности виртуального ввода-вывода имеются в i5/OS, виртуальный SCSI не поддерживается разделами i5/OS в системах System p5. Механизмы ввода и вывода операционной системы i5/OS значительно отличаются от подобных механизмов AIX 5L и Linux.

I/O subsystem для i5/OS обеспечивает ввод-вывод для разделов i5/OS. Она обеспечивается как модель (machine type/model) 9411-100. Только одна 9411-100 требуется для сервера с i5/OS. Разделы i5/OS не могут использовать I/O-слоты в CEC и модули расширения (RIO drawers), а разделы AIX 5L или Linux не могут использовать ввод-вывод i5/OS.

Поддержка виртуального Ethernet

Разделы AIX 5L, Linux и i5/OS поддерживают виртуальный Ethernet.

На рисунке 1-5 показан пример системы IBM *@server p5* с несколькими микроразделами; два из них (i5/OS) имеют собственную систему ресурсов, использующую возможности виртуального соединения Ethernet.

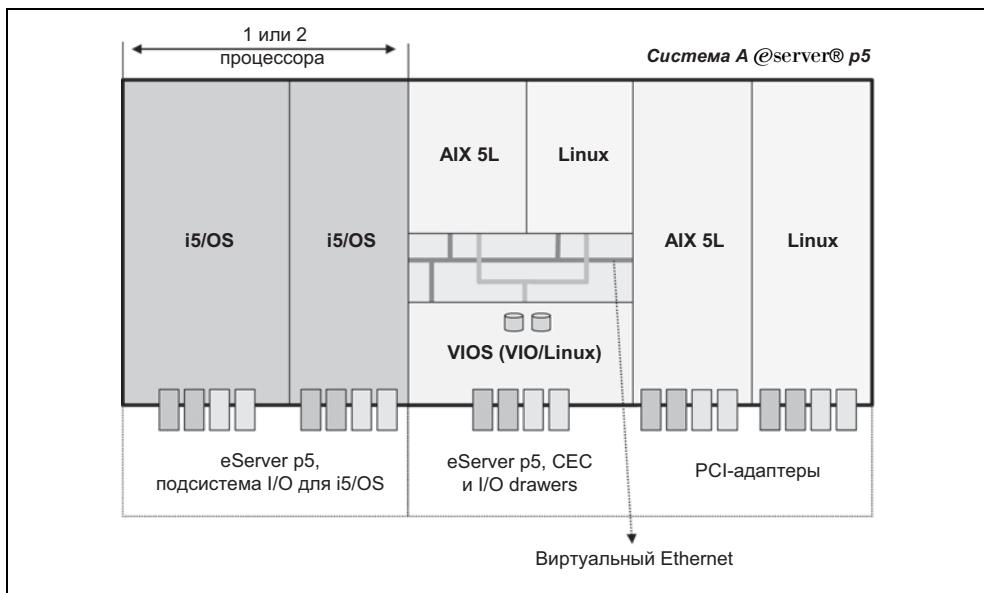


Рис. 1-5. Консолидированные разделы i5/OS и UNIX в системе p5

Примечание. В p5-570 для разделов i5/OS может использоваться только один процессор (до 10 микроразделов i5/OS). В p5-590 или p5-595 могут выделяться до двух процессоров для разделов i5/OS (максимум до 20 микроразделов i5/OS).

Минимальные требования раздела i5/OS

Каждый раздел i5/OS имеет следующие минимальные требования:

- ▶ Процессор (минимум 0,1 на раздел)
- ▶ Память (128 МБ; рекомендуется 256 МБ)
- ▶ Один или более процессоров ввода-вывода (I/O processor, IOP), в зависимости от конфигурации
- ▶ LAN-адаптер
- ▶ Коммуникационный адаптер электронной поддержки клиента (Communication adapter for Electronic Customer Support)
- ▶ Медиа-адаптер для ленточных/DVD устройств
- ▶ Дисковый адаптер и дисковые устройства (рекомендуется зеркализованная пара или массив RAID)
- ▶ По заказу: Процессор IOP и адаптер IOA для консоли iSeries Operations Console/Twinaxial

Более подробную информацию о консолидации разделов i5/OS в системах IBM @server p5, их установке и конфигурации можно получить в книге *i5/OS on IBM @server p5 Models – A Guide to Planning, Implementation, and Operation*, SG24-8001.

1.5.4. Итоги

В таблице 1-2 показана краткая сводка взаимоотношений функций APV с операционными системами, поддерживаемыми данными системами.

Таблица 1-2. Операционные системы, поддерживаемые для функций APV

Система / функция APV	VIOS V 1.1 и V1.2	VIOS Linux	i5/OS	AIX 5L V5.2 ML2	AIX 5L V5.3	RHEL AS V3	RHEL AS V4	SLES V9
Разделы с выделенными процессорами	X	X	X	X	X	X	X	X
Микроразделы	X	X	X		X	X	X	X
Виртуальный терминал	X	X		X	X	X	X	X
Виртуальная консоль клиент/сервер		X	X			X	X	X
Виртуальный Ethernet	X	X	X		X	X	X	X
Загрузка с виртуального Ethernet					X	X	X	X
Совместно используемый адаптер Ethernet (SEA)	X							
Мост Ethernet с STP-поддержкой		X				X	X	X
Виртуальный SCSI-сервер	X	X						
Виртуальный SCSI-клиент					X	X	X	X
Загрузка с диска виртуального SCSI-клиента					X	X	X	X
Виртуальная лента			X					
Виртуальный CD	X	X	X		X	X	X	X
Загрузка с виртуального CD					X	X	X	X
Partition Load Manager				X	X			
Integrated Virtualization Manager	X				X	X	X	X
Динамические операции (DLPAR) с ЦП	X	X	X	X	X		X	X
Динамические операции с ОЗУ	X		X	X	X			
Динамические операции с физическими адаптерами	X		X	X	X			
Динамические операции с виртуальными адаптерами	X		X		X			
Онлайновое сканирование устройств виртуального SCSI	X		X		X		X	X

1.6. Сравнение двух технологий виртуализации компаний IBM

Система IBM System z9™ оставила глубокий след в области использования разделов и виртуализации. Некоторые из новых функций виртуализации, внедренные в системы IBM System p5, были унаследованы от IBM System z9 и ее предшественников. Если вы знакомы с концепциями использования разделов и виртуализации в системе System z9, то вам следует знать, что некоторые из них напоминают концепции системы IBM System p5, но не идентичны им.

В системе z9 имеются два варианта виртуализации: PR/SM™ и z/VM®. PR/SM обеспечивает использование логических разделов и базовую виртуализацию, а z/VM предлагает усовершенствованную технологию виртуализации. z/VM может развертываться в LPAR системы System z9 и обеспечивает виртуальные машины (Virtual Machine, VM) с виртуальными ресурсами. Возможности виртуализации системы System p5 находятся где-то между этими двумя вариантами виртуализации системы System z9.

В таблице 1-3 приводится краткий обзор возможностей виртуализации, имеющихся в системах IBM System z9 и System p5. Некоторые их различия кратко описываются далее.

Таблица 1-3. Сравнение возможностей виртуализации систем IBM System z9 и System p5

Функция / возможность	Технология виртуализации IBM System z9		Технология виртуализации IBM System p5
ПО, предоставляющее возможности виртуализации	Processor Resource/Systems Manager (PR/SM).	z/VM.	Гипервизор POWER.
Максимальное количество виртуализованных серверов	Максимум 60 разделов Logical Partition (LPAR), в зависимости от модели.	Произвольное количество виртуальных машин Virtual Machine (VM), также называемых Guest, ограниченное только ресурсами.	Максимум 254 раздела Logical Partition (LPAR) в зависимости от модели, максимум 10 на один процессор.
Совместное использование процессорных ресурсов	Разделам LPAR назначаются логические процессоры и взвешенные доли центральных процессоров или выделенные процессоры. Разделы LPAR с совместно используемыми процессорами могут быть с верхним пределом (capped) или без него (uncapped).	Виртуальным машинам VM назначаются совместно используемые или выделенные виртуальные процессоры и абсолютные или относительные взвешенные доли виртуальных процессоров. VM с совместно используемыми процессорами могут быть без верхнего предела, с «мягким» (soft capped) или с «жестким» (hard capped) верхним пределом.	Разделам LPAR назначаются либо выделенные физические ЦП, либо процентные доли физических ЦП и определенное количество виртуальных процессоров. LPAR с долями ЦП могут быть с верхним пределом или без него.
Управление распределением нагрузки между разделами	Intelligent Resource Director (IRD) с разделами z/OS®.	Virtual Machine Resource Manager (VMRM).	Partition Load Manager (PLM) с разделами AIX 5L.

Продолжение табл.

Функция / возможность	Технология виртуализации IBM System z9	Продолжение табл. Технология виртуализации IBM System p5
Совместное использование ресурсов памяти	Память LPAR является фиксированной и закрытой; для разделов LPAR с z/OS размеры памяти могут динамически изменяться с некоторыми ограничениями.	Порции памяти VM могут совместно использоваться в режиме только чтения или чтения-записи вместе с другими VM; изменение размеров памяти VM требует процедуры перезагрузки для этой VM
Виртуальный обмен данными между разделами	TCP/IP с HiperSockets™.	TCP/IP с виртуальными HiperSockets, TCP/IP и другие протоколы через виртуальный Ethernet с поддержкой IEEE 802.1Q VLAN.
Совместное использование соединений с внешними сетями	Enhanced Multiple Image Facility (EMIF) мультиплексирует Open Systems Adapter (OSA) между несколькими разделами LPAR.	Виртуальный коммутатор Ethernet создает мост между виртуальным Ethernet и внешним Ethernet через OSA, z/VM также пользуется преимуществом мультиплексного доступа к OSA через EMIF.
Совместное использование ресурсов ввода-вывода.	EMIF мультиплексирует каналы и устройства между несколькими разделами LPAR.	z/VM обеспечивает виртуальные устройства, такие как мини-диски, являющиеся разделами физических дисков, и она обеспечивает совместный доступ к физическим устройствам.
Поддерживаемая OS.	z/OS, Linux, z/VM и другие операционные системы zSeries®	VIOS обеспечивает виртуализированные диски, которые могут быть разделами физических дисков и доступ к которым осуществляется через виртуальные SCSI-адAPTERЫ.
		AIX 5L, Linux и i5/OS на некоторых моделях.

Механизмы совместного использования процессорных ресурсов в системах IBM System z9 и System p5 похожи. Количество виртуальных процессоров в разделе LPAR системы System p5 или в виртуальной машине z/VM может превосходить количество установленных физических процессоров. Есть некоторые различия в установлении верхнего предела, но они за рамками данного обсуждения.

В сервере System z9, интегрированное управление распределением нагрузки между разделами и в разделах может выполнять Intelligent Resource Director (IRD) для LPAR-разделов z/OS, в то время как в System p5 распределением нагрузки между разделами управляет Partition Load Manager (PLM), а внутри разделов – Workload Manager (WLM) операционной системы AIX 5L. Так как PLM и WLM не являются интегрированными, то PLM в системе System p5, как и VMRM в системе System z9, отслеживает приоритеты и цели приложений WLM, а IRD может регулировать выделение ресурсов, если какие-то цели производительности приложений были пропущены.

Использование разделами LPAR памяти в системах System z9 и System p5 основано на ее разделении. Таким образом, сумма участков памяти, назначенных раз-

делам LPAR, не может превосходить объем физически установленной памяти. Виртуальная машина z/VM может совместно использовать память, осуществляя страничную организацию VM. Таким образом, сумма назначенных участков памяти может превосходить объем физически установленной памяти, что происходит в большинстве случаев.

Механизм управления страницами автоматически и динамически регулирует назначение физической памяти для виртуальных машин.

System p5 и z/VM обеспечивают виртуальный Ethernet и реализуют коммутаторы и мосты виртуального Ethernet, которые работают на Уровне 2 и могли бы использоваться с любым протоколом Уровня 3. Сокеты HiperSocket PR/SM работают на Уровне 3 и обеспечивают обмен только по протоколу IP. Совместно используемый адаптер OSA поддерживает доступ к внешним сетям на Уровне 2 и Уровне 3.

Между System z9 PR/SM и System p5 имеется существенное различие, касающееся совместного использования ресурсов ввода-вывода и доступа к внешним сетям:

- ▶ В System z9 с PR/SM доступ к дискам, лентам, сетевым адаптерам и другим ресурсам ввода-вывода мультиплексируется с помощью EMIF. Таким образом, общими физическими ресурсами могут владеть несколько разделов LPAR в системе System z9
- ▶ z/VM, в дополнение к функциям PR/SM, позволяет распределять виртуальные устройства между виртуальными машинами. Имеется возможность мостового доступа от виртуальных Ethernet-сетей к внешним сетям (подобно System p5).
- ▶ В System p5 создаются виртуальные диски, резервируемые физическими дисками, и виртуальные Ethernet-сети могут связываться мостами с адаптерами физической сети Ethernet. Этими физическими ресурсами в системе System p5 владеет Virtual I/O Server.

Виртуальный ввод-вывод и виртуальный Ethernet базируются на системном ПО и микрокоде гипервизора в System p5 и при использовании z/VM, тогда как основная часть EMIF реализована в аппаратной части и микрокоде System z9. Поэтому процессорная нагрузка обработки ввода-вывода с помощью EMIF в System z9 ниже, чем при виртуальном вводе-выводе в System p5 или z/VM. Она почти та же, как в случае выделенных физических I/O-адаптеров в System p5.

z/VM работает в разделе LPAR системы System z9; поэтому PR/SM и z/VM по существу являются вложенными. В системе System z9 реализуются два уровня интерпретационного исполнения, чтобы обеспечить исполнение аппаратных ресурсов виртуальных машин в логических разделах z/VM. z/VM может быть саморазмещающей (self-hosting), что означает возможность запуска z/VM в guest z/VM и, следовательно, работы z/VM в z/VM в разделе LPAR системы System z9. z/VM может размещать в себе z/VM практически до любого уровня. У вас нет возможности вкладывать разделы LPAR систем System z9 или System p5.



2

Ценность технологии Advanced POWER Virtualization

В семействе серверов IBM System p5 предлагается ряд возможностей, позволяющих организациям и предприятиям привести свою вычислительную инфраструктуру в соответствие с целями бизнеса.

В этой главе приведено несколько сценариев возможных реализаций системы System p5, показывающих преимущества функций виртуализации. Показано, как клиенты могут использовать подобные функции для создания гибких и экономичных инфраструктур. В данных сценариях сделана попытка отразить ситуации, которые могут возникнуть в действительности.

2.1. Упрощение ИТ-систем и оптимизация ТСО¹

В этом сценарии показано, как деловая необходимость вынуждает клиента совершенствовать свою ИТ-инфраструктуру для поддержки новых требований бизнеса, ускорения отклика на запросы своих клиентов с одновременным удержанием инвестиций под своим контролем. Данный клиент предполагает модернизировать существующий у него комплекс посредством замены процессоров в некоторых системах и полной замены остальных систем новым оборудованием, и сравнивает это с возможностью консолидировать все приложения в новой системе System p5. Этот пример показан на рисунке 2-1.

Существующий комплекс					
Сервер	Приложение	Кол-во ЦП	Лицензии на Websphere	Лицензии на DB2	
Сервер А	ERP	4		4	
Сервер В	Отдел персонала	2		2	
Сервер С	Система сбыта	2		2	
Сервер D	Сервер приложений	4	4		
Сервер Е	Киоск данных	4		4	

Намерения заказчика в отношении отдельных серверов:

- Увеличение производительности в 1,5 раза для бизнес-приложений
- Критически важные приложения ERP и Системы поставок требуют до двукратного увеличения текущей производительности при пиковой нагрузке
- Снизить риск конфликтов по процессорным лицензиям (ожидание лицензий на совместно используемые процессоры)
- Снизить риск конфликтов по коммутируемым портам как в LAN, так и в SAN (серверы имеют резервные соединения с LAN и SAN; добавление коммутируемых портов требует новых устройств и перекладки кабелей)

Планируемая модернизация (обычный подход)

Сервер	Приложение	Кол-во ЦП	Лицензии на Websphere	Лицензии на DB2	Инвестиции
Сервер А	ERP	6		6	Замена ЦП, добавление ЦП
Сервер В	Отдел персонала	2		2	Новый сервер
Сервер С	Система сбыта	4		4	Добавление ЦП
Сервер D	Сервер приложений	4	4		Новый сервер
Сервер Е	Киоск данных	4		4	Замена ЦП

Рис. 2-1. Примерный сценарий с обычным подходом

Главными целями рабочей ИТ-группы в компании являются:

- Улучшение использования лицензий на ПО – увеличение количества приложений конечного пользователя, сервисов и объема обрабатываемых транзакций с сохранением инвестиций в программное обеспечение.
- Уменьшение стоимости владения – вложение средств в экономичное решение (аппаратные ресурсы и операционная система, лицензирование ПО, обслуживание аппаратных и программных ресурсов).

¹ Total cost of ownership – общая стоимость владения. Большинство сокращений объясняется в конце книги. Для удобства читателя они приводятся и в ходе обсуждения. Прим. перев.

- ▶ Уменьшение затрат сил и средств для работы на пиковой нагрузке (стоимость *запасной процессорной мощности*) с помощью двух факторов: 1) разделением системных ресурсов между приложениями (ЦП, ОЗУ, ввод-вывод и лицензии на ПО); 2) вычислениями «по требованию»
- ▶ Уменьшение сложности – работа с основной частью ИТ-инфраструктуры в готовой поставке без дополнительных внешних расходов (порты LAN и SAN и системное администрирование).

Системы IBM System p5 и IBM *@server* p5 показали свое превосходство в широком спектре сравнительного тестирования по параметрам обработки транзакций, планированию ресурсов предприятия (ERP), JavaTM, работе с веб-сервисами, в системе Business Intelligence¹, в Системе поставок, файловых сервисах и обеспечению высокопроизводительных вычислений. При некоторых видах нагрузок в сравниваемых системах использовалось в 1,5–4 раза больше ЦП, чем требовалось серверу System p5 для обеспечения той же производительности. Более подробно о результатах сравнительного тестирования систем System p5 вы можете узнать по адресу:

<http://www.ibm.com/@server/benchmarks>

Данное решение с System p5 создавалось со следующими допущениями:

- ▶ Инфраструктура в этом сценарии является работоспособной (UNIX-серверы и ПО, подключение к LAN и SAN, резервное копирование данных и системное администрирование).
- ▶ Лицензии на ПО являются действующими и предусматривают поддержку ПО по контрактам.
- ▶ Приложения конечного пользователя могут легко переноситься с других ОС UNIX в AIX 5L или Linux.
- ▶ Лицензии на ПО и приложения допускают их перемещение между системами с UNIX.
- ▶ При сравнении производительности на один процессор системы System p5 (1,65 ГГц в данном сценарии) и других UNIX-систем использовался умеренный коэффициент 1,5.
- ▶ Несмотря на то, что могло бы потребоваться ПО сторонних производителей, оно не упоминается для упрощения обсуждения.

Сравнение обоих вариантов – модернизации имеющихся серверов (пять систем, имеющих в сумме 20 процессоров) и консолидации их в новой системе System p5 (одна система с 11 активными процессорами и пятью разделами LPAR) – показывает, что дороже осуществляется модернизация имеющихся серверов. Использование функций APV в решении с System p5 обеспечивает следующие преимущества:

- ▶ Наиболее критичными приложениями являются ERP и Система поставок. Их пиковые нагрузки возникают в разное время (у ERP два пика в месяц, а Система поставок работает по ночам).
- ▶ Другие приложения используются конечными пользователями и требуют малого объема ресурсов в ночное время.

¹ Business Intelligence – система поддержки принятия решений, система бизнес-аналитики. Прим. науч. ред.

- ▶ ERP и Система поставок могут забирать вычислительные ресурсы у некритических приложений, например у Киоска данных (не требующегося ночью), или у приложений с малыми или временными потребностями в ресурсах, например у Отдела персонала.
- ▶ Лицензии IBM на ПО процессоров для WebSphere® и DB2 могут быть использованы одновременно несколькими микроразделами, если они не превышают значений, предусмотренных лицензиями на процентные доли использования процессоров¹.

Данное решение предусматривает в системе два сервера Virtual I/O Server. Компонент ERP, по причине интенсивного использования сети и хранилища, обеспечивается собственной системой соединений. Данное решение и связанные с ним проблемы показаны на рисунке 2-2.

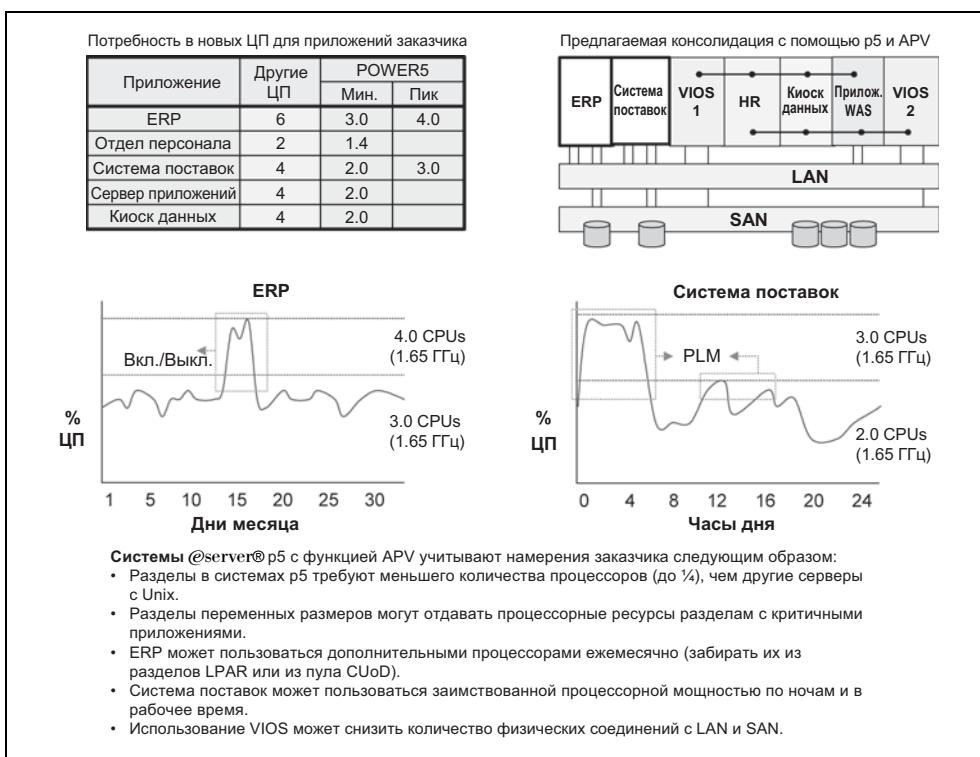


Рис. 2-2. Консолидация бизнес-приложений в системе p5

На рисунке 2-3 показаны детали планирования решения. Мы настоятельно рекомендуем прочитать разделы 3.3 «Введение в микроразделы» и 3.7 «Лицензирование ПО в виртуализированной среде» для получения представления об идеях, значениях и методах, используемых на этом рисунке.

¹ Entitlement – также «выделенная процессорная мощность». Прим. перев.

Proposed consolidation using APV (@server® p5 system approach)								
Раздел	Приложение	Процессорные единицы			VIOS	Лицензии на Websphere	Лицензии на DB2	PLM Group
		Мин.	Сред.	Макс.				
LPAR1	ERP	3.0	3.0	6.0	Выделенный		3.0	
LPAR2	Отдел персонала	1.0	1.4	2.0	Capped	VIOS1, VIOS2		Oценочно
LPAR3	Система поставок	2.0	2.2	4.0	Capped	VIOS1, VIOS2		Group A
LPAR4	Сервер приложений	1.4	2.0	3.0	Capped	VIOS1, VIOS2	2.0	
LPAR5	Киоск данных	0.2	2.0		Uncapped	VIOS1, VIOS2	3 (VPs)	Group A
VIOS1		0.2	0.2	0.2	Capped			
VIOS2		0.2	0.2	0.2	Capped			
On/Off	Не определено		1.0			1 On/Off	1 On/Off	

Система @server® p5

 Выгоды, получаемые заказчиком от консолидации с помощью @server® p5:
 • Увеличение производительности приложений существующих систем более чем 1,5 раза.
 • Критически важным приложениям ERP и Системе поставок может быть предоставлено до трехкратной имеющейся процессорной мощности в пиковые периоды (в динамическом, программируемом или автоматическом режиме)
 • Не увеличивается количество процессорных лицензий.
 • Высвобождающиеся процессорные лицензии могут использоваться для новых приложений.
 • Высвобождающиеся порты в коммутаторах LAN и SAN могут использоваться для новых приложений.
 • Улучшение доступности приложений (резерв ЦП, ОЗУ и адаптеров) и резервирования системы (питание, вентиляторы, часы [p590-595], адаптеры, диски, резервный VIOS).

Рис. 2-3. Технические подробности консолидации в системе p5

Эти решения различаются следующим:

- ▶ Клиент может смешивать различные нагрузки в глобальном процессорном пуле и определять на основании бизнес-целей, выделяется ли приложениям именно тот объем процессорных ресурсов, который им необходим.
- ▶ Высокие рабочие характеристики процессоров POWER5 и возможность создания микроразделов позволяют клиенту высвободить как аппаратные, так и программные ресурсы и использовать их для новых проектов или нагрузок.
- ▶ Стоимость, риски и усилия по обеспечению работы при пиковых нагрузках или в незапланированных бизнес-ситуациях могут быть слишком большими для обычных серверов (отклонение транзакций, большое время отклика, время на развертывание дополнительных ресурсов, непредвиденные расходы, неустойки и т. д.), но с подобными ситуациями можно легко справиться с помощью виртуализованной среды.

На рисунке 2-4 показаны преимущества, которые может понять клиент после оценки решения с IBM @server p5.

2.2. Доступность бизнес-приложений

Клиентам необходима доступность своих бизнес-приложений даже при работе на консолидированных серверах. Такая доступность может опираться на функции RAS аппаратных ресурсов и операционных систем (как объяснялось в 1.3 «Характеристики RAS виртуализированных систем»), конфигурацию резервирования системы и функции динамического восстановления, все из которых предназначены для создания очень надежной системы.

Другие Unix-серверы и старые модели IBM pSeries

Система @server®p5 с APV

Всего активных процессоров в системе	Существующий комплекс	Модернизация существующего комплекса	Консолидация в системе с @server®p5	Выгоды от системы с @server®p5
Общее количество процессоров в системе	16	20	11 + 1 On/Off	Почти половина процессоров
Количество серверов (аппаратных)	5	5	1	Только один сервер
Количество портов в коммутаторах LAN	10	10	8	Экономия портов LAN
Количество портов в коммутаторах SAN	10	10	6	Экономия портов SAN
Количество процессорных лицензий для IBM DB2	12	16	10 + 1 On/Off	Экономия лицензий
Количество процессорных лицензий для IBM WebSphere	4	2	2 + 1 On/Off	Экономия лицензий
Процессорный ресурс для ERP (базовый ресурс + On/Off)	1X (базовый ресурс)	2X	1.5X - 2X	Нет простоя ЦП
Процессорный ресурс для Системы поставок (базовый + замыкаемый)	1X (базовый ресурс)	2X	1.5X - 2X	Нет простоя ЦП
Максимальный процессорный ресурс для ERP	1X (базовый ресурс)	2X	3X+ (1)	Снижение риска конфликтов
Максимальный процессорный ресурс для Системы поставок	1X (базовый ресурс)	2X	3X+ (1)	Снижение риска конфликтов
Временные малые серверы с существующими ресурсами	НЕТ	НЕТ	Да	Больше гибкости

(1) ERP и Система поставок могут забирать до 2,0 ЦП у других приложений и 1,0 ЦП у CUoD в критические моменты

Рис. 2-4. Сводка дополнительных преимуществ решения на базе системы с p5

Этот подход не решает всех проблем, связанных с обслуживанием выполняющихся приложений, так как могут возникать события, оказывающие влияние на всю систему в целом, например:

- ▶ Ремонт системы электроснабжения в информационном центре.
- ▶ Перемещение одного из серверов в другое место.
- ▶ Более длительный срок модернизации, чем это допускается конечным пользователем.
- ▶ Непредвиденная необходимость краткосрочного выделения ресурсов для критичных приложений, вызывающего временное перемещение некритичных приложений на другие системы.
- ▶ Общий отказ, влияющий на всю систему.

Следовательно, хорошей стратегией улучшения доступности бизнес-приложений является использование двух и более систем в информационном центре с распределением между ними бизнес-приложений. В данном разделе обсуждаются два подхода в рамках этой стратегии (два сервера с распределенной нагрузкой), не содержащих сложного ПО, инструментов или конфигураций: использование только функции Advanced POWER Virtualization, структуры системы с внешним хранилищем данных (предпочтительно, размещенного в SAN) и, по выбору, программного обеспечения IBM HACMP.

2.2.1. Улучшенные средства перемещения приложений

В этом сценарии показана система В с сервером IBM @server p5, поддерживающая несколько приложений в трех разделах (один из разделов используется для тестирования ПО). Эту систему необходимо установить в одном из подразделений, чтобы начать новый проект, и приложения будут перемещаться (временно), пока не прибудет новая система или другая система не будет модернизирована до объема ресурсов, позволяющего размещать такие приложения на постоянной основе. Эта ситуация отображена на рисунке 2-5.

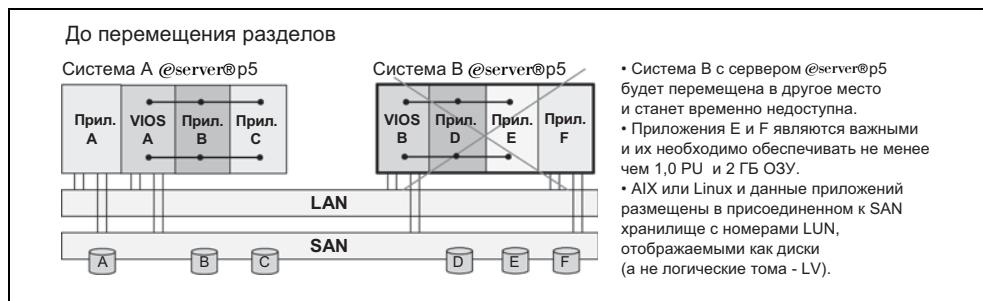


Рис. 2-5. Сценарий перемещения или восстановления приложения

Достоинства систем с APV становятся их отличительным признаком при сравнении с другими UNIX-системами (даже с аппаратными разделами), так как виртуализация решает наиболее важную задачу: найти один сервер с процессорами, памятью и адаптерами для соединений LAN и SAN, выделенный операционной системе для каждого перемещаемого приложения, не трогая других приложений.

Восстановление приложения не является такой же простой процедурой, как его перезапуск на существующей операционной системе с другими выполняемыми приложениями. Большинство приложений требуют отдельных ресурсов (например, банковские приложения), специального промежуточного ПО и особого сетевого конфигурирования или специальных уровней операционной системы и промежуточного ПО. Более того, перемещение приложения в существующую операционную систему подобно *миграции* приложения по времени, усилиям и возможности неудач, которые она подразумевает.

В этом сценарии новая хост-система А имеет достаточно ресурсов для размещения вычислительной мощности, необходимой приложениям Е и F с их собственной операционной системой. На рисунке 2-6 можно видеть, что вычислительная мощность для приложения А, критичном для организации, остается той же самой.

Система А @server®p5		Проц. единицы		Память, ГБ	
LPAR / Приложение		До	После	До	После
VIOS		0.2	0.2	0.6	0.6
Приложение А		4.0	4.0	12	12
Приложение В		1.8	1.4	4	4
Приложение С		1.0	0.2	3	2
Приложение Е		0.0	1.0	0	2
Приложение F		0.0	1.2	0	3
Неиспользуемые/неназначенные On-Off		1.0	0.0	4	0

Порядок заимствования процессоров и памяти

1. Неиспользуемые/неназначенные ресурсы
2. Непроизводственные приложения
3. Некритичные приложения
4. Ресурсы On/Off
5. Критичные приложения

Рис. 2-6. Динамическая реконфигурация системных ресурсов в системе А с процессорами p5

На рисунке 2-7 показано, как приложения Е и F могут перемещаться в системе А при помощи следующих простых шагов:

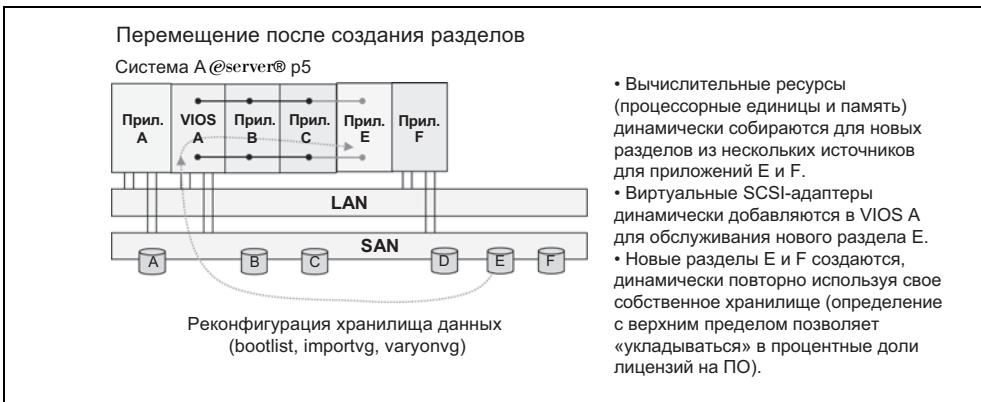


Рис. 2-7. Система с новыми разделами из другой системы

- ▶ Новые временные виртуальные SCSI-адаптеры создаются в сервере Virtual I/O Server A для обеспечения виртуальных SCSI-сервисов в новом разделе для приложения F.
- ▶ Номера LUN в SAN, принадлежащие приложению E, реконфигурируются для сервера Virtual I/O Server A, и для них назначаются связи непосредственно с новым разделом E.
- ▶ Вычислительные ресурсы высвобождаются из существующих разделов (особенно процессорные единицы) либо забираются из неназначенных пулов (например, память и адаптеры) или из пулов «по требованию».
- ▶ Новые разделы E и F создаются динамически, и их профили указывают на правильные устройства загрузки (виртуальному SCSI-диску назначается связь с SAN для раздела E, SAN – для раздела F).
- ▶ Новые разделы E и F запускаются. Так как они используют свои первоначальные образы систем, им не требуется дополнительной реконфигурации (может быть необходимым конфигурирование сети, если соединения VLAN или физического Ethernet различаются).

Примечание. Динамическое перераспределение памяти может занять много времени в зависимости от объема и использования памяти в разделе-доноре. Рекомендуется иметь в системе достаточно неназначенной или находящейся в CUoD памяти для поддержки временного перемещения разделов из других систем вместо использования больших объемов памяти из существующих разделов.

В таблице 2-1 обобщены выгоды, получаемые от систем с APV вашей организацией при необходимости перемещения приложений в нужный момент.

Таблица 2-1. Преимущества систем с APV для восстановления приложений

Выгоды, получаемые организацией	Другие статические UNIX-серверы?	Системы IBM @server p5
Динамически выделяются временные серверы для тестов, перемещения аппаратных ресурсов или новых проектов (чтобы позволить запускать новые проекты или тесты, не ожидая предоставления аппаратных ресурсов)	Нет, если только клиент не желает смешивать приложения в одной и той же операционной системе.	Да, у вас есть достаточно ресурсов для удовлетворения минимальных требований (ЦП, ОЗУ, ввод-вывод).
Динамически и легко восстанавливаются приложения из других серверов без конфигурирования кластера с высокой доступностью (ручное восстановление)	Нет, если только клиент не желает смешивать приложения в одной и той же операционной системе и менять им.	Да, вы собираете ресурсы в других серверах для размещения разделов отказавшей системы.
Динамически увеличиваются вычислительные мощности критических приложений для сохранения бизнес-ситуации или расширения масштаба (быстрое перемещение некритичных разделов для высвобождения дополнительных ресурсов критичным разделам при исчерпании ресурсов «по требованию»)	Нет, если критический сервер не позволяет динамического получения большей мощности.	Да, вы собираете ресурсы в других серверах для размещения разделов, высвобожденных из критичных систем.

2.2.2. Решения с высокой доступностью с НАСМР и APV

В этом сценарии клиенту из телекоммуникационного сектора требуется решение с высокой доступностью к основным приложениям. Клиенту также необходимы непроизводственные приложения для поддержки развития и обеспечения качества (QA) основных приложений. В соответствии с характером деятельности в проекте жестко устанавливается нижний допустимый предел на количество вычислительной мощности, позволяющий приложениям работать в нужных временных рамках обработки (окнах обработки пакетов) и соблюдать минимальное время отклика на запросы конечных пользователей.

Клиент может реализовать это решение с помощью двух систем, применяя метод совместного производства (*crossed production*), то есть разделы в режиме ожидания для кластеров НАСМР и некритичные процессорные пулы рядом с критичными приложениями (процессоры и память On/Off и непроизводственные разделы, отдающие ресурсы критичным приложениям).

На рисунке 2-8 показана предлагаемая конфигурация.

Примечание. НАСМР поддерживается как для разделов AIX 5L с прямым вводом-выводом (выделенные адаптеры), так и для клиентских разделов AIX 5L, использующих ввод-вывод через Virtual I/O Server. В кластере НАСМР все узлы должны быть одного типа; следовательно, разделы AIX 5L с выделенным вводом-выводом не могут смешиваться с клиентскими разделами AIX 5L, использующими Virtual I/O Server.

Система А @server p5 и APV					Система В @server p5 и APV				
LPAR / Приложение	ЕС PU	ОЗУ ГБ	VIOS	Кластер	LPAR / Приложение	ЕС PU	ОЗУ ГБ	VIOS	Кластер
Биллинг – Параллельная БД	5.0	18		БД	Биллинг – Параллельная БД	5.0	18		БД
Медиатор – Режим ожидания	0.4	4		НАСМР	Медиатор – Основная система	2.0	6		НАСМР
Отчеты – Основная система	3.0	8		НАСМР	Отчеты – Ожидание	0.4	6		НАСМР
БД CRM – Режим ожидания	0.4	7		НАСМР	БД CRM – основная	2.2	7		НАСМР
Прил. А CRM	2.3	7			Прил. В CRM	2.2	7		
CRM Разработка	1.2	3	X		Медиатор Разр./QA	0.8	3	X	
Биллинг Разр./QA	1.5	4	X		CRM QA	1.2	4	X	
VIOS A	0.2	1			VIOS B	0.2	1		
CUoD / On-Off	2.0	4			CUoD / On-Off	2.0	4		

Рис. 2-8. Конфигурация двух систем с совместными кластерами

Это решение ценно для организации тем, что сбои ПО или аппаратуры, вызывающие перенос приложений на резервные системы, могут управляться обычными способами, но серверы, работающие в режиме ожидания, могут быть достаточно расширены для запуска критичных баз данных и приложений (используя доли величиной в 1/10 процессора POWER для процессорных единиц микроразделов), и НАСМР (или другое промежуточное ПО) можно запрограммировать на заимствование процессорных единиц у других (конкретных) разделов, чтобы динамически выполнить восстановление.

На рисунке 2-9 приведен пример восстановления приложения Отчеты на системе В.

Система А @server@p5 и APV					Система В @server@p5 и APV				
LPAR / Приложение	ЕС PU	RAM ГБ	VIOS	Кластер	LPAR / Приложение	ЕС PU	RAM ГБ	VIOS	Кластер
Биллинг – Параллельная БД	5.0	18		БД	Биллинг – Параллельная БД	5.0	18		БД
Медиатор – режим ожидания	0.4	4		НАСМР	Медиатор – основная система	2.0	6		НАСМР
Отчеты – основная система	3.0	8		НАСМР	Отчеты – режим ожидания	3.0	6		НАСМР
БД CRM – режим ожидания	0.4	7		НАСМР	БД CRM – основная система	2.2	7		НАСМР
Прил. А CRM	2.3	7			Прил. В CRM	2.2	7		
CRM Разр.	1.2	3	X		Медиатор Разр./QA	0.6	3	X	
Биллинг Разр./QA	1.5	4	X		CRM QA	0.5	4	X	
VIOS A	0.2	1			VIOS B	0.2	1		
CUoD / On-Off	2.0	4			CUoD / On-Off	1.0	4		

Автоматическое восстановление приложений, автоматическое перемещение процессорных единиц, ручное перемещение памяти только по необходимости (для ускорения освобождения памяти после восстановления)

Рис. 2-9. Пример восстановления после отказа в предложенной конфигурации

В «Минимизации времени простоя с помощью использования технологии IBM НАСМР при модернизации ПО в среде LPAR одной системы IBM pSeries, IBM and Availant 2004» (*Minimizing downtime by using IBM HACMP to perform software upgrades in a single in frame IBM pSeries LPAR environment, IBM and Availant 2004*) показаны примеры скриптов для динамического перемещения ресурсов. Их можно посмотреть по адресу:

http://www.ibm.com/servers/@server/pseries/software/whitepapers/hacmp_lpar.pdf

Этот подход дает клиенту экономию аппаратных и программных ресурсов и расходов на обслуживание, но позволяет сохранить те же соглашения об уровне предоставляемых услуг, как и при работе с разделами фиксированного размера (например, аппаратными разделами). На рисунке 2-10 показан сценарий наихудшего случая: отказ всей системы (риск которого минимизирован структурой системы System p5) и уровень предоставления услуг, который может быть обеспечен для организации.

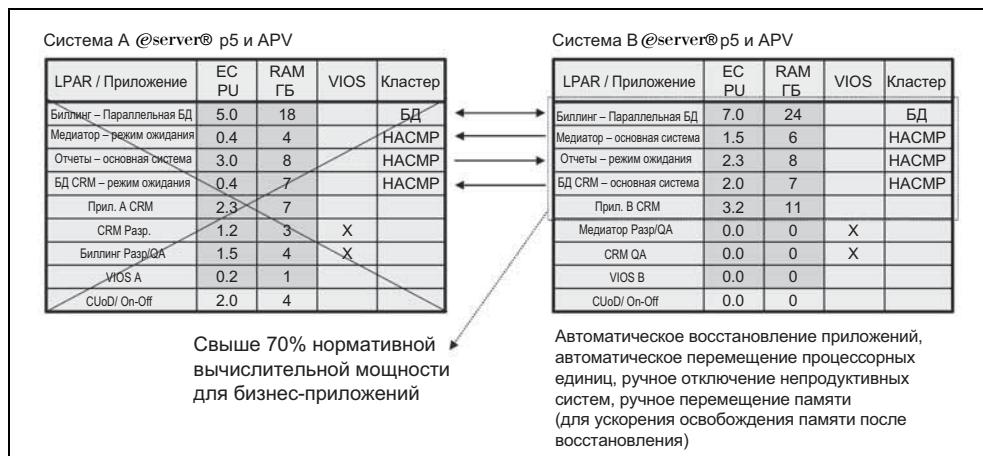


Рис. 2-10. Пример восстановления с перемещением всей системы на другую систему

2.3. Улучшение решений обеспечения непрерывности бизнеса

Функции динамической работы с разделами и виртуализации систем становятся отличным средством для проектирования и реализации у клиентов эффективных решений, позволяющих восстанавливать критичные приложения при отказе в масштабах системы или всего предприятия.

В следующем сценарии показаны три основные системы организации, размещенные в трех разных местах. Имеется еще одно место с ИТ-ресурсами, предназначенными для репликации данных и восстановления бизнес-операций. На рисунке 2-11 показана возможная конфигурация системы для сбора всех реплик производственных систем; такая система обеспечивает непрерывность работы, действуя как глобальный процессорный пул, способный динамически реконфигурироваться при замене производственной системы.

Имея до 10 разделов на процессор, одна система с несколькими процессорами может одновременно работать с приложениями одной или нескольких удаленных систем, а также со своими собственными приложениями для репликации данных остальных мест размещения. Иногда возникают затруднения в использовании других систем из-за наличия в них малого количества разделов, низкой степени разбиения и зависимости от процессорных ресурсов.

На рисунке 2-12 показан пример возможной ситуации восстановления.

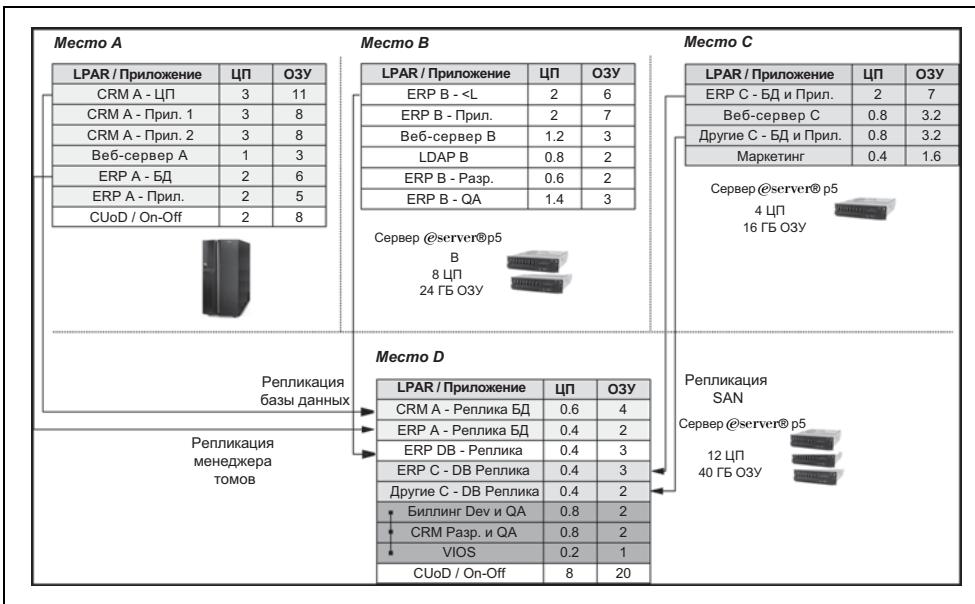


Рис. 2-11. Сценарий с системой для восстановления бизнес-операций

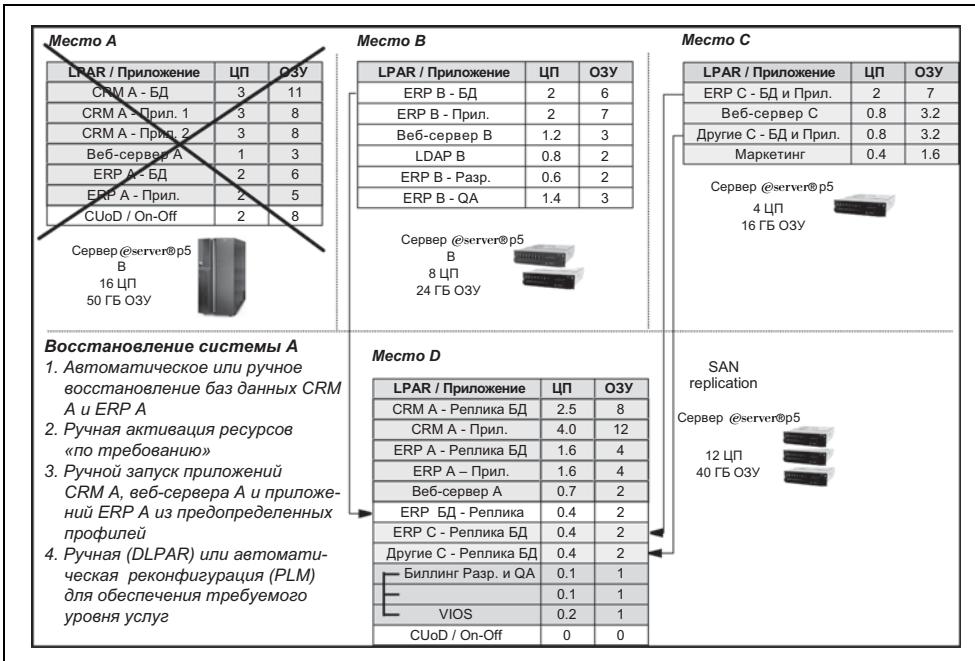


Рис. 2-12. Восстановление системы А перемещением в систему В

Ценность систем с APV для такого решения заключается в том, что одна-единственная система может действовать как глобальный процессорный пул, готовый к своей динамической реконфигурации и замене одной и более систем. При этом снижаются затраты по следующим причинам:

- ▶ Большинство процессоров системы находятся в неактивном состоянии. Только одноранговые разделы, которым необходима репликация данных, являются активными (например, БД), и они конфигурируются солями процессора, так как не имеют нагрузки конечного пользователя.
- ▶ Большинство процессорных лицензий совместно используются разделами и уменьшены вследствие малых размеров разделов. Кроме того, большинство поставщиков ПО разрешают использовать процессорные лицензии неиспользуемой системы в другой системе, предназначеннной для восстановления приложений.
- ▶ Почти нет ограничений на распределение процессорных ресурсов (процессорные единицы и ОЗУ) между восстанавливаемыми приложениями и точную подгонку вычислительной мощности под их потребности.



3

Технологии Virtualization Engine в серверах System p5

В этой главе обсуждаются различные технологии, использующиеся в системах IBM System p5 и IBM @server p5. Будут рассматриваться следующие темы:

- ▶ Новые функции в версии 1.2 сервера Virtual I/O Server
- ▶ Функция Advanced POWER Virtualization
- ▶ Введение в микроразделы
- ▶ Начальные сведения об одновременной многопоточной обработке
- ▶ Начальные сведения о POWER Hypervisor
- ▶ Лицензирование ПО в виртуализованной среде
- ▶ Ознакомление с виртуальным и общим Ethernet
- ▶ Ознакомление с виртуальным SCSI
- ▶ Ознакомление с Partition Load Manager
- ▶ Integrated Virtualization Manager
- ▶ Динамические операции с LPAR
- ▶ Концепции виртуального ввода-вывода в Linux

3.1. Новые функции в версии 1.2 сервера Virtual I/O Server

Этот раздел содержит краткий обзор новых функций, включенных в версию 1.2 сервера Virtual I/O Server (VIOS), таких как:

- ▶ Поддержка виртуальных оптических устройств
- ▶ Переход на резерв с помощью Shared Ethernet Adapter
- ▶ Integrated Virtualization Manager
- ▶ Новые команды для пула хранения
- ▶ Усовершенствования в HMC, облегчающие конфигурирование и обслуживание

Многие из улучшений в Virtual I/O Server Version 1.2 направлены на упрощение конфигурирования и администрирования виртуализированной среды ввода-вывода.

Примечание. Для использования всех новых функций, перечисленных в этом разделе, может потребоваться обновление микрокода системы, программы HMC и сервера VIOS.

3.1.1. Виртуальные DVD-RAM, DVD-ROM и CD-ROM

Виртуальный SCSI (VSCSI) позволяет клиентским разделам совместно использовать физические устройства хранения данных (SCSI и Fibre Channel). В версии 1.2 сервера Virtual I/O Server добавляется поддержка таких оптических устройств, как DVD-RAM и DVD-ROM. Поддержка CD-ROM была введена в предыдущих версиях.

Запись в совместно используемое оптическое устройство в настоящее время ограничена записью на DVD-RAM. Поддержка устройств DVD+RW и DVD-RW не предусмотрена.

Физическое устройство хранения должно принадлежать серверу VIOS, и связь для него устанавливается таким же образом, как и связь виртуального диска с адаптером виртуального SCSI-сервера, с помощью команды `mkvdev`.

Виртуальное оптическое устройство может назначаться одновременно только одному клиентскому разделу. Чтобы использовать это совместно используемое устройство в другом клиентском разделе, оно должно быть прежде всего удалено из владеющего им в настоящее время раздела и переназначено разделу, который будет его использовать. Это является преимуществом перед динамическим LPAR, так как вам не нужно вручную перемещать адаптер этого устройства.

Подробнее о совместном использовании оптических устройств см. в разделе 3.9 «Ознакомление с виртуальным SCSI».

3.1.2. Переход на резерв с помощью общего Ethernet-адаптера

В версии 1.2 сервера Virtual I/O Server вводится новый способ конфигурирования резервных серверов VIOS, обеспечивающий более высокую доступность для внешнего сетевого доступа через совместно используемые адаптеры – Shared Ethernet Adapter (SEA).

Переход на резерв с помощью SEA (Shared Ethernet Adapter failover) обеспечивает избыточность путем конфигурирования резервного SEA в другом разделе сер-

вера ViIOS, и этот адаптер может быть использован, если основной SEA отказывает. Подключение к внешней сети клиентских логических разделов продолжает осуществляться без перебоев.

Конфигурирование перехода на резерв с помощью SEA может начинаться с создания виртуального адаптера с флагом доступа к внешней сети (флагом транка – trunk). Этот виртуальный адаптер должен иметь тот же PVID или VLAN ID, как и у соответствующего виртуального адаптера на резервном сервере VIOS. Он использует значение приоритета, присвоенное виртуальным Ethernet-адаптерам при их создании, чтобы определить, какой SEA будет служить в качестве основного, а какой – в качестве резервного. Адаптеру SEA, у которого виртуальный Ethernet сконфигурирован с меньшим числовым значением приоритета, будет отдаваться предпочтение при определении основного адаптера.

Чтобы вести обмен информацией друг между другом для определения, когда нужно выполнять переход на резерв, адAPTERы SEA в режиме перехода на резерв используют выделенную для такого трафика VLAN, называемую контрольным каналом (control channel). Виртуальный Ethernet (созданный с PVID, уникальным в данной системе) должен создаваться на каждом VIOS, обеспечивающем адAPTER SEA для перехода на резерв. Этот виртуальный Ethernet затем определяется как виртуальный Ethernet контрольного канала, когда каждый SEA создается в режиме перехода на резерв. С помощью контрольного канала резервный SEA определяет, когда отказывает основной адаптер, и сетевой трафик из клиентских логических разделов передается через резервный адаптер. Когда основной SEA восстанавливается после своего отказа, он снова начинает активно пропускать через себя весь сетевой трафик.

Более подробно о функции перехода на резерв SEA failover рассказывается в разделе 5.1.3 «Высокая доступность для обмена данными с внешними сетями».

3.1.3. Integrated Virtualization Manager

Интегрированный менеджер виртуализации IVM (Integrated Virtualization Manager) является базовым решением по администрированию аппаратных ресурсов, содержащимся в программном обеспечении VIO версии 1.2 и наследующим ключевые функции консоли управления оборудованием HMC (Hardware Management Console).

IVM применяется для администрирования систем System p5 с разделами с помощью веб-ориентированного графического интерфейса, не требуя HMC. При этом сокращается оборудование, необходимое для внедрения технологии виртуализации, особенно для систем начального уровня. Такое решение подходит для небольших и функционально простых сред, в которых установлено только несколько серверов и не нужны все функции HMC.

Примечание. На момент написания книги функция IVM была недоступна для разделов сервера Virtual I/O Server на системах IBM `@server` p5 моделей 570, 575, 590 и 595.

Более подробную информацию о принципах работы, установке и конфигурировании IVM можно получить в разделе 3.11 «Integrated Virtualization Manager». Еще больше можно узнать в книге *Virtual I/O Server Integrated Virtualization Manager*, REDP-4061.

3.1.4. Новые команды пула хранения

Подобно группам томов, пулы хранения данных (storage pools) являются наборами из одного и более физических томов, абстрагирующих организацию лежащих в их основе дисков. Физические тома, составляющие пул хранения, могут быть различных размеров и типов.

Использование пулов хранения не требует глубоких знаний по администрированию групп томов и логических томов для создания и назначения логического хранилища для клиентского раздела. Создаваемые с помощью пула хранения устройства не ограничиваются размерами отдельных физических томов. Пулы хранения создаются и администрируются следующими командами:

- `mksp` Создает пул хранения, используя физические тома, указанные как параметры команды.
- `chsp` Добавляет или удаляет физические тома пула хранения или устанавливает пул хранения по умолчанию.
- `lssp` Показывает информацию о пулах хранения.
- `mkbdsp` Закрепляет хранилище из пула хранения за виртуальным SCSI-адаптером.
- `rmbdsp` Удаляет хранилище из виртуального SCSI-адаптера и возвращает его в пул хранения.

Пулом хранения по умолчанию является `rootvg`. Мы рекомендуем создавать другой пул хранения, в котором вы определите ваши устройства заднего плана (backing devices), используемые в качестве виртуальных дисков в вашем клиентском разделе.

Устройства заднего плана создаются с помощью команды `mkbdsp`. Всего за один шаг вы можете создать устройство заднего плана определенного размера и связать его с адаптером виртуального SCSI-сервера, назначенного для соответствующего клиентского раздела.

Примечание. При назначении целых физических томов в качестве устройств заднего плана они не могут быть частью пула хранения. В этом случае вы должны непосредственно устанавливать связь для физического диска.

3.1.5. Улучшения HMC

Программное обеспечение HMC предоставляет улучшенный графический интерфейс для облегчения конфигурирования и обслуживания виртуальных адаптеров ввода-вывода при администрировании серверов. Эта новая функция усовершенствована с целью упрощения обслуживания среды Virtual I/O с помощью HMC начиная с версии 5.1.

Для облегчения конфигурирования виртуальных SCSI-адаптеров HMC может динамически добавлять виртуальный адаптер SCSI-сервера в VIOS при создании клиентского раздела или при добавлении нового адаптера виртуального SCSI-клиента в раздел. Это позволяет вам создавать клиентские и серверные SCSI-адаптеры за один шаг. Затем вы должны будете добавить адаптер виртуального SCSI-сервера в соответствующий профиль раздела сервера VIOS, чтобы сделать изменение постоянным.

Для облегчения обслуживания и изменения конфигурации НМС обеспечивает обзор топологий виртуального Ethernet и виртуального SCSI, сконфигурированных в сервере Virtual I/O Server.

Дальнейшую информацию и подробности конфигурирования можно получить в разделе 3.12 «Динамические операции с LPAR».

3.2. Функция Advanced POWER Virtualization

В этом разделе приведена информация о структуре пакета и способах заказа функции Advanced POWER Virtualization, доступной в системах IBM System p5 и IBM @server p5.

Функция Advanced POWER Virtualization является комбинацией аппаратной активации и программного обеспечения и содержит следующие компоненты, которые поставляются вместе как платная опция:

- ▶ Микрокодовая активация Micro-Partitioning
- ▶ Инсталляционный образ для ПО Virtual I/O Server, который поддерживает:
 - Shared Ethernet Adapter
 - сервер Virtual SCSI
 - Integrated Virtualization Manager (IVM) для поддерживаемых систем
- ▶ Partition Load Manager (поддерживается только для администрируемых НМС-систем и не является частью POWER Hypervisor и Virtual I/O Server FC 1965 в системах IBM OpenPower).

Virtual Ethernet доступен и без этой функции для серверов, присоединенных к НМС или администрируемых с помощью IVM.

Если аппаратная опция указывается при первоначальном заказе системы, то микрокод поставляется с активированной поддержкой Micro-Partitioning и Virtual I/O Server. Для заказов по модернизации компания IBM будет отправлять ключ активации микрокода (подобный ключу CUoD).

Клиенты могут посетить веб-сайт: <http://www-912.ibm.com/pod/pod> чтобы посмотреть текущие коды активации для конкретного сервера, введя тип и серийный номер машины. Код активации для функции Advanced POWER Virtualization имеет определение типа VET в окне с результатами.

Для систем, присоединенных к НМС, на рисунке 3-1 показано окно НМС, в котором вы можете активировать Virtualization Engine Technologies.

В случае использования IVM в сервере Virtual I/O Server для администрирования одной системы на рисунке 3-2 показано меню Advanced System Management Interface (ASMI) для активации Virtualization Engine Technologies. Более подробную информацию об этой процедуре вы можете получить в книге Virtual I/O Server Integrated Virtualization Manager, REDP-4061.

Virtual I/O Server и Partition Load Manager (PLM) являются лицензируемыми программными компонентами функции Advanced POWER Virtualization. Они предусматривают одну единицу оплаты за каждый установленный процессор, включая обслуживание ПО. Первоначальная оплата лицензии на ПО для Virtual I/O Server и PLM включена в цену функции Advanced POWER Virtualization.

В таблице 3-1 приведен обзор функций Advanced POWER Virtualization в системах IBM System и IBM @server p5.

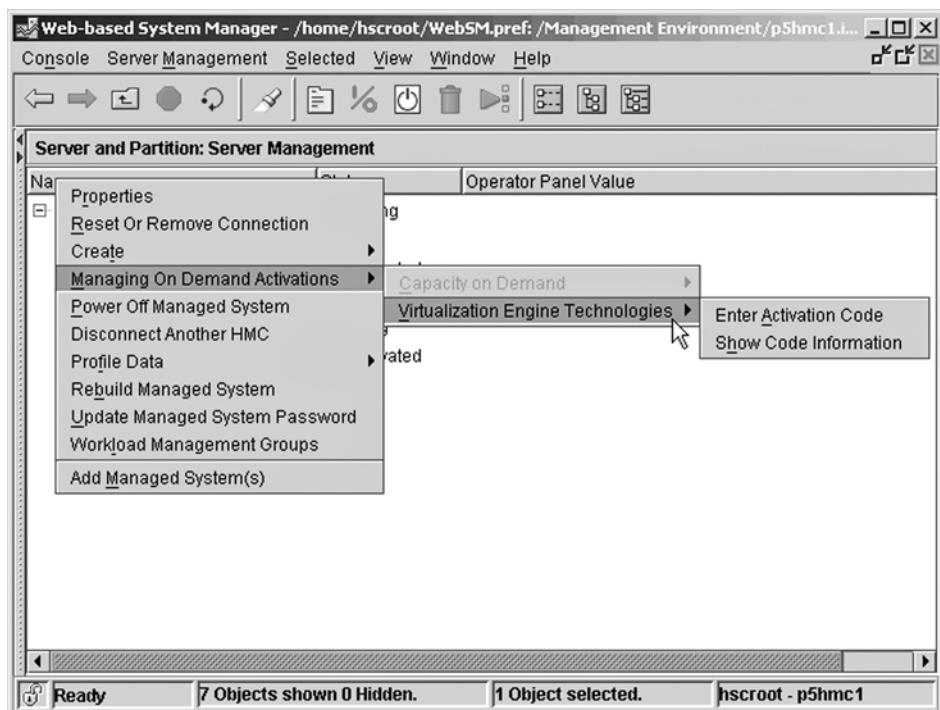


Рис. 3-1. Окно НМС для активации Virtualization Engine Technologies

Таблица 3-1. Обзор кодов функции APV

Серверы	Feature Code	Включена в базовую конфигурацию?	IVM поддерживается?
9115-505	7432	Нет	Да
9110-510	7432	Нет	Да
9123-710 ^a	1965	Нет	Да
9111-520	7940	Нет	Да
9131-52A	7940	Нет	Да
9124-720 ^a	1965	Нет	Да
9113-550	7941	Нет	Да
9133-55A	7941	Нет	Да
9117-570	7942	Нет	Нет
9118-575	7944	Нет	Нет
9119-590	7992	Да	Нет
9119-595	7992	Да	Нет

^a PLM не поставляется с FC 1965

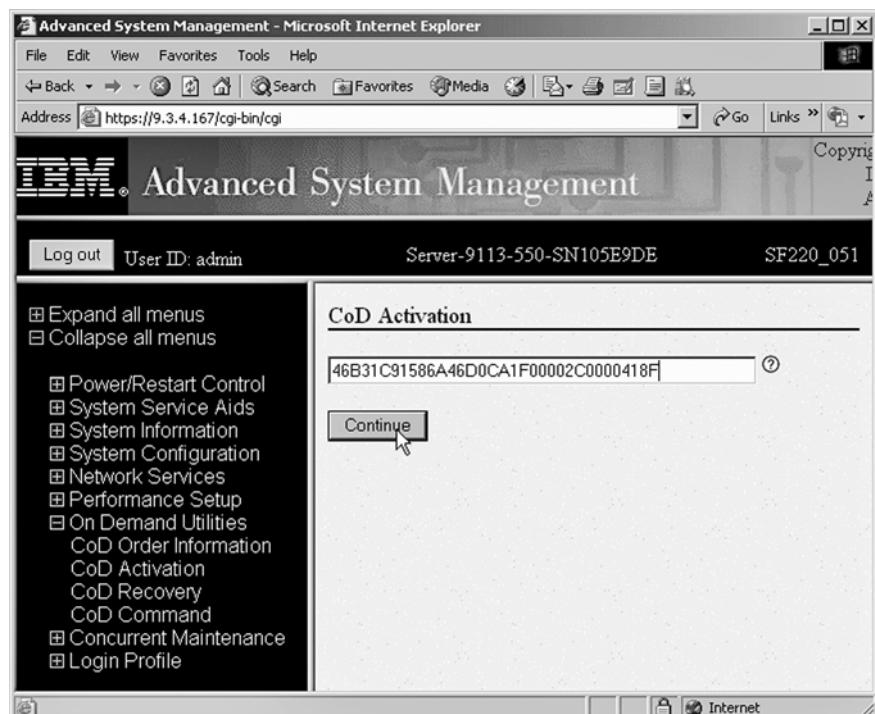


Рис. 3-2. Меню ASMI для активации Virtualization Engine Technologies

Функция Advanced POWER Virtualization конфигурируется опционально и оплачивается для всех упомянутых выше систем, кроме 9119-590 и 595, которые включают функцию Advanced POWER Virtualization как часть базовой конфигурации системы. Обслуживание ПО для всех упомянутых систем оплачивается дополнительно.

Для каждой заказанной лицензии Virtual I/O Server также подается заявка на обслуживание Software Maintenance (SWMA) сроком на один год (5771-VIO) либо на три года (5773-VIO). Вам необходимо покупать лицензию на каждый активный процессор сервера. Для присоединенной к НМС системе процессорная лицензия позволяет вам устанавливать несколько разделов Virtual I/O Server на одном сервере для обеспечения резервирования и распределения нагрузки ввода-вывода между несколькими разделами Virtual I/O Server.

Поддерживаемыми Virtual I/O-клиентами являются:

- ▶ AIX 5L Version 5.3
- ▶ SUSE LINUX Enterprise Server 9 for POWER
- ▶ Red Hat Enterprise Linux AS 3 for POWER (обновление 2 или более позднее)
- ▶ Red Hat Enterprise Linux AS 4 for POWER или более поздний

Virtual I/O Server обеспечивает клиентским разделам (Linux или AIX 5L) виртуальный SCSI-сервер и функцию виртуального ввода-вывода Shared Ethernet Adapter и интерфейс администрирования IVM для систем без НМС. Этот раздел

POWER5 не предназначен для работы приложений конечного пользователя или для входа пользователя в систему.

Для каждой заказанной лицензии на Partition Load Manager V1.1 (5765-G31) также необходимо подавать заявку на обслуживание Software Maintenance (SWMA) сроком на один год (5771-PLM) или на три года (5773-PLM). Цена обслуживания ПО для Partition Load Manager устанавливается с учетом каждого процессора, на основе процессорной группы (класса сервера).

Partition Load Manager for AIX 5L помогает клиентам максимально увеличить использование ресурсов процессоров и памяти в серверах IBM System p5, IBM @server p5 и IBM @server pSeries, поддерживающих работу с динамическими логическими разделами. В пределах ограничений, устанавливаемых политикой пользователя, ресурсы автоматически перемещаются из разделов с меньшими потребностями в разделы с высокими потребностями. Ресурсы, которые могли бы в другом случае остаться неиспользованными, теперь могут использоваться более полно.

3.3. Введение в микроразделы

Micro-Partitioning представляет собой возможность разделить вычислительную мощность физического процессора на доли в виде процессорных единиц и распределить ее между несколькими логическими разделами. Эта функция является опциональной, и для нее вы должны получать и вводить код активации для большинства моделей IBM System и IBM @server, кроме p5-590 и p5-595, в которых она автоматически включается в конфигурацию.

Преимуществом Micro-Partitioning является то, что эта функция позволяет увеличить общее использование ресурсов ЦП в пределах администрируемой системы. Более высокая степень разбиения при размещении ресурсов ЦП в логическом разделе означает эффективное использование вычислительной мощности.

В этом разделе обсуждаются следующие темы, касающиеся Micro-Partitioning:

- ▶ Разделы с общими процессорами
- ▶ Обзор общего процессорного пула
- ▶ Нарашивание ресурсов «по требованию» (CUoD)
- ▶ Динамическое освобождение процессоров и резервирование процессоров
- ▶ Динамические разделы
- ▶ Учитываемые факторы

3.3.1. Разделы с общими процессорами

При виртуализации физических процессоров в системах POWER5 вводится уровень абстракции, реализуемый в аппаратном микрокоде. Для операционной системы виртуальный процессор является тем же самым, что и физический процессор.

Ключевым преимуществом реализации разделов аппаратным способом является возможность для любой операционной системы работать с технологией POWER5 с малыми модификациями или вообще без них. Опционально, для оптимизации производительности, операционная система может быть дополнена возможностью более глубокого использования процессорных пулов, например,

с помощью добровольного возврата тактов ЦП в аппаратные ресурсы, когда в них нет необходимости. AIX 5L Version 5.3 является первой версией AIX 5L, содержащей такие усовершенствования.

Функция Micro-Partitioning позволяет нескольким разделам совместно использовать один физический процессор. Логические разделы, использующие Micro-Partitioning, называются разделами, работающими на общих процессорах (shared processor partitions).

Один раздел может быть определен с процессорной мощностью, измеряемой в виде десятой части (0,10) процессорной единицы. Она представляет собой 1/10 часть физического процессора. Каждый процессор может совместно использоваться максимум десятью разделами с общим процессором. Диспетчеризация и выделение времени разделам с общим процессором происходит на физическом процессоре под управлением гипервизора POWER.

Micro-Partitioning поддерживается во всей серии продуктов POWER5, от самых базовых и до самых совершенных систем. В таблице 2-1 показано максимальное количество логических разделов и разделов с общими процессорами, поддерживаемых в различных моделях.

Таблица 3-2. Обзор Micro-Partitioning

Сервер/Модель	505/510/520/52A	550	55A	570	575	590	595
Процессоры	2	4	8	16	16	32	64
Разделы с выделенным процессором	2	4	8	16	16	32	64
Разделы с общим процессором	20	40	80	160	160	254	254

Важно подчеркнуть, что хотя указанные максимальные значения поддерживаются аппаратурой, но практические пределы, основанные на производственных запросах нагрузки, могут быть существенно ниже.

Разделам с общим процессором по-прежнему требуется выделенная память, но необходимые этим разделам ресурсы ввода-вывода могут обеспечиваться с помощью виртуального Ethernet и виртуального SCSI. При использовании всех функций виртуализации в настоящее время можно поддерживать до 254 разделов с общим процессором.

Разделы с общим процессором создаются и администрируются с помощью НМС. Приступая к созданию раздела, вы должны выбрать между разделом с общим процессором и разделом с выделенным процессором.

При настройке раздела вам нужно определить принадлежащие ему ресурсы, то есть ресурсы ввода-вывода и памяти. Для разделов с общим процессором вам необходимо сконфигурировать следующие дополнительные параметры:

- ▶ Минимальное, желаемое и максимальное количество единиц процессорной мощности
- ▶ Режим совместного использования процессора – с верхним пределом (capped) или без него (uncapped)
- ▶ Минимальное, желаемое и максимальное количество виртуальных процессоров

Эти настройки являются темой следующих разделов.

Единицы процессорной мощности

Процессорная мощность может конфигурироваться долями по 1/100 от мощности процессора. Минимальное количество процессорной мощности, которое можно назначить разделу, равно 1/10 от мощности процессора.

В НМС процессорная мощность определяется в процессорных единицах (processing unit, PU). Минимальная мощность в 1/10 от мощности процессора обозначается как 0,1 процессорной единицы. Для назначения процессорной мощности в 75% от мощности процессора в НМС определяется значение в 0,75 процессорных единиц.

В системе с двумя процессорами разделу может быть назначено максимум 2,0 процессорных единиц. Процессорные единицы, заданные в НМС, используются при количественном определении минимального, желаемого и максимального объема процессорной мощности для раздела.

После активации раздела процессорная мощность обычно называется выделенной, или назначенной, мощностью (capacity entitlement – CE, entitled capacity – EC).

Режимы с верхним пределом и без верхнего предела

В функции микроразделов имеется специальный процессорный режим, определяющий максимум процессорной мощности, предоставляемой микроразделам из общего процессорного пула. Такими процессорными режимами являются:

Режим с верхним пределом (Capped mode)	Количество процессорных единиц, предоставляемых разделу одновременно, никогда не превышает гарантированной процессорной мощности (выделенная мощность гарантируется системой, но ее нельзя превысить даже при наличии ресурсов в общем процессорном пуле).
Режим без верхнего предела (Uncapped mode)	Процессорная мощность, предоставляемая разделу одновременно, может превышать гарантированную процессорную мощность, когда в общем процессорном пуле имеются ресурсы. В режиме без верхнего предела вам необходимо определять вес такого раздела.

Если свободные процессорные единицы требуются нескольким логическим разделам без верхнего предела, то администрируемая система распределяет свободные процессорные единицы между логическими разделами пропорционально весу каждого логического раздела в режиме без верхнего предела. Чем выше значение такого веса у логического раздела, тем больше процессорных единиц он получает.

Вес в режиме без верхнего предела задается целым числом от 0 до 255. По умолчанию значение этого веса для логических разделов без верхнего предела устанавливается равным 128. Доля раздела вычисляется делением его весового значения переменной мощности на сумму весовых значений переменной мощности всех разделов, не имеющих верхнего предела. Если вы установите вес в режиме без верхнего предела равным 0, то администрируемая система будет считать этот логический раздел логическим разделом с верхним пределом.

Логический раздел с нулевым весом в режиме без верхнего предела не может использовать процессорных единиц больше, чем назначено для данного логического раздела.

Вес 0 позволяет автоматически функционирующему ПО обеспечивать эквивалентную функцию в виде динамической операции с LPAR по изменению режима без верхнего предела на режим с верхним пределом.

Виртуальные процессоры

Виртуальный процессор является отображением, или представлением, физического процессора для операционной системы раздела, использующего общий процессорный пул. Предоставленная разделу процессорная мощность, является ли она целой процессорной единицей или ее долей, будет распределяться микроподходом сервера равномерно между виртуальными процессорами, чтобы поддерживать нагрузку. Например, если у логического раздела есть 1,60 процессорных единиц и два виртуальных процессора, то каждый виртуальный процессор будет иметь 0,80 процессорных единиц.

Выбор оптимального количества виртуальных процессоров зависит от нагрузки в разделе. Некоторые разделы выигрывают от большего параллелизма, в то время как другим необходимо больше вычислительной мощности.

По умолчанию задаваемое вами количество процессорных единиц округляется до минимального количества виртуальных процессоров, необходимых для обеспечения назначенного количества процессорных единиц. Настройки по умолчанию поддерживают баланс между виртуальными процессорами и процессорными единицами. Например:

- ▶ Если вы зададите 0,50 процессорных единиц, то будет назначен один виртуальный процессор.
- ▶ Если вы зададите 2,25 процессорных единиц, то будут назначены три виртуальных процессора.

Вы также можете воспользоваться вкладкой Advanced в профиле вашего раздела, чтобы изменить конфигурацию, определенную по умолчанию, и назначить большее количество виртуальных процессоров.

В общем процессорном пуле раздел будет иметь виртуальных процессоров не меньше назначенной ему процессорной мощности. Если вы сделаете количество виртуальных процессоров слишком малым, то ограничите процессорную мощность раздела без верхнего предела. Если у вашего раздела 0,50 процессорных единиц и один виртуальный процессор, то раздел не может превысить 1,00 процессорных единиц, так как он может выполнять одновременно только одно задание, которое не может превысить 1,00 процессорных единиц. Но если тому же разделу с 0,50 процессорными единицами было назначено два виртуальных процессора и процессорные ресурсы являются доступными, то раздел может использовать дополнительно еще 1,50 процессорных единиц.

Минимальное количество процессорных единиц, которые вы можете иметь для каждого виртуального процессора, зависит от модели сервера. Максимальное количество процессорных единиц, которые вы можете иметь для каждого виртуального процессора, всегда равно 1,00. Это означает, что логический раздел не может использовать процессорных единиц больше, чем ему назначено виртуальных процессоров, даже если логический раздел не имеет верхнего предела.

Свертывание виртуальных процессоров

Начиная с пакета обновлений maintenance level 3, AIX 5L V5.3 обеспечивает улучшенное администрирование виртуальных процессоров. Эта функция заключается в лучшем использовании общего процессорного пула с помощью минимизации задействования виртуальных процессоров, пристаивающих большую часть времени. Важным преимуществом данной функции является улучшенное определение сходства процессоров (processor affinity) в условиях большого количества длительно пристаивающих разделов с общими процессорами, и это приводит к эффективному использованию процессорных циклов. Средний цикл диспетчеризации виртуальных процессоров увеличивается, и достигается лучшее использование кэшей и снижение нагрузки на гипервизор.

Функция свертывания виртуальных процессоров (virtual processor folding) характеризуется следующим:

- ▶ Пристыкающие виртуальные процессоры не удаляются динамически из раздела. Они переводятся в «спящее», или неактивное, состояние и активируются только при возрастании нагрузки.
- ▶ Выгоды от этой функции нет при полной занятости разделов.
- ▶ Если функция выключена, то все определенные для раздела виртуальные процессоры размещаются на физических процессорах.
- ▶ Виртуальные процессоры, имеющие прикрепленные ресурсы, например, по командам `bindprocessor` или `rset`, не исключаются из процесса деактивации.
- ▶ Эта функция может включаться и выключаться. По умолчанию она включена.

Когда виртуальный процессор деактивирован, то для него не планируется выполнение потоков, если только поток не связан с этим ЦП.

Примечание. В разделе с разделяемым процессором есть только один узел сходства и, следовательно, только одна глобальная очередь выполнения узла.

Настраиваемым параметром этой функции является `vpm_xvcpus`, по умолчанию равный 0, что означает включенное состояние функции. Используйте команду `schedo` для изменения этого настраиваемого параметра.

Выделенные процессоры

Выделенные процессоры – это целые процессоры, назначенные единственному разделу. Если вы выберете вариант с назначением выделенных процессоров логическому разделу, то будете должны назначить этому разделу хотя бы один процессор. В одном и том же разделе вы не можете смешивать общие и выделенные процессоры.

По умолчанию отключенный логический раздел, использующий выделенные процессоры, будет предоставлять свои процессоры общему процессорному пулу. Когда эти процессоры находятся в общем процессорном пуле, то раздел без верхнего предела, которому требуется больше процессорной мощности, может использовать эти пристыкающие процессорные ресурсы. Но при включении выделенного раздела в тот момент, когда раздел без верхнего предела использует эти процессоры, активируемый раздел получает обратно все свои процессорные ресурсы. Если вы хотите предотвратить использование выделенных процес-

соров в общем процессорном пуле, то можете отключить эту функцию в НМС, убрав флажок **Allow idle processor to be shared** в свойствах раздела.

Примечание. Опция «Allow idle processor to be shared» (Разрешение совместного использования простаивающего процессора) активирована по умолчанию. Она не является частью свойств профиля и не может изменяться динамически.

3.3.2. Обзор общего процессорного пула

Общий процессорный пул является группой физических процессоров, не выделенных ни одному логическому разделу. Технология Micro-Partitioning в сочетании с гипервизором POWER облегчает распределение процессорных единиц между логическими разделами в общем процессорном пуле.

В разделе с разделяемым процессором нет фиксированных взаимоотношений виртуальных и физических процессоров. Гипервизор POWER может использовать любой физический процессор в общем процессорном пуле при планировании размещения виртуального процессора. По умолчанию он пытается использовать тот же самый физический процессор, но это не может всегда гарантироваться. Гипервизор POWER использует для виртуальных процессоров принцип «домашнего узла», позволяющий выбирать для планируемого виртуального процессора наилучший с позиции сходства памяти (memory affinity), доступный физический процессор.

Планирование по сходству (affinity scheduling) предназначено для предохранения содержимого кэшей памяти таким образом, чтобы набор рабочих данных задачи мог читаться или записываться в самый короткий возможный период времени. Управление по сходству активно используется в гипервизоре POWER потому, что каждый раздел имеет совершенно различный контекст. В настоящее время имеется один общий процессорный пул, и все виртуальные процессоры неявно связываются с одним и тем же пулом.

На рисунке 3-3 показаны отношения между двумя разделами, использующими общий процессорный пул на одном физическом ЦП. Один раздел имеет два виртуальных процессора, а другой – один виртуальный процессор. На рисунке также показано, как выделенная мощность равномерно распределяется на нескольких виртуальных процессорах.

Когда вы устанавливаете профиль раздела, то определяете желаемое, минимальное и максимальное значения, необходимые вам для этого профиля. При запуске раздела система выбирает процессорную долю мощности раздела из этого заданного диапазона мощности. Выбранное таким образом значение представляет назначенную мощность, зарезервированную для этого раздела. Эта мощность не может быть использована для запуска другого раздела с общим процессором, то есть мощность не может быть переназначена.

При запуске раздела предпочтение отдается желаемому значению, но это значение не может использоваться всегда, так как в системе может не оказаться достаточно неназначенной мощности. В таком случае выбирается другое значение, которое должно быть больше или равно атрибуту минимальной мощности. Иначе раздел не будет запущен.

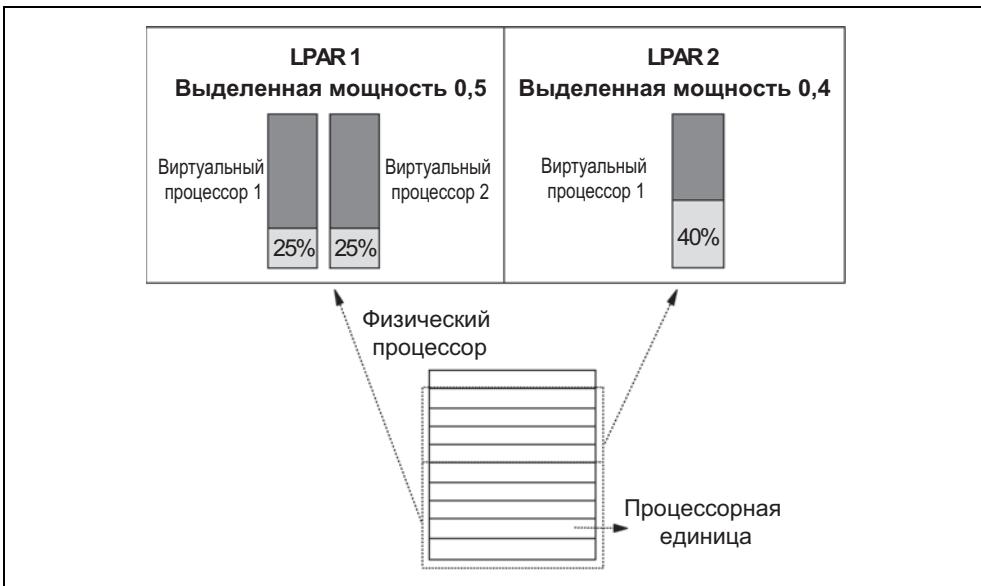


Рис. 3-3. Распределение выделенной мощности между виртуальными процессорами

Выделенная процессорная мощность распределяется между разделами в последовательности, соответствующей запуску разделов. Например, рассмотрим общий процессорный пул, имеющий 2,0 процессорные единицы.

Разделы 1, 2 и 3 активируются в такой последовательности:

- ▶ Активируется раздел 1:
Мин. = 1,0; Макс. = 2,0; Жел. = 1,5;
Выделяемая доля мощности: 1,5
- ▶ Активируется раздел 2:
Мин. = 1,0; Макс. = 2,0; Жел. = 1,0.
Раздел 2 не запускается, так как не удовлетворяется требование минимальной мощности
- ▶ Активируется раздел 3:
Мин. = 0,1; Макс. = 2,0; Жел. = 0,8.
Выделяемая доля мощности: 0,5

Максимальное значение используется только как верхний предел для динамических операций.

На рисунке 3-4 показано использование раздела общего процессорного пула с верхним пределом. Разделам, использующим режим с верхним пределом, не может назначаться из общего процессорного пула процессорная мощность, превышающая значение заданной выделенной мощности.

На рисунке 3-5 показано использование общего процессорного пула разделом без верхнего предела. Раздел без верхнего предела может назначать простаивающую процессорную мощность, если ему нужна мощность, превышающая выделенную.



Рис. 3-4. Разделы с общим процессором и верхним пределом

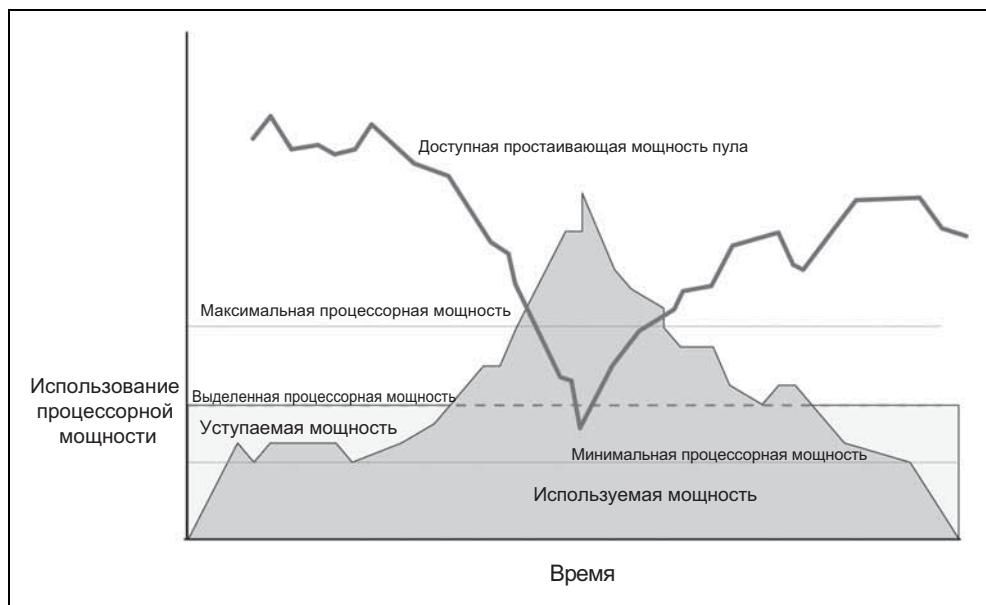


Рис. 3-5. Раздел с общим процессором и без верхнего предела

3.3.3. Наращивание ресурсов «по требованию» (CUoD)

Наращивание ресурсов по требованию – CUoD (Capacity Upgrade on Demand) увеличивает гибкость работы и конфигурирования в системах IBM System p5, IBM *@server* p5 и pSeries. Поставляемая в виде набора платных предложений, функция CUoD позволяет добавлять дополнительные ресурсы по мере необходимости. В оперативном режиме можно получать процессоры и память, чтобы удовлетворять увеличивающиеся потребности нагрузки. Если система сконфигурирована с динамическими LPAR, то это может выполняться без влияния на операции.

При активации процессора, настроенного для Capacity Upgrade on Demand в системе с определенными разделами с общими процессорами, активируемый процессор автоматически назначается в общий процессорный пул. Вы затем можете решить динамически добавить этот процессор в раздел с выделенным процессором или динамически добавить выделенную мощность в разделы с общим процессором.

Чтобы удалить процессор с функцией Capacity Upgrade on Demand (например, при использовании функции On/Off Capacity Upgrade on Demand – «включение-выключение ресурсов по требованию» для временной активации процессоров пользователями), вы должны перед деактивацией процессора убедиться в достаточноном количестве остающихся процессорных единиц. Вы можете динамически удалять необходимую выделенную мощность из разделов.

Разновидностью Capacity Upgrade on Demand является функция Reserve CUoD – «резервное наращивание ресурсов по требованию». Она представляет собой *автономный* способ активации временно используемых ресурсов. Reserve CUoD позволяет вам помещать некоторое количество неактивных процессоров в общий процессорный пул сервера, и они затем становятся доступными менеджеру ресурсов пула. Когда сервер обнаруживает, что базовые (оплаченные/активные) процессоры в разделах без верхнего предела используются на 100% и необходимо не менее 10% дополнительного процессора, то на баланс счета Reserve CUoD выставляется оплата «процессорного дня» – *Processor Day* (действующего в течение 24 часов). Следующий «процессорный день» будет оплачиваться для каждого дополнительного процессора, предоставленного в пользование на основании правила 10%. По истечении 24-часового периода и при отсутствии дальнейшей необходимости в дополнительной мощности «процессорные дни» не будут выставляться к оплате до следующего пика потребления производительности.

3.3.4. Динамическое освобождение процессоров и резервирование процессоров

Если количество сбоев физического процессора в общем процессорном пуле достигло порогового значения и этот процессор нуждается в выводе из системы (блокировке), то гипервизор POWER проанализирует среду системы для определения действия, которое следует предпринять для замены этого процессорного ресурса. Для выхода из этой ситуации возможны следующие варианты действий:

- ▶ Если есть доступный CUoD-процессор, то гипервизор POWER будет прозрачно подключать этот процессор к общему процессорному пулу и не будет происходить потеря мощности разделами.

- Если есть не менее 1,0 доступной неназначенной процессорной мощности, то она может использоваться для замены мощности, потерянной вследствие отказа процессора.

Если нет неназначенных ресурсов достаточного объема, то гипервизор POWER будет определять, сколько должен потерять мощности каждый раздел, чтобы исключить 1,00 процессорных единиц из общего процессорного пула. После того как каждый раздел отдаст процессорную мощность и виртуальные процессоры, отказавший процессор будет выведен из системы сервисным процессором и гипервизором.

Объем ресурсов, запрашиваемых у каждого микрораздела, пропорционален общему объему выделенной мощности в разделе. Этот процесс основан на объеме ресурсов, которые могут быть отданы, и контролируется минимальной процессорной мощностью раздела, определенной в атрибуте *min* в профиле раздела.

3.3.5. Динамические разделы

Разделы с AIX 5L версии 5.2 поддерживаются на серверах с выделенными процессорами. Они также поддерживают динамическое перемещение следующих ресурсов:

- Одного выделенного процессора
- Области памяти 256 МБ
- Одного слота адаптера ввода-вывода
- Раздел с AIX 5L версии 5.3 состоит из выделенных процессоров или общих процессоров конкретной выделенной мощности, работающих в режиме с верхним пределом или без верхнего предела, выделенной области памяти и слотов виртуальных или физических адаптеров ввода-вывода. Все эти ресурсы могут быть динамически изменены.

Для разделов с выделенными процессорами можно только динамически добавлять, перемещать или удалять целые процессоры. Когда вы динамически удаляете процессор из раздела с выделенными процессорами, то он затем назначается в общий процессорный пул.

Для разделов с разделяемыми процессорами также динамически можно:

- Удалять, перемещать или добавлять процессорную мощность.
- Изменять вес атрибута режима без верхнего предела.
- Добавлять и удалять виртуальные процессоры.
- Изменять режимы процессоров между режимами с верхним пределом и без верхнего предела.

3.3.6. Учитываемые факторы

Необходимо учитывать следующие факторы при реализации разделов с общими процессорами:

- Минимальный размер раздела с общим процессором равен 0,1 процессорных единиц физического процессора. Поэтому количество разделов с общим процессором, которые вы можете создавать для системы, зависит главным образом от количества процессоров в системе.

- ▶ Максимальное количество разделов в сервере равно 254.
- ▶ Максимальное количество виртуальных процессоров в разделе равно 64.
- ▶ Минимальное количество процессорных единиц, которые вы можете иметь в каждом виртуальном процессоре, зависит от модели сервера. Максимальное количество процессорных единиц, которые вы можете иметь для каждого виртуального процессора, всегда равно 1,00. Это означает, что логический раздел не может использовать процессорных единиц больше, чем количество назначенных виртуальных процессоров, даже если логический раздел не имеет верхнего предела.
- ▶ Смесь выделенных и общих процессоров в одном и том же разделе не поддерживается.
- ▶ Если вы динамически удаляете виртуальный процессор, то у вас нет возможности задать данные для идентификации именно того виртуального процессора, который нужно удалить. Удаляемый виртуальный процессор будет определять операционная система.
- ▶ Для общих процессоров администрирование в AIX 5L по сходству можно считать бесполезным. AIX 5L будет продолжать использовать информацию домена по сходству, предоставляемую микрокодом, для построения связей виртуальных процессоров и памяти и будет продолжать оказывать предпочтение последнему работающему процессору при повторной диспетчеризации потока.
- ▶ Раздел без верхнего предела с весом 0 имеет то же влияние на производительность, что и раздел с верхним пределом. НМС может динамически изменять либо вес, либо режим раздела – с режима с верхним пределом на режим без верхнего предела.

Диспетчеризация виртуальных процессоров

Существуют дополнительные вычислительные функции, связанные с обслуживанием онлайновых виртуальных процессоров, поэтому вам следует внимательно учитывать их потребности в ресурсах, прежде чем выбирать значения для таких атрибутов.

В системе AIX 5L V5.3 ML3 введена новая функция для облегчения администрирования виртуальных процессоров.

В процессе планирования работы виртуальных процессоров им определяется задержка диспетчеризации. Когда виртуальный процессор переводится в рабочее состояние, то он помещается гипервизором POWER в очередь выполнения и ждет указания диспетчера. Время между этими двумя событиями называется задержкой диспетчеризации.

Задержка диспетчеризации виртуального процессора зависит от выделенной мощности раздела и количества виртуальных процессоров, находящихся в онлайновом режиме этого раздела. Выделенная мощность равномерно распределяется между этими онлайновыми виртуальными процессорами, и количество онлайновых виртуальных процессоров влияет на длительность диспетчеризации каждого виртуального процессора. Чем меньше цикл диспетчеризации, тем большая задержка диспетчеризации.

На время написания книги задержка диспетчеризации для наихудшего случая была равна 18 миллисекундам, а минимальный цикл диспетчеризации, поддер-

живаемый на уровне виртуальных процессоров, равнялся одной миллисекунде. Эта задержка основывается на минимальной выделенной мощности раздела в 1/10 мощности физического процессора и периоде ротации в 10 миллисекунд диспетчерского «колеса» гипервизора. Можно легко зритально представить, что работа виртуального процессора планируется на выполнение в первой и последней частях двух 10-миллисекундных интервалов. В общем случае, если эти задержки слишком велики, клиенты могут увеличить выделенную мощность, минимизировать количество онлайновых виртуальных процессоров без уменьшения выделенной мощности или использовать разделы с выделенными процессорами.

Количество виртуальных процессоров

Обычно значение атрибутов минимальных, желаемых и максимальных виртуальных процессоров каким-то образом должно соответствовать значениям атрибутов минимальной, желаемой и максимальной мощности. Для разделов без верхнего предела должен быть определен какой-то допуск, так как им разрешено потреблять мощность, превышающую их выделенную мощность.

Если раздел не имеет верхнего предела, то администратор может захотеть определить атрибуты желаемого и максимального количества виртуальных процессоров, превышающие соответствующие атрибуты выделенной мощности. Точное значение специфично для конкретной системы, но назначение на 50–100 % больше, чем в режиме с ограничением, представляется разумным.

В таблице 3-3 приведено несколько таких разумных настроек количества виртуальных процессоров и процессорных единиц для режимов с верхним пределом и без него.

Таблица 3-3. Разумные настройки для разделов с разделяемыми процессорами

Мин. кол. VP ^a	Жел. кол. VP	Макс. кол. VP	Мин. PU ^b	Жел. PU	Макс. PU	Capped
1	2	4	0,1	2,0	4,0	Да
1	3 или 4	6 или 8	0,1	2,0	8,0	Нет
2	2	6	2,0	2,0	6,0	Да
2	3 или 4	8 или 10	2,0	2,0	10,0	Нет

а – Виртуальные процессоры

б – Процессорные единицы

Взаимоотношения виртуального и физического процессоров

В разделе с общим процессором нет фиксированных взаимоотношений между виртуальным и физическим процессорами. Гипервизор POWER будет пытаться использовать физический процессор с тем же сходством памяти, как и у виртуального процессора, но это не гарантируется. У виртуального процессора есть понятие «домашнего» физического процессора. Если гипервизор не может найти физический процессор с тем же сходством памяти, то он постепенно расширяет границы поиска и включает в него процессоры с меньшим сходством памяти, пока не найдет тот, который сможет использовать. Из этого следует предполо-

жение, что сходство памяти будет меныши в разделах с разделяемыми процессорами.

Также предполагается, что изменчивость нагрузки будет увеличиваться в разделах с разделяемыми процессорами, так как здесь существуют задержки, связанные с планированием виртуальных процессоров и прерываниями. Также может увеличивать изменчивость и многопоточная обработка SMT, так как она добавляет еще один уровень разделения ресурсов, способный приводить к ситуации, когда один поток мешает выполнению другого потока.

Следовательно, если приложение чувствительно к использованию кэша или не выдерживает изменчивости нагрузки, то его следует развертывать в разделе с выделенным процессором и отключенной SMT. В разделах с выделенными процессорами одному разделу назначается целый процессор. Процессоры не используются совместно с другими разделами и планируются гипервизором POWER. Разделы с выделенными процессорами должны явно создаваться системным администратором с помощью консоли Hardware Management Console.

Данные сходства процессоров и памяти обеспечиваются только в разделах с выделенными процессорами. В разделе с общим процессором считается, что все процессоры имеют одно и то же сходство. Информация сходства предоставляется через API-интерфейсы RSET.

3.4. Ознакомление с процессором POWER5

POWER5 является процессором архитектуры POWER самого последнего поколения. Он надстраивает архитектуру POWER4 и является совместимым с ней, предлагая дополнительные функции и заметное увеличение производительности. Вот некоторые из многочисленных источников с описанием процессора POWER5:

- ▶ <http://www.ibm.com/servers/@server/pseries/news/related/2004/m2040.pdf>
- ▶ <http://researchweb.watson.ibm.com/journal/abstracts/rd/494/sinharoy.html>
- ▶ *IBM @server p5 520 Technical Overview and Introduction*, REDP-9111

POWER5 изготавливается по 0,13-микронной технологии Copper SOI в 8-слойном процессе. На площади чуть меньше 400 кв. миллиметров размещается более 275 миллионов транзисторов.

Каждый чип POWER5 содержит два независимых ядра, каждое из которых способно управлять двумя параллельными аппаратными потоками, и это делает каждый чип четырехканальным симметричным мультипроцессором с точки зрения операционной системы. В архитектуре POWER5 используются 120 регистров с плавающей точкой, 120 целочисленных регистров и восемь блоков исполнения.

В реализации POWER5 поддерживается до 2 ТБ физической памяти со встроенным контроллером памяти и используются три уровня кэширования:

- | | |
|------------------|--|
| Уровень 1 | 64 КБ, 2-канальный ассоциативный кэш с множественным доступом для команд и 32 КБ, 4-канальный ассоциативный кэш с множественным доступом для данных. |
| Уровень 2 | Совместно используется двумя ядрами чипа. 1,9 МБ, 10-канальный ассоциативный кэш с множественным доступом (комбинированный, для команд и данных). |

Уровень 3 Кэш L3 находится вне чипа, но его контроллер и дескрипторы находятся в чипе. 36 МБ, 12-канальный ассоциативный кэш с множественным доступом. Кэш L3 также совместно используется двумястроенными в чип ядрами, является исключением L2. Кэш L3 имеет собственную шину, работающую с частотой, в два раза меньшей тактовой частоты ядра.

Обмен чипа с чипом, чипа с памятью и чипа с вводом-выводом осуществляется по своим отдельным специализированным шинам.

Последняя версия процессора POWER5 называется POWER5+. Процессор POWER5+ имеет более высокую производительность, чем его предшественник, и в нем использована 90-нанометровая технология. Эти процессоры имеют следующие различия:

- ▶ POWER5+ поддерживает два новых размера страниц: размер страницы 64 КБ и размер страницы 16 ГБ в дополнение к размерам страниц 4 КБ и 16 МБ, поддерживаемым предыдущими процессорами POWER.
- ▶ POWER5+ поддерживает сегменты размером 1 ТБ.
- ▶ POWER5+ поддерживает несколько размеров страниц для одного сегмента.
- ▶ POWER5+ имеет в два раза больший буфер быстрого преобразования адреса (translation look-aside buffer, TLB), чем POWER5.
- ▶ Введены четыре новые команды для целочисленного округления данных с плавающей точкой в одной операции: обычное округление – traditional round (в меньшую сторону, если разряд меньше 0,5, иначе – в большую сторону); наименьшее целое, большее, чем разряд – ceiling (округление всегда в большую сторону); наибольшее целое, не превышающее разряд – floor (округление всегда в меньшую сторону), и отбрасывание разряда – truncate.
- ▶ Усовершенствованы команды `lwarx` и `stwcx`.
- ▶ В блоке наблюдения за производительностью Performance Monitor Unit (PMU) добавлены новые события для новых размеров страниц и сегментов.
- ▶ Добавлены четыре очереди чтения контроллера памяти.
- ▶ Возможна запись в кэш за один цикл данных размером менее 64 байт.
- ▶ Усовершенствования кэша L2: в L2 процессора POWER5 используются буфера хранения для данных размером в половину строки, а в POWER5+ используются буфера хранения полной строки.
- ▶ Активирована упаковка очередей (FPU).
- ▶ Обеспечиваются более высокие частоты процессора: в POWER5 используется 9S2 CMOS, а в POWER5+ используется 10S CMOS. Первоначально тактовая частота процессора POWER5+ будет равна 1,9 ГГц.

3.5. Начальные сведения об одновременной многопоточной обработке (SMT)

Обычные процессоры выполняют команды из одного потока команд, и, несмотря на прогресс в микропроцессорных архитектурах, использование блоков исполнения современных микропроцессоров остается на низком уровне. Обычной картиной во многих средах является средняя степень использования блоков исполнения, равная приблизительно 25 процентам.

Реализованная в процессоре POWER5 технология SMT (simultaneous multithreading) позволяет выбирать команды более чем из одного потока. Отличительной чертой этой реализации является способность планировать команды на исполнение параллельно из всех потоков. С помощью SMT система динамически подстраивается к среде, позволяя командам выполнятся по возможности из каждого потока, а командам из одного потока – использовать все блоки исполнения, если в другом потоке возникает событие с длительной задержкой.

3.5.1. Режим SMT процессора POWER5

В режиме SMT процессор POWER5 использует два отдельных программных счетчика, по одному для каждого потока. Команды выбираются поочередно из этих двух потоков. Два потока совместно используют кэш команд.

Но не всем приложениям выгодно использование SMT. По этой причине в процессоре POWER5 поддерживается режим однопоточного исполнения single-threaded (ST). В этом режиме процессор POWER5 отдает все ресурсы физического процессора активному потоку. Процессор POWER5 использует только один программный счетчик и в каждом цикле выбирает команды для этого потока.

В AIX 5L V5.3 есть возможность динамически переключаться между режимами ST и SMT (без перезагрузки). Разделы Linux требуют перезапуска для изменения режима SMT.

Наибольшая выгода от SMT достигается при наличии многочисленных параллельно исполняемых потоков, как правило, в коммерческих средах – например, в веб-сервере или сервере базы данных. Нагрузки с интенсивным обменом данными, с высокой степенью использования памяти или с однопоточными высокопроизводительными вычислениями обычно лучше работают в режиме ST.

3.5.2. SMT и AIX 5L

Планировщик операционной системы AIX 5L распределяет исполнение потоков в логических процессорах. В выделенном и виртуальном процессорах имеются один или два логических процессора, в зависимости от того, активирован режим SMT или нет. Если SMT включен, то оба логических процессора всегда находятся в одном и том же разделе. На рисунке 3-6 показаны взаимоотношения между физическими, выделенными и логическими процессорами для разделов с выделенным и общим процессорами. Возможна ситуация, когда одновременно некоторые разделы с общим процессором будут в режиме SMT, а другие – с отключенным SMT.

Управление SMT в AIX 5L

Активация SMT управляется в AIX 5L командой `smtctl` или с помощью SMIT. Режим SMT может включаться или выключаться динамически в логическом разделе или при очередной перезагрузке операционной системы.

Установка режима SMT из командной строки

Команда `smtctl` должна запускаться пользователями с правами root.

С командой `smtctl` связаны два флага `-m` и `-w`, которые определяются следующим образом:

- `-m off` Режим SMT будет выключен.
- `-m on` Режим SMT будет включен.

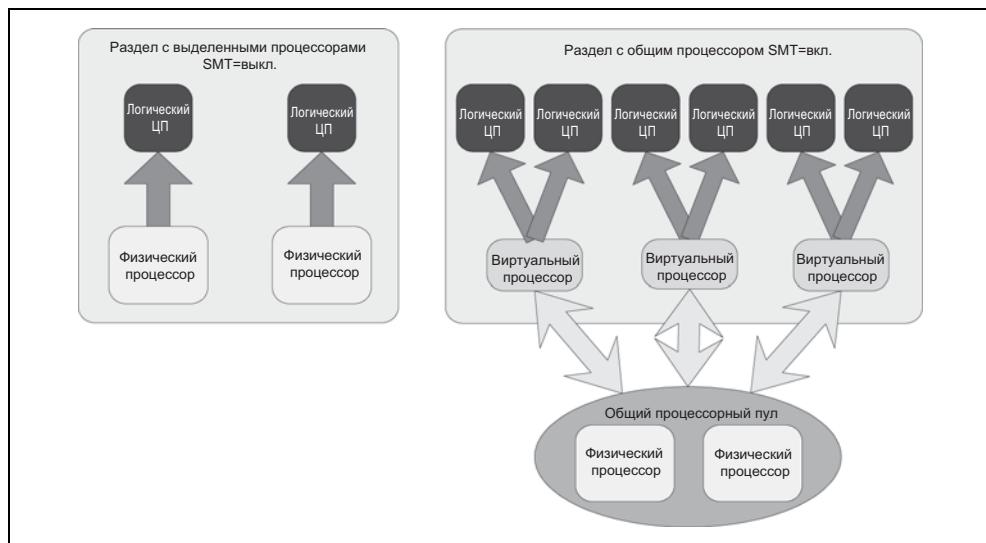


Рис. 3-6. Физические, виртуальные и логические процессоры

- w boot Вводит в действие изменение режима SMT при очередной и последующих перезагрузках.
- w now Немедленно вводит в действие изменение режима, но не сохраняется при перезагрузке.

Команда `smtctl` не перестраивает загрузочный образ. Если вы хотите изменить режим SMT, установленный по умолчанию в AIX 5L, то для перестройки загрузочного образа должна использоваться команда `bosboot`. Загрузочный образ в AIX 5L версии 5.3 был расширен и включает индикатор, управляющий режимом SMT, установленным по умолчанию.

Примечание. Если не введен ни флаг `-w boot`, ни флаг `-w now`, то изменение режима происходит немедленно и будет сохраняться при перезагрузках. Загрузочный образ должен быть изменен с помощью команды `bosboot` для сохранения изменения режима при последующих перезагрузках, независимо от использования флага `-w`.

Команда `smtctl`, введенная без флага, будет показывать текущее состояние SMT в разделе. Вот пример команды `smtctl`:

```
# smtctl
This system is SMT capable.
SMT is currently enabled.
SMT boot mode is set to enabled.
Processor 0 has 2 SMT threads
SMT thread 0 is bound with processor 0
SMT thread 1 is bound with processor 0
```

Установка режима SMT с помощью SMIT

Используйте быстрый путь `smitty smt` для входа в панель управления SMIT SMT. Из главной панели SMIT последовательность выбора выглядит следующим образом: **Performance & Resource Scheduling** → **Simultaneous Multi-Threading Mode** → **Change SMT Mode**. На рисунке 3-7 показана панель SMIT SMT.

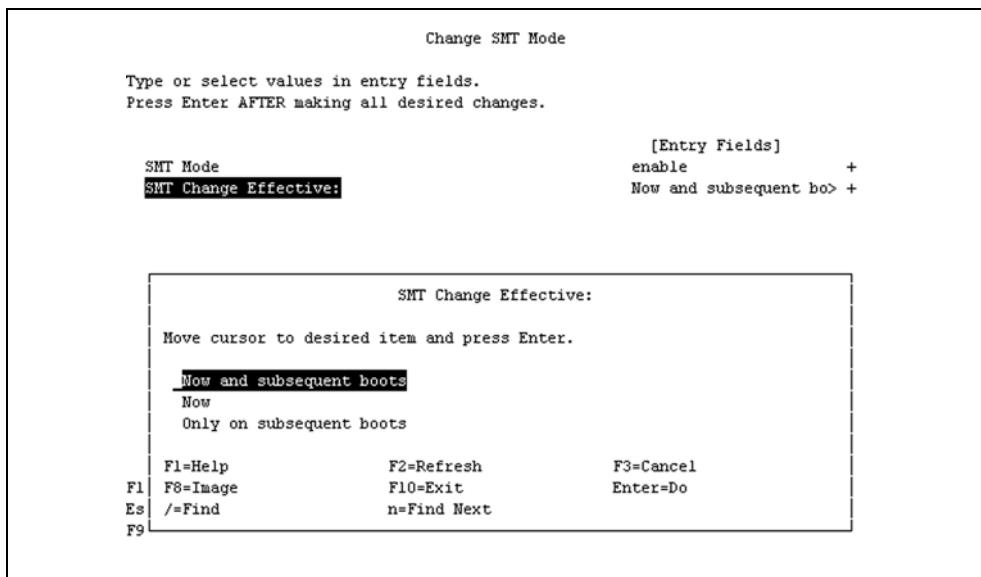


Рис. 3-7. Панель SMIT SMT с опциями

Мониторинг и настройка производительности режима SMT

В AIX 5L версии 5.3 имеются команды или расширенные опции к существующим командам для мониторинга и настройки системных параметров режима SMT.

Мониторинг SMT

Работа режима SMT требует от операционной системы обеспечения статистических данных об использовании конкретных логических процессоров. Команда `mpstat` используется для отображения статистики о производительности всех логических процессоров, работающих в логическом разделе. Команда `mpstat` описывается в подразделе «Инструменты логического процессора» раздела 6.5.2.

Настройка SMT

Для настройки SMT в AIX 5L версии 5.3 был добавлен настраиваемый параметр к команде `schedo`. Параметр `smt_snooze_delay` может использоваться для определения времени, в течение которого поток будет работать в холостом цикле перед посылкой системного вызова `h_cede` гипервизору POWER.

В разделах с выделенными процессорами процессор перейдет в режим ST, если при активном состоянии двух потоков гипервизор получает вызов `cede` из операционной системы для одного из потоков. Неактивный поток будет переведен в «спящее» состояние, и будет исключаться любая дополнительная обработка,

вводимая SMT, что позволит оставшемуся активному потоку выполняться быстрее. Если smt_snooze_delay установлен в -1, то поток не будет уступать ресурсы при ожидании в холостом цикле. Если параметр snooze delay установлен в 0, то поток будет уступать ресурсы немедленно по вхождении в холостое ожидание. Если от последнего активного потока получен запрос об уступке – вызов cede, то процессор делается при необходимости доступным для циклов гипервизора или немедленно возвращается операционной системе.

Для разделов с общими процессорами, когда активными являются несколько потоков и гипервизору посыпается системный вызов cede, пославший этот вызов поток переводится в «спящее» состояние. Если вызов cede поступает от последнего активного потока, то этот процессор считается свободным.

Чтобы максимально улучшить время отклика¹, установите smt_snooze_delay в 0; чтобы максимально увеличить пропускную способность, установите его в -1. Значением по умолчанию является 0.

3.5.3. Управление SMT в Linux

Для включения или выключения SMT при загрузке используйте следующую опцию загрузки в командной строке приглашения загрузки:

boot: linux smt-enabled=on

Измените on на off (выключено), чтобы деактивировать SMT при загрузке. По умолчанию SMT равен on (включено).

3.6. Начальные сведения о POWER Hypervisor

Гипервизор POWER лежит в основе технологии IBM Virtualization Engine, реализованной в линейке продукции на базе процессоров POWER5. В сочетании с возможностями, конструктивно заложенными в процессор POWER5, гипервизор POWER обеспечивает функции активации других системных технологий, включая микроразделы, виртуальные процессоры, виртуальный коммутатор, совместимый с IEEE VLAN, виртуальные SCSI-адAPTERЫ и виртуальные консоли.

Гипервизор POWER является уровнем микрокода, расположенным между размещаемой операционной системой и аппаратным сервером, как показано на рисунке 3-8. Гипервизор POWER устанавливается и активируется всегда, независимо от конфигурации системы.

Гипервизор POWER выполняет следующие задачи:

- ▶ Усиливает целостность разделов с помощью уровня защиты между логическими разделами.
- ▶ Обеспечивает уровень абстрагирования между физическими аппаратными ресурсами и использующими их логическими разделами. Он контролирует размещение виртуальных процессоров на физических процессорах. Он запоминает и восстанавливает всю информацию о состоянии процессоров при коммутации логических процессоров.
- ▶ Управляет механизмом аппаратных прерываний ввода-вывода для логических разделов.

¹ Имеется в виду уменьшение времени отклика. Прим. науч. ред.

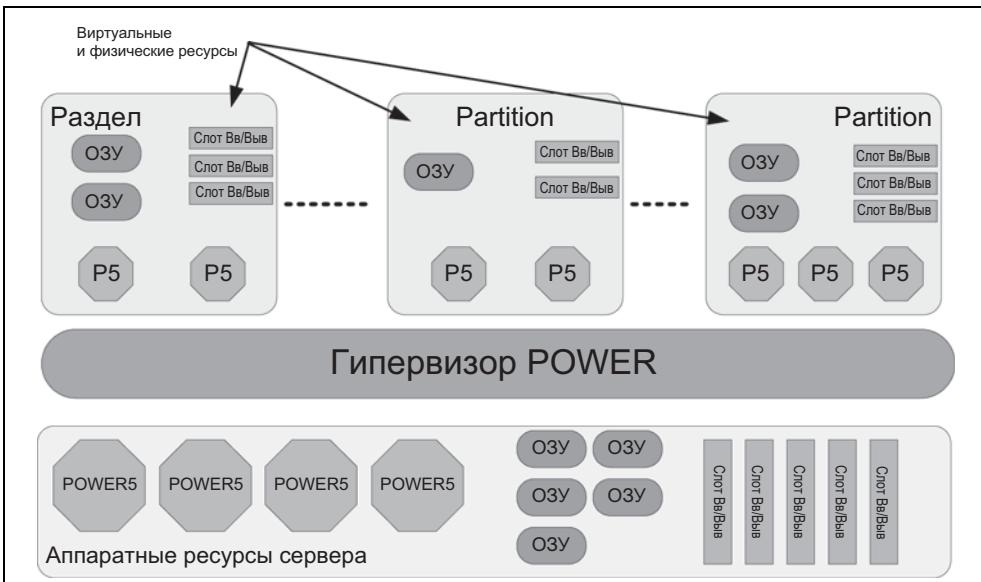


Рис. 3-8. Абстрагирование гипервизором POWER физического серверного оборудования

Микрокод гипервизора и размещаемые операционные системы обмениваются между собой с помощью вызовов гипервизора hcall (hypervisor call).

Гипервизор POWER позволяет нескольким экземплярам операционных систем работать на серверах POWER5 в конкурентном режиме. Поддерживаемые операционные системы перечислены в разделе 1.5 «Поддержка операционных систем».

3.6.1. Диспетчеризация гипервизором POWER виртуальных процессоров

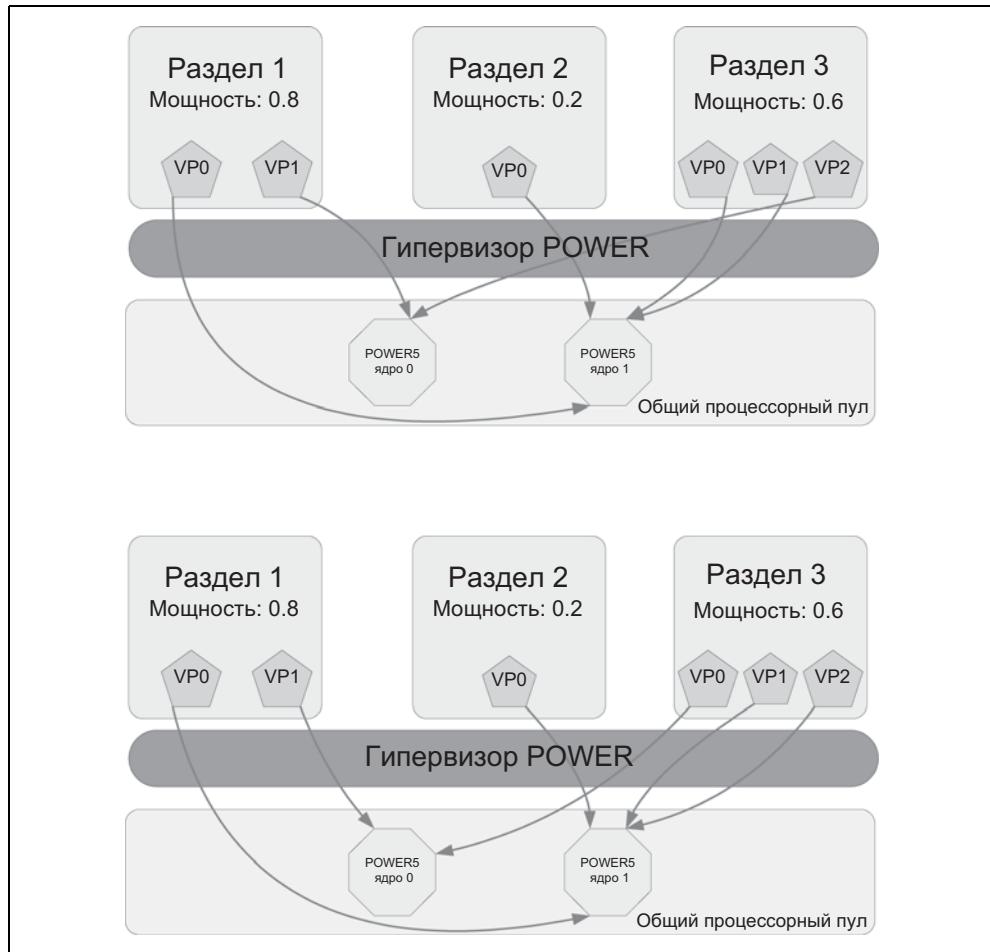
Разделам с общими процессорами для их нагрузок предоставляются один или несколько виртуальных процессоров. Количество виртуальных процессоров в отдельном разделе и во всех разделах не обязательно должно быть связано с количеством физических процессоров в общем процессорном пуле, за исключением того, что каждый физический процессор может поддерживать максимум десять виртуальных процессоров.

Гипервизор POWER управляет распределением имеющихся в общем пуле циклов физических процессоров. Гипервизор POWER использует 10-миллисекундный цикл диспетчеризации; каждому виртуальному процессору гарантируется получение назначенней ему доли процессорных циклов в течение каждого 10-миллисекундного окна диспетчеризации.

Для оптимизации использования физического процессора виртуальный процессор будет уступать ресурсы физического процессора, если у него нет работы и если он переходит в состояние ожидания, например, при ожидании блокировки или завершении ввода-вывода. Виртуальный процессор может уступать физический процессор с помощью вызова гипервизора hcall. Более подробно о вызовах гипервизора POWER можно узнать в Приложении В «Вызовы confer и cede гипервизора POWER».

Механизм диспетчеризации

Для пояснения работы этого механизма рассмотрим три раздела с двумя, одним и тремя виртуальными процессорами. Эти шесть виртуальных процессоров связываются с двумя ядрами физического процессора POWER5, как показано на рисунке 3-9.



На рисунке 3-10 показаны два цикла диспетчеризации гипервизора POWER для двух разделов при распределении шести виртуальных процессоров на двух физических ЦП.

Разделу 1 определены выделенная мощность 0,8 процессорных единиц и два виртуальных процессора. Этому разделу выделяется эквивалент 80 процентов от одного физического процессора в каждом 10-миллисекундном окне диспетче-

ризации общего процессорного пула. Эта рабочая нагрузка использует 40 % каждого физического процессора в течение каждого интервала диспетчеризации.

Раздел 2 сконфигурирован с одним виртуальным процессором и мощностью 0,2 процессорных единиц, предоставляющей ему до 20 процентов использования физического процессора в течение каждого интервала диспетчеризации. В этом примере показан наихудший случай задержки диспетчеризации для данного виртуального процессора, когда использованы 2 мсек в начале интервала диспетчеризации 1 и последние 2 мсек интервала диспетчеризации 2, и между выделениями процессора проходит 16 мсек.

Примечание. Виртуальный процессор может перераспределяться несколько раз в течение одного интервала диспетчеризации. В первом интервале диспетчеризации нагрузке, выполняющейся на виртуальном процессоре 1 в LPAR 1, ресурсы физического процессора предоставляются с перерывом. Такое происходит, если операционная система уступает циклы и повторно активируется вызовом prod hcall.



Рис. 3-10. Диспетчерилизация процессора с микроразделами

Раздел 3 содержит три виртуальных процессора и имеет мощность 0,6 процессорных единиц. Каждый из трех виртуальных процессоров раздела потребляет 20 процентов мощности физического процессора в каждом интервале диспетчериизации, но для виртуальных процессоров 0 и 2 назначаемый им физический процессор меняется от интервала к интервалу.

Сходство процессоров

В гипервизоре POWER предусмотрено назначение для потоков того же самого физического процессора, в котором они выполнялись в предыдущем цикле диспетчериизации. Такое назначение называется назначением по сходству процессора (processor affinity). Гипервизор POWER будет всегда пытаться размещать виртуальный процессор на том же самом физическом процессоре, на котором он работал в предыдущем интервале, и в зависимости от использования ресурсов

будет расширять границы поиска в сторону другого процессора на чипе POWER5, затем – другого чипа на том же самом многочиповом модуле (MCM) и далее – охватывать чип другого МСМ. Целью поиска по сходству процессора является удержание потока как можно ближе к его данным и оптимизация использования кэшей.

Системный мониторинг и статистика

Совместное использование системных ресурсов, например, с помощью микроразделов и SMT предъявляет более высокие требования к традиционным средствам сбора информации и предоставления отчетов о производительности, имеющимся в AIX 5L. В архитектуре POWER5 вводится новый регистр (в каждом ядре) – регистр использования ресурсов процессора (Processor Utilization Resource Register, PURR). Этот регистр обеспечивает раздел точным подсчетом циклов для измерения активности на предоставленных в физическом процессоре отрезках времени.

PURR и средства сбора информации и предоставления отчетов о производительности обсуждаются в разделе 6.5 «Мониторинг виртуализованной среды».

Вызовы `hcall` мониторинга гипервизора

AIX 5L версии 5.3 обеспечивает команды `lparstat` и `mpstat` для отображения статистики гипервизора и сходства виртуальных процессоров. Эти команды подробно обсуждаются в разделе 6.5, «Мониторинг виртуализированной среды».

3.6.2. Гипервизор POWER и виртуальный ввод-вывод

У гипервизора POWER нет своих физических устройств ввода-вывода, и он не обеспечивает для них виртуальные интерфейсы. Все физические устройства ввода-вывода в системе принадлежат логическим разделам.

Примечание. Общие устройства ввода-вывода принадлежат серверу Virtual I/O Server, который обеспечивает доступ к реальным аппаратным ресурсам, на которых базируется виртуальное устройство.

Для поддержки виртуального ввода-вывода гипервизор POWER обеспечивает:

- ▶ Управление и конфигурирование структур для виртуальных адаптеров
- ▶ Контролируемый и безопасный доступ разделов к физическим адаптерам ввода-вывода
- ▶ Виртуализация прерываний и управление

Виды поддерживаемого ввода-вывода

Гипервизором POWER поддерживаются три типа виртуальных адаптеров ввода-вывода:

- ▶ SCSI
- ▶ Ethernet
- ▶ System Port (виртуальная консоль)

Примечание. Сервер Virtual I/O Server поддерживает оптические устройства. Они представляются клиентским разделам как виртуальные SCSI-устройства.

Виртуальные адаптеры ввода-вывода определяются системными администраторами при определении логических разделов. Конфигурационная информация для виртуальных адаптеров предоставляется операционной системе раздела посредством системного микрокаода.

Виртуальный SCSI рассматривается детально в разделе 3.9 «Ознакомление с виртуальным SCSI»; виртуальный Ethernet и общий Ethernet-адаптер обсуждаются в разделе 3.8 «Ознакомление с виртуальным и общим Ethernet».

3.6.3. Системный порт (поддержка виртуального ТTY/консоли)

В каждом разделе необходимо иметь доступ к системной консоли. Такие задачи, как установка операционной системы настройка сети и работа по анализу некоторых проблем, требуют выделенной системной консоли. Гипервизор POWER обеспечивает виртуальную консоль с помощью виртуального ТTY или последовательного адаптера и набора вызовов гипервизора для работы на них.

В зависимости от конфигурации системы, консоль операционной системы может обеспечиваться виртуальным ТTY консоли Hardware Management Console (HMC) или эмулятором терминала, подключенным к физическим системным портам сервисного процессора системы.

3.7. Лицензирование ПО в виртуализованной среде

Для расширения набора предложений с предоставлением ресурсов «по требованию», которые согласовывались бы с развертыванием и принятием инструментов виртуализации систем IBM System p5 и IBM *@server* p5, IBM и ряд независимых изготовителей ПО (ISV) рассмотрели новые методы лицензирования для лучшего удовлетворения потребностей клиентов, связанных с консолидацией бизнес-приложений и промежуточным ПО в виртуализованных средах.

3.7.1. Лицензирование IBM i5/OS

Клиенты, желающие, чтобы i5/OS работала на системах IBM *@server* p5, могут приобрести лицензии (license entitlements) i5/OS, цена которых определяется по количеству процессоров. Условия программного лицензирования (такие как методология подсчета процессоров, агрегирование и возможность переноса) для i5/OS одинаковы для серверов IBM *@server* p5 и серверов i5.

Существует верхний предел на количество лицензий для процессоров i5/OS, совместно используемых разделами, которые клиент может назначать и запускать в системе IBM *@server* p5. На сервере IBM *@server* p5 модели 570 может действовать не более одной процессорной лицензии i5/OS, а на IBM *@server* p5 моделей 590 и 595 может действовать не более двух процессорных лицензий i5/OS.

В оставшейся части этого раздела рассматриваются главные аспекты программного лицензирования для систем IBM *@server* p5, сконфигурированных

с операционными системами IBM AIX 5L и Linux. За более подробной информацией о лицензировании для разделов с i5/OS в системе IBM **@server** p5 обращайтесь к вашему представителю IBM по продажам.

3.7.2. Методы лицензирования ПО для операционных систем UNIX

Хотя обсуждение программного лицензирования по количеству серверов не предусмотрено содержанием этой книги, следует отметить, что термин сервер определяется ISV для целей лицензирования. В большинстве случаев из-за того, что каждый раздел имеет свою операционную систему и аппаратные ресурсы (раздел либо с выделенными процессорами, либо микрораздел), раздел, в котором установлено ПО, считается сервером. В этом случае ПО оплачивается однократно для каждого раздела, в котором оно установлено и работает, независимо от процессоров в данном разделе.

Многие из новых методов лицензирования предлагаются по количеству процессоров, поэтому для определения необходимых затрат на лицензии для ПО важно установить, сколько имеется процессоров (физических или виртуальных), на которых это ПО будет работать. Оставшаяся часть этого раздела применима только к методам лицензирования по количеству процессоров.

Клиентам следует использовать для определения лицензий по количеству процессоров количество активных процессоров, заявленных компанией IBM в качестве ядер в системе IBM **@server** p5. Например, если IBM конфигурирует p5-570 с восемью установленными процессорами, шесть из которых активны, то p5-570 имеет для лицензирования шесть активных ядер из восьми установленных, независимо от количества чипов или процессорных карт.

3.7.3. Факторы лицензирования в виртуализованной системе

Клиентам, планирующим приобретение ПО для системы IBM **@server** p5, работающей с разделами, следует понять факторы лицензирования, так как оплата лицензий зависит от способа использования операционной системой процессоров в системе.

Активные процессоры и аппаратные границы

В лицензировании по количеству процессоров границей для лицензирования является количество активных процессоров в системе (назначенных и неназначенных), так как только активные процессоры могут быть механизмом исполнения для ПО. Это распространяется на любой тип ПО, лицензируемого по количеству процессоров.

Большинство ISV считает разделы с выделенными процессорами в системе IBM **@server** p5 независимыми серверами. В таком случае лицензии на ПО должны приобретаться для всех процессоров в разделе и для всех разделов, в которых установлено данное ПО. Компания IBM использует эту классификацию разделов с выделенными процессорами для отдельных видов ПО IBM.

Количество процессоров для конкретного раздела может варьироваться с течением времени вследствие динамических LPAR-операций, но общее количество лицензий должно быть равным или превышать общее число процессоров, одновременно используемых этим ПО.

Неназначенные процессоры и разделы с выделенными процессорами

В системе, содержащей разделы только с выделенными процессорами (без общего процессорного пула, так как в системе не активирована APV), активные процессоры в пуле неназначенных процессоров могут увеличивать количество процессоров для ПО в разделе, если они добавляются, даже временно, при назначении их динамическими операциями с LPAR. Клиентам следует иметь в виду, что ISV могут потребовать лицензирования для максимального количества процессоров в каждом разделе, где установлено ПО (для максимального числа процессоров в профиле раздела).

IBM требует оплаты за количество процессоров в разделе с выделенными процессорами, даже за процессоры, временно используемые при динамических операциях с LPAR. Так как эти лицензии приобретаются с единовременной оплатой, то лицензирование ПО компании IBM является инкрементальным. Например, клиент может установить раздел с AIX 5L в системе с тремя процессорами для DB2 из пула с восемью активными процессорами (без других разделов с AIX 5L и DB2), затем нарастить этот раздел еще двумя процессорами, а позднее – освободить один процессор из раздела; в этом случае у клиента будет инкрементальное лицензирование, начинающееся с трех процессоров для AIX 5L и DB2, с добавлением лицензий на два дополнительных процессора как для AIX 5L, так и для DB2, и всего будет пять процессорных лицензий.

Процессоры с наращиванием ресурсов по требованию (CUoD)

Процессоры в пуле CUoD не засчитываются при лицензировании, пока с ними не произойдет следующее:

- ▶ Они становятся временно или постоянно активными как часть общего процессорного пула в системах с Advanced POWER Virtualization.
- ▶ Они становятся временно или постоянно активными и назначаются в разделы в системах без Advanced POWER Virtualization.

Клиенты могут предусматривать лицензии на отдельные виды ПО IBM для временного использования на своих системах. Такие лицензии могут использоваться по количеству процессоров или дней и соответствовать возможному временному использованию процессоров CUoD в существующих или новых разделах с AIX 5L или Linux. Например, временные лицензии на ПО для AIX 5L могут использоваться либо для активных процессоров (неназначенных процессоров в системах без APV), либо для новых разделов (создаваемых из неназначенных процессоров или из общего процессорного пула в системах с APV), либо для постоянно активированных процессоров CUoD, либо для временно активированных процессоров On/Off.

Более подробно о процессорах «по требованию» On/Off можно узнать в разделе 3.3.3 «Нарашивание ресурсов по требованию».

Процессоры в общем процессорном пуле

Все процессоры, которые становятся активными и невыделенными, входят в состав общего процессорного пула; следовательно, количество процессоров в общем процессорном пуле равно количеству активных невыделенных процессоров системы.

Доля мощности, выделяемая для микроразделов

Выделенная доля мощности является реальной вычислительной мощностью, которую предоставляют разделу, даже когда его запускают, или выполняются динамические операции с LPAR при его работе. Доля мощности применяется только для времени выполнения и является гарантированным количеством процессорных единиц, которое может потреблять микрораздел. Есть два способа определения раздела с разделяемым процессором: задать режим с верхним пределом (capped) или без верхнего предела (uncapped).

Для микроразделов с верхним пределом доля процессорной мощности является также максимальной процессорной мощностью, которую раздел может использовать, и ее первое значение задается при запуске. Нужно помнить, что при динамических LPAR-операциях можно добавлять процессорные единицы к заданной доле мощности в зависимости как от системных ресурсов в общем процессорном пуле, так и от максимального количества процессорных единиц, допускаемых для данного раздела.

Для микрораздела без верхнего предела доля мощности, предоставляемая разделу, не ограничивает доступ к процессорной мощности. Микрораздел без верхнего предела может использовать мощность, превышающую заданную долю, если есть свободные ресурсы в общем процессорном пуле. Фактором, ограничивающим микрораздел без верхнего предела, является количество определенных виртуальных процессоров. Микрораздел может использовать столько физических процессоров, сколько их имеется в общем процессорном пуле, так как каждый виртуальный процессор одновременно размещается на одном физическом процессоре.

Виртуальные процессоры для микроразделов

Для операционных систем микроразделов создаются виртуальные процессоры, чтобы включить механизм, распределяющий физические процессоры общего процессорного пула между такими микроразделами. Операционная система в микроразделе обращается с виртуальными процессорами как с отдельными системными объектами, и аппаратная часть размещает виртуальные процессоры на физических процессорах в режиме разделения времени.

Когда раздел работает в режиме без верхнего предела и превышает максимальное количество процессорных единиц, есть высокая вероятность того, что его виртуальные процессоры будут размещаться на физических процессорах одновременно. Поэтому для одного микрораздела без верхнего предела максимальное количество виртуальных процессоров, работающих на физических процессорах общего процессорного пула в один и тот же момент времени, равно меньшему из значений, определяющих число виртуальных процессоров и число физических процессоров в общем процессорном пуле.

Сравнение выделяемой доли мощности с виртуальными процессорами

По определению предоставляемая ПО максимальная вычислительная мощность, которая должна лицензироваться в микроразделе с верхним пределом, всегда равна выделяемой мощности. Ее можно расширить до максимального количества процессорных единиц, определяемых в профиле микрораздела. Выделяемая мощность становится решающим фактором лицензирования ПО для микрораздела с верхним пределом и может измеряться для целей аудита при определении

реального использования системы и вычисления возможных потребностей в дополнительных лицензиях.

Для микроразделов без верхнего предела максимальная вычислительная мощность, предоставляемая ПО, зависит от количества виртуальных процессоров в операционной системе и количества физических процессоров в общем процессорном пуле. Следовательно, количество виртуальных процессоров становится решающим фактором лицензирования ПО для микроразделов без верхнего предела и максимум определяется числом физических процессоров в общем процессорном пуле.

На рисунке 3-11 показаны границы лицензирования ПО по количеству процессоров.

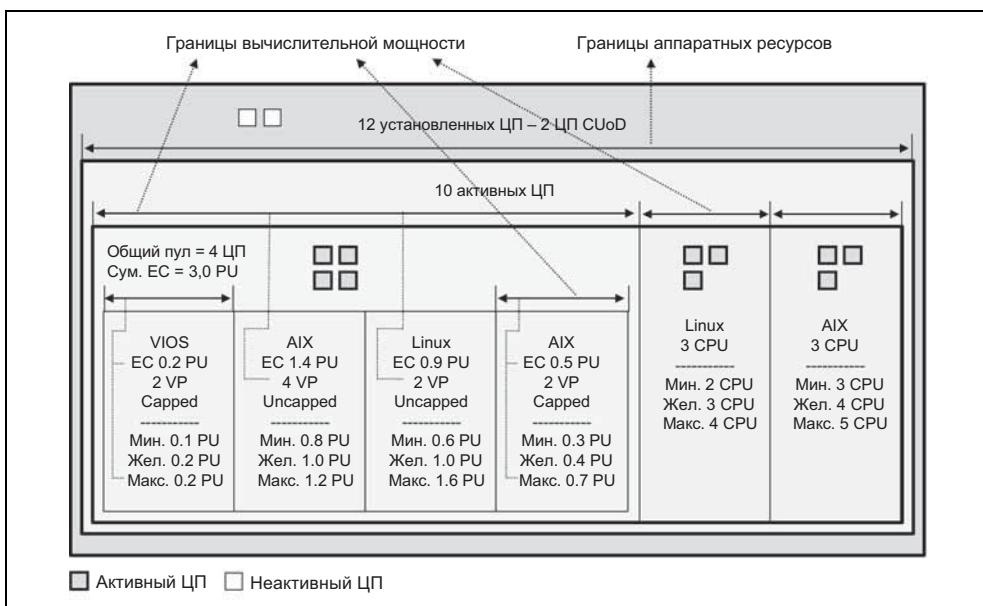


Рис. 3-11. Границы лицензирования ПО по количеству процессоров

3.7.4. Планирование и обеспечение лицензий программного обеспечения IBM

Клиенты, приобретающие ПО IBM, лицензируемое по количеству процессоров, должны учитывать следующие критерии лицензирования (для ПО, поставляемого не IBM, пользователям следует обращаться к представителям по продажам соответствующих ISV):

- ▶ IBM AIX 5L и отдельные программы ПО pSeries лицензируются по количеству процессоров, и лицензии требуются только для тех разделов, в которых установлены операционная система AIX и программы IBM.
- ▶ Для IBM AIX 5L и отдельных программ ПО pSeries может предлагаться лицензирование по количеству процессоров/дней при временном использовании.
- ▶ У дистрибуторов Linux имеются свои методы лицензирования, и лицензии требуются только для тех разделов, в которых установлена эта операционная система.

- ▶ ПО Advanced POWER Virtualization (VIO и PLM) лицензируется по количеству процессоров, и лицензии требуются для всех систем.
- ▶ ПО Advanced POWER Virtualization (VIO и PLM) применяется с лицензированием по количеству процессоров/дней, и лицензии требуются при активации процессоров CUoD On/Off.
- ▶ Отдельные программы ПО IBM, соответствующие IBM Passport Advantage® и лицензируемые по количеству процессоров, могут квалифицироваться по условиям Sub-Capacity, поэтому лицензии требуются только для тех разделов, в которых эти программы установлены. Для соответствия условиям Sub-Capacity клиент должен принять условия IBM International Passport Advantage Agreement Attachment for Sub-Capacity Terms.
- ▶ На момент написания книги отдельные программы ПО IBM *@server* p5 и программы ПО IBM соответствовали временному режиму On/Off. Чтобы соответствовать расценкам на Capacity Upgrade on Demand в режиме On/Off, у клиентов должны быть включены временные ресурсы на соответствующей аппаратуре, и до использования должен быть подписан контракт Amendment for iSeries and pSeries Temporary Capacity Upgrade on Demand Software.
- ▶ Для IBM AIX 5L и отдельных программ ПО IBM программные лицензии могут разделяться между микроразделами с верхним пределом (единственный тип разделов с долями процессора для лицензирования); следовательно, несколько микроразделов, использующих ПО IBM с агрегированной мощностью с позиций процессорных единиц (планируемой мощностью при установке ПО, выделяемыми долями во время выполнения), могут использовать меньше процессорных лицензий, чем при их отдельной оценке.

Примечание. На момент написания книги разделение процессорных лицензий применялось только к AIX 5L, НАСМР и отдельным программам IBM и только к микроразделам с верхним пределом в общем процессорном пуле. Другие связанные с AIX 5L программы IBM могут применяться для разделения лицензий. Обратитесь к вашему представителю по продажам компании IBM за информацией о текущем состоянии отдельных связанных с AIX 5L продуктах IBM, которые могут применяться для разделения лицензий между микроразделами с верхним пределом.

Только отдельное ПО IBM для систем IBM *@server* p5 классифицируется для лицензирования «по требованию». При планировании оплаты лицензий ПО по количеству процессоров для систем IBM *@server* p5 клиенту следует различать следующие варианты:

**Первоначальное
планирование
лицензирования**

Клиент подсчитывает базовые лицензии на доли мощности, основываясь на правилах лицензирования и факторах лицензирования (все, исключая выделенные доли для операционных систем). Клиент покупает процессорные лицензии, основываясь на планируемых потребностях, и максимумом является количество активных процессоров в системе. Клиент также может купить временные лицензии On/Off на отдельное ПО, связанное с pSeries.

Дополнительное лицензирование

Клиент проверяет реальное использование программных лицензий, планирует потребности и определяет лицензии на дополнительные процессорные доли (также лицензии на временные ресурсы On/Off), основываясь на правилах лицензирования и факторах лицензирования (включая доли мощности для операционных систем).

Лицензирование «по требованию»

Клиент обращается в компанию IBM или к ее бизнес-партнеру с заявлением на регистрацию в программе Passport Advantage Program. Клиент выполняет процедуры метода лицензирования (лицензирование sub-capacity для отдельных программ, соответствующих IBM Passport Advantage) и устанавливает систему мониторинга с помощью IBM Tivoli® License Manager для ПО IBM. Компания IBM уведомляется об использовании программных лицензий, а клиент уведомляется компанией IBM о необходимости регулировки лицензируемых долей мощности, если к нему это применимо.

Для первоначального плана лицензирования клиент может использовать подход, приведенный ниже и показанный в сводке на рисунке 3-12:

1. Определяется количество процессоров, которым необходимы программные лицензии, для разделов с выделенными процессорами (планируемое значение между желаемым и максимальным).
2. Определяется количество процессорных единиц, которым необходимы программные лицензии для микроразделов (MP) с верхним пределом (планируемое значение между желаемым и максимальным). Процессорные единицы округляются до следующего целого значения. Проверяется, позволяет ли программа разделение процессорной лицензии.
3. Определяется количество виртуальных процессоров, которым необходимы программные лицензии, для микро-разделов без верхнего предела (планируемое значение между желаемым и максимальным количеством виртуальных процессоров и процессоров общего пула).

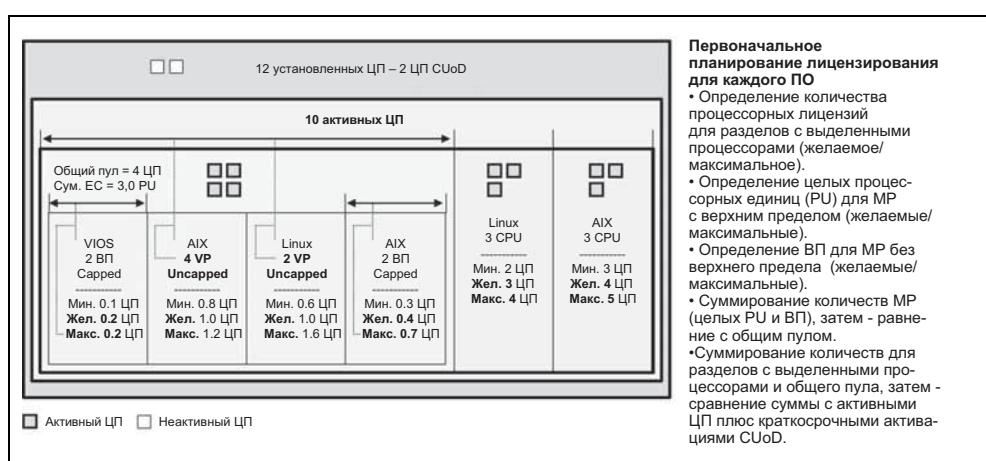


Рис. 3-12. Пример первоначального планирования лицензирования

4. Суммируются отдельные количества процессоров (целые процессорные единицы для микроразделов с верхним пределом, количество виртуальных процессоров для микроразделов без верхнего предела), и берется меньшее из значений суммы и количества процессоров общего пула.
5. Суммируются планируемые лицензии для разделов с выделенными процессорами и планируемые лицензии для общего пула, и берется меньшее значение из суммы и количества планируемых активных процессоров (активных при установке плюс активируемых из ресурсов CUoD).

3.7.5. Лицензирование Sub-capacity для программного обеспечения IBM

Лицензирование Sub-capacity позволяет получать лицензию на ПО для использования на процессорной мощности, меньшей, чем полный процессор, в системах IBM **@server p5**, когда эта программа используется в двух и более разделах.

Лицензирование Sub-capacity дает клиенту возможность получить выгоду от аппаратного разделения, включающего такие передовые функции виртуализации IBM, как общие процессорные пулы, микроразделы (позволяющие разделять процессорные лицензии) и динамическое перемещение ресурсов, с помощью гибкой лицензионной поддержки программного обеспечения.

Ниже приведены важные факторы, которые нужно учитывать при использовании лицензий Sub-capacity на программное обеспечение IBM:

- ▶ Лицензирование Sub-capacity применимо к отдельным программам IBM Passport Advantage, лицензируемым по количеству процессоров.
- ▶ Клиент соглашается с условиями присоединения к International Passport Advantage Agreement и подает заявление по форме Passport Advantage Enrollment Form в компанию IBM или ее бизнес-партнеру.
- ▶ Клиент должен использовать IBM Tivoli License Manager for IBM Software для мониторинга использования программы и ежеквартально представлять в IBM отчет по использованию по форме IBM.
- ▶ На момент написания книги к ПО IBM **@server p5** не предъявлялось требований мониторинга с помощью ITLM.
- ▶ IBM Tivoli License Manager (ITLM) позволяет осуществлять мониторинг использования процессорных лицензий во всех разделах, где установлена программа отслеживаемого ПО IBM (в разделах с выделенными процессорами и микроразделах с верхним пределом и без него).

В разделах системы IBM **@server p5**, в которых он установлен, менеджер ITLM отслеживает общее использование процессорных лицензий, обнаруживает изменения в конфигурации системы при динамических LPAR-операциях и CUoD, а также при активации процессоров On/Off в CUoD и периодически уведомляет IBM о таких изменениях и различиях в лицензировании. Так как лицензии на программы ПО IBM являются инкрементальными, то клиенту нужно приобретать дополнительные программные лицензии IBM, когда использование программ превышает общую выделенную лицензированную мощность (рисунок 3-13).

Для микроразделов с верхним пределом ITLM становится инструментом постоянного измерения выделенной операционной системе доли мощности и позволяет вам сопоставлять первоначально предоставляемые по лицензии ресурсы с действительным потреблением в общем процессорном пуле.

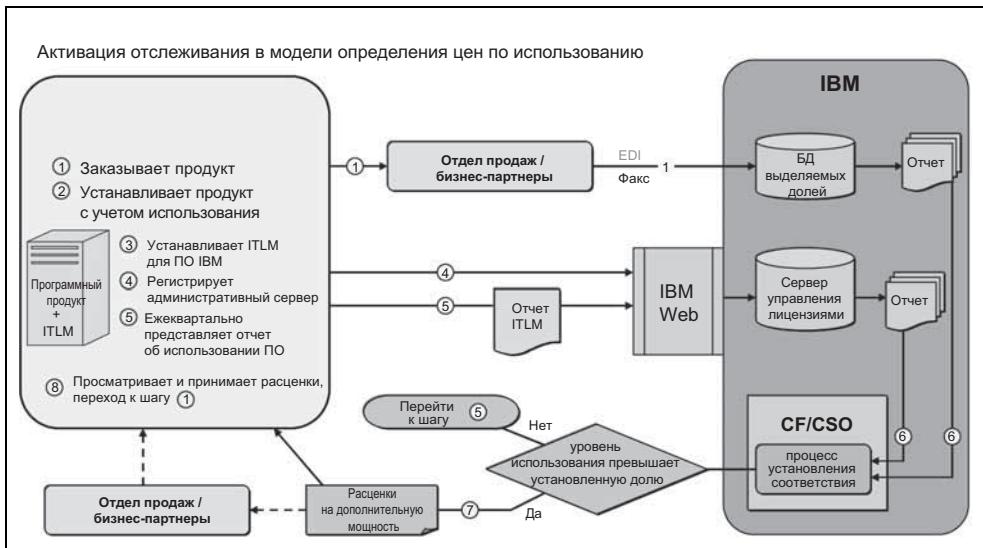


Рис. 3-13. Роль IBM Tivoli License Manager в установлении соответствия мощности

Дополнительную информацию об условиях лицензирования Sub-capacity отдельных программ IBM для вашего региона можно узнать у вашего представителя IBM или на сайте:

<http://www.ibm.com/software/passportadvantage>

Более подробную информацию об использовании IBM Tivoli License Manager для отслеживания ПО IBM можно получить в книге *Introducing IBM Tivoli License Manager*, SG24-6888.

В таблице 3-4 приведена сводка основных характеристик лицензирования отдельных программ IBM для систем с IBM @server p5.

Таблица 3-4. Характеристики лицензирования отдельного ПО IBM

ПО IBM	Вся система (активные ЦП)	CUoD On/Off (временно)	Лицензии на общие процессоры	Контракт лицензирования Sub-capacity
VIO V1.1 и V1.2	X	X	X	-
PLM V1	X	X	X	-
AIX 5L V5.2 и V5.3	-	X	X	-
Performance AIDE V3	-	X	-	-
GPFS V21	-	X	-	-
HACMP V5.2 и V5.3	-	X	X	-
LoadLeveler® V3	-	X	-	-
Parallel Environment for AIX V4	-	X	-	-
Parallel ESSL for Linux pSeries V31	-	X	-	-
Parallel ESSL for AIX V3	-	X	-	-

Продолжение табл.

ПО IBM	Вся система (активные ЦП)	CUoD On/Off (временно)	Лицензии на об- щие процессоры	Контракт лицензи- рования Sub-capacity
CSM V1	-	X	-	-
ESSL V4	-	X	-	-
Lotus® Domino® V6.5	-	X	-	-
TXSeries® V5.0	-	-	X	X
WebSphere MQ V5.3 и V6.0	-	X	X	X
WebSphere MQSeries® Workflow V3.5 и V3.6	-	X	X	X
WebSphere Application Server V5.0, V5.1, V6.0	-	X	X	X
WebSphere Data Interchange V3.2	-	-	X	X
WebSphere Everyplace® Connection Manager without WAP V5.1	-	-	X	X
WebSphere Business Integration Event Broker V5.0	-	-	X	X
WebSphere Business Integration Message Broker V5.0	-	-	X	X
WebSphere Business Integration Message Broker with Rules and For- matter Extension V5.0	-	-	X	X
WebSphere Business Integration Server Foundation, V5.1	-	X	-	-
WebSphere Portal Enable for Multi- platforms V5.0, V5.1	-	X	X	X
WebSphere Portal Extend for Multi- platforms V5.0, V5.1	-	X	X	X
WebSphere InterChange Server V4.3	-	X	X	X
DB2 Data Links Manager V8.2	-	X	X	X
DB2 Net Search Extender V8.2	-	X	X	X
DB2 UDB Data Warehouse Enter- prise Edition V8.2	-	X	X	X
DB2 UDB Enterprise Server Edition V8.2	-	X	X	X

3.7.6. Лицензирование программного обеспечения IBM

Следующие сценарии помогут вам понять, как лицензируется ПО для системы IBM *@server p5* в соответствии с вашими потребностями. Следует отметить, что получение лицензий на ПО является обязанностью клиента.

В таблице 3-5 показано планирование лицензирования для 16-процессорной системы с 14 активными процессорами. В данном случае для разделов DLPAR1, DLPAR2, DLPAR5 и DLPAR6 имеется непосредственная оценка планируемых процессорных лицензий. Разделы DLPAR3 и DLPAR4 являются микроразделами с верхним пределом, и их лицензирование зависит от использования процессорных единиц во время выполнения; следовательно, клиент планирует лицензии с перекрытием реальных будущих потребностей, чтобы воспользоваться преимуществами общих процессорных лицензий для отдельных программ IBM (AIX 5L, НАСМР и ПО IBM по контракту лицензирования Sub-capacity).

Для осуществления такого плана лицензирования клиент соглашается с IBM International Passport Advantage Agreement Attachment for Sub-Capacity Terms, подписывается на программу лицензирования Sub-capacity и устанавливает ITLM для отслеживания использования лицензий на программное обеспечение.

В этом примере есть высокая вероятность превышения программных лицензий для микрораздела с верхним пределом DLPAR3, так как по оценке предусмотрена одна процессорная лицензия и установлен максимум в две процессорные лицензии на основании максимального количества процессорных единиц в профиле раздела.

Таблица 3-5. Оценка лицензирования для первоначального приобретения процессорных лицензий

	DLPAR1	DLPAR2	DLPAR3	DLPAR4	DLPAR5	DLPAR6	CUoD (не активировано)
Операционная система	AIX 5L V5.3	Linux	AIX 5L V5.3	AIX 5L V5.3	Linux	AIX 5L V5.3	Не определено
Дополнительное ПО IBM @server p5	НАСМР		НАСМР	НАСМР			
Дополнительное ПО IBM	DB2	DB2	DB2 / Web-Sphere Application Server	Domino	Web-Sphere Application Server		
Тип раздела	Выделенный	Выделенный	Микро	Микро	Микро	Микро	
Физические процессоры	4	3	7				2
Максимально используется виртуальных процессоров	Не определено	Не определено	2	3	5	8	
Capped / uncapped	Не определено	Не определено	Capped	Capped	Uncapped	Uncapped	

Продолжение табл.

	DLPAR1	DLPAR2	DLPAR3	DLPAR4	DLPAR5	DLPAR6	CUoD (не активировано)
Максимум процессорных единиц	Не определено	Не определено	2,0	3,0	5,0 (VP)	7,0 (Пул)	
Оценка выделяемой мощности	Не определено	Не определено	0,8	1,4	2,4	2,4	
Оценка процессоров для лицензий	4	3	1-2	2-3	5	7	
Планируемые клиентом процессорные лицензии	4	3	1	2	5	7	
Процессорные лицензии AIX = 11+1	4		Округление (0,8+1,4) = 3			7	1 On/Off
Процессорные лицензии НАСМР = 7+1	4		Округление (0,8+1,4) = 3				1 On/Off
Процессорные лицензии DB2 = 8	4	3	1				
WebSphere Application Server, процессорные лицензии = 6 +1			1		5		1 On/Off
Domino, процессорные лицензии = 14				14 (система)			

В таблице 3-6 показана та же конфигурация pSeries через шесть месяцев. К этому моменту произошло несколько событий:

- ▶ В системе были динамически отрегулированы доли мощности, выделяемые микроразделам.
- ▶ Клиент изменил системные профили нескольких разделов.
- ▶ Динамические операции с LPAR перераспределили процессорные единицы между разделами, и программные лицензии типа On/Off не были исчерпаны.
- ▶ Клиент решил постоянно активировать один процессор для повышения общей производительности микроразделов. У одного неактивного процессора все еще остались лицензии On/Off.
- ▶ IBM Tivoli License Manager отслеживает и представляет отчеты по изменениям лицензирования .
- ▶ Клиент установил WebSphere Application Server в разделе DLPAR4.

Таблица 3-6. Пример лицензирования для установленной системы

	DLPAR1	DLPAR2	DLPAR3	DLPAR4	DLPAR5	DLPAR6	CUoD (не активировано)
Операционная система	AIX 5L V5.3	Linux	AIX 5L V5.3	AIX 5L V5.3	Linux	AIX 5L V5.3	Не определено
Дополнительное ПО IBM @server p5	HACMP		HACMP	HACMP			
Дополнительное ПО IBM	DB2	DB2	DB2 / Web-Sphere Application Server	Domino/ Web-Sphere Application Server	Web-Sphere Application Server		
Тип раздела	Выделенный	Выделенный	Микро	Микро	Микро	Микро	
Физические процессоры	4	3	8				1
Реально используется физических процессоров	4+1 On/Off	3	8+1 On/Off				
Максимально используется виртуальных процессоров	Не определено	Не определено	2	3	5	10	
Capped / uncapped	Не определено	Не определено	Capped	Capped	Uncapped	Uncapped	
Максимум процессорных единиц	Не определено	Не определено	2,0	3,0	5,0 (VP)	9,0 (Пул)	
Максимальная реальная доля мощности	Не определено	Не определено	1,7	2,2	4,2	3,4	
Используется процессоров для лицензий без общих процессоров	5	3	2	3	5	9	
Необходимое количество лицензий AIX с общими процессорами = 14			Округление $(1,7 + 2,2) = 4$			9	

Продолжение табл.

	DLPAR1	DLPAR2	DLPAR3	DLPAR4	DLPAR5	DLPAR6	CUoD (не активировано)
Клиентские процессорные лицензии для AIX = 13+1	4		4			8	1 On/Off
Необходимое количество процессорных лицензий НАСМР = 9	5		Округление (1,7 + 2,2) = 4				
Клиентские процессорные лицензии для НАСМР = 8+1	4		4				1 On/Off
Представленные в отчете ITLM лицензии с общим процессором для DB2 = 10, клиентские лицензии DB2 = 10	5	3	2				
Представленные в отчете ITLM лицензии с общим процессором для WAS = 9, клиентские лицензии WAS = 9+1			Округление (1,7 + 2,2) = 4		5		1 On/Off
Клиентские процессорные лицензии Domino = 15				15 (система)			

3.7.7. Лицензирование операционной системы Linux

Условия поставки операционной системы Linux предоставляются дистрибуторами Linux, но все базовые операционные системы Linux лицензируются согласно GPL. Цены дистрибуторов для Linux включают стоимость носителей, упаковки/доставки и документации, и они могут предлагать дополнительные программы с другими лицензиями, а также пакеты сервиса и поддержки.

IBM предлагает возможность заказа и оплаты дистрибутивов Novell SUSE LINUX и Red Hat, Inc. Linux для систем IBM **@server** p5. Это предложение включает поставку носителей программ при первоначальном заказе системы IBM **@server** p5. Клиенты или авторизованные бизнес-партнеры отвечают за установку ОС Linux, если заказ последовал за лицензионным соглашением между клиентом и дистрибутором Linux.

Клиенты должны учитывать количество виртуальных процессоров в микроразделах для целей расширяемости и лицензирования (разделов без верхнего предела) при установке Linux в виртуализированной системе IBM **@server** p5.

Каждый дистрибутор Linux устанавливает свой метод определения цены для своего дистрибутива, сервиса и поддержки. Проконсультируйтесь об этом на веб-сайте дистрибутора или по адресам:

<http://www.novell.com/products/linuxenterpris@server8/pricing.html>

<https://www.redhat.com/software/rhel/compare/server/>

3.8. Ознакомление с виртуальным и разделяемым Ethernet

Виртуальный Ethernet (Virtual Ethernet) обеспечивает возможность коммуникаций между разделами без необходимости назначения физических сетевых адаптеров для каждого раздела. Виртуальный Ethernet позволяет администратору определять соединения между разделами «в памяти», которые управляются на системном уровне (при взаимодействии гипервизора POWER и операционных систем). Характеристики таких соединений подобны характеристикам физических широкополосных соединений Ethernet, и ими поддерживаются протоколы отраслевых стандартов (такие как IPv4, IPv6, ICMP или ARP). Общий Ethernet (Shared Ethernet) обеспечивает нескольким разделам совместное использование физических адаптеров для доступа к внешним сетям.

Для виртуального Ethernet необходимы IBM System p5 или IBM **@server** pSeries либо с AIX 5L Version 5.3, либо с Linux соответствующего уровня и Hardware Management Console (HMC) или Integrated Virtualization Manager (IVM), чтобы определить устройства виртуального Ethernet. Для виртуального Ethernet не требуется приобретения дополнительных функций или оборудования, таких как Advanced POWER Virtualization Feature, которая необходима для адаптеров общего Ethernet (SEA) и виртуальных серверов ввода-вывода (VIOS).

Ознакомление с принципами работы виртуального и общего Ethernet в System p5 разбито на следующие разделы:

- ▶ Даётся общий обзор понятий виртуальной сети и ее использования с AIX 5L.
- ▶ Происходит ознакомление с виртуальным Ethernet в System p5.
- ▶ Объясняется совместное использование физических адаптеров Ethernet в System p5, позволяющее нескольким разделам осуществлять доступ к внешним сетям.

На этом ознакомление с основными принципами работы виртуального и общего Ethernet завершается, и они иллюстрируются примером.

3.8.1. Виртуальная сеть

В этом разделе обсуждаются общие принципы технологии виртуальной сети (Virtual LAN, VLAN). После перечисления преимуществ VLAN приводится пример конкретной реализации в AIX 5L.

Обзор виртуальной сети

Виртуальная сеть (VLAN) – это технология, используемая для создания виртуальных сетевых сегментов (network segments), также называемых сетевыми разделами (network partitions), поверх физических коммутирующих устройств. Виртуальная сеть является понятием Уровня 2 (L2), поэтому она работает под протоколом TCP/IP. При соответствующей конфигурации один коммутатор может поддерживать несколько VLAN, и определение VLAN также может охватывать несколько коммутаторов. Сети VLAN на одном коммутаторе могут быть разъединены или перекрываться, в зависимости от назначенных им портов коммутатора.

Типичная VLAN является одним широковещательным доменом, обеспечивающим взаимную связь всех узлов VLAN без какой-либо маршрутизации (L3-ретрансляции) и без установления мостов между сетями VLAN (L2-ретрансляции). Для TCP/IP это означает, что все интерфейсы узлов в одной VLAN, как правило, совместно используют одну IP-маску подсети/сети и могут преобразовывать все IP-адреса этой VLAN в MAC-адреса с помощью протокола преобразования адресов ARP (Address Resolution Protocol). Даже если VLAN охватывает несколько коммутаторов, с точки зрения TCP/IP ко всем узлам одной VLAN можно осуществить переход за один шаг. Но обмен с узлами в других VLAN осуществляется совершенно иначе: их IP-адреса не могут (и это не нужно) быть преобразованы с помощью ARP, так как эти узлы достигаются через дополнительный шаг в маршрутизаторе (router) уровня L3 (который UNIX-администраторы иногда называют шлюзом – gateway).

На рисунке 3-14 две VLAN (VLAN 1 и 2) определены на трех коммутаторах (коммутаторы А, В и С). К этим трем коммутаторам подключены семь хост-узлов (A-1, A-2, B-1, B-2, B-3, C-1 и C-2). Топология физической сети представляет собой дерево, что типично для сети без резервирования:

- ▶ Коммутатор А
 - Узел А-1
 - Узел А-2
- ▶ Коммутатор В
 - Узел В-1
 - Узел В-2
 - Узел В-3
- ▶ Коммутатор С
 - Узел С-1
 - Узел С-2

Во многих случаях топология физической сети должна учитывать такие физические ограничения среды, как помещения, стены, этажи, здания и комплексы зданий и многое другое. Но сети VLAN могут не зависеть от физической топологии:

- ▶ VLAN 1
 - Узел А-1
 - Узел В-1
 - Узел В-2
 - Узел С-1

- ▶ VLAN 2
 - Узел А-2
 - Узел В-3
 - Узел С-2

Хотя узлы С-1 и С-2 физически подключены к одному коммутатору С, трафик между двумя узлами может быть блокирован. Для обеспечения обмена между VLAN 1 и 2 должна быть установлена маршрутизация уровня L3 или созданы мосты между сетями VLAN; это обычно может осуществляться с помощью устройства уровня L3, например маршрутизатора или брандмауэра, встроенного в коммутатор А.

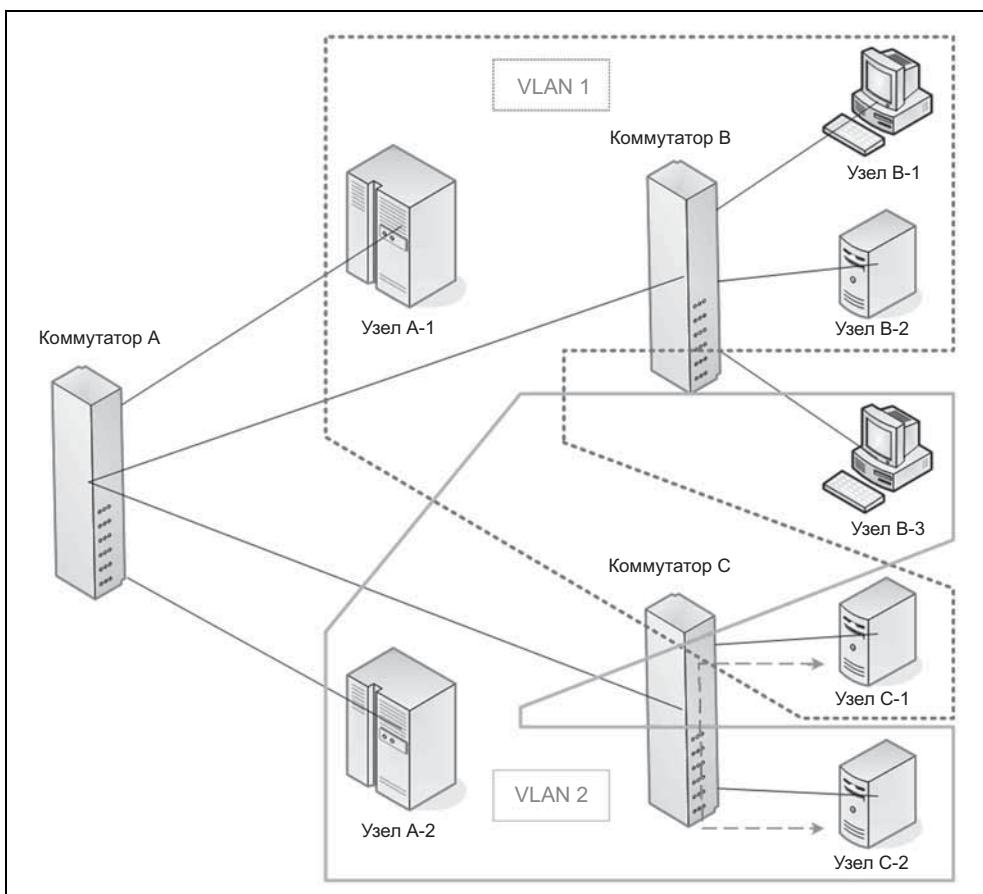


Рис. 3-14. Пример VLAN

Рассмотрим соединения между коммутаторами: по ним осуществляется трафик для обеих сетей – VLAN 1 и VLAN 2. Следовательно, должна быть только одно физическое соединение от В к А, а не по одному для каждой VLAN. Коммутаторы не будут перепутаны и не будут смешивать трафик разных VLAN, так как пакеты, идущие через объединенные (транковые – trunk) порты по соединению, будут соответственно маркироваться.

Преимущества виртуальной сети

Применение технологии VLAN обеспечивает более гибкое развертывание сети по сравнению с традиционной сетевой технологией. Оно помогает преодолеть физические ограничения среды и уменьшить количество необходимых коммутаторов, портов, адаптеров, кабельной проводки и подключений. Такое упрощение физического развертывания сети не дается без затрат: конфигурация коммутаторов и хостов при использовании VLAN становится более сложной. Но общий уровень сложности не повышается; она лишь смещается от физической к виртуальной стороне.

Сети VLAN также дают возможность улучшить производительность сети. При разделении сети на различные VLAN вы также разделяете широковещательные домены. Поэтому когда узел выполняет широковещательную рассылку, то при ее получении прерываются только узлы одной VLAN. Причина в том, что обычно широковещательная рассылка не ретранслируется маршрутизаторами. Вам нужно это помнить, если вы реализуете сети VLAN и желаете использовать протоколы, ориентированные на широковещательную рассылку, например BOOTP или DHCP для IP-автоконфигурации.

Распространенной практикой является также использование сетей VLAN, если в среде реализуются Jumbo Frames технологии Gigabit Ethernet, и не все узлы или коммутаторы способны использовать или быть совместимыми с Jumbo Frames. Jumbo Frames позволяют иметь максимальный размер пакета (MTU), равный 9000 вместо установленного по умолчанию в Ethernet значения 1500. Это может повышать пропускную способность и уменьшать процессорную нагрузку в принимающем узле в сценарии с интенсивной нагрузкой, например, при резервном копировании файлов по сети.

Сети VLAN могут обеспечивать дополнительную защиту, позволяя администратору блокировать пакеты, передаваемые из домена в домен одного коммутатора, тем самым предоставляя дополнительное средство контроля видимости трафика сети для конкретных Ethernet-портов коммутатора. Между сетями VLAN могут устанавливаться фильтры пакетов и брандмауэры, и может реализоваться преобразование сетевых адресов NAT (Network Address Translation). Сети VLAN могут повысить защиту системы от атак.

Поддержка виртуальных сетей в AIX 5L

К некоторым из технологий реализации VLAN относятся:

- ▶ Port-based VLAN (с ориентацией на порты)
- ▶ Layer-2 VLAN (Уровня 2)
- ▶ Policy-based VLAN (с ориентацией на политики)
- ▶ IEEE 802.1Q VLAN

Поддержка VLAN в AIX 5L базируется на реализации IEEE 802.1Q VLAN. AIX 5L может быть также использована с ориентированной на порты VLAN, но она полностью прозрачна к AIX 5L. Поддержка VLAN специально не определена в технологии Advanced POWER Virtualization на IBM System p5, но доступна на всех системах IBM System p5, IBM **@server** pSeries и поддерживаемых системах RS/6000 с соответствующим уровнем AIX 5L.

Поддержка IEEE 802.1Q VLAN осуществляется разрешением драйверу устройства AIX 5L VLAN добавлять метку VLAN ID к каждому Ethernet-кадру, как показано на рисунке 3-15, и ограничением Ethernet-коммутаторами приема кадров только портами с авторизацией для данного VLAN ID.

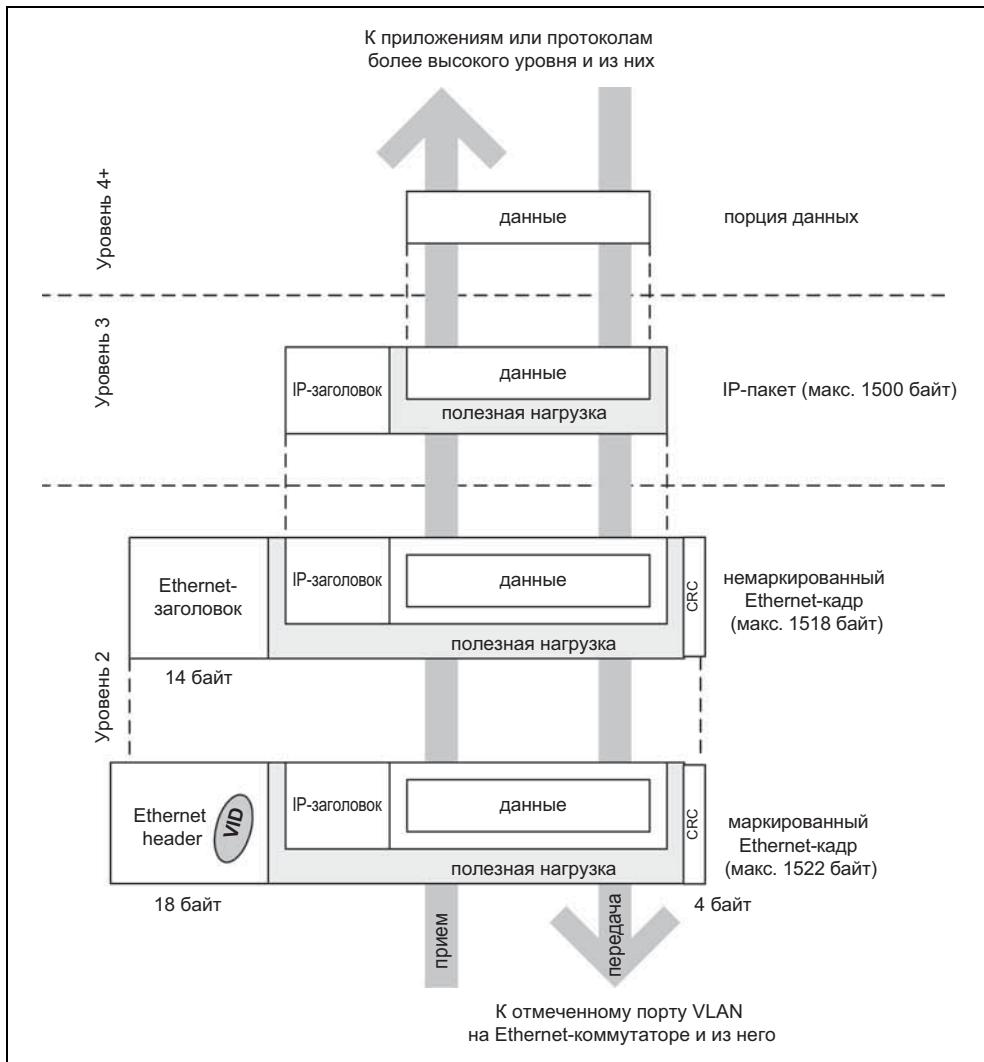


Рис. 3-15. Метка VID добавляется в расширенный Ethernet-заголовок

Метка VLAN ID помещается в Ethernet-заголовок и не образует дополнительного заголовка. Чтобы это можно было сделать, размер Ethernet-кадра для маркируемых кадров был увеличен с 1518 байт до 1522 байт и формат Ethernet-заголовка был слегка изменен при вводе в действие IEEE802.1Q. Следовательно, в противоположность, например, протоколу Point-to-Point-Protocol-over-Ethernet (PPPoE), который обычно используется для xDSL с MTU, равным 1492, вам не нужно бес-

покоиться об уменьшении MTU TCP/IP, равного 1500, в связи с маркировкой VLAN ID.

Примечание. Вам не нужно уменьшать размер MTU TCP/IP, равный по умолчанию 1500, для Ethernet благодаря дополнительным 4 байтам, введенным с помощью IEEE 802.1Q VLAN.

Внимание. Если вы увеличиваете размер MTU TCP/IP для виртуального Ethernet, который реализуется с помощью гипервизора POWER, о чем рассказывается в разделе 3.8.2 «Построение сетей между разделами с помощью виртуального Ethernet», то должны учитывать дополнительные 4 байта, введенные с помощью IEEE 802.1Q VLAN: максимальный MTU равен 65394 без VLAN и 65390 байтам с VLAN. Это вызвано пределом в 65408 байт для кадров виртуальной Ethernet, передаваемых через гипервизор POWER. (Ethernet-заголовки имеют размеры 14 и 18 байт соответственно, но нет необходимости в 4 байтах CRC в гипервизоре POWER.)

Порт на коммутаторе с функцией VLAN имеет по умолчанию идентификатор порта для виртуальной сети PVID (Port virtual LAN ID), указывающий на VLAN по умолчанию, которой принадлежит порт. Коммутатор добавляет метку PVID к немаркированным пакетам, получаемым этим портом. Кроме сети с PVID порт может принадлежать дополнительным сетям VLAN и иметь присвоенные ему метки VLAN ID, указывающие на дополнительные сети VLAN, которым принадлежит порт.

- ▶ Порт коммутатора только с PVID называется *немаркированным (untagged) портом*. Немаркированные порты используются для соединения VLAN-неосведомленных хостов.
- ▶ Порт с PVID и дополнительными VID называется *маркированным (tagged) портом*. Маркированные порты используются для соединения VLAN-осведомленных хостов.

VLAN-осведомленность (VLAN-aware) означает, что хост является совместимым с IEEE 802.1Q и может интерпретировать VLAN-метки, то есть истолковывать, добавлять и удалять их из Ethernet-кадров. VLAN-неосведомленный (VLAN-unaware) хост может прийти в замешательство при получении маркированного Ethernet-кадра. Может произойти потеря кадра с индикацией ошибки кадра.

Получение пакетов маркированным портом

Маркированный порт использует следующие правила при получении пакетов от хоста:

1. Маркированный порт получает немаркированный пакет: Пакет будет про-маркирован PVID, затем ретранслирован.
2. Маркированный порт получает пакет, маркированный PVID или одним из присвоенных VID: Пакет будет ретранслироваться без модификации.
3. Маркированный порт получает пакет, маркированный любым VLAN ID, отличающимся от PVID или от любого присвоенного дополнительного VID: Пакет будет отбрасываться.

Следовательно, маркированный порт будет принимать только немаркированные пакеты и пакеты с метками VLAN ID (PVID или дополнительными VID) тех сетей VLAN, которым этот порт был назначен. Второй случай является наиболее типичным.

Получение пакетов немаркированным портом

Порту коммутатора, сконфигурированному в немаркированном режиме, разрешено иметь только PVID и принимать только немаркированные пакеты или пакеты, маркированные PVID. Функция немаркированного порта позволяет системам, не понимающим VLAN-маркировку (VLAN-неосведомленным хостам), обмениваться с другими системами, использующими стандартный Ethernet.

Немаркированный порт использует следующие правила при приеме пакета от хоста:

1. Немаркированный порт получает немаркированный пакет: Пакет маркируется PVID, затем ретранслируется.
2. Немаркированный порт получает пакет, маркованный PVID: Пакет ретранслируется без модификации.
3. Немаркированный порт получает пакет, маркованный любым VLAN ID, отличающимся от PVID: Пакет отбрасывается.

Первый случай является наиболее типичным; другие два случая не возникают в правильно сконфигурированной системе.

После успешного получения пакета через маркированный или немаркированный порт коммутатору не нужно выполнять дальнейшую обработку немаркированных пакетов, и он обрабатывает только маркированные пакеты. По этой причине несколько VLAN могут совместно использовать одно физическое соединение с соседним коммутатором. Это физическое соединение выполняется через транковые порты соответственно со всеми назначеными VLAN.

Отправка пакетов маркированным или немаркированным портом

Перед отправкой пакета коммутатор должен определить порт назначения пакета на основании MAC-адреса места назначения пакета. Порт назначения должен иметь PVID или VID, совпадающий с VLAN ID пакета. Если пакет является широковещательным (или многоадресным – multicast), то он ретранслируется всем (или нескольким) портам VLAN, даже с использованием соединений с другими коммутаторами. Если не может быть определен действительный порт назначения, то пакет просто отбрасывается. В конце, после внутренней ретрансляции пакета портам назначения коммутатора, перед отправкой пакета принимающему хосту, метка VLAN ID может удаляться или нет, в зависимости от типа порта:

- ▶ Пакет отправляется маркированным портом: Метка PVID или VID остается на пакете.

- ▶ Пакет отправляется немаркированным портом: Метка PVID удаляется с пакета.

Следовательно, маркированные и немаркированные порты коммутатора ведут себя одинаково в отношении принимаемых пакетов, но их поведение различно при отправке пакетов.

АдAPTERы и интерфейсы Ethernet в AIX 5L

В AIX 5L есть различия между сетевым адаптером и сетевым интерфейсом:

- Сетевой адаптер** Представляет собой устройство Уровня 2, например Ethernet-адаптер ent0 имеет MAC-адрес, такой как 06:56:C0:00:20:03.
- Сетевой интерфейс** Представляет собой устройство Уровня 3, например Ethernet-интерфейс en0 имеет IP-адрес, такой как 9.3.5.195.

Типично сетевой интерфейс связан с сетевым адаптером, например Ethernet-интерфейс en0 связан с Ethernet-адаптером ent0. В AIX 5L также есть некоторые сетевые интерфейсы, которые не связаны с сетевым адаптером, например интерфейс обратной связи (loopback) lo0 или интерфейс виртуального IP-адреса VIPA (Virtual IP Address), такой как vi0, если он определен.

Примечание. В Linux нет различий между сетевым адаптером и сетевым интерфейсом в отношении именования устройств: для них обоих определяется только одно имя устройства. В Linux сетевое устройство eth0 представляет собой сетевой адаптер и сетевой интерфейс, и это устройство имеет атрибуты как Уровня 2, так и Уровня 3, такие как MAC-адрес и IP-адрес.

При использовании VLAN, EtherChannel (EC), Link Aggregation, (LA) или Network Interface Backup (NIB) с AIX 5L общая концепция заключается в том, что Ethernet-адаптеры объединяются с другими Ethernet-адаптерами, как показано на рисунке 3-16. EtherChannel и Link Aggregation будут обсуждаться подробно в разделе 5.1.2 «Использование Link Aggregation или EtherChannel для внешних сетей».

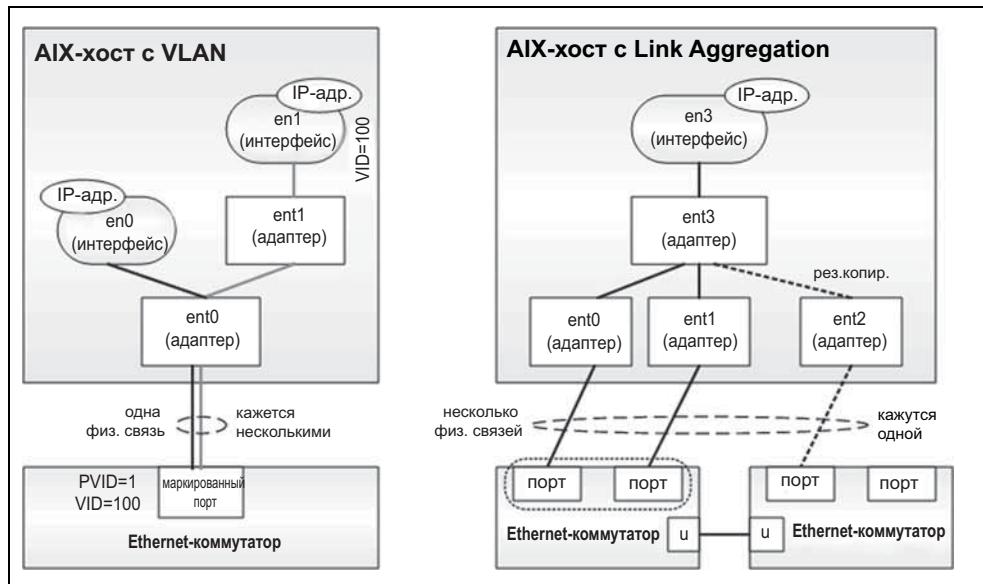


Рис. 3-16. АдAPTERы и интерфейсы с сетями VLAN (слева) и LA (справа)

При конфигурировании сетей VLAN на физическом Ethernet-адаптере в AIX 5L для каждого VLAN ID, конфигурируемого администратором, будет автоматически создаваться другой Ethernet-адаптер, представляющий эту VLAN. Есть небольшое отличие в отношении того, что происходит с первоначальными адаптерами: в случае EC, LA и NIB, адAPTERы-компоненты больше не будут доступны для другого использования, например для связывания с интерфейсом. В противоположность этому, при создании VLAN-адаптера подключаемый Ethernet-адаптер будет оставаться в доступном состоянии и интерфейс по-прежнему может быть связан с ним в дополнение к VLAN-адаптеру.

Если у вас имеется один реальный Ethernet-адаптер с именем устройства ent0, который подключен к маркированному порту коммутатора с PVID=1 и VID=100, то администратор сгенерирует дополнительное имя устройства ent1 для VLAN с VID=100. Первоначальное имя устройства ent0 будет представлять немаркированный порт VLAN с PVID=1. Ethernet-интерфейсы могут размещаться на обоих адаптерах: en0 может быть помещен поверх ent0, а en1 – на ent1, и будут сконфигурированы разные IP-адреса для en0 и en1. Это показано на рисунке 3-16.

3.8.2. Построение сетей между разделами с помощью виртуального Ethernet

Микрокод гипервизора POWER реализует виртуальный Ethernet-коммутатор по типу IEEE 802.1Q VLAN. Подобно физическому Ethernet-коммутатору IEEE 802.1Q он может поддерживать маркированные и немаркированные порты. Так как виртуальному коммутатору в действительности не нужны порты, то виртуальные порты соответствуют непосредственно виртуальным Ethernet-адаптерам, которые могут быть назначены разделам LPAR из HMC или IVM. В данном случае нет необходимости явно подключать адаптер виртуального Ethernet к порту виртуального Ethernet-коммутатора. Но используя аналогию с физическими Ethernet-коммутаторами, порт виртуального Ethernet-коммутатора конфигурируется, когда вы конфигурируете адаптер виртуального Ethernet в HMC или IVM.

Для AIX 5L адаптер виртуального Ethernet ненамного отличается от адаптера реального Ethernet. Он может использоваться:

- ▶ Для конфигурирования на нем Ethernet-интерфейса с IP-адресами
- ▶ Для конфигурирования на нем VLAN-адаптеров (по одному для каждого VID)
- ▶ Как компонент адаптера Network Interface Backup

Но он не может использоваться для EtherChannel или Link Aggregation

Виртуальный Ethernet-коммутатор гипервизора POWER может поддерживать кадры виртуального Ethernet размером до 65408 байт, что значительно больше того, что поддерживают физические коммутаторы: 1522 байта являются стандартом, а 9000 байт поддерживаются с помощью Gigabit Ethernet Jumbo Frames. Таким образом, с помощью виртуального Ethernet гипервизора POWER вы можете увеличить размер MTU TCP/IP до 65394 (= 65408 – 14 для заголовка, без CRC) в случае без VLAN и до 65390 (= 65408 – 14 – 4 для VLAN, снова без CRC), если вы используете VLAN. Увеличение размера MTU полезно для производительности, так как при этом уменьшается объем обработки благодаря заголовкам и уменьшается количество прерываний, на которые должен реагировать драйвер устройства.

3.8.3. Совместное использование физических Ethernet-адаптеров

Существует два подхода к подключению виртуального Ethernet, обеспечивающие обмен между разделами на одном сервере, к внешней сети:

Маршрутизация

Ретрансляция IP-пакетов на Уровне 3

Построение мостов

Ретрансляция Ethernet-кадров на Уровне 2

Маршрутизация

Обеспечивая функции IP-ретрансляции в разделе AIX 5L или Linux с помощью виртуальных или физических Ethernet-адаптеров, раздел может работать как маршрутизатор. На рисунке 3-17 показан пример конфигурации. Клиентские разделы будут иметь свои наборы маршрутов по умолчанию для раздела, по которым трафик направляется во внешнюю сеть.

Ограничение. В конфигурации такого типа раздел, маршрутизирующий трафик во внешнюю сеть, не может быть сервером VIOS, так как вы не можете активировать IP-ретрансляцию через интерфейс командной строки VIOS.

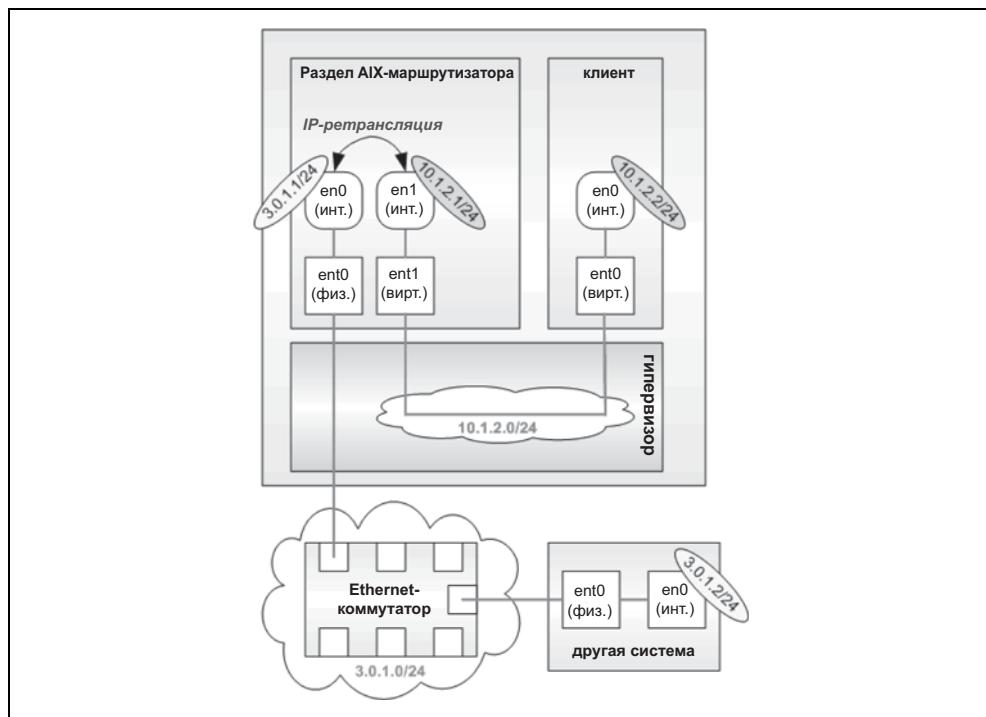


Рис. 3-17. Соединение с внешней сетью с помощью маршрутизации

Подход с маршрутизацией имеет следующие характеристики:

- ▶ Он не требует приобретения функции Advanced POWER Virtualization и использования виртуального сервера ввода-вывода (VIOS).
- ▶ В таких разделах с маршрутизацией можно реализовывать IP-фильтрацию, брандмауэры или качество обслуживания (QoS).
- ▶ Разделы с маршрутизацией могли бы также действовать в качестве конечных точек для IPsec-туннелей, обеспечивая шифрованный обмен по внешним сетям для всех разделов без необходимости конфигурирования IPSec во всех разделах.
- ▶ Может обеспечиваться высокая доступность с помощью реализации нескольких разделов с маршрутизацией и конфигурирования множественных соединений по технологии IP-multipathing в клиентах или с помощью реализации передачи IP-адресов при отказе между разделами с маршрутизацией, обсуждаемой в разделе 5.1.3 «Высокая доступность для обмена с внешними сетями».

Общий Ethernet-адаптер

Общий Ethernet-адаптер (Shared Ethernet Adapter, SEA) может использоваться для подключения физической сети Ethernet к виртуальной сети Ethernet. Он также обеспечивает возможность нескольким клиентским разделам совместно использовать один физический адаптер. С помощью SEA вы можете соединять внутренние и внешние VLAN, используя физический адаптер. SEA, размещенный в виртуальном сервере ввода-вывода VIOS, действует как мост Уровня 2 между внутренней и внешней сетями.

SEA является сетевым мостом Уровня 2, безопасно транспортирующим сетевой трафик между виртуальными сетями Ethernet и реальными сетевыми адаптерами. Служба общего Ethernet-адаптера (Shared Ethernet Adapter, SEA) работает в виртуальном сервере ввода-вывода (Virtual I/O Server). Она не может выполняться в разделе AIX 5L общего назначения.

Подсказка. Раздел с Linux также может обеспечивать мостовую функцию с помощью команды brctl.

Для использования SEA есть несколько ограничений:

- ▶ SEA требует гипервизора POWER Hypervisor, функции Advanced POWER Virtualization и установки Virtual I/O Server.
- ▶ SEA не может использоваться с более ранними версиями ОС, чем AIX 5L Version 5.3, так как драйверы устройств для виртуального Ethernet имеются только для AIX 5L Version 5.3 и Linux. Следовательно, для раздела с AIX 5L Version 5.2 будет нужен физический Ethernet-адаптер.

Адаптер SEA позволяет разделам осуществлять обмен за пределами системы без необходимости выделения физического слота ввода-вывода и физического сетевого адаптера клиентскому разделу. Общий сетевой адаптер SEA имеет следующие характеристики:

- ▶ MAC-адреса виртуального Ethernet виртуальных Ethernet-адаптеров являются видимыми для внешних систем (используя команду arp -a).

- Поддерживается одноадресная, широковещательная и многоадресная рассылка, поэтому протоколы, ориентированные на широковещательную или многоадресную рассылку, такие как Address Resolution Protocol (ARP), Dynamic Host Configuration Protocol (DHCP), Boot Protocol (BOOTP) и Neighbor Discovery Protocol (NDP), могут работать с SEA.

Чтобы создать мост для сетевого трафика между виртуальным Ethernet и внешней сетью, сервер VIOS необходимо конфигурировать хотя бы с одним физическим Ethernet-адаптером. Один адаптер SEA может совместно использоваться несколькими виртуальными Ethernet-адаптерами, и каждый из них может поддерживать несколько VLAN. На рисунке 3-18 показан пример конфигурации SEA с одним физическим и двумя виртуальными Ethernet-адаптерами. Общий Ethernet-адаптер SEA может включать в себя до 16 виртуальных Ethernet-адаптеров, совместно использующих физический доступ.

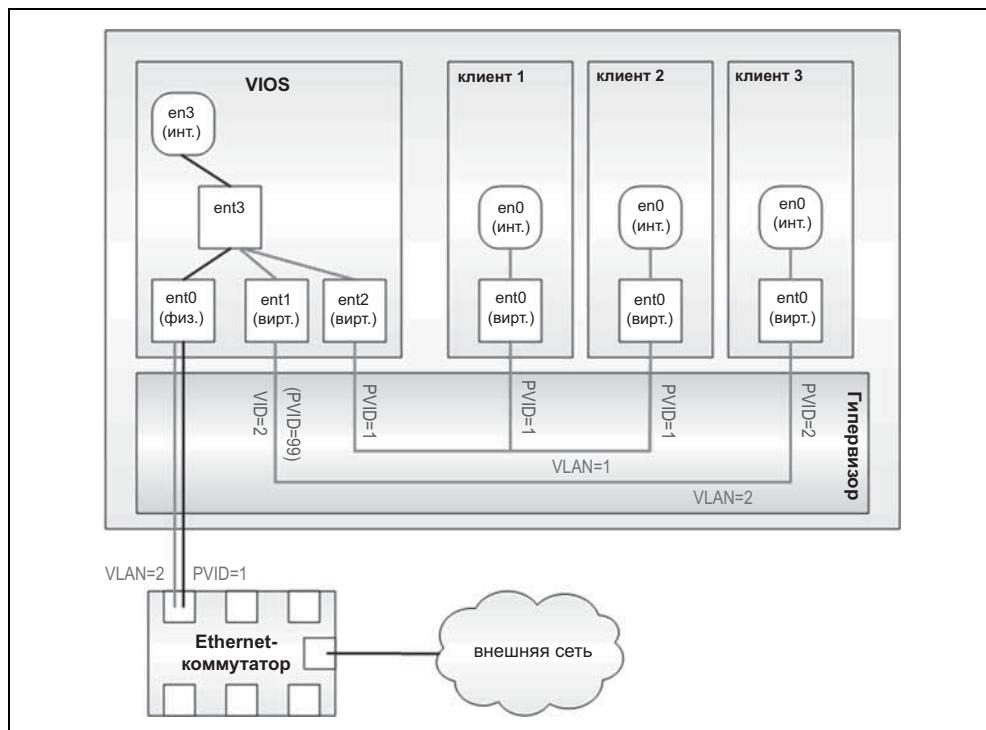


Рис. 3-18. Общий Ethernet-адаптер

Виртуальный Ethernet-адаптер, подключаемый к SEA, должен иметь отмеченным поле флагка **Access External Networks** (называемого в некоторых ранних версиях HMC *транковым флагком – trunk flag*). Когда Ethernet-кадр отправляется виртуальным Ethernet-адаптером в клиентском разделе гипервизора POWER, гипервизор POWER ищет MAC-адрес места назначения внутри VLAN. Если такого MAC-адреса в пределах этой VLAN не существует, то он ретранслирует кадр виртуальному Ethernet-адаптеру во VLAN, у которой включена опция Access External

Networks. Этот виртуальный Ethernet-адаптер соответствует порту моста Уровня 2, а физический Ethernet-адаптер образует другой порт того же моста.

Примечание. Общий виртуальный адаптер не требует IP-конфигурирования для выполнения функции Ethernet-моста. Но IP-конфигурирование в SEA оказывается очень удобным, так как тогда можно войти на сервер VIOS через TCP/IP, например, для выполнения динамических LPAR-операций или осуществления удаленного входа в систему.

SEA направляет пакеты, основываясь на метках VLAN ID. Один из виртуальных адаптеров в SEA-адаптере сервера VIOS может быть определен как PVID-адаптер по умолчанию. Ethernet-кадры без каких-либо меток VLAN ID, получаемые SEA из внешней сети, ретранслируются этому адаптеру, и им присваивается PVID по умолчанию. На рисунке 3-18 адаптером по умолчанию назначен ent2, поэтому все немаркированные кадры, получаемые ent0 из внешней сети, будут ретранслироваться ent2. Так как ent1 не является PVID-адаптером по умолчанию, то в этом адаптере будет использоваться только VID=2, а на PVID=99 адаптера ent1 не будет обращаться внимания. Так можно сделать для любого неиспользуемого VLAN ID. В другом варианте ent1 и ent2 также могут объединяться в один виртуальный адаптер ent1 с PVID=1 и VID=2 с флагком адаптера по умолчанию.

Когда SEA получает или отправляет IP-пакеты (IPv4 или IPv6), которые больше, чем MTU адаптера, через который ретранслируется пакет, то либо выполняется IP-фрагментация, либо отправителю возвращается сообщение ICMP packet too big message (сообщение ICMP о слишком большом пакете), если в IP-заголовке задано do not fragment (не фрагментировать). Это используется, например, в Path MTU discovery.

Теоретически один адаптер может действовать в качестве единого контакта с внешними сетями для всех клиентских разделов. В зависимости от количества клиентских разделов и создаваемой ими сетевой нагрузки может стать критически важной проблема производительности. Так как SEA зависит от виртуального ввода-вывода, то он потребляет процессорное время для всех видов обмена. Для ЦП может создаваться значительная нагрузка при использовании виртуального Ethernet и SEA.

Есть несколько разных способов конфигурирования физических и виртуальных Ethernet-адаптеров в общие Ethernet-адаптеры для максимального увеличения пропускной способности:

- ▶ С помощью Link Aggregation (EtherChannel) можно агрегировать несколько физических сетевых адаптеров. В разделе 5.1.2 «Использование Link Aggregation или EtherChannel для внешних сетей» вы можете найти более подробную информацию.
- ▶ С помощью нескольких SEA можно обеспечить больше очередей и увеличить производительность.

Другими аспектами, которые следует учесть, являются доступность (см. 5.1.3 «Высокая доступность для обмена с внешними сетями») и возможность соединения с различными сетями.

Использовать маршрутизацию или мосты?

В сценарии консолидации, когда множество существующих серверов консолидируются в несколько систем или когда разделы LPAR часто перемещаются с одной системы на другую, предпочтительным вариантом чаще становятся мосты, так как не нужно изменять сетевую топологию, и IP-подсети и IP-адреса консолидированных серверов могут оставаться немодифицированными. Даже в существующей схеме с несколькими VLAN могут применяться мосты.

Маршрутизация может заслуживать внимания, если кроме базовой ретрансляции пакетов на центральном месте стоит выполнение таких дополнительных функций, как IP-фильтрация, брандмауэры, QoS Routing или IPsec-туннелирование. Маршрутизация также является предпочтительным подходом, если внешняя сеть является коммутируемым Ethernet Уровня 3 с использованием протокола динамической маршрутизации OSPF, имеющимся во многих средах IBM System z9. Для некоторых сред одним из факторов может быть также то, что подход с маршрутизацией не требует использования сервера VIOS и приобретения функции Advanced POWER Virtualization.

В заключение нужно добавить, что в большинстве типичных сред создание мостов будет наиболее приемлемым и даже более простым в конфигурировании вариантом, поэтому его следует рассматривать в качестве подхода по умолчанию.

3.8.4. Пример конфигурации виртуального и общего Ethernet

После ознакомления с основными принципами работы сетей VLAN, виртуального Ethernet и общих Ethernet-адаптеров в предыдущих разделах в этом разделе более подробно рассматривается, как работает обмен между разделами и с внешними сетями, с использованием примера конфигурации на рисунке 3-19.

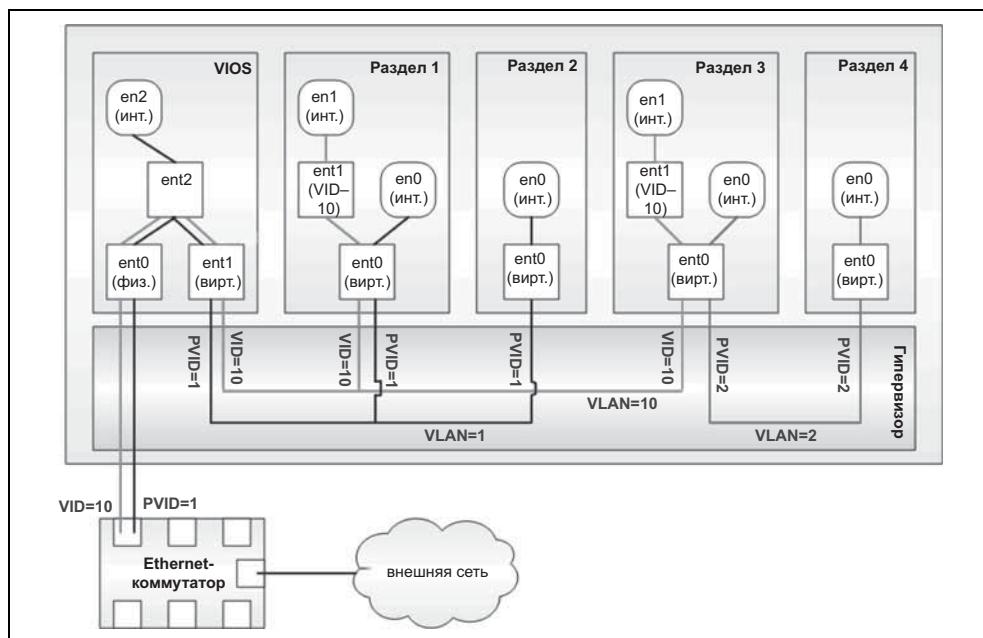


Рис. 3-19. Пример конфигурации VLAN

В этой конфигурации используются четыре клиентских раздела (разделы 1 – 4), в которых работают AIX 5L и один виртуальный сервер ввода-вывода Virtual I/O Server (VIOS). Каждый из клиентских разделов определен с одним виртуальным Ethernet-адаптером. Сервер VIOS имеет общий Ethernet-адаптер (SEA), действующий как мост для трафика во внешнюю сеть.

Построение сетей между разделами

Раздел 2 и раздел 4 используют только метку порта виртуальной сети PVID (Port virtual LAN ID). Это означает следующее:

- ▶ Операционная система, работающая в таком разделе, не осведомлена о сетях VLAN.
- ▶ Принимаются только пакеты для VLAN, определенных как PVID.
- ▶ Из пакетов, маркированных для своих VLAN, метка убирается гипервизором POWER до получения их разделами.
- ▶ К пакетам, отправляемым этими разделами, гипервизором POWER прикрепляется VLAN-метка для VLAN, определенных как PVID.

Кроме PVID, виртуальные Ethernet-адAPTERы в разделе 1 и разделе 3 также сконфигурированы для VLAN 10 с помощью VLAN Ethernet-адаптера (ent1) и сетевого интерфейса (en1) с использованием команды `smitty vlan` на AIX 5L (с использованием команды `vconfig` в Linux). Это означает следующее:

- ▶ Пакеты, передаваемые через сетевые интерфейсы en1, маркируются для VLAN 10 с помощью VLAN Ethernet-адаптера ent1 в AIX 5L.
- ▶ Сетевыми интерфейсами en1 принимаются только пакеты для VLAN 10.
- ▶ Пакеты, передаваемые через en0, не маркируются AIX 5L, но автоматически маркируются гипервизором POWER для VLAN, определенной как PVID.
- ▶ Сетевыми интерфейсами en0 принимаются только пакеты для VLAN, определенной как PVID.

В конфигурации, показанной на рисунке 3-19, в виртуальном сервере ввода-вывода (VIOS) сети VLAN 1 и VLAN 10 через общий Ethernet-адаптер (SEA) соединяются мостом с внешним Ethernet-коммутатором. Но сам VIOS может обмениваться только с VLAN 1 через свой сетевой интерфейс en2, подключенный к SEA. Так как это связано с PVID, то метки VLAN автоматически добавляются и удаляются гипервизором POWER при отправке пакетов в другие внутренние разделы и получении пакетов из них через интерфейс en2.

В таблице 3-7 представлена сводка о том, какие разделы в конфигурации виртуального Ethernet из рисунка 3-19 могут вести взаимный внутренний обмен (также показано, через какие сетевые интерфейсы они могут вести обмен).

Таблица 3-7. Обмен между разделами VLAN

Внутренняя VLAN	Раздел / сетевой интерфейс
1	Раздел 1 / en0 Раздел 2 / en0 VIOS / en2
2	Раздел 3 / en0 Раздел 4 / en0
10	Раздел 1 / en1 Раздел 3 / en1

Если бы VIOS потребовалась возможность также обмениваться и с VLAN 10, то стали бы нужны дополнительный Ethernet-адаптер и сетевой интерфейс с IP-адресом для VLAN 10, как показано слева на рисунке 3-20. VLAN-неосведомленного виртуального Ethernet-адаптера только с PVID, как показано слева на рисунке 3-20, было бы достаточно; здесь нет необходимости во VLAN-осведомленном Ethernet-адаптере (ent4), как показано в центре рисунка 3-20. Более простая конфигурация только с PVID справилась бы с этой задачей, так как VIOS уже имеет доступ к VLAN 1 через сетевой интерфейс (en2), подключенный к SEA (ent2). Как вариант, вы могли бы связать дополнительный Ethernet-адаптер VLAN (ent3) с SEA (ent2), как показано справа на рисунке 3-20.

Примечание. Хотя можно сконфигурировать несколько IP-адресов в VIOS, рекомендуется иметь не более одного адреса, так как это является допущением для некоторых команд интерфейса командной строки. Таким образом, VIOS может иметь только один IP-адрес или не иметь IP-адреса.

IP-адрес необходим на сервере VIOS для обеспечения обмена с HMC через RMC, что является предпосылкой для выполнения динамических операций с LPAR. Таким образом, рекомендуется иметь в точности один IP-адрес на сервере VIOS, если вы хотите иметь возможность использовать динамический LPAR с VIOS.

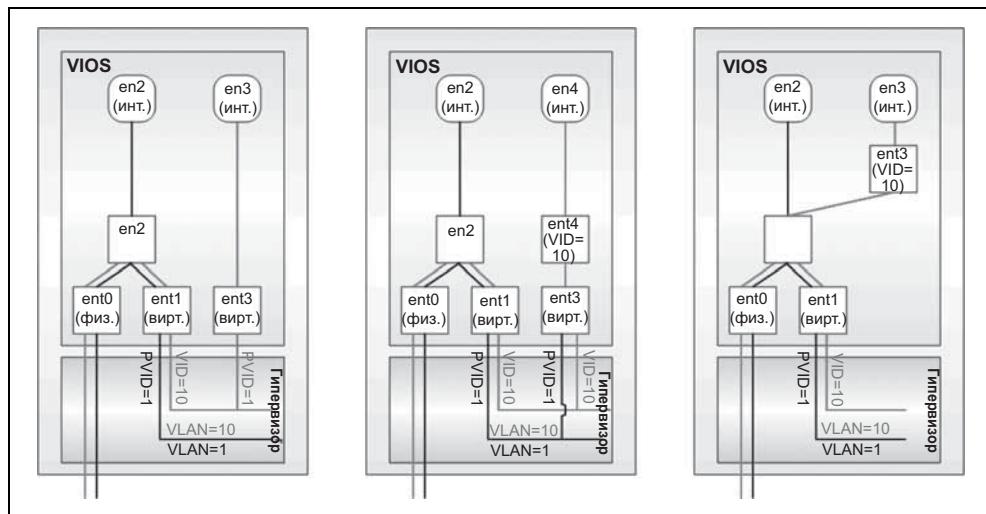


Рис. 3-20. Добавление виртуальных Ethernet-адаптеров в VIOS для сетей VLAN

Обмен с внешними сетями

Общий Ethernet-адаптер (SEA) на рисунке 3-19 сконфигурирован по умолчанию с PVID 1 и адаптером ent1. Это означает, что немаркированные пакеты или пакеты с VID = 1, получаемые SEA из внешней сети, ретранслируются адаптеру ent1. Виртуальный Ethernet-адаптер ent1 имеет дополнительный VID 10. Следовательно, пакеты, маркованные VID 10, будут ретранслироваться также и ent1.

Обработка исходящего во внешнюю сеть трафика зависит от метки VLAN на исходящих пакетах:

- ▶ Пакеты, маркованные VLAN 1, что соответствует PVID виртуального Ethernet-адаптера ent1, размаркировываются гипервизором POWER перед тем, как их примет ent1, пересылаются по мосту в ent0 с помощью SEA и отправляются наружу во внешнюю сеть.
- ▶ Пакеты, маркованные сетью VLAN, отличающейся от PVID 1 виртуального Ethernet-адаптера ent1, такой как VID 10, обрабатываются с неизменяемой VLAN-меткой.

В примере конфигурации виртуального Ethernet и VLAN рисунка 3-19 раздел 1 и раздел 2 имеют доступ к внешнему Ethernet через сетевой интерфейс ent0, использующий PVID 1.

- ▶ Так как пакеты с VLAN 1 используют PVID, то гипервизор POWER будет удалять VLAN-метки перед приемом этих пакетов ent0 разделов 1 и 2.
- ▶ Так как VLAN 1 является также PVID интерфейса ent1 адаптера SEA в сервере VIOS, то эти пакеты будут обрабатываться адаптером SEA без VLAN-меток и будут отправляться наружу во внешнюю сеть немаркованными.
- ▶ Следовательно, VLAN-неосведомленные устройства назначения во внешней сети будут также способны принимать эти пакеты.

Раздел 1 и раздел 3 имеют доступ к внешнему Ethernet через сетевой интерфейс en1 и VLAN 10.

- ▶ Эти пакеты посылаются наружу Ethernet-адаптером VLAN ent1, маркованные VLAN 10, через физический Ethernet-адаптер ent0.
- ▶ Виртуальный Ethernet-адаптер ent1 адаптера SEA в VIOS также использует VID 10 и будет принимать пакеты из гипервизора POWER с неизмененной VLAN-меткой. Затем пакет будет отправлен наружу через ent0 с неизмененной VLAN-меткой.
- ▶ Следовательно, эти пакеты будут способны принимать только устройства назначения с функцией VLAN.

Раздел 4 не имеет доступа к внешнему Ethernet.

В таблице 3-8 представлена сводка о том, какие разделы в конфигурации виртуального Ethernet из рисунка 3-19 и через какие сетевые интерфейсы могут вести обмен с внешними VLAN.

Таблица 3-8. Обмен VLAN с внешней сетью

Внешняя VLAN	Раздел / сетевой интерфейс
1	Раздел 1 / en0 Раздел 2 / en0 VIOS / en2
10	Раздел 1 / en1 Раздел 3 / en1

Если эта конфигурация должна быть расширена для обеспечения раздела 4 обменом с устройствами внешней сети, но не делая раздел 4 VLAN-осведомленным, то могут быть рассмотрены следующие варианты:

- ▶ В раздел 4 можно добавить дополнительный физический Ethernet-адаптер.
- ▶ В раздел 4 можно добавить дополнительный виртуальный Ethernet-адаптер ent1 с PVID=1: Тогда раздел 4 стал бы способен обмениваться с устройствами внешней сети с помощью VLAN=1 по умолчанию.
- ▶ В раздел 4 можно добавить дополнительный виртуальный Ethernet-адаптер ent1 с PVID=10: Тогда раздел 4 стал бы способен обмениваться с устройствами внешней сети с помощью VLAN=10.
- ▶ Можно добавить VLAN 2 как дополнительный VID в ent1 раздела VIOS, тем самым соединяя мостом VLAN 2 с внешним Ethernet точно так же, как VLAN 10: Тогда раздел 4 стал бы способен обмениваться с устройствами внешней сети с помощью VLAN=2. Это работало бы только тогда, когда VLAN 2 была бы также известна внешнему Ethernet и у VLAN2 были бы устройства во внешней сети.
- ▶ Раздел 3 мог бы работать в качестве маршрутизатора между VLAN 2 и VLAN 10 при активировании IP-ретрансляции в разделе 3 и добавлении маршрута по умолчанию через раздел 3 в раздел 4.

3.8.5. Ограничения и учитываемые факторы

Об ограничениях и учитываемых факторах для виртуального Ethernet и общих Ethernet-адаптеров (SEA) можно узнать в разделе 5.1.8 «Ограничения и учитываемые факторы» в конце обсуждения характеристик усовершенствованного виртуального Ethernet.

3.9. Ознакомление с виртуальным SCSI

Виртуальный ввод-вывод имеет отношение к виртуализированной реализации протокола SCSI. Виртуальный SCSI требует аппаратуры POWER5 с активированной функцией Advanced POWER Virtualization. Она обеспечивает поддержку виртуального SCSI для AIX 5L Version 5.3 и Linux (см. 1.2.8 «Поддержка нескольких операционных систем»).

Виртуальный ввод-вывод обуславливается следующими факторами:

- ▶ Усовершенствованными технологическими возможностями современных аппаратных средств и операционных систем, таких как POWER5 и IBM AIX 5L Version 5.3.
- ▶ Ценностью предложения, обеспечивающего вычисления по требованию и серверную консолидацию. Кроме того, виртуальный ввод-вывод дает более экономичную модель ввода-вывода, эффективнее использующую физические ресурсы с помощью совместного их использования.

На момент написания книги функции виртуализации платформы POWER5 поддерживали до 254 разделов, в то время как серверное оборудование обеспечивало только до 240 слотов ввода-вывода и 192 внутренних SCSI-дисков на одну машину. При необходимости обеспечить для каждого раздела, как правило, один слот ввода-вывода для подключения диска и еще один – для подключения к сети создавалось ограничение для количества разделов. Чтобы преодолеть эти физические ограничения, необходимо совместно использовать ресурсы ввода-вывода. Виртуальный SCSI дает средство выполнить это для устройств хранения SCSI.

Примечание. Вам встретятся в данной книге различные термины, относящиеся к самым разным компонентам, связанным с виртуальным SCSI. В зависимости от контекста эти термины могут варьироваться. В случае SCSI обычно употребляются термины *инициатор* (*initiator*) и *цель* (*target*), поэтому вам могут встретиться такие термины, как *инициатор виртуального SCSI* (*virtual SCSI initiator*) и *цель виртуального SCSI* (*virtual SCSI target*). В случае HMC используются термины *серверный адаптер виртуального SCSI* (*virtual SCSI server adapter*) и *клиентский адаптер виртуального SCSI* (*virtual SCSI client adapter*). Как правило, они обозначают одно и то же. При описании клиент-серверных отношений между разделами, участвующими в виртуальном SCSI, используются термины *размещающий раздел* (*hosting partition*) (означающий VIOS) и *размещаемый раздел* (*hosted partition*) (означающий клиентский раздел).

3.9.1. Доступ разделов к виртуальным SCSI-устройствам

В последующих разделах описываются архитектура виртуального SCSI и используемые протоколы.

Обзор клиент-серверной архитектуры виртуального SCSI

Виртуальный SCSI основан на взаимоотношениях клиент-сервер. Виртуальный сервер ввода-вывода VIOS владеет физическими ресурсами и действует как сервер или, в терминологии SCSI, целевое устройство. Логические разделы осуществляют доступ к ресурсам виртуального SCSI, обеспечиваемым сервером VIOS, как клиенты.

Виртуальные адAPTERы ввода-вывода конфигурируются с помощью HMC или Integrated Virtualization Manager (на системах меньших размеров). Предоставление ресурсов виртуального диска обеспечивается сервером VIOS.

Часто VIOS также называют размещающим разделом (*hosting partition*), а клиентские разделы – размещаемыми разделами (*hosted partitions*).

Физические диски, принадлежащие VIOS, могут либо экспортirоваться и целиком назначаться клиентскому разделу, либо разделяться на несколько логических томов. Эти логические тома затем могут назначаться различным разделам. Таким образом, виртуальный SCSI обеспечивает совместное использование как адAPTERов, так и дисковых устройств.

Чтобы сделать физический или логический том доступным клиентскому разделу, он назначается серверному адAPTERу виртуального SCSI в VIOS. АдAPTER виртуального SCSI представляется с помощью устройства vhost следующим образом:

vhost0 Available Virtual SCSI Server Adapter

Виртуализуемый диск или логический том представляются стандартным типом устройства AIX 5L: hdisk или логическим томом.

Примечание. Физический SCSI-диск или LUN назначается адAPTERу VSCSI по такой же процедуре, как и для логического тома. Один адAPTER VSCSI может иметь несколько назначенных ему физических дисков или номеров LUN, обраzуя несколько целевых устройств VSCSI.

Клиентский раздел осуществляет доступ к своим назначенным дискам через клиентский адаптер виртуального SCSI. Клиентский адаптер виртуального SCSI видит стандартные устройства SCSI и логические номера устройств LUN через свой виртуальный адаптер. Команды следующего примера показывают, как диски появляются в клиентском разделе AIX 5L:

```
# lsdev -Cc disk -s vscsi
hdisk2 Available Virtual SCSI Disk Drive

# lscfg -vpl hdisk2
hdisk2 111.520.10DDEC-V3-C5-T1-L810000000000 Virtual SCSI Disk Drive
```

SCSI-адаптер vhost представляет собой то же самое, что и обычный SCSI-адаптер, так как вы можете иметь из него доступ к нескольким дискам. Тем не менее, этот адаптер vhost может быть связан только с одним клиентским адаптером виртуального SCSI; любые диски, связанные с таким адаптером vhost, будут видимыми только для клиентского раздела, имеющего клиентский адаптер VSCSI, связанный с адаптером vhost.

Связывание адаптеров виртуального SCSI сервера VIOS с клиентскими адаптерами виртуального SCSI выполняется в НМС. Подробнее об этом можно узнать в разделе 4.4 «Базовый сценарий с сервером VIOS».

На рисунке 3-21 показан пример, в котором один физический диск разделяется на два логических тома внутри сервера VIOS. Каждому из двух клиентских разделов присваивается один логический том, к которому он осуществляет доступ через виртуальный адаптер ввода-вывода (клиентский адаптер VSCSI). Внутри раздела диск виден как обычный hdisk.

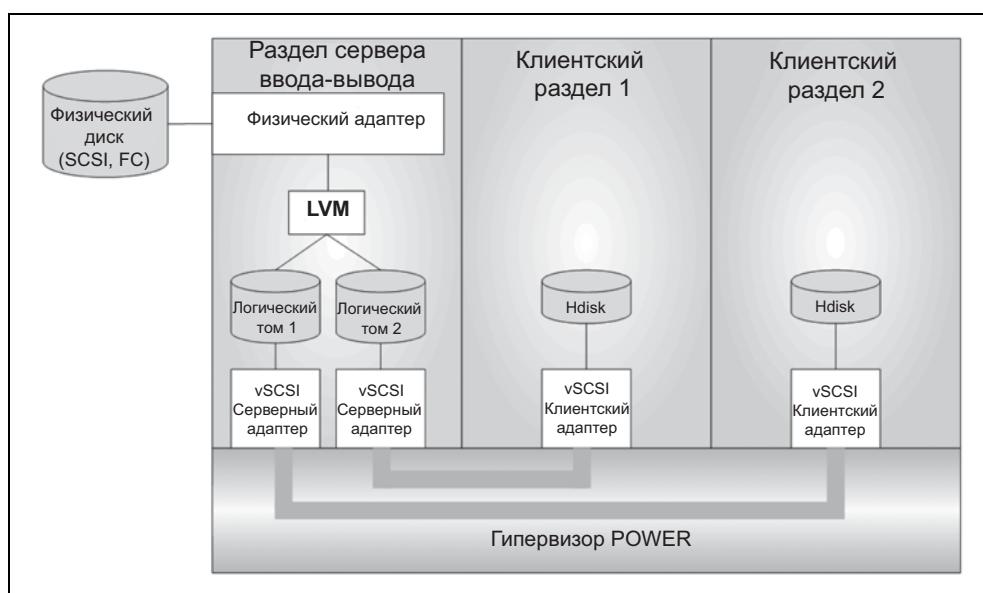


Рис. 3-21. Обзор архитектуры виртуального SCSI

Удаленный прямой доступ к памяти SCSI

Семейство стандартов SCSI обеспечивает множество различных транспортных протоколов, определяющих правила обмена информацией между инициаторами и целями SCSI. В виртуальном SCSI используется SCSI RDMA Protocol (SRP), который определяет правила обмена SCSI-информацией в среде, в которой инициаторы и цели SCSI имеют возможность непосредственно передавать информацию между соответствующими им адресными пространствами.

Запросы и ответы SCSI посылаются с помощью адаптеров виртуального SCSI, осуществляющих обмен через гипервизор POWER.

Тем не менее действительная передача данных выполняется непосредственно между буфером данных в клиентском разделе и физическим адаптером в VIOS с помощью протокола логического удаленного прямого доступа к памяти LRDMA (Logical Remote Direct Memory Access).

На рисунке 3-22 продемонстрирована передача данных с помощью LRDMA. Инициатор VSCSI клиентского раздела использует гипервизор для запроса доступа к данным из целевого устройства VSCSI. VIOS затем определяет, из какого физического адаптера необходимо передать эти данные, и отправляет его адрес гипервизору. Гипервизор связывает адрес этого физического адаптера с адресом буфера данных клиентского раздела, чтобы установить передачу данных непосредственно из физического адаптера VIOS в буфер данных клиентского раздела.

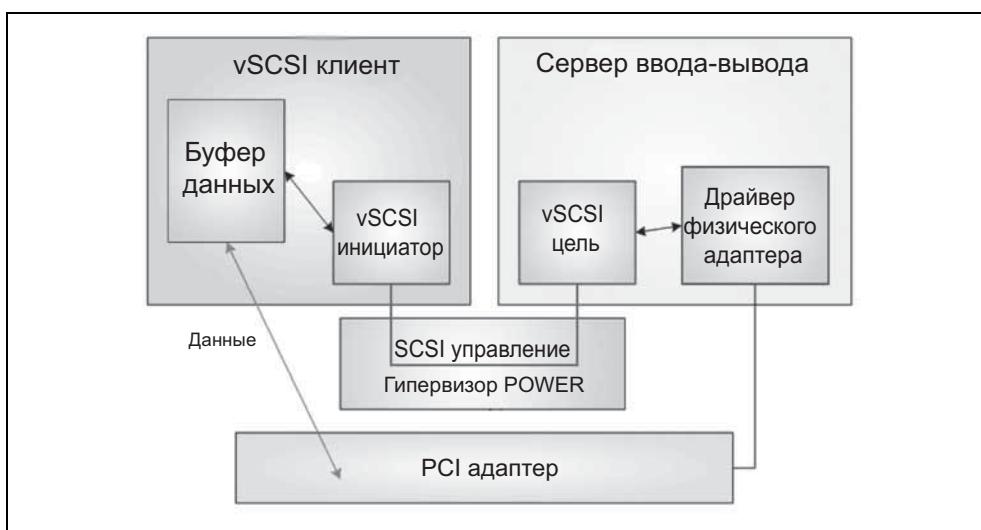


Рис. 3-22. Логический удаленный прямой доступ к памяти

Конфигурирование устройств AIX 5L для виртуального SCSI

Виртуальные адAPTERы ввода-вывода подключаются к виртуальному хост-мосту (host bridge), который воспринимается AIX 5L так же, как хост-мост PCI. Он представляется в ODM как устройство шины, родительским элементом для которого является sysplanar0. Виртуальные адAPTERы ввода-вывода представляются как адAPTERные устройства с виртуальным хост-мостом в качестве их родительского

элемента. В VIOS каждый логический том или физический том, экспортируемый в клиентский раздел, представляется виртуальным целевым устройством, которое является дочерним элементом серверного адаптера виртуального SCSI.

В клиентском разделе экспортируемые диски являются видимыми как обычные диски hdisk, но они определяются в подклассе vscsi. Они имеют в качестве родительского элемента клиентский адаптер виртуального SCSI.

На рисунках 3-23 и 3-24 показаны взаимоотношения устройства, используемых AIX 5L и VIOS для виртуального SCSI и их физических «двойников».

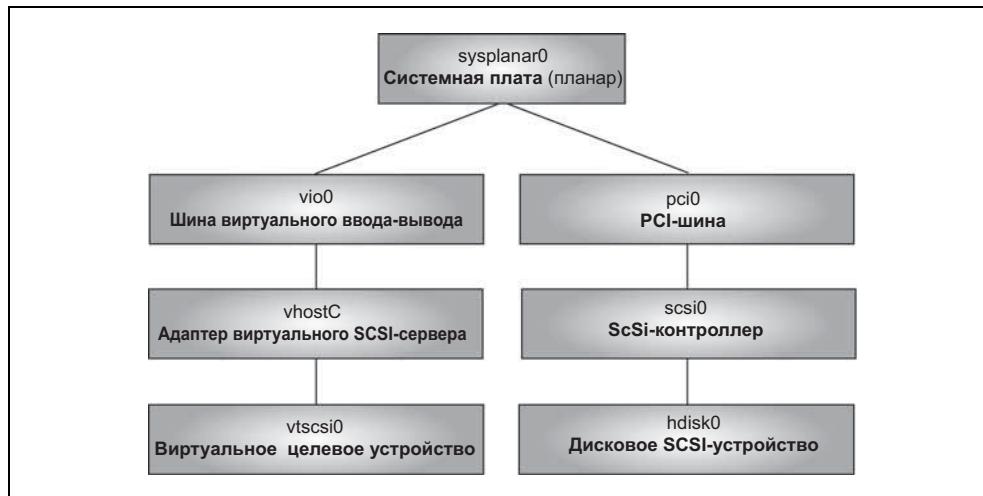


Рис. 3-23. Взаимоотношения устройств виртуального SCSI в VIOS

На рисунке 3-23 показаны взаимоотношения между физическим диском и целевым устройством SCSI в VIOS. Контроллер SCSI находится на одном иерархическом уровне с серверным адаптером виртуального SCSI, и это означает, что адаптер VSCSI может считаться тем же самым, что и адаптер SCSI.

На рисунке 3-24 показано, что клиентский адаптер клиентского раздела виртуального SCSI является тем же самым, что и контроллер SCSI, у каждого имеется SCSI-диск в качестве дочернего устройства. Клиентский адаптер VSCSI может иметь в качестве дочерних элементов несколько виртуальных SCSI-дисков, что вполне соответствует обычному адаптеру SCSI. Что касается физических SCSI-адаптеров, то они соединяются по принципу «один к одному», означающему, что только один клиентский адаптер виртуального SCSI может быть соединен с одним серверным адаптером виртуального SCSI.

Динамические разделы для устройств виртуального SCSI

Ресурсы виртуального SCSI могут динамически назначаться и удаляться в НМС. Серверные и клиентские адаптеры виртуального SCSI могут назначаться разделу и удаляться из него с помощью динамических операций с логическими разделами после настройки этих ресурсов в VIOS и их выделения в AIX 5L.

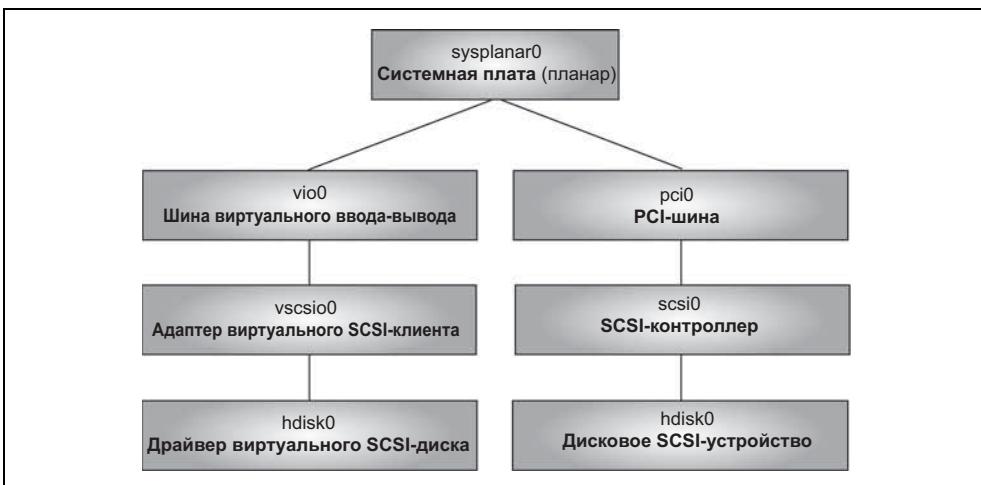


Рис. 3-24. Взаимоотношения устройств виртуального SCSI в клиентском разделе AIX 5L

При связывании физического устройства с виртуальным хост-адаптером перемещение виртуального SCSI-диска между клиентскими разделами похоже на перемещение физического адаптера. Переназначение адаптера VSCSI в серверном разделе новому клиентскому разделу сопровождается созданием нового клиентского адаптера клиентского раздела VSCSI. В клиенте выполняется команда `cfgmgr` и становится доступным новый диск VSCSI.

Оптические устройства виртуального SCSI

В Virtual I/O Server V1.2 добавлена дополнительная поддержка оптических устройств, включая DVD-ROM и DVD-RAM. В настоящее время DVD-RAM является единственным поддерживаемым оптическим устройством с записью.

Запись DVD+RW и DVD-RW не поддерживается.

Добавлено ограничение для реализации виртуального оптического устройства, заключающееся в том, что только одно виртуальное целевое оптическое устройство может быть создано для любого конкретного физического оптического устройства. Это гарантирует, что два клиента виртуального SCSI не смогут записывать на один и тот же носитель DVD-RAM и портить друг другу данные.

Конфигурирование оптического устройства в качестве устройства виртуального SCSI осуществляется так же, как и конфигурирование диска или логического тома в устройство VSCSI. С использованием либо нового, либо заранее определенного адаптера `vhost`, связанного с клиентским разделом, выполняется следующая команда:

```
$ mkdev -vdev cd0 -vadapter vhost0
```

При этом будет создано виртуальное целевое оптическое SCSI-устройство в VIOS со следующим сообщением:

```
vtopt0 Available Virtual Target Device - Optical Media
```

В клиентском разделе выполните команду `cfgmgr`, и устройство cd0 будет сконфигурировано для использования. Теперь можно подмонтировать CD-устройство, используя команду `mkdvd` для носителя DVD-RAM.

Для экспорта этого физического оптического устройства в другой клиентский раздел выполните следующее:

- ▶ Удалите существующее виртуальное оптическое устройство из связанного с ним клиентского раздела.
- ▶ Удалите виртуальную оптическую SCSI-цель в VIOS.
- ▶ Выполните команду `mkdev` в VIOS с использованием выбранного клиентского `vhost`-адаптера.
- ▶ Выполните команду `cfgmgr` в новом клиенте для конфигурирования устройства cd0.

3.9.2. Основные учитываемые факторы

Необходимо принимать во внимание следующие факторы при реализации виртуального SCSI:

- ▶ На момент написания книги виртуальный SCSI поддерживал Fibre Channel, параллельный SCSI, SCSI RAID-устройства и оптические устройства, включая DVD-RAM и DVD-ROM. Другие протоколы, например SSA и для ленточных устройств, не поддерживались.
- ▶ Логический том в VIOS, используемый как виртуальный SCSI-диск, не может превышать размер 1 ТБ или охватывать несколько физических томов.
- ▶ Протокол SCSI определяет обязательные и необязательные команды. Хотя виртуальный SCSI поддерживает все обязательные команды, не все необязательные команды поддерживаются.

Важно. Частедование (striping), зеркалирование, активация перемещения поврежденных блоков и использование логических томов в качестве виртуальных дисков в VIOS с охватом нескольких физических томов не рекомендуются.

Для подтверждения того, что логический том не охватывает несколько дисков, выполните следующее:

```
$ lslv -pv app_vg  
app_vg:N/A  
PV          COPIES      IN BAND DISTRIBUTION  
hdisk5     320:000:000  99% 000:319:001:000:000
```

В результате списка должен появиться только один диск.

Факторы, связанные с установкой и миграцией

При установке и миграции следует учитывать следующие главные факторы:

- ▶ Рекомендуется планировать размеры группы томов rootvg клиентского раздела до создания логических томов. Увеличение rootvg с помощью расширения связанного с ним логического тома VIOS не поддерживается. О расширении групп томов см. «Увеличение группы томов клиентского раздела» в разделе 6.4.4.

- ▶ В настоящее время миграция с физического SCSI-диска на виртуальное SCSI-устройство не поддерживается. Все виртуальные SCSI-устройства, созданные в VIOS, считаются новыми устройствами. При миграции с физического на виртуальное устройство потребуются резервное копирование и восстановление данных.
- ▶ В VIOS применяется несколько методов для уникальной маркировки диска, используемого в качестве виртуального SCSI-диска. К этим методам относятся:
 - Уникальный идентификатор устройства – Unique device identifier (UDID)
 - Идентификатор тома IEEE – IEEE volume identifier
 - Физический идентификатор тома – Physical Volume Identifier (PVID)

Независимо от используемого метода виртуальное устройство всегда будет выглядеть одинаково для клиента VSCSI. Используемый метод не оказывает влияния на схему физического хранилища, управляемого VIOS.

Предпочтительным методом идентификации диска для виртуальных дисков является UDID. Метод UDID используют MPIO-устройства. В идеальном случае когда-то в будущем произойдет такое слияние устройств, при котором все они смогут использовать метод UDID. В настоящее время в большинстве программных продуктов для дисковых хранилищ с множественным доступом без MPIO используется метод PVID, а не UDID. Когда произойдет слияние всех устройств к использованию метода UDID, существующие устройства, создаваемые с использованием старых методов (таких как PVID или IEEE volume ID), будут по-прежнему использоваться без их изменения. Клиентам следует знать, что для использования преимуществ некоторых будущих действий или функций, выполняемых в LPAR VIOS, старым устройствам может вначале потребоваться миграция данных, то есть некоторый вид резервного копирования и восстановления прикрепленных дисков. При этом могут выполняться действия из следующего неполного списка:

- Преобразование из среды без MPIO в MPIO.
- Преобразование дисковой идентификации из PVID-метода в UDID-метод.
- Удаление и повторное обнаружение записей Disk Storage ODM.
- При некоторых обстоятельствах обновление ПО множественного доступа без MPIO.
- Возможные будущие усовершенствования в виртуальном вводе-выводе.
- ▶ Сам по себе виртуальный SCSI не имеет каких-либо ограничений, касающихся количества поддерживаемых устройств или адаптеров. На момент написания книги VIOS поддерживал максимум 1024 виртуальных слота ввода-вывода на одном сервере. Для одного раздела может быть назначено до 256 виртуальных слотов ввода-вывода. Для реализации каждого слота ввода-вывода необходимы некоторые ресурсы. Следовательно, размер VIOS накладывает ограничение на количество виртуальных адаптеров, которые могут быть сконфигурированы.

Учет факторов производительности

Использование виртуальных SCSI-устройств оказывается на производительности. Важно понимать, что для выполнения вызовов SCSI Remote DMA и гипервизора POWER виртуальный SCSI будет занимать дополнительные циклы ЦП при обработке запросов ввода-вывода. При более интенсивной нагрузке ввода-вывода на виртуальные SCSI-устройства в VIOS будет использоваться больше циклов ЦП.

Виртуальному SCSI необходимо около 20 процентов мощности ЦП для обеспечения потоковой обработки больших блоков или даже 70 процентов для наихудшего случая с малыми блоками и большим объемом нагрузки транзакций для среды с двойным дисковым адаптером. При условии доступности достаточной вычислительной мощности ЦП по производительности виртуальный SCSI сравним с выделенными устройствами ввода-вывода.

К подходящим для виртуального SCSI применением относятся загрузочные диски для операционных систем или веб-серверы, на которых обычно буферизуется большой объем данных. При планировании конфигурации виртуального ввода-вывода производительность является важным аспектом, заслуживающим тщательного учета.

Виртуальный SCSI работает на низком уровне приоритетов прерываний, в то время как виртуальный Ethernet работает с прерываниями высоких приоритетов, что вызвано различиями задержек между дисками и сетевыми адаптерами. Клиент, создающий большой объем сетевой деятельности, потенциально способен влиять на производительность клиента, которому необходим большой объем работы с дисками, из-за более высокого приоритета прерываний виртуального Ethernet. Для разделения этих двух клиентов с высокой потребностью в производительности можно предусмотреть либо большой VIOS с увеличенными процессорными ресурсами, либо второй VIOS.

3.10. Ознакомление с Partition Load Manager

ПО Partition Load Manager (PLM) является частью функции Advanced POWER Virtualization и помогает клиентам добиваться максимального использования ресурсов процессора и памяти динамических логических разделов с LPAR, в которых работает AIX 5L.

Partition Load Manager является менеджером ресурсов, назначающим и перемещающим ресурсы на основе определенных политик и использования ресурсов. PLM управляет памятью, выделенными процессорами и разделами, использующими технологию микроразделов для перераспределения ресурсов. Это придает дополнительную гибкость уже гибкой технологии микроразделов, обеспечиваемой гипервизором POWER.

Тем не менее у PLM нет сведений о степени важности нагрузок, выполняемых в разделах, и он не может перераспределять приоритеты на основе изменения типов нагрузок. PLM не управляет разделами с Linux и i5/OS.

Partition Load Manager устанавливается в разделе или в другой системе с AIX 5L Version 5.2 ML4 или AIX 5L Version 5.3. PLM, и клиенты не поддерживаются Linux или i5OS. В разделе или системе с Partition Load Manager у вас также могут быть установлены другие приложения. Один экземпляр Partition Load Manager может управлять только одним сервером.

Partition Load Manager использует клиент-серверную модель для отчетов и управления использованием ресурсов. Клиенты (управляемые разделы) уведомляют PLM-сервер при недостаточном или излишнем использовании ресурсов. При получении уведомления об одном из этих событий PLM-сервер принимает решение о распределении ресурсов на основе файла политики, определяемого администратором.

На рисунке 3-25 показан общий вид компонентов Partition Load Manager. На этом рисунке PLM-сервер будет получать уведомление от раздела базы данных, которому требуются дополнительные процессорные ресурсы. Раздел веб-сервера также будет уведомлять PLM-сервер об избытке процессорных ресурсов. С помощью файла политики PLM-сервер будет определять, что приемлемым будет взять процессорные ресурсы у раздела веб-сервера и назначить их разделу базы данных и затем осуществить это перераспределение ресурсов, используя HMC.

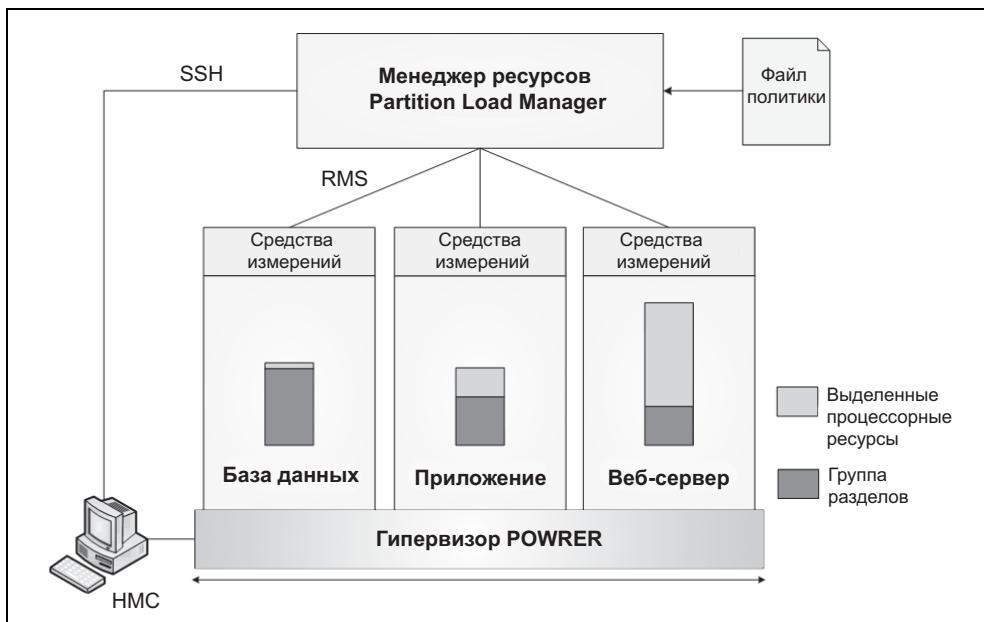


Рис. 3-25. Общий вид Partition Load Manager

После возврата раздела базы данных к нормальной нагрузке PLM-сервер будет определять, необходимо ли перераспределять ресурсы и возвращать их разделу веб-сервера.

3.11. Integrated Virtualization Manager

Этот раздел в кратком виде знакомит с Integrated Virtualization Manager. Более подробную информацию и детальные шаги конфигурирования можно найти в книге *Virtual I/O Server Integrated Virtualization Manager*, REDP-4061.

Основное решение по управлению оборудованием компания IBM разработала на основе специализированного сервера, названного консолью управления оборудованием HMC (Hardware Management Console) и выполненного в виде отдельного настольного или устанавливаемого в стойку ПК.

HMC является централизованным пунктом управления аппаратными ресурсами. С одной консоли HMC можно управлять несколькими системами POWER5, а с двух HMC можно управлять одним набором серверов в конфигурации двойной активации.

НМС управляет оборудованием, используя стандартное Ethernet-соединение с сервисным процессором каждой системы POWER5. Взаимодействуя с сервисным процессором, НМС может создавать логические разделы, управлять ими и изменять их, изменять конфигурацию оборудования управляемой системы и управлять сервисными вызовами.

Для небольшой или распределенной среды всех функций НМС не требуется и может не подходить вариант с установкой дополнительного персонального компьютера.

IBM разработала IVM – решение по управлению оборудованием, наследующее большинство функций НМС, но ограничивающее управлением одного сервера без необходимости иметь выделенный ПК. Это решение позволяет администратору уменьшать время установки системы. IVM встроен в продукт Virtual I/O Server, обеспечивающий виртуализацию ввода-вывода и процессоров в системах POWER5.

3.11.1. Основные правила установки IVM

Так как одной из целей IVM является администрирование, то к конфигурации и установке сервера применимы некоторые свойственные ему правила. Для управления системой с использованием IVM в помощь вам предлагаются следующие основные принципы:

- ▶ Система конфигурируется в режиме *Factory Default*, означающем, что предопределается один раздел с сервисными полномочиями, который владеет всеми аппаратными ресурсами. Если система не конфигурируется в режиме Factory Default потому, что она уже имеет разделы или подключена к НМС, то вы можете вернуть систему в режим Factory Default с помощью ASMI.
- ▶ Предопределенный раздел автоматически запускается при включении системы. К этому разделу подключены физическая панель управления и последовательные порты.
- ▶ Должна быть активирована функция APV. При заказе этой функции вместе с системой она активирована по умолчанию; если этого нет, то она может быть активирована с помощью ASMI.
- ▶ В предопределенном разделе должен быть установлен Virtual I/O Server Version 1.2 или более поздней версии.

Затем VIOS автоматически распределяет все ресурсы ввода-вывода. Все другие LPAR конфигурируются с помощью IVM, встроенного в VIOS. У клиентских разделов нет сконфигурированных физических устройств ввода-вывода. Они осуществляют доступ к дискам, сети и оптическим устройствам только как виртуальные устройства через VIOS. Эта конфигурация может быть выполнена с помощью GUI или интерфейса командной строки в VIOS. Чтобы настроить конфигурацию системы, администратор может использовать браузер для соединения с IVM.

На рисунке 3-26 показан пример конфигурирования с помощью IVM. VIOS владеет всеми физическими адаптерами, в то время как другие два раздела конфигурируются для использования только виртуальных устройств.

Тесное взаимодействие VIOS и IVM позволяет администратору управлять системой с разделами без НМС. Программное обеспечение, которое обычно работало

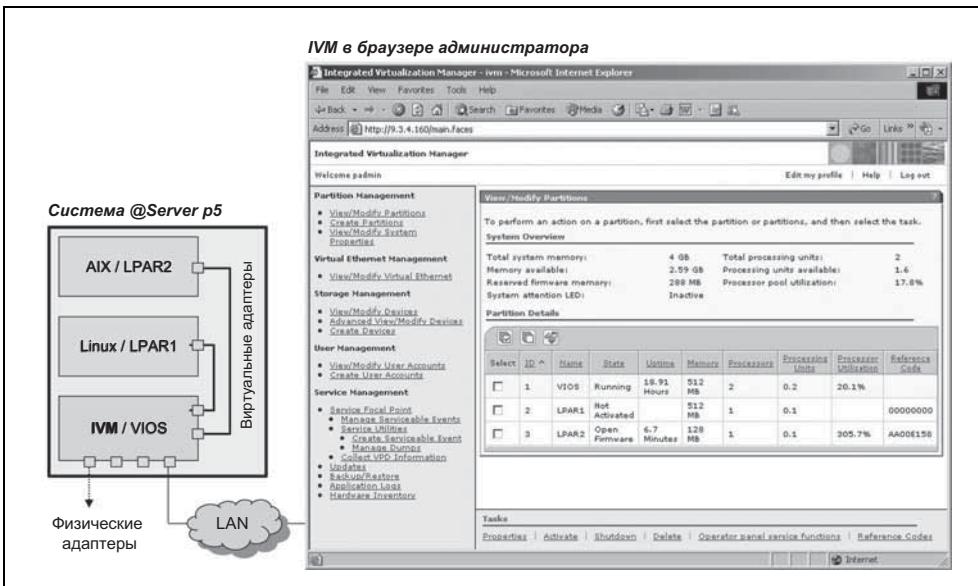


Рис. 3-26. Конфигурирование Integrated Virtualization Manager

в НМС, было переработано для встраивания в VIOS с уменьшением его функций до объема, необходимого для конфигурационной модели с IVM. Так как IVM работает с использованием системных ресурсов, то при его разработке стремились, чтобы потребление этих ресурсов было сведено к минимуму.

IVM не взаимодействует с сервисным процессором системы. В VIOS было разработано специальное устройство, названное *виртуальным каналом управления* – VMC (*Virtual Management Channel*) и обеспечивающее прямое конфигурирование гипервизора без необходимости установки дополнительного сетевого соединения. Это устройство активируется по умолчанию при установке VIOS в качестве первого раздела.

VMC позволяет IVM обеспечивать базовые функции логических разделов:

- ▶ Конфигурирование логических разделов
- ▶ Действия по загрузке, запуску и останову для отдельных разделов
- ▶ Отображение состояния разделов
- ▶ Управление виртуальным Ethernet
- ▶ Управление виртуальным хранилищем
- ▶ Обеспечение базового управления системой

Так как IVM выполняется в одном LPAR, то его сервисные функции ограничены, и необходимо использовать ASMI. Например, включение системы должно осуществляться физическим нажатием кнопки подачи питания на систему или с помощью удаленного доступа к ASMI, так как IVM не работает при выключенном системе. ASMI и IVM вместе обеспечивают базовое, но эффективное решение для одного сервера с разделами.

Управление LPAR с помощью IVM осуществляется через веб-интерфейс, разработанный, чтобы сделать администрирование более удобным и быстрым, чем у полного решения с HMC. Будучи интегрированным в программный код VIOS, IVM также выполняет все задачи виртуализации, которые обычно требуют запуска команд VIOS.

В IVM управление системой осуществляется по-другому, чем в HMC. Новый администратор системы POWER5 быстро научится необходимым приемам, а специалисту по HMC придется перед использованием IVM изучить эти различия.

3.11.2. Конфигурирование разделов с помощью IVM

Конфигурирование LPAR выполняется назначением процессоров, памяти и виртуального ввода-вывода с помощью мастера с графическим веб-интерфейсом. На каждом шаге этого процесса администратору задаются простые вопросы и предлагается набор возможных ответов. Большинство параметров, связанных с установкой LPAR, спрятаны во время создания для облегчения настройки и могут быть при необходимости изменены после создания раздела в его свойствах.

Назначенные LPAR ресурсы выделяются немедленно и не могут быть далее доступными другим разделам, независимо от того, находится ли этот LPAR в активном или в выключенном состоянии. Такое поведение делает управление более прямым и отличающимся от управления системой с помощью HMC, в котором допускается передача ресурсов.

Разделы LPAR в управляемой с помощью IVM системе изолируются точно так же, как во всех системах POWER5, и могут взаимодействовать только с использованием виртуальных устройств. Только IVM позволяет выполнять следующие ограниченные действия с другими разделами LPAR:

- ▶ Включение и выключение питания
- ▶ Постепенное закрытие операционной системы
- ▶ Создание и удаление
- ▶ Просмотр и изменение конфигурации

Так как IVM не обеспечивает динамических операций для клиентских разделов, то в нем нет конфигурирования минимального, желаемого и максимального значений.

Процессоры

LPAR может быть определен либо с выделенными, либо с общими процессорами. Когда для раздела выбираются общие процессоры, то мастер позволяет администратору выбрать только количество активируемых виртуальных процессоров. Каждому виртуальному процессору неявно присваивается 0,1 процессорных единиц, и LPAR создается в режиме без верхнего предела с весом 128.

Количество процессорных единиц, режим без верхнего предела и вес можно изменять, модифицируя конфигурацию LPAR после его создания.

Виртуальный Ethernet

Управляемая IVM система конфигурируется с четырьмя предопределенными виртуальными сетями Ethernet, и у каждой из них имеется идентификатор виртуального Ethernet в диапазоне от 1 до 4. В каждом LPAR может быть до двух виртуальных Ethernet-адаптеров, которые могут подключаться к любой из четырех виртуальных сетей системы.

Каждую виртуальную Ethernet-сеть VIOS может связать мостом с физической сетью с помощью только одного физического адаптера. Один физический адаптер может связывать мостом только одну виртуальную сеть.

Виртуальная Ethernet-сеть является загрузочным устройством и может использоваться для установки операционной системы LPAR.

Виртуальное хранилище

Каждый LPAR снабжается одним или несколькими виртуальными SCSI-дисками, использующими один виртуальный SCSI-адаптер. Виртуальные диски являются загрузочными устройствами и считаются операционной системой обычными SCSI-дисками.

Виртуальное оптическое устройство

Каждое оптическое устройство, имеющееся в разделе VIOS (либо CD-ROM, либо DVD-ROM, либо DVD-RAM), может быть виртуализовано и назначено любому логическому разделу в однозначном соответствии и используя тот же виртуальный SCSI-адаптер, который предоставлен для виртуальных дисков. Виртуальные оптические устройства могут использоваться для установки операционной системы и, в случае DVD-RAM, для выполнения резервного копирования.

Виртуальный TTY

Чтобы устанавливать и управлять LPAR, IVM обеспечивает среду виртуального терминала для работы с консолью LPAR. Когда определяется новый LPAR, то ему автоматически назначается клиентский виртуальный последовательный адаптер, используемый в качестве консольного устройства по умолчанию. С помощью IVM создается соответствующий серверный виртуальный терминальный адаптер, который связывается с виртуальным клиентом клиентского LPAR.

3.12. Динамические операции с LPAR

Вам необходимо знать несколько вещей для выполнения динамических операций с LPAR, которые касаются как виртуализованных, так и невиртуализованных серверных ресурсов:

- ▶ Убедитесь в том, что такие ресурсы, как физические и виртуальные адаптеры, добавляемые и перемещаемые между разделами, не используются другими разделами. Это означает соответствующую очистку на клиентской стороне с удалением их из системы или выводом их из работы PCI-процедурами горячего подключения с помощью SMIT, если они являются физическими адаптерами.
- ▶ У вас не будет возможности динамического добавления дополнительной памяти для раздела, у которого уже достигнут максимум, определенный в его профиле.

- ▶ НМС должен иметь возможность связи с логическими разделами через сеть для RMC-соединений.
- ▶ Учитывайте факторы производительности при удалении памяти из логических разделов.
- ▶ Выполняемые приложения должны уметь работать с динамическими LPAR. если они динамически выделяют и освобождают ресурсы, то есть быть способными изменять свои размеры и приспосабливаться к изменениям в аппаратных ресурсах.

В разделе 6.1 «Динамические операции с LPAR» показано, как вам выполнять такие операции в работающей системе.

3.13. Концепции виртуального ввода-вывода в Linux

Кроме AIX 5L в IBM System p5 также может использоваться Linux. Linux может устанавливаться в разделе с выделенным или общим процессором. Работающий в разделе Linux может использовать физические и виртуальные устройства. Он также может участвовать в виртуальном Ethernet и осуществлять доступ к внешним сетям через общие Ethernet-адAPTERЫ – SEA (Shared Ethernet Adapters). Раздел Linux может использовать виртуальные SCSI-диски. Linux также может обеспечивать для других разделов с Linux некоторые из виртуализованных сервисов, которые IBM Virtual I/O Server обычно обеспечивает для разделов с AIX 5L и Linux.

Общими являются следующие термины и определения:

Виртуальный клиент ввода-вывода

Любой раздел, использующий виртуализованные устройства, обеспечиваемые другими разделами.

Виртуальный сервер ввода-вывода

Любой сервер раздела ввода-вывода (Virtual I/O Server, VIOS), предоставляющий виртуализованные устройства для использования другими разделами.

Более точно: существуют два разных типа VIOS, доступных для System p5:

APV VIOS

Advanced POWER Virtualization Virtual I/O Server компании IBM для pSeries p5. Это специализированное функциональное устройство, которое может использоваться только как VIOS и не предназначено для выполнения приложений общего назначения.

Linux VIOS

Реализация набора функций VIOS в Linux.

Виртуальный клиент ввода-вывода (VIO-клиент) и виртуальный сервер ввода-вывода (VIO-сервер) являются ролями. По такому определению система одновременно могла бы быть и VIO-клиентом, и VIO-сервером. В большинстве случаев этого следует избегать, так как при этом усложняется администрирование вследствие сложных зависимостей.

Важно. В остальной части этой книги, за исключением данной главы, термины *Virtual I/O Server* или *VIOS*, упоминаемые без особых пояснений, означают APV VIOS. В этом разделе мы подчеркнуто называем это устройство APV VIOS, чтобы отличать от Linux VIOS.

В разделе System p5 могут размещаться четыре различных типа систем:

- ▶ AIX 5L: может быть только VIO-клиентом.
- ▶ Linux: может быть VIO-клиентом и VIO-сервером.
- ▶ APV VIOS: только VIO-сервер.
- ▶ i5/OS на отдельных системах.

Теперь мы объясним основные принципы работы VIO-клиентов и VIO-серверов Linux. Более детальное описание и практическое руководство вы можете найти в публикациях, перечисленных в разделе 3.13.5 «Что читать дальше».

3.13.1. Драйверы устройств Linux для виртуальных устройств IBM System p5

Компания IBM сотрудничала с разработчиками Linux в создании драйверов устройств для ядра Linux 2.6, и это позволяет Linux использовать функции виртуализации IBM System p5.

В таблице 3-9 приведены все модули этого ядра для виртуальных устройств IBM System p5.

Таблица 3-9. Модули ядра для виртуальных устройств IBM System p5

Модуль ядра Linux 2.6	Поддерживаемое виртуальное устройство	Местонахождение файлов исходного кода относительно /usr/src/linux/drivers/
hvcs	сервер виртуальной консоли	char/hvc*
ibmveth	виртуальный Ethernet	net/ibmveth*
ibmvscsis	виртуальный SCSI-клиент/инициатор	scsi/ibmvscsi*
ibmvscsis	виртуальный SCSI-сервер/цель	scsi/ibmvscsi*

Исходный код ядра Linux 2.6 может быть загружен с адреса:

<ftp://ftp.kernel.org/pub/linux/kernel/v2.6/>

Предварительно скомпилированные модули ядра Linux включены в некоторые дистрибутивы Linux.

3.13.2. Linux как VIO-клиент

Linux, работающий в разделе System p5, может использовать виртуальные Ethernet-адаптеры и виртуальные устройства, обеспечиваемые серверами VIOS. Linux VIO-клиент может использовать одновременно как APV VIOS, так и Linux VIOS.

Виртуальная консоль

System p5 обеспечивает виртуальную консоль /dev/hvc0 каждому разделу с Linux.

Виртуальный Ethernet

Для использования виртуальных Ethernet-адаптеров с Linux должен быть загружен модуль ядра Linux ibmveth. Если используются сети IEEE 802.1Q VLAN, то, кроме этого, должен быть доступен модуль ядра Linux 8021q. Виртуальные Ethernet-адAPTERы используют ту же самую схему именования, как и физические Ethernet-адAPTERы, например eth0 для первого адAPTERа. Сети VLAN конфигурируются с помощью команды `vconfig`.

Linux может использовать построение общих сетей с другими разделами и использовать совместный доступ к внешним сетям с другими разделами с Linux и AIX 5L, например, через общий Ethernet-адAPTER SEA (Shared Ethernet Adapter) APV VIOS.

Виртуальный SCSI-клиент

Виртуальный SCSI-клиент IBM для Linux реализуется модулем ядра Linux ibmvscsic. При загрузке этого модуля ядра он будет сканировать и автоматически обнаруживать любые виртуальные SCSI-диски, обеспечиваемые серверами VIOS. Обнаружение также может быть запущено вручную после добавления дополнительных виртуальных SCSI-дисков к виртуальному SCSI-адAPTERу в VIOS.

Виртуальные SCSI-диски будут именоваться точно так же, как обычные SCSI-диски, например /dev/sda для первого SCSI-диска или /dev/sdb3 для третьего раздела второго SCSI-диска.

MPIO

Linux поддерживает базовые реализации и специальные реализации некоторых изготовителей множественного ввода-вывода MPIO (Multi-Path I/O), и некоторые изготовители обеспечивают дополнительные драйверы устройств с функцией MPIO для Linux.

Внимание. В настоящее время Linux VIO-клиент не поддерживает MPIO для доступа к одним и тем же дискам, использующим два VIOS.

Следует помнить, что MPIO также может быть реализован в VIOS для обеспечения резервного доступа к внешним дискам для VIO-клиента. Но реализация MPIO в VIOS, а не в VIO-клиенте, не обеспечивает в той же степени высокую доступность VIO-клиента, так как VIO-клиент должен выключаться при закрытии единственного VIOS, например, когда VIOS обновляется. Различия между MPIO в VIO-клиенте и в VIOS приведены на рисунке 3-27.

Зеркалирование

Linux может зеркаливать диски с помощью RAID-Tools. Таким образом, для резервирования вы можете зеркаливать каждый виртуальный диск, обеспечиваемый одним VIOS, на другом виртуальном диске, обеспечиваемом другим VIOS.

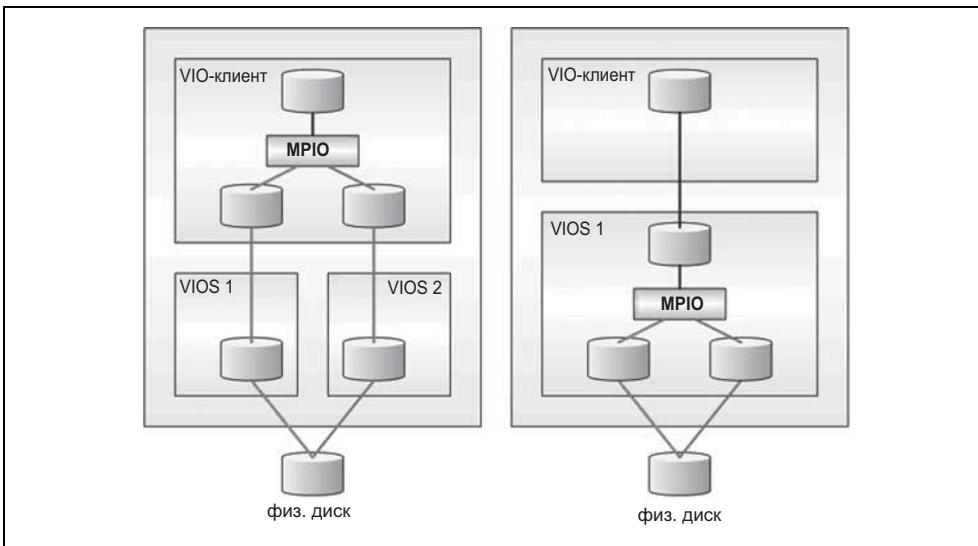


Рис. 3-27. Реализация MPIO в VIO-клиенте и в VIO-сервере

Внимание. Будьте внимательны, так как использование RAID-Tools Linux для зеркалирования разделов boot и root и предоставление системе с Linux возможности загружаться с зеркальзованных дисков могут потребовать модификации схемы загрузки по умолчанию.

Следует помнить, что зеркалирование также может реализовываться в VIOS для обеспечения резервного доступа к внешним дискам для VIO-клиента. Но реализация зеркалирования в VIOS, а не в VIO-клиенте, не обеспечивает в той же степени высокую доступность VIO-клиента, так как VIO-клиент должен выключаться при закрытии единственного VIOS, например, когда VIOS обновляется. Различия между зеркалированием в VIO-клиенте и в VIOS приведены на рисунке 3-28.

Примечание. В настоящее время LVM-зеркалирование виртуальных дисков в APV VIOS не рекомендуется.

LVM

Менеджер логических томов Linux OS Logical Volume Manager может использовать любую смесь виртуальных и физических дисков.

Внимание. Будьте внимательны, так как использование LVM Linux для root-раздела и предоставление системе с Linux возможности загружаться с root-файлом в логическом томе могут потребовать модификации схемы загрузки по умолчанию.

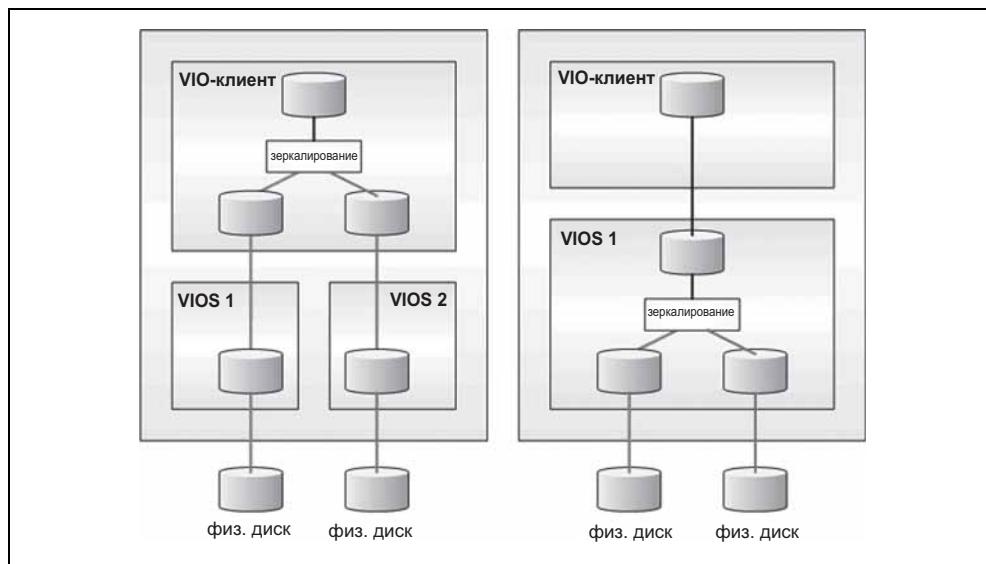


Рис. 3-28. Реализация зеркалирования в VIO-клиенте и в VIO-сервере

3.13.3. Linux как VIO-сервер

Linux также может обеспечивать другим Linux-разделам некоторые сервисы виртуализации, которые APV VIOS IBM System p5 обычно обеспечивает разделам с AIX 5L и с Linux.

Ограничение. Linux VIO Server не поддерживается AIX 5L. Для VIO-клиентов с AIX 5L поддерживается только APV VIOS.

Создание Ethernet-мостов

Для обеспечения функции построения мостов на Уровне 2 в Linux, например, между виртуальными и физическими Ethernet-адаптерами функция построения мостов активируется в процессе создания ядра. Должна быть установлена утилита bridge-utils RPM, обеспечивающая доступ к команде `brctl`, которая используется для установки и конфигурирования моста. Для ограничения доступа может использоваться команда `ipfilter`.

Создание моста между физическим и виртуальным Ethernet-адаптерами в случае Linux показано на рисунке 3-29. Обратите внимание, что IP-адрес для обеспечивающего мост Linux-раздела определен на `br0`, а не на `eth0`, который теперь является членом моста.

Маршрутизация

Linux также может действовать в качестве маршрутизатора к внешним сетям. Должна быть активирована IP-ретрансляция. Для ограничения доступа может использоваться команда `ipfilter`.

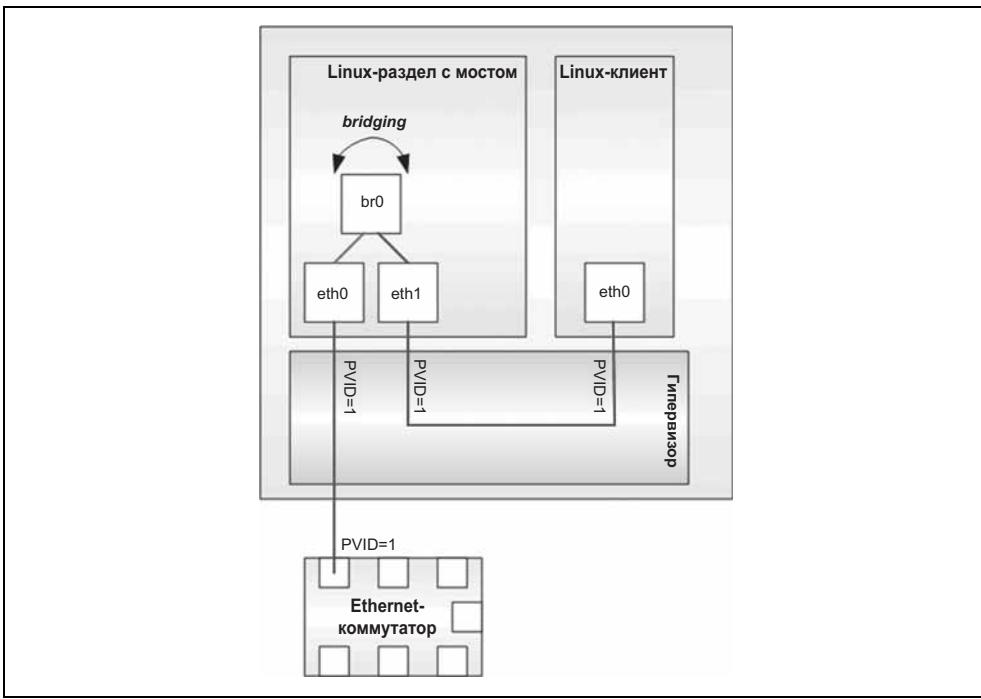


Рис. 3-29. Связывание мостом виртуального и физического Ethernet-адаптеров с Linux

Виртуальный SCSI-сервер

Виртуальный SCSI-сервер реализуется с помощью модуля ядра Linux `ibmvscsis`. Вначале нужно сконфигурировать виртуальные серверные SCSI-адAPTERы. Затем можно добавить следующее для использования в качестве виртуальных дисков для Linux VIO-клиентов:

- ▶ Физические диски, например `/dev/sdc`
- ▶ Разделы физических дисков, например `/dev/sdd2`
- ▶ Логические тома, например `/dev/datavg/lv01`
- ▶ Файлы, монтируемые с обратной связью (loopback-mounted files)

Преимущество использования файлов с обратной связью состоит в том, что клонирование и резервное копирование виртуальных дисков могут легко выполняться с помощью команды `cpr`. При этом виртуальные диски не должны быть неактивными.

3.13.4. Учитываемые факторы

При использовании Linux в качестве VIO-клиента или VIO-сервера следует учитывать следующее:

- ▶ Поддерживаемыми VIO-клиентами являются AIX 5L V5.3 и специальные дистрибутивы Linux.
- ▶ Для VIO-клиентов AIX 5L V5.3 IBM поддерживает только APV VIOS.

- ▶ Linux VIO-клиенты могут использовать APV VIOS и Linux VIO Server.
- ▶ Использование APV VIOS может потребовать приобретения дополнительной функции, зависящей от модели IBM System p5.
- ▶ Linux VIO Server в настоящее время доступен только в SUSE Linux Enterprise Server 9 (SLES V9).
- ▶ В настоящее время не поддерживается MPIO на Linux-клиентах, использующих виртуальные диски.
- ▶ Динамические LPAR в Linux-разделах используют подсистему Hot-Plugging, и их поведение отличается от AIX 5L. Некоторые операции требуют перезагрузки системы с Linux, например изменение размеров памяти.

3.13.5. Что читать дальше

Ниже приведены источники дополнительной информации, которые могут помочь при конфигурировании и использовании функций Advanced POWER Virtualization с Linux VIO-клиентами и VIO-серверами:

- ▶ *Linux for pSeries installation and administration (SLES 9)*, by Chris Walden (Установка и администрирование Linux for pSeries (SLES 9), Крис Валден), по адресу: <http://www-128.ibm.com/developerworks/linux/library/l-pow-pinstall/>
- ▶ *Linux virtualization on POWER5: A hands-on setup guide*, by John Engel (Linux-виртуализация в POWER5: практическое руководство по настройке, Джон Энгел) <engel@us.ibm.com>, опубликовано IBM DeveloperWorks, по адресу: <http://www-128.ibm.com/developerworks/edu/l-dw-linux-pow-virutal.html>
- ▶ *POWER5 Virtualization: How to set up the SUSE Linux Virtual I/O Server*, by Nigel Griffiths, (POWER5 Virtualization: как настроить SUSE Linux Virtual I/O Server, Найджел Гриффитс) <nag@uk.ibm.com>, по адресу: <http://www-128.ibm.com/developerworks/@server/library/es-susevio/>



4

Установка Virtual I/O Server: базовые настройки

Эта глава знакомит с основами конфигурирования виртуальной среды в системе IBM System p5. Полностью рассмотрен процесс построения сервера ввода-вывода VIOS и приведен базовый сценарий его конфигурирования вместе с клиентскими разделами.

Основные темы главы:

- ▶ Начальные сведения
- ▶ Создание раздела VIOS
- ▶ Установка ПО Virtual I/O Server
- ▶ Базовый сценарий для VIOS
- ▶ Взаимодействие с клиентскими UNIX-разделами

4.1. Начальные сведения

В этом разделе приведена следующая информация об операционной среде сервера VIOS:

- ▶ Интерфейс командной строки VIOS, также называемый IOSCLI
- ▶ Управляемые аппаратные ресурсы
- ▶ Структура пакета ПО и поддержка

4.1.1. Интерфейс командной строки

VIOS обеспечивает ограниченный интерфейс командной строки (IOSCLI) с возможностью создания скриптов. Все конфигурации VIOS следует выполнять из этого интерфейса IOSCLI с помощью имеющейся ограниченной оболочки (Restricted Shell).

Важно. В среде shell-оболочки oem_setup_env не следует выполнять какое-либо конфигурирование группы томов и создание логических томов.

Через интерфейс командной строки выполняются следующие задачи администрирования VIOS:

- ▶ Управление устройствами (физическими, виртуальными и LVM)
- ▶ Конфигурирование сети
- ▶ Установка и обновление ПО
- ▶ Обеспечение безопасности
- ▶ Управление пользователями
- ▶ Установка программного обеспечения OEM
- ▶ Задачи обслуживания

Для первоначального входа в систему VIOS используйте идентификатор пользователя `padmin`, являющегося главным администратором. После входа в систему необходимо сменить пароль. Пароля по умолчанию не существует¹.

При входе в VIOS вы попадаете в ограниченную оболочку Korn Shell. Ограниченный Korn Shell работает так же, как и стандартный Korn Shell, но с некоторыми ограничениями. Эти ограничения не позволяют пользователю делать следующее:

- ▶ Изменять текущий рабочий каталог.
- ▶ Задавать значения переменных SHELL, ENV или PATH.
- ▶ Определять имя пути команды, содержащей перенаправленный вывод команды с признаками `>`, `>|`, `<>` или `>`.

Из-за этих ограничений у вас нет возможности запускать команды, недоступные по вашей переменной PATH. Эти ограничения не дают вам непосредственно отправлять вывод команды в файл и требуют вместо этого перенаправлять через конвейер вывод в команду tee.

После того, как вы вошли в систему, вы можете ввести команду `help` для получения обзора поддерживаемых команд, приведенного в примере 4-1.

¹ При первом входе в систему VIOS пароль не спрашивают, а принуждают установить (новый) пароль сразу после ввода login-имени `padmin`. Прим. науч. ред.

Пример 4-1. Команды, поддерживаемые в Virtual I/O Server Version 1.2

```
$ help
Install Commands (Команды установки)           Security Commands (Команды управления
                                                    безопасностью)
  ioslevel                                         lsfailedlogin
  license                                          lsgcl
  lssw
  oem_platform_level
  oem_setup_env
  remote_management
  updateios

LAN Commands (Команды управления сетью)          UserID Commands (Команды управления
                                                    пользователями)
  cfglnagg
  cfgnamesrv
  entstat
  hostmap
  hostname
  lsnetsvc
  lstcpip
  mktcpip
  netstat
  optimizenet
  ping
  rmtcpip
  startnetsvc
  stopnetsvc
  traceroute

Device Commands (Команды управления уст-       Maintenance Commands (Команды обслуживания)
ройствами)                                     backupios
  chdev
  chpath
  cfgdev
  lsdev
  lsmap
  lspath
  mkpath
  mkvdev
  mkvt
  rmdev
  rmpath
  rmvdev

Physical Volume Commands (Команды управле-     Shell Commands (Команды Shell)
ния физическими томами)
  lspv
                                                    mount
                                                    pdump
                                                    restorevgstruct
                                                    savevgstruct
                                                    showmount
                                                    shutdown
                                                    snap
                                                    startsysdump
                                                    startrace
                                                    stoptrace
                                                    sysstat
                                                    topas
                                                    umount
                                                    viostat
```

migratepv	awk
	cat
Logical Volume Commands (Команды управления логическими томами)	chmod
chlv	clear
cplv	cp
extendlv	date
lslv	ftp
mklv	grep
mklvcopy	head
rmlv	ls
rmlvcopy	man
	mkdir
Volume Group Commands (Команды управления группами томов)	more
activatevg	mv
chvg	rm
deactivatevg	sed
exportvg	stty
extendvg	tail
importvg	tee
lsvg	vi
mirrorios	wall
mkvg	wc
reducevg	who
syncvg	
unmirrorios	
Storage Pool Commands (Команды управления путем хранения)	
chsp	
lssp	
mkbdsp	
mksp	
mbdsp	

Для получения более полной справки по этим командам используйте команду **help**, как показано в примере 4-2.

Пример 4-2. Команда help

```
$ help errlog
Usage: errlog [-ls | -rm Days]

Displays or clears the error log.

-ls Displays information about errors in the error log file in a detailed format.
-rm Deletes all entries from the error log older than the number of days specified by
the Days parameter.
```

Интерфейс командной строки VIOS поддерживает два режима выполнения:

- Традиционный режим
- Интерактивный режим

Традиционный режим предназначен для исполнения одной команды. В этом режиме вы за один раз запускаете одну команду из строки приглашения Shell. Например, для получения списка всех виртуальных устройств введите следующее:

```
#ioscli lsdev -virtual
```

Для упрощения ввода по сравнению с традиционным Shell для каждой подкоманды был создан свой псевдоним (alias). При наличии набора псевдонимов вам не нужно вводить команду `ioscli`. Например, для получения списка всех устройств с типом `adapter` вы можете ввести следующее:

```
#lsdev -type adapter
```

В интерактивном режиме пользователь будет представлен приглашением команды `ioscli`, если исполнит команду `ioscli` без подкоманд и аргументов. После этого команды `ioscli` будут запускаться одна за другой без повторного ввода `ioscli`. Например, для перехода в интерактивный режим введите:

```
#ioscli
```

В интерактивном режиме для получения списка всех виртуальных устройств введите:

```
#lsdev -virtual
```

Такие внешние команды, как `grep` или `sed`, не могут запускаться из строки приглашения в интерактивном режиме. Сначала вы должны выйти из интерактивного режима, введя `quit` или `exit`.

4.1.2. Управляемые аппаратные ресурсы

Функция Advanced POWER Virtualization, активирующая микроразделы на сервере, обеспечивает средство установки VIOS. Также необходим логический раздел с ресурсами, достаточными для совместного использования с другими разделами. Для создания VIOS должен быть доступен следующий минимальный объем аппаратных ресурсов:

Сервер POWER5	Машина с функцией виртуального ввода-вывода.
Аппаратная консоль (Hardware Management Console)	Для создания раздела и назначения ресурсов необходима НМС или должен быть установлен менеджер IVM либо в предопределенном разделе, либо в предустановленном.
Адаптер дисковой памяти	Серверному разделу необходим хотя бы один адаптер дисковой памяти.
Физический диск	Если вы хотите сделать ваш диск общим для клиентских разделов, то вам необходим диск с объемом, позволяющим создавать на нем логические тома достаточно большого размера.
Ethernet-адаптер	Этот адаптер необходим, если вы хотите обеспечивать безопасную маршрутизацию сетевого трафика из виртуального Ethernet в реальный сетевой адаптер.

Оперативная память Необходимо не менее 512 МБ памяти. Так же как и для операционной системы, сложность подсистемы ввода-вывода и количество виртуальных устройств диктуют необходимый объем памяти, например, в случае использования множественного доступа к устройствам хранения SAN.

Virtual I/O Server V1.2 предназначен для работы в определенных конфигурациях с системами хранения данных IBM и других производителей.

Обратитесь к вашему представителю компании IBM или ее бизнес-партнеру за последней информацией и имеющимися конфигурациями.

Виртуальные устройства, экспортируемые в клиентские разделы с помощью VIOS, должны присоединяться через один из следующих физических адаптеров:

- ▶ PCI 4-Channel Ultra3 SCSI RAID Adapter (FC 2498)
- ▶ PCI-X Dual Channel Ultra320 SCSI RAID Adapter (FC 5703)
- ▶ Dual Channel SCSI RAID Enablement Card (FC 5709)
- ▶ PCI-X Dual Channel Ultra320 SCSI Adapter (FC 5712)
- ▶ 2 Gigabit Fibre Channel PCI-X Adapter (FC 5716)
- ▶ 2 Gigabit Fibre Channel Adapter for 64-bit PCI Bus (FC 6228)
- ▶ 2 Gigabit Fibre Channel PCI-X Adapter (FC 6239)

Рекомендуется проводить тщательное планирование перед тем, как вы начнете конфигурирование и установку своего VIOS и клиентских разделов. В зависимости от типа нагрузки и потребностей приложения рассмотрите возможность смешивания виртуальных и физических устройств. Например, если ваше приложение выигрывает от быстрого дискового доступа, то предусмотрите выделение этому разделу физического адаптера.

4.1.3. Структура пакета ПО и поддержка

Установка раздела с Virtual I/O Server выполняется со специального компакт-диска формата `mksysb`, поставляемого клиентам, заказавшим функцию Advanced POWER Virtualization, за дополнительную плату. Для моделей p5-590 и p5-595 эта функция уже включена в комплект поставки. ПО Virtual I/O Server поддерживается только в разделах Virtual I/O Server.

Установка Virtual I/O Server V1.2 с DVD-носителя может быть выполнена следующими способами:

- ▶ Непосредственно с носителя (назначением дисковода DVD-ROM разделу и загрузкой с данного носителя).
- ▶ С помощью HMC (вставкой носителя в дисковод DVD-ROM на HMC и командой `installios`).
- ▶ С помощью DVD-носителя, сервера сетевой инсталляции (NIM) и команды `smitty installios` (для работы между NIM и HMC требуется доступ через `secure shell`).

Важно

- ▶ В случае использования IVM не нужно назначать DVD-ROM разделу, так как VIOS устанавливается в предопределенный раздел.
- ▶ Команда `installios` неприменима для систем, управляемых с помощью IVM.

Более подробно об установке Virtual I/O Server рассказывается в разделе 4.3 «Установка ПО Virtual I/O Server».

4.2. Создание раздела с Virtual I/O Server

В этом разделе дается пошаговая процедура создания логического раздела с виртуальным сервером ввода-вывода и установки программного обеспечения VIOS.

4.2.1. Определение раздела с Virtual I/O Server

Здесь показано, как создать логический раздел, в который вы будете устанавливать виртуальный сервер ввода-вывода с именем VIO_Server1 на HMC.

Если одним из главных факторов является производительность, мы рекомендуем назначить VIOS выделенный процессор. В нашем тестовом случае у нас будет общий процессор для нашего раздела с VIOS. Для создания виртуального сервера ввода-вывода выполните следующие шаги:

1. На рисунке 4-1 показана HMC с четырьмя подключенными управляемыми системами. Для нашей настройки базовой VIOS-конфигурации мы будем использовать управляемую систему с именем P520_ITSO.
Создание нашего первого виртуального сервера ввода-вывода будет проходить через последовательность следующих окон:
2. Щелкните правой кнопкой мыши на управляемой системе P520_ITSO и выберите *Create Logical Partition* (Создать Логический раздел), как показано на рисунке 4-2, чтобы запустить мастер Create Logical Partition Wizard (Мастер создания логических разделов).
3. Введите имя и идентификатор раздела и выберите кнопку *Virtual I/O Server*, как показано на рисунке 4-3.
4. Пропустите определение группы управления нагрузкой (workload management group), выбрав кнопку *Нет*, и щелкните по *Next* (далее) (рисунок 4-4).
5. У вас есть возможность изменить имя профиля (Profile name), как показано на рисунке 4-5. Если это не нужно, то вы можете оставить его значение по умолчанию. В нашем случае мы зададим имя профиля одинаковым с именем нашего раздела.

Примечание. Если отмечено поле флагка *Use all resources in the system* (Использовать все ресурсы в системе), то определяемый логический раздел получит все ресурсы в управляемой системе.

6. Выберите настройки памяти, как показано на рисунке 4-6.

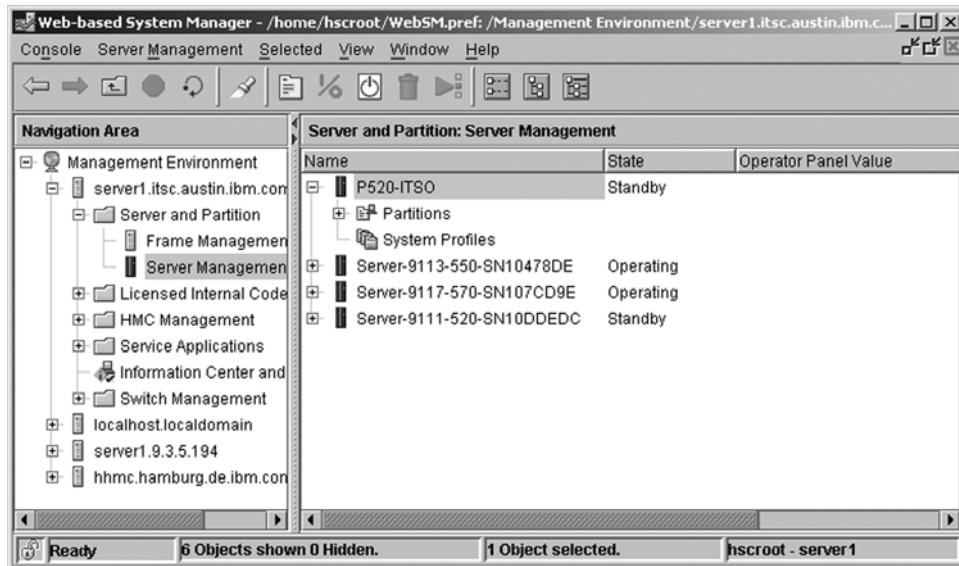


Рис. 4-1. Вид аппаратной консоли (Hardware Management Console)

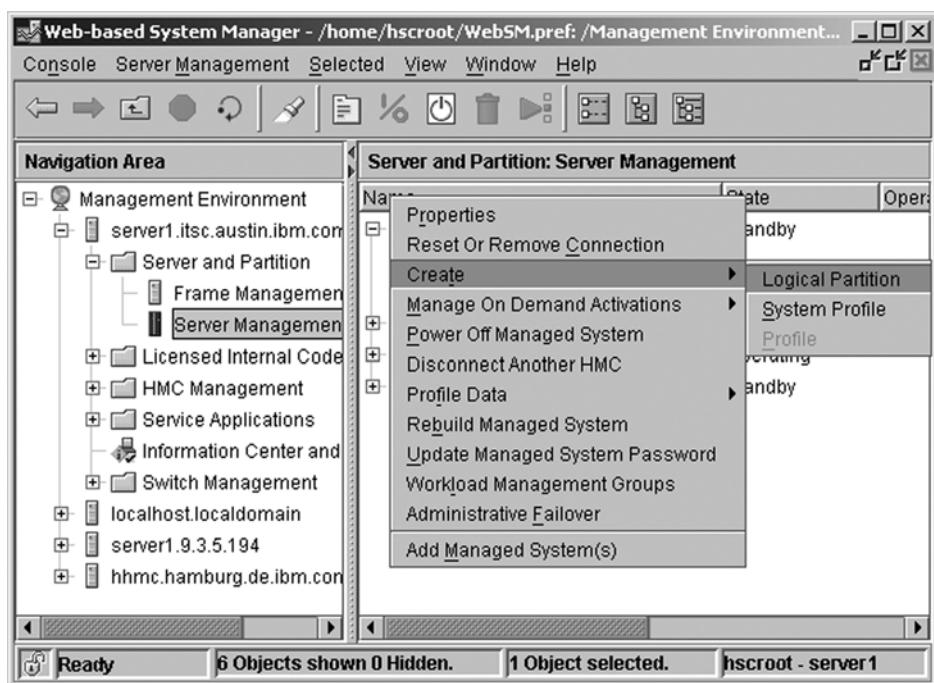


Рис. 4-2. Запуск мастера Create Logical Partition Wizard

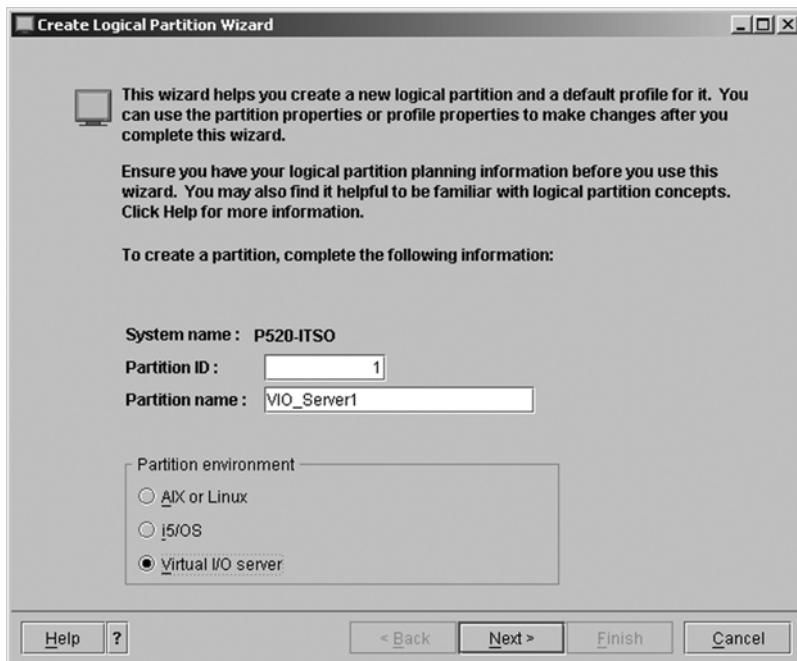


Рис. 4-3. Определение идентификатора и имени раздела

Ограничение. К рисунку 4-6 применяются следующие ограничения:

- ▶ Если управляемая система не способна обеспечить минимальный объем памяти, то раздел не будет запускаться.
- ▶ Вы не сможете динамически увеличивать объем памяти в разделе сверх заданного максимума. Если вы хотите иметь память сверх этого максимума, то раздел нужно деактивировать, обновить профиль и затем повторно активировать раздел.

Важно. Минимально рекомендуемый объем памяти для VIOS равен 512 МБ. Так же как и для операционной системы, сложность подсистемы ввода-вывода и количество виртуальных устройств диктуют на необходимый объем памяти, например, в случае использования множественного доступа к устройствам хранения SAN.

7. Выберите флагок **Shared (Общий)** для способа выделения процессора, как показано на рисунке 4-7.

Примечание. При высоких нагрузках на VIOS мы рекомендуем использовать в разделе выделенный процессор. Мы используем общий процессор из-за ограниченного количества физических процессоров в нашей управляемой системе и стремления к простоте настройки в данном случае.

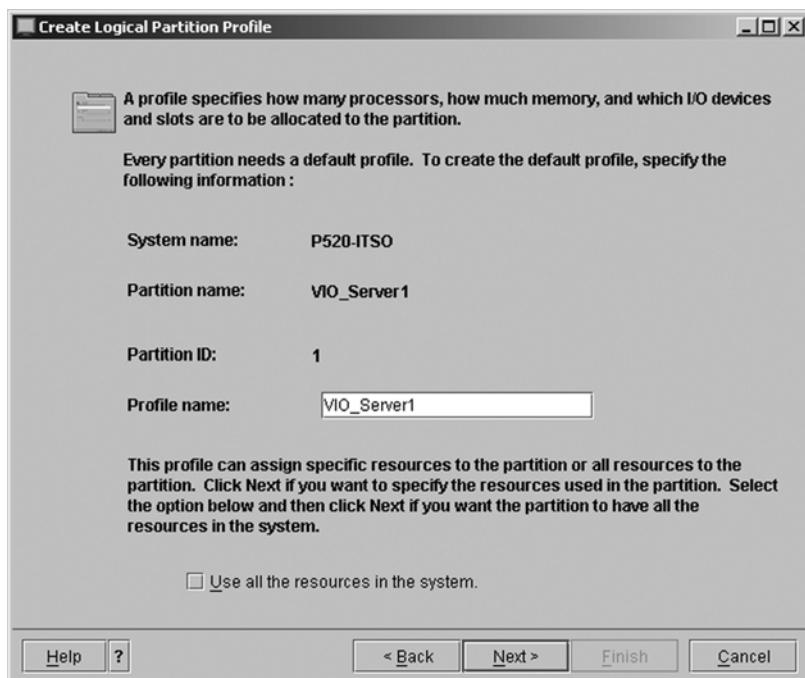


Рис. 4-4. Пропуск группы управления нагрузкой

8. Выберите настройки общего процессора и задайте процессорные единицы, как показано на рисунке 4-8.

Ограничение. К настройкам процессора применяются следующие ограничения:

- Раздел не запустится, если управляемая система не может обеспечить минимальное количество процессорных единиц.
- Вы не сможете динамически увеличивать количество процессорных единиц сверх заданного максимума. Если вы хотите увеличить количество процессорных единиц, то раздел нужно деактивировать, обновить профиль и затем повторно активировать раздел.

Примечание. В последней версии Web-based System Manager поле Desired processing units (Желаемые процессорные единицы) отображается ниже поля Minimum processing units (Минимальные процессорные единицы). В предыдущих версиях они расположены в обратном порядке.

9. Задайте режим общего использования процессора (processing sharing mode) и настройки виртуальных процессоров, как показано на рисунке 4-9, войдя в это окно щелчком по кнопке Advanced (Дополнительно). Щелкните OK, когда вы завершите эти настройки.

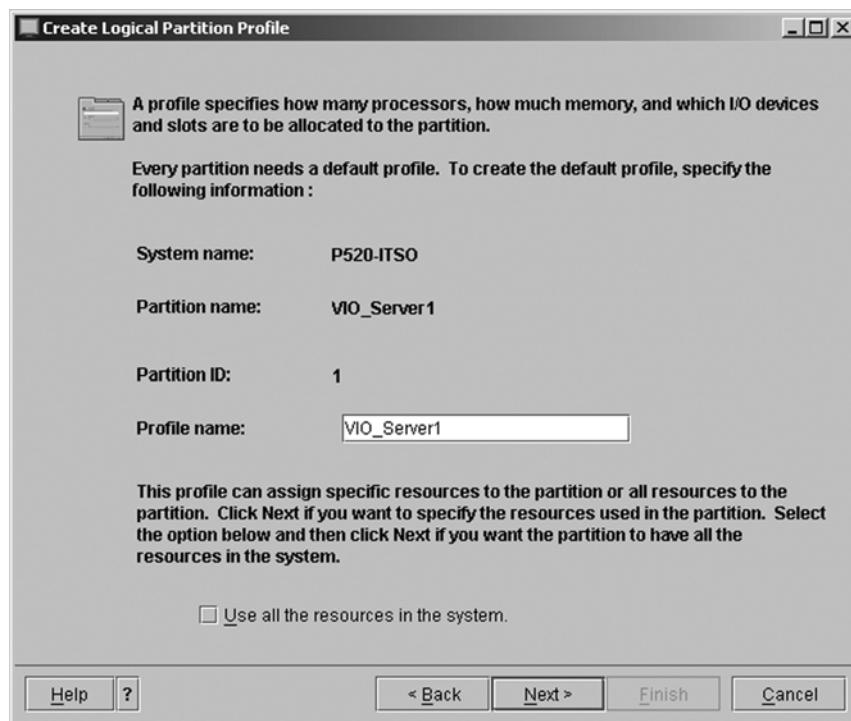


Рис. 4-5. Определение имени профиля раздела

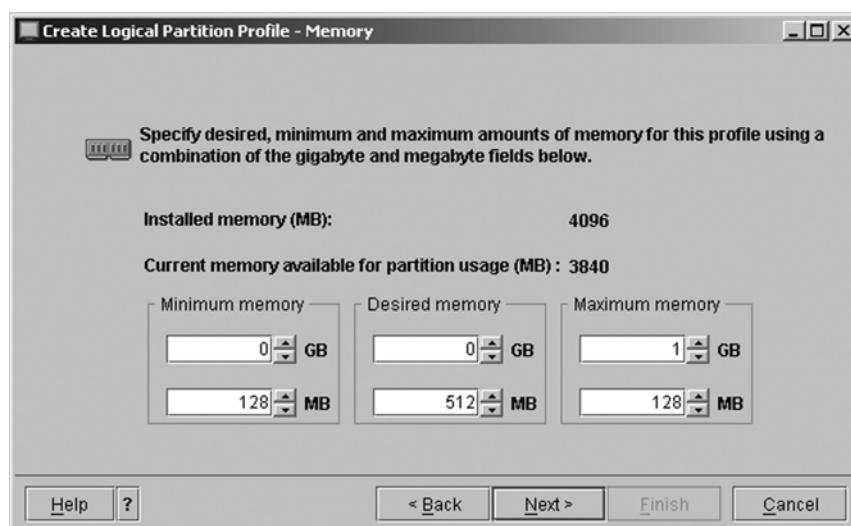


Рис. 4-6. Настройки памяти разделов

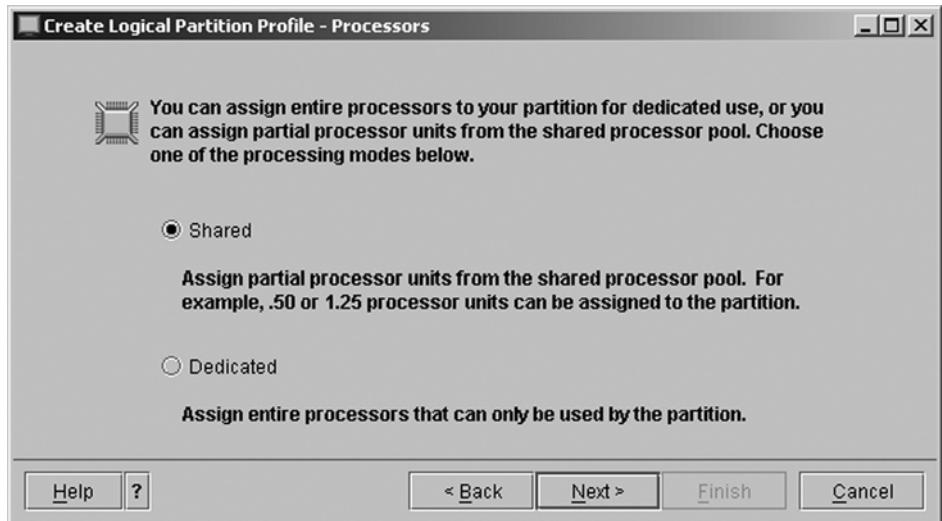


Рис. 4-7. Использование общего процессора

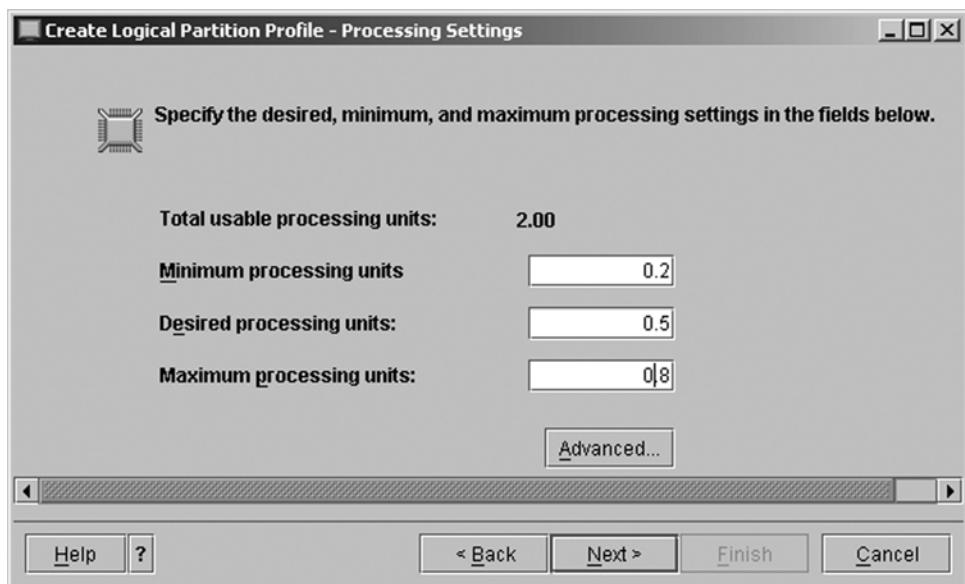


Рис. 4-8. Настройки общего процессора

Примечание. См. раздел 3.3 «Введение в микроразделы» для получения более подробной информации о процессорных единицах, режимах с верхним пределом (capped) и без верхнего предела (uncapped) и виртуальных процессорах.

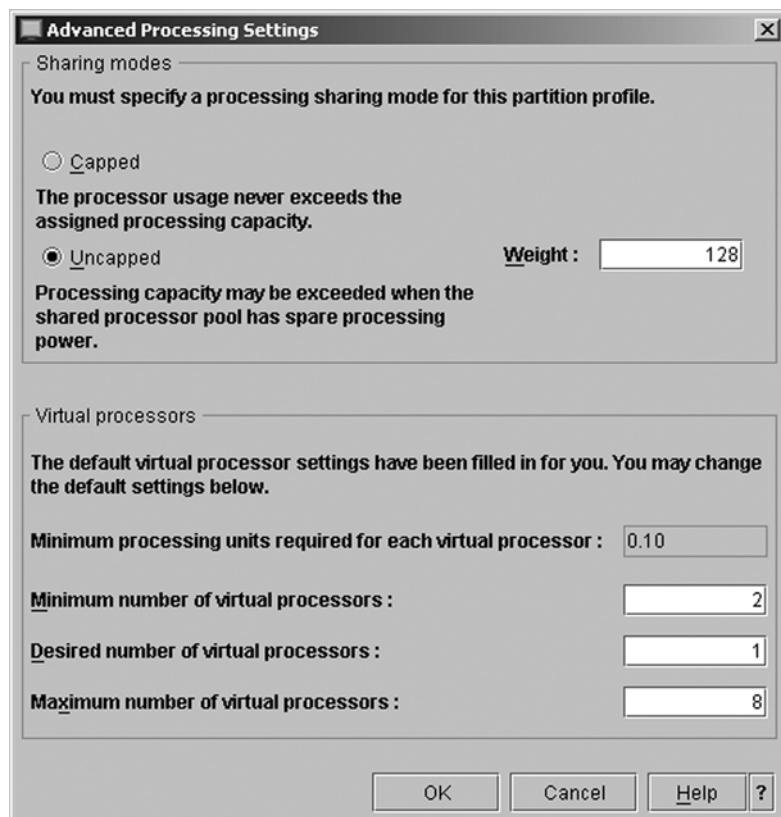


Рис. 4-9. Режим общего процессора и настройки виртуальных процессоров

10. Выберите физические ресурсы, которые вы хотите выделить вашему виртуальному серверу ввода-вывода. Для настройки нашей базовой конфигурации нам будет нужен контроллер хранилища с локальными дисками, один CD-привод и Ethernet-адаптер. На рисунке 4-10 показан вариант выбора для нашей базовой настройки. Щелкните по **Next**, когда закончите делать выбор.
11. Пропустите настройку пулов ввода-вывода (I/O pools), предлагаемую на рисунке 4-11, щелкнув по **Next**.
12. Пропустите определение виртуальных адаптеров ввода-вывода, выбрав кнопку **No** (см. рис. 4-12).

Примечание. Так как запуск виртуальных адаптеров связан с проведением тщательного планирования, то мы рекомендуем в данный момент воздержаться от создания виртуальных адаптеров.

13. Пропустите настройки разделов управления питанием, показанные на рисунке 4-13, щелкнув по **Next**.

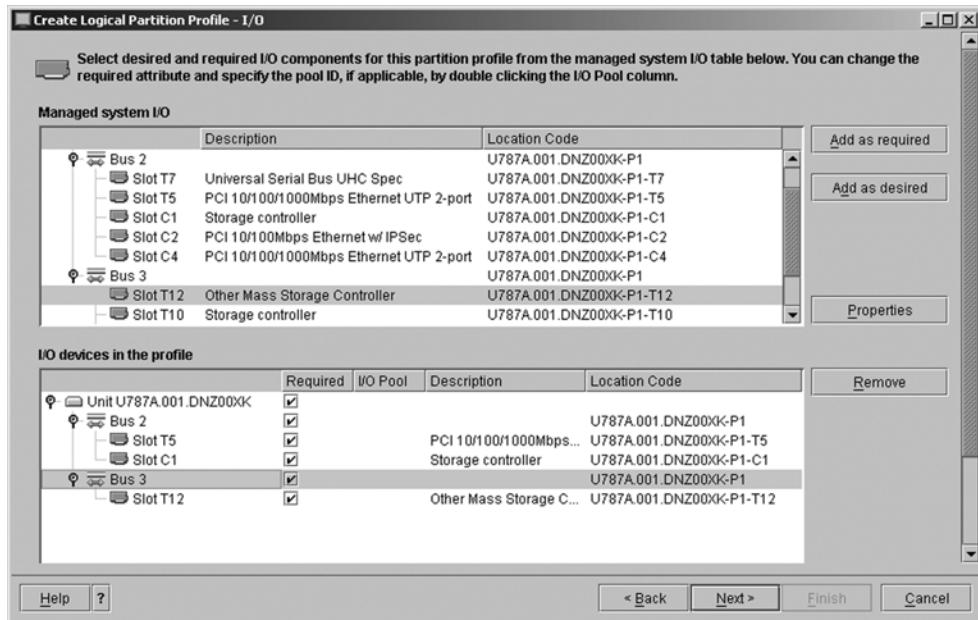


Рис. 4-10. Выбор физических компонентов ввода-вывода

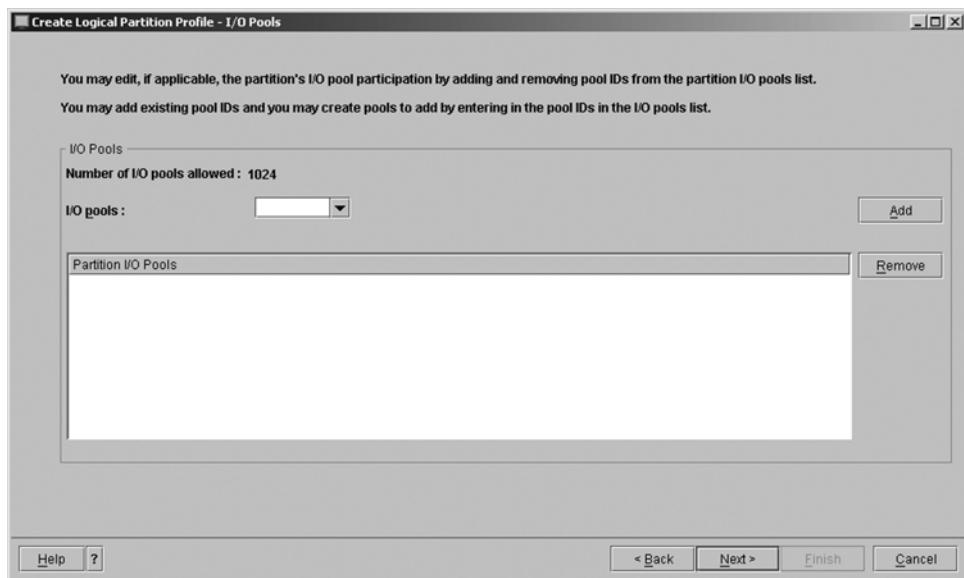


Рис. 4-11. Настройки пула ввода-вывода

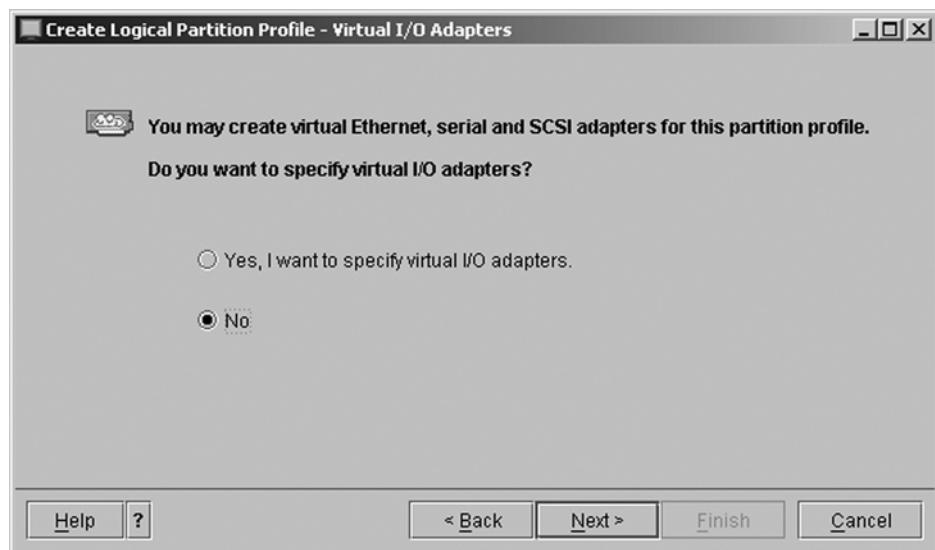


Рис. 4-12. Пропуск определения виртуальных адаптеров ввода-вывода

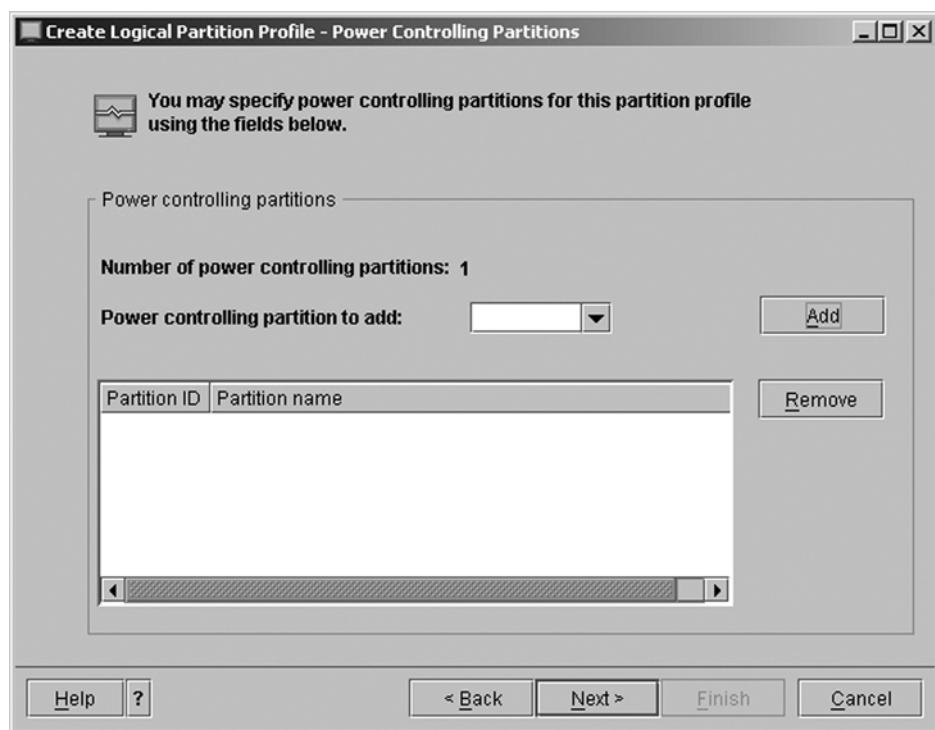


Рис. 4-13. Пропуск настроек для разделов управления питанием

14. Выберите **Normal (Обычный)** для настройки вашего режима загрузки, как показано на рисунке 4-14.

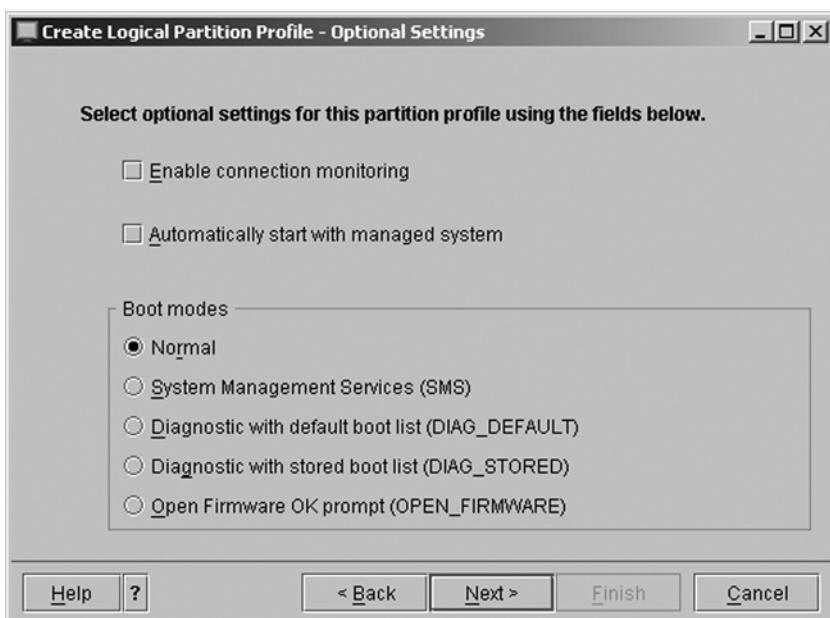


Рис. 4-14. Выбор настройки режима загрузки

15. Внимательно проверьте ранее выбранные вами настройки для раздела, показанные на рисунке 4-15, и затем запустите мастер создания разделов, щелкнув по **Finish** (Готово).

При выполнении создания раздела появится рабочее окно (см. рис. 4-16).

Через несколько секунд после выполнения обработки в окне состояния у вас будет возможность увидеть раздел, определенный в основной папке Server Management под вкладкой Partitions (см. рис. 4-17).

4.3. Установка ПО Virtual I/O Server

В этом разделе описывается реальная установка программного обеспечения Virtual I/O Server V1.2 на ранее созданный раздел виртуального ввода-вывода под именем VIO_Server1. Существуют три поддерживаемых способа установки ПО Virtual I/O Server V1.2:

- ▶ С помощью загрузки с дисковода CD/DVD, выделенного разделу VIOS
- ▶ Установка ПО VIOS с консоли HMC
- ▶ Установка ПО с помощью NIM и HMC

Ниже приведены шаги процедуры установки с помощью установочного CD/DVD-устройства:

- 1 . Поместите DVD-диск Virtual I/O Server V1.2 в CD/DVD-привод.

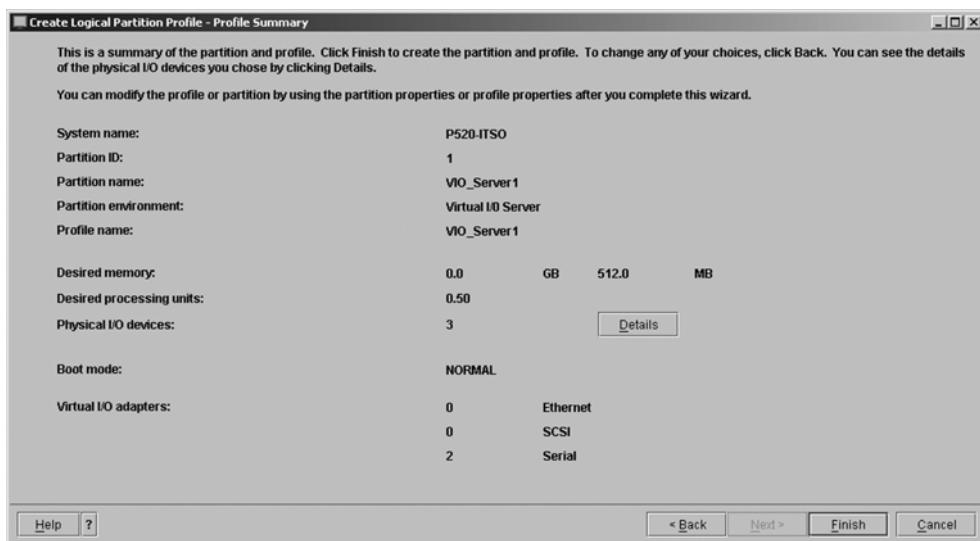


Рис. 4-15. Общий вид настроек раздела

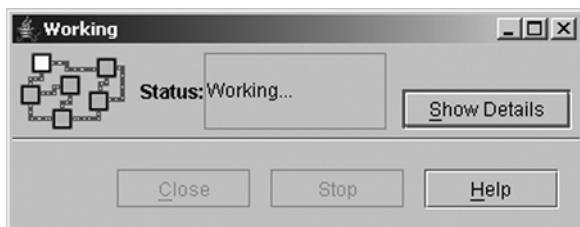


Рис. 4-16. Окно состояния

2. Активируйте раздел VIO_Server1, щелкнув правой кнопкой мыши по имени раздела и выбрав кнопку **Activate (Активировать)**, как показано на рисунке 4-18. Выберите использованный вами при создании этого раздела профиль по умолчанию.
3. Выберите профиль VIO_Server1, установите флажок *Open a terminal window or console session* (Открыть окно терминала или сессию консоли), как показано на рисунке 4-19, и щелкните по кнопке **Advanced**.
4. В выпадающем списке **Boot Mode** (Режим загрузки) выберите **SMS (System Management Services)**, как показано на рисунке 4-20, и щелкните **OK**.
5. На рисунке 4-21 показано меню SMS сервера IBM @server pSeries после загрузки раздела в режиме SMS.
6. Выполните следующие шаги для продолжения процедуры и загрузки раздела **Virtual I/O Server**:
 - a. Выберите 5 для **Select Boot Options** (Выбор опций загрузки) и нажмите **Enter**.
 - b. Выберите 1 для **Select Install/Boot Device** (Выбор устройства установки/загрузки) и нажмите **Enter**.

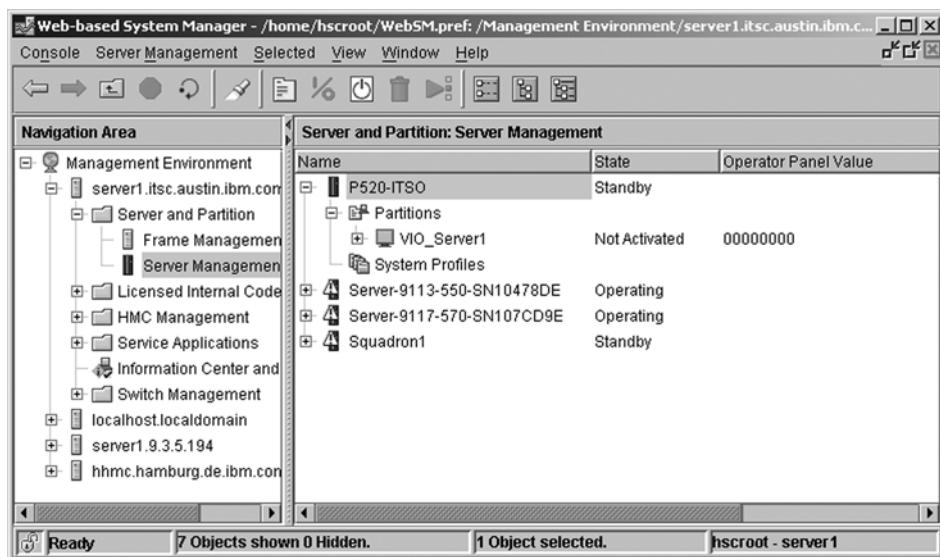


Рис. 4-17. Теперь в окне показан новый созданный раздел VIO_Server1

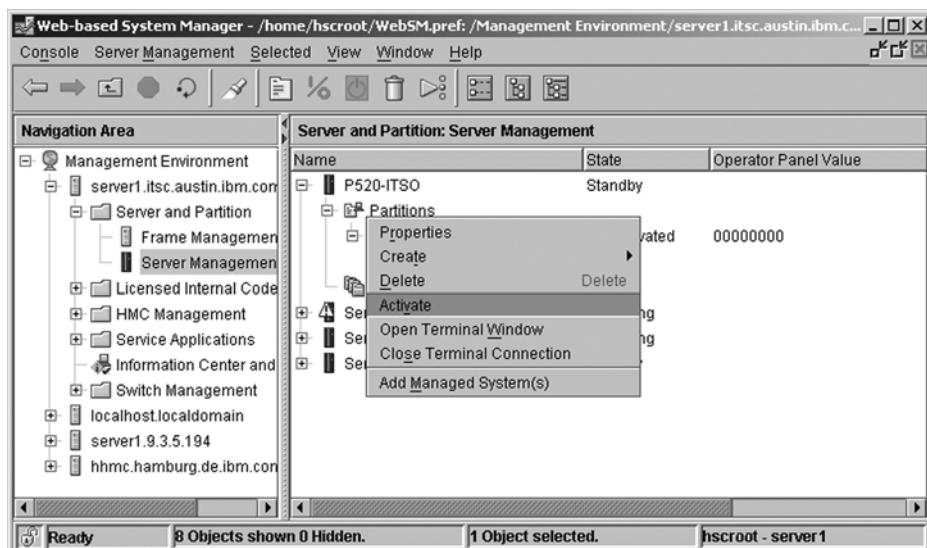


Рис. 4-18. Активация раздела VIO_Server1

- c. Выберите 3 для CD/DVD и нажмите Enter.
- d. Выберите 4 для IDE и нажмите Enter.
- e. Выберите 1 для IDE CD-ROM и нажмите Enter.
- f. Выберите 2 для Normal Mode Boot (Загрузка в обычном режиме) и нажмите Enter.
- g. Подтвердите ваш выбор, установив 1 для Yes, и нажмите Enter.

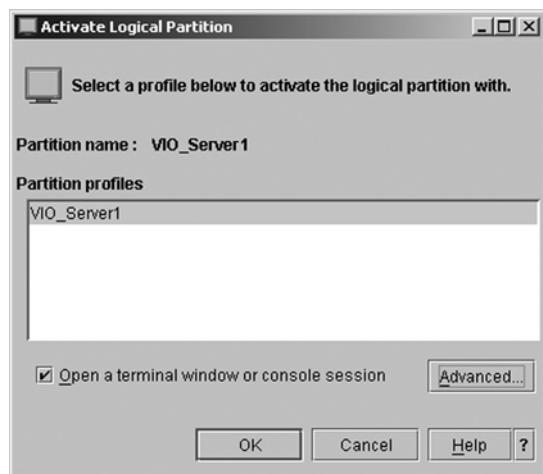


Рис. 4-19. Выбор профиля

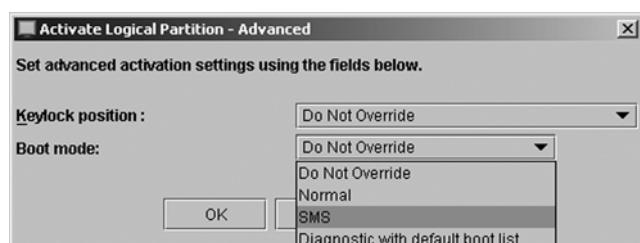


Рис. 4-20. Выбор режима загрузки SMS

7. По окончании процедуры установки используйте для входа в систему имя пользователя `padmin`. При первоначальном входе в систему вас попросят ввести пароль.

После успешного входа в систему вы будете помещены в интерфейс командной строки (CLI) сервера VIOS. Введите следующую команду для принятия лицензии:

```
$ license -accept
```

Теперь вы готовы использовать только что установленное программное обеспечение VIOS.

4.4. Базовый сценарий для VIOS

В этом разделе приведена простая конфигурация, состоящая из одного раздела VIOS, обслуживающего виртуальные SCSI-устройства для четырех логических разделов. На рисунке 4-22 вы можете увидеть сценарий создания этой базовой конфигурации.

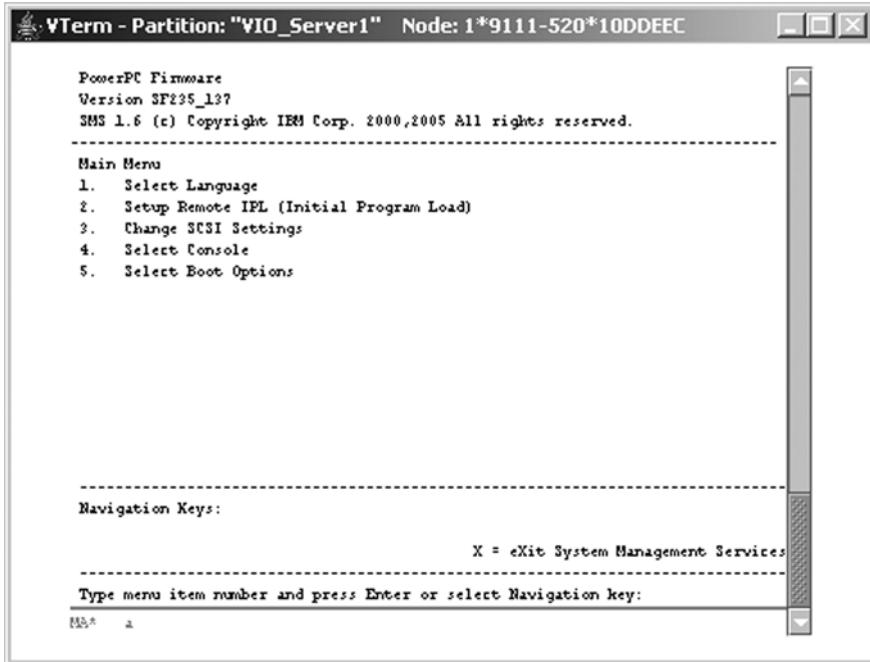


Рис. 4-21. Меню SMS

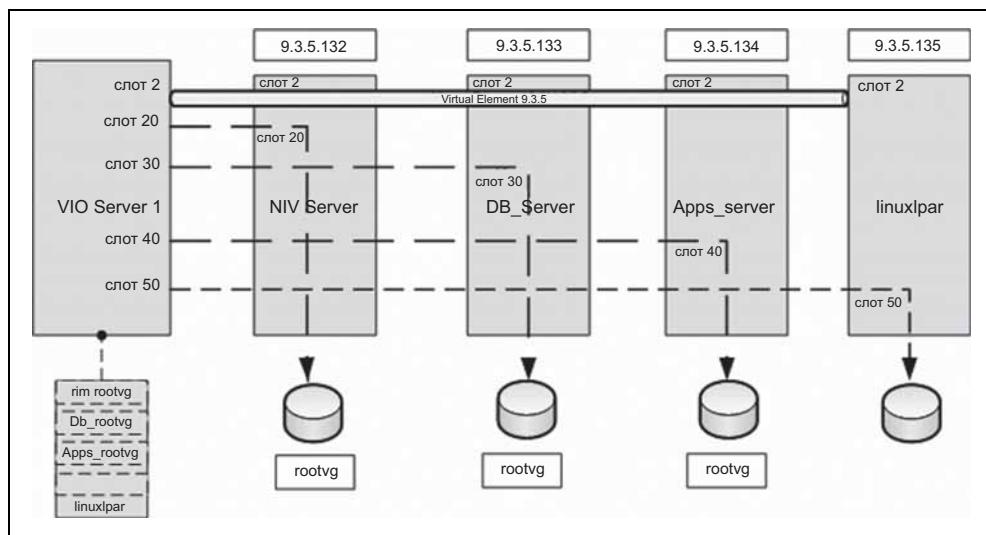


Рис. 4-22. Сценарий создания базового VIOS

На основании рисунка 4-22 в следующих разделах будет описана последовательность конфигурирования наших виртуальных Ethernet- и SCSI-адаптеров.

4.4.1. Создание виртуального Ethernet-адаптера для VIOS

Виртуальный Ethernet-адаптер является логическим адаптером, эмулирующим функцию физического адаптера ввода-вывода в логическом разделе. Виртуальные Ethernet-адAPTERы обеспечивают обмен с другими логическими разделами внутри управляемой системы без использования реального оборудования и кабельной проводки.

Для создания такого адаптера выполните следующие шаги:

1. Щелкните правой кнопкой мыши по профилю раздела VIO_Server1 и выберите Properties (Свойства) (см. рис. 4-23).

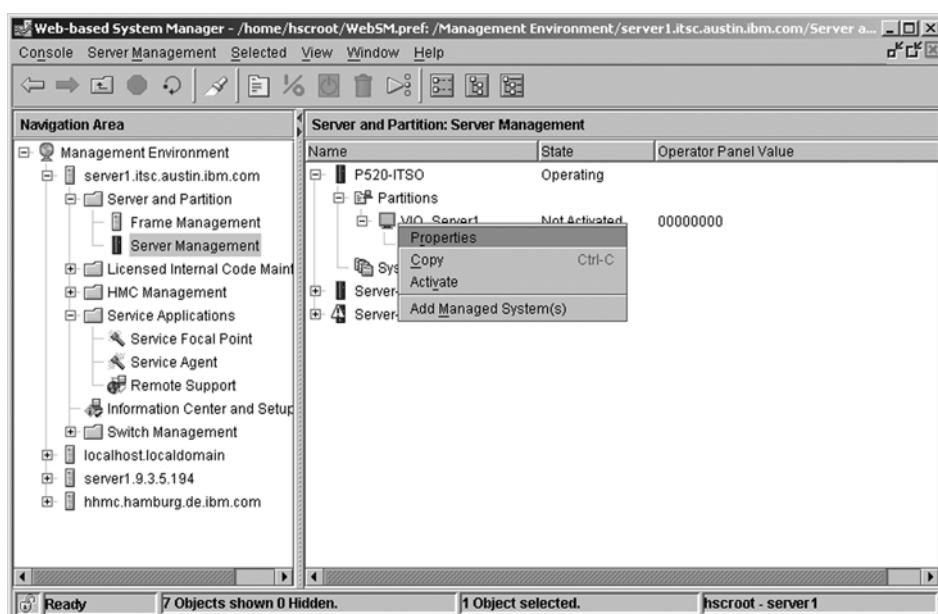


Рис. 4-23. Добавление виртуального Ethernet

2. Щелкните вкладку **Virtual I/O Adapters** (Виртуальные адAPTERы ввода-вывода) и затем щелкните вкладку **Ethernet**, как показано на рисунке 4-24.
3. Щелкните по кнопке **Create adapter** (Создать адаптер) и введите настройки. В свойствах виртуального Ethernet-адаптера выберите номер слота для виртуального адаптера и идентификатор виртуальной сети (Virtual LAN ID), а затем установите флажок **Access External network** (Доступ к внешней сети), чтобы использовать этот адаптер как шлюз между сетями VLAN и внешней сетью. Этот виртуальный Ethernet будет сконфигурирован как общий Ethernet-адаптер (см. рис. 4-25).
4. Вы можете установить флажок **IEEE 802.1Q compatible adapter** (Совместимый с IEEE 802.1Q адаптер), если вы хотите добавить дополнительные идентификаторы сетей VLAN. В нашем случае мы этого делать не будем, так как конфигурация должна оставаться базовой.

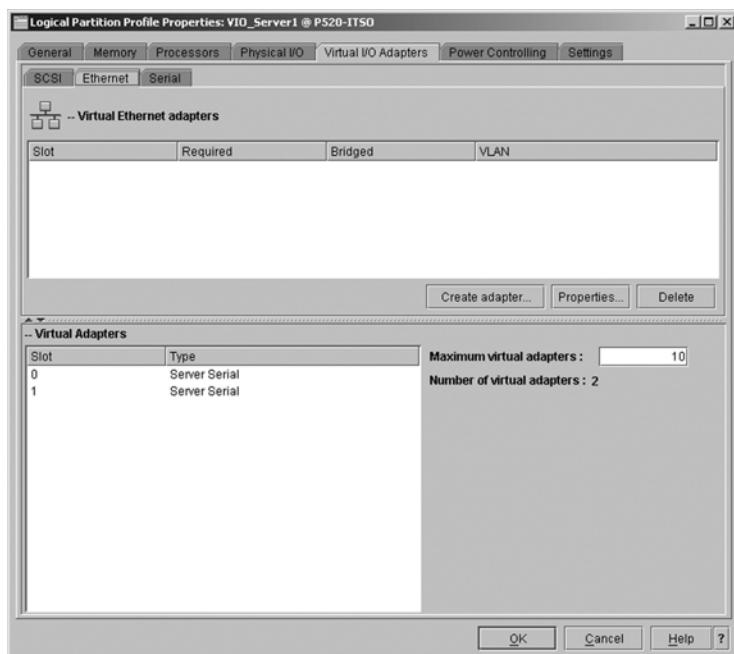


Рис. 4-24. Вкладка виртуального Ethernet

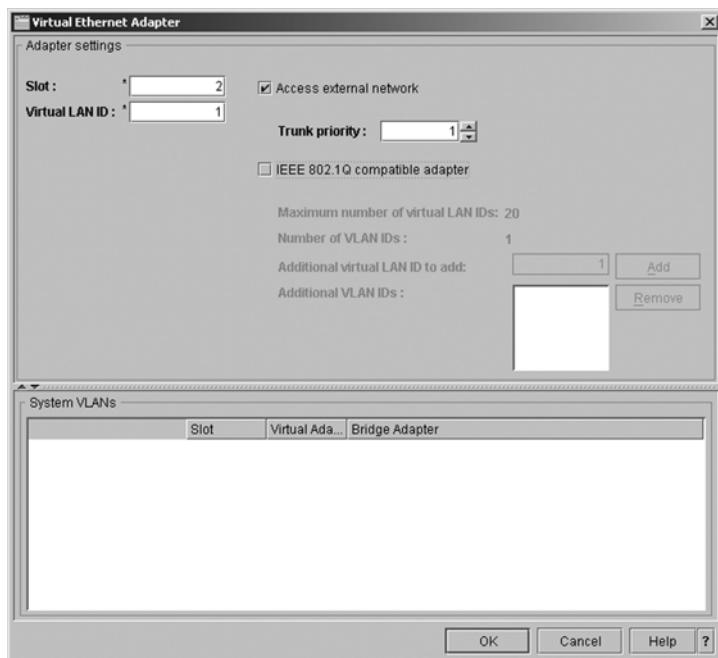


Рис. 4-25. Свойства виртуального Ethernet

Примечание. Установка флагка **Access External Networks** имеет смысл только для раздела VIOS. Не устанавливайте этот флагок при конфигурировании виртуальных Ethernet-адаптеров клиентских разделов. Не создавайте больше одного Ethernet-адаптера с флагком Access External Networks в одной VLAN.

5. Щелкните OK – и виртуальный Ethernet-адаптер готов для конфигурирования из интерфейса командной строки (CLI) VIOS. (См. рис. 4-26).

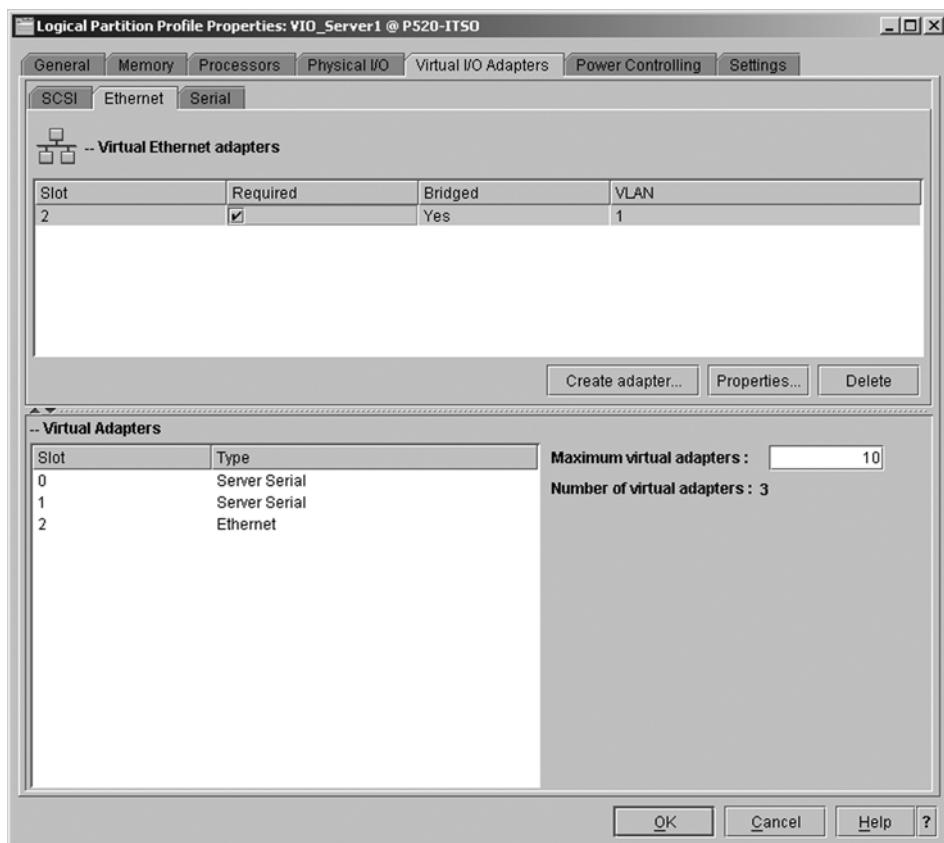


Рис. 4-26. Вкладка нового виртуального Ethernet-адаптера

4.4.2. Создание серверных виртуальных SCSI-адаптеров

В этом разделе рассказывается, как определить серверные виртуальные SCSI-адAPTERы для использования клиентскими разделами (NIM_server, DB_server, Apps_server и linuxlpar). Выполните следующие шаги для создания серверных виртуальных SCSI-адаптеров:

1. Щелкните правой кнопкой мыши по профилю раздела VIO_Server1 и выберите пункт **Properties** (см. рис. 4-23).
2. Щелкните по вкладке **Virtual I/O Adapters** и затем щелкните по вкладке **SCSI**, как показано на рисунке 4-27.

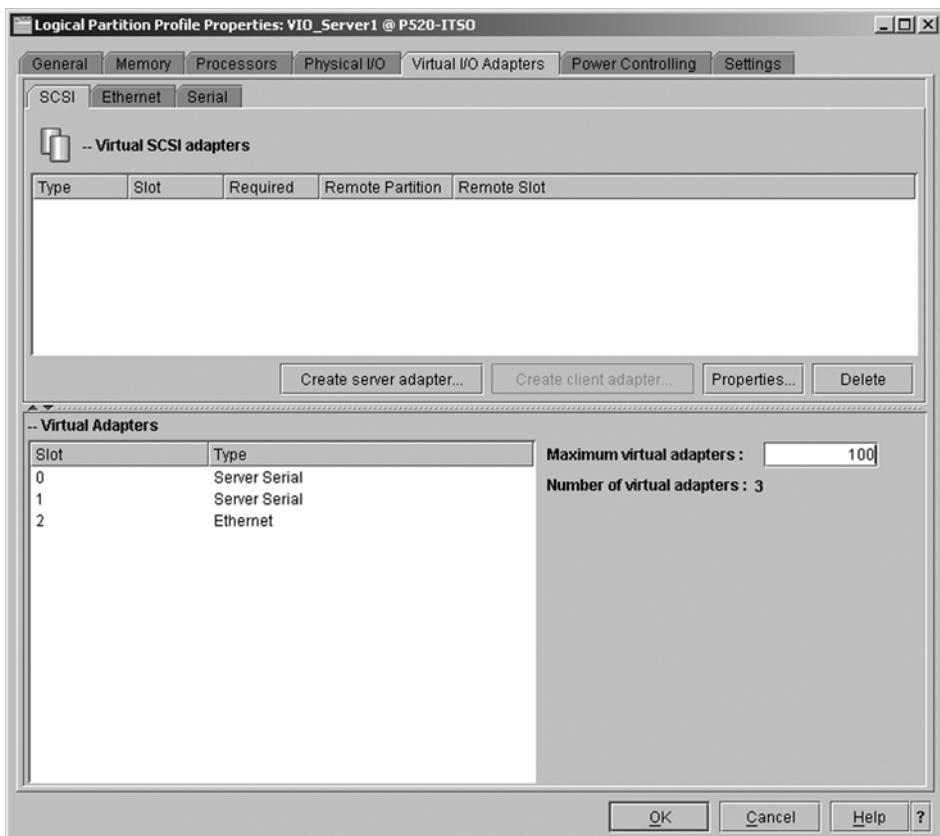


Рис. 4-27. Вкладка свойств SCSI

3. Задайте параметр **Maximum virtual adapters** (Максимальное количество виртуальных адаптеров) для адаптеров, которые будут поддерживаться VIOS, и щелкните по **Create server adapter** (Создать серверный адаптер). Выпадающий список под **Client partition** (Клиентский раздел) позволяет вам выбрать, какой раздел может использовать этот слот, как показано на рисунке 4-28. После этого щелкните **OK**.

Примечание. По нашему опыту, лучше будет, если номер слота сервера будет совпадать с номером слота клиента. Вы сэкономите много времени, когда будете устанавливать позднее связи между слотами.

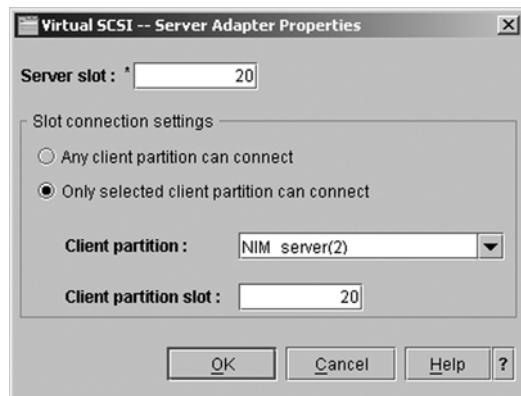


Рис. 4-28. Свойства слота серверного адаптера

Начиная с HMC Version 5 Release 1, теперь показываются связи между слотами, что делает этот процесс более управляемым для администраторов при работе с большим количеством виртуальных адаптеров (см. рис. 4-29).

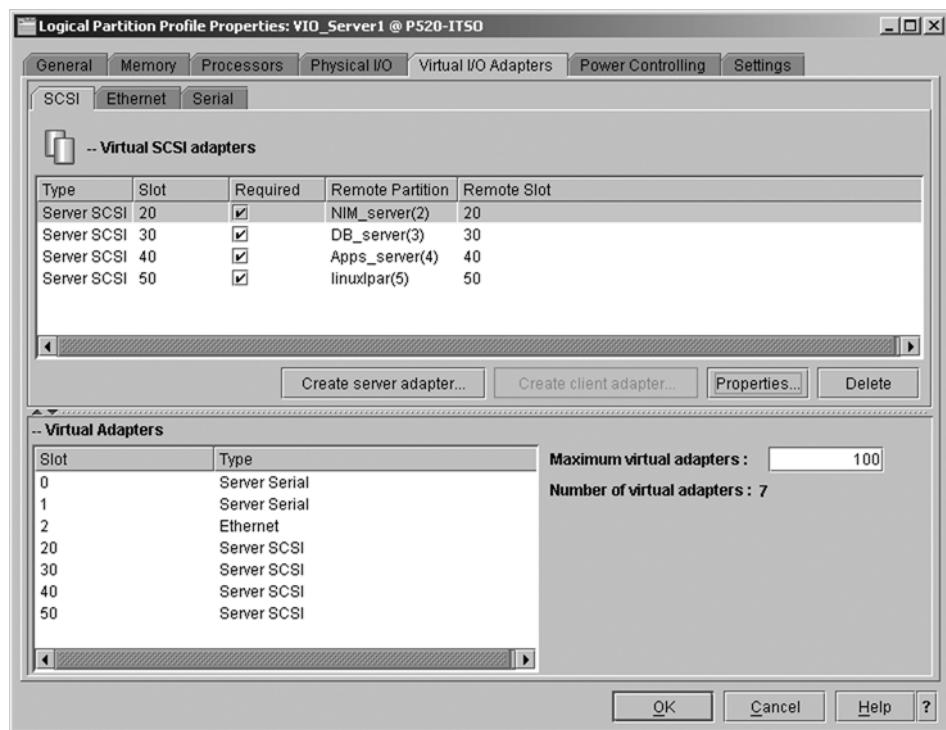


Рис. 4-29. Вкладка свойств VIOS SCSI

Важно. При установленном флагке Required (Требуется) вы не сможете динамически перемещать ресурсы между разделами.

4.4.3. Создание клиентских разделов

В этом разделе показано, как создать четыре клиентских раздела для нашего базового сценария виртуального ввода-вывода. Определение параметров похоже на создание нашего раздела VIOS, но вместо того, чтобы щелкнуть по Virtual I/O, мы выберем AIX or Linux (AIX или Linux). Создание наших клиентских разделов можно увидеть на рисунке 4-30. Выполните следующие шаги для создания клиентских разделов:

1. Перезапустите мастер Create Logical Partition Wizard, как вы это делали в начале предыдущей процедуры 4.2.1 «Определение раздела с Virtual I/O Server». Обратитесь к рисунку 4-2, чтобы увидеть это окно.
2. Установите флажок AIX or Linux и введите идентификатор и имя раздела, как показано на рисунке 4-30.

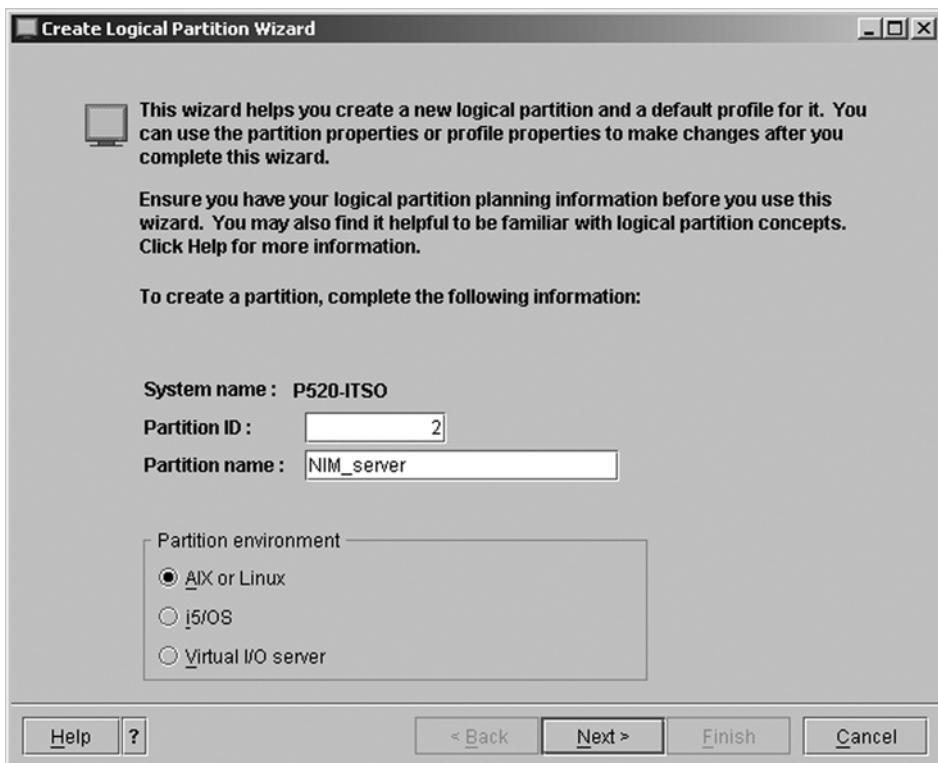


Рис. 4-30. Создание раздела NIM_server

3. Повторите шаги 4–15 раздела 4.2.1 «Определение раздела с Virtual I/O Server» со следующими исключениями:
 - a. В шаге 6 используйте значения 256/512/768 МВ для настройки минимального/желаемого/максимального объема памяти. «Боевые» системы могут потребовать дополнительной памяти.
 - b. В шаге 8 используйте значения 0.1/0.2/0.4 процессорных единиц для настройки минимального/желаемого/максимального значений. «Боевые» системы могут потребовать дополнительных процессорных единиц.
 - c. В шаге 9 используйте значения 1/2/4 для настройки минимального/желаемого/максимального количества виртуальных процессоров.
 - d. Пропустите шаг 10, щелкнув по **Next**, вместо того, чтобы выбрать вариант создания физических компонентов ввода-вывода.

Для трех других разделов (DB_server, Apps_server и linuxlpar) шаги те же самые, кроме изменений, сделанных в шаге 3. Определенные нами значения предназначались для тестовых целей. Вам следует учитывать свои требования к размерам сервера, чтобы правильно задать размеры разделов вашей базы данных и приложений.

На рисунке 4-31 показан результат создания вами клиентских разделов.

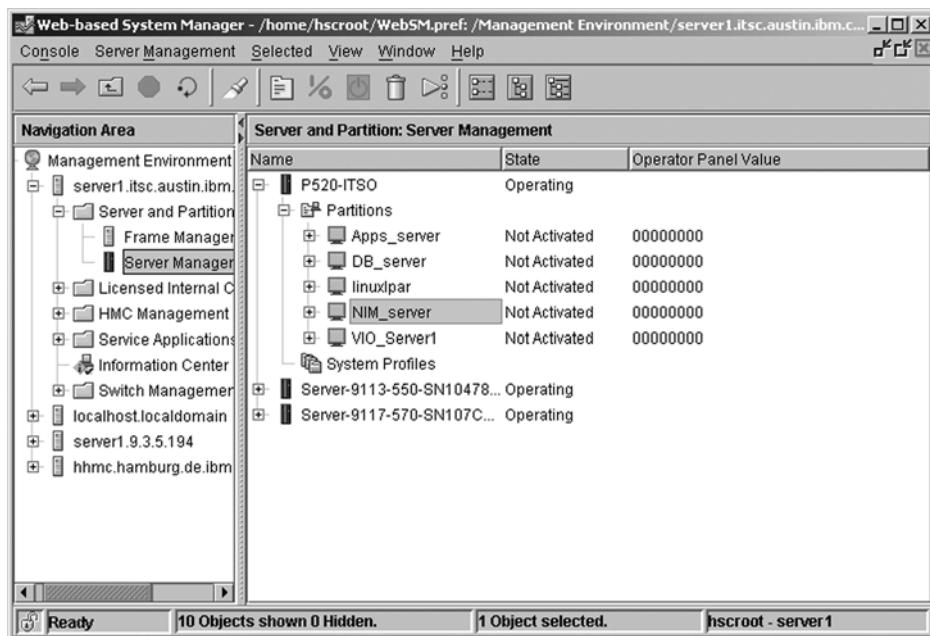


Рис. 4-31. Вид НМС с созданными новыми разделами

4.4.4. Создание виртуальных Ethernet-адаптеров для клиентских разделов

Эти шаги точно совпадают с созданием виртуального Ethernet-адаптера для VIO_Server1, но не устанавливайте флажок Access External Networks. Выполните следующие шаги для создания виртуальных Ethernet-адаптеров:

1. В НМС щелкните правой кнопкой мыши по имени профиля клиентского раздела NIM_server и выберите в меню пункт Properties (рисунок 4-32).
2. Добавьте виртуальный Ethernet-адаптер, выбрав вкладку Virtual I/O, вкладку Ethernet и щелкнув по кнопке Create adapter. Определите номер слота, исходя из рисунка 4-22, для виртуального Ethernet-адаптера. После этого щелкните OK.

Примечание. Идентификатор VLAN устанавливается в 1, так как мы хотим, чтобы все наши разделы были в одной и той же VLAN. Номер слота и идентификатор VLAN в нашем базовом сценарии одинаковы для разделов NIM_server, DB_server, Apps_server и linuxlpar. Это сделано для простоты достижения нашей цели.

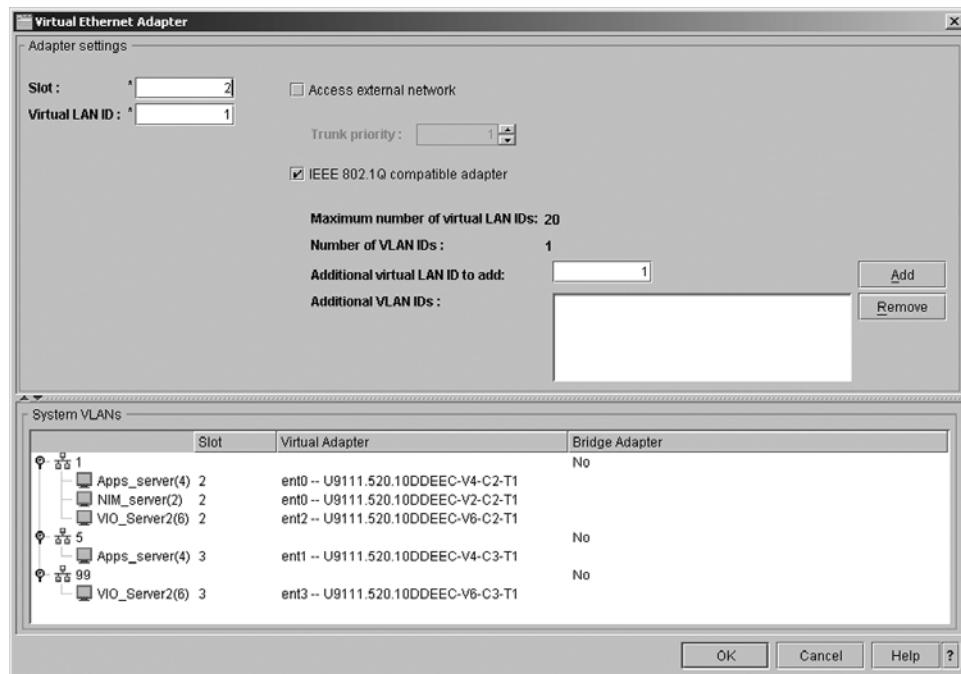


Рис. 4-32. Свойства виртуального Ethernet

4.4.5. Создание виртуального SCSI-адаптера для клиентских разделов

Используйте следующие шаги для определения клиентского виртуального SCSI-адаптера:

1. В HMC щелкните правой кнопкой по имени профиля клиентского раздела NIM_server и выберите в меню пункт **Properties** (рисунок 4-33).
2. Добавьте виртуальный SCSI-адаптер, выбрав вкладку **Virtual I/O**, а затем – вкладку **SCSI** и щелкнув по кнопке **Create adapter**. Определите номер слота, исходя из рисунка 4-22, для виртуального SCSI-адаптера. Выберите значения **Server partition** (Серверный раздел) и **Server partition slot** (Слот серверного раздела) и щелкните **OK**.

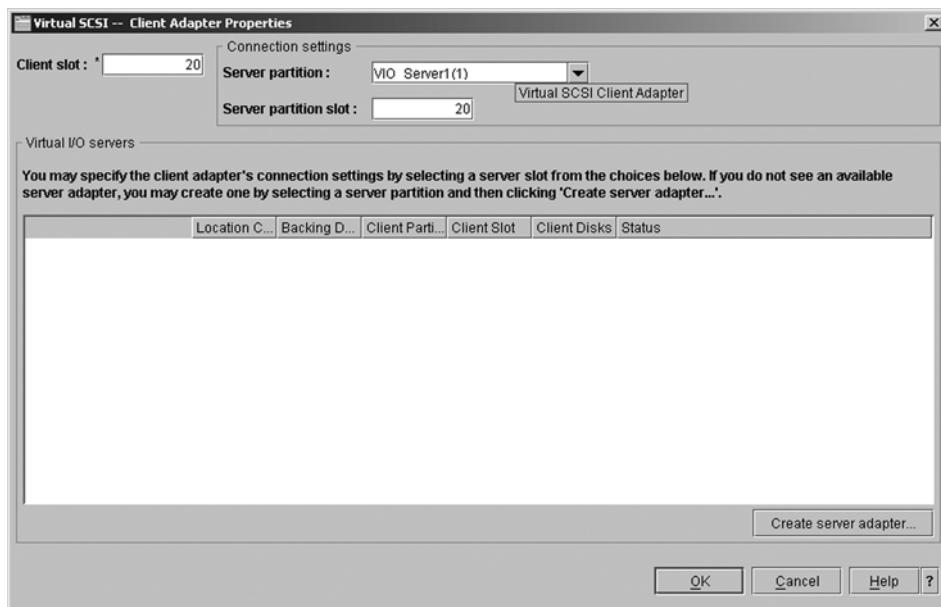


Рис. 4-33. Задание свойств клиентского виртуального SCSI-адаптера

4.4.6. Определение групп томов и логических томов

Выполните следующие шаги, чтобы определить логические тома, необходимые для создания виртуального диска для rootvg клиентского раздела в соответствии с нашим базовым сценарием:

1. Войдите в систему с идентификатором пользователя `padmin` и запустите команду `cfgdev`, чтобы перестроить список видимых устройств, используемых VIOS.

Теперь серверу VIOS доступны серверные виртуальные SCSI-адаптеры. Имена этих адаптеров будут иметь вид `vhostx`, где `x` является номером, присвоенным системой.

2. Используйте команду `lsdev -virtual`, чтобы убедиться в доступности вашего нового виртуального SCSI-адаптера, как показано в примере 4-3.

Пример 4-3. Интерфейс командной строки виртуального ввода-вывода

```
$ lsdev -virtual
name      status description
ent2     Available Virtual I/O Ethernet Adapter (l-lan)
vhost0   Available Virtual SCSI Server Adapter
vhost1   Available Virtual SCSI Server Adapter
vhost2   Available Virtual SCSI Server Adapter
vhost3   Available Virtual SCSI Server Adapter
vsas0    Available LPAR Virtual Serial Adapter
```

Если устройства недоступны, то придется их определить. Вы можете использовать команду `rmdev -dev vhost0 -recursive` для каждого устройства и затем, если необходимо, перезагрузить VIOS. При перезагрузке менеджер конфигурации configuration manager будет распознавать оборудование и заново создаст vhost-устройства.

В нашем базовом сценарии мы будем создавать группу томов с именем `rootvg_clients` на диске `hdisk2` и создавать логические разделы, служащие загрузочными дисками для наших клиентских разделов.

3. Создайте группу томов и назначьте `hdisk2` для `rootvg_clients` командой `mkvg`.

Пример 4-4. Реальное выполнение команды mkvg

```
$ mkvg -f -vg rootvg_clients hdisk2
rootvg_clients
```

4. Определите все логические тома, которые будут представляться для клиентских разделов как диски `hdisks`. В нашем случае эти логическими томами будут наши `rootvg` для клиентских разделов (см. пример 4-5).

Пример 4-5. Создание логических томов

```
$ mklv -lv rootvg_dbsrv rootvg_clients 10G
rootvg_dbsrv
$ mklv -lv rootvg_apps rootvg_clients 10G
rootvg_apps
$ mklv -lv rootvg_nim rootvg_clients 10G
rootvg_nim
$ mklv -lv rootvg_lnx rootvg_clients 2G
rootvg_lnx
```

5. Определите SCSI-связи для создания виртуального целевого устройства, которое связывается с логическим томом, определенным вами в предыдущем шаге. В примере 4-6 мы имеем четыре виртуальных хост-устройства в VIOS. Этими vhost-устройствами являются устройства, которые мы будем связывать с нашим логическим томом.

Пример 4-6. Создание связей виртуальных устройств

```
$ lsdev -vpd | grep vhost
vhost3 U9111.520.10DDEC-V1-C50      Virtual SCSI Server Adapter
vhost2 U9111.520.10DDEC-V1-C40      Virtual SCSI Server Adapter
vhost1 U9111.520.10DDEC-V1-C30      Virtual SCSI Server Adapter
vhost0 U9111.520.10DDEC-V1-C20      Virtual SCSI Server Adapter
$ mkvdev -vdev rootvg_nim -vadapter vhost0 -dev vnim
vnim Available
$ mkvdev -vdev rootvg_dbsrv -vadapter vhost1 -dev vdbsrv
vdbsrv Available
$ mkvdev -vdev rootvg_apps -vadapter vhost2 -dev vapps
vapps Available
$ mkvdev -vdev rootvg_lnx -vadapter vhost3 -dev vlnx
vlnx Available
$ lsdev -virtual
name    status   description
ent2    Available Virtual I/O Ethernet Adapter (1-lan)
vhost0  Available Virtual SCSI Server Adapter
vhost1  Available Virtual SCSI Server Adapter
vhost2  Available Virtual SCSI Server Adapter
vhost3  Available Virtual SCSI Server Adapter
vsas0   Available LPAR Virtual Serial Adapter
vapps   Available Virtual Target Device - Logical Volume
vdbsrv  Available Virtual Target Device - Logical Volume
vlnx    Available Virtual Target Device - Logical Volume
vnim    Available Virtual Target Device - Logical Volume
```

Примечание. Судя по результатам выполнения команды `lsdev -vpd`, связи точно соответствуют предусмотренной нами нумерации слотов (см. рис. 4-22). Например, у устройства `vhost0` номер слота в VIOS равен 20 (U9111.520.10DDEC-V1-C20), и оно будет выделено разделу `NIM_server`. У раздела `NIM_server` слот его виртуального SCSI-устройства установлен равным 20. Это сделано для более легкого связывания виртуальных SCSI-устройств на серверной и клиентской стороне.

6. Используйте команду `lsmap`, чтобы убедиться в правильности всех логических связей между вновь созданными устройствами, как показано в примере 4-7.

Пример 4-7. Проверка связывания

```
$ lsmap -vadapter vhost0
SVSA          Physloc           Client Partition ID
-----
vhost0        U9111.520.10DDEC-V1-C20      0x00000000
VTD           vnim
LUN           0x8100000000000000
Backing device rootvg_nim
Physloc
```

7. Теперь вы готовы к установке AIX 5L V5.3 или Linux в каждом разделе. В данный момент диски должны быть доступными для использования клиентскими разделами.

Подсказка. Тот же подход применим к созданию групп томов, которые будут использоваться для хранения данных.

4.4.7. Создание общего Ethernet-адаптера (SEA)

Для создания общего Ethernet-адаптера выполните следующие шаги:

1. С помощью команды `lsdev` в VIOS проверьте доступность транкового Ethernet-адаптера (см. пример 4-8).

Пример 4-8. Проверка для SEA

```
$ lsdev -virtual
name          status description
ent2          Available Virtual I/O Ethernet Adapter (l-lan)
vhost0         Available Virtual SCSI Server Adapter
vhost1         Available Virtual SCSI Server Adapter
vhost2         Available Virtual SCSI Server Adapter
vhost3         Available Virtual SCSI Server Adapter
vsat0          Available LPAR Virtual Serial Adapter
vapps          Available Virtual Target Device - Logical Volume
vdbsrv         Available Virtual Target Device - Logical Volume
vlnx           Available Virtual Target Device - Logical Volume
vnim           Available Virtual Target Device - Logical Volume
```

2. Выберите физический Ethernet-адаптер, который подходит для создания SEA. Командой `lsdev` будет выведен список доступных физических адаптеров (см. пример 4-9).

Пример 4-9. Проверка физического Ethernet-адаптера

```
$ lsdev -type adapter
name          status description
ent0          Available 2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent1          Available 2-Port 10/100/1000 Base-TX PCI-X Adapter (1410890
ent2          Available Virtual I/O Ethernet Adapter (l-lan)
ide0          Defined ATA/IDE Controller Device
sisccsia0     Available PCI-X Dual Channel Ultra320 SCSI Adapter
vhost0         Available Virtual SCSI Server Adapter
vhost1         Available Virtual SCSI Server Adapter
vhost2         Available Virtual SCSI Server Adapter
vhost3         Available Virtual SCSI Server Adapter
vsat0          Available LPAR Virtual Serial Adapter
```

3. Используйте команду `mkvdev` для создания нового устройства `ent3` в качестве SEA. Устройство `ent0` будет использоваться как физический Ethernet-адаптер, а устройство `ent2` – как виртуальный Ethernet-адаптер (пример 4-10).

Пример 4-10. Создание SEA

```
$ mkvdev -sea ent0 -vadapter ent2 -default ent2 -defaultid 1  
ent3 Available  
en3  
et3
```

4. Проверьте доступность вновь созданного SEA (пример 4-11).

Пример 4-11. Проверка устройства SEA

```
$ lsdev -virtual  
name      status    description  
ent2      Available Virtual I/O Ethernet Adapter (l-lan)  
vhost0    Available Virtual SCSI Server Adapter  
vhost1    Available Virtual SCSI Server Adapter  
vhost2    Available Virtual SCSI Server Adapter  
vhost3    Available Virtual SCSI Server Adapter  
vsas0     Available LPAR Virtual Serial Adapter  
vapps     Available Virtual Target Device - Logical Volume  
vdbsrv    Available Virtual Target Device - Logical Volume  
vlnx      Available Virtual Target Device - Logical Volume  
vnim      Available Virtual Target Device - Logical Volume  
ent3      Available Shared Ethernet Adapter
```

Адаптер SEA будет образовывать мост, позволяющий VLAN, действующей между разделами, вести обмен с внешней сетью.

По нашему базовому сценарию, у нас имеется одно физическое соединение с общедоступной сетью через физический Ethernet, поэтому мы должны пойти дальше и сконфигурировать общий Ethernet-адаптер как мост между общедоступной сетью и действующей между разделами VLAN.

В нашем сценарии мы используем следующие значения (таблица 4-1).

Таблица 4-1. Сетевые настройки

Настройки	Значение
hostname (имя хоста)	VIO_Server1
IP-address (IP-адрес)	9.3.5.130
netmask (маска сети)	255.255.255.0
gateway (шлюз)	9.3.5.41

С помощью команды `mktcpip` сконфигурируйте вновь созданный интерфейс `ent3` адаптера SEA (см. пример 4-12).

Пример 4-12. Определение SEA

```
$ mktcpip -hostname VIO_Server1 -inetaddr 9.3.5.130 -interface en3 -netmask  
255.255.255.0 -gateway 9.3.5.41
```

4.4.8. Установка AIX 5L в клиентском разделе

В этом разделе описывается метод установки AIX 5L Version 5.3 в ранее определенный клиентский раздел. Вы можете воспользоваться собственным предпочтительным методом, но для нашего базового сценария мы выбрали вариант установки в разделы DB_server и Apps_server с помощью Менеджера сетевой инсталляции (Network Installation Manager), прилагаемого к AIX 5L Version 5.3. Мы также будем использовать виртуальные Ethernet-адAPTERы для сетевой загрузки и виртуальные SCSI-диски, которые мы ранее назначили клиентским разделам для rootvg.

При работе с сервером Network Installation Manager (NIM) пользуйтесь *NIM: From A to Z in AIX 4.3*, SG24-5524 или *Partitioning Implementations for IBM @server p5 Servers*, SG24-7039, содержащими подробную информацию о создании NIM-серверов и установке AIX 5L с помощью NIM.

Полагая, что NIM-мастер сконфигурирован (в нашем случае мы установили раздел NIM_server в качестве нашего NIM-мастера), основные шаги выполнения установки AIX 5L с помощью NIM будут выглядеть так:

1. Создайте клиента NIM-машины DB_server и определения в вашем NIM-мастере. (См. пример 4-13, показывающий, как проверить ресурсы).

Пример 4-13. Проверка выделения ресурсов

```
# lsnim -l DB_server
DB_server:
class          - machines
type           - standalone
connect        - shell
platform       - chrp
netboot_kernel - mp
if1            - network1 DB_server b60f90003002
cable_type1   - N/A
Cstate         - BOS installation has been enabled
prev_state     - ready for a NIM operation
Mstate         - not running
boot           - boot
lpp_source     - 53d_lppsource
nim_script     - nim_script
spot           - 53d_spot
control        - master
```

2. Инициируйте процесс установки, активировав клиентский раздел DB_server в режиме SMS (см. рис. 4-34).
3. Задайте IP-адрес NIM-мастера и адрес клиента выбором опции 2 Setup Remote IPL (Настройка удаленной загрузки) (см. пример 4-14).

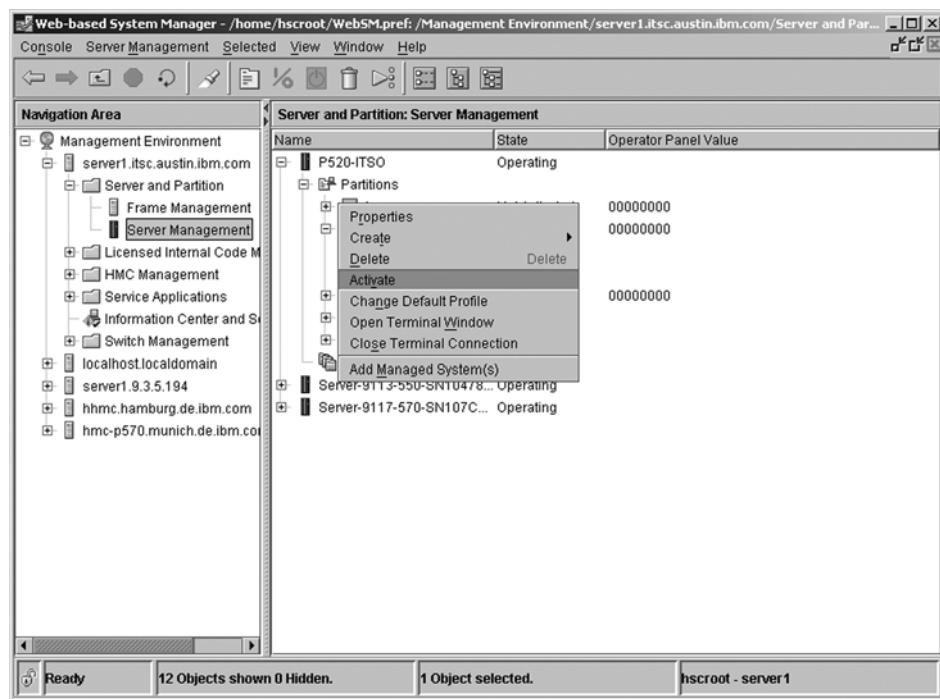


Рис. 4-34. Активация раздела DB_server

Пример 4-14. Меню SMS Firmware (Микрокод SMS)

```

Version SF235_137
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.

-----
Main Menu
1. Select Language
2. Setup Remote IPL (Initial Program Load)
3. Change SCSI Settings
4. Select Console
5. Select Boot Options

```

4. Выберите опцию 1, как показано в примере 4-15.

Пример 4-15. АдAPTERы, доступные для сетевой загрузки

```

PowerPC Firmware
Version SF235_137
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.

-----
NIC Adapters
Device Location Code Hardware
Address

```

1. Interpartition Logical LAN U9111.520.10DDEC-V3-C2-T1 b60f90003002
 2. Interpartition Logical LAN U9111.520.10DDEC-V3-C3-T1 b60f90003003
-

Примечание. Interpartition Logical LAN (Логическая сеть между разделами) с номером 1 является виртуальным Ethernet-адаптером, определенным в NIM-мастере для клиента DB_server.

```
DB_server:  
class          - machines  
type           - standalone  
connect        - shell  
platform       - chrp  
netboot_kernel - mp  
if1            - network1 DB_server b60f90003002
```

5. Выберите опцию 1 для IP Parameters (IP-параметры), затем пройдите через каждую опцию и определите IP-адреса, как показано в примере 4-16.

Пример 4-16. Определение IP-адресов

```
PowerPC Firmware  
Version SF235_137  
SMS 1.6 (c) Copyright IBM Corp. 2000,2005 All rights reserved.  
  
-----  
IP Parameters  
Interpartition Logical LAN: U9111.520.10DDEC-V3-C2-T1  
1. Client IP Address [9.3.5.133]  
2. Server IP Address [9.3.5.132]  
3. Gateway IP Address [9.3.5.41]  
4. Subnet Mask [255.255.255.000]
```

6. Вернитесь в главное меню и выполните загрузку раздела через сеть.

4.4.9. Зеркалирование rootvg сервера VIOS

После завершения установки VIOS зеркалирование его группы томов rootvg на втором физическом диске может быть выполнено с помощью нижеперечисленных команд. Зеркалирование VIOS rootvg выполняется так:

1. Используйте команду `extendvg` для включения диска hdisk1 в группу томов rootvg. Здесь применима та же идея LVM; вы не можете использовать hdisk, принадлежащий другой группе томов; диск должен быть того же самого или большего размера.
2. Используйте команду `lspv`, как показано в примере 4-17, для проверки, была ли rootvg расширена для включения hdisk1.

Пример 4-17. Вывод команды lspv

```
$ lspv  
NAME      PVID      VG      STATUS  
hdisk0   00cddeec2dce312d  rootvg  active
```

hdisk1	00cddeec87e69f91	rootvg active
hdisk2	00cddeec68220f19	rootvg_clients active
hdisk3	00cddeec685b3e88	None

3. Используйте команду `mirrorios` для зеркалирования rootvg на hdisk1, как показано в примере 4-18. С флагом `-f` команда `mirrorios` будет автоматически перезагружать раздел VIOS.

Внимание. Если вы намереваетесь иметь в rootvg логические разделы, используемые как виртуальные SCSI-устройства, то вначале запустите команду `mirrorios`. Логические тома нельзя зеркаливать при их использовании в качестве виртуальных SCSI-дисков.

Пример 4-18. Команда mirrorios

```
$ mirrorios -f hdisk1
```

4. Проверьте, выполнено ли зеркалирование логических томов и обновлена ли обычная последовательность загрузки, как показано в примере 4-19.

Пример 4-19. Логические разделы связаны с двумя физическими разделами

```
$ lsvg -lv rootvg
rootvg:
LV NAME      TYPE    LPs   PPs   PVs   LV STATE      MOUNT POINT
hd5          boot     1     2     2   closed/syncd  N/A
hd6          paging   8    16     2   open/syncd   N/A
paging00     paging  16    32     2   open/syncd   N/A
hd8          jfs2log  1     2     2   open/syncd   N/A
hd4          jfs2     3     6     2   open/syncd   /
hd2          jfs2    20    40     2   open/syncd  /usr
hd9var       jfs2     9    18     2   open/syncd  /var
hd3          jfs2    21    42     2   open/syncd  /tmp
hd1          jfs2   160   320    2   open/syncd  /home
hd10opt      jfs2     1     2     2   open/syncd /opt
lg_dumplv    sysdump 16    16     1   open/syncd  N/A

$ bootlist -mode normal -ls
hdisk0 blv=hd5
hdisk1 blv=hd5
```

4.5. Взаимодействие с клиентскими UNIX-разделами

В следующем разделе описано, как VIOS предоставляет ресурсы разделам с AIX 5L или Linux. Этими ресурсами могут быть устройства хранения и оптические устройства через виртуальный SCSI или сетевые соединения через виртуальный Ethernet. Хотя виртуальные сервисы ввода-вывода могут обеспечиваться в Linux-разделе, работающем как сервер ресурсов хранения и соединений Ethernet, этот раздел руководства применим только для функции VIOS в технологии Advanced POWER Virtualization, VIOS V1.1 или V1.2.

На момент написания книги разделы с i5/OS в системах IBM @server p5 не взаимодействовали с серверами VIOS, поэтому здесь ничего не говорится о клиентских разделах с i5/OS.

4.5.1. Виртуальные SCSI-сервисы

В случае виртуального SCSI взаимодействие между VIOS и клиентским разделом с AIX 5L или Linux активируется, когда в конфигурациях как серверного виртуального SCSI-адаптера, так и клиентского виртуального SCSI-адаптера совпадают номера слотов их профилей разделов, и операционные системы распознают их виртуальный адаптер (с помощью команды `cfgmgr` для динамически добавляемых виртуальных SCSI-адаптеров).

После активации взаимодействия между серверными и клиентскими виртуальными SCSI-адаптерами необходимо связать ресурсы хранения из VIOS с клиентским разделом. Клиентский раздел конфигурирует и использует ресурсы хранения при его запуске или при реконфигурации во время выполнения.

Эти процессы протекают следующим образом:

- ▶ HMC активирует взаимодействие между виртуальными SCSI-адаптерами.
- ▶ В VIOS выполняется связывание ресурсов хранения.
- ▶ Клиентский раздел использует хранилище после своей загрузки или при исполнении команды `cfgmgr`.

Более подробно реализация виртуального SCSI описана в разделе 3.9 «Ознакомление с виртуальным SCSI».

На рисунке 4-35 показана последовательность активации виртуальных SCSI-ресурсов для клиентов с AIX 5L или Linux. Обратите внимание, что разделы VIOS и клиентов не нужно перезапускать, когда новые серверные и клиентские виртуальные SCSI-адаптеры создаются с помощью меню динамического LPAR в HMC и когда активируются динамические операции с LPAR в операционных системах.



Рис. 4-35. Базовая последовательность конфигурирования виртуальных SCSI-ресурсов

Для обеспечения взаимодействия клиентских разделов с AIX 5L или Linux с виртуальными SCSI-ресурсами необходимо выполнить следующие шаги:

1. Спланируйте, какой виртуальный слот будет использоваться в VIOS для серверного виртуального SCSI-адаптера, а какой слот в клиентском разделе с AIX 5L или Linux – для клиентского виртуального SCSI-адаптера (у каждого раздела есть свой пул виртуальных слотов). В VIOS учитывайте диапазон номеров слотов для виртуальных адаптеров, обслуживающих конкретный раздел (например, слоты 20–29 для серверных виртуальных SCSI-адаптеров клиентских разделов с AIX 5L).
2. Определите серверный виртуальный SCSI-адаптер в VIOS.
3. Определите клиентский SCSI-адаптер в клиентском разделе с AIX 5L или Linux.
4. Свяжите необходимые SCSI-ресурсы командой `mkvdev`, как описано в разделе 4.4.6 «Определение групп томов и логических томов».

После того как VIOS и клиентские разделы с AIX 5L или Linux стали взаимодействовать друг с другом, в VIOS может оперативно устанавливаться связывание виртуальных устройств и клиентские разделы с AIX 5L или Linux оперативно могут реконфигурироваться для их использования, как показано на рисунке 4-36.



Рис. 4-36. Базовая последовательность конфигурирования виртуальных SCSI-ресурсов

На рисунке 4-37 показаны шаги создания новых серверных и клиентских виртуальных SCSI-адаптеров и определения первоначальных ресурсов хранения для клиентского раздела с AIX 5L.

Виртуальные SCSI-сервисы для разделов с AIX 5L

Клиентские разделы с AIX 5L могут использовать виртуальные SCSI-устройства с определенными в VIOS связями после запуска операционной системы или реконфигурирования системы командами `cfgmgr` или `mkdev`.

Ресурсы дискового типа с определенными связями (физические тома или логические тома, охватываемые VIOS) появляются в клиентском разделе с AIX 5L как устройства типа `hdisk` (например, `hdisk0`, `hdisk1`). Клиентский виртуальный SCSI-адаптер может использовать эти устройства, как и любое физически подключен-

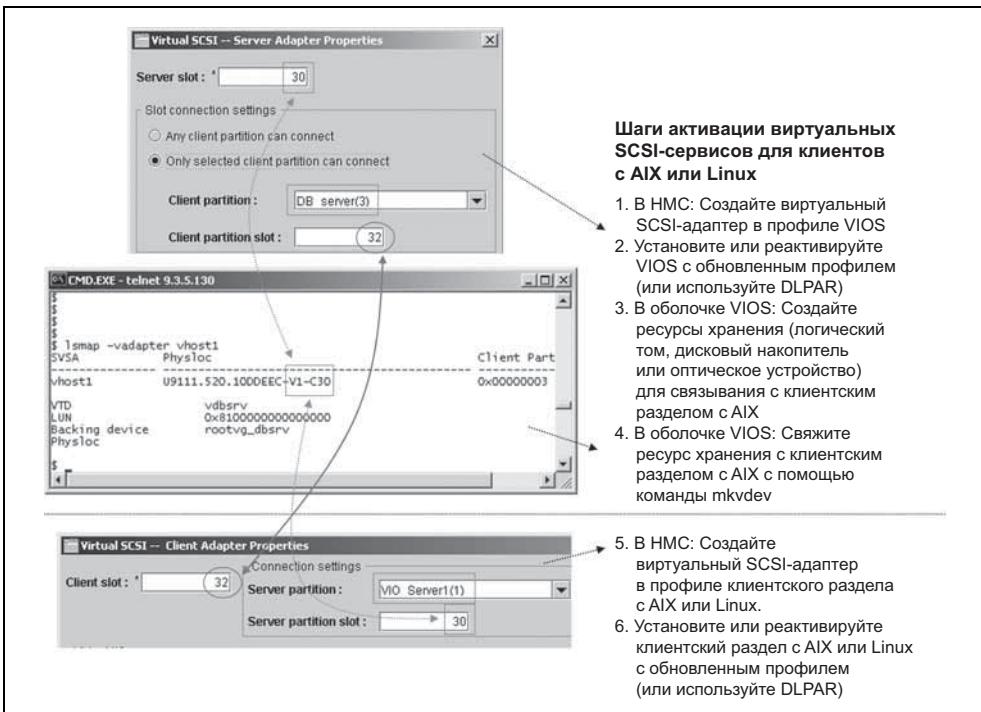


Рис. 4-37. Шаги по активации виртуального SCSI-сервиса для клиентского раздела с AIX 5L

ное hdisk-устройство, для загрузки, подкачки, зеркалирования или для любой другой поддерживаемой AIX 5L функции. Ресурсы оптического типа (DVD-ROM и DVD-RAM) появляются как устройства типа CD (например, cd0).

Виртуальные SCSI-сервисы для разделов с Linux

Конфигурирование для разделов с Linux с целью позволить им использовать виртуальные SCSI-адAPTERы и устройства выполняется подобно конфигурированию для разделов с AIX 5L. Разделы с Linux обращаются с виртуальными SCSI-устройствами как с дисковыми накопителями SCSI. Пользователи должны знать о следующих факторах, касающихся клиентских разделов с Linux:

- ▶ Виртуальные SCSI-адAPTERы в клиентском разделе с Linux перечисляются в каталоге /sys/class/scsi_host; такой каталог содержит управляющие файлы для каждого виртуального адAPTERа. Клиентские виртуальные SCSI-адAPTERы могут требовать повторного сканирования для только что связанных из VIOS виртуальных SCSI-ресурсов с помощью тех же самых интерфейсов, как любой другой SCSI-адAPTER.
- ▶ Виртуальные дисковые накопители SCSI в клиентском разделе с Linux могут быть видны в дереве устройств как устройства типа sdx (например, sda). Обратите внимание, что Linux показывает в дереве устройств как виртуальные дисковые накопители (виртуальные ресурсы хранения SCSI, связываемые из VIOS), так и дисковые разделы, созданные внутри этих виртуальных дисковых накопителей.

4.5.2. Виртуальные Ethernet-ресурсы

Вторым типом ресурсов, вызывающих взаимодействие между клиентскими разделами с AIX 5L или Linux и VIOS, является общий Ethernet-адаптер в VIOS. Эта функция позволяет разделам с AIX 5L соединяться с внешними сетями без физического адаптера.

Реализация виртуальных Ethernet-адаптеров основана на определении сетевых интерфейсов, которые осуществляют соединение через гипервизор POWER с виртуальным Ethernet-коммутатором системы, поддерживающим IEEE VLAN. Все разделы, ведущие обмен в виртуальной Ethernet-сети, являются одноранговыми. В системе могут быть определены до 4096 отдельных сетей IEEE VLAN. Каждый раздел может иметь до 65533 виртуальных Ethernet-адаптеров, подключенных к виртуальному Ethernet-коммутатору, и каждый адаптер может быть подключен к 21 различной сети IEEE VLAN (20 VID и 1 PVID).

Подробнее о технических деталях реализации виртуального Ethernet рассказывается в разделе 3.8 «Ознакомление с виртуальным и разделяемым Ethernet».

Активация и настройка виртуального Ethernet-адаптера в клиентском разделе с AIX 5L или Linux не требует какого-либо специального оборудования или программного обеспечения. После предоставления разделу конкретного виртуального Ethernet внутри этого раздела создается сетевое устройство. Затем пользователь может настроить соответствующую TCP/IP-конфигурацию для информационного обмена с другими разделами..

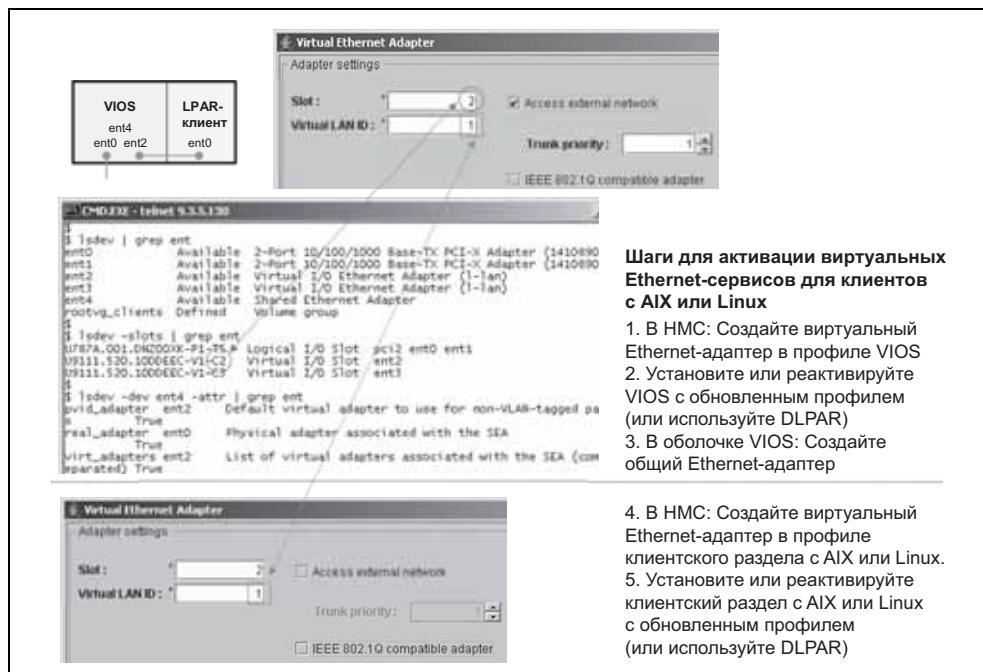


Рис. 4-38. Последовательность шагов, необходимых для активации подключения к виртуальному Ethernet

Чтобы позволить клиентскому разделу с AIX 5L или Linux обмениваться с внешними Ethernet-сетями, VIOS действует как мост и ретранслирует IP-пакеты из клиентского виртуального Ethernet-адаптера. Это делается с помощью создания разделяемого Ethernet-адаптера (SEA) в VIOS, который соединяет серверный виртуальный Ethernet-адаптер и физический Ethernet-адаптер в сервере.

На рисунке 4-38 приведены пример пошаговой процедуры конфигурирования виртуальных Ethernet-сервисов и детали, касающиеся разделяемого Ethernet-адаптера в VIOS.

Более подробную информацию о разделяемом Ethernet-адаптере можно получить в разделе 3.8 «Ознакомление с виртуальным и разделяемым Ethernet».

Примечание. Нужно помнить, что адAPTERы виртуального SCSI или виртуального Ethernet, создаваемые с помощью динамических операций с LPAR, не отражаются в профилях раздела и существуют только во время раздела.



5

Установка Virtual I/O: расширенная конфигурация

Эта глава начинается с обсуждения дополнительных тем, касающихся Virtual I/O, которые важно понять, прежде чем приступить к установке, включая:

- ▶ Обеспечение повышения доступности VIOS.
Сюда входит обсуждение того, когда следует использовать Virtual I/O Server и как можно достичь высокой доступности этих серверов для связи с внешними сетями. Мы применим эти концепции в трех сценариях и продемонстрируем, как использовать дополнительные настройки, позволяющие увеличить уровень избыточности:
 - Сценарий 1: Зеркалирование логического тома
 - Сценарий 2: Обработка отказа SEA
 - Сценарий 3: MPIO в клиенте с SAN в VIOS
- ▶ Также мы продемонстрируем, как активировать раздел перед началом установки Linux на клиент VIO.
- ▶ И наконец, мы перечислим поддерживаемые конфигурации, куда включен раздел с советами относительно использования IBM TotalStorage Solutions, HACMP и GPFS в визуализируемом окружении.

5.1. Обеспечение повышения доступности VIOS

В этот раздел входит обсуждение требований и концепций для повышения доступности Virtual I/O Server и того, когда следует использовать несколько серверов Virtual I/O Server для предоставления Virtual SCSI и общего Ethernet клиентским разделам.

При определении количества серверов Virtual I/O Server во внимание принимается несколько соображений. Допустимое время простоя клиента и рассчитанная средняя загрузка подсистемы ввода-вывода для сети и хранилища, возможность управления системой – это факторы, которые необходимо учитывать при планировании создаваемых Virtual I/O Server.

Для небольших систем с ограниченными ресурсами и ограниченным количеством адаптеров ввода-вывода второй VIOS может не получить необходимых ресурсов. При ограниченном количестве адаптеров ввода-вывода добавление второго сервера VIOS может значительно сказаться на производительности, предоставляемой клиенту, если дополнительные адаптеры ввода-вывода увеличивают избыточность, а не пропускную способность.

Для больших систем существует меньше ограничений по ресурсам, так что несколько VIOS могут быть размещены без влияния на общую производительность клиента. Дополнительные адаптеры ввода-вывода при использовании дополнительными Virtual I/O Server влияют как на пропускную способность, так и на избыточность.

Ограничение. IVM поддерживает только один Virtual I/O Server.

5.1.1. Обеспечение повышения доступности путем увеличения количества Virtual I/O Server

Хотя избыточность может быть заложена в сам VIOS с помощью MPIO и LVM-зеркалирования (RAID) для устройств хранения и агрегирования каналов для сетевых устройств, Virtual I/O Server должен отдавать предпочтение клиенту. Запланированные перерывы в работе, такие как программные обновления (см. 6.4.1 «Конкурентные программные обновления для VIOS»), и незапланированные перерывы в работе, такие как простои по виду оборудования, мешают обеспечению доступности 24x7.

Наличие нескольких Virtual I/O Server, предоставляющих доступ к одним и тем же ресурсам, обеспечивает клиенту хорошую избыточность, так же как и наличие дополнительных адаптеров, подключенных к клиенту напрямую.

Избыточность Virtual SCSI

Если на клиенте доступен MPIO, каждый VIOS может представлять собой виртуальное SCSI-устройство, физически подключенное к одному и тому же физическому диску. Это обеспечивает избыточность как для самого VIOS, так и для каждого из адаптеров, коммутаторов или устройств, находящихся между VIOS и диском.

При использовании зеркалирования логического тома на клиенте каждый VIOS может быть представлен виртуальным устройством SCSI, физически подключенным к своему диску и используемым при обычном в AIX 5L зеркалировании группы томов на клиенте. Таким образом, потенциально достигается больший уро-

вень надежности благодаря исключению диска как единственного потенциального источника ошибки. Зеркалирование группы томов клиента требуется также и в том случае, когда логический том VIOS используется как виртуальное SCSI устройство на клиенте. В этом случае виртуальные устройства SCSI ассоциируются с разными дисками SCSI, каждый из которых управляется одним из двух VIOS.

Рисунок 5-1 демонстрирует дополнительные настройки при использовании MPIO и LVM зеркалирования в клиенте VIO одновременно: два диска Virtual I/O Server на одном клиенте. Клиент использует MPIO для доступа к SAN-диску и зеркалирование LVM для доступа к SCSI-дискам. С точки зрения клиента, следующие ситуации должны обрабатываться без отключения клиента:

- ▶ Любой путь на SAN-диске может отказать и, тем не менее, клиент все равно должен иметь возможность получить доступ к данным на SAN-диске другим путем. Для восстановления сбояного пути на SAN-диске после восстановления не предпринимается никаких действий.
- ▶ Ошибка диска SCSI приводит к устареванию разделов (stale partitions)¹ для группы томов с назначенными виртуальными дисками, однако клиент все равно сможет получить доступ к данным на другой копии зеркального диска. После устранения ошибки на SCSI-диске или перезагрузки VIOS все устаревшие разделы будут синхронизированы с помощью команды **varyonvg**.
- ▶ Любой из VIOS может быть перезагружен, что приведет к временному отказу в доступе к пути на SAN-диске и устареванию разделов в группе томов на SCSI-дисках, как и в предыдущем случае.

Примечание. Если через настройки вам доступны и зеркалирование, и MPIO, MPIO является предпочтительным вариантом для добавления в клиент избыточных дисков. LVM-зеркалирование приводит к устареванию разделов, требующему синхронизации, а MPIO – нет. При использовании служб SAN также можно использовать зеркалирование.

Дальнейшие примеры настройки виртуального SCSI см. 5.6.1 «Поддерживаемые настройки VSCSI».

Доступность общего Ethernet

Наличие встроенного агрегирования каналов для VIOS защищает VIOS от ошибок адаптеров и сетевых концентраторов, однако не устраняет зависимости клиента от сервера. Раздел клиента все равно требует наличия уровня абстрагирования от VIOS, чего можно достичь с помощью технологий резервирования сетевого интерфейса (network interface backup), нескольких маршрутов IP с обнаружением сбояных маршрутизаторов (Dead Gateway Detection) или перехвата SEA (Shared Ethernet Adapter failover), которые будут описаны далее.

Перехват SEA является нововведением в VIOS v1.2 и предоставляет клиенту избыточность Ethernet на виртуальном уровне. Клиент получает один стандартный виртуальный Ethernet-адаптер, располагающийся на двух VIOS. Два Virtual I/O Server используют канал управления для определения того, кто из них предостав-

¹ Из контекста ясно, что имеются в виду разделы Logical Partitions (компоненты LVM).
Прим. науч. ред.

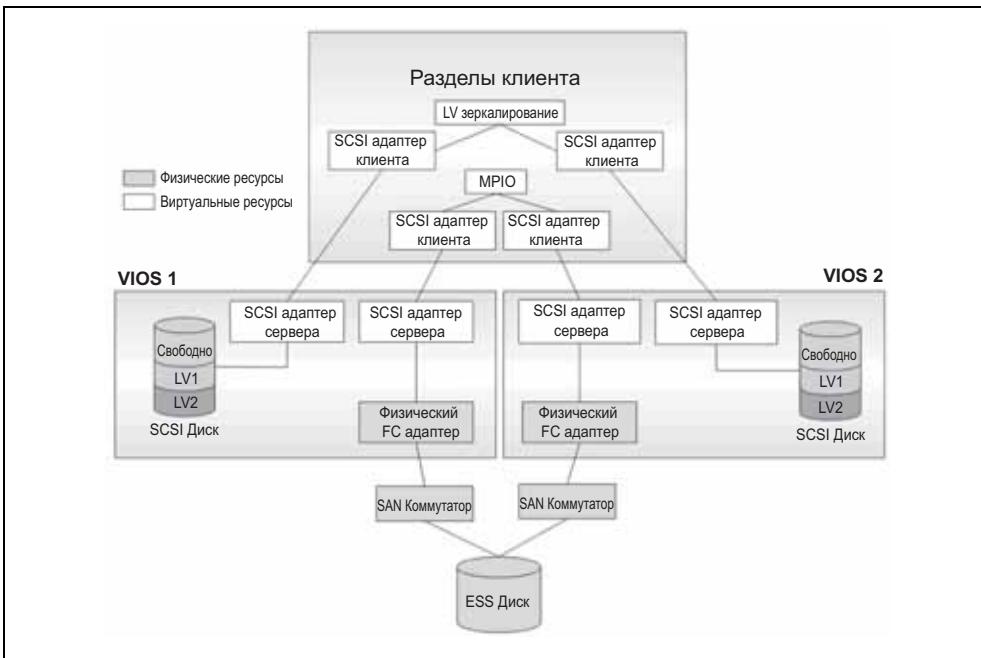


Рис. 5-1. MPIO и зеркалирование для двух VIOS

ляет Ethernet-службу клиенту. С помощью активного мониторинга двух VIOS ошибка любого из них приведет к тому, что оставшийся VIOS возьмет Ethernet-службу для клиента под свое управление. Клиент не обладает специальным протоколом или программными настройками и использует виртуальный Ethernet-адаптер так, как если бы он был размещен только на одном VIOS.

На рисунке 5-2 показана типичная настройка, комбинирующая агрегирование каналов и дублирование SEA: два Virtual I/O Server имеют настроенное агрегирование каналов через два Ethernet-адаптера для расширения полосы пропускания и большей избыточности. Настроены общие Ethernet-адAPTERЫ. Канал управления для общих Ethernet-адаптеров полностью отделен. При запуске клиентского раздела его сетевой трафик будет обслуживаться VIOS 1 с максимальным назначенным приоритетом. Если VIOS 1 недоступен, VIOS 2 определит это через канал управления и перехватит управление сетевыми службами для раздела клиента.

Примечание. Помните, что меньшее число означает больший приоритет.

Вот некоторые темы, касающиеся повышения доступности сети и производительности, которые будут подробнее обсуждаться в следующем разделе:

- ▶ Производительность и доступность общих Ethernet-адаптеров может быть увеличена с помощью агрегирования каналов или EtherChannel.
- ▶ Обсуждаются подходы к устранению ситуации, когда общие Ethernet-адаптеры становятся единой точкой отказа для доступа к сети.
- ▶ Обсуждаются детали реализации в гипервизоре POWER и его влияние на производительность.

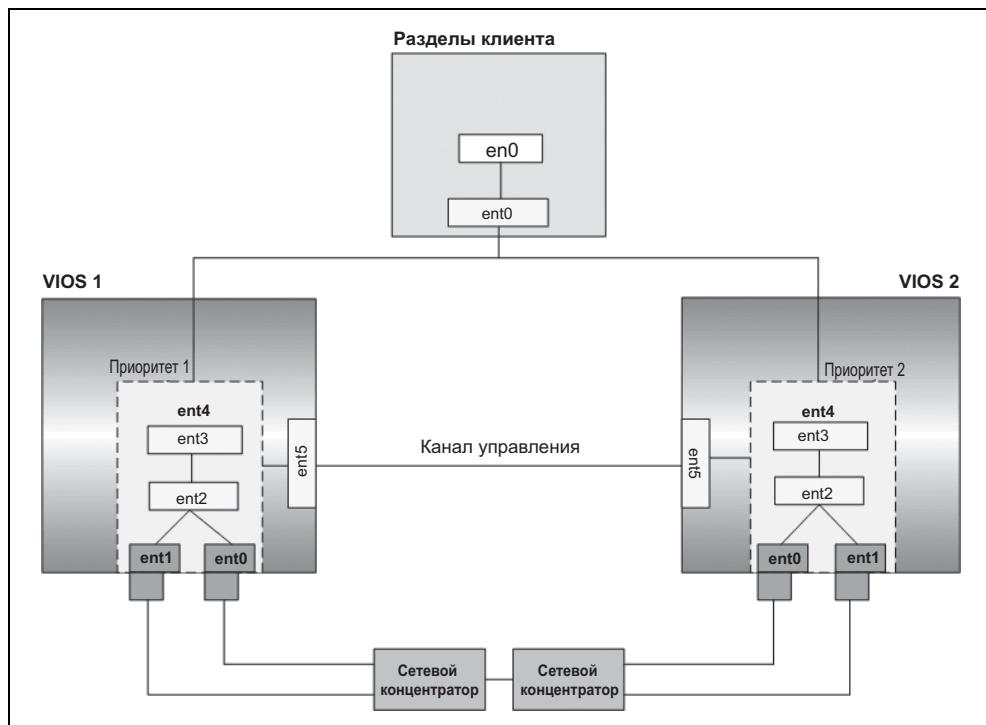


Рис. 5-2. Дублирование общего Ethernet-адаптера

Обсуждение дополнительных концепций виртуального и общего Ethernet завершается подведением итогов их преимуществ, ограничений и соглашений.

5.1.2. Использование агрегирования каналов или EthernetChannel для внешних сетей

Агрегирование каналов – это технология агрегирования сетевых портов, позволяющая нескольким Ethernet-адаптерам объединяться в один Ethernet псевдоадаптер. Эта технология обычно используется для преодоления ограничений полосы пропускания отдельных сетевых адаптеров и преодоления проблемы «бутылочного горлышка» при разделении одного сетевого адаптера между несколькими клиентскими разделами.

Основным достоинством агрегирования каналов является то, что оно обеспечивает полосу пропускания всех своих адаптеров, присутствующих в этой сети. Если адаптер возвращает ошибку, пакеты автоматически посыпаются на следующий доступный адаптер без разрыва существующего соединения с пользователем. Адаптер автоматически возвращается к работе в агрегировании каналов сразу после восстановления работоспособности. Поэтому агрегирование каналов увеличивает еще и доступность. Ошибка канала или адаптера приводит к снижению скорости, но не к обрыву соединения.

Тем не менее агрегирование каналов не является законченным решением для обеспечения высокой доступности, так как все агрегируемые каналы должны быть подключены к одному коммутатору. Это ограничение можно обойти с помощью запасного адаптера: вы можете добавить один дополнительный канал к агрегированным каналам, который будет подключен к другому Ethernet-коммутатору в той же VLAN. Этот одиночный канал может использоваться в виде резерва.

В качестве примера агрегирования каналов: ent0 и ent1 могут быть агрегированы в ent2. Система считает эти агрегированные адAPTERЫ одним адAPTERом. Интерфейс en2 будет настроен на единственный IP-адрес. Тем не менее IP будет настроен, как и для любого другого Ethernet-адAPTERа. Кроме этого все адAPTERы в агрегировании каналов получают одинаковый аппаратный (MAC) адрес, так что они будут считаться удаленными системами одним адAPTERом.

В AIX 5L поддерживаются два варианта агрегирования каналов:

- ▶ Cisco EtherChannel (EC)
- ▶ IEEE 802.3ad Link Aggregation (LA)

EC является специфической для Cisco реализацией аппаратного агрегирования, тогда как LA следует стандарту IEEE 802.3ad. Таблица 5-1 демонстрирует основные различия между EC и LA.

Таблица 5-1. Основные различия между EC- и LA-агрегированием

Cisco EtherChannel	IEEE 802.3ad Link Aggregation
Специфический для Cisco	Открытый стандарт
Требует настройки коммутатора	Может потребоваться незначительная настройка коммутатора для включения агрегирования. Может потребоваться начальная настройка коммутатора
Поддерживает различные модели распределения пакетов	Поддерживает только стандартную модель распределения пакетов

Главным достоинством применения LA является то, что если коммутатор поддерживает *Link Aggregation Control Protocol* (LACP), производить дополнительные настройки портов коммутатора не требуется. Преимущество EC заключается в поддержке различных моделей распределения пакетов. Это означает, что на сбалансированность загрузки агрегируемых адAPTERов можно влиять. В оставшейся части этой книги мы будем использовать Link Aggregation, где это только возможно, так как он считается очень универсальным термином.

Рисунок 5-3 демонстрирует агрегирование четырех плюс одного адAPTERа в единственное псевдо-Ethernet-устройство, включая резервную копию. Ethernet-адAPTERы от ent0 до ent3 агрегируются для расширения полосы пропускания и должны быть подключены к одному Ethernet-коммутатору, тогда как ent4 подключается к другому коммутатору, но используется только в качестве резервной копии, если основной Ethernet-коммутатор откажет. АдAPTERы от ent0 до ent4 теперь эксклюзивно доступны через псевдо-Ethernet-адAPTER ent5 и его интерфейс en5. Вы не можете, например, подключить сетевой интерфейс en0 к ent0, пока ent0 является членом EtherChannel или агрегирования каналов.

Примечание. Агрегирование каналов или EthernetChannel на виртуальных Ethernet-адаптерах не поддерживается. Тем не менее вы можете использовать возможность резервного сетевого интерфейса агрегирования каналов с виртуальными Ethernet-адаптерами.

Об агрегировании каналов с только одним основным Ethernet-адаптером и одним резервным адаптером говорят как о работе с резервным сетевым интерфейсом (Network Interface Backup NIB).

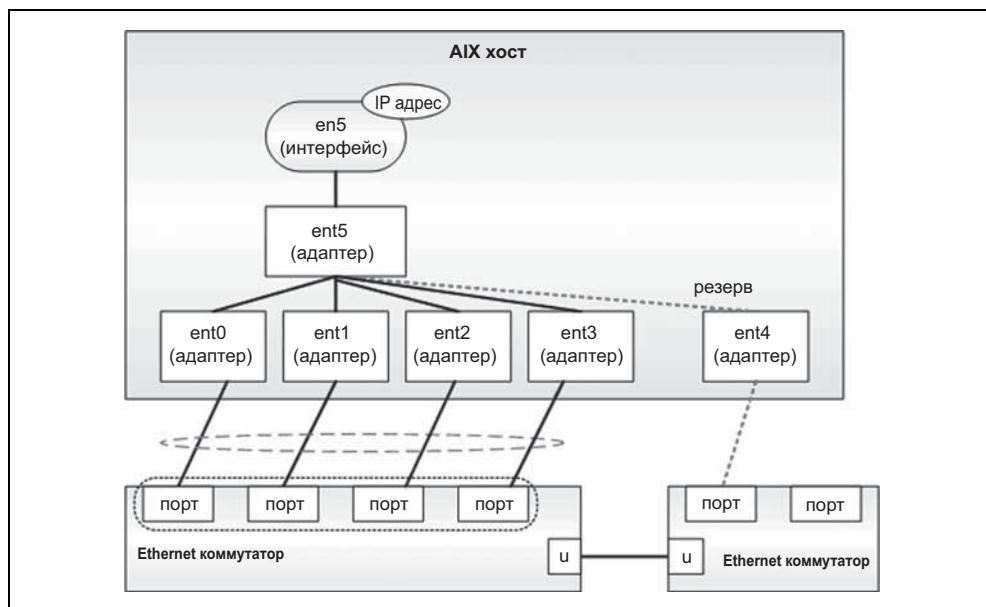


Рис. 5-3. Агрегирование каналов (EtherChannel) на AIX 5L

5.1.3. Обеспечение повышения доступности для связи с внешними сетями

При настройке одиночного Virtual I/O Server связь с внешними сетями прервется, если VIOS станет недоступным. Клиенты VIO ощутят этот обрыв, если они используют SEA для доступа к внешним сетям. Связь через SEA прервется, как только отключится VIOS, и восстановится, как только VIOS опять заработает. Внутренняя связь между разделами через виртуальное Ethernet-соединение в то время, когда VIOS недоступен, останется незатронутой. Клиенты VIO не должны перезагружаться или перенастраиваться каким-либо образом после восстановления связи через SEA. В основном, если это касается виртуального Ethernet, перезагрузка VIOS с SEA затрагивает клиентов, так же как и при отключении и переподключении связи с физическим Ethernet-коммутатором.

Если временная ошибка связи с внешними сетями недопустима, реализуется более одного экземпляра объекта, связывающего клиенты с внешним миром, и специальные функции восстановления после ошибок.

Ограничение. При использовании интегрированного менеджера виртуализации Integrated Virtualization Manager (IVM) вы можете использовать только один Virtual I/O Server и все физические устройства принадлежат серверу VIOS. Поэтому, если вы используете IVM, вы не можете настроить высокую доступность для связи с внешними сетями, как описано в этом разделе. Эта глава касается только систем, управляемых консолью управления Hardware Management Console (HMC).

Альтернативные способы повышения доступности в AIX 5L

Для физических Ethernet-адаптеров для повышения доступности связи в AIX 5L можно использовать:

- ▶ На уровне 2 (Ethernet), как показано на рисунке 5-4, при наличии резервного адаптера EthernetChannel, агрегирования каналов или резервного сетевого интерфейса (NIB).
- ▶ На уровне 3 (TCP/IP), как показано на рисунке 5-5, при многопутевом IP (IP-MP) одним из следующих способов:
 - С помощью механизма обнаружения сбойных маршрутизаторов (DGD)
 - С виртуальными IP-адресами (VIPA) и протоколами динамической маршрутизации, такими как Open Shortest path First (OSPF)
- ▶ Перехват локальных IP-адресов (IPAT) с помощью кластеров обеспечения высокой доступности или программного обеспечения для автоматизации, такого как HACMP для AIX 5L или Tivoli System Automation (TSA).

Далее приведено несколько важных соглашений, касающихся использования агрегирования каналов для виртуальных Ethernet-адаптеров вместо физических.

Ограничение. Агрегирование каналов для более чем одного виртуального Ethernet-адаптера не поддерживается. Поддерживается только один основной виртуальный Ethernet-адаптер плюс один резервный виртуальный Ethernet-адаптер. Поэтому допускается использование только двух виртуальных адаптеров, одного активного и одного ожидающего, как показано на рисунке 5-4.

Важно. При использовании NIB с виртуальными Ethernet-адаптерами обязательно использовать функцию ping-to-address, для того чтобы иметь возможность обнаруживать сетевые ошибки, так как для виртуальных Ethernet-адаптеров не существует аналога ошибки канала, активизирующей резервный адаптер.

Тем не менее вы можете иметь несколько активных физических Ethernet-адаптеров с агрегированными каналами плюс один резервный виртуальный Ethernet-адаптер.

Ограничение. При настройке NIB для двух виртуальных Ethernet-адаптеров две используемые внутренние сети должны быть разделены в гипервизоре POWER, и поэтому вы должны использовать два различных PVID для двух адаптеров в клиенте и не можете использовать на них дополнительные VID. Далее для двух отдельных внутренних VLAN создается соединение типа мост (bridge) на одну внешнюю VLAN.

Многопутевой IP с протоколом DGD может сделать более доступным только исходящее соединение, тогда как использование VIPA с применением протоколов динамической маршрутизации позволяет сделать более доступными как входящие, так и исходящие соединения. Реализация протоколов динамической маршрутизации может быть достаточно сложной, так как в их реализации должно участвовать сетевое оборудование.

Примечание. VIPA и OSFP используются вместе с IBM *@server* zSeries and IBM System z9 Geographically Dispersed Parallel Sysplex™ (GDPS®). Поэтому, если вы хотите повысить доступность общего Ethernet на System p5 в окружении System z9, вам следует учесть, что внешние сетевые устройства уже могут быть настроены для работы с OSPF.

Примечание. Обсуждение DGD, VIPS, OSFP и GDPS выходит за границы рассмотрения данной книги.

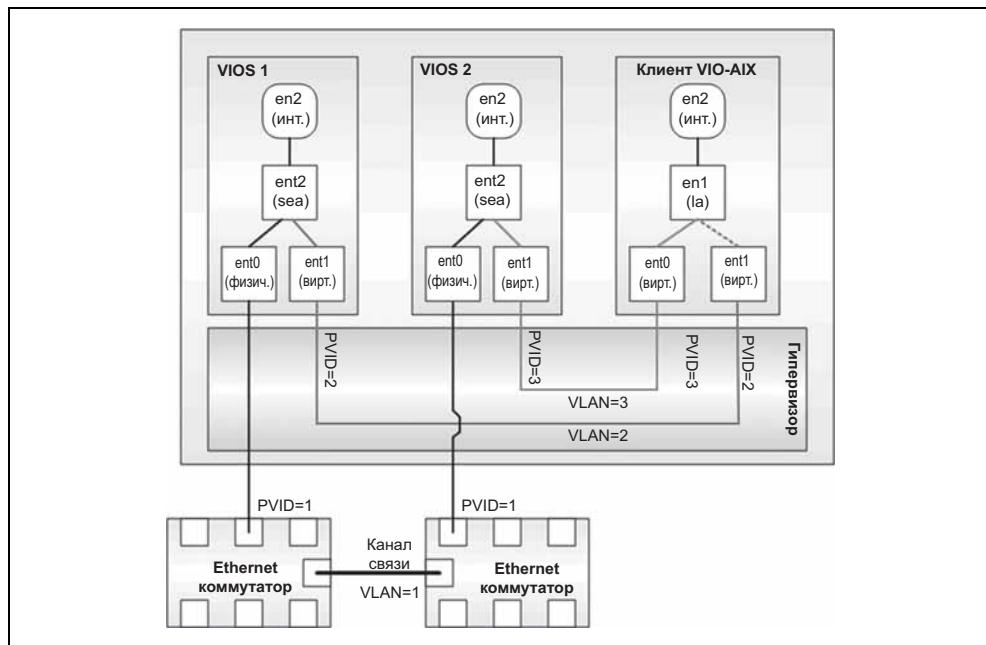


Рис. 5-4. Резервирование сетевого интерфейса (NIB) с двумя VIOS

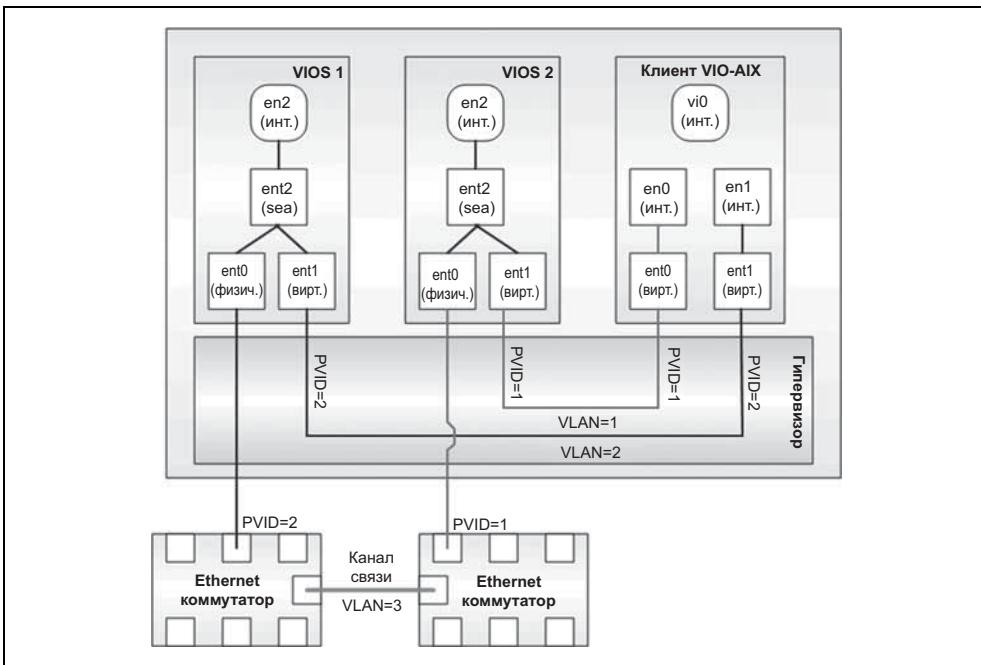


Рис. 5-5. Многопутевой IP в клиенте, использующий два SEA различных VIOS

Где реализовывать высокую доступность сети – на клиенте или сервере?

Выше описаны различные варианты, такие как NIB, IPMP и IPAT, предоставляемые AIX 5L и доступные для использования разделом – клиентом общего Ethernet для повышения доступности при связи с внешними сетями. Наиболее распространенные подходы предполагают, что для клиентских разделов требуется несколько виртуальных Ethernet-адаптеров и логика перехвата реализуется на этих клиентских разделах, что, в свою очередь, усложняет настройку разделов клиента.

Следующие два раздела этой книги описывают два подхода к повышению доступности для доступа к внешним сетям (перехват ошибок маршрутизатора и перехват общих Ethernet-адаптеров), которые обычно не предполагают, что логика перехвата реализована на разделах клиента, и поэтому упрощают настройку клиента.

Перехват ошибок маршрутизатора

При использовании маршрутизации (пересылка на уровне 3) вместо соединения типа мост (пересылка на уровне 2) для соединения внутренних сетей (сетей между разделами) с внешними сетями возможно использование двух разделов-маршрутизаторов и IP-перехвата (IPAT), настроенного между этими разделами-маршрутзаторами для повышения доступности подключения к внешним сетям. Далее разделы клиента будут использовать высокодоступный IP-адрес в качестве маршрута по умолчанию, что упрощает настройку для этих разделов и концентрирует всю сложность восстановления после ошибок внутри разделов-маршрутизаторов. Этот подход показан на рисунке 5-6.

Для реализации IPAT в разделах-маршрутизаторах требуется дополнительное программное обеспечение, такое как HACMP для AIX 5L, Tivoli System Automation для AIX 5L или Linux, Heartbeat для Linux, или реализация протокола Virtual Router Redundancy Protocol (VRRP) на Linux например, подобные подходы описаны в Linux on IBM **@server** zSeries and S/390: Virtual Router Redundancy Protocol on VM Guest LANs, REDP-3657.

Примечание. Обсуждение HACMP, TSA, Heartbeat и VRRP выходит за границы рассмотрения данной книги.

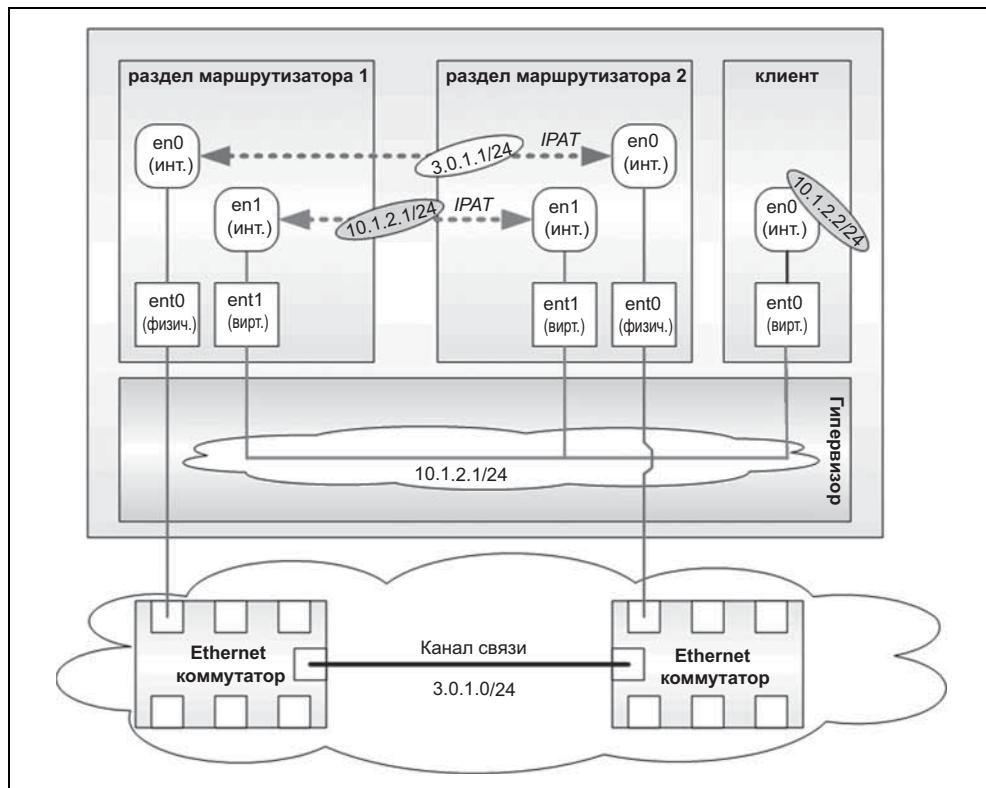


Рис. 5-6. Перехват маршрутизатора

Перехват общих Ethernet-адаптеров

Начиная с Virtual I/O Server V 1.2 существует очень прямолинейное решение для повышения доступности общего Ethernet – *Перехват общих Ethernet-адаптеров* (SEA Failover). Перехват SEA реализуется на VIOS, а не на клиенте и поддерживает упрощенный мостовой (уровня 2) подход для доступа к внешним сетям. Перехват SEA поддерживает IEEE 802.1Q VLAN-маркировку в качестве основной особенности SEA.

Перехват SEA работает следующим образом: два VIOS образуют функциональность моста SEA с автоматическим восстановлением после сбоев (перехватом), если один VIOS отказывает, отключается или SEA теряет доступ к внешней сети через физический Ethernet-адаптер. Вы также можете инициировать перехват вручную.

Как показано на рисунке 5-7, оба VIOS подключаются к одной виртуальной и физической Ethernet-сети и VLAN, и оба виртуальных Ethernet-адаптера обеих SEA будут иметь установленный *флаг доступа к внешним сетям (access to external network)*, называемый в предыдущих версиях *флагом транка (trunk)*. Дополнительное виртуальное Ethernet-соединение настраивается как отдельная VLAN между двумя VIOS и должно быть подключено к общим Ethernet-адаптерам (SEA) в виде *канала управления (control channel)*, а не как обычный член SEA. Эта VLAN служит каналом для обмена keep-alive или heartbeat-сообщениями между двумя VIOS и поэтому управляет перехватом функциональности моста. К Ethernet-адаптерам управляющего канала не должно быть подключено никаких сетевых интерфейсов; адаптер управляющего канала должен быть выделенным и должен находиться в выделенной VLAN, которая не используется ни для чего другого.

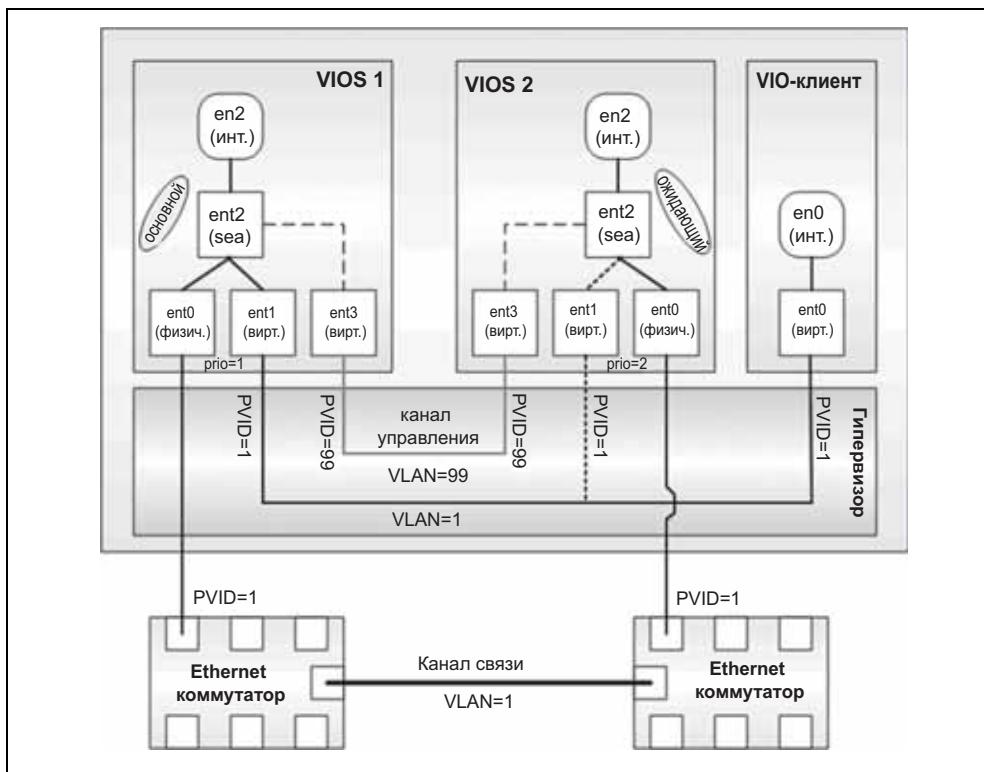


Рис. 5-7. Базовая настройка SEA

Вам нужно выбрать разные приоритеты для двух SEA с помощью установки этого приоритета для всех виртуальных Ethernet-адаптеров каждого SEA. Значение приоритета определяет, какой из двух SEA будет основным (активным) и какой

будет резервным (ожидающим). Чем меньше значение приоритета, тем больше приоритет, поэтому приоритет = 1 означает наивысший приоритет.

SEA может быть настроен таким образом, чтобы он периодически пытался посылать ping-запросы на указанный IP-адрес для проверки того, что соединение доступно. Это похоже на IP-адрес для проверки через ping, который может быть настроен в Network Interface Backup.

Существуют четыре ситуации, приводящие к активизации восстановления после ошибок SEA:

1. Ожидавший SEA определяет, что сообщения keep-alive от активного SEA больше не поступают по каналу управления.
2. Активный SEA определяет потерю физического канала из сообщения драйвера устройства физического Ethernet-адаптера.
3. На VIOS с активным SEA ручной перехват может быть инициирован с помощью установки активного SEA в режим ожидания.
4. Активный SEA определяет, что он больше не может «пинговать» заданный IP-адрес.

Остановка поступления сообщений keep-alive происходит, когда VIOS с активным SEA отключается или зависает, прекращает отвечать или деактивируется с HMC.

Существует несколько типов ошибок, которые не активизируют перехват SEA, так как сообщения keep-alive пересыпаются только по каналу управления. Через сеть SEA сообщения keep-alive не пересыпаются, по крайней мере не через внешнюю сеть. Однако возможность перехвата SEA может быть настроена на периодическое «пингование» IP-адреса, так как через него можно определить некоторые сетевые ошибки. Вы наверняка знаете эту особенность NIB AIX 5L.

Важно. Общие Ethernet-адAPTERы должны иметь сетевые интерфейсы с назначенными IP-адресами, которые можно использовать для периодического теста доступности.

Эти IP-адреса должны быть уникальными и вам следует использовать различные IP-адреса для обеих SEA.

SEA должен иметь IP-адреса для предоставления обратного адреса при отсылке пакетов ICMP-Echo-Request и приеме пакетов ICMP-Echo-Reply, получаемых при пинговании данных IP-адресов. Эти IP-адреса должны быть разными.

Поэтому, так как альтернативная конфигурация для восстановления SEA после ошибок на рисунке 5-8 будет практически эквивалентом той, что приведена на рисунке 5-7, вы не можете заставить SEA периодически пинговать указанный IP-адрес. Мы рекомендуем связать интерфейс и IP-адрес с SEA, как показано на рисунке 5-7.

Преимущество технологии перехвата SEA (SEA Failover) по сравнению с резервированием сетевого интерфейса (Network Interface Backup), помимо прочего, заключается в поддержке маркировки VLAN и упрощении настройки клиента Virtual I/O. Это возможно благодаря тому, что присутствует только один Ethernet-адаптер, один маршрутизатор по умолчанию и никакой логики поддержки восстановления после ошибок в клиенте Virtual I/O. Вся избыточность и касающиеся повышения

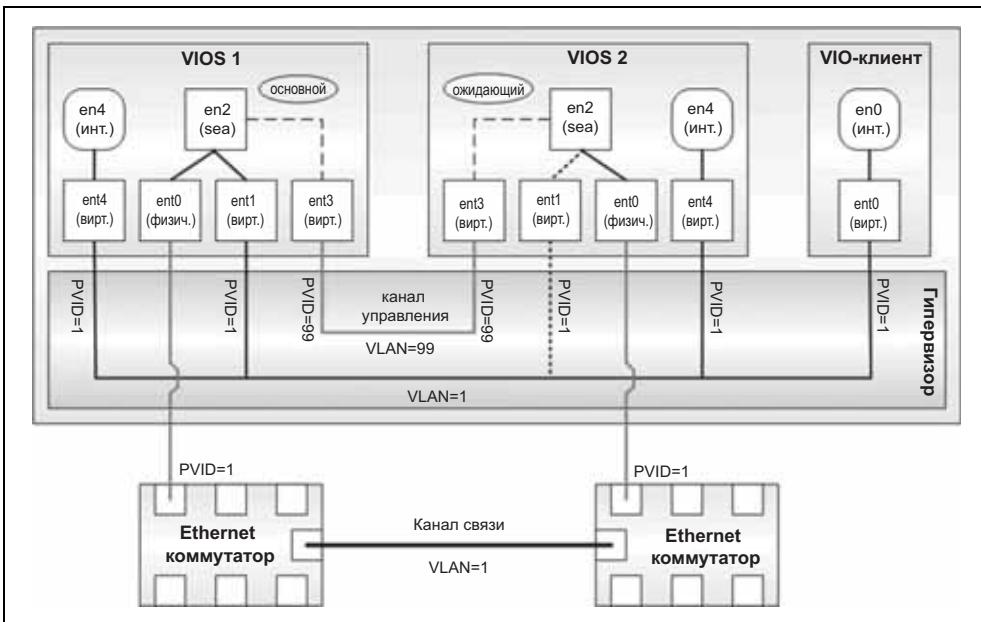


Рис. 5-8. Альтернативная настройка восстановления SEA после ошибок

шения доступности функции реализуются в VIOS. Это означает, что виртуализация имеет потенциал для более гибкого размещения и лучшего использования ресурсов, а также для упрощения и разделения интересов.

В следующем разделе более подробно обсуждаются архитектурные соглашения.

Когда следует использовать перехват SEA или резервирование сетевого интерфейса

В предыдущем разделе мы описывали два различных подхода повышения доступности для подключения через мост (уровень 2) к внешним сетям с помощью избыточного общего Ethernet-адаптера для двух серверов Virtual I/O Server:

- ▶ Резервирование сетевого интерфейса (NIB) (см. рисунок 5-4 и рисунок 5-9)
- ▶ Перехват общих Ethernet-адаптеров (SEA Failover) (см. рисунок 5-7)

Мы сравним эти два подхода:

- ▶ Перехват SEA реализуется на Virtual I/O Server, тогда как NIB реализуется на клиенте.
- ▶ Перехват SEA упрощает настройку нескольких клиентов.

Перехват SEA упрощает настройку клиентов, так как для этого нужно иметь только один адаптер и VLAN и никакой логики восстановления после ошибок.

При использовании подхода с NIB все клиенты должны иметь второй виртуальный Ethernet-адаптер, настроенный на другую VLAN, и адаптер агрегирования каналов с поддержкой NIB. Это значительно усложняет настройку клиента.

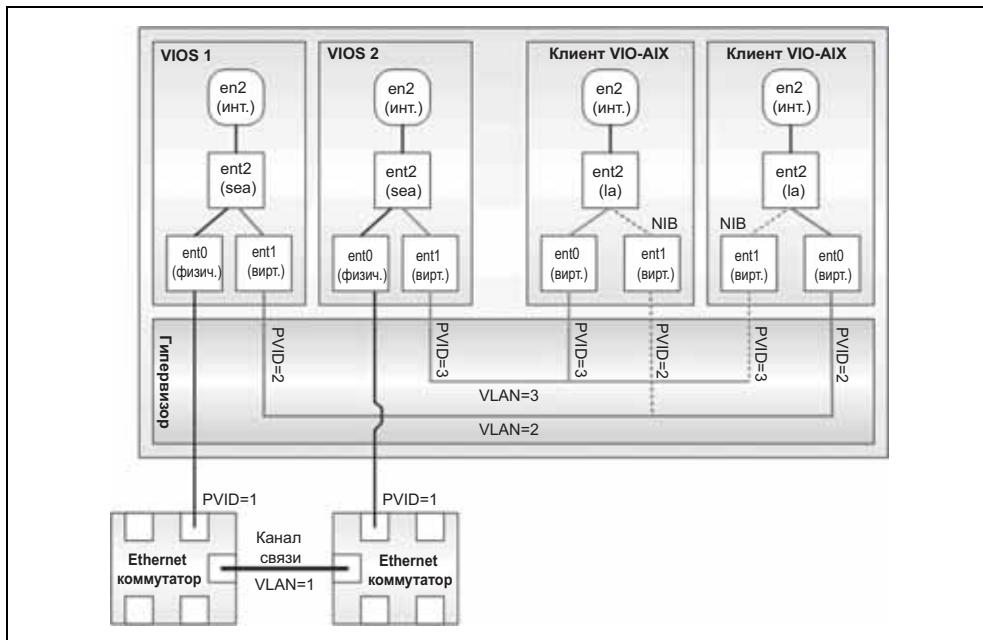


Рис. 5-9. Резервирование сетевого интерфейса (NIB) для нескольких клиентов

- Перехват SEA можно использовать совместно с IEEE802.1Q VLAN-метками, а NIB – нет.

При использовании NIB две используемые внутренние сети должны быть разделены в гипервизоре POWER, так что вам нужно использовать два различных PVID для двух адаптеров на клиенте и вы не можете использовать на нем дополнительные VID. Далее между двумя разными внутренними VLAN настраивается мост во внешнюю VLAN.

NIB позволяет более оптимально использовать доступные ресурсы:

- При использовании перехвата SEA в каждый момент времени активно используется только один SEA, а другой простояивает. Поэтому пропускная способность физического адаптера ожидающего SEA не используется.
- При использовании NIB вы можете распределять клиенты между обоими SEA таким образом, что одна половина в качестве основного адаптера использует первый SEA, а другая половина использует второй SEA, как показано на рисунке 5-9. Поэтому используется пропускная способность физических Ethernet-адаптеров обоих SEA.

В большинстве случаев преимущества перехвата SEA перевешивают достоинства NIB, так что для повышения доступности мостового доступа к внешним сетям обычно используется перехват SEA.

Примечание. Так как перехват SEA является нововведением в APV Virtual I/O Server V 1.2, в более ранних версиях для повышения доступности мостового доступа к внешним сетям доступен только NIB-подход.

Ограничение. Как в перехвате SEA, так и в NIB существует несколько ограничений: они не выполняют проверки достижимости указанного IP-адреса через резервный путь, пока доступен основной путь. Поэтому вы не можете узнать, есть ли у вас в наличии работающий резервный канал на случай отказа основного.

Резюме касательно способов обеспечения высокой доступности для подключения к внешним сетям

В таблице 5-2 подытожены альтернативные подходы к повышению доступности общего доступа к внешним сетям, рассмотренные в предыдущем разделе.

Таблица 5-2. Резюме касательно способов повышения доступности для доступа к внешним сетям

	Реализация на сервере	Реализация на клиенте
Уровень 2/ мост	Перехват SEA	NIB
Уровень 3/ маршрутизатор	Перехват маршрутизатора	IPMP, VIPA и IPAT

Среди этих подходов наиболее подходящим для типичного сценария виртуализации IMB System p5 является перехват SEA, так что в оставшейся части книги мы акцентируем свое внимание на перехвате SEA. Если вы собираетесь реализовывать один из альтернативных подходов, вам следует обратиться к соответствующим публикациям.

5.1.4. Управление системой на Virtual I/O Server

Избыточность облегчает управление системой, тогда как единые точки отказа часто приводят к усложнению администрирования. При использовании двух Virtual I/O Server исключается слой физической зависимости от ресурсов.

Поддержка системы может выполняться на виртуальном сервере ввода-вывода или внешнем устройстве, к которому он подключен через сеть или SAN-коммутатор. При использовании Virtual SCSI и общего Ethernet, размещенного на втором Virtual I/O Server, перезагрузка или отключение VIOS от внешних устройств возможны без обрыва соединения с клиентом. Если раздел клиента работает на MPIO и использует перехват SEA, во время обслуживания системы и после ее завершения на разделе клиента не нужно выполнять никаких действий. Это улучшает время доступности клиента и уменьшает необходимость в администрировании раздела клиента.

Запуск и перезагрузка VIOS, сетевого коммутатора или SAN-коммутатора упрощены и обособлены, так как клиент больше не зависит от доступности всего окружения.

На рисунке 5-10 раздел клиента имеет Virtual SCSI-устройства и виртуальный Ethernet-адаптер, размещенные на двух Virtual I/O Server. Клиент имеет MPIO для виртуальных SCSI-устройств и SEA Failover для виртуального Ethernet. Во время выключения VIOS 2 для проведения обслуживания раздел клиента продолжает использовать доступ к сети и SAN-хранилищу через VIOS 1.

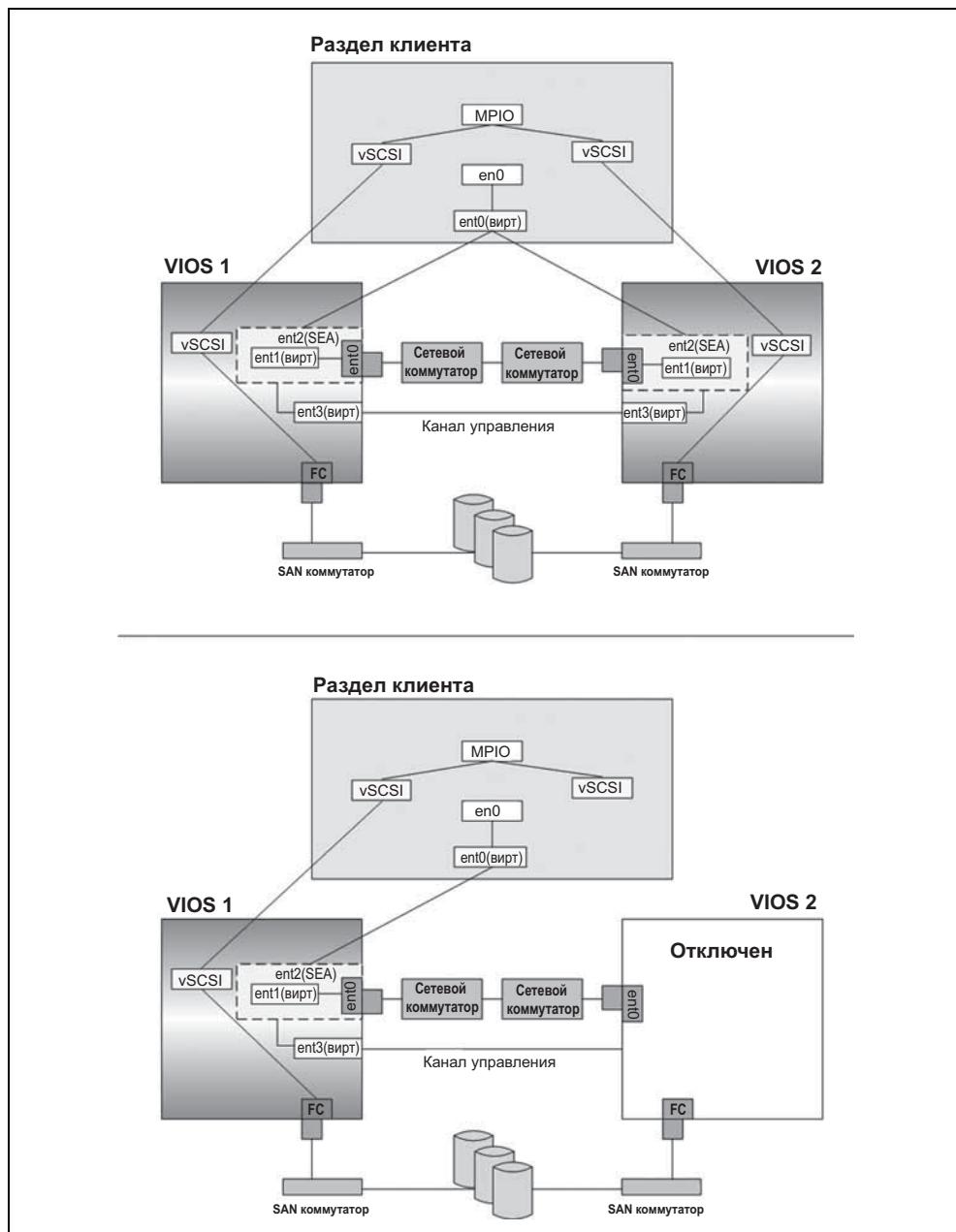


Рис. 5-10. Избыточные виртуальные серверы ввода-вывода во время проведения обслуживания

Как только VIOS 2 возвращается к работе, клиент продолжает использовать путь MPIO через VIOS 1, а виртуальный Ethernet переключается на основной общий Ethernet-адаптер VIOS

5.1.5. Реализация виртуального Ethernet в гипервизоре POWER

Виртуальное Ethernet-соединение использует технологию VLAN для обеспечения доступности из раздела только к данным, которые к ним относятся. Гипервизор POWER предоставляет функцию виртуального Ethernet-коммутатора на основе стандарта IEEE 802.1Q VLAN, который позволяет разделам общаться в пределах одного сервера. Соединение базируется на внутренней реализации гипервизора POWER, который перемещает данные между разделами. Эта часть книги описывает различные элементы виртуального Ethernet и связанные с ним различные типы работ. Рисунок 5-11 упрощенно иллюстрирует основные черты сети между разделами на уровне взаимодействия драйверов устройств и гипервизора.

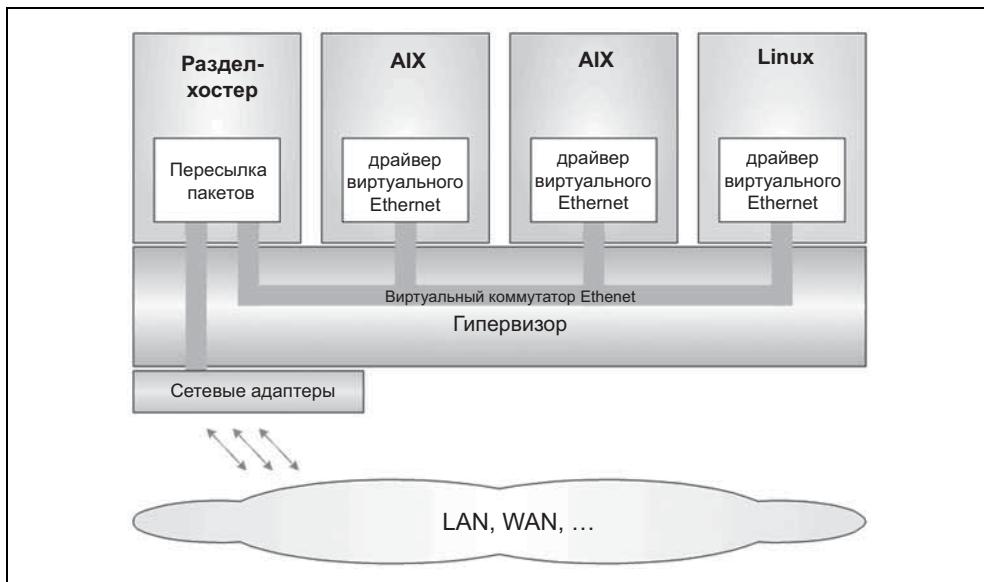


Рис. 5-11. Логический вид VLAN между разделами

Создание виртуального Ethernet-адаптера

Разделы, которые связаны через виртуальный Ethernet, имеют дополнительные каналы в памяти, реализованные в гипервизоре:

- Создание канала в памяти между разделами происходит автоматически при настройке виртуального Ethernet-адаптера для разделов на НМС или ИВМ.
- Ядро AIX 5L или Linux автоматически создает виртуальное сетевое устройство для каждого канала в памяти, обозначенного внутренним кодом (firmware) POWER5.

- ▶ Менеджер настройки (configuration manager) AIX 5L создает необходимые ODM объекты для:
 - Устройства Ethernet сетевого адаптера (ent*) в доступном состоянии (available)
 - Устройства Ethernet сетевого интерфейса (en* и et*) в определенном состоянии (defined)

Уникальный 6-байтовый адрес Media Access Control (MAC) (также называемый адресом Ethernet, аппаратным адресом или адресом уровня 2) генерируется при создании виртуального Ethernet-устройства на HMC или IVM. Системе может быть назначен префикс, так что она будет автоматически генерировать MAC адреса в системе, содержащие указанный системный префикс плюс генерируемую алгоритмом уникальную часть для каждого адаптера. Поэтому сгенерированные адреса не будут конфликтовать с адресами для других сетевых устройств.

Виртуальный Ethernet можно также использовать в качестве загрузочного устройства, что позволяет выполнять задачи наподобие инсталляции операционной системы через NIM.

Динамические операции с разделами для виртуальных Ethernet-устройств

Виртуальные Ethernet-ресурсы можно назначать и удалять динамически через динамические LPAR-операции. На HMC или IVM виртуальные Ethernet-адAPTERЫ назначения и сервера могут добавляться и удаляться из раздела при помощи динамических логических разделов. Создание физических и виртуальных Ethernet-адAPTERов на Virtual I/O Server может быть выполнено динамически. После добавления адAPTERа на HMC, виртуального ли физического, нужно запустить команду cfgmgr на разделе AIX 5L и команду cfgdev на Virtual I/O Server.

5.1.6. Замечания о производительности Virtual I/O Server

Скорость передачи виртуального Ethernet-адAPTERа находится в пределах нескольких гигабит в секунду, в зависимости от размера передачи (MTU) и общей производительности системы. Виртуальное Ethernet-соединение обычно требует больше циклов процессора, чем соединение через физические Ethernet-адAPTERы. Это происходит потому, что современные физические Ethernet-адAPTERы содержат множество функций, разгружающих центральный процессор от некоторых работ, например вычисления контрольных сумм и верификации, модуляции по прерываниям и переборки пакетов. Эти адAPTERы используют прямой доступ к памяти (DMA) для передачи данных между адAPTERом и оперативной памятью, т.е. используют всего несколько тактов центрального процессора для настройки и совсем не используют их для передачи.

Для разделов общего процессорного пула производительность будет ограничена настройками раздела (например, назначенная мощность и количество процессоров). Малые разделы благодаря тому, что на них приходится меньше времени центрального процессора, при взаимодействии друг с другом будут показывать большую задержку из-за благодаря частого переключению контекстов.

Для разделов с выделенными процессорами пропускная способность будет сравнима с гигабитным Ethernet для малых пакетов и значительно лучшей для больших пакетов. Для больших пакетов виртуальное Ethernet-соединение ограничено скоростью полосы пропускания копирования из памяти.

Совет. Обычно приложения, требующие широкой полосы пропускания, не размещают на малых разделах общего процессорного пула. Для приложений, требующих широкой полосы пропускания, целесообразнее использовать физические адAPTERы.

Несколько серверов Virtual I/O Server также имеют определенные соглашения о производительности, особенно большие установки. Если множество разделов клиентов используют и общий Ethernet и Virtual SCSI, загрузка на VIOS становится излишней.

Совет. Принято использовать различные VIOS для разделения конкурирующих работ, таких как чувствительные к сетевым задержкам приложения и приложения с интенсивными операциями ввода-вывода для обеспечения этих работ соответствующими ресурсами. В подобном окружении вам следует отдельить виртуальный Ethernet от Virtual SCSI и разместить их на различных VIOS.

Для Virtual I/O Server, единолично выделенного на Virtual SCSI и отдельно выделенного разделяемого Ethernet, идея реализации драйверов устройств становится очевидной. При планировании избыточности наличие пары для каждого (устройства) повышает отказоустойчивость окружения и упрощает администрирование каждого Virtual I/O Server для улучшения обслуживания.

На рисунке 5-12 один раздел клиента обслуживается четырьмя Virtual I/O Server: два Virtual SCSI и два виртуальных Ethernet. Изоляция каждого типа ввода-вывода гарантирует, что Ethernet не будет испытывать нужду в ресурсах со стороны SCSI и наоборот.

Несмотря на то что данный сценарий покрывает требования к VIOS, для увеличения полосы пропускания между сервером VIOS и ресурсами потребуются дополнительные адAPTERы оптоволоконной связи и Ethernet-адAPTERы. MPIO будет использоваться на VIOS 1 и 2 для создания дополнительных оптоволоконных соединений для SAN, а агрегирование каналов будет использоваться Ethernet-адAPTERами для повышения полосы пропускания сети.

При создании по этому методу Virtual I/O Server и клиентских разделов можно использовать множество различных сценариев. Единственное ограничение на количество поддерживаемых конфигураций – это максимальное количество поддерживаемых клиентов и максимальное количество слотов ввода-вывода на машине. Группировка разделов клиента по требованиям к вводу-выводу и дальнейшее планирование Virtual I/O Server вокруг этих групп являются хорошей отправной точкой при начальной настройке новой большой машины.

5.1.7. Достоинства виртуального Ethernet и общих Ethernet-адAPTERов

Благодаря тому что на большинстве машин количество разделов может быть большим, чем количество слотов ввода-вывода, виртуальный Ethernet и общие Ethernet-адAPTERы (SEA) предоставляют большие возможности для настройки и экономии:

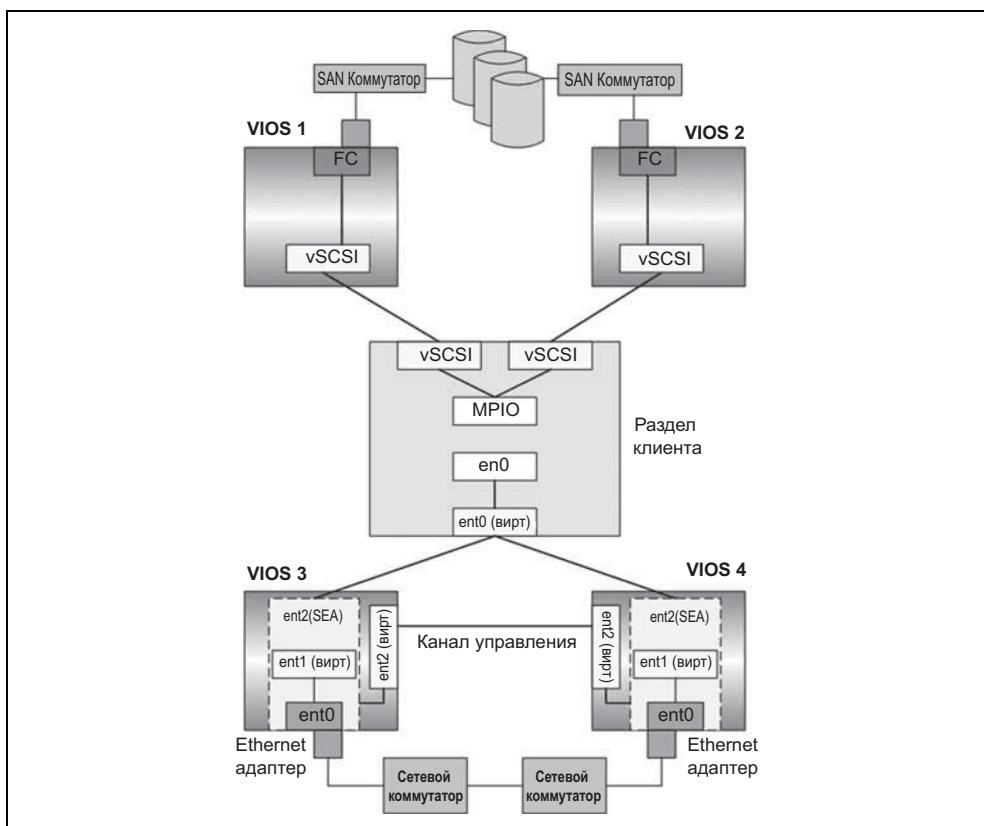


Рис. 5-12. Отдельные серверы виртуального ввода-вывода для каждого ресурса

- ▶ Обеспечения разделов внутри одной системы высокоскоростной связью друг с другом.
- ▶ Объединения доступа к физическим сетям.
- ▶ Поддержание на надлежащем уровне безопасности благодаря изоляции с использованием VLAN.

VLAN создает логические Ethernet-соединения между разделами и предназначена для предотвращения влияния ошибок или потери работоспособности операционной системы на связь между остальными функционирующими операционными системами. Виртуальное Ethernet-соединение может быть также *мостовым* или *маршрутизируемым* по отношению к внешним сетям, для того чтобы разрешить разделам работать с внешними сетями без использования физических сетевых адаптеров.

Реализация повышения доступности через перехват маршрутизатора или перехват общих Ethernet-адаптеров упрощают настройку клиента VIO.

5.1.8. Ограничения и соглашения

При реализации виртуальных Ethernet и общих Ethernet-адаптеров на Virtual I/O Server необходимо учитывать следующие ограничения:

- ▶ Виртуальный Ethernet требует систему POWER5 и HMC или IVM для определения виртуальных Ethernet-адаптеров.
- ▶ Виртуальный Ethernet доступен для всех систем POWER5, а общие Ethernet-адAPTERЫ и Virtual I/O Server могут потребовать закупки дополнительного оборудования для некоторых моделей.
- ▶ Виртуальный Ethernet может использоваться для разделов с разделяемым и выделенным процессорами.
- ▶ Для каждого раздела может быть определено максимально 256 виртуальных Ethernet-адаптеров.
- ▶ Каждый виртуальный Ethernet-адаптер может быть ассоциирован максимум с 21 VLAN (20 VID и 1 PVID).
- ▶ Система может поддерживать до 4096 различных VLAN, как определено в стандарте IEEE802.1Q.
- ▶ Раздел должен работать на AIX 5L версии 5.3 или Linux с ядром 2.6 или ядром, поддерживающим виртуальный Ethernet.
- ▶ Внутри раздела разрешается смешивать виртуальные Ethernet-соединения с реальными сетевыми адаптерами.
- ▶ Виртуальный Ethernet может соединять разделы только внутри одной системы.
- ▶ Поддерживается виртуальное Ethernet-соединение между разделами AIX 5L и Linux.
- ▶ Виртуальное Ethernet-соединение между разделом AIX 5L или Linux с разделом i5/OS может работать; однако на момент написания книги эта возможность еще не поддерживалась.
- ▶ Виртуальный Ethernet использует процессор системы для всех операций связи, в то время как сетевые адAPTERЫ разгружают процессор от большинства подобных нагрузок. В результате виртуальный Ethernet значительно сильнее загружает процессор системы.
- ▶ Разделяемому Ethernet-адаптеру может быть назначено до 16 виртуальных Ethernet-адаптеров с 21 VLAN (20 VID и 1 PVID), если это один физический сетевой адаптер.
- ▶ Число разделов, которые можно подсоединить к VLAN, не ограничено. На практике сетевой трафик ограничивает количество клиентов, которое может обслужить отдельный адаптер.
- ▶ Для повышения доступности виртуального Ethernet-соединения с внешними сетями реализуются два Virtual I/O Server с перехватом общего Ethernet-адаптера или другой механизм обеспечения высокой доступности.
- ▶ Вы не можете использовать перехват SEA с интегрированным менеджером виртуализации (IVM), так как IVM поддерживает только один виртуальный сервер ввода-вывода.

5.2. Сценарий 1: Зеркалирование логического тома

В этом сценарии мы модифицируем нашу базовую конфигурацию и обратим внимание на второй сервер ввода-вывода с именем VIO_Server2, который будет предоставлять дополнительные диски для разделов нашего клиента. Эта настройка обеспечит доступность rootvg через зеркалирование логического тома на стороне клиента. См. иллюстрацию настройки зеркалирования логического тома на рисунке 5-13.

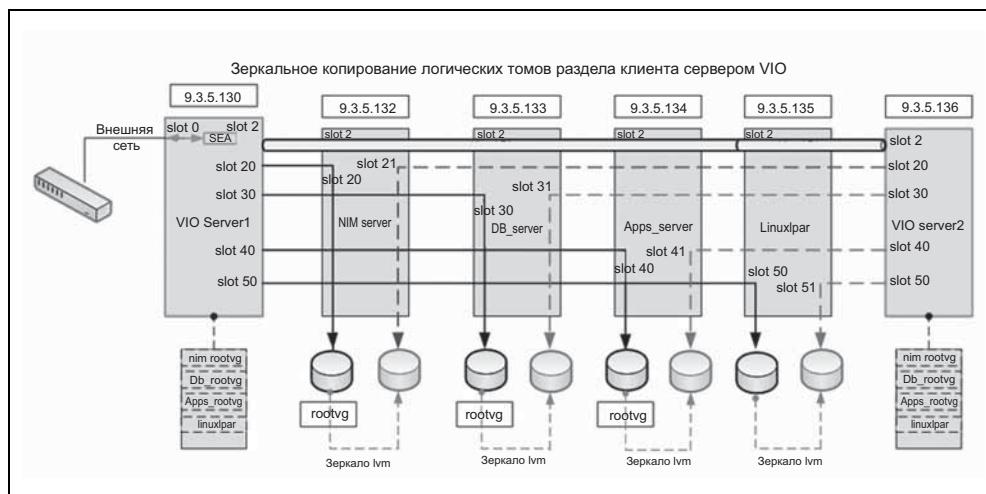


Рис. 5-13. Сценарий зеркалирования LVM

В основном шаги, необходимые для создания второго Virtual I/O Server, который будет служить дополнительным hdisk для раздела клиента, совпадают с шагами по созданию первого. Следуйте шагам, изложенным ниже, для определения решения LVM:

1. Следуйте инструкциям из раздела 4.2.1 «Определение раздела Virtual I/O Server» для создания вашего второго Virtual I/O Server, за исключением следующего:
 - a. Используйте имя VIO_Server2 на шагах 3 и 5.
 - b. На шаге 10 выделите неиспользуемой слот Ethernet-адаптера и другой слот контроллера хранения для предоставления нашему Virtual I/O Server физических устройств. В нашем случае мы выделили слот шины 2 C4 и слот шины 3 T10 контроллерам Ethernet и хранения, как показано на рисунке 5-14.
2. Следуйте инструкциям в 4.3 «Установка программного обеспечения для Virtual I/O Server» для установки программного обеспечения для Virtual I/O Server для раздела VIO_Server2.
3. Создайте Virtual SCSI-адаптер для раздела VIO_Server2, который будет хранить устройства логических томов, разделяемых разделами клиента. Обратитесь к описанию создания Virtual SCSI-адаптеров в 4.4.2 «Создание серверных адаптеров Virtual SCSI».

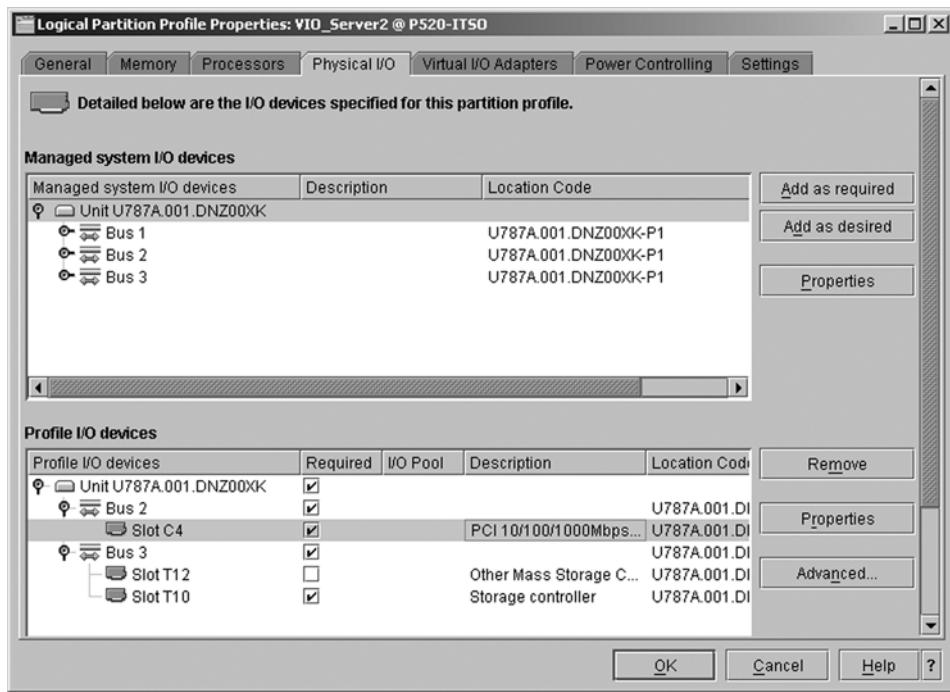


Рис. 5-14. Выбор физических компонент для VIO_Server2

4. Руководствуясь рисунком 5-13 создайте клиентские адаптеры Virtual SCSI (на разделах NIM_server, DB_server, Apps_server и linuxlpar), которые будут отображаться на Virtual SCSI-слоты на VIO_Server2. Обратитесь к инструкции по созданию Virtual SCSI-адаптеров в 4.4.2 «Создание серверных адаптеров Virtual SCSI»¹.
5. На рисунке 5-15 показано как будет выглядеть отображение для раздела VIO_Server2 с HMC.
6. Теперь вы готовы к созданию группы томов и логических томов на втором Virtual I/O Server. Более подробную информацию вы можете найти в 4.4.6 «Определение группы томов и логических томов».
7. Проверьте список устройств на вашем разделе VIO_Server2 с помощью команды `lsdev`, как показано в примере 5-1.

¹ Здесь ошибка – эту информацию можно найти в разделе 4.4.5 «Создание виртуальных SCSI-адаптеров для разделов клиентов». Прим. науч. ред.

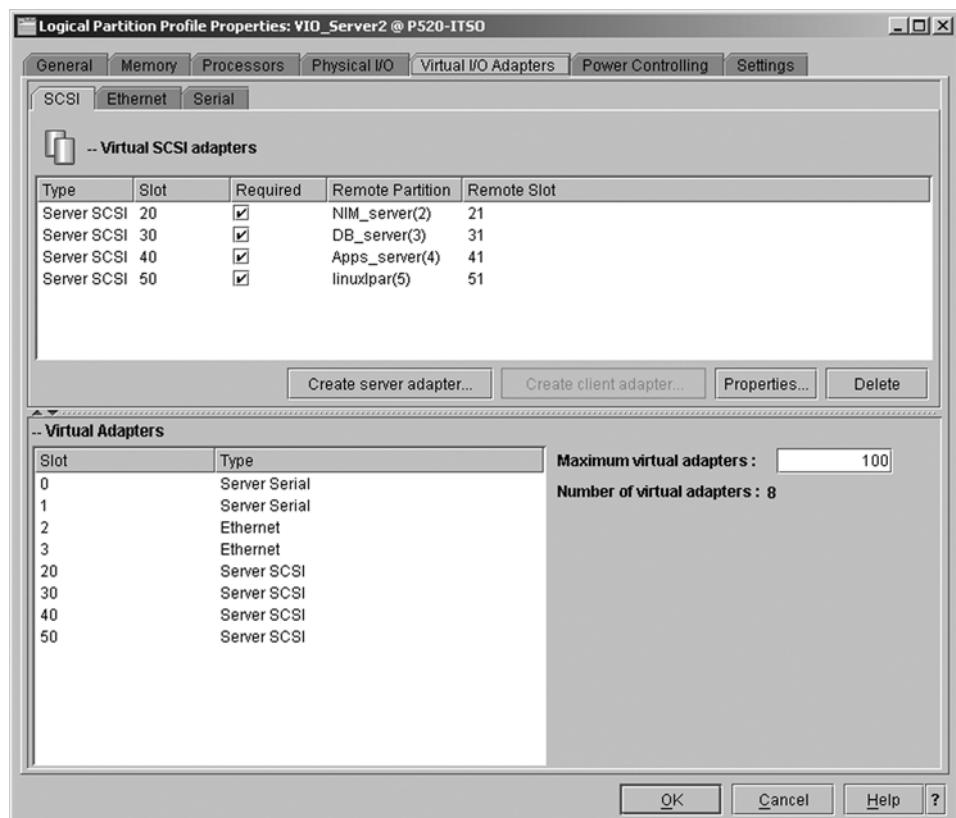


Рис. 5-15. Свойства Virtual SCSI-раздела VIO_Server2

Пример 5-1. Раздел VIO_Server2

```
$ lsdev -virtual
name      status      description
ent0      Available   Virtual I/O Ethernet Adapter (l-lan)
ent1      Available   Virtual I/O Ethernet Adapter (l-lan)
vhost0    Available   Virtual SCSI Server Adapter
vhost1    Available   Virtual SCSI Server Adapter
vhost2    Available   Virtual SCSI Server Adapter
vhost3    Available   Virtual SCSI Server Adapter
vsa0      Available   LPAR Virtual Serial Adapter
vapps     Available   Virtual Target Device - Logical Volume
vdbsrv   Available   Virtual Target Device - Logical Volume
vlnx     Available   Virtual Target Device - Logical Volume
vnim     Available   Virtual Target Device - Logical Volume
```

8. Когда вы активизируете новые разделы, вы будете иметь в наличии hdisk1, как показано в примере 5-2. Отзеркалируйте ваш rootvg как вы обычно это делаете на AIX 5L.

Пример 5-2. Раздел NIM_Server с hdisk1, предоставленным VIO_Server2

```
# hostname
NIM_server
# lsvp
hdisk0      00cddeecb17bf4d0          rootvg      active
hdisk1      none                      None
# lsdev -Cc disk
hdisk0 Available Virtual SCSI Disk Drive
hdisk1 Available Virtual SCSI Disk Drive
# extendvg rootvg hdisk1
0516-1254 extendvg: Changing the PVID in the ODM.
# lspv
hdisk0      00cddeecb17bf4d0          rootvg      active
hdisk1      00cddeec9c2d47f6          rootvg      active
# mirrorvg rootvg hdisk1
0516-1124 mirrorvg: Quorum requirement turned off, reboot system for this
to take effect for rootvg.
0516-1126 mirrorvg: rootvg successfully mirrored, user should perform
bosboot of system to initialize boot records. Then, user must modify
bootlist to include: hdisk1 hdisk0.
# bosboot -a -d /dev/hdisk1
bosboot: Boot image is 23779 512 byte blocks.
# bootlist -m normal hdisk0 hdisk1
# bootlist -m normal -r
hdisk0 blv=hd5
hdisk1 blv=hd5
```

5.3. Сценарий 2: Перехват SEA

Этот сценарий покажет вам, как изменить существующий SEA для использования перехвата SEA и как создать второй SEA в режиме перехвата на VIO_Server2, который станет резервным путем на случай, если первичный VIO_Server1 станет недоступным. Это новая возможность, появившаяся в Virtual I/O Server V1.2.

Высокая доступность общего Ethernet-адаптера достигается с помощью создания дополнительного виртуального Ethernet-адаптера на каждом Virtual I/O Server, определяя его в виде канала управления для каждого SEA и с помощью изменения двух атрибутов на каждом из двух разделяемых Ethernet-адаптеров. Канал управления используется для передачи пакетов сердцебиения между двумя VIO, серверами. В Virtual I/O Server V1.2 доступна новая возможность для создания виртуального Ethernet-адаптера через кнопку выбора Access External Networks. Здесь вам нужно указать основной и дополнительный адаптеры с помощью назначения различных чисел приоритетов.

Рисунок 5-16 демонстрирует готовое решение для повышения доступности общего Ethernet-адаптера.

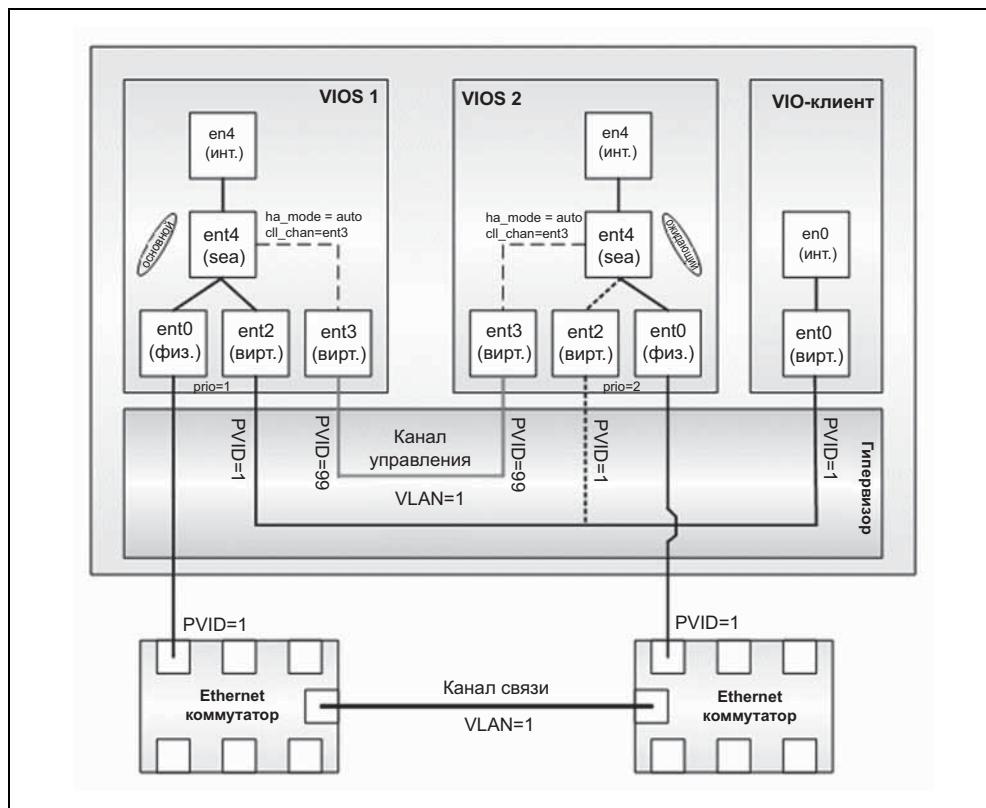


Рис. 5-16. Настройка повышения доступности адаптера SEA

Следующие шаги проведут вас через процесс настройки второго вспомогательного SEA-адаптера:

1. Создайте динамически виртуальный Ethernet-адаптер для каждого Virtual I/O Server, который будет выполнять работу канала управления:
 - a. На разделе VIO_Server1 выполните щелчок правой кнопкой мыши и выберите Dynamic Logical Partitioning Virtual Adapter Resources Add/Remove, как показано на рисунке 5-17.
 - b. Щелкните на кнопке Create Adapter и затем введите значение ID для Virtual LAN, как показано на рисунке 5-18 (значение Slot заполняется автоматически; целесообразно использовать значение по умолчанию). Когда закончите, нажмите OK.
 - c. Выполните те же шаги для профиля VIO_Server2.

Примечание. Для двух адаптеров, которые мы хотим создать, нам нужно получить уникальный для них общий PVID (например, оба адаптера должны иметь одинаковый PVID и этот PVID не должен использоваться другими адаптерами на этой машине). Как показано на рисунке 5-16, мы ввели значение VLAN 99.

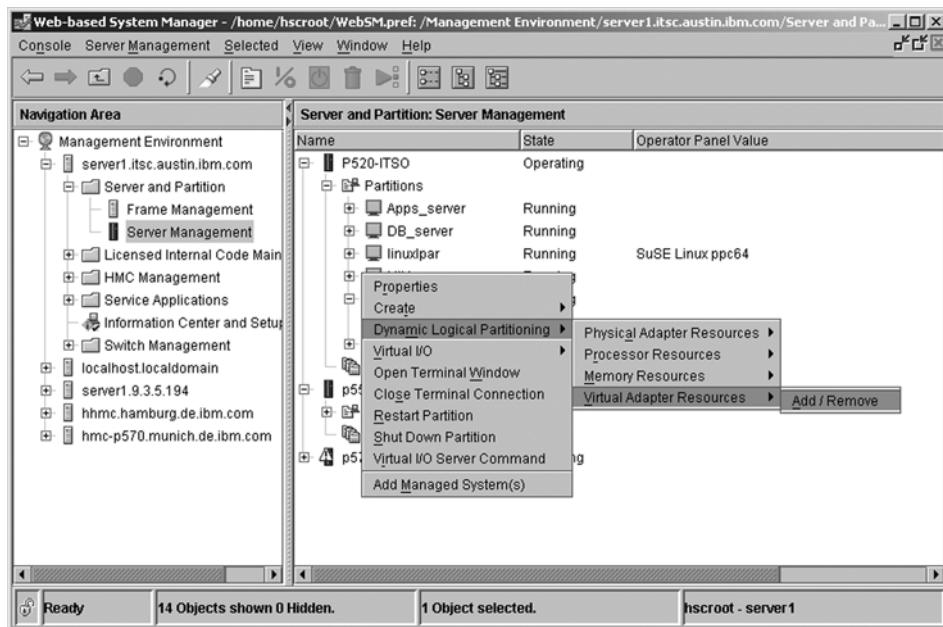


Рис. 5-17. Динамическая операция с LPAR по добавлению виртуального Ethernet на VIO_Server1

2. Сделайте щелчок правой кнопкой мыши на раздел VIO_Server2 и создайте виртуальный Ethernet-адаптер. Значение ID для Virtual LAN должно быть таким же, как и у основного SEA (в нашем примере 1), и кнопка выбора Access External Network должна быть отмечена. Флаг Trunk Priority для этого адаптера устанавливаем равным 2. Этим мы указываем, что SEA-адаптер на VIO_Server2 будет резервным. Когда закончите, нажмите **OK**.
3. Для того чтобы адаптер стал доступным на обоих серверах ввода-вывода, вам необходимо войти в систему с правами администратора и запустить для каждого VIOS команду **cfdev**, так как они добавляются динамически.

Совет. Вот настройки для виртуальных устройств на каждом VIOS:

На VIO_Server1:

- Свойства виртуального Ethernet SEA:
 - Virtual LAN ID 1
 - Trunk priority 1
 - Нажата кнопка Access External Network
- Виртуальный Ethernet для канала управления использует Virtual LAN ID 99

На VIO_Server2:

- Свойства виртуального Ethernet SEA:

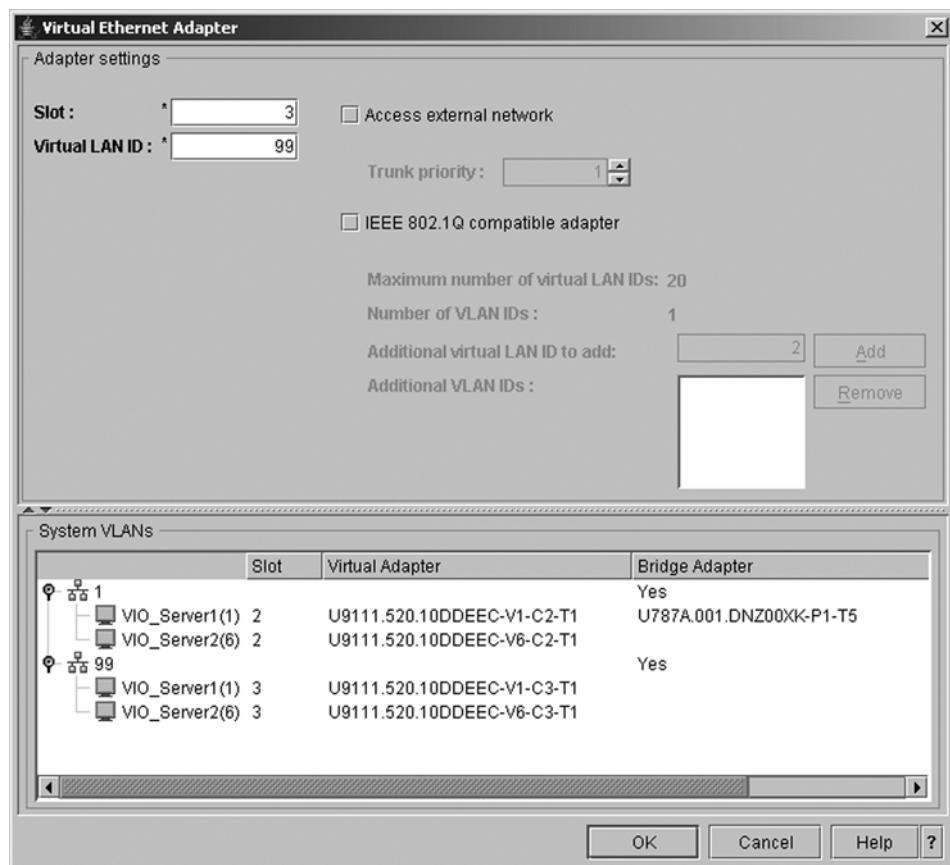


Рис. 5-18. Слоты виртуального Ethernet и значение ID для Virtual LAN (PVID)

- Virtual LAN ID 1
 - Trunk priority 2
 - Нажата кнопка Access External Network
- Виртуальный Ethernet для канала управления использует Virtual LAN ID 99
4. Измените устройство адаптера SEA на VIO_Server1 с помощью команды chdev, как показано в примере 5-3.

Пример 5-3. Общий Ethernet-адаптер на VIO_Server1

```
$chdev -dev ent4 -attr ha_mode=auto ctl_chan=ent3
ent4 changed
```

Укажите устройство адаптера SEA на VIO_Server2 с помощью команды mkdev¹, как показано в примере 5-4.

¹ Как видно из примера, используется команда mkvdev. Прим. науч. ред.

Пример 5-4. Общий Ethernet-адаптер на VIO_Server2

```
$mkvdev -sea ent0 -vadapter ent2 -default ent2 -defaultid 1 -attr ha_mode=auto  
ctl_chan=ent3  
ent4 Available  
en4  
et4
```

Совет. Несовпадение SEA и перехвата SEA может привести к широковещательному шторму в сети и повлиять на стабильность вашей сети. При обновлении от SEA до перехвата SEA обязательно нужно изменить VIOS с SEA на перехват SEA до создания второго SEA с включенным перехватом SEA.

5. Проверьте свойства адаптера SEA на обоих Virtual I/O Server SEA-как показано в примере 5-5.

Пример 5-5. Проверьте и измените свойства адаптера SEA

```
$ lsdev -dev ent4 -attr  
attribute value description user_settable  
ctl_channent3Control Channel adapter for SEA failover True  
ha_mode auto High Availability Mode True  
netaddr Address to ping True  
pvid 1 PVID to use for the SEA device True  
pvid_adapter ent2 Default virtual adapter to use for non-VLAN-tagged  
packets True  
real_adapter ent0 Physical adapter associated with the SEA True  
thread 0 Thread mode enabled (1) or disabled (0) True  
virt_adapters ent2 List of virtual adapters associated with the SEA (comma  
separated) True
```

6. Создайте IP-адрес общего Ethernet-адаптера на VIO_Server2 с помощью команды mktcpip как показано в примере 5-6.

Пример 5-6. Создание IP-адреса для общего Ethernet-адаптера

```
$ mktcpip -hostname VIO_Server2 -interface en4 -inetaddr 9.3.5.136 -netmask  
255.255.255.0 -gateway 9.3.5.41 -nsrvaddr 9.3.4.2 -nsrvdomain  
itsc.austin.ibm.com
```

Тестирование перехвата SEA

Для проверки ваших настроек и конфигурации вам нужно протестировать перехват SEA.

Настройки тестирования

Для тестирования того, что перехват SEA действительно работает ожидаемым образом, вам нужно открыть удаленную сессию командного интерпретатора (shell) через адаптер SEA к любому разделу VIO-клиента с любой системы во

внешней сети, например с вашей рабочей станции. Из командного интерпретатора попробуйте запустить любую команду, порождающую длительный вывод в консоль. Теперь вы готовы к тестированию.

Тестируемые случаи

Данная серия описывает тестируемые случаи, которые вам нужно проверить для того, чтобы быть уверенным, что ваши настройки перехвата SEA работают надлежащим образом.

Внимание. Если ваши Virtual I/O Server в дополнение к SEA предоставляют Virtual SCSI диски, в случае отключения или ошибки виртуальных серверов ввода-вывода на VIO-клиентах окажутся устаревшие разделы (stale partitions). Поэтому после каждого теста вам нужно не забыть ресинхронизировать все устаревшие разделы перед продолжением тестирования.

1. Ручной перехват:
 - a. Установите на основном адаптере ha_mode в режим ожидания: SEA должен выполнить обработку ошибки:
`chdev -dev ent2 -attr ha_mode=standby`
 - b. Сбросьте ha_mode в автоматический режим на основном адаптере: SEA должен выполнить откат после ошибки:
`chdev -dev ent2 -attr ha_mode=auto`
2. Отключение VIOS:
 - a. Перезагрузите основной VIOS: SEA должен выполнить обработку ошибки.
 - b. Далее основной VIOS опять загрузится: SEA должен выполнить откат после ошибки.
3. Ошибка VIOS:
 - a. Отключите основной раздел с HMC: SEA должен выполнить обработку ошибки.
 - b. Включите и загрузите VIOS: SEA должен выполнить откат после ошибки.
4. Физическая ошибка канала связи:
 - a. Физически отключите адаптер на основном SEA: SEA должен выполнить обработку ошибки.
 - b. Переподключите адаптер на основном SEA: SEA должен выполнить откат после ошибки.
5. Обратная последовательность загрузки:
 - a. Отключите оба VIOS.
 - b. Загрузите резервный VIOS: SEA должен задействовать свой резерв.
 - c. Загрузите основной VIOS: SEA должен выполнить откат.

5.4. Сценарий 3: MPIO на клиенте с SAN в VIOS

Этот раздел описывает настройку усложненного сценария с двумя Virtual I/O Server, подключенными к DS4200. Оба Virtual I/O Server обслуживают один и тот же DS4200 LUN на клиентском разделе. Использование MPIO с PCM по умолчанию на клиентском разделе дает избыточный доступ к обслуживаемому LUN. Этот сценарий показан на рисунке 5-19.

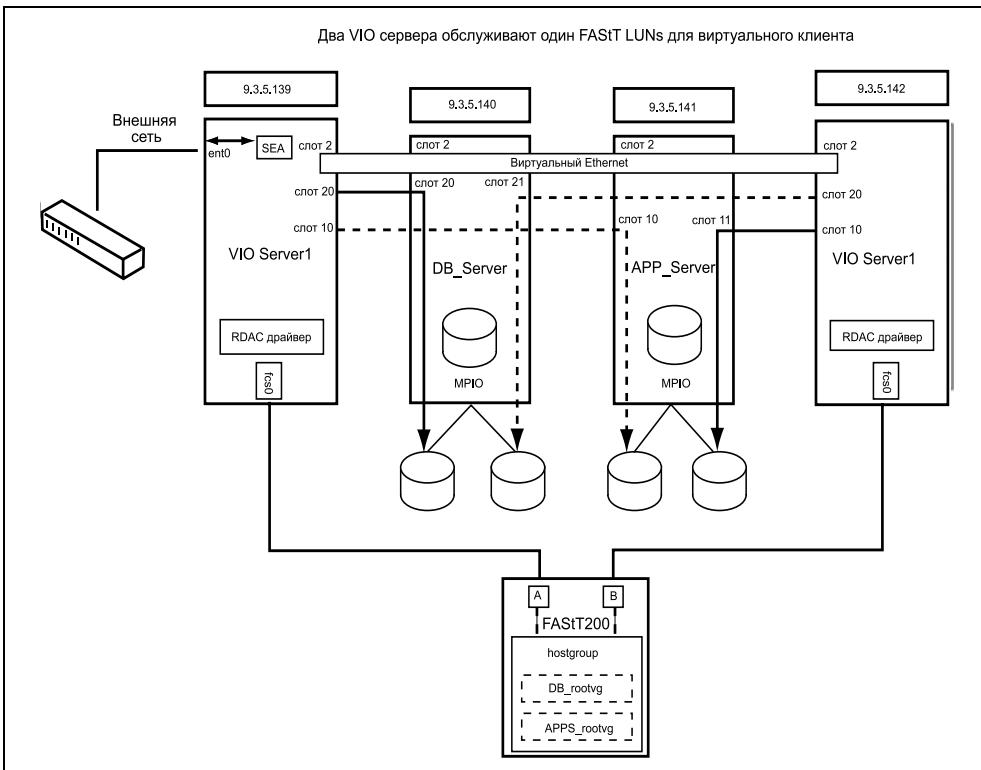


Рис. 5-19. Присоединение SAN к нескольким Virtual I/O Server

В этом сценарии мы подключаем каждый Virtual I/O Server напрямую к DS4200 с помощью одного оптоволоконного адаптера. На Virtual I/O Server RDAC-драйвер уже присутствует независимо от наличия нескольких оптоволоконных адаптеров.

Вы настроили шесть 50 Гб LUN на DS4200, принадлежащих одной хост-группе. Хост-группа состоит из VIO Server1 и VIO Server2, так что оба Virtual I/O Server подключены к одному и тому же диску.

На рисунке 5-20 показан общий вид отображения хост-группы на DS4200.

В этом сценарии мы подключаем каждый Virtual I/O Server напрямую к DS4200 с помощью оптоволоконного адаптера. В нашем случае мы используем эти настройки для демонстрации базовых настроек.

При использовании одного оптоволоконного адаптера для каждого Virtual I/O Server вам нужно иметь дополнительный коммутатор для поддержки такой конфигурации. В этом случае важно, чтобы зонирование SAN было настроено таким образом, чтобы отдельный НВА в каждой VIOS LPAR был зонирован, чтобы видеть оба контроллера хранилищ в FAStT. Если второй НВА используется для повышения избыточности, администратор хранилища должен быть уверен, что каждый НВА зонирован только на один контроллер DS4200. Правила присоединения элементов хранилища FAStT к AIX можно найти в документации к Storage Manager для продуктов Storage Manager.

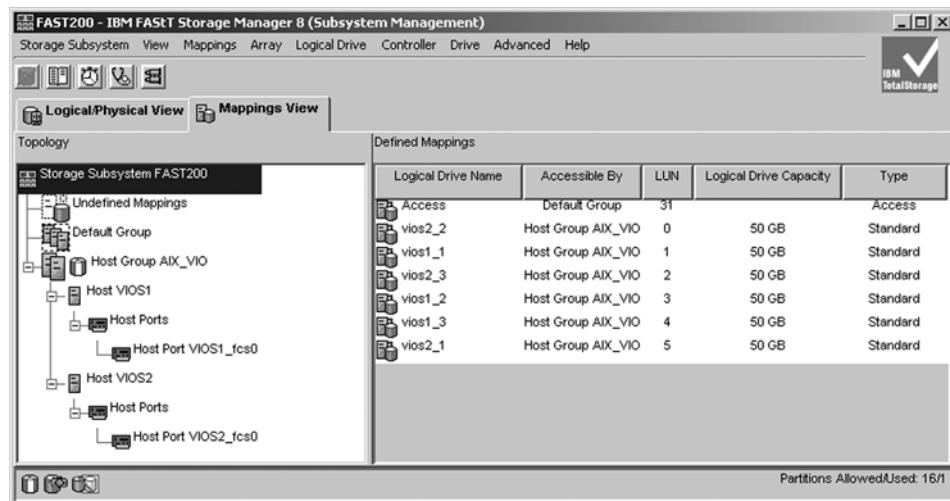


Рис. 5-20. Общий вид настройки DS4200

На Virtual I/O Server RDAC-драйвер уже присутствует независимо от наличия нескольких оптоволоконных адаптеров.

5.4.1. Настройка НМС

Для настройки данного сценария вам потребуется выполнить следующие шаги:

1. Создайте два раздела Virtual I/O Server и назовите их VIO_Server_SAN1 и VIO_Server_SAN1 следуя инструкциям в 4.2 «Создание разделов Virtual I/O Server». На шаге 10 выберите оптоволоконный адаптер в дополнение к показанному физическому адаптеру.
2. Установите оба Virtual I/O Server руководствуясь инструкцией в 4.3 «Установка программного обеспечения для Virtual I/O Server».
3. После успешной установки вы можете отображать общемировые имена (WWN, world wide names) оптоволоконного адаптера на созданную LUN. Используйте команду `cfdev` для того, чтобы сделать диски доступными для Virtual I/O Server. После успешного присоединения к дискам остановите оба Virtual I/O Server.
4. Создайте два клиентских раздела с именами DB_server и APP_server, следуя инструкции в 4.4.3 «Создание клиентского раздела». Не указывайте сейчас никакого виртуального устройства. Рисунок 5-21 демонстрирует все разделы, которые нам нужно настроить для этого сценария: два Virtual I/O Server уже установлены и выключены, и два клиентских раздела уже созданы.
5. Определите виртуальный адаптер ввода-вывода на Virtual I/O Server. Выберите профиль раздела VIO_Server_SAN1, сделайте на нем щелчок правой кнопки мыши и выберите **Properties**, чтобы открыть окно свойств профиля локального раздела.

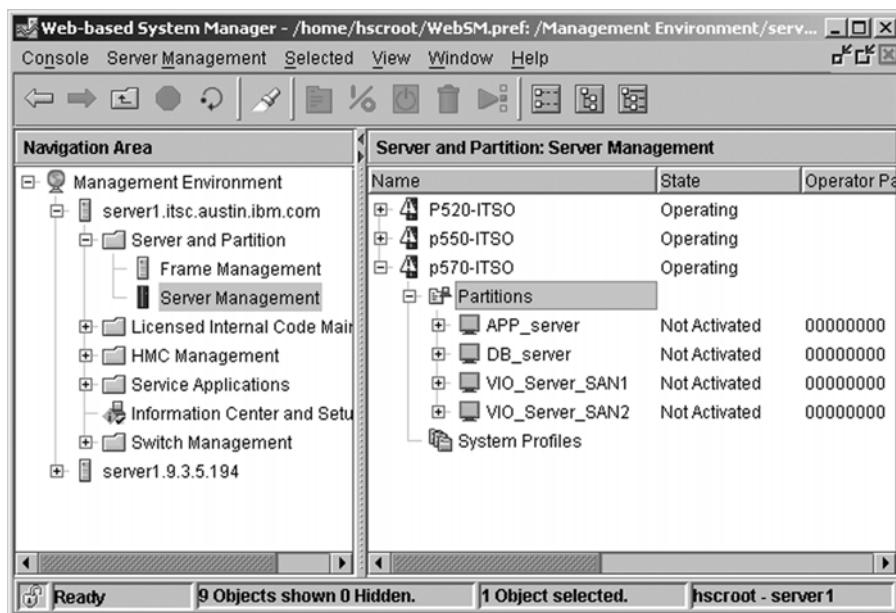


Рис. 5-21. Начальные настройки сценария

6. Выберите закладку **Virtual I/O Adapters** и измените номер максимума виртуальных адаптеров на 10. Затем нажмите **Create Server adapter**, как показано на рисунке 5-22.
7. Укажите слот номер 10 для слота сервера и выберите APP_Server-клиент в качестве удаленного раздела для подключения к слоту 10, как показано на рисунке 5-23.
8. Повторите этот шаг для раздела BD_Server, используя слот 20 для сервера и слот 20 для клиентского раздела. Рисунок 5-24 демонстрирует созданный SCSI-адаптер на Virtual I/O Server.
9. Нажмите закладку **Ethernet** для создания виртуального Ethernet-адаптера с PVID 1 и включите флаг **Trunk Priority**¹. Эта часть SEA будет предоставлять IP-доступ к разделам клиента.
10. Нажмите OK и повторите шаги с 5 по 8 для VIO_Server_SAN2, за исключением указания клиентского раздела для SCSI-адаптера сервера:
 - Укажите 11 в качестве слота клиентского раздела для APP_server.
 - Укажите 21 в качестве слота клиентского раздела для DB_server.
 На рисунке 5-25 показан общий вид созданного SCSI-адаптера сервера на разделе VIO_Server_SAN2.
11. Запустите оба Virtual I/O Server.
12. Укажите виртуальный адаптер ввода-вывода для раздела клиента. Начните с раздела APP_Server и выберите профиль, сделайте щелчок правой кнопкой

¹ Здесь ошибка: нужно отметить флаг Access external network. Прим. науч. ред.

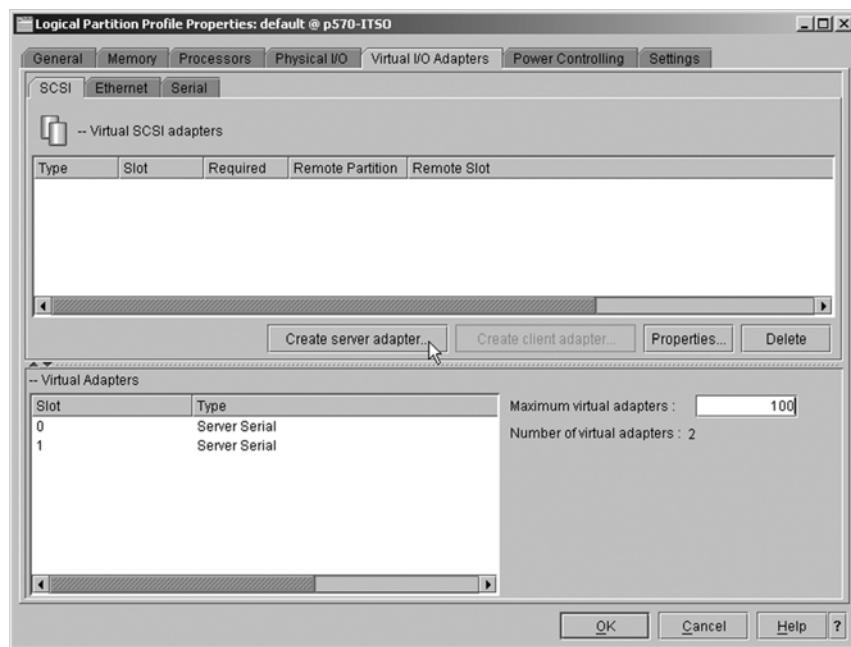


Рис. 5-22. Окно свойств профиля локального раздела

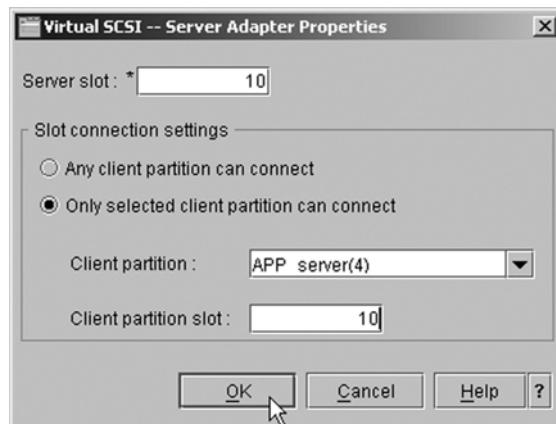


Рис. 5-23. Virtual SCSI: окно свойств адаптера сервера

мыши и выберите **Properties** для открытия окна свойств профиля локального раздела.

13. Выберите закладку **Virtual I/O Adapter** и измените количество максимума виртуальных адаптеров на 100. Затем нажмите **Create Client Adapter**. Заполните значения, как показано на рисунке 5-26.

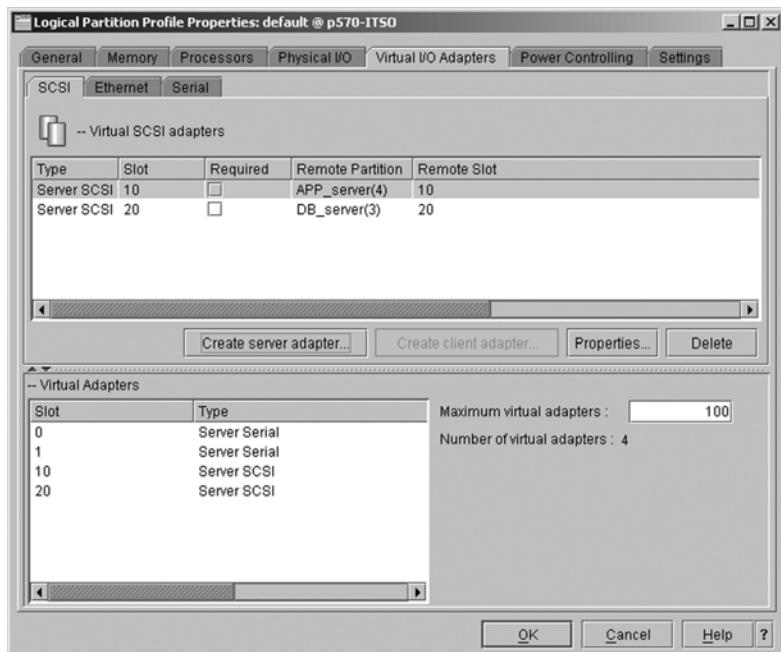


Рис. 5-24. Общий вид SCSI-адаптера сервера на VIO_Server_SAN1

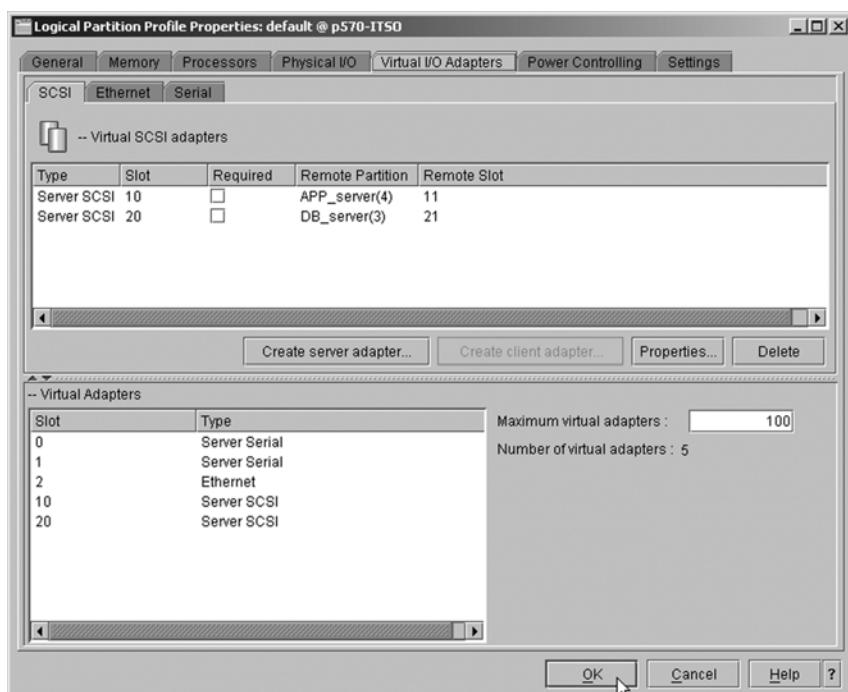


Рис. 5-25. Внешний вид SCSI-адаптера сервера на VIO_Server_SAN2

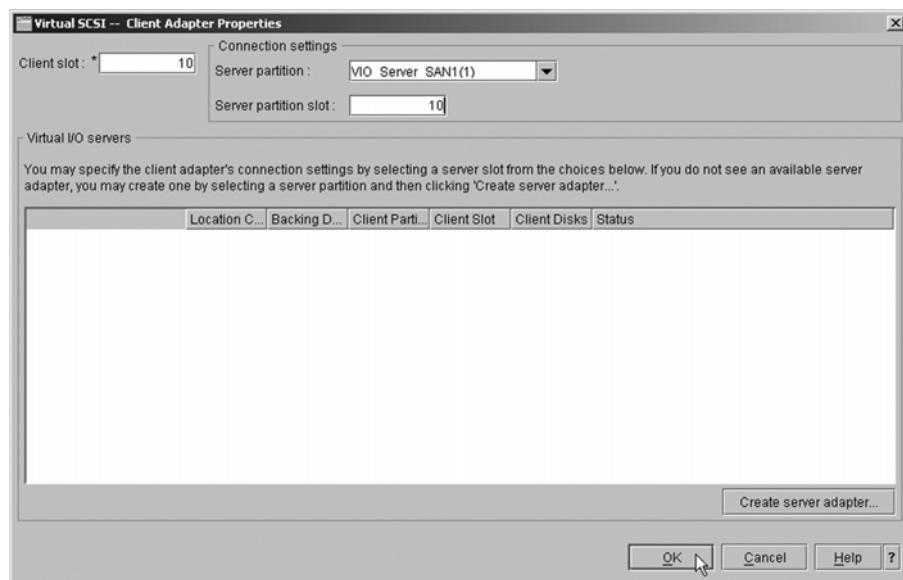


Рис. 5-26. Virtual SCSI: окно свойств клиентского адаптера

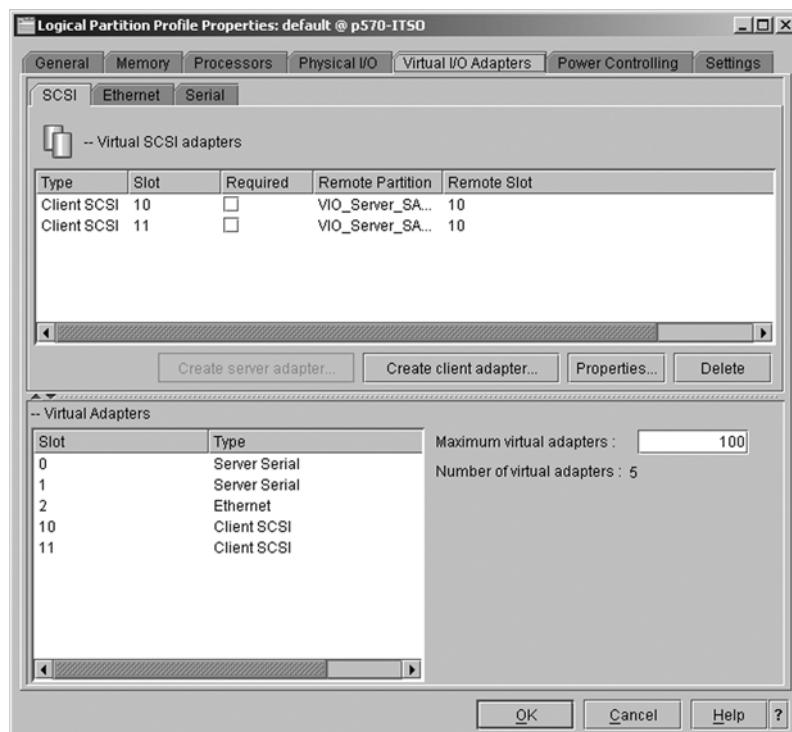


Рис. 5-27. Внешний вид адаптера SCSI клиента на разделе APP_server

14. Повторите шаг 13 для второго SCSI-адаптера клиента, подключенного к VIO_Server SAN2. Выберите клиентский слот номер 11 и слот раздела сервера номер 10. На рисунке 5-27 представлен общий вид после создания обоих клиентских SCSI-адаптеров.
15. Нажмите на закладку **Ethernet** и создайте виртуальный Ethernet-адаптер с PVID 12 с неотмеченной кнопкой выбора **Access External Network**.
16. Повторите шаги с 12 по 15 для раздела DB_server и измените номер слота, когда будете указывать SCSI-адаптер клиента значениями, указанными в таблице 5-3.

Таблица 5-3. Указание SCSI-адаптера клиента для раздела SB_server

Слот клиента	Раздел сервера	Слот раздела сервера
20	VIO_Server_SAN1	20
21	VIO_Server_SAN2	20

На Рисунке 5-28 показан общий вид созданного SCSI-адаптера клиента на разделе DB_server.

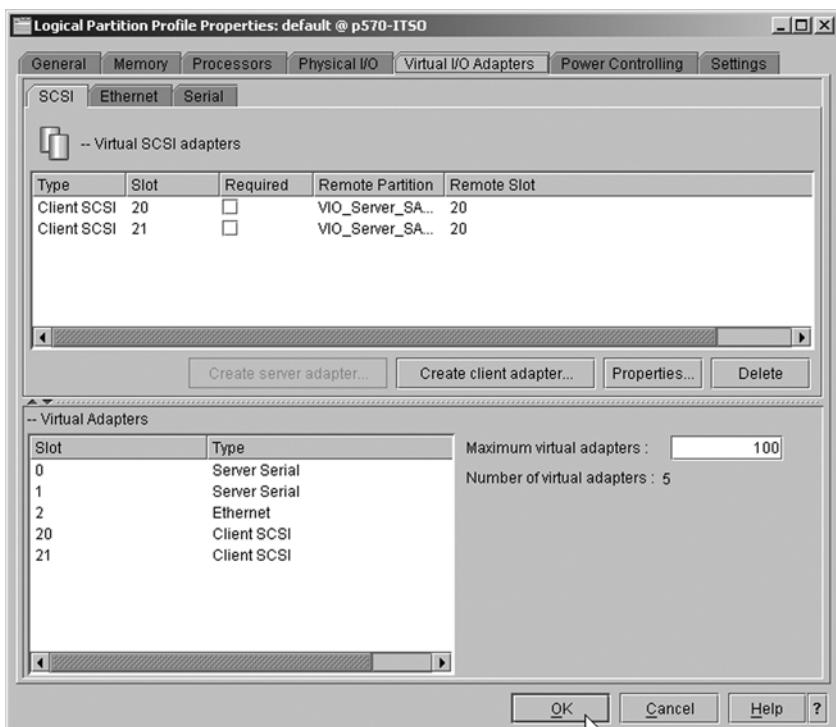


Рис. 5-28. Внешний вид SCSI-адаптера клиента на разделе DB_server

5.4.2. Настройка Virtual I/O Server

Для настройки дисков на VIO_Server_SAN1 следуйте нижеперечисленным шагам:

1. Откройте окно терминала для раздела VIO_Server_SAN2; выбрав этот раздел, сделайте щелчок правой кнопкой мыши и выберите Open Terminal Window.
2. Настройте SEA, как описано в шагах с 1 по 4 в 4.4.7 «Создание общего Ethernet-адаптера». В нашем примере мы используем сетевые настройки, указанные в таблице 5-4.

Таблица 5-4. Сетевые настройки

Настройка	Значение
имя хоста	VIO_Server_SAN1
IP-адрес	9.3.5.139
сетевая маска	255.255.255.0
шлюз	9.3.5.41

3. Проверьте подключенные DS4200 диски с помощью команды lsdev с флагом -type, как показано на рисунке 5-7.

Пример 5-7. Проверка дисков DS4200

```
$ lsdev -type disk
name status description
hdisk0      Available 16 Bit LVD SCSI Disk Drive
hdisk1      Available 16 Bit LVD SCSI Disk Drive
hdisk2      Available 3542      (200) Disk Array Device
hdisk3      Available 3542      (200) Disk Array Device
hdisk4      Available 3542      (200) Disk Array Device
hdisk5      Available 3542      (200) Disk Array Device
hdisk6      Available 3542      (200) Disk Array Device
hdisk7      Available 3542      (200) Disk Array Device
```

Вывод этой команды перечисляет два внутренних SCSI-диска и шесть дисков DS4200.

4. Проверьте адаптер vhost с помощью команды lsmap, с флагом -all как показано в примере 5-8. Номер слота, который мы настроили на HMC, показан в колонке Phyloc. SCSI-адаптер сервера, который мы настроили в слоте 10, показывает локацию C10.

Пример 5-8. Список vhost адаптеров

```
$ lsmap -all
SVSA          Physloc                           Client PartitionID
-----
vhost0        U9117.570.107CD9E-V1-C10           0x00000000
VTD           NO VIRTUAL TARGET DEVICE FOUND
```

SVSA	Physloc	Client PartitionID
vhost1	U9117.570.107CD9E-V1-C20	0x00000000
VTD	NO VIRTUAL TARGET DEVICE FOUND	

5. Проверьте свойства hdisk2 и атрибут reserve_policy с помощью команды lsdev, как показано в примере 5-9.

Пример 5-9. Отображение атрибутов hdisk для VIOS

attribute	value	description	user_settable
PR_key_value	none	Persistant Reserve Key Value	True
cache_method	fast_write	Write Caching method	False
ieee_volname	600A0B80000BDC160000051642F3A58E	IEEE Unique volume name	False
lun_id	0x0000000000000000	Logical Unit Number	False
max_transfer	0x100000	Maximum TRANSFER Size	True
prefetch_mult	1	Multiple of blocks to prefetch on read	False
pvid	none	Physical volume identifier	False
q_type	simple	Queuing Type	False
queue_depth	10	Queue Depth	True
raid_level	5	RAID Level	False
reassign_to	120	Reassign Timeout value	True
reserve_policy	single_path	Reserve Policy	True
rw_timeout	30	Read/Write Timeout value	True
scsi_id	0xef	SCSI ID	False
size	51200	Size in Mbytes	False
write_cache	yes	Write Caching enabled	False

6. Измените атрибут reserve_policy с single_path на no_reserve с помощью команды chdev, как показано в примере 5-10.

Пример 5-10. Настройка атрибута на no_reserve

```
$ chdev -dev hdisk2 -attr r reserve_policy=no_reserve
hdisk2 changed
```

7. Проверьте еще раз с помощью команды lsdev, что reserve_policy теперь установлен в no_reserve, как показано в примере 5-11.

Пример 5-11. Вывод атрибутов hdisk attribute после изменения атрибута reserve_policy

attribute	value	description	user_settable
PR_key_value	none	Persistant Reserve Key Value	True
cache_method	fast_write	Write Caching method	False
ieee_volname	600A0B80000BDC160000051642F3A58E	IEEE Unique volume name	False
lun_id	0x0000000000000000	Logical Unit Number	False
max_transfer	0x100000	Maximum TRANSFER Size	True
prefetch_mult	1	Multiple of blocks to prefetch on read	False

pvid	none	Physical volume identifier	False
q_type	simple	Queuing Type	False
queue_depth	10	Queue Depth	True
raid_level	5	RAID Level	False
reassign_to	120	Reassign Timeout value	True
reserve_policy	no_reserve	Reserve Policy	True
rw_timeout	30	Read/Write Timeout value	True
scsi_id	0xef	SCSI ID	False
size	51200	Size in Mbytes	False
write_cache	yes	Write Caching enabled	False

8. Повторите шаги с 5 по 7 для hdisk3.
9. Для оптоволоконного адаптера измените атрибут fc_err_recov на fast_fail и dyntrk на yes. Используйте команду chdev, как показано в примере 5-12.

Пример 5-12. Изменение атрибутов для оптоволоконного адаптера

```
$ chdev -dev fscsi0 -attr fc_err_recov=fast_fail dyntrk=yes -perm
fscsi0 changed
$ lsdev -dev fscsi0 -attr
attribute    value      description          user_settable
attach        al        How this adapter is CONNECTED   False
dyntrk        yes       Dynamic Tracking of FC Devices True
fc_err_recov fast_fail FC Fabric Event Error RECOVERY Policy True
scsi_id       0x1       Adapter SCSI ID           False
sw_fc_class  3         FC Class for Fabric       True
```

Примечание. Атрибут fast_fail изменяется для того, чтобы, если драйвер оптоволоконного адаптера обнаружит событие соединения, такое как обрыв связи между устройством хранения и коммутатором, любой новый ввод/вывод или последующие попытки неудавшейся операции ввода-вывода сразу были признаны неудачными до того, как драйвер адаптера определит, что связь через оптоволокно восстановлена. По умолчанию этот атрибут установлен в delayed_fail.

Установка атрибута dyntrk в yes делает AIX 5L терпимой к изменению в прокладке кабелей в SAN. Обратите внимание, что эта функция поддерживается не на всех системах хранения. Узнайте наличие поддержки этой функции у вашего поставщика.

10. Перезагрузите Virtual I/O Server для того, чтобы изменения вступили в силу. Вам нужно перезагрузиться только после того, как вы измените атрибуты оптоволоконного канала.
11. Загрузите VIO_Server_SAN2. Повторите шаги с 1 по 10, за исключением создания SEA. Так как в данном сценарии мы не концентрируем внимание на избыточности доступа к внешним сетям, настройте физический Ethernet-адаптер с помощью значений, показанных в таблице 5-4. В нашем примере VIO_Server_SAN2 имеет IP-адрес 9.3.5.124.

Проверьте на обеих Virtual I/O Server чтобы, vhost0 и vhost1 имели одинаковые номера слотов при запуске команды lsmmap.

12. Убедитесь, что диски, которые вы хотите отобразить на клиентов, имеют одинаковые номера hdisk и LUN на обеих Virtual I/O Server с помощью команды lsdev, как показано в примере 5-13.

Пример 5-13. Определение LUN ID на диске DS4200

```
$ lsdev -dev hdisk2 -vpd
hdisk2          U7879.001.DQD186N-P1-C3-T1-W200400A0B8110D0F-L0  3542
(200) Disk Array Device
  PLATFORM SPECIFIC
    Name: disk
    Node: disk
    Device Type: block
$ lsdev -dev hdisk3 -vpd
hdisk3          U7879.001.DQD186N-P1-C3-T1-W200400A0B8110D0F-L10000000000000
3542          (200) Disk Array Device
  PLATFORM SPECIFIC
    Name: disk
    Node: disk
    Device Type: block
```

13. Отобразите hdisk на адаптер vhost с помощью команды mkvdev, как показано в примере 5-14.

Пример 5-14. Отображение диска на адаптер vhost

```
$ mkvdev -vdev hdisk2 -v adapter vhost0 -dev app_server r
app_server Available
```

Важно. При подключении к одному диску нескольких серверов ввода-вывода, в качестве резервного устройства доступен только hdisk. Вы не можете создавать группы томов на этих дисках и использовать логические тома в качестве резервных устройств.

14. Повторите шаг 13 для hdisk3, отобразите этот диск на адаптер vhost1 и назовите Virtual SCSI-устройство db_server. В примере 5-15 показан вывод команды lsmap после отображения обоих дисков на адаптер vhost.

Пример 5-15. Вывод lsmap -all после отображения дисков на VIO_Server_SAN1

```
$ lsmap -all
SVSA          Physloc                               Client PartitionID
-----
vhost0        U9117.570.107CD9E-V2-C10           0x00000000
VTd          app_server
LUN          0x8100000000000000
Backing device      hdisk2
Physloc       U7879.001.DQD186K-P1-C3-T1-W200500A0B8110D0F-L0
SVSA          Physloc                               Client PartitionID
-----
```

vhost1	U9117.570.107CD9E-V2-C20	0x00000000
VTD	db_server	
LUN	0x8100000000000000	
Backing device	hdisk3	
Physloc	U7879.001.DQD186K-P1-C3-T1-W200500A0B8110D0F-L1000000000000	

15. Загрузите VIO_Server_SAN2 и повторите шаги с 13 по 14. Пример 5-16 показывает вывод команды lsmap после отображения обоих дисков на адаптер vhost.

Пример 5-16. Вывод lsmap –all после отображения дисков на VIO_Server_SAN2

\$ lsmap -all		
SVSA	Physloc	Client PartitionID
vhost0	U9117.570.107CD9E-V2-C10	0x00000004
VTD	app_server	
LUN	0x8100000000000000	
Backing device	hdisk2	
Physloc	U7879.001.DQD186K-P1-C3-T1-W200500A0B8110D0F-L0	
SVSA	Physloc	Client Partition ID
vhost1	U9117.570.107CD9E-V2-C20	0x00000003
VTD	db_server	
LUN	0x8100000000000000	
Backing device	hdisk3	
Physloc	U7879.001.DQD186K-P1-C3-T1-W200500A0B8110D0F-L1000000000000	

16. Установите два клиентских раздела с помощью NIM или назначьте оптическое устройство как желаемый ресурс для установки AIX 5L. За инструкциями касательно установки AIX 5L обратитесь к 4.4.8 «Установка клиентского раздела AIX 5L».

5.4.3. Работа с MPIO на клиентских разделах

После установки клиентского раздела настройте MPIO для клиентских разделов. Включите режим проверки состояния (health check) для диска, чтобы иметь возможность получать достоверную информацию о состоянии диска в случае, если Virtual I/O Server отключится, а затем вновь запустится.

Настройте клиентский раздел DB_server с активным путем через VIO_Server_SAN1 и клиентский раздел APP_server с активным путем через VIO_Server_SAN2.

Следуйте нижеприведенным шагам по настройке клиентских разделов:

1. Загрузите ваш клиентский раздел APP_server.
2. Проверьте настройку MPIO с помощью команд, указанных в примере 5-17. В этом сценарии отображается только настраиваемый hdisk.

Пример 5-17. Проверка настройки hdisk для клиентских разделов

```
# lspv
hdisk0      00c7cd9eabe9f4bf          rootvg      active
# lsdev -Cc disk
hdisk0 Available Virtual SCSI Disk Drive
```

3. Запустите команду `lspath` для проверки того, что диск подключен с использованием двух различных путей. Пример 5-18 демонстрирует, что `hdisk0` подключен с помощью `VSCSI0`, а адаптер `VSCSI1` указывает на другой Virtual I/O Server. Оба Virtual I/O Server загружены и работают. Оба пути включены.

Пример 5-18. Проверка путей `hdisk0`

```
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

4. Настройте клиентский раздел с помощью команды `chdev` и переключите его в режим обновления путей, изменив атрибут `hcheck_interval` для `hdisk0`. Для проверки настроек атрибута используйте команду `lsattr`, как показано в примере 5-19.

Пример 5-19. Отображение атрибутов `hdisk0`

```
# lsattr -El hdisk0
PCM          PCM/friend/vscsi           Path Control Module  False
algorithm    fail_over                  Algorithm          True
hcheck_cmd   test_unit_rdy            Health Check Command  True
hcheck_interval 0                    Health Check Interval  True
hcheck_mode   nonactive               Health Check Mode   True
max_transfer 0x40000                Maximum TRANSFER Size  True
pvid         00c7cd9eabdeaf32000000000000000000 Physical volume identifier False
queue_depth   3                     Queue DEPTH        False
reserve_policy no_reserve           Reserve Policy     True
```

Так как атрибут `hcheck_interval` установлен в 0, клиентский раздел не обновляет пути при использовании команды `lspath` в случае ошибки активного пути. Для включения функции проверки состояния используйте команду `chdev`, как показано в примере 5-20. В этом примере мы используем проверку состояния с интервалом в 60 секунд.

Пример 5-20. Изменение интервала проверки состояния

```
# chdev -l hdisk0 -a hcheck_interval=60 -P
hdisk0 changed
# lsattr -El hdisk0
PCM          PCM/friend/vscsi           Path Control Module  False
algorithm    fail_over                  Algorithm          True
hcheck_cmd   test_unit_rdy            Health Check Command  True
hcheck_interval 60                  Health Check Interval  True
hcheck_mode   nonactive               Health Check Mode   True
```

max_transfer	0x40000	Maximum TRANSFER Size	True
pvid	00c7cd9eabdeaf320000000000000000	Physical volume identifier	False
queue_depth	3	Queue DEPTH	False
reserve_policy	no_reserve	Reserve Policy	True

Примечание. MPIO на клиентском разделе выполняет алгоритм fail_over. Это значит, что в каждый момент активен только один путь. Если вы отключите виртуальный сервер, который служит неактивным путем, режим выбора пути не изменится и ошибки не возникнут, так как ввод-вывод этот путь не использует.

- Установите приоритетный путь для этого раздела, для того чтобы путь проходил через VIO_Server2. Настройки по умолчанию равны 1 для обоих, как показано в примере 5-21. В этом случае вам не нужно указывать путь специально и система будет автоматически выбирать path0 как активный путь. Приоритет 1 – наивысший, и вы имеете возможность задавать приоритет в пределах от 1 до 255.

Пример 5-21. Вывод приоритета пути с помощью lspath

```
# lspath -AHE -l hdisk0 -p vscsi0
attribute value description user_settab
priority 1 Priority True
# lspath -AHE -l hdisk0 -p vscsi1
attribute value description user_settab
priority 1 Priority True
```

Для определения того, какой путь будет использовать раздел VIO_Server_SAN2, используйте команду lscfg и проверьте номер слота для устройства SCSI, как показано в примере 5-22.

Пример 5-22. Посмотрите, какой родитель какому пути принадлежит

```
# lscfg -vl vscsi0
vscsi0      U9117.570.107CD9E-V4-C10-T1 Virtual SCSI Client Adapter
            Device Specific.(YL).....U9117.570.107CD9E-V4-C10-T1
# lscfg -vl vscsi1
vscsi1      U9117.570.107CD9E-V4-C11-T1 Virtual SCSI Client Adapter
            Device Specific.(YL).....U9117.570.107CD9E-V4-C11-T1
```

В нашем примере четные номера слотов настроены указывать на раздел VIO_Server_SAN2. Для настройки активного пути, использующего устройство VSCSI1, оставьте приоритет равным 1, т.е. наивысшим. Измените приоритет пути, использующего VSCSI0 на 2, как показано в примере 5-23.

Пример 5-23. Изменение приоритета пути

```
# chpath -l hdisk0 -p vscsi0 -a priority=2
path Changed
```

6. Перезагрузите клиентский раздел, чтобы изменения вступили в силу. Оба изменения требуют перезагрузки.
7. Повторите эти шаги для раздела DB_server, но установите приоритет для пути VSCSI1 в 2.

5.5. Запуск установки Linux на клиенте VIO

Нижеприведенные шаги описывают, как загрузить в SMS раздел linuxpar и установить SUSE Linux Enterprise Server 9 на назначенному Virtual SCSI диске:

1. Загрузите раздел linuxpar с помощью привода CD или DVD, как на обычном AIX 5L разделе.
2. После загрузки первый экран будет похож на приведенный в примере 5-24.
3. Напечатайте *install* и нажмите Enter для запуска установки.

Пример 5-24. Установка SUSE Linux Enterprise Server 9 на раздел linuxpar

```
Welcome to SuSE Linux (SLES9 preview)

Use "install"      to boot the pSeries 64bit kernel
Use "install32"    to boot the 32bit RS/6000 kernel

You can pass the option "noinitrd"  to skip the installer
Example: install noinitrd root=/dev/sda4

Welcome to yaboot version 1.3.11.SuSE
Enter "help" to get some basic usage information
boot:install
```

Примечание. После ввода *install* и нажатия Enter вы попадете в Yaboot – инсталлятор SUSE Linux, где вы сможете выбрать настройки установки перед началом непосредственной установки Linux.

После успешной установки Linux раздел отобразится, как *SUSE Linux ppc64* в колонке operator окна HMC, как показано на рисунке 5-29.

5.6. Поддерживаемые конфигурации

В этом разделе обсуждаются различные поддерживаемые конфигурации окружения Virtual I/O Server. Описаны следующие конфигурации:

- ▶ Поддерживаемые конфигурации VSCSI на Virtual I/O Server:
 - Зеркалируемые VSCSI устройства на клиенте и сервере
 - Многопутевые конфигурации в SAN-окружении
- ▶ Конфигурации с виртуальным Ethernet
- ▶ Виртуальные устройства и поддержка HACMP
- ▶ Виртуальные устройства и поддержка GPFS

Описываемые в данном разделе конфигурации не представляют собой полный перечень поддерживаемых конфигураций. Они демонстрируют только набор

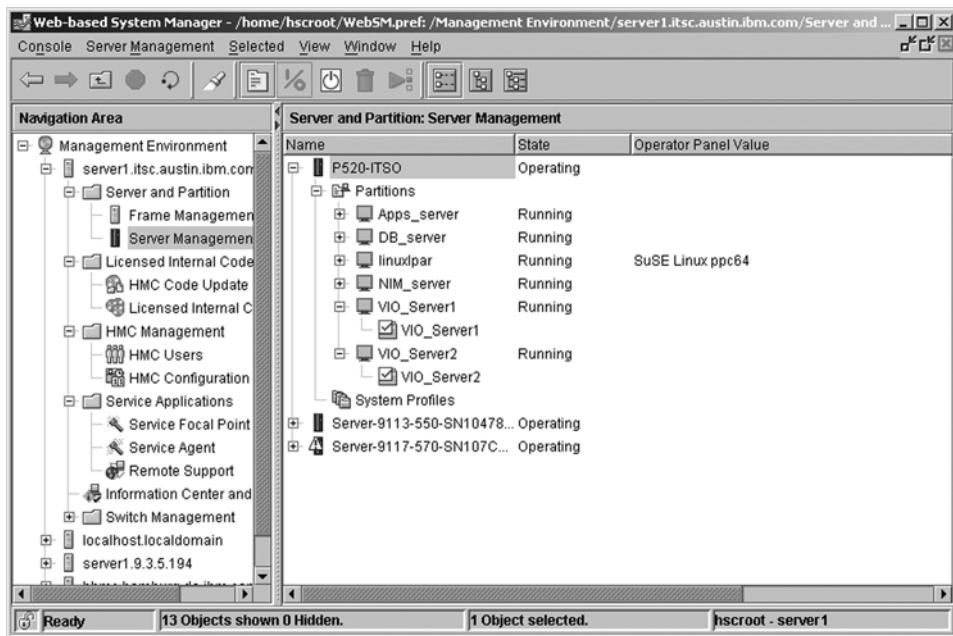


Рис. 5-29. Установленный SUSE-раздел

наиболее широко адаптируемых конфигураций, которые отвечают требованиям большинства рабочих сред.

Последнюю информацию о поддерживаемых окружениях Virtual I/O Server вы можете найти на следующем веб-сайте:

<http://techsupport.services.ibm.com/server/vios/documentation/datasheet.html>

5.6.1. Поддерживаемые конфигурации VSCSI

Данное описание конфигураций VSCSI описывает поддерживаемые и рекомендуемые конфигурации для использования Virtual I/O Server. Оно дополняет обсуждение в 5.1 «Обеспечение повышения доступности путем увеличения количества Virtual I/O Server». Для понимания вам потребуется знание принципов избыточности и доступности.

Для получения информации об использовании логических томов на виртуальном сервере ввода-вывода в качестве виртуального диска для раздела клиента обратитесь к 3.9 «Введение в Virtual SCSI».

Поддерживаемые конфигурации с зеркалированием VSCSI

Рисунок 5-30 демонстрирует поддерживаемые способы зеркалирования дисков для одного Virtual I/O Server.

На Virtual I/O Server вы либо настраиваете два логических тома и отображаете их на виртуальный адаптер, назначенный клиентскому разделу, или напрямую от-

бражаете hdisk на виртуальный адаптер. На клиентском разделе отображаемые устройства появятся в виде двух дисков. Клиент зеркаливает два виртуальных диска в соответствии со стандартом зеркаливания AIX 5L LVM.

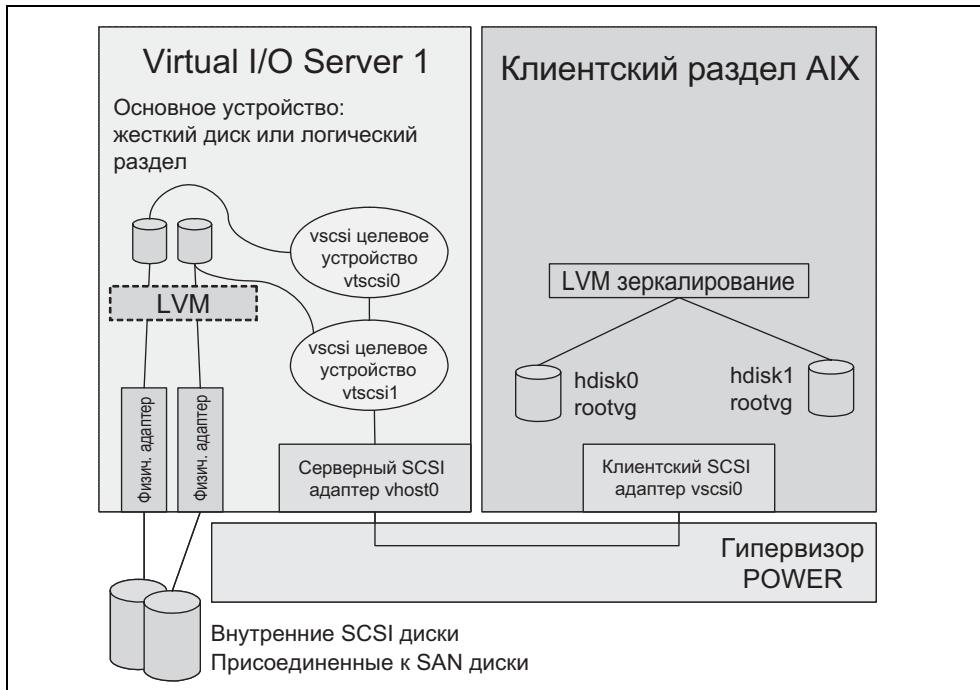


Рис. 5-30. Поддерживаемый и рекомендуемый способы зеркаливания виртуальных дисков

Важно. На момент написания книги мы не можем рекомендовать вам зеркаливать или чередовать (stripe) логические тома, используемые в качестве виртуальных дисков на Virtual I/O Server с помощью LVM. Логический том на VIOS, используемый в качестве виртуального диска, также должен не охватывать несколько дисков.

Вы можете проверить, что логический том ограничен отдельным диском с помощью команды `lslv -pv lvname`. Вывод этой команды должен вывести только один диск.

Мы рекомендуем использовать зеркаливание, чередование или объединение физических дисков с помощью LVM на клиенте VIO или использовать подобные возможности RAID-адаптеров или подсистемы хранения данных с Virtual I/O Server. Поэтому для обеспечения избыточности диска на Virtual I/O Server можно использовать аппаратный RAID5-массив. Рисунок 5-31 демонстрирует Virtual I/O Server, настроенный на SCSI RAID-адаптере. На момент написания книги поддерживались следующие адаптеры:

- ▶ PCI-X Dual Channel Ultra320 SCSI RAID Adapter (FC 5703)
- ▶ Dual Channel SCSI RAID Enablement Card (FC 5709)

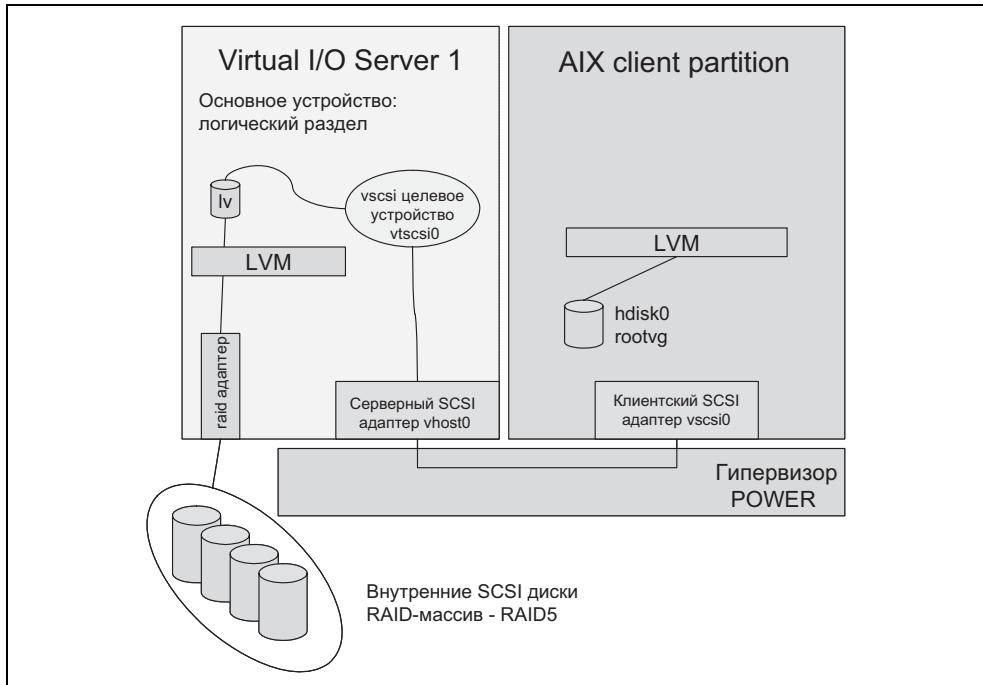


Рис. 5-31. Конфигурация RAID5, использующая RAID-адаптер на Virtual I/O Server

Внимание. Для этой конфигурации поддерживается только аппаратный RAID.

После создания RAID5-массива он появится как один из жестких дисков на Virtual I/O Server. Далее вы можете разделить большой диск на логические тома и отобразить их на ваш клиентский раздел.

При использовании этой настройки мы рекомендуем вам спланировать два дополнительных диска для установки Virtual I/O Server, который должен зеркалироваться на два диска. В противном случае отображаемые вами на клиентский раздел логические тома будут созданы на rootvg Virtual I/O Server.

Важно. Когда вы планируете зеркаливать rootvg вашего Virtual I/O Server с помощью команды `mirrorios`, будьте внимательны, так как вам нужно сделать это перед созданием дополнительных логических томов в rootvg, которые мы используем в качестве виртуальных дисков на клиентах.

При планировании нескольких Virtual I/O Server, Рисунок 5-32 показывает поддерживаемый и рекомендуемый способы зеркалирования виртуальных дисков на клиентских разделах.

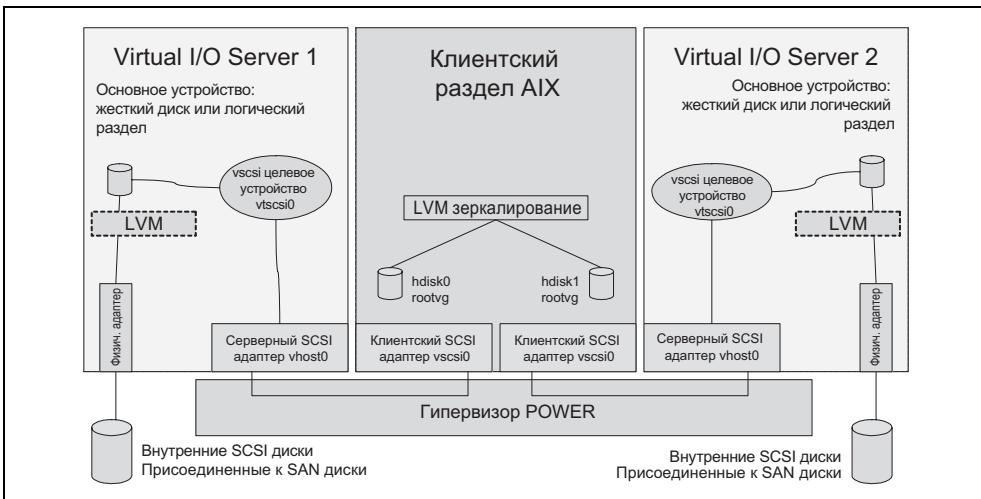


Рис. 5-32. Рекомендуемый способ зеркалирования виртуальных дисков для двух VIOS

Примечание. На интегрируемом виртуальном менеджере виртуализации (IVM) невозможно использовать более одного Virtual I/O Server. Такие конфигурации поддерживаются только на присоединенных к HMC системах.

Вы можете либо настроить логические тома на каждом сервере Virtual I/O Server и отобразить их на адаптер vhost, назначенному тому же клиентскому разделу, или напрямую отобразить hdisk на соответствующий адаптер vhost. Отображаемый логический том или диск будет настроен как hdisk на стороне клиента, каждый из которых принадлежит различным VIOS. Далее используйте LVM-зеркалирование на сайте клиента.

Поддерживаемые многопутевые конфигурации в SAN-окружении

При обсуждении поддерживаемых конфигураций с MPIO вам нужно отличать два различных сценария:

- ▶ Один Virtual I/O Server подключает LUN к SAN более чем через один путь. В этом случае вам нужно только обеспечить Virtual I/O Server многопутевым программным обеспечением.
- ▶ Несколько серверов VIOS подключены к одному LUN и являются резервными по отношению к одному клиенту. В этом случае клиентский раздел использует MPIO для доступа к виртуальному диску как к одному устройству. Вы можете также принять решение использовать многопутевое программное обеспечение на Virtual I/O Server для доступа к LUN более чем через один путь для перехвата пути и балансировки загрузки.

Примечание. Эта книга касается только решений IBM. Для получения информации о поддержке решений хранения данных от других поставщиков свяжитесь с вашим представителем IBM или напрямую с вашим поставщиком хранилищ данных для получения спецификаций на поддерживаемые конфигурации.

Поддерживаемые решения IBM TotalStorage

На момент написания книги TotalStorage Enterprise Server включал следующие модели:

- ▶ 2105 Enterprise Storage Server® (модели 800, 750, и Fxx)
- ▶ 2107 Модель 921 IBM TotalStorage DS8100
- ▶ 2107 Модель 922 IBM TotalStorage DS8300
- ▶ 2107 Модель 9A2 IBM TotalStorage DS8300
- ▶ 2107 Модель 92E IBM TotalStorage DS8000 Expansion Unit
- ▶ 2107 Модель 9AE IBM TotalStorage DS8000 Expansion Unit
- ▶ 1750 Модель 511 IBM TotalStorage DS6800
- ▶ 1750 Модель EX1 IBM TotalStorage DS6000 Expansion Unit

При подключении Virtual I/O Server к IBM TotalStorage семейства DS на Virtual I/O Server используется RDAC-драйвер. На момент написания книги IBM-семейство DS TotalStorage DS включало следующие модели:

- ▶ DS4100 (FAStT100)
- ▶ DS4200 (FAStT200)
- ▶ DS4200 (FAStT600)
- ▶ DS4400 (FAStT700)
- ▶ DS4500 (FAStT900)
- ▶ FAStT500 Storage Server

Смотрите информацию о других поддерживаемых решениях хранения данных IBM, таких как TotalStorage SAN Volume Controller, на сайте IBM:

<http://techsupport.services.ibm.com/server/vios/documentation/datasheet.html>

Использование MPIO или другого многопутевого программного обеспечения

Существует много мноногопутевых программных продуктов, поддерживаемых Virtual I/O Server. Следующие примеры сценариев демонстрируют, какое многопутевое программное обеспечение поддерживается в различных конфигурациях.

Virtual I/O Server использует несколько методов уникальной идентификации диска для каждого Virtual SCSI-диска. Это:

- ▶ Уникальный идентификатор устройства (unique device identifier, UDID), используемый MPIO
- ▶ Идентификатор тома IEEE volume identifier, используемый RDAC с семейством продуктов DS4000
- ▶ идентификатор физического тома (physical volume identifier, PVID), используемый другим многопутевым программным обеспечением

Большинство не-MPIO многопутевых программных решений используют метод PVID вместо метода UDID. Из-за того, что с методом PVID связано несколько различных форматов данных, клиент в не-MPIO окружении должен не забывать, что некоторые действия, производимые в будущем над разделом Virtual I/O Server, могут потребовать миграции данных, например некоторые типы резервирования и восстановления подключаемых дисков. Эти действия могут включать, хотя и не ограничены следующими действиями:

- ▶ Преобразованием не-MPIO окружения в MPIO
- ▶ Преобразованием метода PVID в UDID для идентификации диска
- ▶ Удаления и повторного обнаружения хранилища данных (записи в ODM)
- ▶ Обновление не-MPIO прогопутевого программного обеспечения по определенным причинам
- ▶ Возможное в будущем расширение VIO

Для всех конфигураций мы настоятельно советуем использовать MPIO с соответствующим модулем управления путями во избежание лишних хлопот при будущей миграции.

Примечание. Используйте команду `oem_setup_env` для установки и настройки многопутевого окружения на Virtual I/O Server. Все остальные конфигурации должны создаваться из интерфейса командной строки `padm` во избежание неправильных настроек.

Поддерживаемые сценарии при использовании Virtual I/O Server

Рисунок 5-33 представляет конфигурацию с MPIO и только одним Virtual I/O Server, подключенным к системам IBM TotalStorage Enterprise Server. LUN соединяется через два оптоволоконных адаптера к Virtual I/O Server для увеличения избыточности или пропускной способности.

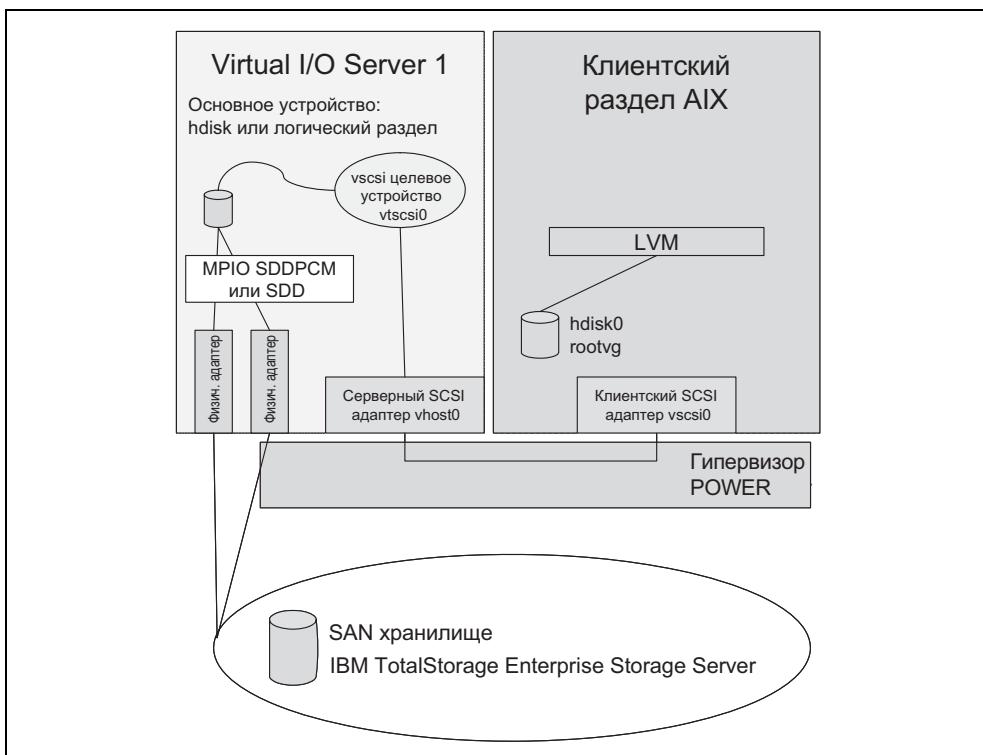


Рис. 5-33. Использование MPIO на Virtual I/O Server с IBM TotalStorage

Так как диск подключен только к одному Virtual I/O Server, вы можете создать логические тома и отобразить их на адаптер vhost, назначенный соответствующему клиентскому разделу. Также вы можете отобразить диск непосредственно на адаптер vhost.

Для присоединения IBM TotalStorage Enterprise Server только к одному серверу VIOS поддерживается MPIO с SDDPCM или SSD.

Внимание. Метод MPIO является предпочтительным для подключения систем IBM TotalStorage Enterprise Server. Виртуальное устройство диска, созданное с помощью SSD, потребует в будущем миграционных работ.

Рисунок 5-34 демонстрирует конфигурацию только с одним Virtual I/O Server для IBM TotalStorage семейства DS, использующим RDAC.

Для подключения IBM TotalStorage семейства DS поддерживается только драйвер RDAC. Так как RDAC-драйвер использует идентификатор тома IEEE, миграционные работы не требуются.

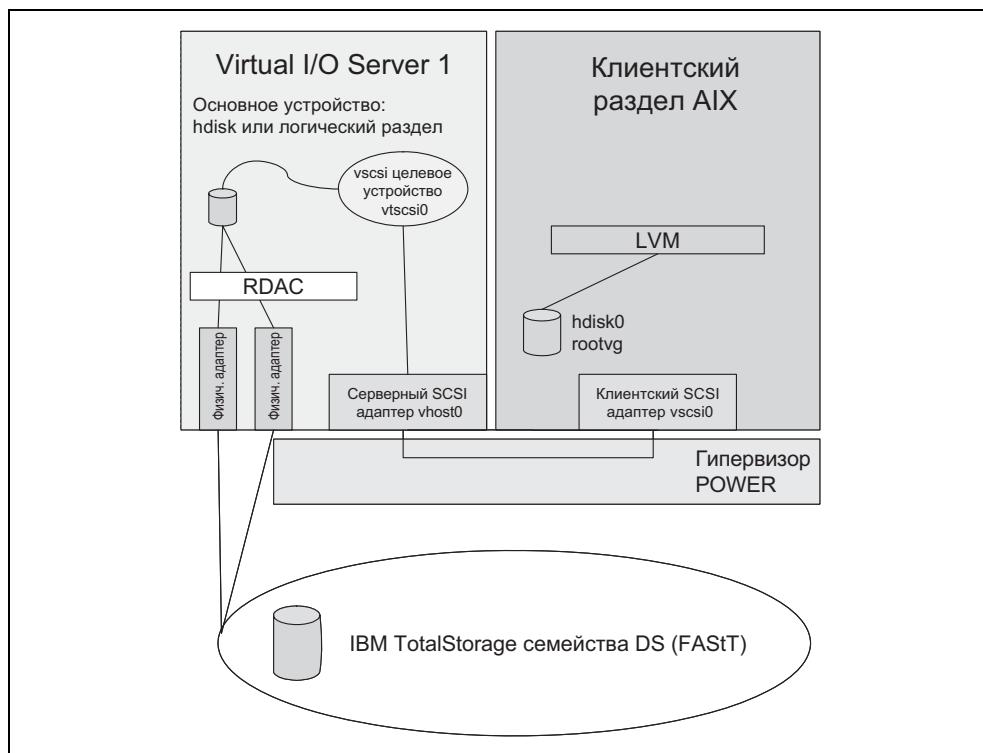


Рис. 5-34. Использование RDAC на Virtual I/O Server с IBM TotalStorage

Поддерживаемые сценарии, использующие несколько серверов ввода-вывода
Для конфигурации с несколькими серверами ввода-вывода. Рисунок 5-35 демонстрирует поддерживаемую конфигурацию для подключения систем IBM Total-

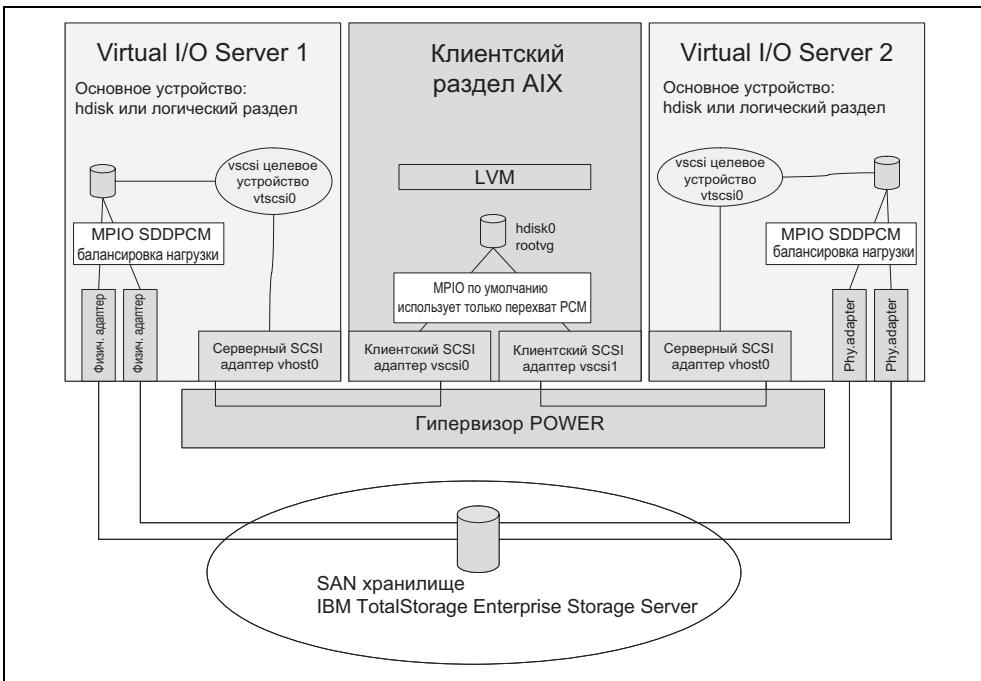


Рис. 5-35. Конфигурация для нескольких Virtual I/O Server и IBM ESS

Storage Enterprise Server при помощи многопутевого программного обеспечения на Virtual I/O Server для дополнительной избыточности и полосы пропускания.

Внимание. При использовании нескольких Virtual I/O Server и экспортации нескольких LUN на клиентский раздел поддерживается только отображение жестких дисков (логические тома не поддерживаются).

Присоединение IBM TotalStorage Enterprise Storage Server к Virtual I/O Server поддерживается только на MPIO с помощью SDDPCM.

При установленном на Virtual I/O Server пакете исправлений 6.2 или выше эта конфигурация также поддерживается при подключении IBM TotalStorage семейства DS при помощи драйвера RDAC-как показано на рисунке 5-36.

На обоих Virtual I/O Server вам нужно установить атрибут hdisk *reserve_policy* в no. Этот атрибут не позволяет Virtual I/O Server установить флаг резервирования на диске во время отображения. Компонент MPIO на клиентском разделе возьмет на себя ответственность по управлению диском.

На клиентском разделе MPIO поддерживается использование PCM по умолчанию, что позволяет управлять только политикой обработки ошибок, но не балансировкой загрузки. Активен только один путь к дискам; другие пути используются в случаях, когда активный путь становится недоступным, например когда Virtual I/O Server, обслуживающий диски через активный путь, перезагружается.

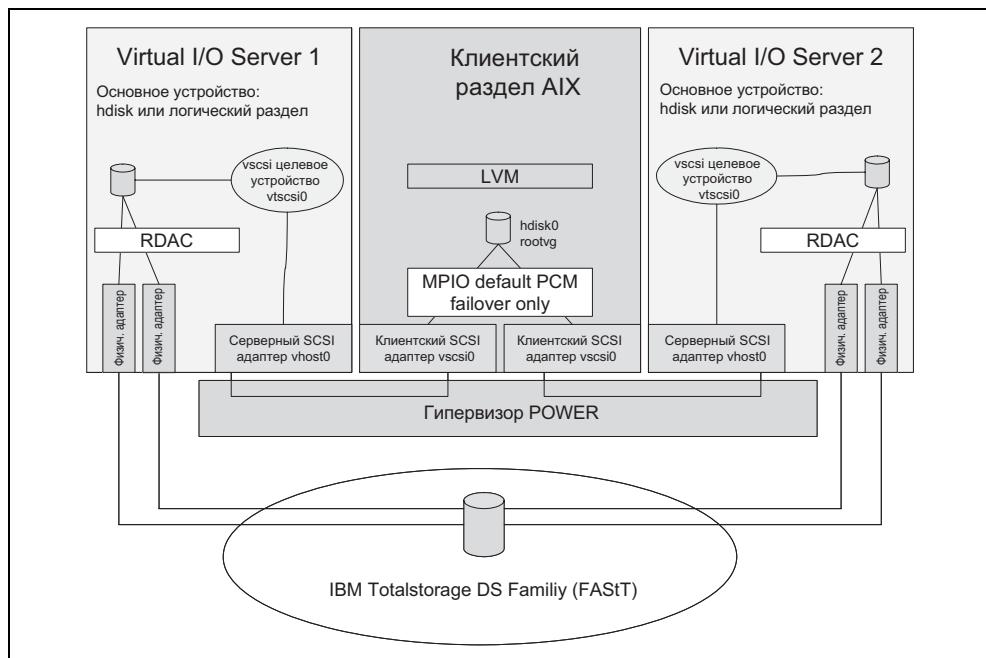


Рис. 5-36. Конфигурация для нескольких Virtual I/O Server и IBM FASTT

На стороне клиента вы можете изменить активный путь. Пользователи могут вручную настроить путь для клиентов, позволяя вам распределить нагрузку среди Virtual I/O Server. За подробными шагами по настройке обратитесь к 5.4 «Сценарий 3: MPIO на клиенте с SAN в VIOS».

Поддерживаемые Ethernet-конфигурации

Перехват общих Ethernet-адаптеров (SEA) является новой возможностью в Virtual I/O Server версии 1.2. Это позволяет вам использовать два различных подхода к конфигурации для повышения доступности для внешних сетей с помощью SEA на нескольких Virtual I/O Server.

Рисунок 5-37 показывает поддерживаемые конфигурации при использовании резервного сетевого интерфейса на клиентских разделах. Эта конфигурация ограничена только использованием с PVID.

Рисунок 5-38 демонстрирует поддержку перехвата SEA. Также эта конфигурация позволяет вам настраивать повышенную доступность к сети извне при использовании VLAN-маркеров и упрощает настройку клиентского раздела.

За дополнительными объяснениями этих двух конфигураций обратитесь к 5.1.3 «Повышение доступности для связи с внешними сетями». За другими поддерживаемыми конфигурациями для виртуального Ethernet и SEA обращайтесь к документации в InfoCenter.

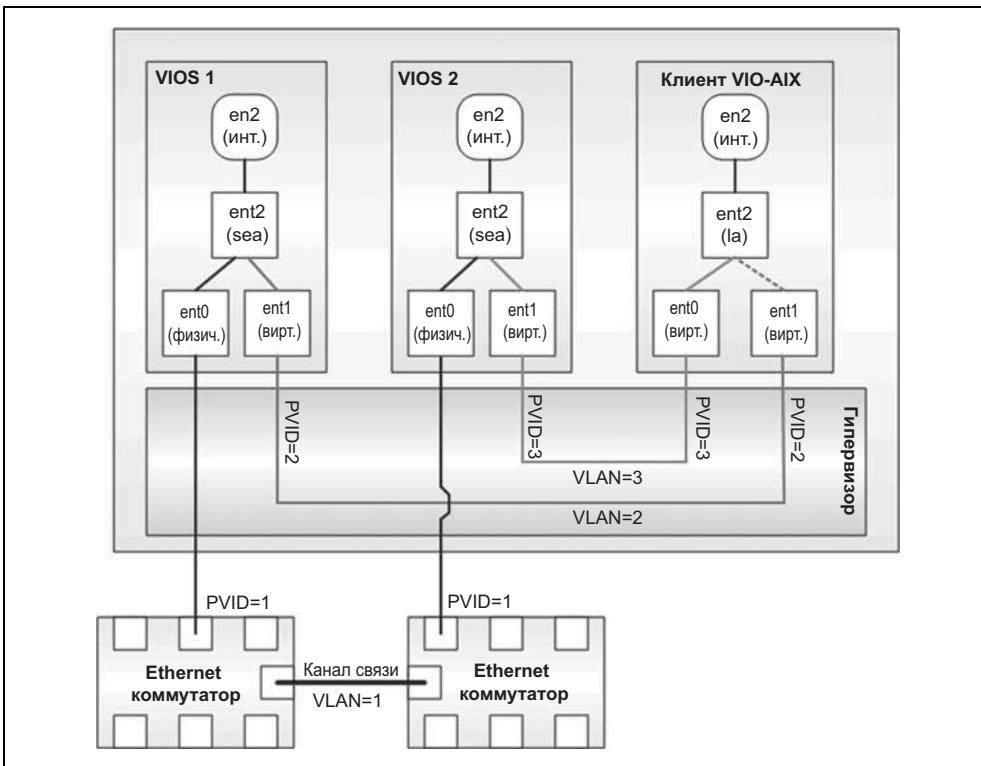


Рис. 5-37. Настройка резервного сетевого интерфейса

5.6.3. НАСМР для клиентов виртуального ввода-вывода

Программное обеспечение IBM НАСМР предоставляет вычислительную среду, которая обеспечивает работоспособность критически важных приложений и быстро устраняет последствия аппаратных и программных ошибок на отдельных серверах AIX 5L или разделах. Программное обеспечение НАСМР – это высокодоступная система, которая следит за тем, чтобы приложениям были доступны критически важные ресурсы на кластерах, состоящих из серверов или разделов. Повышенная доступность – это комбинация специального программного обеспечения с аппаратным обеспечением, минимизирующая время простоев с помощью быстрого восстановления служб, когда система, компонент или приложение отказывают.

Для того чтобы помочь клиентам увеличить доступность для их виртуализованных серверов, IBM протестировала и сертифицировала НАСМР с возможностями Advanced POWER Virtualization Virtual SCSI и виртуального Ethernet. Клиенты могут проектировать кластеры с высокой доступностью для разделов с выделенными процессорами или микроразделами, которые используют службы Virtual I/O Server для своих критических приложений. Информацию об обновлениях в поддержке НАСМР для Advanced POWER Virtualization можно найти на:

<http://www.ibm.com/support/techdocs/atstrmstr.nsf/WebIndex/FLASH10390>

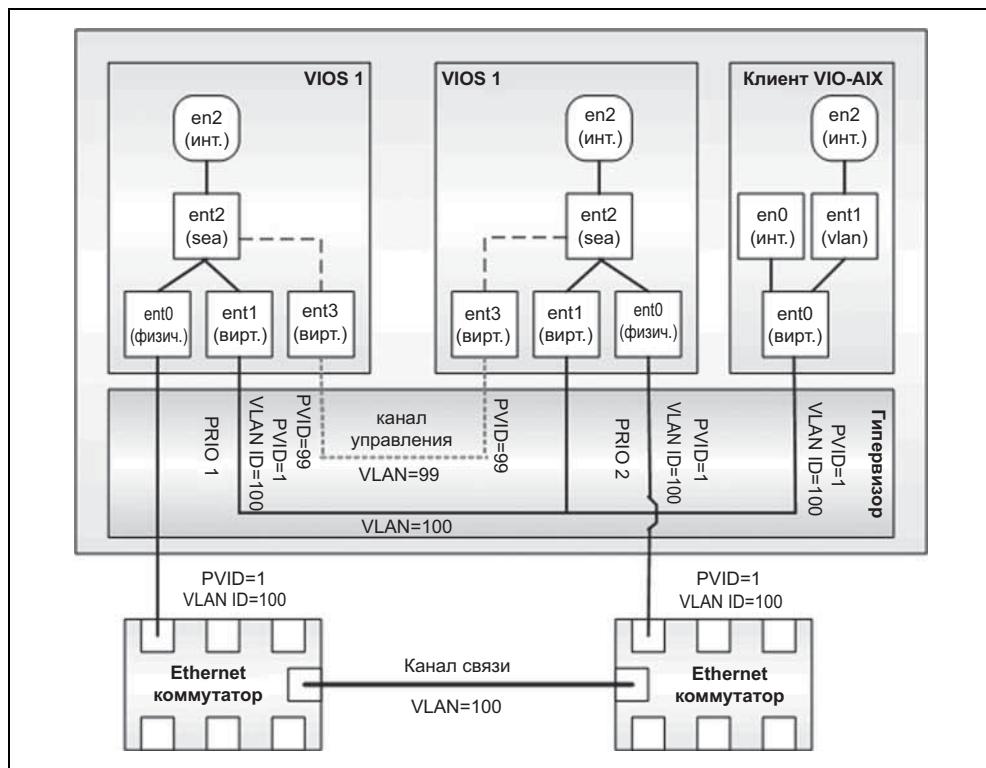


Рис. 5-38. Конфигурация перехвата SEA

Задачи, связанные с обслуживанием ресурсов хранения активного кластера на каждом сервере Virtual I/O Server, обслуживающем клиентские разделы AIX 5L (узлы кластера), обрабатываются комбинацией уровней сертифицированного программного обеспечения и конфигурацией дисковых приводов и групп томов. Наиболее важными соглашениями являются:

- ▶ Внешние хранилища информации (группа томов и логические тома) обрабатываются на уровне раздела AIX 5L. Virtual I/O Server соединяют внешние хранилища с узлами НАСМР/AIX 5L (клиентскими разделами).
- ▶ На аппаратном уровне не используется резервирование дисков. Доступ к хранилищам представляет собой комбинацию НАСМР/AIX 5L в зависимости от конфигурации.
- ▶ Все группы томов должны быть группами Enhanced Concurrent независимо от того, используются они в конкурентном режиме или нет.
- ▶ Если узлы НАСМР/AIX 5L используют доступ к группе томов через Virtual I/O Server, все узлы должны иметь доступ через Virtual I/O Server.
- ▶ Все узлы НАСМР/AIX 5L должны использовать группы томов одного типа, конкурентные либо неконкурентные.

На рисунке 5-39 показаны базовые задачи хранения на клиентских разделах AIX 5L и НАСМР.

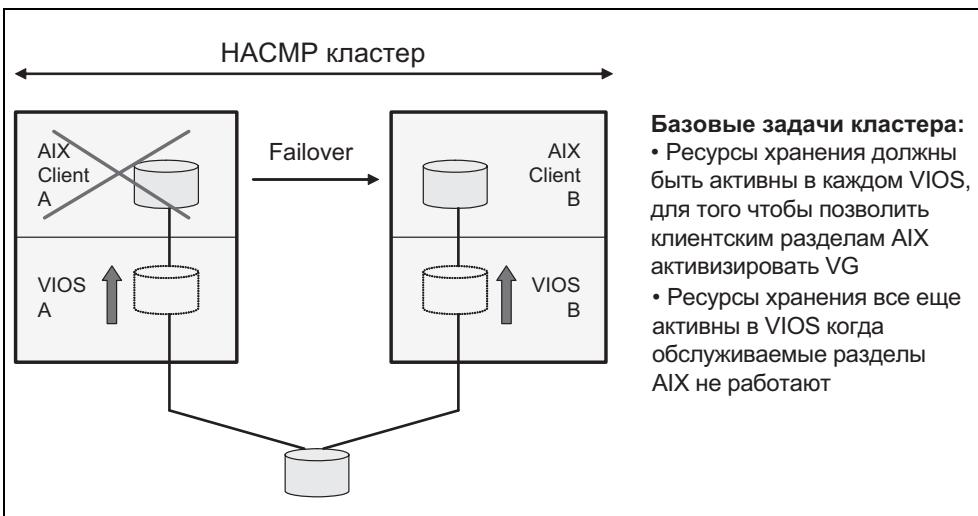


Рис. 5-39. Базовые задачи хранения на клиентских разделах AIX 5L и HACMP

Этот раздел описывал поддерживаемые конфигурации при использовании функций HACMP с виртуальным Ethernet и Virtual SCSI. Такие конфигурации должны выполняться с минимальным уровнем программного обеспечения и специфических настроек виртуального Ethernet и Virtual SCSI-дисков.

Требования к программному обеспечению

Уровни программного обеспечения, необходимого как для Virtual I/O Server, так и для клиентских разделов AIX 5L, показаны в таблице 5-5.

Таблица 5-5. Минимальные уровни программного обеспечения для настройки HACMP с APV

Программное обеспечение	Версия	Уровень поддержки	APAR/Исправления
AIX 5L V5.3	5.3	5300-02	IY70082, IY72974
набор файлов rsct.basic.hacmp	2.4.2.1		
набор файлов rsct.basic.rte	2.4.2.2		
набор файлов rsct.compat.basic.hacmp	2.4.2.0		
HACMP	5.1		IY66556
HACMP	5.2		IY68370, IY68387
HACMP	5.3		
Virtual I/O Server	1.1	Пакет исправлений 6.2	IY71303
Virtual I/O Server	1.2		

Клиент может найти исправления и наборы файлов для AIX 5L и НАСМР на:
<http://techsupport.services.ibm.com/server/vios/download/home.html>

Пакет исправлений для VIOS можно найти на:
<http://www.ibm.com/servers/@server/support/pseries/aixfixes.html>

HACMP и Virtual SCSI

Группа томов должна быть определена в режиме Enhanced Concurrent. Обычно режим Enhanced Concurrent рекомендуется для общих групп томов в кластерах НАСМР, так как тома доступны для нескольких узлов НАСМР, в результате чего перехват при отказе узла происходит быстрее.

Если на ожидающих узлах используются файловые системы, они не монтируются до момента перехвата, так как группа томов находится в режиме активного чтения/записи только на домашнем узле; ожидающие узлы владеют группами томов в пассивном режиме, что не позволяет получить доступ к логическим томам или файловой системе. Если доступ к общим томам (логические тома прямого доступа) осуществляется в режиме Enhanced Concurrent, эти тома доступны из нескольких узлов, так что доступом нужно управлять на более высоком уровне, наподобие базы данных (СУБД).

Если узел кластера осуществляет доступ к общим томам через Virtual SCSI, все узлы должны работать одинаково. Это значит, что диски не могут быть разделены между разделами с помощью Virtual SCSI и одновременно быть доступны серверу или разделу в режиме прямого доступа.

Все группы томов конструируются и поддерживаются на этих общих дисках с узлов НАСМР с помощью C-SPOC, а не с Virtual I/O Server.

Примечание. Разделы, использующие Virtual I/O-нельзя смешивать в кластере НАСМР с разделами, использующими выделенные адAPTERы, осуществляющие доступ к тем же разделяемым дискам.

HACMP и виртуальный Ethernet

Следует использовать перехват IP-адресов (IPAT) через синонимы. IPAT через замещение и перехват MAC-адресов не поддерживаются. В общем случае для поддерживающих его сетей НАСМР рекомендуется использовать IPAT через синонимы.

Все виртуальные Ethernet-интерфейсы, назначенные на НАСМР-должны считаться одноадаптерными сетями. Для этого настройте файл netmon.cf для включения списка клиентов для пингования. Он используется для мониторинга и обнаружения ошибок сетевого интерфейса. Благодаря природе виртуального Ethernet, другие механизмы обнаружения ошибок сетевого интерфейса неэффективны.

Если Virtual I/O Server имеет только один физический интерфейс для сети (вместо, например, двух интерфейсов с агрегированием Ethernet), ошибки такого физического интерфейса будут определяться с помощью НАСМР на клиентском разделе AIX 5L. Тем не менее такая ошибка изолирует узел от сети. Так что мы рекомендуем в этом случае использовать на клиентском разделе AIX 5L второй виртуальный Ethernet.

Существует несколько способов настройки клиентского раздела AIX 5L и повышения доступности ресурсов Virtual I/O Server с HACMP, хотя мы рекомендуем использовать как минимум два Virtual I/O Server для создания разделов (не обязательно ожидающих), так чтобы Virtual I/O Server не становился единой точкой отказа (SPOF). Эта конфигурация позволяет также обеспечивать онлайновые службы для Virtual I/O с улучшенным временем доступа для пользовательских приложений. На рисунке 5-40 приведен пример кластера HACMP между двумя клиентскими разделами AIX 5L.

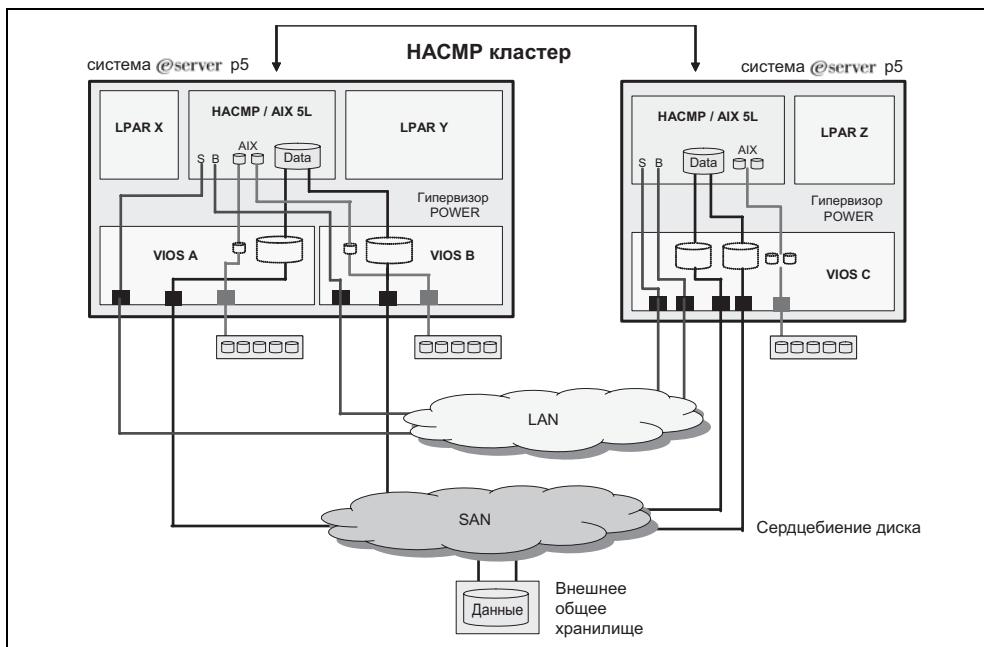


Рис. 5-40. Пример кластера HACMP между двумя клиентскими разделами AIX 5L

Рисунок 5-41 демонстрирует основные соглашения при настройке клиентского раздела AIX 5L с HACMP как части высокодоступного кластера узлов, использующих службы виртуального Ethernet и Virtual SCSI из двух Virtual I/O Server.

Для планирования и настройки кластеров HACMP с клиентскими разделами AIX 5L вам полезно будет ознакомиться со следующими публикациями:

High Availability ClusterMulti-Processing for AIX: Concepts and Facilities Guide, SC23-4864

High Availability ClusterMulti-Processing for AIX: Planning and Installation Guide, SC23-4861

Implementing Highly Available Cluster Multi-Processing Cookbook, SG24-6769

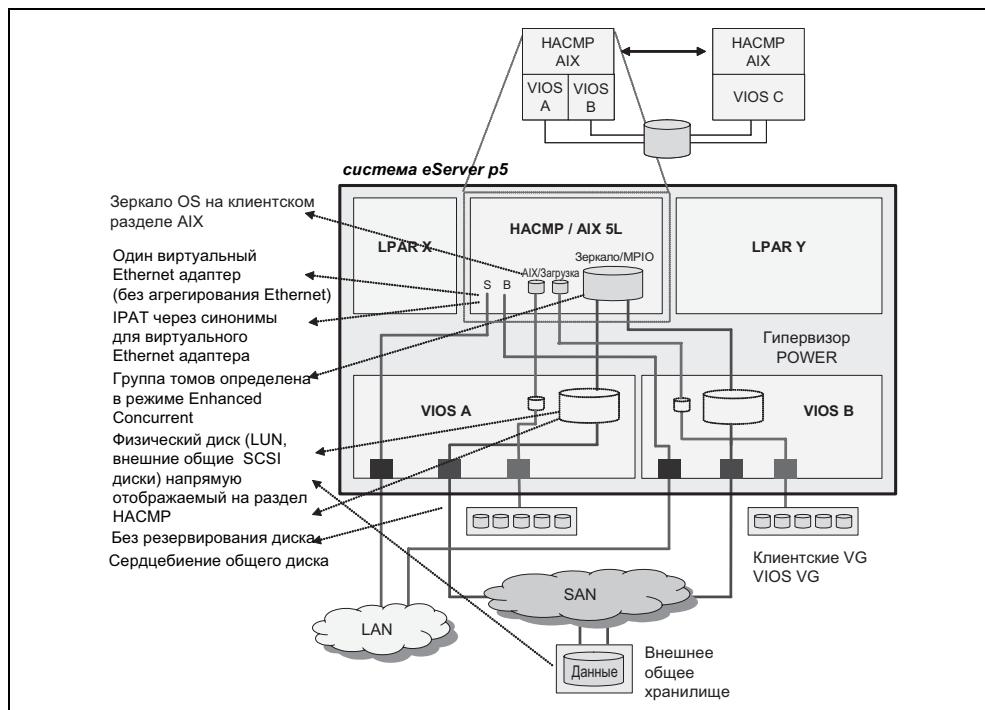


Рис. 5-41. Пример клиентского раздела AIX 5L с HACMP, использующего два VIOS

5.6.4. General Parallel Filesystem (GPFS)

На данный момент **параллельная файловая система General Parallel Filesystem (GPFS)** не поддерживается для виртуальных Ethernet или Virtual SCSI-дисков на AIX 5L или Linux. Более новую информацию вы можете прочитать в GPFS FAQ на: http://publib.boulder.ibm.com/infocenter/clresctr/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfsclustersfaq.html

6



Управление системой

В этом разделе обсуждаются следующие темы:

- ▶ Динамические операции с LPAR
- ▶ Резервное копирование и восстановление сервера Virtual I/O Server
- ▶ Пересоздание сервера Virtual I/O Server в случае невозможности восстановления
- ▶ Обслуживание сервера Virtual I/O Server
- ▶ Мониторинг виртуализованной среды
- ▶ Подбор необходимого количества ресурсов для сервера Virtual I/O Server

6.1. Динамические операции с LPAR

В этой секции объясняется, как динамически перемещать ресурсы, что может быть полезным при обслуживании вашей виртуализованной среды.

6.1.1. Динамическое удаление памяти

Следующие шаги представляют собой путь для динамического удаления памяти из логического раздела:

1. Щелкните правой кнопкой мыши на тот логический раздел, в котором вы хотите инициировать динамическую операцию. Первое окно для любой динамической операции с LPAR будет похоже на показанное на рис. 6-1.

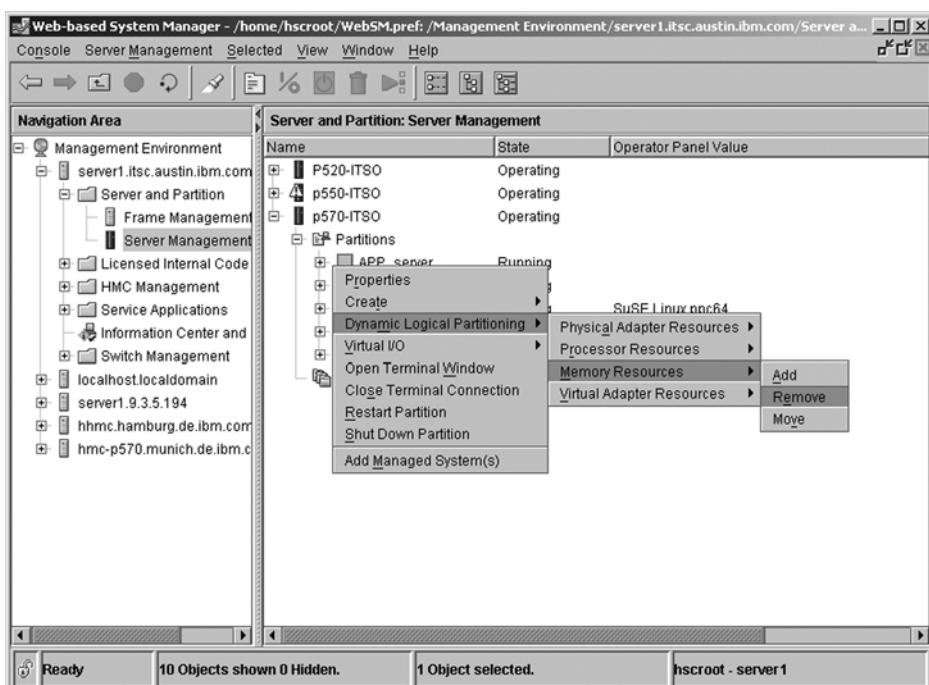


Рис. 6-1. Начальное окно динамической операции с LPAR

Настройки памяти перед динамической операцией:

```
# lsattr -El mem0
goodsize 512 Amount of usable physical memory in Mbytes False
size      512 Total amount of physical memory in Mbytes False
```

На рис. 6-2 показана закладка для уменьшения количества памяти. Сделайте необходимые изменения, как показано на рисунке.

2. Щелкните **OK** после изменений. На рис. 6-3 показано окно статуса.

Следующая команда показывает эффект удаления памяти:

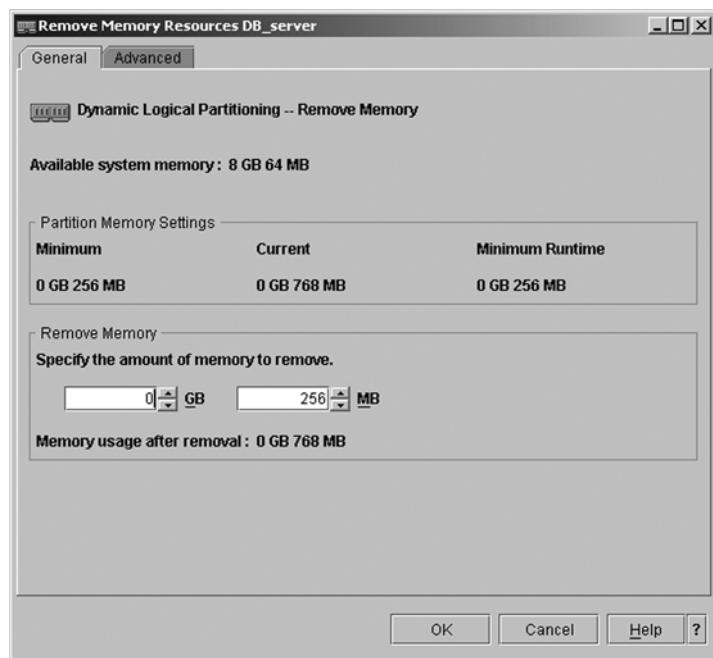


Рис. 6-2. Динамическое удаление 256 МБ памяти

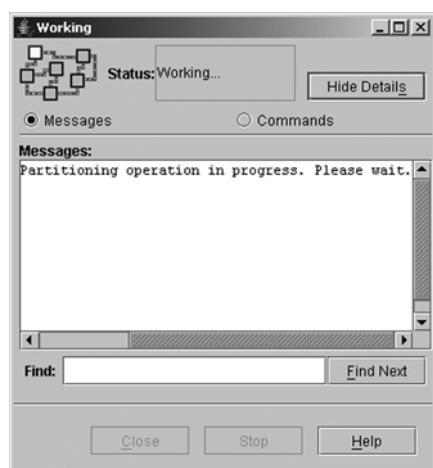


Рис. 6-3. Окно статуса

```
# lsattr -El mem0
goodsize 256 Amount of usable physical memory in Mbytes False
size      256 Total amount of physical memory in Mbytes False
```

6.1.2. Динамическое удаление виртуальных адаптеров

Следующие шаги представляют собой путь для динамического удаления виртуальных адаптеров из раздела:

1. Повторите шаг 1, но на этот раз выберите Virtual Adapter Resources, затем кнопку Add/Remove.
2. Выберите адаптер, который вы хотите удалить (рис. 6-4), и щелкните кнопку Delete.

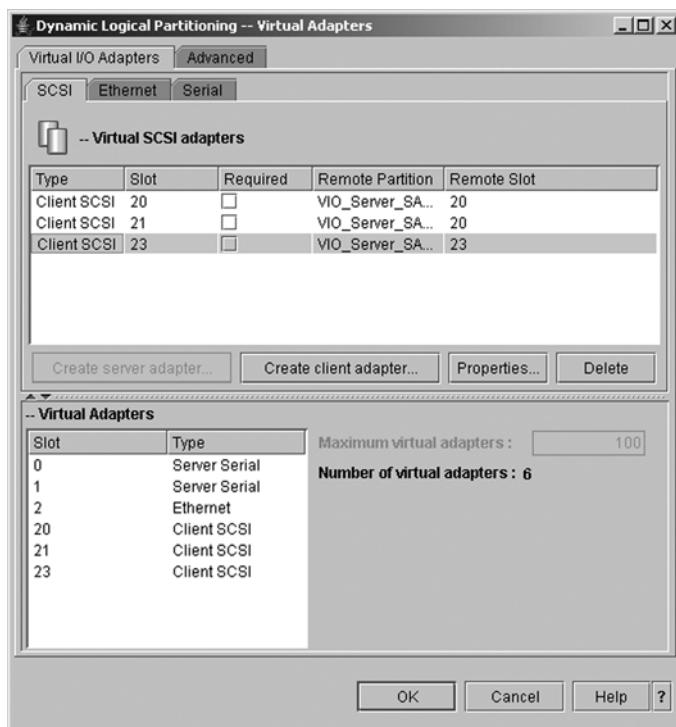


Рис. 6-4. Окно динамических операций с виртуальными адаптерами

3. По завершении щелкните OK.

6.1.3. Динамическое удаление процессоров

Следующие шаги представляют собой путь для динамического удаления процессоров:

1. Щелкните правой кнопкой мыши на тот логический раздел, в котором вы хотите инициировать динамическую операцию, как показано на рис. 6-5.
2. На рис. 6-6 показаны текущие вычислительные единицы и удаление 0.1 вычислительной единицы.
3. По завершении щелкните OK.

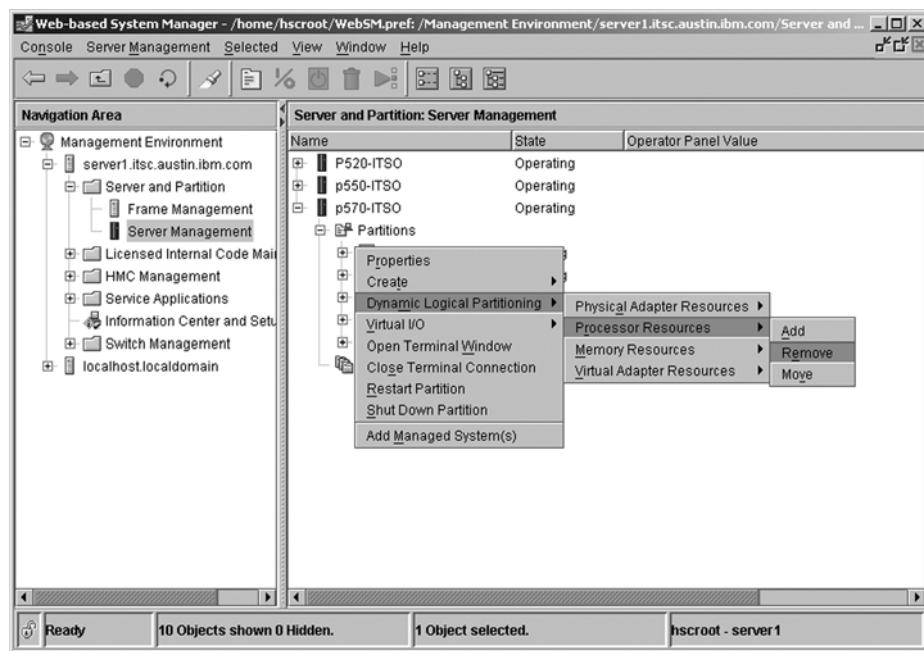


Рис. 6-5. Динамическая операция с вычислительными единицами ЦП

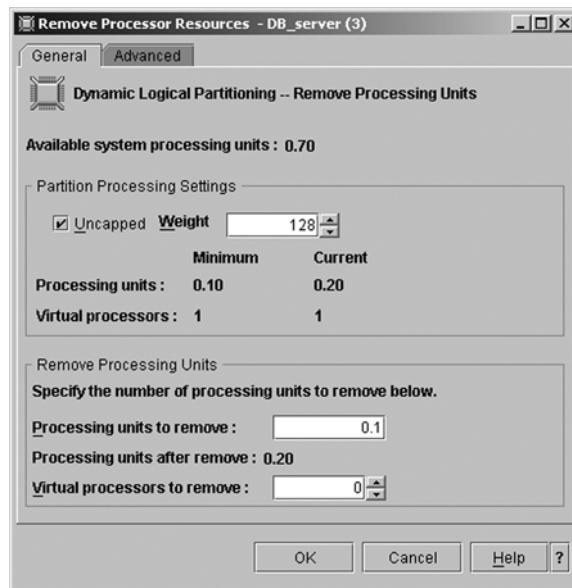


Рис. 6-6. Динамическая операция над LPAR для удаления 0.1 вычислительной единицы

6.1.4. Динамическое добавление адаптеров

Следующие шаги представляют собой один из путей для динамического добавления адаптеров:

1. Повторите шаг 1, но на этот раз выберите **Virtual Adapter Resources** и затем кнопку **Add/Remove**.
2. Следующее окно будет похоже на представленное на рис. 6-7. Щелкните на кнопку **Create client adapter**.

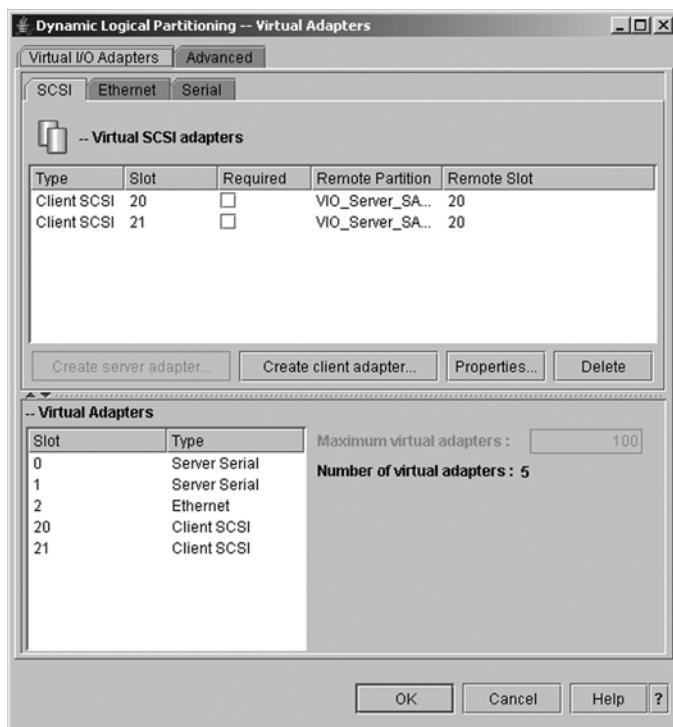


Рис. 6-7. Окно динамического добавления виртуальных адаптеров

3. На рис. 6-8 показано окно после нажатия **Create client adapter**. Введите номер слота для клиентского адаптера в верхней левой части окна. В части **Connection settings** выберите раздел сервера **Virtual I/O Server**, используя кнопку для выпадающего списка, и введите соответствующий номер слота в сервере **Virtual I/O Server** для серверного адаптера.

Внимание. Шаг 3 подразумевает, что у вас уже есть доступный серверный адаптер виртуального SCSI и вы знаете номер слота на стороне сервера. Тогда вы можете сразу щелкнуть кнопку **OK** для создания клиентского адаптера. Если нет, переходите к шагу 4 и динамически создайте серверный SCSI-адаптер.

4. В диалоговом разделе Virtual I/O Server (рис. 6-8) выберите сервер Virtual I/O Server, в котором вы хотите создать адаптер, отметьте его и щелкните кнопку Create server adapter.

Внимание. Кнопка Create server adapter в нижней правой части окна – это новая опция на HMC, которая позволяет вам динамически создавать виртуальный SCSI-адаптер на сервере Virtual I/O Server. Учтите, что, когда вы выполняете динамические операции с LPAR и щелкаете кнопку Create server adapter, операция динамически добавит адаптер в раздел, но не обновит профиль раздела. Вы должны не забыть отредактировать профиль раздела, если вы хотите сохранить это изменение и после перезагрузки.

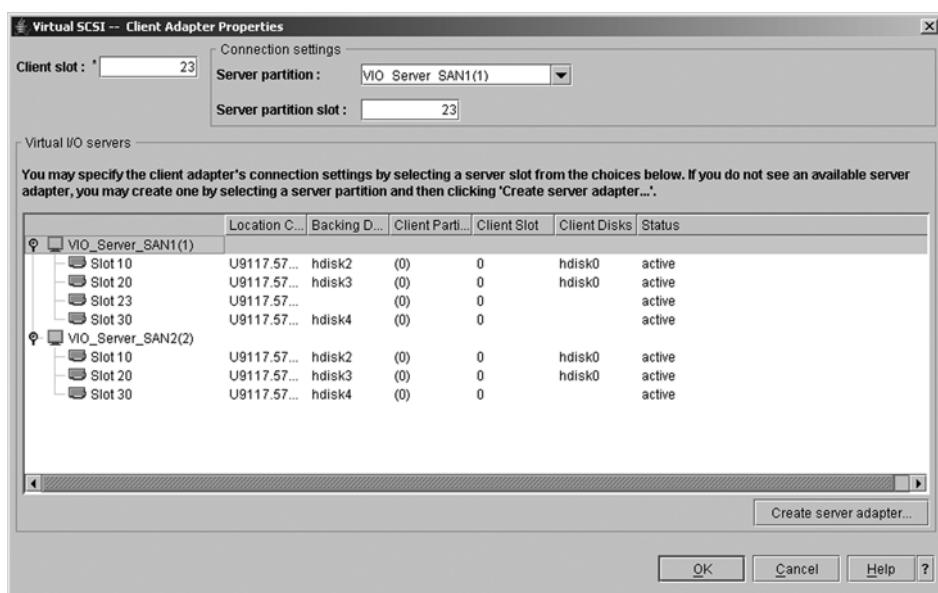


Рис. 6-8. Окно Virtual SCSI client adapter properties

5. На рис. 6-9 показано окно виртуальных адаптеров. Щелкните Create server adapter и укажите подходящий номер слота.
6. Результат этой операции показан на рис. 6-10.
7. По завершении щелкните OK.

6.1.5. Динамическое добавление памяти

Следуйте нижеуказанным шагам для динамического добавления дополнительной памяти в логический раздел:

1. Щелкните правой кнопкой мыши на логический раздел и выберите Dynamic Logical Partitioning Memory resources Add (см. рис. 6-11).
2. Выберите то количество памяти, которое вы хотите добавить (см. рис. 6-12).
3. По завершении щелкните OK. Появится окно статуса (рис. 6-13).

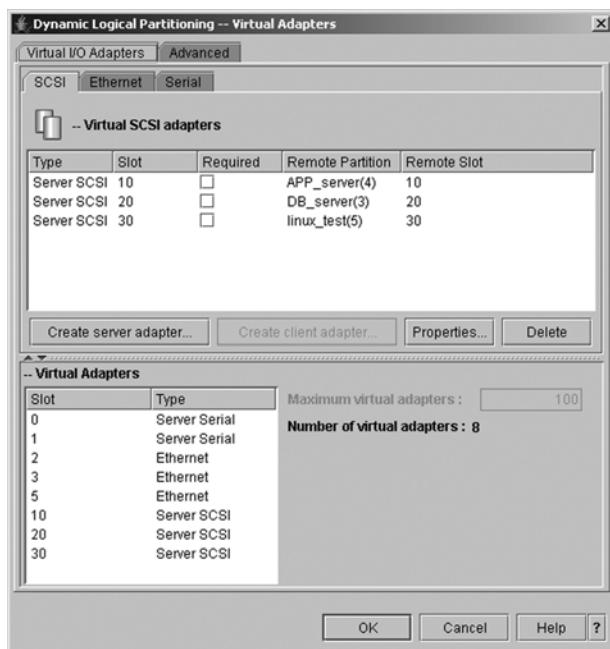


Рис. 6-9. Окно динамического создания серверного адаптера



Рис. 6-10. Окно после создания серверного адаптера virtual SCSI

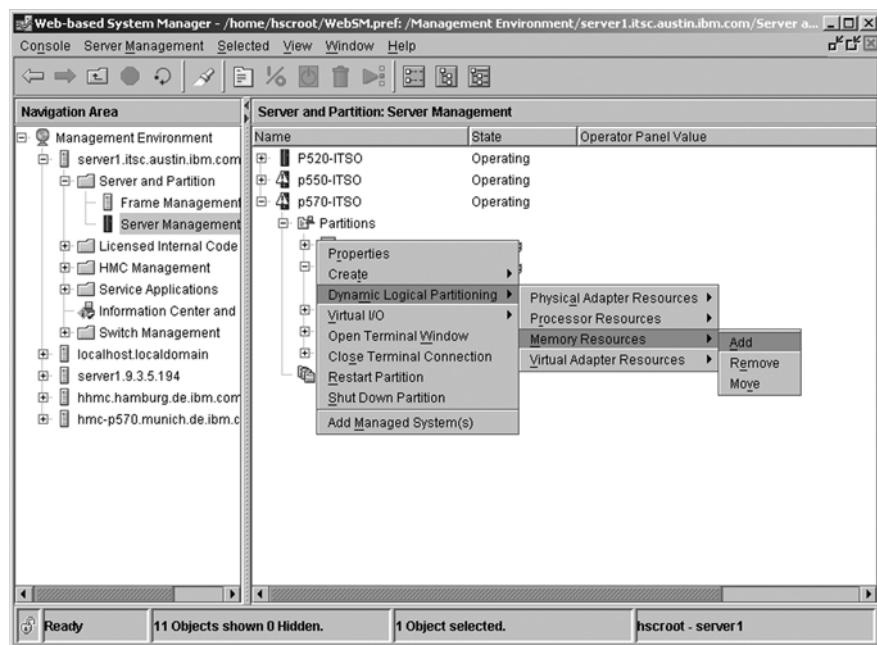


Рис. 6-11. Динамическая операция с памятью LPAR

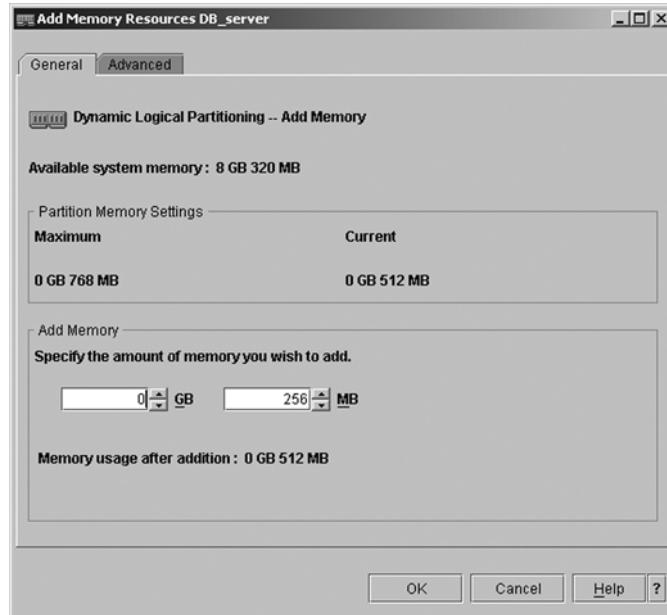


Рис. 6-12. Дополнительные 256МБ памяти будут добавлены динамически

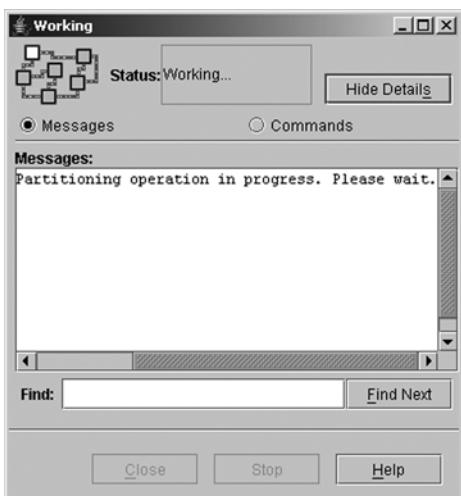


Рис. 6-13. Идет динамическая операция с LPAR

6.1.6. Просмотр топологии на HMC

Совет. На HMC есть опция помощи администраторам в просмотре топологии виртуального SCSI и виртуальной LAN на сервере VIOS.

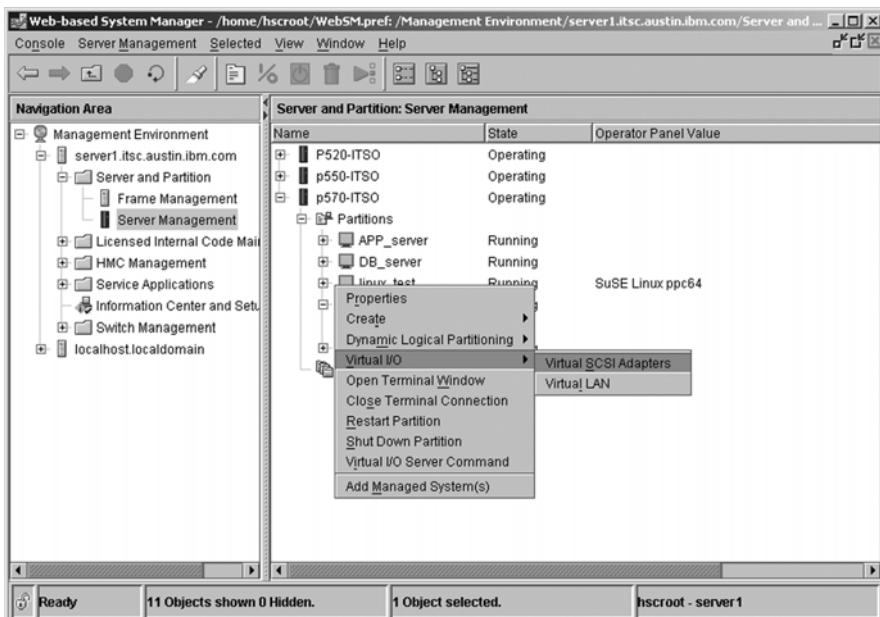


Рис. 6-14. Выбор просмотра топологии Virtual I/O

На следующих рисунках показана новая функция, имеющаяся на НМС:

- ▶ Щелкните правой кнопкой мыши на том разделе Virtual I/O Server, для которого вы хотите просмотреть топологию. Выберите Virtual I/O Virtual SCSI adapter, как показано на рис. 6-14
- ▶ На рис. 6-15 показана топология после нажатия Virtual SCSI Adapter.

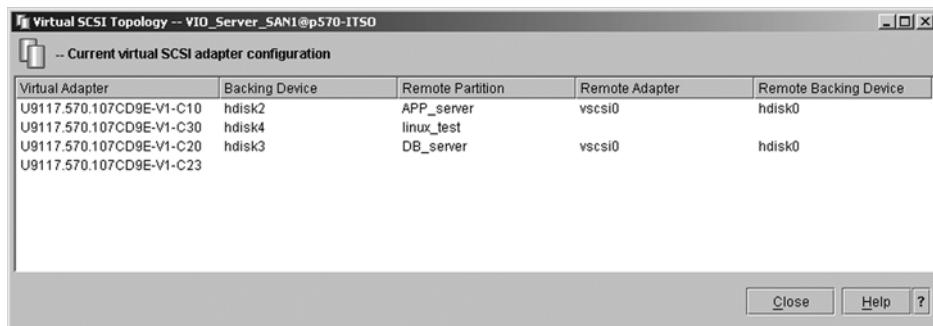


Рис. 6-15. Топология виртуальных SCSI-адаптеров на VIOS

6.2. Резервное копирование и восстановление сервера Virtual I/O Server

В этом разделе показан способ резервного копирования и восстановления сервера Virtual I/O Server.

6.2.1. Резервное копирование Virtual I/O Server

Интерфейс командной строки Virtual I/O Server предоставляет команду `backupios` для создания установочного образа группы томов `rootvg` либо на загрузочную ленту, либо на многотомный CD/DVD. Возможно также создание установочного образа NIM на файловой системе. Дополнительно информация о настройках раздела, включая виртуальные устройства ввода-вывода, должна быть сохранена на НМС. Резервная копия данных клиента должна быть создана с клиентского раздела для обеспечения целостности данных.

Команда `backupios` поддерживает следующие устройства резервного копирования:

- ▶ Лента
- ▶ Файловая система
- ▶ CD
- ▶ DVD

В следующих трех разделах мы сделаем резервную копию Virtual I/O Server, используя ленту, DVD и файловую систему. Резервное копирование на CD не представлено, так как оно выполняется аналогично копированию на DVD.

6.2.2. Резервное копирование на ленту

В примере 6-1 показан результат выполнения команды `backupios` с ключом `-tape`.

Пример 6-1. Резервное копирование на ленту

```
$ backupios -tape /dev/rmt0
Creating information file (/image.data) for rootvg..
Creating tape boot image.....
Creating list of files to back up.
Backing up 23622 files.....
23622 of 23622 files (100%)
0512-038 mksysb: Backup Completed Successfully.
bosboot: Boot image is 26916 512 byte blocks.
bosboot: Boot image is 26916 512 byte blocks.
```

Результат этой команды – загрузочная лента, которая дает возможность легкого восстановления Virtual I/O Server, как показано в разделе «Восстановление с ленты».

6.2.3. Резервное копирование на DVD

Существует два типа DVD, которые могут быть использованы для резервного копирования: DVD-RAM и DVD-R. DVD-RAM может поддерживать форматы `-cd` и `-udf`, а DVD-R поддерживает только формат `-cd`. Устройство DVD не может быть виртуализовано и присвоено клиентскому разделу во время выполнения `backupios`. Удалите устройство из клиента и отображений виртуального SCSI на сервере перед выполнением резервного копирования.

В примере 6-2 показано, как выполнить резервное копирование VIOS на DVD-RAM.

Пример 6-2. Резервное копирование на DVD

```
$ backupios -cd /dev/cd0 -udf
Creating information file for volume group datapool..
Creating list of files to back up.
Backing up 6 files
6 of 6 files (100%)
0512-038 savevg: Backup Completed Successfully.
Backup in progress. This command can take a considerable amount of time
to complete, please be patient...
Initializing mkcd log: /var/adm/ras/mkcd.log...
Verifying command parameters...
Creating image.data file...
Creating temporary file system: /mkcd/mksysb_image...
Creating mksysb image...
Creating list of files to back up.
Backing up 27129 files.....
27129 of 27129 files (100%)
0512-038 mksysb: Backup Completed Successfully.
Populating the CD or DVD file system...
```

```
Copying backup to the CD or DVD file system...
.....
.....
.....
.....
Building chrp boot image...
Removing temporary file system: /mkcd/mksysb_image...
```

6.2.4. Резервное копирование на файловую систему

Результат работы команды `backupios` – образ резервной копии в формате tar. Этот файл будет сохранен в каталоге, указанном флагом `-file`. В примере 6-3 показаны создание подкатаога `backup_loc` и выполнение команды `backupios`.

Пример 6-3. Резервное копирование на файловую систему

```
$ mkdir /home/padmin/backup_loc
$ backupios -file /home/padmin/backup_loc
Creating information file for volume group datapool..
Creating list of files to back up.
Backing up 6 files
6 of 6 files (100%)
0512-038 savevg: Backup Completed Successfully.
Backup in progress. This command can take a considerable amount of time
to complete, please be patient...
```

Команда `ls` показывает, что резервное копирование успешно создало tar-файл:

```
-rw-r--r-- 1 root staff 653363200 Jan 11 21:13 nim_resources.tar
```

Когда требуется только образ резервной копии VIOS, а не файл с ресурсами NIM, используется флаг `-mksysb`. С этим флагом требуется указание файла и каталога для сохранения резервной копии, например:

```
$ backupios -file /home/padmin/backup_loc/VIOS.img -mksysb
```

Для резервных копий используется флаг `-file`, резервирование на смонтированную файловую систему NFS полезно для сохранения образа резервной копии на другую систему.

6.2.5. Восстановление сервера Virtual I/O Server

В следующих разделах показан процесс восстановления сервера Virtual I/O Server в зависимости от выбранного вами формата резервной копии.

Восстановление с ленты

Для восстановления сервера Virtual I/O Server с ленты загрузите раздел Virtual I/O Server в меню SMS и выберите ленточное устройство в качестве загрузочного. Затем продолжайте, как при обычной установке AIX 5L. На рис. 6-16 показан выбор ленточного устройства для восстановления с предварительно созданной на ленте резервной копии.

```

PowerPC Firmware
Version SF220_010
SMS 1.5 (c) Copyright IBM Corp. 2000,2003 All rights reserved.

-----
Select Device
Device Current Device
Number Position Name
1. - SCSI Tape
   ( loc=U787A.001.DN200XX-P1-T10-L1-L0 )

-----

Navigation keys:
M = return to Main Menu
ESC key = return to previous screen      X = eXit System Management Services
-----
Type the number of the menu item and press Enter or select Navigation Key:1_
MA* a                                     p1 25/076

```

Рис. 6-16. Выбор ленточного устройства для восстановления Virtual I/O Server

Восстановление с DVD

Для восстановления Virtual I/O Server с DVD загрузите раздел Virtual I/O Server в меню SMS и выберите устройство DVD в качестве загрузочного. Продолжите, как при обычной установке AIX 5L.

Восстановление с файловой системы

Восстановление Virtual I/O Server из резервной копии на файловой системе выполняется командой `installios` с НМС или AIX 5L. Для восстановления tar-файл должен быть расположен либо на НМС, либо на доступном по NFS каталоге, либо на DVD. Чтобы сделать созданной командой `backupios` tar-файл доступным для восстановления, мы выполнили следующие шаги:

1. Создали каталог `backup`, используя команду `mkdir /home/padmin/backup`.
 2. Проверили, что сервер NFS экспортирует файловую систему, командой `showmount nfs_server`.
 3. Смонтировали проэкспортированную файловую систему NFS в каталог `backup`.
 4. Скопировали tar-файл, созданный в разделе 6.2.4 «Резервное копирование на файловую систему» в подмонтированный каталог NFS, используя следующую команду:
- ```
$ cp /home/padmin/backup_loc/nim_resources.tar /home/padmin/backup
```

На этой стадии резервная копия готова для восстановления на раздел Virtual I/O Server, используя команду `installios` с HMC или раздела AIX 5L, являющегося сервером NIM. Процедура восстановления остановит раздел Virtual I/O Server, если он еще работает. Ниже представлена справка по использованию команды:

```
hscroot@server1:~> installios -?
installios: usage: installios [-s managed_sys -S netmask -p partition
 -r profile -i client_addr -d source_dir -m mac_addr
 -g gateway [-P speed] [-D duplex] [-n] [-l language]]
 | -u
```

В команде `installios` ключ `-s managed_sys` требует имени HMC, ключ `-p partition` требует имени раздела VIOS и ключ `-r profile` требует профиля раздела, который вы хотите использовать для загрузки раздела VIOS во время восстановления.

Если вы не указали флаг `-m` и не указали MAC-адрес восстанавливаемого сервера VIOS, восстановление займет большее время, так как команда `installios` остановит сервер VIOS и загрузит его в SMS для определения MAC-адреса. Ниже представлен пример использования команды:

```
hscroot@server1:~> installios -s p550-ITSO -S 255.255.255.0 -p testvios2 -r
default -i 9.3.5.125 -d 9.3.5.126:/export_fs -m 00:02:55:d3:dc:34 -g 9.3.5.41
```

Выполняя эту команду, NIMOL на HMC выполняет процесс NIM и монтирует проэкспортированную файловую систему для обработки тар-файла `backupios`, созданного на VIOS предварительно. Затем HMC NIM продолжает нормальную установку VIOS и одна финальная перезагрузка раздела завершит установку. Для восстановления с сервера NIM AIX 5L используется та же команда:

```
installios -?
Usage: installios [-h hmc -s managed_sys -p partition -r profile]
 -S netmask -i client_addr -g gateway -d source_dir
 [-P speed] [-D duplex] [-n] [-N] [-l language] [-L location] | -u[f|U]
```

Эта команда может быть выполнена либо из командной строки, либо используя команду `smitty installios`. Для выполнения этой команды должен быть настроен SSH между сервером NIM и HMC. В примере 6-4 показан вывод команды `smitty installios`, где указывается та же информация, что и при запуске `installios` из командной строки.

#### Пример 6-4. Меню AIX 5L smitty installios

---

```
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP] [Entry Fields]
* Select or specify software source [cd0] +
to initialize environment

HMC Name []
Managed System Name []
Partition Name []
Partition Profile Name []
Primary Network Install Interface []

* IP Address Used by Machine []
* Subnetmask Used by Machine []
* Default Gateway Used by Machine []
```

|                                             |         |   |
|---------------------------------------------|---------|---|
| Network Speed Setting                       | [100]   | + |
| Network Duplex Setting                      | [full]  | + |
| Language to Install                         | [en_US] | + |
| Configure Client Network After Installation | [yes]   | + |
| ACCEPT I/O Server License Agreement?        | [no]    | + |
| [BOTTOM]                                    |         |   |

---

Команда AIX 5L `installios` использует команды SSH на НМС для остановки LPAR и его перезагрузки для получения MAC-адреса и запуска установки NIM. Для выполнения такого восстановления может использоваться резервная копия на DVD или в файле.

### 6.3. Пересоздание сервера Virtual I/O Server

В этом разделе показано, что делать при отсутствии работоспособных устройств резервного копирования или образов резервных копий. В этом случае вы должны установить новый Virtual I/O Server.

В дальнейшем мы подразумеваем, что определения разделов VIOS и всех клиентов все еще доступны на НМС. Мы покажем, как мы пересоздали нашу конфигурацию сети и SCSI.

В дополнение к обычным резервным копиям, созданным командой `backupios`, мы рекомендуем сохранять следующие настройки:

- ▶ Сетевые настройки  
Команды: `netstat -state`, `netstat -routinfo` и `netstat -dev Device -attr`
- ▶ Все SCSI устройства – физические и логические тома  
Команды: `lspv`, `lsvg` и `lsvg -lv VolumeGroup`
- ▶ Все физические и логические адаптеры  
Команды: `lsdev -type adapter`
- ▶ Соответствие между физическими, логическими и виртуальными устройствами  
Команды: `lsmmap -all` и `lsmmap -all -net`

Имея эту информацию, вы можете перенастроить ваш сервер Virtual I/O Server вручную. В следующих разделах мы покажем команды, которые мы использовали для получения необходимой информации, и команды для пересоздания конфигурации. Важная информация из вывода команд выделена шрифтом. В зависимости от вашей среды команды могут отличаться от показанных в примерах.

Для того чтобы начать пересоздание сервера Virtual I/O Server, вы должны знать, какие диски использовались для самого Virtual I/O Server, а какие – для групп томов виртуального ввода-вывода.

- ▶ Команда `lspv` показывает нам, что Virtual I/O Server был установлен на `hdisk0`. Первый шаг – установка нового Virtual I/O Server с установочного носителя на диск `hdisk0`. Эта команда может быть запущена из окружения команды `diag` или из другого окружения AIX 5L:

|                     |                  |                           |        |
|---------------------|------------------|---------------------------|--------|
| <code>hdisk0</code> | 00cddedc01300ed3 | <code>rootvg</code>       | active |
| <code>hdisk1</code> | 00cddedc143815fb | <code>None</code>         |        |
| <code>hdisk2</code> | 00cddedc4d209163 | <code>client_disks</code> | active |
| <code>hdisk3</code> | 00cddedc4d2091f8 | <code>datavg</code>       | active |

См. 4-3, «Установка Virtual I/O Server» для деталей процедуры установки. Пересоздание сервера Virtual I/O Server выполнено в два шага:

1. Пересоздание конфигурации SCSI.
2. Пересоздание конфигурации сети.

### 6.3.1. Пересоздание конфигурации SCSI

Команда `lspv` также показывает нам, что есть две дополнительные группы томов, расположенные на Virtual I/O Server (`client_disks` и `datavg`):

|        |                  |              |        |
|--------|------------------|--------------|--------|
| hdisk0 | 00cddedc01300ed3 | rootvg       | active |
| hdisk1 | 00cddedc143815fb | None         |        |
| hdisk2 | 00cddedc4d209163 | client_disks | active |
| hdisk3 | 00cddedc4d2091f8 | datavg       | active |

Следующие команды импортируют эту информацию в ODM новой системы Virtual I/O Server:

```
importvg -vg client_disks hdisk2
importvg -vg datavg hdisk3
```

В примере 6-5 мы смотрим на отображение между логическими и физическими томами и серверными адаптерами виртуального SCSI.

#### Пример 6-5. lsmap -all

```
$ lsmap -all
SVSA Physloc Client Partition
ID

vhost0 U9111.520.10DDEDC-V4-C30 0x00000000
VTD vtscsi2
LUN 0x8100000000000000
Backing device data1lv
Physloc
SVSA Physloc Client Partition
ID

vhost2 U9111.520.10DDEDC-V4-C10 0x00000006
VTD vtscsi0
LUN 0x8100000000000000
Backing device rootvg_ztest0
Physloc
VTD vtscsi1
LUN 0x8200000000000000
Backing device hdisk1
Physloc U787A.001.DNZ00XY-P1-T10-L4-L0
```

Серверный адаптер виртуального SCSI vhost0 (определенный в слоте 30 на HMC) отображается в логический том data1lv через виртуальное целевое устройство (Virtual Target Device) vtscsi2.

Серверный адаптер виртуального SCSI vhost2 имеет два виртуальных целевых устройства vtscsi0 и vtscsi1. Они отображаются в логический том rootvg\_ztest0 и физический том hdisk1 на vhost2 (определенный в слоте 10 на НМС).

Следующие команды использовались нами для создания необходимых нам виртуальных целевых устройств:

```
mkvdev -vdev datalv -vadapter vhost0
mkvdev -vdev rootvg_ztest0 -vadapter vhost2
mkvdev -vdev hdisk1 -vadapter vhost2
```

**Замечание.** Имена виртуальных целевых устройств генерируются автоматически, если вы не указываете их явно, используя флаг -dev команды mkvdev.

### 6.3.2. Пересоздание конфигурации сети

После успешного пересоздания конфигурации SCSI мы пересоздадим конфигурацию сети.

Команда netstat -state показывает нам, что en2 – единственный активный сетевой адаптер<sup>1</sup>:

| Name | Mtu   | Network | Address        | Ipkts | Ierrs | Opkts | Oerrs | Coll |
|------|-------|---------|----------------|-------|-------|-------|-------|------|
| en2  | 1500  | link#2  | 0.d.60.a.58.a4 | 2477  | 0     | 777   | 0     | 0    |
| en2  | 1500  | 9.3.5   | 9.3.5.147      | 2477  | 0     | 777   | 0     | 0    |
| lo0  | 16896 | link#1  |                | 153   | 0     | 158   | 0     | 0    |
| lo0  | 16896 | 127     | 127.0.0.1      | 153   | 0     | 158   | 0     | 0    |
| lo0  | 16896 | ::1     |                | 153   | 0     | 158   | 0     | 0    |

Используя команду lsmap -all -net, мы определили, что ent2 – общий адаптер Ethernet (SEA), отображающий физический адаптер ent0 в виртуальный адаптер ent1:

| SVEA           | Physloc                    |
|----------------|----------------------------|
| -----          |                            |
| ent1           | U9111.520.10DDEDC-V4-C2-T1 |
| SEA            | ent2                       |
| Backing device | ent0                       |
| Physloc        | U787A.001.DNZ00XY-P1-C2-T1 |

Информация о маршрутизаторе по умолчанию определяется командой netstat -routinfo:

| Routing tables                               | Destination | Gateway   | Flags | Wt | Policy | If  | Cost | Config_Cost |
|----------------------------------------------|-------------|-----------|-------|----|--------|-----|------|-------------|
| Route Tree for Protocol Family 2 (Internet): | default     | 9.3.5.41  | UG    | 1  | -      | en2 | 0    | 0           |
|                                              | 9.3.5.0     | 9.3.5.147 | UHSb  | 1  | -      | en2 | 0    | =>          |
|                                              | 9.3.5/24    | 9.3.5.147 | U     | 1  | -      | en2 | 0    | 0           |
|                                              | 9.3.5.147   | 127.0.0.1 | UGHS  | 1  | -      | lo0 | 0    | 0           |
|                                              | 9.3.5.255   | 9.3.5.147 | UHSb  | 1  | -      | en2 | 0    | 0           |
|                                              | 127/8       | 127.0.0.1 | U     | 1  | -      | 0   | 0    | 0           |

<sup>1</sup> Точнее, сетевой интерфейс. Он находится на сетевом адаптере ent2. Прим. науч. ред.

Для получения маски подсети мы использовали команду `lsdev -dev en2 -attr`:

|                                                                                                                                                                                          |                                                                                      |                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|----------------------------|
| <code>netmask</code>                                                                                                                                                                     | <code>255.255.255.0 Subnet Mask</code>                                               | <code>True</code>          |
| Последняя необходимая нам информация – это виртуальный адаптер по умолчанию и PVID по умолчанию для общего адаптера Ethernet. Это показано командой <code>lsdev -dev ent2 -attr</code> : |                                                                                      |                            |
| <code>attribute value description</code>                                                                                                                                                 |                                                                                      | <code>user_settable</code> |
| <code>pvid</code>                                                                                                                                                                        | <code>1 PVID to use for the SEA device</code>                                        | <code>True</code>          |
| <code>pvid_adapter</code>                                                                                                                                                                | <code>ent1 Default virtual adapter to use for non-VLAN-tagged packets</code>         | <code>True</code>          |
| <code>real_adapter</code>                                                                                                                                                                | <code>ent0 Physical adapter associated with the SEA</code>                           | <code>True</code>          |
| <code>virt_adapters</code>                                                                                                                                                               | <code>ent1 List of virtual adapters associated with the SEA (comma separated)</code> | <code>True</code>          |

Следующие команды пересоздают нашу сетевую конфигурацию:

```
mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 1
mktcpip -hostname p51iosrv2 -inetaddr 9.3.5.147 -interface en2 -start -netmask
255.255.255.0 -gateway 9.3.5.41
```

Эти шаги завершают базовое пересоздание сервера Virtual I/O Server.

## 6.4. Обслуживание сервера Virtual I/O Server

В этом разделе обсуждается обслуживание сервера VIOS. Наши темы включают устройства с горячей заменой и восстановление после сбоя диска как на VIOS, так и на разделе клиента. Мы начнем с методов обновления ПО VIOS на активной конфигурации VIOS без прерывания доступности сервисов VIOS для его клиента.

### 6.4.1. Параллельные обновления ПО VIOS

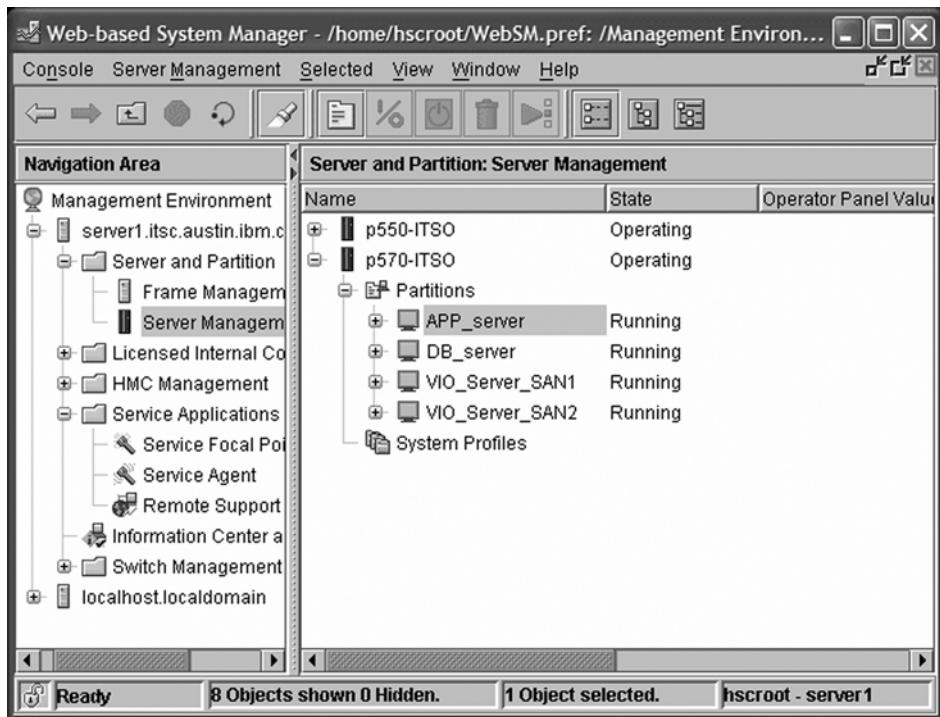
В этом разделе рассказывается о шагах для обновления серверов Virtual I/O Server в среде нескольких VIOS. Простое обновление ПО показано в разделе «Обновление VIOS».

Мы настроили два раздела Virtual I/O Server и два клиентских раздела для того, чтобы показать обновление Virtual I/O Server в среде с зеркалированием и MPIO. В этой среде возможно обеспечение функционирования клиентов 24x7 без необходимости прерывания работы клиента во время обновления ПО VIOS.

Клиентский раздел DB\_Server настроен на наличие одного виртуального диска с сервера VIO\_Server\_SAN1 и другого виртуального диска с сервера VIO\_Server SAN2. В этом клиентском разделе мы настроили зеркалирование через LVM. Подробности действий для настройки этого сценария вы можете найти в разделе 5.2, «Сценарий 1: Зеркалирование через LVM».

Клиентский раздел APP\_server настроен как клиент MPIO. Оба сервера Virtual I/O Server предоставляют пути к одному диску SAN для этого клиентского раздела. Подробности действий для настройки этой конфигурации вы можете найти в разделе 5.4 «Сценарий 3: MPIO на клиенте с SAN в VIOS».

Конфигурация системы показана на рис. 6-17.



**Рис. 6-17.** Конфигурация для параллельного обновления ПО

В этой тестовой настройке оба сервера Virtual I/O Server имеют установленный уровень (oslevel) Version 1.1.2.62 и будут обновлены до Version 1.2, используя Fix Pack 7. Обновления Virtual I/O Server можно найти на URL:

<http://techsupport.services.ibm.com/server/vios/download/home.html>

На клиентских разделах выполняется AIX 5L V5.3 ML 03. Шаги, показанные здесь, типичны для всех обновлений Virtual I/O Server в окружении нескольких VIOS, вне зависимости от уровня ОС.

До того как мы начнем обновлять Virtual I/O Server, мы проверим наши клиентские разделы<sup>1</sup>, чтобы убедиться, что не было предыдущих действий, таких как перезагрузки, приведшие к устареванию (stale) разделов<sup>2</sup> на клиентах или устареванию путей. В этих случаях могут быть неработоспособны пути или зеркала LVM.

В среде с зеркалированием выполните следующие шаги для подготовки:

- 1 . В клиентском разделе, в нашем случае DB\_Server, выполните команду `lsvg` и проверьте, есть ли устаревшие (stale) разделы:

```
lsvg -l rootvg
rootvg:
```

<sup>1</sup> Имеются в виду LPAR. Прим. науч. ред.

<sup>2</sup> Имеются в виду Stale physical partitions (PP) – компонент LVM. Здесь значение слова «раздел» («partition») зависит от контекста. Прим. науч. ред.

| LV NAME | TYPE    | LPs | PPs | PVs | LV STATE     | MOUNT POINT |
|---------|---------|-----|-----|-----|--------------|-------------|
| hd5     | boot    | 1   | 2   | 2   | closed/syncd | N/A         |
| hd6     | paging  | 4   | 8   | 2   | open/syncd   | N/A         |
| hd8     | jfs2log | 1   | 2   | 2   | open/stale   | N/A         |
| hd4     | jfs2    | 1   | 2   | 2   | open/stale   | /           |
| hd2     | jfs2    | 5   | 10  | 2   | open/stale   | /usr        |
| hd9var  | jfs2    | 1   | 2   | 2   | open/stale   | /var        |
| hd3     | jfs2    | 1   | 2   | 2   | open/stale   | /tmp        |
| hd1     | jfs2    | 1   | 2   | 2   | open/stale   | /home       |
| hd10opt | jfs2    | 1   | 2   | 2   | open/stale   | /opt        |

2. Если вы видите устаревшие разделы, ресинхронизируйте группу томов, используя команду `varyonvg`. Убедитесь, что Virtual I/O Server работает и все отображения находятся в корректном состоянии:

```
varyonvg rootvg
```

3. После выполнения этой команды проверьте еще раз, используя команду `lsvg`, что группа томов синхронизована:

```
lsvg -l rootvg
```

rootvg:

| LV NAME | TYPE    | LPs | PPs | PVs | LV STATE     | MOUNT POINT |
|---------|---------|-----|-----|-----|--------------|-------------|
| hd5     | boot    | 1   | 2   | 2   | closed/syncd | N/A         |
| hd6     | paging  | 4   | 8   | 2   | open/syncd   | N/A         |
| hd8     | jfs2log | 1   | 2   | 2   | open/syncd   | N/A         |
| hd4     | jfs2    | 1   | 2   | 2   | open/syncd   | /           |
| hd2     | jfs2    | 5   | 10  | 2   | open/syncd   | /usr        |
| hd9var  | jfs2    | 1   | 2   | 2   | open/syncd   | /var        |
| hd3     | jfs2    | 1   | 2   | 2   | open/syncd   | /tmp        |
| hd1     | jfs2    | 1   | 2   | 2   | open/syncd   | /home       |
| hd10opt | jfs2    | 1   | 2   | 2   | open/syncd   | /opt        |

В среде MPIO выполните следующие шаги для подготовки:

1. В клиентском разделе, в нашем случае APP\_Server, проверьте состояние, используя команду `lspath`. Оба пути должны быть в состоянии enabled. Если один из них пропал или в состоянии disabled, проверьте возможные причины. Если ранее перезагружался Virtual I/O Server и атрибут `health_check` не установлен attribute, вам может понадобиться его установить.
2. Выполнить следующую команду для проверки путей:

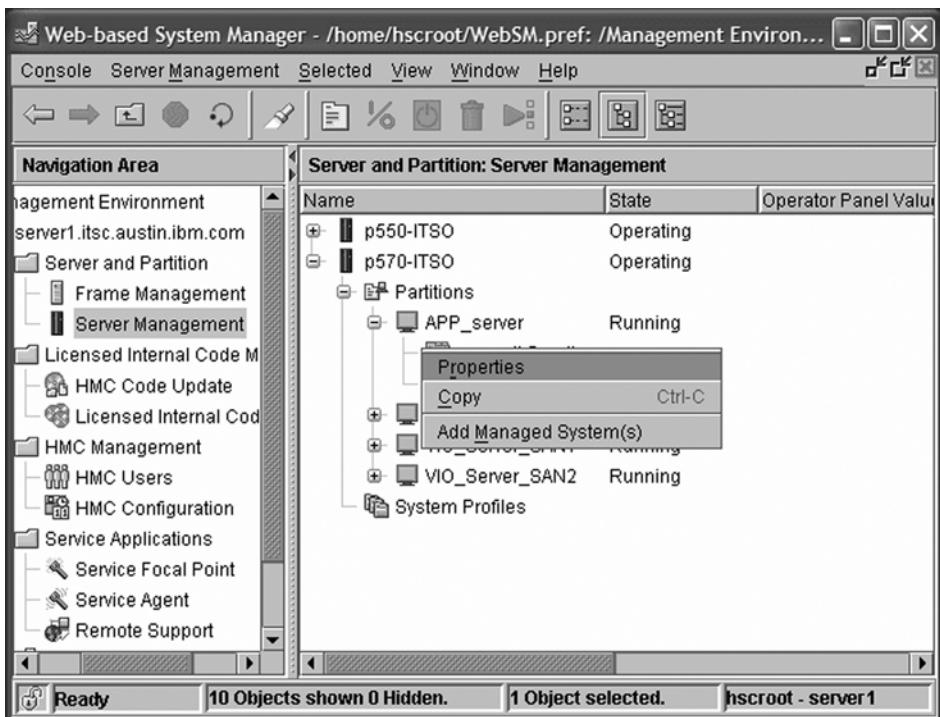
```
lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

3. После подтверждения того, что оба пути функционируют без проблем, отключите путь к тому Virtual I/O Server, который будет обновляться первым. В этом примере мы начнем с обновления VIO\_Server\_SAN1. Команда `lspath` показывает нам, что один путь подключен через адаптер `vscsi0`, а другой – через адаптер `vscsi1`.

Выполните команду `lscfg` для определения номера слота адаптера `vscsi0`:

```
lscfg -vl vscsi0
vscsi0 U9117.570.107CD9E-V4-C10-T1 Virtual SCSI Client Adapter
Device Specific.(YL).....U9117.570.107CD9E-V4-C10-T1
```

- Для обнаружения того, к какому Virtual I/O Server подключен адаптер vscsi0, посмотрите активный профиль клиентского раздела на НМС.
- Идите в раздел Server Management и выберите активный профиль клиентского раздела, как показано на рис. 6-18.



**Рис. 6-18.** Свойства профайла на НМС

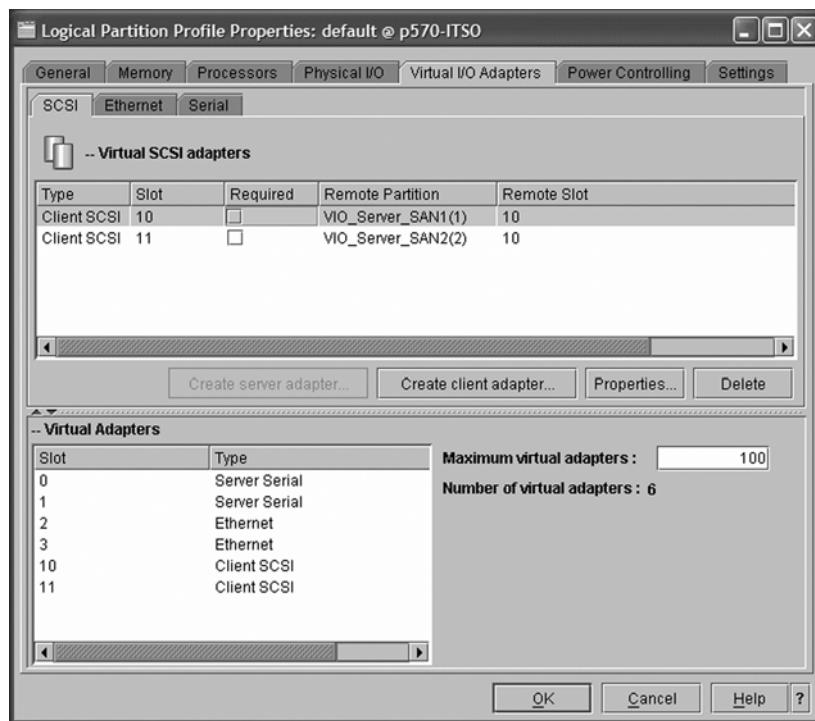
Щелкните правой кнопкой на профайле и выберите Properties. В окне Properties выберите закладку Virtual I/O Adapters.

На рис. 6-19 показано, что виртуальный адаптер SCSI в слоте 10 подключен к разделу VIO\_Server-SAN1.

- Отключите путь vscsi0 для hdisk0, используя команду chpath.

```
lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
chpath -l hdisk0 -p vscsi0 -s disable
paths Changed
lspath
Disabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

Теперь мы можем начать обновление раздела VIO\_Server\_SAN1.



**Рис. 6-19.** Окно свойств LPAR

1. В разделе VIO\_Server\_SAN1 мы настроили устройство CD-ROM и вставили в него диск с обновлением.

Для обновления выполните команду `updateios`; результаты показаны в примере 6-6.

#### **Пример 6-6.** Вывод команды `updateios`

```
$ updateios -dev cd0 -install -accept

installp PREVIEW: installation will not actually occur.

+-----+
 Pre-installation Verification...
+-----+
Verifying selections...done
Verifying requisites...done
Results...
WARNINGS

Problems described in this section are not likely to be the source of any
immediate or serious failures, but further actions may be necessary or
desired.
```

Already Installed

The following filesets which you selected are either already installed or effectively installed through superseding filesets.

|                                           |                                   |
|-------------------------------------------|-----------------------------------|
| devices.pci.4f11c800.rte 5.3.0.0          | # 2-Port Asynchronous EIA-232 ... |
| devices.pci.4f11c800.diag 5.3.0.0         | # 2-port Asynchronous EIA-232 ... |
| devices.pci.1410d002.com 5.3.0.0          | # Common PCI iSCSI TOE Adapter... |
| devices.pci.1410d002.rte 5.3.0.0          | # 1000 Base-TX PCI-X iSCSI TOE... |
| devices.pci.1410d002.diag 5.3.0.0         | # 1000 Base-TX PCI-X iSCSI TOE... |
| devices.pci.1410cf02.rte 5.3.0.0          | # 1000 Base-SX PCI-X iSCSI TOE... |
| devices.pci.1410cf02.diag 5.3.0.0         | # 1000 Base-SX PCI-X iSCSI TOE... |
| invscount.ldb 2.2.0.2                     | # Inventory Scout Logic Database  |
| csm.diagnostics 1.4.1.0                   | # Cluster Systems Management P... |
| csm.core 1.4.1.0                          | # Cluster Systems Management Core |
| csm.client 1.4.1.0                        | # Cluster Systems Management C... |
| csm.dsh 1.4.1.0                           | # Cluster Systems Management Dsh  |
| devices.pci.2b101a05.diag 5.3.0.20        | # GXT120P Graphics Adapter Dia... |
| devices.pci.14103302.diag 5.3.0.20        | # GXT135P Graphics Adapter Dia... |
| devices.pci.14105400.diag 5.3.0.20        | # GXT500P/GXT550P Graphics Ada... |
| devices.pci.2b102005.diag 5.3.0.20        | # GXT130P Graphics Adapter Dia... |
| devices.pci.isa.rte 5.3.0.10              | # ISA Bus Bridge Software (CHRP)  |
| devices.common.IBM.fddi.rte 5.3.0.10      | # Common FDDI Software            |
| devices.common.IBM.tokenring.rte 5.3.0.10 | # Common Token Ring Software      |
| devices.common.IBM.hdlc.rte 5.3.0.10      | # Common HDLC Software            |
| devices.pci.331121b9.rte 5.3.0.10         | # IBM PCI 2-Port Multiprotocol... |
| devices.pci.14107c00.diag 5.3.0.10        | # PCI ATM Adapter (14107c00) D... |
| devices.pci.1410e601.diag 5.3.0.10        | # IBM Cryptographic Accelerato... |
| devices.pci.14101800.diag 5.3.0.10        | # PCI Tokenring Adapter Diagn...  |
| devices.pci.14103e00.diag 5.3.0.10        | # IBM PCI Tokenring Adapter (1... |
| devices.pci.23100020.diag 5.3.0.10        | # IBM PCI 10/100 Mb Ethernet A... |
| devices.pci.22100020.diag 5.3.0.10        | # PCI Ethernet Adapter Diagnos... |
| devices.pci.14102e00.diag 5.3.0.10        | # IBM PCI SCSI RAID Adapter Di... |
| devices.pci.14105e01.diag 5.3.0.10        | # 622Mbps ATM PCI Adapter Diag... |
| devices.isa_sio.chrp.ecp.rte 5.3.0.10     | # CHRP IEEE1284 Parallel Port ... |
| devices.serial.gio.X11 5.3.0.10           | # AIXwindows Serial Graphics I... |
| devices.serial.sb1.X11 5.3.0.10           | # AIXwindows 6094-030 Spacebal... |
| devices.serial.tablet1.X11 5.3.0.10       | # AIXwindows Serial Tablet Inp... |
| perl.libext 2.1.0.10                      | # Perl Library Extensions         |
| devices.pci.77101223.rte 5.3.0.10         | # PCI FC Adapter (77101223) Ru... |
| devices.pci.5a107512.rte 5.3.0.10         | # IDE Adapter Driver for Promi... |
| bos.txt.spell 5.3.0.10                    | # Writer's Tools Commands         |
| devices.pci.14107d01.X11 5.3.0.10         | # AIXwindows GXT300P Graphics ... |
| devices.pci.14106e01.X11 5.3.0.10         | # AIXwindows GXT4000P Graphics... |
| devices.pci.14107001.X11 5.3.0.10         | # AIXwindows GXT6000P Graphics... |
| devices.pci.14108e00.X11 5.3.0.10         | # AIXwindows GXT3000P Graphics... |
| devices.pci.1410b800.X11 5.3.0.10         | # AIXwindows GXT2000P Graphics... |
| devices.pci.14101b02.X11 5.3.0.10         | # AIXwindows GXT6500P Graphics... |
| devices.pci.14101c02.X11 5.3.0.10         | # AIXwindows GXT4500P Graphics... |
| devices.pci.14106902.diag 5.3.0.10        | # 10/100/1000 Base-TX PCI-X Ad... |
| devices.pci.1410ff01.diag 5.3.0.10        | # 10/100 Mbps Ethernet PCI Ada... |

```

devices.pci.00100100.com 5.3.0.10 # Common Symbios PCI SCSI I/O ...
devices.pci.14100401.diag 5.3.0.10 # Gigabit Ethernet-SX PCI Adap...
devices.pci.1410ba02.rte 5.3.0.10 # 10 Gigabit-SR Ethernet PCI-X...
devices.common.IBM.ide.rte 5.3.0.10 # Common IDE I/O Controller So...

NOTE: Base level filesets may be reinstalled using the "Force"
option (-F flag), or they may be removed, using the deinstall or
"Remove Software Products" facility (-u flag), and then reinstalled.

<< End of Warning Section >>

SUCSESSES

Filesets listed in this section passed pre-installation verification
and will be installed.

Mandatory Fileset Updates

(being installed automatically due to their importance)
bos.rte.install 5.3.0.30 # LPP Install Commands

<< End of Success Section >>

FILESET STATISTICS

262 Selected to be installed, of which:
 1 Passed pre-installation verification
 50 Already installed (directly or via superseding filesets)
 211 Deferred (see *NOTE below)

 1 Total to be installed

*NOTE The deferred filesets mentioned above will be processed after the installp
 update and its requisites are successfully installed.

RESOURCES

Estimated system resource requirements for filesets being installed:
 (All sizes are in 512-byte blocks)
 Filesystem Needed Space Free Space
 /usr 7256 887280

 TOTAL: 7256 887280

NOTE: "Needed Space" values are calculated from data available prior
to installation. These are the estimated resources required for the
entire operation. Further resource checks will be made during
installation to verify that these initial estimates are sufficient.

End of installp PREVIEW. No apply operation has actually occurred.

Continue the installation [y|n]?


```

---

Ведите у и нажмите Enter для начала обновления. Вывод команды updateios  
очень детальный и убран из этого примера.

- После успешного обновления вы должны снова принять условия лицензии:

```
$ license -accept
```

Для проверки обновления запустите команду `ioslevel`:

```
$ ioslevel
1.2.0.0
```

- Перезагрузите Virtual I/O Server.

После того как Virtual I/O Server снова запустится, нам необходимо зайти в клиентские разделы, использующие первый обновленный VIOS для ресинхронизации группы томов на отзеркальном клиентском разделе и для изменения статуса пути на клиентском разделе MPIO. Это обновит зеркало, – включив в него результаты дисковых операций, которые проводились во время обновления ПО.

В отзеркальном окружении выполните команду `lsvg` на клиентских разделах для проверки статуса группы томов:

```
lsvg -l rootvg
rootvg:
 LV NAME TYPE LPs PPs PVs LV STATE MOUNT POINT
 hd5 boot 1 2 2 closed/syncd N/A
 hd6 paging 4 8 2 open/syncd N/A
 hd8 jfs2log 1 2 2 open/stale N/A
 hd4 jfs2 1 2 2 open/syncd /
 hd2 jfs2 5 10 2 open/syncd /usr
 hd9var jfs2 1 2 2 open/stale /var
 hd3 jfs2 1 2 2 open/syncd /tmp
 hd1 jfs2 1 2 2 open/syncd /home
 hd10opt jfs2 1 2 2 open/syncd /opt
```

Запустите команду `varyonvg` для синхронизации группы томов:

```
varyonvg rootvg
lsvg -l rootvg
rootvg:
 LV NAME TYPE LPs PPs PVs LV STATE MOUNT POINT
 hd5 boot 1 2 2 closed/syncd N/A
 hd6 paging 4 8 2 open/syncd N/A
 hd8 jfs2log 1 2 2 open/syncd N/A
 hd4 jfs2 1 2 2 open/syncd /
 hd2 jfs2 5 10 2 open/syncd /usr
 hd9var jfs2 1 2 2 open/syncd /var
 hd3 jfs2 1 2 2 open/syncd /tmp
 hd1 jfs2 1 2 2 open/syncd /home
 hd10opt jfs2 1 2 2 open/syncd /opt
```

- В окружении MPIO зайдите в ваши клиентские разделы и проверьте статус пути командой `lspath`:

```
lspath
Disabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

- Включите путь `vscsi0` и отключите путь `vscsi1`:

```
chpath -l hdisk0 -p vscsi0 -s enable
paths Changed
lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
chpath -l hdisk0 -p vscsi1 -s disable
paths Changed
lspath
Enabled hdisk0 vscsi0
Disabled hdisk0 vscsi1
```

6. Обновите второй Virtual I/O Server и повторите вышеуказанные шаги.
7. После завершения обновления и перезагрузки раздела второго Virtual I/O Server зайдите в клиентский раздел для синхронизации rootvg, используя команду `varyonvg`, и включения пути, используя команду `chpath`, как показано в примерах выше.

Это завершает обновление конфигурации VIOS с избыточностью с сохранением доступности клиентского приложения.

#### 6.4.2. Устройства с горячей заменой

Аналогично AIX 5L, VIOS включает функцию поддержки горячего подключения устройств, таких как диски и адаптеры PCI, в сервер и активизации их в разделе без перезагрузки.

До начала на HMC должен быть присвоен пустой слот разделу VIOS. Эта задача может быть выполнена через динамические операции с LPAR, но профиль раздела VIOS также должен быть обновлен, чтобы новый адаптер остался сконфигурированным на VIOS после перезагрузки.

Для того чтобы начать, используйте команду `diagmenu` для захода в диагностическое меню VIOS. Это меню очень похоже на диагностическое меню AIX 5L и дает вам те же четыре возможности в начальном экране:

- ▶ Diagnostic Routines
- ▶ Advanced Diagnostic Routines
- ▶ Task Selection
- ▶ Resource Selection

Функция Hot Plug Tasks находится под пунктом меню Task Selection. В этом пункте меню есть пункты PCI hot plug tasks, RAID hot plug devices и SCSI and SCSI RAID hot plug manager, как показано в примере 6-7.

##### Пример 6-7. Меню команды diagmenu Hot Plug Task

---

```
Hot Plug Task
801004
Move cursor to desired item and press Enter.
PCI Hot Plug Manager
RAID Hot Plug Devices
SCSI and SCSI RAID Hot Plug Manager
```

---

Меню PCI используется для добавления, идентификации или замены адаптеров PCI, назначенных сейчас системе VIOS. Пункт RAID hot plug devices используется для добавления хранилищ RAID, подключаемых к SCSI RAID-адаптеру. Меню SCSI and SCSI RAID manager используется для добавления или замены дисков и настройки SCSI RAID.

### **Добавление адаптера PCI с горячей заменой**

В тестовом VIOS была следующая конфигурация адаптеров PCI:

| # Slot                  | Description                                  | Device(s) |
|-------------------------|----------------------------------------------|-----------|
| U787B.001.DNW0974-P1-C1 | PCI-X capable, 64 bit, 133MHz slot ent0      |           |
| U787B.001.DNW0974-P1-C2 | PCI-X capable, 64 bit, 133MHz slot Empty     |           |
| U787B.001.DNW0974-P1-C3 | PCI-X capable, 64 bit, 133MHz slot Empty     |           |
| U787B.001.DNW0974-P1-C4 | PCI-X capable, 64 bit, 133MHz slot sisioa0   |           |
| U787B.001.DNW0974-P1-C5 | PCI-X capable, 64 bit, 133MHz slot pci5 lai0 |           |

В этом VIOS слоты C2 и C3 являются пустыми слотами PCI с горячей заменой. Слоты C1, C4 и C5 – также слоты адаптеров с горячей заменой, с адаптером Ethernet в C1, адаптером SCSI RAID в C4 и графическим адаптером в C5.

Для добавления адаптера выберите в меню Add a PCI Hot Plug Adapter, и появится список доступных слотов, как показано в примере 6-8.

#### **Пример 6-8. Экран Add a PCI Hot Plug Adapter**

|  |                                                                        |                                          |
|--|------------------------------------------------------------------------|------------------------------------------|
|  | Add a PCI Hot Plug Adapter                                             |                                          |
|  | Move cursor to desired item and press Enter. Use arrow keys to scroll. |                                          |
|  | # Slot                                                                 | Description Device(s)                    |
|  | U787B.001.DNW0974-P1-C2                                                | PCI-X capable, 64 bit, 133MHz slot Empty |
|  | U787B.001.DNW0974-P1-C3                                                | PCI-X capable, 64 bit, 133MHz slot Empty |

В этом примере выбран слот C2, и вывод показан в примере 6-9.

#### **Пример 6-9. Добавление адаптера PCI с горячей заменой в слот 2**

```
Command: running stdout: yes stderr: no
Before command completion, additional instructions may appear below.
The visual indicator for the specified PCI slot has
been set to the identify state. Press Enter to continue
or enter x to exit.
```

Дальнейшее добавление адаптера выполняется так же, как и в обычном LPAR AIX 5L, со следующими этапами:

- ▶ Мигает индикатор для определения месторасположения адаптера
- ▶ Установка адаптера
- ▶ Завершение задачи установки адаптера с использованием команды diagmenu

После успешного добавления адаптера команда diagmenu выдаст следующее:

Add Operation Complete.

После этого шага необходимо запустить команду `cfgdev` для настройки устройства VIOS. После этого, выбрав опцию просмотра доступных адаптеров с горячей заменой в `diagmenu`, в слоте C2 появится новый адаптер Fibre Channel:

| # Slot                  | Description                        | Device(s)  |
|-------------------------|------------------------------------|------------|
| U787B.001.DNW0974-P1-C1 | PCI-X capable, 64 bit, 133MHz slot | ent0       |
| U787B.001.DNW0974-P1-C2 | PCI-X capable, 64 bit, 133MHz slot | fcs0       |
| U787B.001.DNW0974-P1-C3 | PCI-X capable, 64 bit, 133MHz slot | Empty      |
| U787B.001.DNW0974-P1-C4 | PCI-X capable, 64 bit, 133MHz slot | sisioa0    |
| U787B.001.DNW0974-P1-C5 | PCI-X capable, 64 bit, 133MHz slot | pcis1 lai0 |

Этот адаптер Fibre Channel теперь готов к подключению к SAN и имеет (несколько) LUN, выданных VIOS для виртуализации.

### Добавление диска SCSI с горячей заменой

Для добавления диска SCSI с горячей заменой в RAID-корзину на VIOS вам опять нужно использовать команду `diagmenu`. В пункте task selection выберите Hot Plug Tasks, затем SCSI and SCSI RAID Hot Plug Manager. Результат показан в примере 6-10.

#### Пример 6-10. Меню SCSI and SCSI RAID Hot Plug

---

```
Make selection, use Enter to continue.

List Hot Swap Enclosure Devices
 This selection lists all SCSI hot swap slots and their contents.
Identify a Device Attached to a SCSI Hot Swap Enclosure Device
 This selection sets the Identify indication.
Attach a Device to an SCSI Hot Swap Enclosure Device
 This selection sets the Add indication and prepares
 the slot for insertion of a device.
Replace/Remove a Device Attached to an SCSI Hot Swap Enclosure Device
 This selection sets the Remove indication and prepares
 the device for removal.
Configure Added/Replaced Devices
 This selection runs the configuration manager on the
 parent adapter where devices have been added or replaced.
```

---

После выбора пункта Attach a Device to a SCSI Hot Swap Enclosure Device вы увидите список свободных слотов в RAID-корзине, которую нужно подключить SCSI-диск. Ниже показан список свободных слотов SCSI Hot Swap Enclosure:

```
slot 3 [empty slot]
```

В этом примере слот 3 – единственный доступный пустой слот. Когда вы выберите пустой слот, появится экран, предлагающий вам подключить устройство и нажать Enter для возвращения LED в состояние normal.

После этого вернитесь в предыдущее меню и выберите в нем пункт List Hot Swap Enclosure Devices; вывод показан в примере 6-11.

#### Пример 6-11. Просмотр Hot Swap Enclosure Devices

---

```
The following is a list of SCSI Hot Swap Enclosure Devices. Status information
about a slot can be viewed.
```

```
Make selection, use Enter to continue.
U787B.001.DNW0974-
```

```

ses0 P1-T14-L15-L0
slot 1 P1-T14-L8-L0 hdisk3
slot 2 P1-T14-L5-L0 hdisk2
slot 3
slot 4 P1-T14-L3-L0 hdisk0

```

---

Слот 3 до сих пор показан без устройства hdisk в нем, но в состоянии populated. Запустите команду `cfgdev` для инициализации диска из командной строки VIOS; после возврата в это меню вы увидите, что в слоте 3 появился hdisk1:

```

U787B.001.DNW0974-
ses0 P1-T14-L15-L0
slot 1 P1-T14-L8-L0 hdisk3
slot 2 P1-T14-L5-L0 hdisk2
slot 3 P1-T14-L4-L0 hdisk1
slot 4 P1-T14-L3-L0 hdisk0

```

### 6.4.3. Восстановление после сбоя диска на VIOS

Если на Virtual I/O Server происходит сбой диска, то вам нужно выполнить несколько шагов для достижения полного восстановления. Хотя замена сбояного диска является ключевым шагом к восстановлению, необходимо также выполнить очистку VIOS и клиентского раздела до выполнения этапа перестройки.

Для просмотра произошедших на VIOS ошибок используется команда `errlog`. В нашем случае выдается следующий отчет:

| IDENTIFIER | TIMESTAMP  | T | C | RESOURCE_NAME | DESCRIPTION               |
|------------|------------|---|---|---------------|---------------------------|
| 613E5F38   | 0114201370 | P | H | LVDD          | I/O ERROR DETECTED BY LVM |
| 8647C4E2   | 0114201370 | P | H | hdisk1        | DISK OPERATION ERROR      |
| 613E5F38   | 0114201370 | P | H | LVDD          | I/O ERROR DETECTED BY LVM |
| 8647C4E2   | 0114201370 | P | H | hdisk1        | DISK OPERATION ERROR      |
| 613E5F38   | 0114201370 | P | H | LVDD          | I/O ERROR DETECTED BY LVM |
| 8647C4E2   | 0114201370 | P | H | hdisk1        | DISK OPERATION ERROR      |

Как показано в отчете об ошибках, на VIOS отказал hdisk1. Диск не является частью RAID и не отзеркалирован, так что его сбой является фатальной ошибкой на VIOS, так как потеряны логические тома.

**Замечание.** Для предотвращения возможных сбоев из-за потери SCSI-диска рекомендуется использование SCSI RAID для клиентских групп томов.

Группа томов `clientvg` имеет логические тома для групп томов `rootvg` клиентов и логический том для группы томов `NIM` клиента.

Виртуальная цель SCSI и устройство все еще доступны в системе, так как они не подключены непосредственно к отказавшему диску:

```

vhost0 Available Virtual SCSI Server Adapter
vhost1 Available Virtual SCSI Server Adapter
vhost2 Available Virtual SCSI Server Adapter
vsa0 Available LPAR Virtual Serial Adapter
vcl1nimvg Available Virtual Target Device - Logical Volume
vcl1rootvg Available Virtual Target Device - Logical Volume
vcl2rootvg Available Virtual Target Device - Logical Volume

```

Однако логические тома, располагавшиеся на сбояном диске, пропали, и необходимо пересоздание логических томов и устройств виртуальных целей SCSI.

Запустите `diagmenu` и выберите `Task Selection Hot Plug Task SCSI and SCSI RAID Hot Plug Manager` для появления следующей функции:

`Replace/Remove a Device Attached to an SCSI Hot Swap Enclosure Device`

Эта функция начнет процедуру замены сбояного диска. Последовательность идентификации диска, его замены и завершения задачи выполняется аналогично процедуре в разделе AIX 5L.

После замены и включения диска в группу томов необходимо создать новые логические тома для клиентских групп томов. Старые устройства виртуальных целей SCSI должны быть удалены командой `rmdev -dev vscsi -target-device`, а затем, используя команду `mkvdev`, их необходимо создать снова, аналогично первоначальным настройкам:

```
mkvdev -vdev {new logical volume} -vadapter {vhost}
```

На клиентской системе следующий вывод показывает состояние группы томов `rootvg`:

```
lsvg -l rootvg
rootvg:
 LV NAME TYPE LPs PPs PVs LV STATE MOUNT POINT
 hd5 boot 1 2 2 closed/syncd N/A
 hd6 paging 32 64 2 open/syncd N/A
 hd8 jfs2log 1 2 2 open/syncd N/A
 hd4 jfs2 1 2 2 open/syncd /
 hd2 jfs2 37 74 2 open/stale /usr
 hd9var jfs2 1 2 2 open/stale /var
 hd3 jfs2 3 6 2 open/stale /tmp
 hd1 jfs2 1 2 2 open/stale /home
 hd10opt jfs2 3 6 2 open/stale /opt
```

В разделе AIX 5L требуется очистка группы томов. Удалите зеркальное отображение с пропавшего диска, после чего удалите диск из группы томов. Удалите определение устройства `hdisk` и запустите команду `cfgmgr` для конфигурирования диска в системе. Процедура очистки необходима, так как логический том, созданный на VIOS, не содержит данных группы томов и будет показан как новый диск `hdisk` после завершения команды `cfgmgr`.

После того как отображение будет выполнено на VIOS, виртуальный SCSI-диск будет доступен и готов к работе. После этого можно выполнить стандартную процедуру зеркалирования группы томов `rootvg`.

### **Восстановление после сбоя VIOS**

Если по каким-либо причинам произошла перезагрузка VIOS без подготовки к этому клиентов, то потребуется выполнение процедуры очистки группы томов. В отличие от отказавшего устройства SCSI на VIOS, клиентские группы томов и логические тома по-прежнему остаются неповрежденными.

После перезагрузки VIOS пропавший перед этим виртуальный SCSI-диск клиента появится снова. Группа томов, которая использовала виртуальный SCSI-диск с этого сервера VIOS, покажет логические тома в состоянии `open/stale`, а диски, размещенные на VIOS, будут помечены как `missing` (пропавшие).

Корректный способ восстановления после этой ситуации – выполнение команды `varyonvg` для всех затронутых групп томов, включая `rootvg`. Эта команда восстановит диск из состояния `missing` в состояние (PV State) `active` и начнет синхронизацию логических томов.

После этого не требуется выполнения никаких дополнительных действий на разделе клиента.

### Отказы путей с MPIO

Если произойдет сбой либо сервера VIOS, либо адаптера Fibre Channel и в клиентском разделе настроен MPIO, то, запустив команду `lspath`, вы заметите сбойный путь:

```
lspath
Failed hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

После перезагрузки VIOS или замены адаптера Fibre Channel сбойный путь автоматически изменит статус на `Enabled`. Если этого не произойдет, вам необходимо выполнить команду `chpath -s Enabled` для принудительной активизации пути. Без выполнения этой команды, в случае сбоя на втором сервере VIOS, произойдет сбой ввода-вывода на клиентском разделе, так как не будет работоспособных путей к дискам.

#### 6.4.4. Дополнительные соображения и рекомендации по обслуживанию

В следующем разделе обсуждаются общие вопросы обслуживания VIOS в нескольких базовых, часто встречающихся сценариях.

##### Увеличение группы томов клиентского раздела

Увеличение размера группы томов клиентского раздела может быть выполнено двумя способами: с прерыванием операций и без прерывания. Метод с прерыванием операций требует деактивизации группы томов, но позволяет рост самих дисков. Метод без прерывания операций требует включения в группу томов дополнительных виртуальных дисков SCSI.

**Ограничение.** Увеличение `rootvg` может быть выполнено только путем включения дополнительных виртуальных дисков SCSI. Ранее включенные в группу диски не могут быть увеличены, как в случае других групп томов.

Для групп томов, отличных от `rootvg`, для увеличения размера должны быть выполнены следующие шаги:

1. На клиентском разделе деактивизируйте группу томов, используя команду `varyoffvg clientvg`.
2. Удалите клиентские SCSI-адAPTERЫ, относящиеся к группе томов.
3. На стороне VIOS выполните следующее:
  - a. Удалите целевое устройство SCSI, используя команду `rmdev -dev vtscsi`.
  - b. Увеличьте логический том, используя команду `extendlv client_lv size physical_disk`.

- c. Пересоздайте целевое устройство SCSI, используя команду `mkvdev -vdev client_lv -vadapter vhost`.
4. На клиенте выполните команду `cfgmgr` для пересоздания клиентских SCSI-адаптеров.
5. Импортируйте группу томов и выполните команду `chvg -g volume_group` для опроса дисков и синхронизации их увеличенного размера с описанием группы томов.

Альтернативно метод увеличения группы томов без прерывания операций выполняется следующим способом:

1. На VIOS, `vhost0` отобразите в `client_lv`, с созданием `vscsi1`.
2. Командой `mklv` создайте дополнительный логический том.
3. Создайте новое устройство виртуального SCSI, отображающееся в существующий SCSI хост-адаптер `vhost0`:  
`mkvdev -vdev new_logical_volume -vadapter vhost0`
4. На клиенте обнаружьте новый диск, используя команду `cfgmgr`, и добавьте его в группу томов, требующую увеличения.

**Важно.** На время написания этой книги не рекомендуется располагать логический том, предназначенный для создания виртуальных SCSI-устройств, на нескольких физических дисках.

## Обновление VIOS

Команда `updateios` используется для обновления ПО VIOS. Работа этой команды показана в примере 6-12.

### Пример 6-12. Пример работы команды updateios

---

```
$ updateios
Command requires option "-accept -cleanup -dev -remove -reject".
Usage: updateios -dev Media [-f] [-install] [-accept]
 updateios -commit | -reject [-f]
 updateios -cleanup
 updateios -remove {-file RemoveListFile | RemoveList}
```

---

Команда `updateios` не выполняет `commit` устанавливаемых обновлений, а оставляет их в состоянии `applied`. Перед проведением следующего обновления необходимо либо подтвердить (`commit`), либо отказатьаться (`reject`) от предыдущего обновления, что выполняется флагами `-commit` и `-reject`.

В некоторых случаях для завершения обновления VIOS требуется перезагрузка раздела. Некоторые обновления могут потребовать также обновления микрокода системы, что может означать недоступность всей системы (для перезагрузки, необходимой для завершения обновления микрокаода).

Пример обновления сервера VIOS без влияния на доступность клиентских приложений можно найти в разделе 6.4.1 «Параллельные обновления ПО VIOS».

## **Сбор информации командой snap для службы поддержки IBM**

При связи со службой технической поддержки IBM перед открытием заявки вам необходимо подготовить следующее:

- ▶ Точная конфигурация вашей системы.
- ▶ Результат работы команды `snap` с VIOS.

Основная коллекция информации `snap` выполняется аналогично стандартной команде AIX 5L `snap`. Создается стандартный `snap`-файл `snap.rax.Z`, который может быть переслан на сайт IBM, указанный вам представителем службы технической поддержки IBM.

Когда с командой `snap` используется ключ `-general`, собирается только общая информация о системе. Без ключа `-general` производится полное сканирование системы и вся информация, включая настройки подсистемы хранения, безопасности, результаты установки, сетевая конфигурация и виртуальные настройки, будут собраны в файл `snap.rax.Z`.

## **Дополнительные соображения по конфигурации с двумя Virtual I/O Server**

Вместе с избыточностью конфигурации с двумя серверами Virtual I/O Server приходит и дополнительное системное обслуживание для поддержки работоспособности этой конфигурации. Как и в случае НАСМР, все изменения, производимые на двух VIOS, связанные с обеспечением резервирования клиентскому разделу, должны быть протестированы. Хотя и не требуется соблюдения жестких требований НАСМР к тестированию, должно быть проведено тестирование после крупных изменений или добавлений для подтверждения того, что требуемая избыточность реально достигнута.

Того же требует и организация резервного копирования двух VIOS. При резервном копировании на оптическое устройство, где VIOS поддерживает контроль над DVD, необходимо планирование, так как клиенты также будут использовать этот VIOS для доступа к оптическому устройству. Назначение одного VIOS клиентом другого для DVD допустимо, но в такой конфигурации поддерживается только DVD-RAM.

## **6.5. Мониторинг виртуализованной среды**

Оскар Уайльд однажды сказал: «Правда редко чиста и никогда не проста» («The truth is rarely pure and never simple»). Это утверждение может быть применено к контролю использования ресурсов системы в виртуализованной среде. В таких средах количество ресурсов, принадлежащих разделу, может изменяться «на лету», и это представляет новые задачи как разработчикам инструментов измерения производительности, так и тем, кто пытается интерпретировать результаты.

Этот раздел начинается с теории об инструментальном измерении использования виртуальных ресурсов, прежде чем перейти к практическому использованию инструментов и представлением некоторых из новых команд AIX 5L, связанных с производительностью.

## **Задавайте правильные вопросы**

Вот немногие вопросы, на которые нужно ответить при проектировании инструментов мониторинга производительности в виртуализованной среде. Во многих случаях на них есть несколько правильных ответов.

При использовании SMT как вы будете измерять использование ресурсов двух логических процессоров? Является ли логический процессор, использующий 50% физического процессора, занятым на 100%? Когда неограниченный (un-capped) виртуальный процессор занят на 100%? Что будет в отчете, если произойдут изменения конфигурации во время мониторинга?

Для того чтобы помочь ответить на эти и другие вопросы, в семействе процессоров POWER5 реализован новый регистр, специально связанный с производительностью, называемый Process Utilization Resource Register (PURR). Регистр PURR отслеживает использование ресурса реального процессора на уровне нити или на уровне раздела. Средства мониторинга производительности AIX 5L в версии AIX 5L V5.3 обновлены. Они были модифицированы так, чтобы отображать эту новую статистику.

Традиционные измерения производительности базировались на опросах, обычно с частотой 100 Гц (каждый опрос соответствует 10 мс «тику» (tick)). Каждый опрос попадает в одну из четырех категорий:

|               |                                                                                |
|---------------|--------------------------------------------------------------------------------|
| <b>user</b>   | Прерываемый код вне ядра AIX 5L                                                |
| <b>sys</b>    | Прерываемый код в ядре AIX 5L и работающая в данный момент нить – не waitproc. |
| <b>iowait</b> | Работающая в данный момент нить – waitproc и есть ожидание ввода-вывода.       |
| <b>idle</b>   | Работающая в данный момент нить – waitproc и нет ожидания ввода-вывода.        |

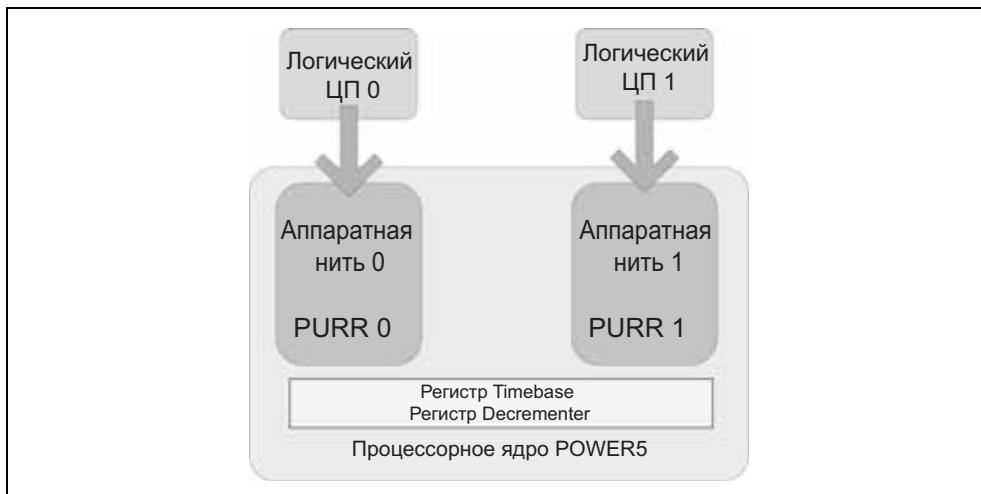
Этот традиционный механизм должен оставаться без изменений для поддержки двоичной совместимости с более ранними версиями.

Этот подход, базирующийся на опросах, не работает в виртуализированной среде, так как предположение, что цикл диспетчеризации всех виртуальных процессоров одинаков, не является истинным. Схожая проблема существует с SMT; если одна нить потребляет 100 процентов времени физического ЦП, то инструмент, базирующийся на опросах, покажет, что система занята на 50 процентов (1 процессор – на 100%, другой – на 0%), но фактически процессор действительно занят на 100 процентов.

### **6.5.1. Process Utilization Resource Register (PURR)**

PURR – простой 64-битный счетчик с теми же единицами для регистров timebase и decrementer, которые предоставляют статистику утилизации процессора на уровне нити. На рисунке 6-20 показаны логические ЦП и связь регистров PURR в пределах одного процессора (ядра – core) POWER5 и двух аппаратных нитей. При активизированном SMT каждая аппаратная нить видна как логический процессор.

Регистр *timebase*, показанный на рис. 6-20, – это просто аппаратный регистр, который увеличивается каждый тик. Регистр *decrementer* предоставляет периодические прерывания.



**Рис. 6-20.** PURR на уровне нитей

Каждый процессорный квант времени один из регистров PURR увеличивается:

- ▶ Инструкциями диспетчеризации нити
- ▶ Нитью, которая окончила диспетчеризировать инструкцию

Сумма двух PURR равна величине в регистре *timebase*. Этот подход является аппроксимацией, так как SMT позволяет выполнять обеим нитям параллельно. Он просто предоставляет обоснованную индикацию того, какая нить использует ресурсы процессора POWER5; однако он не предоставляет механизма для определения разницы в производительности с включенным и выключенным режимами SMT.

### Новые метрики, базирующиеся на PURR

Новые регистры предоставляют некоторую новую статистику.

#### Статистики SMT

Соотношение (дельта PURR)/(дельта *timebase*) за интервал индицируют долю физического процессора, потребленную логическим процессором. Это величина, возвращаемая командами `sar -P ALL` и `mpstat`.

Величина (дельта PURR/дельта TB)\*100 за интервал дает предыдущую величину в процентах и может быть интерпретирована как процент циклов диспетчеризации, данных логическому процессору, или процент физического процессора, потребленный логическим процессором. Это величина, возвращаемая командой `mpstat -s`, которая показывает статистику SMT.

## **Статистика ЦП в разделах с общими процессорами**

В среде с общими процессорами PURR измеряет время, которое виртуальный процессор работает на физическом процессоре. Время раздела, как и процессор, виртуально. При этом гипервизор POWER поддерживает виртуальный Time Base как сумму двух PURR. В общих процессорах с выключенным SMT виртуальный time base – это просто величина, хранящаяся в PURR.

### **Общие процессоры с ограничением**

Для общих процессоров с ограничением подсчет идет следующим образом:

*Назначенный (entitled) PURR* за интервал высчитывается как  $entitlement * time\ base$ .

%user time за интервал высчитывается как:

$\%user = (\delta PURR \text{ in user mode} / entitled PURR) * 100$

### **Общие процессоры без ограничений**

Для общих процессоров без ограничений при подсчетах принимается во внимание переменная мощность. *Entitled PURR* в вышеуказанной формуле заменяется на потребленный (*consumed*) PURR, если потребление больше, чем назначение.

### **Потребление общим процессором физического процессора**

Потребление разделом физического процессора, измеренное за интервал, – это просто сумма потреблений всех его логических процессоров:

$SUM(\delta PURR / \delta TB)$

### **Потребление выделенной мощности раздела**

Потребление выделенной мощности раздела – это просто соотношение потребления физического процессора – physical processor consumption (PPC) к его назначенной мощности (entitlement), показанное в процентах:

$(PPC/ENT) * 100$

### **Емкость резерва общего процессорного пула**

Неиспользованные такты общего процессорного пула уходят в цикл простоя (idle loop) гипервизора POWER. Гипервизор POWER входит в этот цикл, когда все назначенные мощности выданы разделам и нет разделов для диспетчеризации. Время, потраченное в цикле простоя гипервизора, измеряемое в тиках, называется Pool Idle Count (PIC) емкость резерва общего процессорного пула за интервал вычисляется как:

$(\delta PIC / \delta TB)$

и измеряется в количестве процессоров. Только разделы, имеющие атрибут shared-processor pool authority, способны показывать эту величину. Пример с использованием команды lparstat дан в примере 6-21.

### **Утилизация логического процессора**

Это просто сумма традиционных 10 мс, базирующихся на тиках, опросов времени, проведенного в %sys и %user. Если эта величина начинает приближаться к 100%, то это может показывать, что раздел получит пользу от дополнительных виртуальных процессоров.

## 6.5.2. Общесистемные инструменты, модифицированные для виртуализации

Инструменты AIX 5L, представляющие информацию на уровне системы, такие, как команды `iostat`, `vmstat`, `sar` и `time`, при включенном SMT используют базирующуюся на PURR статистику для предоставления величин `%user`, `%system`, `%iowait` и `%idle`.

При выполнении в разделе общего процессорного пула эти команды добавляют два дополнительных столбца со следующей информацией:

- ▶ Потребление физического процессора разделом, показанное как `ps` или `%physc`.
- ▶ Процент выделенной мощности, потребленной разделом, показанный как `ec` или `%entc`.

Это показано в примерах 6-13 и 6-14.

### Пример 6-13. Команда iostat в ограниченном разделе общего пула

---

```
iostat -t 2 4
System configuration: lcpu=2 ent=0.50
tty: tin toutavg-cpu: % user % sys % idle % iowait physc % entc
 0.0 19.3 8.4 77.6 14.0 0.1 0.5 99.9
 0.0 83.2 9.9 75.8 14.2 0.1 0.5 99.5
 0.0 41.1 9.5 76.4 13.9 0.1 0.5 99.6
 0.0 41.0 9.4 76.4 14.1 0.0 0.5 99.7
```

---

### Пример 6-14. Команда sar в ограниченном разделе общего пула

---

```
sar -P ALL 2 2
AIX vio_client2 3 5 00CC489E4C00 08/17/05
System configuration: lcpu=2 ent=0.50
20:13:48 cpu %usr %sys %wio %idle physc %entc
20:13:50 0 19 71 0 9 0.31 61.1
 1 2 75 0 23 0.19 38.7
 - 13 73 0 15 0.50 99.8
20:13:52 0 21 69 0 9 0.31 61.1
 1 2 75 0 23 0.20 39.0
 - 14 71 0 15 0.50 100.2
Average 0 20 70 0 9 0.31 61.1
 1 2 75 0 23 0.19 38.9
 - 13 72 0 15 0.50 100.0
```

---

### Инструменты мониторинга логических процессоров

Инструменты для мониторинга логических процессоров – команды `mpstat` и `sar -P ALL`. При выполнении в разделе с включенным SMT они добавляют столбец `Physical Processor Fraction Consumed` (дельта PURR/дельта TB), показанный как `physc`. Он показывает относительное разделение времени физического процессора между логическими процессорами.

При выполнении в разделе общего пула эти команды добавляют новый столбец Percentage of Entitlement Consumed ( $(PPFC/ENT) * 100$ ), показанный как %entc. Эта величина показывает относительное потребление каждого логического процессора, выраженное в процентах.

Команда `mpstat -s` в примере 6-15 показывает виртуальные и логические процессоры и их загрузку.

#### Пример 6-15. Команда mpstat в режиме SMT

---

```
mpstat -s 2 2
System configuration: lcpu=8 ent=0.5
 Proc0 Proc2 Proc4 Proc6
 49.94% 0.03% 0.03% 0.03%
cpu0 cpu1 cpu2 cpu3 cpu4 cpu5 cpu6 cpu7
24.98% 24.96% 0.01% 0.01% 0.01% 0.01% 0.01% 0.01%

 Proc0 Proc2 Proc4 Proc6
 49.90% 0.03% 0.03% 0.03%
cpu0 cpu1 cpu2 cpu3 cpu4 cpu5 cpu6 cpu7
25.01% 24.89% 0.02% 0.01% 0.01% 0.01% 0.01% 0.01%
```

---

### 6.5.3. Команда topas

Экран статистики ЦП команды `topas` теперь включает информацию о потреблении разделом физического процессора (Physc) и мощности (%Entc), как показано в примере 6-16.

#### Пример 6-16. Экран по умолчанию команды topas

---

| Topas Monitor for host: |             |        | vio_client2   |             | EVENTS/QUEUES |               | FILE/TTY      |       |
|-------------------------|-------------|--------|---------------|-------------|---------------|---------------|---------------|-------|
|                         |             |        | Interval: 5   |             | Cswitch 491   |               | Readch 149.2K |       |
| Kernel                  | 71.3        | #####  |               |             | Syscall       | 808           | Writech       | 19425 |
| User                    | 12.8        | ####   |               |             | Reads         | 155           | Rawin         | 0     |
| Wait                    | 0.2         | #      |               |             | Writes        | 26            | Ttyout        | 105   |
| Idle                    | 15.7        | ####   |               |             | Forks         | 25            | Igets         | 0     |
| <b>Physc =</b>          | <b>0.49</b> |        | <b>%Entc=</b> | <b>98.9</b> | Execs         | 25            | Namei         | 185   |
|                         |             |        |               |             | Runqueue      | 0.8           | Dirblk        | 0     |
|                         |             |        |               |             | Waitqueue     | 0.2           |               |       |
| Network                 | KBPS        | I-Pack | O-Pack        | KB-In       | KB-Out        |               |               |       |
| en0                     | 0.2         | 2.8    |               | 0.1         | 0.1           | PAGING        | MEMORY        |       |
| lo0                     | 0.0         | 0.0    |               | 0.0         | 0.0           | Faults        | Real, MB      | 512   |
|                         |             |        |               |             |               | Steals        | % Comp        | 65.5  |
| Disk                    | Busy%       | KBPS   | TPS           | KB-Read     | KB-Writ       | PgspIn        | 0 % Noncomp   | 34.6  |
| hdisk0                  | 39.8        | 376.8  | 93.4          | 106.4       | 270.4         | PgspOut       | 0 % Client    | 37.5  |
| PageIn                  | 51          |        |               |             |               |               |               |       |
| Name                    |             | PID    | CPU%          | PgSp        | Owner         | PageOut       | 0 PAGING      | SPACE |
| ksh                     |             | 245872 | 15.9          | 84.6        | root          | Sios          | 51 Size, MB   | 512   |
| syncd                   |             | 98426  | 0.2           | 0.5         | root          |               | % Used        | 7.1   |
| topas                   |             | 311370 | 0.1           | 1.0         | root          | NFS (calls/s) | % Free        | 92.8  |
| getty                   |             | 241866 | 0.0           | 0.4         | root          | ServerV2      | 0             |       |

---

```

lrud 16392 0.0 0.1 root ClientV2 0 Press:
gil 57372 0.0 0.1 root ServerV3 0 "h" for help
rpc.lock 172192 0.0 0.2 root ClientV3 0 "q" to quit

```

Команда **topas** имеет новый режим разделения экрана (ключ **-L** или команда **L**). В верхней части экрана показана часть статистики команды **lparstat**, а в нижней части – отсортированный список логических процессоров с некоторым количеством значений, возвращаемых командой **mpstat**, как показано в примере 6-17.

#### **Пример 6-17. Экран монитора LPAR командой topas**

```

Interval: 5 Logical Partition: VIO_client2 Wed Aug 10 17:13:20 2005
Psize: 6 Shared SMT ON Online Memory: 512.0
Ent: 0.50 Mode: Capped Online Logical CPUs: 2
Partition CPU Utilization Online Virtual CPUs: 1
%usr %sys %wait %idle physc %entc %lbusy app vcs wphint %hypv hcalls
 0 0 0 100 0.0 0.70 0.00 5.98 336 0 0.0 3
=====
LCPU minpf majpf intr csw icsw runq lpa scalls usr sys _wt idl pc lcsw
Cpu0 0 0 28 134 66 0 100 10 4 51 0 45 0.00 193
Cpu1 0 0 169 0 0 0 0 0 30 0 70 0.00 143

```

У команды **topas** есть также новый ключ **-D** (или команда **D**), который показывает дисковую статистику, принимая во внимание виртуальный SCSI, как показано в примере 6-18.

#### **Пример 6-18. Экран мониторинга диска командой topas**

```

Topas Monitor for host: vio_client2 Interval: 10 Wed Aug 10 18:28:44 2005
=====
Disk Busy% KBPS TPS KB-R ART MRT KB-W AWT MWT AQW AQD
hdisk0 47.1 3.2K 138.4 1.9K 4.7 21.1 1.3K 7.2 17.6 9.5 7.6

```

Командой **topas -C** можно увидеть использование системных ресурсов другими разделами. Эта команда показывает только разделы с AIX 5L V5.3 ML3 или более поздних версий; она не покажет разделы с сервером VIOS или Linux. Пример вывода этой команды показан в примере 6-19.

#### **Пример 6-19. Экран мониторинга других разделов команды topas**

```

Topas CEC Monitor Interval: 10 Mon Aug 15 16:20:34 2005
Partitions Memory (GB) Processors
Shr: 2 Mon: 2.0 InUse: 1.4 Shr: 1.0 PSz: 0 Shr_PhysB: 0.00
Ded: 2 Avl: - Ded: 2 APP: 0.0 Ded_PhysB: 0.00
Host OS M Mem InU Lp Us Sy Wa Id PhysB Ent %EntC Vcs WPhI
-----shared-----
plmserver A53 S 0.5 0.4 2 5 2 0 93 0.00 0.50 0.7 261 0
vio_client2 A53 S 0.5 0.3 2 0 0 0 99 0.00 0.50 0.7 347 0
-----dedicated-----
db_server A53 C 0.5 0.4 2 0 0 0 99 0.00
app_server A53 C 0.5 0.5 2 0 0 0 100 0.00

```

Экран команды `topas -C` разделяется на регионы заголовка и разделов. Регион разделов делится между выделенными разделами и разделами общего пула. Регион заголовка содержит общую информацию СЕС и разбит на три столбца. Столбец Partitions показывает, сколько разделов каждого типа `topas` нашел в СЕС. Столбец Memory показывает общее количество памяти и количество, выданное разделам (в гигабайтах). Столбец Processor показывает типы процессоров – выделенные и общие. Параметр PSz показывает размер общего процессорного пула; он доступен только разделам с shared-processor pool authority. Параметры Shr\_PhysB и Ded\_PhysB сравнимы со столбцом physc вывода команды `lparstat`, но он исключает время простоя (idle).

Регион разделов содержит список всех разделов, которые команда `topas` нашла в СЕС. Столбец OS показывает тип операционной системы. В примере 6-19 A53 индицирует AIX 5L V5.3. Столбец M показывает режимы раздела – общий (shared), ограниченный (capped) и неограниченный (uncapped). Столбцы Mem и InU показывают сконфигурированную и занятую память, измеряемую в гигабайтах. Столбец Lp показывает, сколько сконфигурировано логических процессоров. В примере 6-19 все разделы имеют один виртуальный процессор с включенным режимом SMT и, следовательно, имеют по два логических процессора. Столбцы Us, Sy, Wa и Id показывают проценты времени работы процессора в режимах user, system, wait и idle. Столбец PhysB – тот же, что и в регионе заголовка. Столбцы Ent, %Ent, Vcsw и PhI указываются только для разделов общего пула. Столбец Ent показывает выделенную мощность, столбец %EntC показывает процент использования выделенной мощности, столбец Vcsw показывает количество переключений виртуального контекста в секунду (virtual context switch rate), и столбец PhI показывает количество фантомных прерываний в секунду (phantom interrupt rate).

Для расширения глобальной информации введите в окне команды `topas` букву g, как показано в примере 6-20. В этом примере некоторые поля не заполнены, например общее количество доступной памяти. Они зарезервированы для обновлений этой команды, которые позволяют `topas` взаимодействовать с НМС для получения этих значений. Возможно вручную указать некоторые из этих параметров, указав их в командной строке.

#### **Пример 6-20. Глобальная информация topas -C с командой g**

| Topas CEC Monitor         |      | Interval: 10 |           | Mon Aug 15 16:25:37 2005 |     |              |              |      |    |       |      |       |      |     |
|---------------------------|------|--------------|-----------|--------------------------|-----|--------------|--------------|------|----|-------|------|-------|------|-----|
| Partition                 | Info | Memory (GB)  | Processor |                          |     |              |              |      |    |       |      |       |      |     |
| Monitored :               | 4    | Monitored :  | 2.0       | Monitored :              | 3   | Shr Physical | Busy:        | 0.01 |    |       |      |       |      |     |
| UnMonitored:              | -    | Unmonitored: | -         | Unmonitored:             | -   | Ded Physical | Busy:        | 0.00 |    |       |      |       |      |     |
| Shared :                  | 2    | Available :  | -         | Available :              | -   |              |              |      |    |       |      |       |      |     |
| Dedicated :               | 2    | UnAllocated: | -         | UnAllocated:             | -   | Hypervisor   |              |      |    |       |      |       |      |     |
| Capped :                  | 4    | Consumed :   | 1.5       | Shared :                 | 1   | Virt Context | Switch:      | 559  |    |       |      |       |      |     |
| Uncapped :                | 0    |              |           | Dedicated :              | 2   | Phantom      | Interrupts : | 0    |    |       |      |       |      |     |
|                           |      |              |           | Pool Size :              | 0   |              |              |      |    |       |      |       |      |     |
|                           |      |              |           | Avail Pool :             | 0.0 |              |              |      |    |       |      |       |      |     |
| Host                      | OS   | M            | Mem       | InU                      | Lp  | Us           | Sy           | Wa   | Id | PhysB | Ent  | %EntC | Vcsw | PhI |
| <hr/> -----shared-----    |      |              |           |                          |     |              |              |      |    |       |      |       |      |     |
| vio_client2               | A53  | S            | 0.5       | 0.5                      | 2   | 0            | 0            | 0    | 99 | 0.00  | 0.50 | 0.9   | 303  | 0   |
| plmserver                 | A53  | S            | 0.5       | 0.4                      | 2   | 0            | 0            | 0    | 99 | 0.00  | 0.50 | 0.7   | 256  | 0   |
| <hr/> -----dedicated----- |      |              |           |                          |     |              |              |      |    |       |      |       |      |     |

```
app_server A53 C 0.5 0.3 2 0 0 99 0.00
db_server A53 C 0.5 0.3 2 0 0 99 0.00
```

---

Команда `topas` доступна на сервере Virtual I/O Server.

#### 6.5.4. Новые команды мониторинга

В AIX 5L V5.3 появились некоторые новые команды мониторинга, специфичные в виртуализованной среде. Они обсуждаются в этом разделе.

##### Команда `lparstat`

Команда `lparstat` показывает информацию о конфигурации и производительности раздела, в котором она выполняется. Эта команда работает на всех системах под управлением AIX 5L V5.3, даже на тех, что не поддерживают LPAR (конечно, без информации о них). Она поддерживается как в выделенных разделах, так и в разделах общего процессорного пула как в режиме SMT, так и нет.

У команды `lparstat` есть четыре режима работы:

|                                                      |                                              |
|------------------------------------------------------|----------------------------------------------|
| <b>Monitoring (Мониторинг)</b>                       | Режим по умолчанию, без дополнительных опций |
| <b>Hypervisor summary</b>                            | С опцией <code>-h</code>                     |
| <b>(Суммарная информация о гипервизоре)</b>          |                                              |
| <b>Hypervisor hcalls (Вызовы гипервизора hcalls)</b> | С опцией <code>-H</code>                     |
| <b>System configuration</b>                          | С опцией <code>-i</code>                     |
| <b>(Конфигурация системы)</b>                        |                                              |

Во всех режимах, кроме режима с опцией `-i`, команда `lparstat` печатает в одной строке сводку о конфигурации системы перед отображением соответствующей информации.

##### Режим мониторинга команды `lparstat`

Без опций команда `lparstat` производит мониторинг использования ресурсов системы. Как и большинство `stat`-команд, команда `lparstat` принимает опциональные параметры – интервал и количество опросов. Вывод команды `lparstat` показан в примере 6-21.

##### Пример 6-21. Режим мониторинга команды `lparstat`

---

```
lparstat 2 4
System configuration: type=Shared mode=Capped smt=On lcput=2 mem=512 psiz=6
ent=0.50
%user %sys %wait %idle physc %entc lbusy app vcsw phint
----- ----- ----- ----- -----
14.2 85.3 0.6 0.0 0.50 99.6 97.7 5.49 311 1
13.7 85.9 0.3 0.0 0.50 99.8 98.5 5.49 321 0
13.9 85.9 0.2 0.0 0.50 100.2 98.2 5.49 319 0
14.7 85.0 0.3 0.0 0.50 100.0 98.7 5.49 353 0
```

---

Из первой строки, в которой показана конфигурация системы, видно, что раздел работает в общем пуле, в ограниченном режиме (type=Shared и mode=Capped), SMT активизирован (smt=On) и имеет два логических ЦП (lcpu=2), из чего можно сделать вывод о том, что сконфигурирован один виртуальный процессор, раздел имеет 512МБ памяти (mem=512) и назначенная процессорная мощность раздела равна 0.5 физического процессора (ent=0.50).

Вывод psiz=6 обозначает, что в общем пуле есть шесть процессоров. Эта информация доступна только тем разделам, у которых в конфигурации HMC отмечен флаг shared-processor pool authority.

Поля %user, %sys, %wait и %idle показывают традиционную статистику ЦП UNIX. Столбец physc показывает, сколько физических процессоров потребляет раздел. Параметр %entc показывает процент мощности, потребленный разделом, а столбец lbusy показывает процент занятости ЦП на уровне пользователя и системы (user и system). Из этих трех столбцов в примере 6-21 вместе с информацией в строке сводки конфигурации мы достаточно просто можем сделать вывод о том, что раздел испытывает нехватку ЦП (CPU bound), потребляя всю свою мощность.

Столбец vcsw показывает количество переключений виртуального контекста. Оно связано с количеством аппаратных выгрузок виртуального процессора. Эта величина показывает загрузку гипервизора POWER.

Столбец phint показывает количество фантомных прерываний, которое получает раздел. Фантомное прерывание – это прерывание, предназначено другому разделу, использующему тот же физический процессор. Например, один раздел начинает операцию ввода-вывода. Пока раздел ожидает окончания операции ввода-вывода, он уступает физический процессор другому разделу. Операция ввода-вывода завершается, и контроллер посыпает прерывание запрашивающему процессору, но так как прерванный раздел, выполняющийся на процессоре, не является точкой назначения прерывания, раздел сообщает «это не для меня» и прерывание отправляется в очередь гипервизором POWER. Фантомные прерывания являются относительно незатратными (в терминах вычислительного времени) и имеют небольшое влияние на производительность системы, кроме тех случаев, когда они происходят в очень большом количестве.

### Суммарная информация о гипервизоре команды lparstat

С опцией -h команда lparstat отображает сводку активности гипервизора POWER, как показано в примере 6-22.

#### Пример 6-22. Сводка активности гипервизора – команда lparstat

---

| # lparstat -h 2 4                                                                   |
|-------------------------------------------------------------------------------------|
| System configuration: type=Shared mode=Capped smt=On lcpu=2 mem=512 psiz=6 ent=0.50 |
| %user %sys %wait %idle physc %entc lbusy app vcsw phint %hypv hcalls                |
| ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----       |
| 14.0 72.0 0.1 13.9 0.50 100.1 50.2 5.48 504 1 15.1 8830                             |
| 14.1 71.7 0.1 14.1 0.50 99.9 48.8 5.50 511 2 15.2 8696                              |
| 14.8 71.2 0.1 13.9 0.50 100.0 49.8 5.49 507 0 15.5 8826                             |
| 13.9 71.5 0.1 14.4 0.50 99.4 47.2 5.49 559 0 17.5 9381                              |

---

Опция -h добавляет два столбца к выводу режима мониторинга (без опций) команды lparstat.

Столбец %hypv показывает количество времени, занятого в гипервизоре POWER за каждый интервал. В примере 6-22 показано, что за каждый интервал в гипервизоре было проведено от 15 до 17 процентов времени. Если эта величина становится слишком большой, можно использовать опцию -H для просмотра того, какой вызов гипервизора вызывает проблемы, как показано в следующем разделе.

Размещаемая операционная система использует вызовы гипервизора (hcalls) для коммуникации с Гипервизором POWER (см. раздел 3.6 «Введение в гипервизор POWER»). Столбец hcalls показывает общее количество вызовов AIX 5L, сделанных к гипервизору POWER. В примере 6-22 показано примерно 4500 вызовов в секунду. Эту величину можно использовать совместно со столбцом %hypv и командой lparstat -H, показанной в следующем разделе, для отслеживания причин высокой загрузки гипервизора.

### Команда lparstat – вызовы гипервизора POWER

Опция -H команды lparstat предоставляет детальный анализ времени, проведенного в гипервизоре POWER. Для каждого вызова (hcall) она показывает:

- ▶ Сколько раз он был сделан.
- ▶ Процент общего времени, потраченный на вызов
- ▶ Процент времени гипервизора POWER, потраченный на вызов
- ▶ Среднее и максимальное время, потраченное на вызов

В примере 6-23 вызов h\_cede полностью доминирует по времени, проведенному на гипервизоре POWER (94.8%); это достаточно распространенный случай. Хотя числа и большие, например максимальное время вызова для h\_cede – 15463728, но оно дано в наносекундах, так что это всего 15.5 мс и среднее время вызова h\_cede менее 0.2 мс.

#### Пример 6-23. Вызовы гипервизора – команда lparstat

```
lparstat -H 2 1
System configuration: type=Shared mode=Capped smt=On lcpu=2 mem=512 psize=6 ent=0.50
Detailed information on Hypervisor Calls
Hypervisor Number of %Total Time %Hypervisor Avg Call Max Call
Call Calls Spent Time Spent Time(ns) Time(ns)
remove 4052 0.2 1.1 430 5877
read 2785 0.1 0.4 194 5322
nclear_mod 0 0.0 0.0 1 0
page_init 2418 0.2 1.4 906 6863
clear_ref 306 0.0 0.0 114 1159
protect 0 0.0 0.0 1 0
put_tce 142 0.0 0.1 1140 2071
xirr 67 0.0 0.0 874 3313
eoi 66 0.0 0.0 729 1067
ipi 0 0.0 0.0 1 405
cppr 66 0.0 0.0 390 685
asr 0 0.0 0.0 1 0
others 0 0.0 0.0 1 0
enter 6404 0.2 1.2 290 5641
```

|                    |     |      |      |        |          |
|--------------------|-----|------|------|--------|----------|
| cede               | 834 | 14.5 | 94.8 | 173511 | 15463728 |
| migrate_dma        | 0   | 0.0  | 0.0  | 1      | 0        |
| put_rtce           | 0   | 0.0  | 0.0  | 1      | 0        |
| confer             | 0   | 0.0  | 0.0  | 1      | 2434     |
| prod               | 152 | 0.0  | 0.0  | 463    | 777      |
| get_ppp            | 1   | 0.0  | 0.0  | 1980   | 2583     |
| set_ppp            | 0   | 0.0  | 0.0  | 1      | 0        |
| purr               | 0   | 0.0  | 0.0  | 1      | 0        |
| pic                | 1   | 0.0  | 0.0  | 2912   | 3849     |
| bulk_remove        | 809 | 0.1  | 0.7  | 1381   | 7114     |
| send_crq           | 61  | 0.0  | 0.1  | 2415   | 6143     |
| copy_rdma          | 0   | 0.0  | 0.0  | 1      | 0        |
| get_tce            | 0   | 0.0  | 0.0  | 1      | 0        |
| send_logical_lan   | 2   | 0.0  | 0.0  | 2600   | 6384     |
| add_logicl_lan_buf | 6   | 0.0  | 0.0  | 521    | 1733     |

---

### Команда lparstat – конфигурация системы

Опция -i команды lparstat дает вывод в формате, существенно отличающемся от других команд lparstat. Он показывает список настроек раздела, определенных на НМС. В примере 6-24 показано использование этой опции.

#### Пример 6-24. Конфигурация системы – команда lparstat

---

```
lparstat -i
Node Name : vio_client2
Partition Name : VIO_client2
Partition Number : 1
Type : Shared-SMT
Mode : Capped
Entitled Capacity : 0.50
Partition Group-ID: 32769
Shared Pool ID : 0
Online Virtual CPUs: 1
Maximum Virtual CPUs: 32
Minimum Virtual CPUs: 1
Online Memory : 512 MB
Maximum Memory : 1024 MB
Minimum Memory : 128 MB
Variable Capacity Weight: 0 ←----- Ограниченный раздел (capped)
Minimum Capacity : 0.10
Maximum Capacity : 2.00
Capacity Increment : 0.01
Maximum Physical CPUs in system: 16
Active Physical CPUs in system: 8
Active CPUs in Pool : 6
Unallocated Capacity: 0.00
Physical CPU Percentage: 50.00%
Unallocated Weight : 0
```

---

**Замечание.** Значение переменной variable capacity weight (вес раздела) в этом примере равно нулю, так как это ограниченный раздел, как можно видеть из пятой строки вывода.

**Замечание.** Формат вывода команды lparstat может быть разным, в зависимости от настроек раздела (SMT вкл/выкл, общие или выделенные процессоры). Это делает его парсинг командами sed, awk, cut, perl и т.п. весьма проблематичным.

### 6.5.5. Команда mpstat

Команда mpstat собирает и отображает статистику производительности всех логических ЦП в разделе. Интерпретация многих из величин, отображаемых этой командой, требует понимания работы гипервизора POWER и процессора POWER5. Опции команды mpstat показаны в таблице 6-1.

**Таблица 6-1.** Опции команды mpstat

| Команда | Опции | Функции                                                                                                                                                          |
|---------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mpstat  | нет   | Показывает статистику по умолчанию; подмножество опции -a.                                                                                                       |
|         | -a    | Показывает все 29 параметров статистики логического процессора.                                                                                                  |
|         | -i    | Показывает статистику прерываний; подмножество опции -a.                                                                                                         |
|         | -d    | Показывает статистику сходства (affinity) и миграции процессора для нитей AIX 5L и статистику диспетчеризации для логических процессоров. Подмножество опции -i. |
|         | -s    | Показывает статистику использования SMT, если режим SMT активирован. Эта информация не показывается с опцией -a.                                                 |

После запуска команда mpstat отображает две секции статистики. В первой секции показана конфигурация системы, которая отображается при старте команды и при изменении конфигурации системы. Во второй секции отображается статистика утилизации, которая будет показываться в интервалы времени и в любое время, когда значения этих метрик отличаются от значений в прошлом интервале времени.

Необязательная опция -w переключает вывод в режим широкого экрана.

В примере 6-25 показан вывод команды mpstat -a. Так как вывод очень широкий, результат разделен на три набора столбцов, со столбцом ЦП (CPU), повторяющимся в каждом выводе. Значение каждого столбца дано в таблице 6-2.

#### Пример 6-25. Команда mpstat

---

```
mpstat -a
System configuration: lcpu=2 ent=0.5
cpu min maj mpcs mpcr dev soft dec ph cs ics bound
 0 134 3 0 0 2 0 105 0 90 47 0
 1 88 0 0 0 2 54 122 0 9 6 0
U - - - - - - - - - - - -
```

|     |      |      |        |       |      |      |      |      |      |      |     |
|-----|------|------|--------|-------|------|------|------|------|------|------|-----|
| ALL | 222  | 3    | 0      | 0     | 4    | 54   | 227  | 0    | 99   | 53   | 0   |
| cpu | rq   | push | S3pull | S3grd | S0rd | S1rd | S2rd | S3rd | S4rd | S5rd |     |
| 0   | 0    | 0    | 0      | 0     | 98.8 | 1.2  | 0.0  | 0.0  | 0.0  | 0.0  | 0.0 |
| 1   | 0    | 0    | 0      | 0     | 90.8 | 9.2  | 0.0  | 0.0  | 0.0  | 0.0  | 0.0 |
| U   | -    | -    | -      | -     | -    | -    | -    | -    | -    | -    | -   |
| ALL | 0    | 0    | 0      | 0     | 97.7 | 2.3  | 0.0  | 0.0  | 0.0  | 0.0  | 0.0 |
| cpu | sysc | us   | sy     | wa    | id   | pc   | %ec  | ilcs | vlcs |      |     |
| 0   | 205  | 5.1  | 84.2   | 0.3   | 10.4 | 0.01 | 2.1  | 11   | 173  |      |     |
| 1   | 47   | 6.3  | 72.7   | 0.2   | 20.8 | 0.01 | 1.4  | 4    | 157  |      |     |
| U   | -    | -    | -      | 0.3   | 96.1 | 0.48 | 96.4 | -    | -    |      |     |
| ALL | 252  | 0.2  | 2.8    | 0.4   | 96.6 | 0.02 | 3.6  | 5    | 330  |      |     |

**Таблица 6-2.** Интерпретация вывода команды mpstat

| Столбец      | Измеряемый параметр                                                                                 | Комментарии                                                                                                   |
|--------------|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| cpu          | Идентификатор логического ЦП                                                                        | У показывает неиспользованные ЦП                                                                              |
| min/maj      | Старшие и младшие (minor и major) сбои страниц (page faults)                                        | Младший сбой страницы не вызывает дискового ввода-вывода, старший – вызывает                                  |
| mpcr<br>mpcs | Количество посланных (mpcs) и принятых (mpcr) прерываний mpс                                        | Прерывания mpс используются для межпроцессорных коммуникаций                                                  |
| dev          | Количество прерываний, инициированных устройствами                                                  | Аппаратное прерывание                                                                                         |
| soft         | Количество прерываний, инициированных ПО                                                            |                                                                                                               |
| dec          | Количество прерываний decrementer                                                                   | Прерывания таймера                                                                                            |
| ph           | Количество фантомных прерываний                                                                     | Количество полученных прерываний, которые предназначены другому разделу, использующему тот же самый процессор |
| cs           | Переключения контекста                                                                              |                                                                                                               |
| ics          | Непреднамеренные переключения контекста                                                             |                                                                                                               |
| bound        | Количество нитей, привязанных к процессору                                                          | Через наборы ресурсов (resource sets) или вызов bindprocessor                                                 |
| rq           | Длина очереди на выполнение                                                                         | Количество нитей, ожидающих выполнения                                                                        |
| push         | Количество нитей, мигрированных на другой процессор при балансировке нагрузки                       |                                                                                                               |
| mig          | Общее количество миграций нитей (на другой логический процессор)                                    | Показывается только в версии команды по умолчанию (без опций)                                                 |
| s3pull       | Количество миграций логического процессора на другой физический процессор в другом МСМ <sup>a</sup> | Измеряет миграции нитей через границы МСМ для ликвидации простоев (простаивающий МСМ)                         |

*Продолжение табл.*

| Столбец        | Измеряемый параметр                                                                                               | Комментарии                                                                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| s3grd          | Количество диспетчеризаций из общей очереди на выполнение                                                         |                                                                                                                                                                                     |
| s0rd           | Процент редиспетчеризаций нити, происходящих на том же логическом процессоре                                      | Это оптимальный случай; нить использует те же самые регистры. Это значение должно быть большим                                                                                      |
| s1rd           | Процент редиспетчеризаций нити, происходящих на том же физическом процессоре, но на другом логическом процессоре  | Эта ситуация почти так же хороша, как и предыдущая; нить использует те же L1-кэши инструкций и данных. Если s0rd имеет небольшое значение, то эта метрика должна быть высокой       |
| s2rd           | Процент редиспетчеризаций нити, происходящих на том же самом чипе, но не на том же ядре (процессор POWER5)        | L1-кэши разные, но в процессоре POWER5 L2-кэши совместно используются ядрами <sup>b</sup> , так что L2- и L3-кэши могут по-прежнему оставаться горячими при редиспетчеризации нити  |
| s3rd           | Процент редиспетчеризаций нити, происходящих на том же самом MCM <sup>a</sup> , но не на том же самом чипе        | Нить остается близко к используемой ею физической памяти                                                                                                                            |
| s4rd           | Процент редиспетчеризаций нити, происходящих на той же самой книжке (book – набор MCM), но не на том же самом MCM | Нить начинает удаляться от своего домашнего месторасположения. Все пути холодные, и пути к ранее размещенным данным становятся длиннее. Эта метрика должна иметь небольшое значение |
| s5rd           | Процент редиспетчеризаций нити, происходящих на другую книжку (набор MCM)                                         | Это самый плохой сценарий. Эта метрика должна иметь небольшое значение. В терминах производительности это значительно более неприемлемо, чем состояние s4rd                         |
| sysc           | Количество системных вызовов                                                                                      |                                                                                                                                                                                     |
| us, sy, wa, id | Утилизация логического ЦП в контексте пользователя системы в ожидании ввода-вывода и в состоянии простоя          |                                                                                                                                                                                     |
| pc             | Потребление физического процессора                                                                                | Доступно только с активированным режимом SMT и в разделах общего пула                                                                                                               |
| %ec            | Процент использованной выделенной мощности                                                                        | Только для разделов общего пула                                                                                                                                                     |
| icls           | Количество непроизвольных переключений контекста логического процессора                                           | Происходит по истечении кванта времени логического процессора                                                                                                                       |
| vccls          | Количество принудительных переключений контекста логического процессора                                           | Принудительные переключения контекста инициируются семейством вызовов (hcalls) h_cede и h_confer                                                                                    |

<sup>a</sup> MCM – Multi Chip Module. Прим. науч. ред.

<sup>b</sup> В процессоре POWER5 один общий L2-кэш размером 1.9 МБ. Прим. науч. ред.

## 6.5.6. Мониторинг с помощью PLM

Partition Load Manager (PLM) можно использовать в режиме мониторинга или в режиме управления. В обоих режимах PLM предоставляет общую информацию об управляемых им разделах. Команда PLM для режима мониторинга – `xlpstat`; пример ее вывода показан в примере 6-26. Синтаксис команды:

```
xlpstat [-r] { -p | -f } filename [interval] [count]
```

Распространенный формат показан в примере 6-26; опция `-p` указывает, что список управляемых разделов должен быть взят из указанного файла политики; обычно используется тот же файл политики, который был указан при старте сервера PLM. Альтернативно можно указать список управляемых разделов в текстовом файле, по одному разделу в строке, и указать имя файла с опцией `-f`. Команда `xlpstat` будет опрашивать статус указанных разделов. В выводе команды активно управляемые разделы не будут отделены от остальных.

Опция `-r` дает вывод в неотформатированном виде, более удобном для парсинга скриптовыми языками.

**Пример 6-26.** Команда `xlpstat`

| # xlpstat -p 2_groups |     |      |       |      |     |       |       |             |  |
|-----------------------|-----|------|-------|------|-----|-------|-------|-------------|--|
| CPU                   |     |      |       |      | MEM |       |       |             |  |
| STAT                  | TYP | CUR  | PCT   | LOAD | CUR | PCT   | PGSTL | HOST        |  |
| group2:               |     |      |       |      |     |       |       |             |  |
| up                    | S   | 0.5  | 4.00  | 0.10 | 512 | 75.17 | 0     | plmserver   |  |
| up                    | S   | 0.50 | 85.45 | 0.44 | 512 | 99.17 | 129   | vio_client2 |  |
| group1:               |     |      |       |      |     |       |       |             |  |
| up                    | D   | 1.00 | 95.09 | 0.19 | 512 | 99.23 | 129   | app_server  |  |
| up                    | D   | 1.00 | 0.39  | 0.09 | 512 | 74.73 | 0     | db_server   |  |

На экране в отдельной строке показан статус каждого раздела; разделы объединены в группы PLM. В примере есть две группы PLM.

Столбец `STAT` показывает, работает раздел (`up`) или нет (`down`). В примере все разделы работают.

В столбце `TYP` показано, какие процессоры использует раздел – общие (`S`), выделенные (`D`) или, если `xlpstat` не может опросить раздел, – `U` (обычно это обозначает наличие проблем с соединением). В примере 6-26 разделы в группе 2 используют общие процессоры, а разделы в группе 1 – выделенные.

Следующие шесть столбцов разбиты на две группы по три: одна – для отображения загруженности ЦП, а другая – для отображения загруженности памяти. В столбце `CUR` показано текущее выделение ЦП и памяти, а в столбце `PCT` – процент загрузки. В столбце `LOAD` показана загрузка процессора, измеренная PLM, а в столбце `PGSTL` – загрузка памяти, измеренная по количеству замещений страниц (page steal rate). В столбце `HOST` показано имя управляемого раздела.

### 6.5.7. Performance Workbench

Performance Workbench – это графический интерфейс для мониторинга процессов и активности системы. Он состоит из двух окон; в первом показана конфигурация раздела, а во втором – наиболее активные процессы, которые могут быть отсортированы по различным метрикам.

Performance Workbench является модулем, подключаемым к среде разработки Eclipse. Он находится в программном наборе bos.perf.gtools.perfbw. Пакет Eclipse IDE для AIX 5L находится в eclipse2.rte. Графическая среда Eclipse требует также установки X11 и Motif.

Для запуска Performance Workbench используется команда `perfwb`. На рисунке 6-21 показано окно Proctmon системы Performance Workbench.

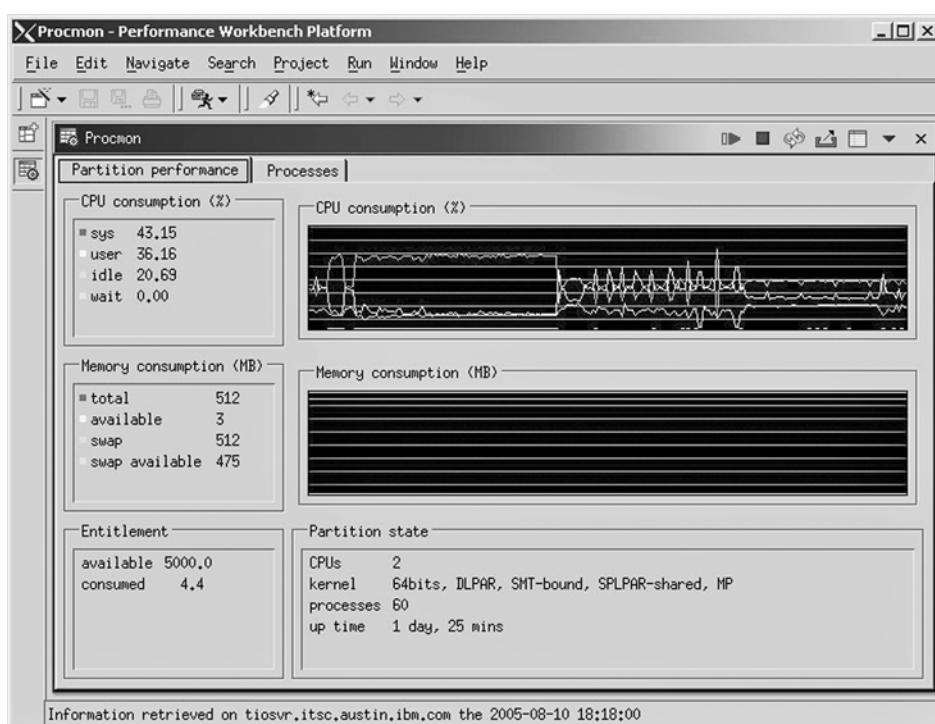


Рис. 6-21. Performance Workbench: окно Proctmon

### 6.5.8. Команда pmon

Команда `pmon` – это общедоступный (freeware) инструмент мониторинга для AIX 5L. Он предоставляет текстовый отчет о ключевых метриках системы, аналогичный команде `topas`. Дополнительный инструмент, `nmon analyzer`, предоставляет простой способ преобразования текстового отчета в графический формат или в формат электронных таблиц.

В версии 10 `pmon` знает о режиме SMT и разделах.

Команда `nmon` доступна на сайте:

[http://www.ibm.com/developerworks/views/download.jsp?contentid=91235&filename=es-nmon\\_analyser.zip&method=ftp&locale=worldwide](http://www.ibm.com/developerworks/views/download.jsp?contentid=91235&filename=es-nmon_analyser.zip&method=ftp&locale=worldwide)

Распакуйте и установите файл `nmon_aix53`, обычно в каталог `/usr/sbin/nmon`. Опционально измените и проверьте опции `iostat` для постоянного ведения истории дискового ввода-вывода. Используйте для этого следующие команды:

```
chdev -l sys0 -a iostat=true
lsattr -D -l sys0 -a iostat
```

Если не поставить эту опцию в положение `true`, возможна выдача предупреждения, но `nmon` будет продолжать выдавать статистику загрузки диска.

Если у вас большое количество дисков (более 200), то установка `iostat` в `true` начнет потреблять процессорное время (примерно 2 процента). По завершении процедуры измерений вы должны установить опцию `iostat` в `false`.

При использовании опции `-L` можно получить информацию о разделах, как показано на рис. 6-22.

```
--nmon-v10p---L=LargePageStats---Host=vio_client2---Refresh=8 secs---14:37.18--
--Shared-CPU-Logical-Partition-Stats-----
-Partition:Number=1 "VIO_client2"-----
-Flags: LPARed DRable SMT-bound Shared Capped PoolAuth-----
-Summary: Entitled= 0.50 Used 0.50 (99.7%) 6.2% of CPUs in System-----
- PoolCPUs= 6 Unused 5.49 8.3% of CPUs in Pool-----
-CPU-Stats----- Capacity----- ID-Memory-----
-Physical(+CUoD) 16 Cap. Processor Min 0.10 LPAR ID Group:Pool 32769:0-----
-Active (in_sys) 8 Cap. Processor Max 2.00 Memory(MB) Min:Max 128:1024-----
-Virtual Online 1 Cap. Increment 0.01 Memory(MB) Online 512-----
-Logical Online 2 Cap. Unallocated 0.00 Memory Region LMB 32MB min-----
-Physical pool 6 Cap. Entitled 0.50 Time-----Seconds-----
-SMT threads/CPU 2 -MinReqVirtualCPU 0.10 Time Dispatch Wheel 0.0100-----
-CPU-----Min-Max Weight----- MaxDispatch Latency 0.0100-----
-Virtual 1 32 Weight Variable 0 Time Pool Idle 5.4870-----
-Logical 1 64 Weight Unallocated 0 Time Total Dispatch 0.4983-----
```

**Рис. 6-22.** Экран LPAR команды `nmon`

Команда `nmon` использует переменную окружения `NMON` для определения формата отображения по умолчанию. Например, вы можете использовать:

```
export NMON=cmt
nmon
```

В результате вы увидите экран использования ЦП и памяти и экран наиболее активных процессов, расположенные один под другим.

### Инструмент `nmon` на VIOS

Последняя версия команды `nmon` может осуществлять мониторинг использования ресурсов на Virtual I/O Server.

Для установки `nmon` на VIOS используйте учетную запись `padmin` для передачи в раздел VIOS по FTP исполняемого файла `nmon`, извлеченного из загруженного ранее пакета. Файл надо переименовать в `nmon`:

```
ftp vio_server
(user padmin)
> put nmon_aix53 nmon
```

Войдите на VIOS командой `telnet` или через НМС и установите биты исполнения на файл:

```
chmod 550 nmon
```

Поменяйте атрибуты системы для ведения истории дискового ввода-вывода; команда аналогична AIX 5L:

```
chdev -dev sys0 -attr iostat=true
lsdev -dev sys0 -attr iostat
```

Теперь вы можете использовать `nmon` в интерактивном режиме.

```
NMON=cmt nmon
```

Вы можете использовать стандартные опции `nmon`.

### Протоколирование информации при помощи `nmon`

Используя `nmon`, вы можете протоколировать использование ресурсов для дальнейшего анализа посредством `nmon analyzer tool`. Это работает как в стандартных разделах AIX 5L, так и в VIOS:

```
nmon -f -t -s <interval> -c <count>
```

Процесс `nmon` выполняется в фоновом режиме, при желании можно выйти (`log off`) из раздела. Для получения лучших результатов параметр `count` не должен превышать значение 1500. Команда создаст файл, именованный в следующем формате:

```
<nom-part>_<date>_<hour>
```

После завершения процесса протоколирования необходимо переслать файл на систему с ПО электронных таблиц Microsoft® Excel® для запуска инструмента анализа `nmon analyzer tool`.

При мониторинге поведения системы в виртуализованной среде необходимо осуществлять одновременный мониторинг VIOS и клиентских разделов. Установите на НМС значение атрибута «Allow shared processor pool utilization authority» в значение `true` хотя бы для одного из контролируемых разделов (это потребует перезагрузки раздела).

### 6.5.9. AIX Performance Toolbox

Пакет AIX Performance Toolbox (PTX®) поддерживает разделы с общими и выделенными процессорами. Он включает в себя мониторы для виртуализированной среды, такие как выделенная мощность (entitlement) и потребленная мощность (consumed entitlement).

### 6.5.10. Осведомленность о динамической реконфигурации

Команды `vmstat`, `iostat`, `sar`, `mpstat` и `lparstat` осведомлены о динамической реконфигурации. Это означает, что они способны обнаружить изменения в системе из-за динамической реконфигурации (динамической операции с LPAR). Эти команды начинают свой вывод с предзаголовка с суммарной информацией о конфигурации, и каждый раз при изменении конфигурации выводится предупреждение и предзаголовок с информацией о новой конфигурации.

В примере 6-27 показано, как это работает для команды `vmstat` при добавлении виртуального процессора. Так как режим SMT активирован, добавление одного виртуального процессора приведет к добавлению двух логических процессоров, в результате будет два события динамической реконфигурации. Строки, выделенные шрифтом, показывают вывод команды `vmstat`, вызванный изменением конфигурации<sup>1</sup>.

#### Пример 6-27. Динамическая реконфигурация и команда vmstat

```
vmstat 2 6
System configuration: lcpsi=2 mem=512MB ent=0.50
kthr memory page faults cpu

r b avm fre re pi po fr sr cy in sy cs us sy id wa pc ec
0 0 92678 16964 0 2 0 0 0 0 2 25 182 0 0 97 2 0.00 1.0
1 0 92678 16964 0 0 0 0 0 0 1 5 147 0 0 99 0 0.00 0.8
System configuration changed. The current iteration values may be inaccurate.
1 0 9309115932 0 207 0 0 0 0 3687411147 3 15 59 23 0.14 28.7
System configuration: lcpsi=3 mem=512MB ent=0.50
kthr memory page faults cpu

r b avm fre re pi po fr sr cy in sy cs us sy id wa pc ec
System configuration changed. The current iteration values may be inaccurate.
0 0 9320115678 0 31 0 0 0 0 2 656 328 1 17 77 5 0.13 25.2
System configuration: lcpsi=4 mem=512MB ent=0.50
kthr memory page faults cpu

r b avm fre re pi po fr sr cy in sy cs us sy id wa pc ec
0 0 93329 15550 0 0 0 0 0 0 2 10 152 0 1 99 0 0.01 1.2
0 0 93329 15550 0 0 0 0 0 0 1 5 150 0 0 99 0 0.00 1.0
```

## 6.6. Соображения по подбору количества ресурсов

Технологии виртуализации серверов, обсуждаемые в этом руководстве, добавляют гибкость в вычислительную инфраструктуру. Но настоящая вычислительная мощность платформы при ее виртуализации не меняется. Однако производительность приложений и возможность реагирования на бизнес-требования при виртуализации улучшаются, так как виртуализация позволяет вам выделять приложениям ресурсы в соответствии с текущими бизнес-требованиями. Инструменты мониторинга ресурсов и рабочей нагрузки постоянно контролируют загрузку системы и оперативно переназначают выделения ресурсов в ответ на любое изменение. В этом и заключается настоящая мощь технологий виртуализации IBM.

В этом разделе даны некоторые руководящие принципы конфигурирования разделов на серверах IBM System p5.

<sup>1</sup> В оригинале книги нужные строки не выделены шрифтом. Это строки «System Configuration». Прим. науч. ред.

Гипервизор POWER является частью всех серверов IBM System p5; невозможно сконфигурировать сервер без гипервизора POWER. Все тесты производительности на серверах IBM System p5, опубликованные IBM, выполнены с гипервизором POWER. Когда вы выбираете сервер с каким-то конкретным rPerf<sup>1</sup>, он (rPerf) является потенциалом производительности, доступной вашим приложениям.

### **6.6.1. Соображения по конфигурации разделов**

В этом разделе обсуждаются некоторые моменты, которые необходимо рассмотреть при настройке разделов на сервере IBM System p5.

#### **Количество разделов**

Как общее правило, количество разделов должно быть по возможности минимальным. Предпочтительнее консолидировать несколько приложений в одном разделе AIX 5L, чем создавать по разделу для каждого приложения. Иногда, по техническим или организационным причинам, это невыполнимо, например настройка AIX 5L под транзакционные приложения баз данных может снизить производительность других типов приложений.

Каждый раздел должен управляться как отдельно стоящий сервер, он требует настройки, резервного копирования и лицензий на ПО. Сохранение небольшого количества разделов оказывает непосредственное влияние на стоимость администрирования и, следовательно, на общую стоимость владения любым сервером.

Далее, каждый раздел имеет связанные с ним ресурсы. Эти ресурсы имеют контекстную информацию и должны управляться ПО виртуализации. Это управление информацией состояния потребляет ресурсы, которые могли бы быть выданы разделам.

#### **Максимумы ресурсов**

Один из параметров определения раздела – это максимальное количество любого конкретного ресурса, будь то память, процессоры или виртуальные слоты ввода-вывода. Может возникнуть желание обозначить максимальные значения этих величин такими, чтобы быть уверенным в том, что всегда будет возможным увеличить количество ресурсов раздела, используя динамическую реконфигурацию. Однако гипервизор POWER, так же как и операционная система, должен содержать структуры данных, которые позволяют ему управлять всеми разделами, даже если они получат максимально возможное количество ресурсов. Гипервизор получит необходимую память, сделав ее недоступной разделам. Таким образом, вы должны указывать только реалистичные значения максимального количества ресурсов, используя запланированные предельные параметры.

#### **Количество виртуальных ЦП**

С максимумами ресурсов связаны количество виртуальных процессоров в любом из разделов и сумма всех виртуальных процессоров во всех разделах общего пула.

<sup>1</sup> rPerf, relative performance – тест для оценки средней коммерческой производительности систем IBM System p5 (и предыдущих серий на базе архитектуры POWER). Подробную информацию о тесте rPerf можно найти на сайте IBM System p5. Прим. науч. ред.

При настройке количество виртуальных процессоров в общем пуле должны быть учтены следующие общие правила:

- ▶ Количество виртуальных процессоров в разделе общего пула не должно превышать количества физических процессоров в общем процессорном пуле.
- ▶ Количество виртуальных процессоров в ограниченном разделе общего пула не должно превышать выделенную мощность, округленную в большую сторону до ближайшего целого числа.
- ▶ Для версий AIX 5L до V5.3 Maintenance Level 3 или для более поздних версий с отключенной функцией свертки виртуальных процессоров (virtual processor folding) сумма всех виртуальных ЦП во всех активных разделах общего пула не должна быть больше, чем количество физических процессоров в общем пуле, умноженное на четыре.
- ▶ Для версий AIX 5L после Version 5.3 ML3 с включенной функцией свертки виртуальных процессоров чрезмерное количество виртуальных ЦП имеет очень маленькое влияние на производительность.

### **Разделы с ограничением или без?**

Ограниченные разделы никогда не превышают выделенной им мощности, даже если они перегружены и в системе есть неиспользуемые ресурсы. Как правило, использование ограниченных разделов неоптимально; используйте разделы без ограничений и устанавливайте приоритет выделения резервной мощности, ис-пользуя веса разделов.

### **6.6.2. Виртуализация и приложения**

Некоторые приложения и ПО промежуточного уровня (middleware) не могут адаптироваться к изменениям динамической реконфигурации, например приложение запускает определенное количество процессов, базируясь на количестве сконфигурированных процессоров. Если приложению потребуется дополнительная вычислительная мощность, добавление дополнительных процессоров не даст эффекта без остановки и перезапуска приложения. При использовании общих процессоров можно изменить мощность виртуальных процессоров для изменения количества вычислительных ресурсов, доступных приложению. Так как количество процессоров не изменилось, нет нужды перезапускать приложение, не осведомленное о динамической реконфигурации.

Если приложение или рабочая среда чувствительно к размеру кэша или не может переносить нестабильность, связанную с совместным использованием ресурсов, оно должно быть размещено в разделе с выделенными процессорами и отключенным SMT. В разделах с выделенными процессорами весь физический процессор эксклюзивно выделяется одному разделу.

### **6.6.3. Управление ресурсами**

PLM и WLM предоставляют средства управления ресурсами и рабочей нагрузкой. Некоторые приложения и ПО промежуточного уровня предоставляют свои собственные средства управления ресурсами. Особенно это относится к СУБД. При использовании инструментов управления ресурсами AIX 5L и гипервизора POWER совместно с такими приложениями и ПО среднего уровня следует соблюдать осторожность.





7

## Partition Load Manager

В этой главе рассказывается о менеджере управления нагрузкой в разделах – Partition Load Manager (PLM). В ней Вам покажут, как установить и настроить его для управления ЦП и памятью. Глава состоит из следующих разделов:

- ▶ Введение в Partition Load Manager
- ▶ Система мониторинга и контроля ресурсов – Resource Monitoring and Control (RMC)
- ▶ Управление ресурсами
- ▶ Установка и настройка Partition Load Manager
- ▶ Реконфигурация по расписанию
- ▶ Советы по настройке и устранение неисправностей PLM
- ▶ Соглашения и ограничения PLM

## **7.1. Введение в Partition Load Manager**

Partition Load Manager (PLM) для AIX 5L предназначен для автоматизации администрирования (распределения) ресурсов ЦП и памяти между логическими разделами в пределах одного комплекса (central electronics complex, CEC). Для улучшения использования системных ресурсов PLM автоматизирует миграцию ресурсов между разделами, основываясь на загрузке и приоритетах разделов; разделы с большими запросами будут получать ресурсы, добровольно отданые или принудительно отобранные у разделов с меньшими запросами. Определяемые пользователями политики управляют тем, как будут перемещаться ресурсы. PLM не будет противоречить определениям разделов в HMC. PLM позволяет администраторам контролировать использование ресурсов в управляемых разделах.

PLM является частью опции Advanced POWER Virtualization. Он поддерживается как в разделах с выделенными процессорами, так и в разделах, работающих в общем процессорном пуле под управлением AIX 5L V5.2 (ML4), AIX 5L V5.3.0 или более новых версий на серверах IBM System p5.

### **7.1.1. Режимы работы PLM**

PLM может быть запущен в одном из двух режимов:

- ▶ Режим мониторинга
- ▶ Режим управления

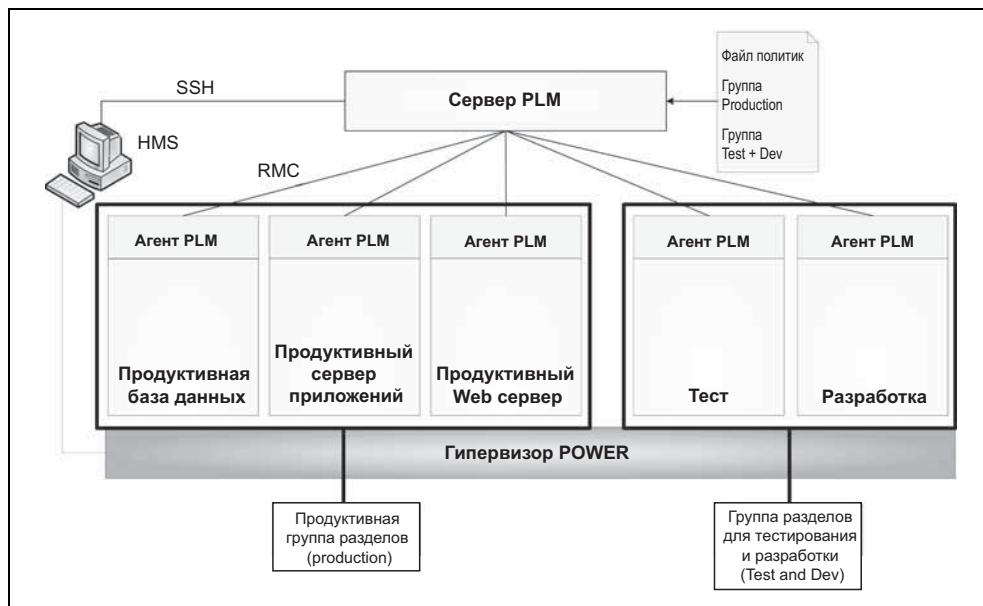
В режиме мониторинга PLM предоставляет наборы статистических отчетов об использовании ресурсов в управляемых разделах. Это обсуждается в главе 6.5.6 «Мониторинг при помощи PLM».

В режиме управления PLM будет инициировать операции динамической реконфигурации (DR) для установки соответствия системных ресурсов загрузке раздела в соответствии с определенной политикой.

### **7.1.2. Модель управления**

PLM использует клиент-серверную модель, показанную на рис. 7-1, для мониторинга и управления ресурсами разделов. Клиенты выступают как агенты (agents) на каждом из управляемых разделов. Сервер PLM конфигурирует каждого из агентов (clients), устанавливая пороговые значения, о достижении которых сервер должен быть проинформирован. Агенты контролируют использование ресурсов раздела и извещают сервер PLM, когда достигаются установленные им пороговые значения (понижение или повышение загрузки относительно порогового значения). Базируясь на определяемой пользователем политике управления ресурсами, сервер вызывает операции динамической реконфигурации (DR) через HMC для перемещения ресурсов из резервного пула в раздел или между разделами.

PLM позволяет создавать группы разделов. Ресурсы в пределах группы управляются независимо. На рис. 7-1 показаны две группы разделов: одна – для производственных разделов и другая – для тестирования и разработки.



**Рис. 7-1.** Архитектура PLM

**Замечания.**

- ▶ Сервер PLM может находиться как в разделе на том же сервере, что и управляемые разделы, так и на отдельной машине. Когда сервер PLM работает в разделе, он способен управлять своим собственным разделом.
- ▶ Несколько серверов Partition Load Manager могут выполняться на одной системе AIX 5L<sup>1</sup>.
- ▶ Различные группы PLM на данном сервере могут управляться разными серверами PLM.
- ▶ Раздел может иметь не более одного менеджера PLM.
- ▶ Необязательно все разделы в системе должны быть управляемыми.
- ▶ Один сервер Partition Load Manager может управлять разделами в пределах только одного управляемого СЕС.
- ▶ Невозможно иметь разделы, работающие в общем процессорном пуле, и разделы с выделенными процессорами в одной группе разделов PLM.
- ▶ Ресурсы ограничены принадлежностью к группе: раздел в одной группе PLM никогда не получит ресурсов от другого раздела из другой группы.
- ▶ В группе разделов должно быть как минимум два активных раздела.

Так как каждый раздел контролируется локально и агенты взаимодействуют с сервером PLM только при возникновении события, PLM потребляет незначительное количество системных и сетевых ресурсов.

<sup>1</sup> Имеется в виду сервер IBM System p5. Прим. науч. ред.

### 7.1.3. Политики управления ресурсами

Политика управления ресурсами (resource management policy) для каждой группы разделов (partition group) указывается в файле политики (policy file), который определяет как управляемое окружение, так и параметры политики управления ресурсами. Детали политик PLM обсуждаются в главе 7.2.5 «Определение групп разделов и политик».

Различные состояния разделов и пороговые значения загрузок показаны на рис. 7-2. Для каждого раздела существуют верхнее и нижнее пороговые значения

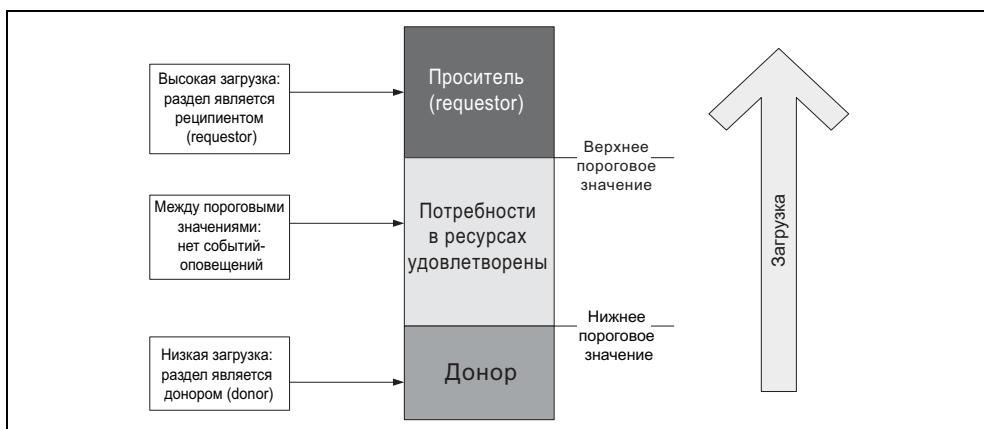


Рис. 7-2. Пороговые значения загрузки ресурсов

загрузки. Каждый раз, когда преодолевается пороговое значение, PLM получает событие Системы управления и контроля – Resource Management and Control (RMC). Когда загрузка ресурса превышает верхнее пороговое значение, раздел PLM считает, что раздел нуждается в дополнительных ресурсах; такой раздел называется *requestor* (запросчик, реципиент). Когда загрузка раздела становится меньше, чем нижнее пороговое значение, раздел становится потенциальным донором (*donor*). При нормальном стечении обстоятельств ресурсы перемещаются от доноров только тогда, когда другой раздел переходит в состояние *requestor* для того же ресурса. Когда загрузка ресурса находится между двумя пороговыми значениями, PLM считает, что доступных ресурсов достаточно.

Существует четыре места, в которых может быть указана политика управления группой ресурсов. В порядке увеличения приоритетов это:

- Значение по умолчанию PLM
- Значение по умолчанию экземпляра сервера PLM
- Спецификация политики группы
- Спецификация политики раздела

PLM имеет встроенные значения по умолчанию для настраиваемых параметров. Если эти параметры больше нигде не были указаны, то значения по умолчанию будут использоваться в политике. Пользователь может также указать значения по умолчанию для всех групп, управляемых сервером (экземпляром PLM, PLM instance), для всех разделов в данной группе (политика группы, group policy) или

для индивидуальных разделов. Значения, указанные в политике раздела (partition policy), имеют преимущество перед всеми остальными.

Файл политики (policy file) после загрузки является статическим; приоритет раздела не изменится при появлении высокоприоритетной работы. Приоритет разделов может быть изменен только путем загрузки новой политики. Файлы политики могут быть изменены «на лету» без установки PLM.

### Размещение ресурсов

Часть определения политики – относительный приоритет каждого раздела в группе. Это достигается путем использования механизма «общих ресурсов» (shares) аналогичного AIX 5L

Workload Manager (WLM). Чем больше «shares» выделено разделу, тем выше его приоритет. Чтобы предотвратить полное истощение количества ресурсов в некоторых разделах, PLM модулирует приоритет раздела на основе его текущего количества ресурсов.

Когда PLM оповещается о том, что раздел перешел в состояние реципиента, он ищет свободные ресурсы в следующем порядке:

- Свободный пул неразмещенных ресурсов
- Донор ресурсов
- Раздел с наименьшим количеством общих (требуемых) ресурсов, имеющий при этом больше ресурсов, чем указано в параметре настройки «guaranteed» (гарантировано).

Если есть доступные ресурсы в пуле свободных ресурсов, они будут отданы запрашивающему разделу. Если доступных ресурсов в пуле нет, будет проверен пул доноров ресурсов. Если есть донор ресурсов, ресурс будет перемещен от него запрашивающему. Количество перемещаемых ресурсов – либо минимальное дельта-значение для обоих разделов, либо количество, которое даст им приоритет, указанный в политике. Если нет доноров ресурсов, проверяется список разделов с большим количеством ресурсов, чем им гарантировано.

Определение, какой из узлов больше или меньше требуется в ресурсах, выполняется путем сравнения того, каким количеством ресурсов владеет раздел относительно его приоритета, определяемого, в свою очередь, по количеству «общих ресурсов». PLM ранжирует разделы, включая раздел-requestor, в списке разделов с количеством ресурсов, превышающих гарантированное значение. Приоритеты разделов определяются по следующему соотношению:

$$\text{Приоритет} = \frac{(\text{текущее количество ресурсов} - \text{гарантированное количество ресурсов})}{\text{общие ресурсы}}.$$

Наименьшее результатирующее значение означает наивысший приоритет; разделы с меньшим значением могут получать ресурсы от разделов с большим значением.

На рис. 7-3 показан процесс перемещения ресурсов ЦП для трех разделов. Раздел 3, загруженный, является просителем. В пуле нет свободных ресурсов, нет и разделов-доноров. PLM ищет разделы с избыточным количеством ресурсов (большим, чем им гарантировано). Остальные два раздела в группе имеют избыточные ресурсы. Раздел 1 имеет наибольшее соотношение «избыточные–общие ресурсы» из всех трех разделов, и ресурсы будут перемещаться от раздела 1 к разделу 3.

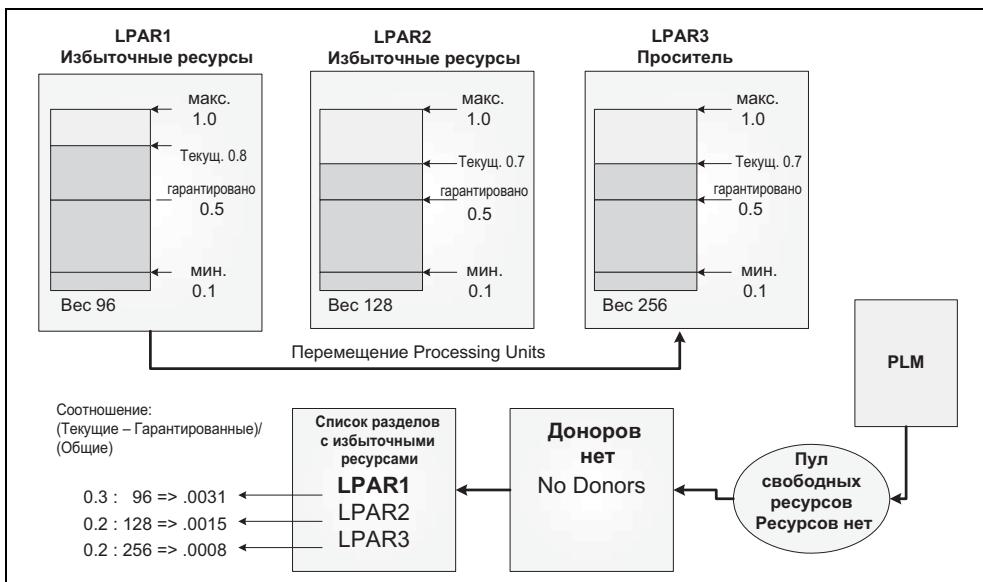


Рис. 7-3. Распределение ресурсов PLM

Если запрос ресурсов не может быть выполнен, он помещается в очередь и проходит повторную оценку, когда появятся свободные ресурсы.

#### Ограничения размещения ресурсов

При указании политик PLM нужно учитывать ограничения:

- Минимальное, гарантированное и максимальное значения должны удовлетворять соотношению: минимальное  $\leq$  гарантированное  $\leq$  максимальное.
- Если минимальное, гарантированное и максимальное значения одинаковы или максимум для группы установлен равным нулю, PLM не будет управлять ресурсом.
- Вне зависимости от приоритета PLM не даст разделу выйти за границы максимума и минимума для каждого ресурса.
- Диапазон максимумов и минимумов PLM должен быть в пределах максимумов и минимумов, установленных на НМС; если нет, то будет использоваться пересечение диапазонов НМС и PLM.
- Если не указаны значения максимумов и минимумов PLM, будут использоваться значения НМС.

#### 7.1.4. Управление памятью

PLM управляет памятью, перемещая LMB (Logical Memory Blocks) между разделами. Размер LMB зависит от количества памяти, установленной в СЕС. Он варьируется в диапазоне от 16 до 256 МБ. Размер LMB можно изменить через Advanced System Management Interface (ASMI) на НМС.

Для определения потребности в памяти PLM использует две метрики:

- Процент утилизации (соотношение используемой и установленной памяти).
- Скорость замещения страниц (page replacement rate).

Детальнее загрузка памяти обсуждается в главе 7.6.2 «Как оценивается загрузка», дополнительные детали управления памятью представлены в главе 7.6.4, «Управление ресурсами памяти».

AIX 5L использует всю доступную ему память. Он не будет выгружать страницы из памяти, если ему не потребуется загрузить другие страницы с диска. Это означает, что если есть избыток памяти, то AIX 5L будет использовать ее и она будет показана как используемая инструментами AIX 5L, даже если нет приложений, использующих ее. По этой причине разделы редко становятся донорами.

### 7.1.5. Управление процессором

Для разделов с выделенными процессорами PLM перемещает физические процессоры (по одному) от разделов, которые не используют их или имеют больший вес избыточности, разделам, которые имеют потребность в них. Это позволяет активным разделам с выделенными процессорами лучше использовать свои ресурсы, например, выравнивая переход от транзакций конца дня к ночным пакетным работам.

Для разделов в общем процессорном пуле PLV управляет выделенной мощностью (CE) и количеством виртуальных процессоров (VP). Когда раздел запрашивает больше процессорной мощности, PLM увеличит выделенную мощность (CE) для запрашивающего раздела, если дополнительная процессорная мощность доступна. PLM может увеличить количество виртуальных процессоров для увеличения потенциальной возможности раздела потреблять процессорные ресурсы при сильной загрузке для разделов с ограничением (capped) и без ограничения (uncapped). PLM может также, наоборот, уменьшить выделенную мощность и количество виртуальных процессоров при низкой нагрузке для более эффективного использования физических процессоров.

**Замечание.** Оптимизация «свертывания» виртуальных процессоров была представлена в AIX 5L V5.3 ML3, делая управление PLM количеством виртуальных процессоров ненужным в большинстве ситуаций, но убиение виртуальных процессоров более эффективно, чем «свертка» (VP folding), так что при некоторых обстоятельствах управление PLM виртуальными процессорами может быть предпочтительным. См. «Свертка виртуальных процессоров».

Управление процессорами более детально обсуждается в главе 7.6.3 «Управление ресурсами ЦП».

### 7.1.6. Resource Monitoring and Control (RMC)

PLM использует подсистему Мониторинга и Контроля Ресурсов – Resource Monitoring and Control (RMC) для сетевых коммуникаций, которые обеспечивают стабильное окружение для мониторинга и управления ресурсами.

Resource Monitoring and Control (RMC) – это структура для обеспечения коммуникаций и обработки событий, использующаяся PLM. В этой главе – краткое вве-

дение в ключевые компоненты и характеристики RMC, необходимые для понимания того, как работает PLM.

Для получения дополнительной информации о RMC, см. руководство A Practical Guide for Resource Monitoring and Control (RMC), SG24-6615 и документацию по AIX 5L.

RMC – это функция, дающая Вам возможность контролировать состояние ресурсов системы и реагировать на пересечение предопределенных пороговых значений, так что Вы можете автоматизировать большое количество рутинных задач. RMC входит в комплект AIX 5L как бесплатная опция и устанавливается по умолчанию вместе с операционной системой. RMC – это часть функционального набора Reliable Scalable Cluster Technology (RSCT).

RMC контролирует ресурсы (дисковое пространство, использование ЦП, статус процессора, процессы приложения и т.д.) и выполняет действия в ответ на определенные условия. RMC может работать в автономном или кластерном (множество систем или разделов) окружении.

RMC позволяет Вам настроить действия-реакции или скрипты, которые управляют общими системными настройками с небольшим участием администратора или без него. Например, Вы можете настроить RMC на автоматическое увеличение файловой системы, если ее использование превысило 95 процентов.

При работе с PLM RMC настраивается в кластерном окружении как домен управления. В домене управления (management domain) узлы (nodes) управляются сервером управления (management server). Сервер управления знает о всех узлах, и все узлы знают о своем сервере управления. Однако управляемые узлы ничего не знают друг о друге. Взаимоотношение между управляющим и управляемыми узлами показано на рис. 7-4.

Сервер управления может размещаться в разделе или на удаленной системе. Сервер управления может управлять своим разделом.

**Замечание.** CSM может выполняться на той же системе, что и сервер управления PLM.

## 7.2. Установка и настройка Partition Load Manager

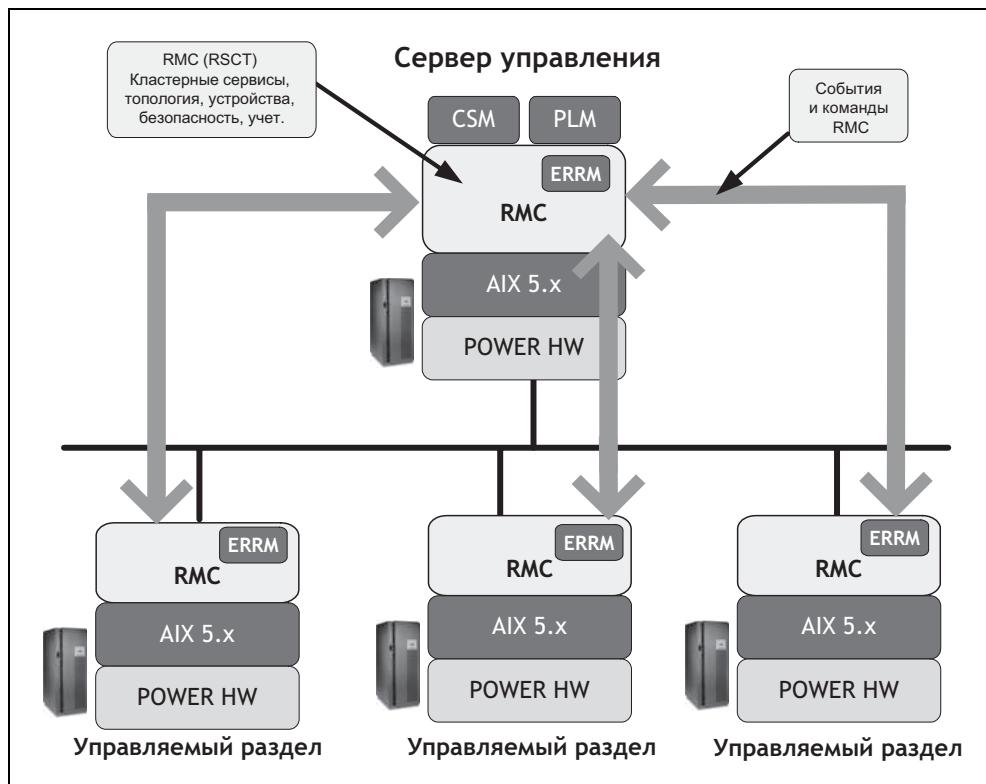
В этой главе проводится обзор установки и настройки PLM, настройки групп разделов и определения политик PLM для управления памятью и ЦП.

Дополнительную информацию можно найти в документации по AIX 5L:

- *Introduction to pSeries Provisioning*, SG24-6389
- *AIX 5L V5.3 Partition Load Manager Guide and Reference*, SC23-4883

Для установки и настройки PLM необходимо выполнить следующие шаги:

1. Подготовить окружение AIX 5L.
2. Установить и настроить OpenSSL и OpenSSH.
3. Создать файл политики.
4. Настроить RMC.
5. Проверить установку.



**Рис. 7-4.** Сервер управления RMC и управляемые разделы

### 7.2.1. Подготовка AIX 5L для PLM

В этой главе рассказывается о необходимой предварительной настройке перед установкой PLM.

#### Разрешение имен

Перед выполнением любой из следующих конфигурационных задач необходимо решить, будете ли Вы использовать полностью квалифицированные или короткие имена хостов. Полностью квалифицированные имена включают добавление имени домена к имени хоста, например my\_server.my\_domain.com. Если Вы выбираете полностью квалифицированные имена, Вы должны убедиться в том, что используемый Вами механизм разрешения имен возвращает полностью квалифицированные имена. Сервер PLM должен быть способен производить разрешение имен контролирующей НМС и всех управляемых разделов.

**Внимание.** Чтобы избежать сложностей именования PLM, RMC и ssh, сетевые имена должны соответствовать именам хостов для управляемых разделов, сервера PLM и НМС.

**Замечание.** Хотя и возможно создать специальный пользователь AIX 5L для работы PLM, использование пользователя root вызовет меньше сложности.

#### Команды rsh и rcp

Во время установки у сервера PLM должна быть возможность запуска удаленных скриптов shell на управляемых разделах и копировать файлы на них. PLM использует команды удаленного shell AIX 5L, rsh, и удаленного копирования, rcp, для этих задач, и они должны быть настроены до его установки. После того как PLM будет полностью настроен, их можно удалить.

Если по соображениям безопасности команды rsh и rcp были заблокированы, выполните следующие шаги для их активизации:

1. Отредактируйте файл .rhosts для пользователя root на каждом управляемом разделе – добавьте туда следующие строки:

```
plmserver1 root
plmserver1.mydomain.com root
```

Иногда более предпочтительным может быть редактирование файла /etc/hosts.equiv.

2. Разрешите использование команд rsh и rcp на каждом LPAR, используя следующие команды:

```
chmod 4554 /usr/sbin/rshd
chmod 4554 /usr/bin/rcp
```

3. Отредактируйте файл /etc/inetd.conf и раскомментируйте следующую строку:

```
shell stream tcp6 nowait root /usr/sbin/rshd rshd
```

4. Перезапустите демон inetd, используя следующую команду:

```
refresh -s inetd
```

5. Протестируйте доступ командой rsh от сервера Partition Load Manager к каждому управляемому разделу, используя следующие команды:

```
rsh root@lpar1 date
rsh root@lpar2 date
```

#### 7.2.2. Установка и настройка SSL и SSH

**Внимание.** Open Secure Shell (OpenSSH) основывается на Open Secure Sockets Layer (OpenSSL). Вы должны установить OpenSSL до установки OpenSSH.

#### OpenSSL

OpenSSL предоставляет безопасные криптографические библиотеки, использующиеся в SSH, он доступен как пакеты RPM на AIX Toolbox for Linux Applications CD, также Вы можете загрузить пакеты с Web-сайта AIX Toolbox for Linux Applications:

<http://www.ibm.com/servers/aix/products/aixos/linux/download.html>

OpenSSL находится в секции AIX Toolbox Cryptographic Content на Web-сайте: см. в рамке на правой стороне страницы. Вы должны заранее иметь или получить

IBM user ID для доступа к этой странице. На время написания этой книги последняя доступная версия – 0.9.7d-2. Загрузите и установите пакет RPM:

- openssl-0.9.7d-2.aix5.1.ppc.rpm

Два следующих пакета OpenSSL, openssl-devel и openssl-doc, необязательные при использовании OpenSSH на AIX 5L. Это средства для разработки и документация для OpenSSH.

Шаги установки:

1. Используйте команду rpm для установки RPM пакета OpenSSL RPM:

```
rpm -Uvh openssl-0.9.7d-2.aix5.1.ppc.rpm
openssl #####
```

2. Если пакет установлен корректно, Вы можете проверить статус установки, используя одну из следующих команд:

```
lslpp -L | grep openssl
openssl 0.9.7d-2 C R Secure Sockets Layer and
rpm -q openssl
openssl-0.9.7d-2
```

## OpenSSH

Программные средства OpenSSH предоставляют функции оболочки (shell) для шифрования сетевого трафика, аутентификации хостов и сетевых пользователей и обеспечения целостности данных. PLM использует SSH для коммуникаций с НМС и управляемыми разделами.

Для получения большей информации об OpenSSH на AIX 5L,смотрите следующий Web-сайт, на котором есть последние версии пакетов в формате installp для AIX 5L:  
<http://sourceforge.net/projects/openssh-aix>

На время написания этой книги последняя версия пакета – 3.8.1p1. Загрузите файл openssh-3.8.1p1\_53.tar.Z для AIX 5L V5.3 в отдельный каталог.

Для установки OpenSSH:

1. Распакуйте файлы из архива:

```
#zcat openssh-3.8.1p1_53.tar.Z | tar xvf -
```

2. Создайте toc-файл командой inutoc.

3. Установите пакеты, используя installp или smitty install\_latest:

```
installp -acv -Y -d . all
```

Опция -d. обозначает, что Вы запускаете команду в том каталоге, куда Вы распаковали файлы; в противном случае используйте соответствующий путь. Опция -Y обозначает, что Вы принимаете условия лицензии.

4. Проверьте установку следующей командой:

```
lslpp -L | grep ssh
openssl.base.client 3.8.0.5302 C F Open Secure Shell Commands
openssl.base.server 3.8.0.5302 C F Open Secure Shell Server
openssl.license 3.8.0.5302 C F Open Secure Shell License
openssl.man.en_US 3.8.0.5302 C F Open Secure Shell
openssl.msg.CA_ES 3.8.0.5302 C F Open Secure Shell Messages -
...
```

Демон sshd находится под контролем AIX SRC. Вы можете запустить, остановить его и просмотреть статус, используя следующие команды:

- ▶ `startsrc -s sshd` или `startsrc -g ssh` (группа)
- ▶ `stopsrc -s sshd` или `stopsrc -g ssh`
- ▶ `lssrc -s sshd` или `lssrc -g ssh`

В руководстве серии *IBM Redbook Managing AIX Server Farms*, SG24-6606 предоставлена информация о настройке OpenSSH в AIX 5L; оно доступно на следующем Web-сайте:

<http://www.redbooks.ibm.com>

## Обмен ключами SSH

PLM использует SSH для безопасного запуска удаленных команд на НМС без запроса ввода пароля. SSH должен быть настроен на разрешение доступа с раздела менеджера PLM пользователя – администратора PLM, в нашем примере – пользователя root. Это осуществляется путем создания пары ключей SSH на сервере PLM и экспортирования открытого ключа для пользователя hscroot на НМС. Это выполняется следующим образом:

1. Войдите в раздел сервера PLM как root.
2. Создайте ключи SSH на сервере Partition Load Manager, используя следующую команду:

```
$ ssh-keygen -t rsa
```

Оставьте passphrase<sup>1</sup> пустым, когда Вас спросят об этом. У команды должен быть следующий вывод:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/ssh/id_rsa):
Created directory '/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /ssh/id_rsa.
Your public key has been saved in /ssh/id_rsa.pub.
The key fingerprint is:
20:f5:d9:49:13:d7:2d:df:14:8c:a3:f6:ac:5e:d7:17 root@plmserver
```

Команда ssh-keygen создает каталог .ssh в домашнем каталоге. Содержание каталога следующее:

```
ls -l .ssh
total 40
-rw----- 1 root system 883 Aug 7 21:59 id_rsa
-rw-r--r-- 1 root system 227 Aug 7 21:59 id_rsa.pub
```

Файл id\_rsa.pub содержит открытый ключ SSH.

3. Добавьте открытый ключ SSH на НМС.

**Важно.** Так как НМС может уже иметь открытые ключи от других разделов, мы должны добавить наш ключ в существующий список открытых ключей на НМС, вместо того чтобы просто скопировать его.

<sup>1</sup> Пароль для шифрования ключа. Прим. науч. ред.

- a. Скопируйте список открытых ключей HMC на сервер PLM, ответьте yes на запрос:

```
$ scp hscroot@590hmc:.ssh/authorized_keys2 ~/ssh/hmc_authorized_keys2
The authenticity of host '590hmc (192.168.255.69)' can't be established.
RSA key fingerprint is 29:4b:1b:eb:1e:30:b6:da:ed:26:c7:0d:f6:2e:19:9a.
Are you sure you want to continue connecting (yes/no)?yes
Warning: Permanently added '590hmc, 192.168.255.69' (RSA) to the list of known hosts.
hscroot@590hmc's password:
authorized_keys2 100% 0 0.0KB/s 00:00
```

Эта команда создаст файл known\_hosts в каталоге .ssh:

```
$ ls -l .ssh
total 4
-rw-r--r-- 1 root system 0 Aug 09 22:07 hmc_authorized_keys2
-rw----- 1 root system 883 Aug 09 21:59 id_rsa
-rw-r--r-- 1 root system 227 Aug 09 21:59 id_rsa.pub
-rw-r--r-- 1 root system 225 Aug 09 22:07 known_hosts
```

В предыдущем примере файл hmc\_authorized\_keys2 пустой, что показывает, что это первый обмен ключами для HMC.

- b. Добавьте открытый ключ сервера PLM в список открытых ключей HMC:

```
$ cat ~/ssh/id_rsa.pub >>~/ssh/hmc_authorized_keys2
$ ls -l .ssh
total 5
-rw-r--r-- 1 root system 227 Aug 09 22:16 hmc_authorized_keys2
-rw----- 1 root system 883 Aug 09 21:59 id_rsa
-rw-r--r-- 1 root system 227 Aug 09 21:59 id_rsa.pub
-rw-r--r-- 1 root system 225 Aug 09 22:07 known_hosts
```

- c. Скопируйте обновленный список открытых ключей обратно на HMC:

```
$ scp .ssh/hmc_authorized_keys2 hscroot@590hmc:.ssh/authorized_keys2
hscroot@590hmc's password:
hmc_authorized_keys2 100% 227 0.2KB/s 00:00
```

- d. Завершите обмен ключами и проверьте конфигурацию SSH.

Начальный обмен SSH между двумя серверами производит обмен ключами SSH. Запустите команду ls удаленно на HMC, введя следующую команду на сервере PLM (Вас не должны спросить пароль):

```
ssh hscroot@590hmc ls
WebSM.pref
websm.script
```

Необходимо повторить эту команду ssh с полностью квалифицированным именем и IP-адресом HMC для полного обмена ключами. Это уменьшит потенциальную возможность возникновения проблем при настройке PLM и RMC:

```
$ ssh hscroot@590hmc.mydomain.com ls
$ ssh hscroot@192.164.10.10 ls
```

- e. Получите имя управляемой системы.

Используйте следующую команду на сервере PLM:

```
ssh hscroot@p5hmc1 lssyscfg -r sys -F name
```

Если имя управляемого раздела не было изменено на HMC, используя закладку Properties на управляемой системе (managed system), имя управляемой системы по умолчанию похоже на следующее:

Server-9119-590-SN02Cxxxx

### 7.2.3. Настройка RMC для PLM

Сервер Partition Load Manager использует Resource Monitoring and Control (RMC) для взаимодействия с управляемыми разделами.

Настройка ACL в RMC состоит из двух компонентов:

- ▶ Аутентификация хостов
- ▶ Авторизация пользователей

Аутентификация хостов включает в себя обмен открытыми ключами между сервером Partition Load Manager и управляемыми разделами. Это позволяет серверу PLM подключаться (создавать сессию) к управляемым разделам.

Авторизация пользователей включает в себя добавление записей в файл ACL для RMC и дать доступ root (на PLM) к требуемому классу ресурсов.

Вы можете настроить RMC, используя скрипт, расположенный в каталоге, или используя GUI Web-based System Manager, как показано в шаге 10 в главе 7.2.6 «Базовая настройка PLM».

Скрипт `plmsetup` автоматизирует обе эти задачи, используя команды удаленного shell. Процедура настройки использует следующие позиционные параметры:

- ▶ Идентификатор пользователя<sup>1</sup>, под учетной записью которого будет выполняться Partition Load Manager.
- ▶ Имя хоста (host name) раздела.
- ▶ Опционально имя сервера PLM, если команда запускается на управляемом разделе.

Для настройки с сервера PLM, используйте следующий синтаксис:

```
/etc/plm/setup/plmsetup lpar_hostname root
```

Параметр `lpar_hostname` – имя управляемого раздела. Скрипт нужно запустить для всех управляемых разделов.

Если удаленный shell недоступен или ненастроен на управляемых разделах, вы можете выполнить эти задачи самостоятельно. Запустите следующий скрипт shell как пользователь root на управляющей системе, на которой будет выполняться Partition Load Manager:

```
/etc/plm/setup/plmsetup lpar_hostname root plmserver
```

Параметр `lpar_hostname` – имя управляемого раздела, `plmserver` – имя системы или раздела, содержащего `plmserver`.

Если `plmserver` управляет своим разделом, Вы увидите сообщение, похожее на следующее:

```
rcp: /tmp/exec_script.335922 and /tmp/exec_script.335922 refer to the same file
(not copied).
```

Это нормальное сообщение.

После того как скрипт успешно выполнится, файл RMC ACL (Access Control), находящийся в каталоге `/var/ct/cfg/ctrmc.acls` на удаленной системе, будет иметь запись, похожую на следующую:

---

<sup>1</sup> Скрипту нужно указать имя пользователя, а не его ID. Прим. науч. ред.

```
tail -1 /var/ct/cfg/ctrmc.acls
root@nimmaster * rw
```

Этот идентификатор пользователя использовался для настройки файлов RMC ACL на управляемых разделах.

Настройка RMC для Partition Load Manager выполняется следующими шагами:

1. Выберите **Set up Management of Logical Partitions**. Аутентифицированное имя пользователя – root.
2. Выберите **Automatically setup with each partition in the policy file**. Имя файла политики – `/etc/plm/policies/plm_example`.
3. Нажмите **OK**.

Это также можно выполнить, используя командную строку, если Вы root на сервере Partition Load Manager. Для того чтобы сделать это, у Вас должен быть доступ к `rsh` и `rcp`. После того как настройка будет выполнена, Вы можете удалить файл `.rhosts`.

#### 7.2.4. Установка Partition Load Manager

Для установки сервера Partition Load Manager выполните следующие шаги:

1. Смонтируйте CD Partition Load Manager на Вашей системе.
2. Используя или команду `installp`, или быстрый путь `smitty install_latest`, установите следующие пакеты (filesets):
  - `plm.license`
  - `plm.server.rte`
  - `plm.sysmgt.websm`

Вы должны установить значение поля Accept Licence в yes, если Вы используете `smitty` или используйте флаг `-Y` при использовании команды `installp`.

#### 7.2.5. Определение групп разделов и политик

PLM использует файл политики, в котором определено, какие разделы будут управляемы, их гарантированные мощности (entitlements), минимальные и максимальные мощности.

Файл политик – стандартный плоский файл AIX 5L, который можно редактировать любым текстовым редактором; однако интерфейс Web-based System Manager к PLM предоставляет мастера для определения политики и заполнения файла политики. В этом документе описывается только использование мастера PLM для определения политики.

**Внимание.** Если Вы редактируете политику самостоятельно, учтите, что файл имеет жесткую структуру, основанную на разделах (stanza). Если эта структура не соблюдена, PLM не сможет использовать его. Вы должны сделать копию гарантированно корректного файла и редактировать только копию. Вы можете использовать команду `xlp1m -C -p policy_file`, где `policy_file` – отредактированный Вами файл, для проверки синтаксиса файла политики.

Определите политику PLM:

- 1 . Создайте новый файл политики.
  - 2 . Определите глобальное окружение и, дополнительно, глобальные настройки.
  - 3 . Определите группы разделов и, дополнительно, настройки групп.
  - 4 . Добавьте разделы в группы и, дополнительно, определите настройки разделов.
- В этой главе представлен краткий обзор параметров конфигурации и настроек. Для детальных шагов конфигурации обратитесь к главе 7.2.6 «Базовая настройка PLM» на странице 352.

### **Конфигурационные параметры и настройки**

Перед детализацией процесса создания файла политики необходимо понимание настроек PLM. Как обозначено выше, настройки могут быть указаны в трех различных местах в мастере политики PLM, и в дополнение к этим трем PLM имеет набор значений по умолчанию для некоторого числа из них. Любой настраиваемый параметр, указанный для раздела, имеет преимущество перед настройкой, указанной для группы, которая, в свою очередь, имеет приоритет перед глобальными значениями, которые имеют приоритет перед значениями PLM по умолчанию.

Глобальные настройки, группы и разделы имеют отдельные закладки в мастере настройки политики PLM.

**Замечание.** Следующие главы описывают каждую конфигурацию и настраиваемые параметры для файлов политики PLM. Хотя эти списки могут быть трудны для изучения, все из процитированных параметров имеют значения по умолчанию. Единственные обязательные части настройки PLM – это:

- Имя группы PLM
- Тип группы PLM: выделенная или общая<sup>1</sup>
- Список разделов в группе

### **Конфигурационные параметры**

PLM имеет четыре параметра для каждого управляемого ресурса, которые управляют тем, как PLM добавляет и удаляет память и процессоры к разделу и от раздела. Они показаны в таблице 7-1.

**Таблица 7-1.** Конфигурационные параметры PLM

| Параметр                               | Мин. | По умолчанию | Макс. | Описание                                                                                                                                                        |
|----------------------------------------|------|--------------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Memory minimum (Минимум памяти)        | -    | -            | -     | Минимальный объем памяти в разделе. PLM никогда не оставит раздел с меньшим количеством памяти, чем указано здесь. Значение по умолчанию – на HMC. <sup>a</sup> |
| Memory guaranteed (Гарантирано памяти) | -    | -            | -     | Гарантированный объем памяти. Значение по умолчанию – «желательное» на HMC.                                                                                     |

<sup>1</sup> Имеется в виду то, на каких процессорах работают разделы – выделенных или из общего пула. Прим. науч. ред.

*Продолжение табл.*

| Параметр                                  | Мин. | По умолчанию | Макс. | Описание                                                                                                                                                                                                                     |
|-------------------------------------------|------|--------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Memory maximum<br>(Максимум памяти)       | -    | -            | -     | Максимальное количество ресурсов памяти, которое PLM позволит иметь разделу. Значение по умолчанию – на HMC.                                                                                                                 |
| Memory shares<br>(«Общие ресурсы» памяти) | 0    | 1            | 255   | Фактор, используемый для определения того, как объем памяти, превышающий гарантированный, распределяется между разделами в группе. Значение 0 указывает, что раздел никогда не получит памяти больше, чем гарантировано.     |
| CPU minimum<br>(Минимум ЦП)               | -    | -            | -     | Минимальная мощность ЦП в разделе. PLM никогда не оставит раздел с меньшим количеством, чем указано здесь. Значение по умолчанию – на HMC.                                                                                   |
| CPU guaranteed<br>(Гарантирано ЦП)        | -    | -            | -     | Гарантиированная мощность ЦП, если загрузка раздела больше нижнего порогового значения средней загрузки ЦП. Значение по умолчанию – «желательное» на HMC.                                                                    |
| CPU maximum<br>(Максимум ЦП)              | -    | -            | -     | Максимальное количество ресурсов ЦП, которое PLM позволяет иметь разделу. Значение по умолчанию – на HMC.                                                                                                                    |
| CPU Shares<br>(«Общие ресурсы» ЦП)        | 0    | 1            | 255   | Фактор, используемый для определения того, как мощность ЦП, превышающая гарантированную, распределяется между разделами в группе. Значение 0 указывает, что раздел никогда не получит ресурсов ЦП больше, чем гарантировано. |

<sup>a</sup> Обратите внимание, что в оригинале руководства написано «HCM», однако имеется в виду HMC – Hardware Management Console.  
Прим. науч. ред.

#### **В чем разница между гарантированным и минимальным значениями?**

PLM будет обеспечивать то, что раздел имеет гарантированное количество ресурсов, если загрузка раздела больше, чем нижнее пороговое значение использования ресурса. Если использование ресурса меньше, чем этот лимит, то раздел становится донором и PLM будет удалять ресурсы до достижения жесткого лимита – минимального значения.

### **Настройки ЦП**

В таблице 7-2 показаны настройки, общие для всех процессорных ресурсов, а в таблице 7-3 показаны настройки, специфичные для виртуальных процессоров в разделах, работающих в общем пуле.

Эти настройки опциональны, они могут использоваться для настройки политики PLM. Не все из этих настроек доступны в закладке *globals* в мастере настройки PLM.

**Таблица 7-2.** Настройки, связанные с ЦП

| Настройка                                                                        | Мин. | По умолчанию | Макс. | Описание                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------------------------------|------|--------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU notify intervals (Количество интервалов для оповещения)                      | 1    | 6            | 100   | Количество непрерывных 10-секундных периодов опроса, в течение которых относящееся к ЦП значение должно превышать (быть ниже) пороговое значение для того, чтобы PLM инициировал действие.                                                                                                                                                                                                     |
| CPU load average high threshold (Верхнее пороговое значение средней загрузки ЦП) | 0.1  | 1.0          | 10.0  | Верхнее пороговое значение средней загрузки процессора. Считается, что раздел со средней загрузкой, большей, чем это значение, нуждается в большем количестве процессорной мощности (проситель).                                                                                                                                                                                               |
| CPU load average low threshold (Нижнее пороговое значение средней загрузки ЦП)   | 0.1  | 0.5          | 1.0   | Нижнее пороговое значение средней загрузки ЦП. Считается, что раздел со средней загрузкой, меньшей, чем это значение, имеет лишнюю процессорную мощность (донор).                                                                                                                                                                                                                              |
| Immediate release of free CPU (Немедленное освобождение свободных ЦП)            | -    | нет          | -     | Показывает, будет или нет неиспользуемая избыточная процессорная мощность удалена из раздела и помещена в общий процессорный пул. Значение «нет» показывает, что лишняя процессорная мощность остается в разделе до тех пор, пока в ней не станет нуждаться другой раздел. Значение «да» показывает, что лишняя процессорная мощность удаляется из раздела, если он больше не нуждается в ней. |

**Таблица 7-3.** Настройки, связанные с виртуальными процессорами

| Настройка                                            | Мин. | По умолчанию | Макс. | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------------|------|--------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Entitled capacity delta (Дельта выделенной мощности) | 1    | 10           | 100   | Процент увеличения СЕ процессора для добавления в или удаления из раздела, работающего в общем процессорном пуле. Значение указывает процент текущей выделенной мощности раздела для добавления или удаления.                                                                                                                                                                                                                                                                                                                            |
| Minimum entitlement per VP (Минимум мощности на ВП)  | 0.1  | 0.5          | 1.0   | Минимальное количество выделенной мощности на виртуальный процессор. Использование этого атрибута защищает раздел от снижения производительности по причине того, что у него есть слишком много виртуальных процессоров относительно выделенной мощности. Когда выделенная мощность удаляется из раздела, виртуальные процессоры также будут удалены, если выделенная мощность на каждый виртуальный процессор становится меньше, чем это число. Значение по умолчанию – 0.5. Минимальное значение – 0.1. Максимальное значение – 1.0.   |
| Maximum entitlement per VP (Максимум мощности на ВП) | 0.1  | 0.8          | 1.0   | Максимальное количество выделенной мощности на виртуальный процессор. Этот атрибут контролирует количество доступной мощности, которая может использоваться uncapped-разделом, работающим в общем пуле. Когда выделенная мощность добавляется в раздел, виртуальные процессоры будут добавлены, если количество выделенной мощности на каждый виртуальный процессор становится выше этого числа. Увеличение количества виртуальных процессоров в uncapped-разделе позволяет разделу использовать больше доступной процессорной мощности. |

## Настройки памяти

В таблице 7-4 показаны настройки, связанные с памятью. Все эти настройки подходят для политик разделов и групп, но только часть используется для общих определений.

**Таблица 7-4.** Настройки, связанные с памятью

| Настройка                                                        | Мин. | По умолчанию | Макс.   | Описание                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------------------------------------|------|--------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Memory notify intervals (Количество интервалов для оповещения)   | 1    | 6            | 100     | Количество непрерывных 10-секундных периодов опроса, в течение которых относящееся к памяти значение должно превышать (быть ниже) пороговое значение для того, чтобы PLM инициировал действие.                                                                                                                                                                                                                 |
| Memory utilization low (Нижнее значение загруженности памяти)    | 1    | 50           | 90      | Если загрузка памяти падает ниже этого порогового значения, то считается, что раздел имеет избыток памяти, и он становится донором. Значение – в процентах. Минимальный разрыв между memory_util_low и memory_util_high – 10 процентов.                                                                                                                                                                        |
| Memory utilization high (Верхнее значение загруженности памяти)  | 1    | 90           | 100     | Верхнее пороговое значение загруженности памяти, при достижении которого раздел считается нуждающимся в большем количестве памяти. Значение – в процентах. Минимальный разрыв между memory_util_low и memory_util_high – 10 процентов.                                                                                                                                                                         |
| Memory page steal high (Верхнее значение изъятия страниц памяти) | 0    | 0            | 231 – 1 | Пороговое значение скорости изъятия страниц (page steal rate), при достижении которого раздел считается нуждающимся в большем количестве памяти. Значение – количество изъятых страниц в секунду. При определении того, нуждается ли раздел в дополнительной памяти, результат проверки этого порогового значения объединяется логическим «И» с результатом верхнего порогового значения загруженности памяти. |
| Memory free unused (Освобождение неиспользуемой памяти)          | -    | Нет          | -       | Показывает, будут или нет избыточные ресурсы памяти удалены из раздела и помещены в резервный пул памяти. Значение «нет» показывает, что лишние ресурсы памяти останутся в разделе, пока они не потребуются другому разделу. Значение «да» показывает, что лишние ресурсы памяти удаляются из раздела, когда он больше не использует их.                                                                       |
| Memory delta (Дельта памяти)                                     | 1    | 1 LMB        | 256     | Количество мегабайт памяти, удаляемой или добавляемой в раздел за одну DR-операцию. Если это значение меньше, чем размер logical memory block (LMB) в системе, то значение округляется до размера LMB. Если это значение больше, чем размер LMB в системе, но не кратно ему, то значение округляется в нижнюю сторону до ближайшего числа, кратного размеру LMB.                                               |

## 7.2.6. Базовая настройка PLM

В этой главе рассказывается о шагах, которые необходимо выполнить для начальной настройки PLM, используя мастер настройки PLM в Web-based System Manager. Общая процедура показана на рис. 7-5.

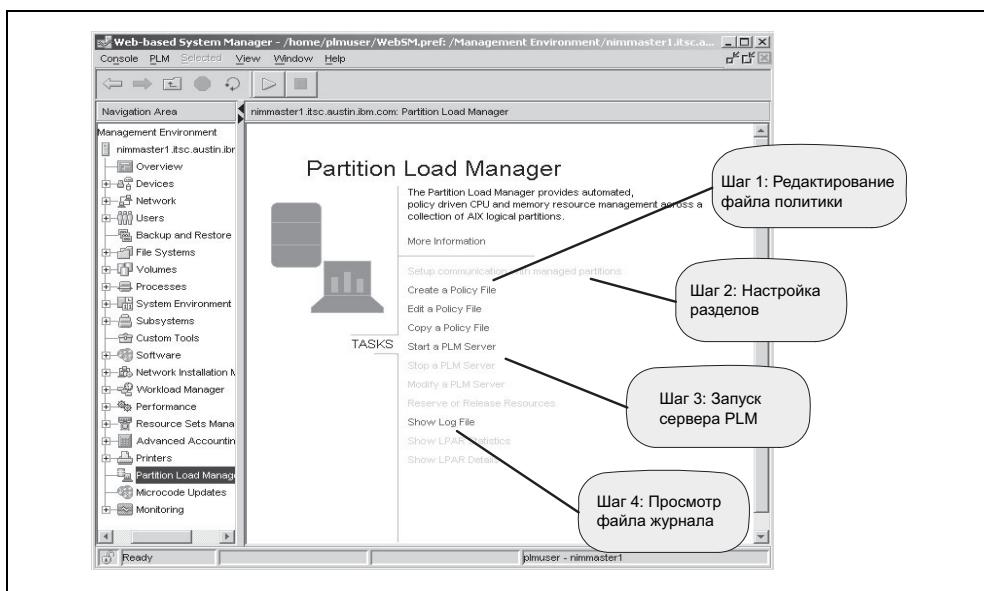


Рис. 7-5. Шаги, необходимые для настройки PLM

### Управление выделенной процессорной мощностью

В первом примере мы настроим PLM на управление выделенной мощностью (CE) двух ограниченных (capped) разделов, работающих в общем процессорном пуле: plmserver и vio\_client2. Эти два раздела имеют определения на НМС, показанные в таблице 7-5.

Таблица 7-5. Начальная конфигурация разделов в пуле

| Ресурс                 | Мин. | Желательно | Макс. |
|------------------------|------|------------|-------|
| Виртуальные процессоры | 1    | 1          | 30    |
| Мощность               | 0.1  | 3          | 5     |
| Память (МБ)            | 256  | 512        | 1024  |

1 . Запустите Web-based System Manager.

Используйте одну из следующих процедур.

- Войдите в сервер PLM как root и выполните команду `wsm`. Если Вы используете команду `wsm`, Вы должны присвоить переменной DISPLAY имя или IP-адрес Вашего терминала X11:  
\$ export DISPLAY=my\_xterm:0

где my\_xterm – имя Вашего терминала X11. Вы должны поставить :0 в конце; это указывает на номер окна X11.

– Используйте основанный на Java клиент Windows® Web-based System Manager (Доступен для загрузки с HMC по адресу

[http://my\\_hmc/remote\\_client.html](http://my_hmc/remote_client.html), где my\_hmc – имя или IP-адрес Вашей HMC).

2. Запустите мастера настройки PLM.

Сделайте двойной щелчок на значке Partition Load Manager или в списке в зоне навигации слева или в основном окне. Откроется окно PLM, как показано на рис. 7-6.

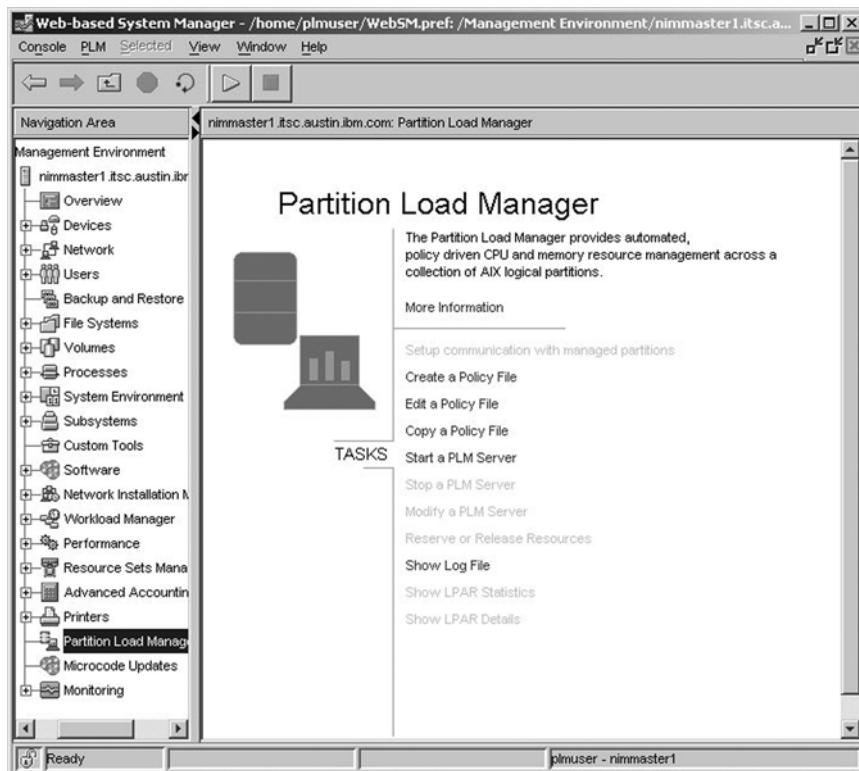


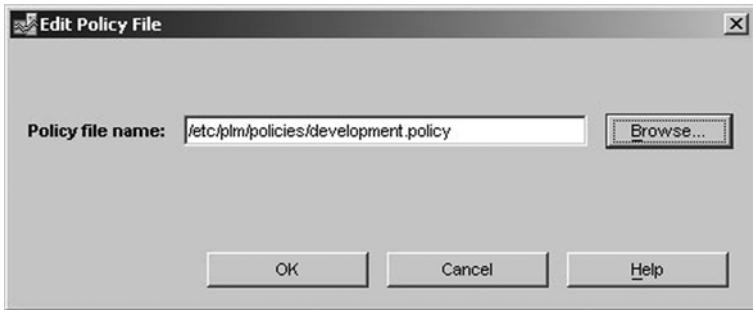
Рис. 7-6. Стартовое окно Partition Load Manager

3. Создайте файл политики, используя мастер.

Щелкните на строку `Create policy file` – откроется окно, показанное на рис. 7-7<sup>1</sup>.

Расположение файла политик по умолчанию – каталог `/etc/plm/policies`. Вы можете создать файл политик в любом каталоге, на которые у Вас есть право записи; имя файла, указываемое в окне, должно быть полным. В этом при-

<sup>1</sup> На рисунке 7-7 показано другое окно – редактирование файла политики. Прим. науч. ред.



**Рис. 7-7.** Закладка General окна Create Policy File мастера PLM

мере, файл политики называется development. Секция комментариев опциональная; в нашем примере мы показываем, к каким разделам относится политика.

4. Заполните информацию в закладке Globals, как показано на рис. 7-8. Поле HMC name – это имя или IP-адрес Вашей НМС. Проверьте еще раз, что имя, которое Вы выбрали, может использоваться в команде ssh без запроса на обмен ключами или ввод пароля.

Поле HMC user name – это имя пользователя, которое использовалось при настройке соединения ssh, обычно это hscroot.

Поле SEC name – это имя, которое Вы получили, когда тестировали соединение ssh между сервером PLM и НМС. Используйте следующую команду с сервера PLM:

```
ssh hscroot@p5hmc1 lssyscfg -r sys -F name
Server590
```

Поле HMC command wait используется для того, чтобы сообщить PLM, сколько времени ждать, в минутах, завершения команды НМС до тайм-аута. Значение по умолчанию – 5 минут. Минимальное значение – 1, максимальное – 60.

5. Определите группу разделов. Все разделы принадлежат группе разделов. Все разделы в любой группе одного типа – выделенные или работающие в общем пуле.

Щелкните на закладку Group, потом – на кнопку Add. Откроется окно Group Definition, как показано на рис. 7-9. В этом примере группа была названа development, максимальное количество физических ЦП для группы было установлено равным 4, что означает, что сумма процессорных мощностей всех разделов в группе не будет превышать четыре ЦП. Мы отключили управление ресурсами памяти, сняв пометку с соответствующего поля.

6. В нашем примере всего два раздела, и мы собираемся установить одинаковые настройки для каждого раздела, так что мы будем использовать настройки группы. Щелкните на закладку Tunables и установите значение политики по умолчанию для всех разделов в группе в открывшемся окне, показанном на рис. 7-10.

Заполните значения параметров настройки. Если Вы оставляете поле пустым, оно принимает значение по умолчанию. В этом примере нет значений

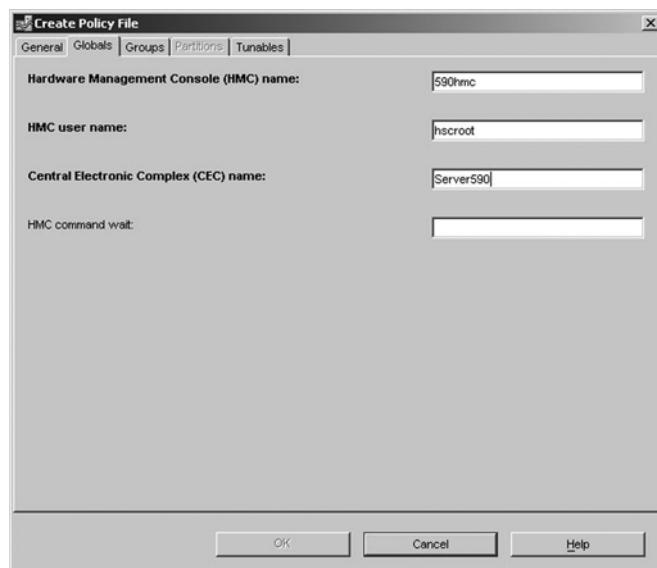


Рис. 7-8. Закладка Globals окна Create Policy File мастера PLM

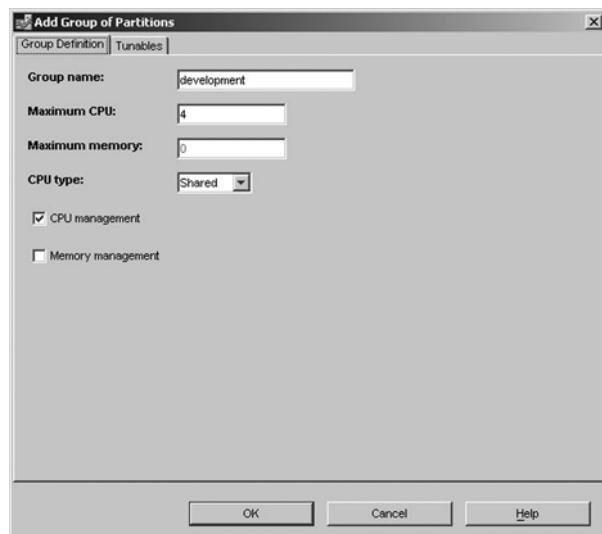
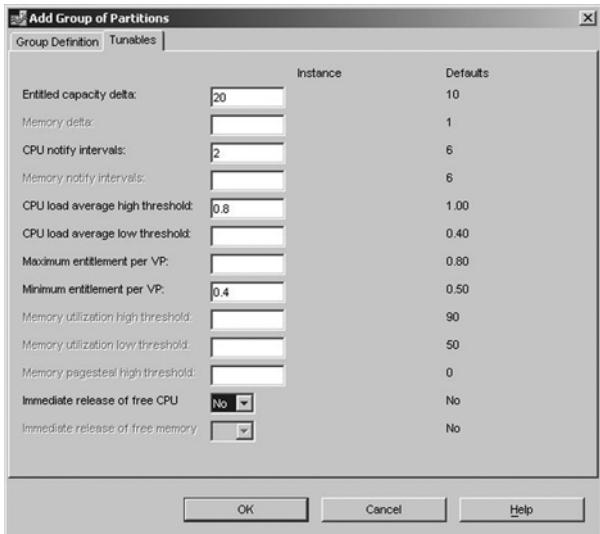


Рис. 7-9. Окно Group Definitions мастера PLM

по умолчанию Instance, так что используются все значения по умолчанию PLM, показанные в правой стороне окна.

В этом примере мы изменили значение параметра entitled capacity delta (дельта мощности) с 10 процентов на 20 процентов, так что PLM будет при



**Рис. 7-10.** Закладка tunables окна Add Group of Partitions мастера PLM

выполнении DR-операций добавлять или убирать 20 процентов текущих ресурсов. Мы уменьшили количество notify intervals (интервалов оповещения) до двух; это означает, что после двух оповещений о нехватке ЦП PLM будет пытаться найти ресурсы для раздела. Мы уменьшили параметр CPU load maximum (максимальная загрузка ЦП) до 80 процентов, это означает, что когда средняя загрузка процессора достигнет 80 процентов, агент пошлет уведомление серверу PLM. Мы также уменьшили значение параметра minimum entitlement per virtual processor (минимальная мощность на виртуальный процессор) до 0.4.

Щелкните кнопку OK, когда закончите, и Вы увидите итоговое определение группы (см. рис. 7-11).

7. Определив группу, мы должны теперь указать, какие разделы в нее входят. Для выполнения этой задачи щелкните на закладку **Partitions** в окне Create Policy File, после этого щелкните на кнопку **Add** во всплывающем окне, как показано на рис. 7-12.

В окне есть выпадающее меню для выбора имени группы, полезное в том случае, если Вы создали несколько групп.

В поле Partition name введите сетевое имя хоста. Оно должно быть тем же, что было указано при настройке RMC. Если Вы использовали полностью квалифицированные имена, здесь Вы также должны использовать их.

**Внимание.** Поле Partition name в интерфейсе – это *не* имя раздела, определенное на HMC, а сетевое имя.

8. Щелкните на закладку **Resource Entitlements** в окне Add Managed Partition для указания политики ресурсов ЦП для этого раздела. Появится окно, показанное на рис. 7-13. Значения по умолчанию для минимальной, желаемой и мак-

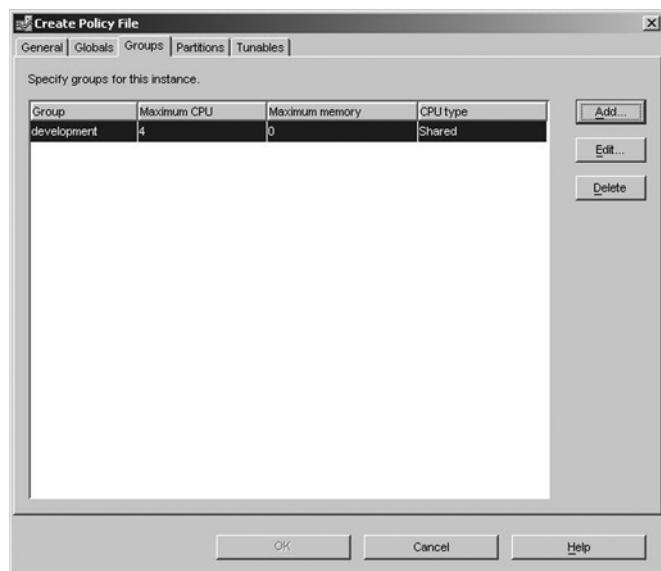


Рис. 7-11. Окно wizard group definition summary мастера PLM

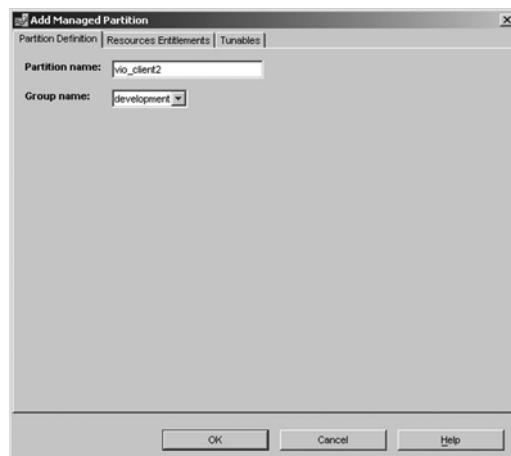
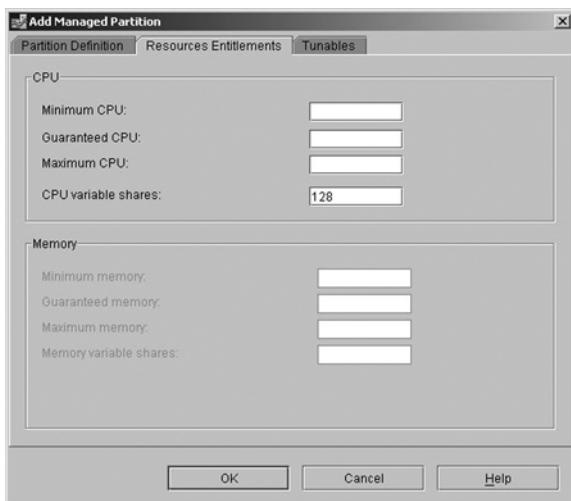


Рис. 7-12. Окно add managed partition мастера PLM

симальной мощности взяты равными значениям минимальной, желаемой и максимальной мощности в определении раздела на НМС. Значение по умолчанию параметра CPU variable shares равно единице.

В этом примере в группе есть два раздела. Мы установим разные значения shares так, что один раздел будет иметь в два раза больше «общих ресурсов», чем другой. Одному разделу мы установим 128 shares, а другому – 64. Мы уви-

дим, как PLM использует эти веса, когда он будет распределять ресурсы между разделами.



**Рис. 7-13.** Окно partition resource entitlement мастера PLM

**Замечание.** Значения параметров настройки PLM, использованные в предыдущих примерах, выбраны только для целей демонстрации. Они не подходят для большинства рабочих систем.

9. Повторите действия начиная с шага 7 для остальных разделов в группе. Когда Вы закончите, закладка **Partitions** окна Create Policy File должна быть похожей на показанную на рис. 7-14.

Вы увидите, что список параметров управления ресурсами памяти заполнен, даже если для этой группы не настраивалось управление памятью.

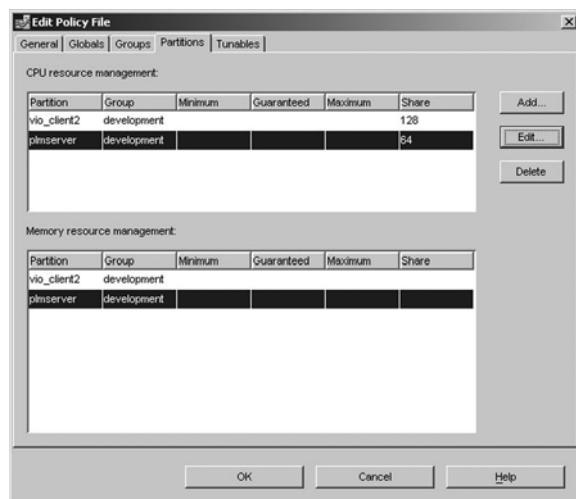
Щелкните кнопку **OK** для создания файла политики. Файл политики находится в том каталоге, который Вы указали в шаге 3. Каталог по умолчанию – `/etc/plm/policies`. Если Вы просмотрите этот файл, его содержимое будет похоже на представленное в примере 7-1.

#### **Пример 7-1.** Файл политики PLM для управления ресурсами ЦП

```
cat /etc/plm/policies/development
#Policy for the development and admin partitions

globals:
 hmc_host_name = 590hmc
 hmc_user_name = hscroot
 hmc_cec_name = Server590

development:
 type = group
 cpu_type = shared
 cpu_maximum = 4
```



**Рис. 7-14.** Окно мастера PLM с заполненной информацией о разделах

```

mem_maximum = 0

vio_client2.mydomain.com:
 type = partition
 group = development
 cpu_shares = 128
 ec_delta = 20
 cpu_intervals = 2
 cpu_load_high = 0.8
 cpu_load_low = 0.3
 cpu_free_unused = yes

plmserver.mydomain.com:
 type = partition
 group = development
 cpu_shares = 64
 ec_delta = 20
 cpu_intervals = 2
 cpu_load_high = 0.8
 cpu_load_low = 0.3
 cpu_free_unused = yes

```

10. Как только файл политики будет определен для управляемых разделов, мы можем установить коммуникации RMC между разделами, если это не было сделано вручную при помощи команды `plmsetup` ранее. Щелкните на `Setup communication with managed partitions`, в результате откроется окно, показанное на рис. 7-15.

В поле `Authenticated user name` выберите `root` из выпадающего меню и введите или выберите при помощи кнопки `Browse` имя файла политики для управляемых разделов. Щелкните `OK`.

Это действие может быть выполнено вручную, как описано в 7.2.3 «Настройка RMC для PLM».

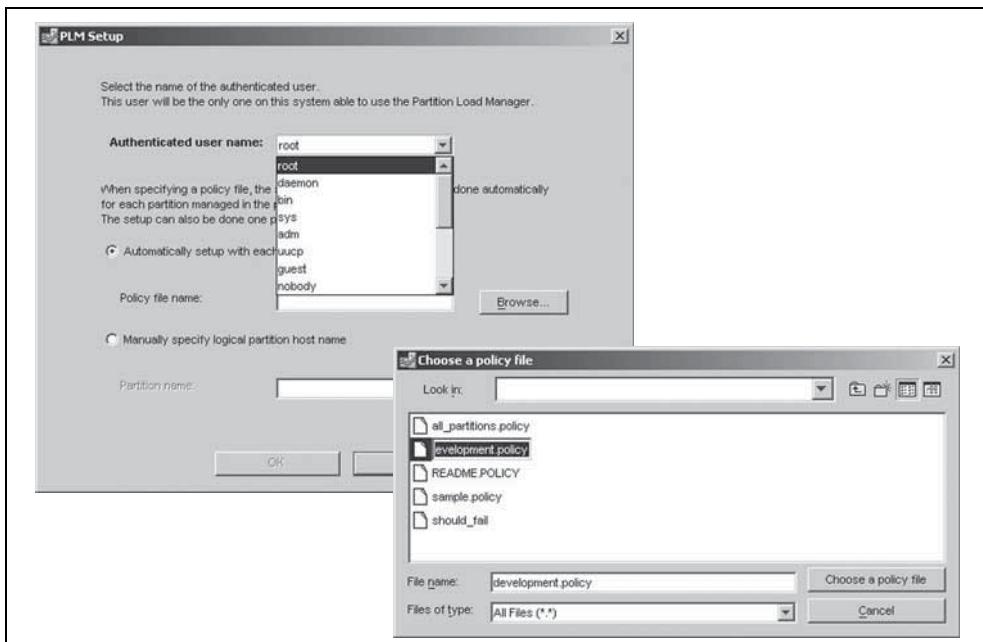


Рис. 7-15. Окно настройки коммуникаций PLM с управляемыми разделами

11. Мы можем теперь запустить PLM-сервер с нашим файлом политики. Просто просмотрите конфигурацию управляемых разделов, используя команду `mpstat`.

PLM может быть запущен из командной строки или используя интерфейс Web-based System Manager. В этом примере мы называем конфигурацию RedbookServer.

- a. Выполните в командной строке:

```
cd /etc/plm/policies
xlplm -S -p ./development.policy -l /var/opt/plm/plm.log RedbookServer
```

По умолчанию система работает в режиме управления, так что нам не надо указывать это в командной строке. Мы можем проверить, что экземпляр сервера выполняется:

```
#xlplm -Q
```

RedbookServer

- b. Из главного окна PLM интерфейса Web-based System Manager, как показано на рис. 7-16.

Это выполнит то же самое действие, как и интерфейс командной строки: создаст сервер PLM с именем RedbookServer в режиме управления, используя ту же самую политику и файлы системного журнала, используемые в командной строке. Вы можете использовать конфигурацию с определенным Вами именем или использовать имя по умолчанию.



**Рис. 7-16.** Запуск сервера PLM

Файл политики предусматривает, что PLM не будет начинать принимать меры до того, как загрузка ЦП не достигнет 0.8. Запустите набор задач, интенсивно использующих процессор, в каждом разделе, и наблюдайте за действиями, предпринимаемые PLM.

Вы можете наблюдать за действиями PLM, используя команду `tail` для просмотра файла журнала:

```
tail /var/opt/plm/plm.log
```

Вы можете проверить новую конфигурацию раздела, используя команду `mpstat`.

12. Из главного окна PLM интерфейса Web-based System Manager Вы можете увидеть статус и статистику PLM, а также модифицировать сервер PLM. Используя следующую команду, Вы можете определить имена выполняющихся экземпляров:

```
xlplm -Q
RedbookServer
```

13. Остановите PLM из главного окна PLM интерфейса Web-based System Manager или используя следующую команду:

```
xlplm -K RedbookServer
```

## Управление памятью

Мы сделали управление процессорами для разделов, работающих в общем пуле; перейдем к управлению памятью. До того как продолжить, Вы должны остановить экземпляр сервера PLM.

В этом втором примере мы настроим PLM на управление сконфигурированной памятью двух разделов с выделенными процессорами: `app_server` и `db_server`.

Поскольку PLM не может управлять различными типами процессора в пределах одной группы, мы должны создать новую группу. Эта новая группа или может быть включена в существующую политику PLM, или мы можем создать новый экземпляр сервера PLM для управления этими разделами.

В этом примере мы выберем первый вариант, чтобы показать управление двумя группами из одного сервера PLM.

Два раздела с выделенными процессорами настроены на НМС идентично. Это показано в таблице 7-6.

**Таблица 7-6.** Начальная конфигурация разделов с выделенными разделами<sup>1</sup>

| Ресурс      | Мин. | Желательно | Макс. |
|-------------|------|------------|-------|
| ЦП          | 1    | 1          | 4     |
| Память (МБ) | 512  | 512        | 3076  |

Шаги настройки:

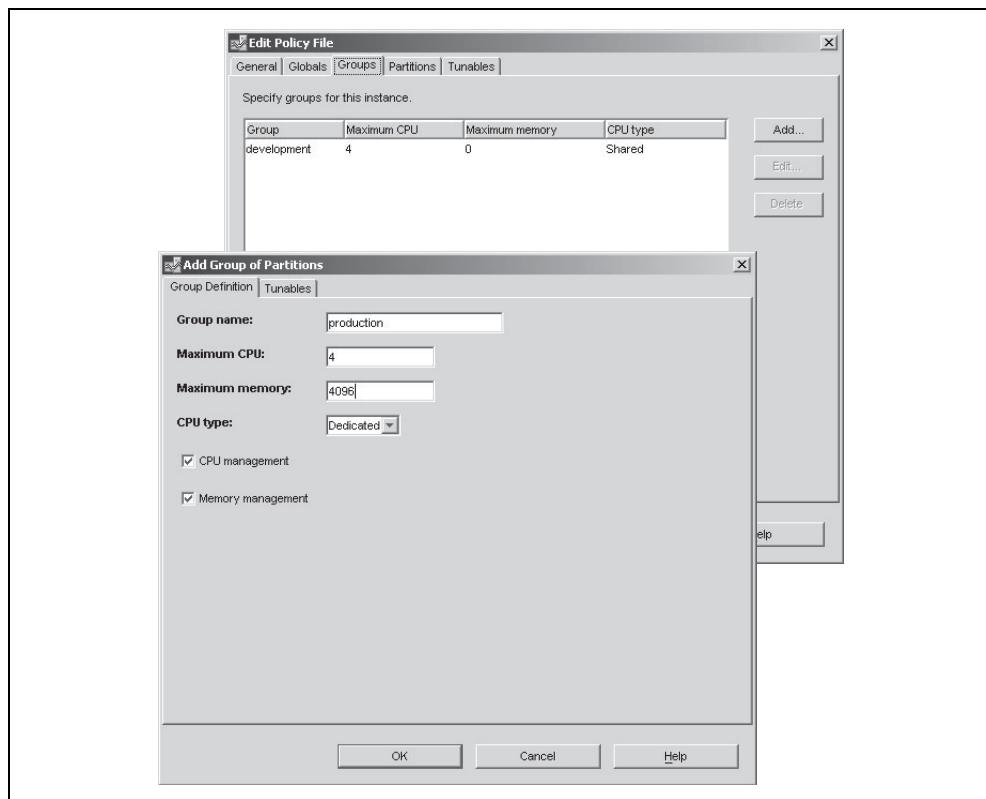
1. Отредактируйте существующий файл политики PLM. В основном окне PLM выберите **Edit a Policy File**. Откроется окно, показанное на рис. 7-17. Введите имя созданного ранее файла политики или сделайте это при помощи кнопки **Browse**. Щелкните **OK**, в результате откроется окно **Edit Policy File**.



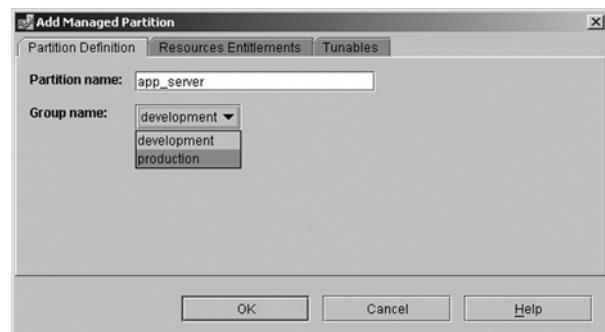
**Рис. 7-17.** Мастер PLM: окно Edit Policy File

2. Щелкните на закладку **Groups**, а затем на кнопку **Add**. Откроется окно, показанное на рис. 7-18 (то же, что использовалось в шаге 5 на 355 в первом сценарии). В этом примере мы создадим группу, которая будет называться **production**. Щелкните **OK**.
3. Определите, какие разделы принадлежат новой группе. Щелкните на закладку **Partitions**, потом на кнопку **Add**. Введите имя управляемого раздела в первой строке и затем щелкните на выпадающее меню **Group name** и выберите группу, созданную в предыдущем шаге, как показано на рис. 7-19.
4. Настройте «общие ресурсы» (shares) для ЦП и памяти. В этой группе оба раздела будут одинаковыми, с одинаковыми весами (128 и для процессора, и для памяти), как показано на рис. 7-20 и рис. 7-21. Щелкните **OK**.
5. Так как политики для обоих разделов идентичны, мы определим политику группы, а не индивидуальные политики групп. Щелкните на закладку **Groups**, в результате мы увидим итоговую информацию о всех определенных группах.

<sup>1</sup> Обратите внимание, что в оригинале руководства ошибочно написано «Shared processor partition initial configuration», т. е. «Начальная конфигурация разделов с общими процессорами». Прим. науч. ред.



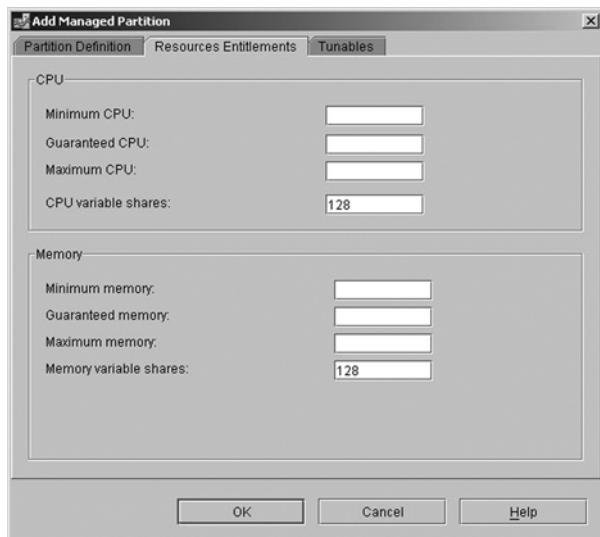
**Рис. 7-18.** Диалог PLM для добавления группы разделов в существующий файл политики



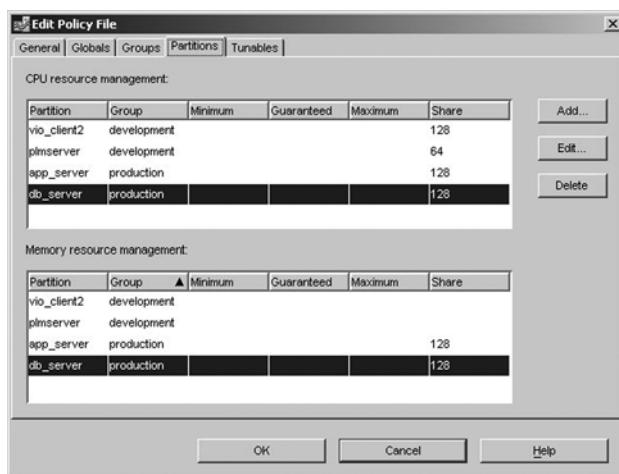
**Рис. 7-19.** Диалог Add Managed Partition мастера PLM

Выберите группу production и щелкните на кнопку Edit; в появившемся окне щелкните на закладку Tunables, как показано на рис. 7-22.

В этом примере параметр memory delta установлен в 256 МБ, так что PLM будет пытаться добавлять (и убирать) память разделу блоками по 256 МБ. Пара-

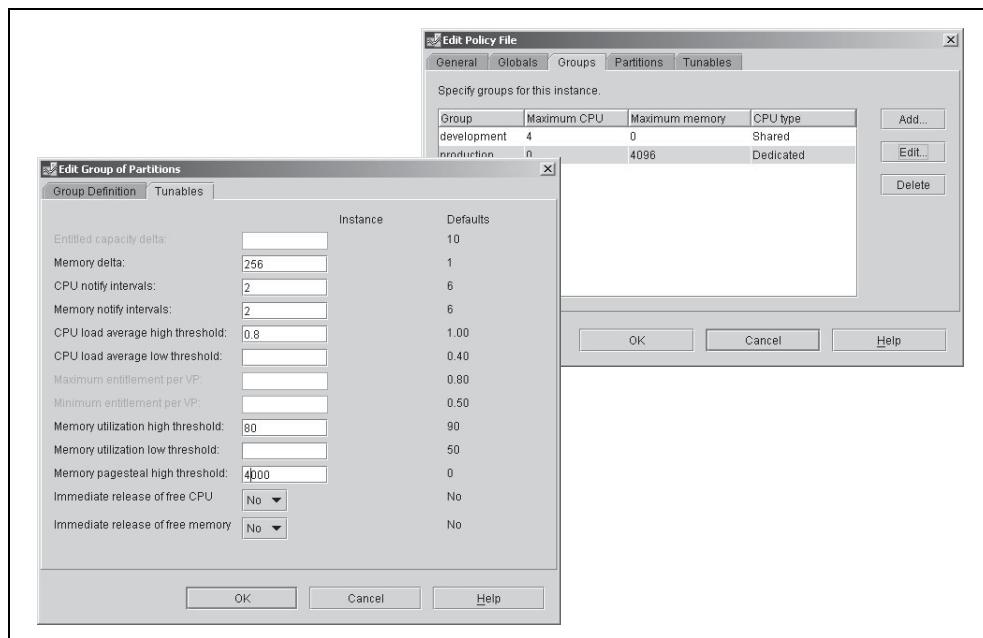


**Рис. 7-20.** Диалог Resource Entitlements мастера PLM.



**Рис. 7-21.** Итоговая информация о разделах в окне Edit Policy File

метр notify count уменьшен со значения по умолчанию «шесть» до двух – как для ЦП, так и для памяти. Параметр CPU load average high threshold (верхнее пороговое значение средней загрузки процессора) уменьшен до 0.8, но нижнее пороговое значение оставлено по умолчанию – 0.4. Пороговое значение виртуальных процессоров недоступно, т.к. эта группа состоит из разделов с выделенными процессорами. Верхнее пороговое значение загрузки памяти (memory high utilization threshold) уменьшено до 0.8, и верхнее пороговое



**Рис. 7-22.** Мастер PLM: установка настроек группы

значение скорости изъятия страниц установлено в 4000 изъятий (page-steals) в секунду.

Когда Вы установите Вашу политику, щелкните **OK**.

6. Добавьте второй раздел в группу, используя ту же самую процедуру, начиная с шага 3.
7. После определения групп разделов мы можем перезапустить сервер PLM, как объяснено в шаге 11 (в первом примере).

### 7.2.7. Интерфейс командной строки Partition Load Manager

Существует две команды PLM для контроля его работы через скрипты:

**xlp1m** Контролирует и проверяет PLM.

**xlpstat** Показывает статистику загрузки логических разделов.

Команда **xlpstat** обсуждается в главе 6.5.6 «Мониторинг при помощи PLM».

Существует шесть форматов команды **xlp1m**, в зависимости от выполняемого действия.

Возможные операции:

- ▶ Запуск сервера PLM (-S)
- ▶ Останов сервера PLM (-K)
- ▶ Модификация сервера PLM (-M)
- ▶ Резервирование ресурсов в группу PLM или освобождение их из группы (-R)

- ▶ Опрос статуса PLM (-Q или -T)
- ▶ Проверка синтаксиса и структуры файла политики PLM (-C)

### **Запуск сервера PLM**

Синтаксис команды `xlplm` для запуска сервера PLM:

```
xlplm -S -p policy_file -l log_file [-o operational_mode] [plm_instance]
```

Параметры `policy_file` и `log_file` обязательные. Файл политики указывает используемую политику; это может быть любой корректный файл политики, созданный либо вручную, либо при помощи мастера настройки PLM. Файл должен удовлетворять стандарту абзацев (*stanza*) файла политик PLM. Этот формат детально описан в файле README POLICY в каталоге /etc/plm/policies и не представлен в этом руководстве.

При использовании ключа `-C`, `xlplm` может проверить синтаксис файла. Синтаксис команды:

```
xlplm -C -p policy_file
```

Файл журнала содержит журнал активности PLM, это может быть любой файл, в который Вы имеете право записи. Мастер PLM предлагает каталог /var/opt/plm/, но он по умолчанию доступен на запись только для пользователя root.

Параметр `operational_mode` – или мониторинг (monitoring), обозначаемый N, или управление (managing), обозначаемый M. Режим работы по умолчанию – управление (M).

Параметр `plm_instance` – это имя, которое будет присвоено экземпляру сервера PLM. Имя по умолчанию – default. Имя должно быть обязательно указано, если на одной системе (в одном разделе) выполняется несколько экземпляров PLM.

### **Останов сервера PLM**

Команда `xlplm -K [ plm_instance ]` используется для остановки сервера PLM. Если имя экземпляра сервера PLM указывалось при старте, то при останове оно также должно быть указано. Для получения списка всех выполняющихся экземпляров сервера PLM используйте команду `xlplm -Q`.

### **Модификация сервера PLM**

Команда `xlplm -M` может быть использована для:

- ▶ Смены активного файла политики.
- ▶ Смены файла журнала.
- ▶ Смены режима работы: мониторинг или управление.

Синтаксис похож на синтаксис команды старта PLM:

```
xlplm -M { -p Policy } { -l Logfile } { -o {M|N} } [Instance]
```

Вы можете изменить более одного аспекта конфигурации в одной команде.

Команду `xlplm -M` можно поместить в файл crontab для периодической смены политики PLM или ротации файлов журналов.

## **Резервирование или освобождение ресурсов для группы PLM**

Резервируя ресурсы для группы разделов (или освобождая их), Вы можете изменить лимиты группы PLM. Синтаксис команды:

```
xlplm -R { -c Amount | -m Amount } [-g Group] [Instance]
```

Вы можете указать только ресурс, управляемый экземпляром сервера PLM, так что, если сервер PLM не управляет памятью, Вы не можете дать запрос на резервирование или освобождение ресурсов памяти.

## **Просмотр и опрос выполняющихся экземпляров сервера PLM**

Существует два формата команды для опроса PLM: -Q и -T. Первый предназначен для опроса экземпляров PLM; второй показывает настройки PLM по умолчанию. Синтаксис двух форматов команды:

```
xlplm -Q [-v] [-r] [-f plm_instance]
```

(Ключ -v показывает текущие значения настроек.) и

```
xlplm -T [-r]
```

В обоих случаях ключ -r делает вывод в формате, разделенном двоеточиями, удобном для выборки части результатов.

В коротком формате команда `xlplm -Q` показывает список всех выполняющихся экземпляров сервера PLM. В длинном формате команда `xlplm -Q -f plm_instance` показывает ключевую информацию, связанную с конкретным экземпляром, как показано в примере 7-2. Значение 0 показывает, что используются значения PLM по умолчанию.

### **Пример 7-2. Опрос статуса экземпляра сервера PLM**

---

```
xlplm -Q -f dedicated
PLM Instance: dedicated
GROUP: group1
 CUR MAX AVAIL RESVD MNGD
CPU: 0.00 4.00 4.00 0.00 Yes
MEM: 0 0 0 0 No

app_server.mydomain.com
 RESOURCES:
 CUR MIN GUAR MAX SHR
CPU: 0.00 0.00 0.00 0.00 50
MEM: 0 0 0 0 1

db_server.mydomain.com
 RESOURCES:
 CUR MIN GUAR MAX SHR
CPU: 0.00 0.00 0.00 0.00 200
MEM: 0 0 0 0 1
```

---

С ключом `-r` (raw) будут показаны те же данные, как показано в примере 7-3.

---

**Пример 7-3.** Опрос статуса экземпляра сервера PLM

---

```
xlplm -Q -rf dedicated
#globals:hmc_host_name:hmc_user_name:hmc_cec_name:policy_file:log_file:mode
globals:192.168.255.69:hscroot:Server590:/etc/plm/policies/dedicated:/etc/plm/p
olicies/dedicated.log:manage
#group:group_name:cpu_type:cpu_maximum:cpu_free:cpu_reserved:mem_maximum:mem_fr
ee:mem_reserved
group:group1:dedicated:4.00:4.00:0.00:0:0:0
#partition:status:host_name:group_name:cpu_minimum:cpu_guaranteed:cpu_maximum:c
pu_shares:cpu_current:mem_minumum:mem_guaranteed:mem_
maximum:mem_shares:mem_current:cpu_intervals:cpu_free_unused:cpu_load_high:cpu_
load_low:ec_per_vp_max:ec_per_vp_min:ec_delta:mem_int
ervals:mem_free_unused:mem_util_high:mem_low:mem_pgstl_high:mem_delta
partition:up:app_server.mydomain.com:group1:-1.00:-1.00:-1.00:50:0.00:-1:-1:-1:
1:0:2:0:1.00:0.40:0.80:0.50:10:6:0:90:50:0:1
partition:up:db_server.mydomain.com:group1:-1.00:-1.00:-1.00:200:0.00:-1:-1:-1:
1:0:2:0:1.00:0.40:0.80:0.50:10:6:0:90:50:0:1s
```

---

**Проверка синтаксиса файла политики PLM**

Команда `xlplm -C -p policy_file` проверяет синтаксис файла политики. Она не проверяет, что политика удовлетворяет определению раздела на HMC.

**Измерение загрузки PLM**

Чтобы узнать загрузку памяти и процессоров с точки зрения PLM, используйте команду `lsrsrc`, как показано в примере 7-4.

---

**Пример 7-4.** Оценка загрузки ресурсов PLM

---

```
$ lsrsrc -Ad IBM.LPAR
Resource Dynamic Attributes for IBM.LPAR
resource 1:
 ConfigChanged = 0
 CPULoad = 0.0878906
 CPUUtil = 0.39
 MemLoad = 92.5613
 MemPgSteal = 149320
 CPUZone = 2
 MemZone = 2
```

---

## 7.3. Реконфигурация по расписанию

Возможна реконфигурация разделов и политик НМС по расписанию (и отложенная по времени). В этой главе детально объясняются такие операции.

### 7.3.1. Реконфигурация раздела

НМС предоставляет механизм выполнять однократную отложенную или периодическую реконфигурацию управляемых разделов; управление находится в окне Scheduled Operations окна HMC Configuration, как показано на рис. 7-23.

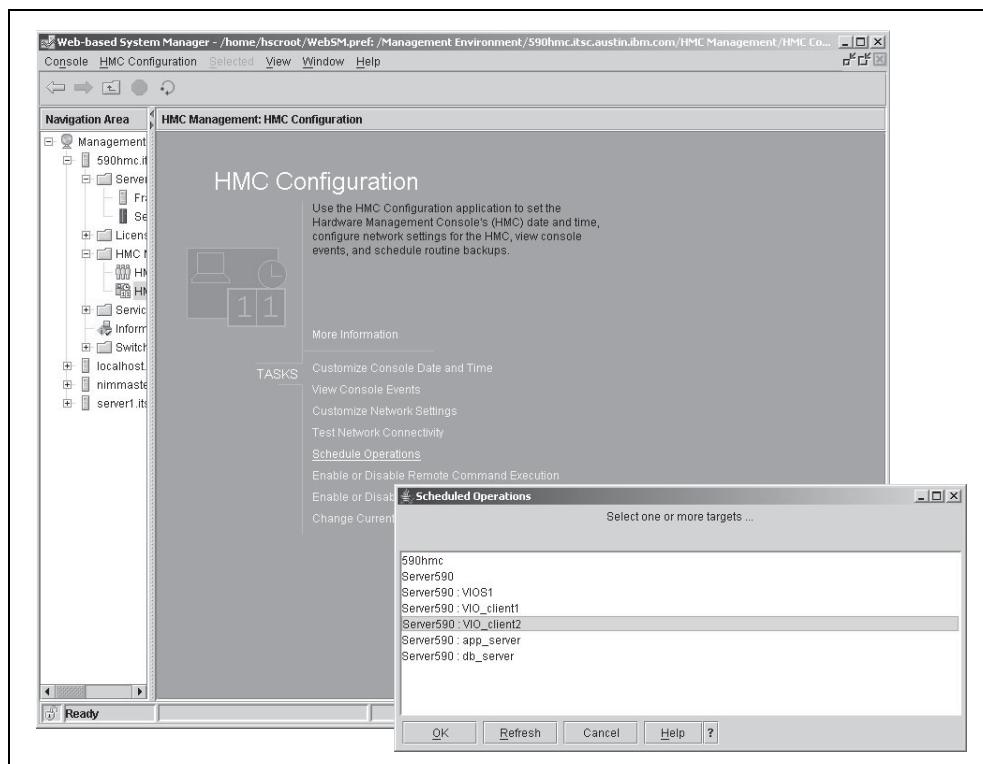


Рис. 7-23. Окна HMC Configuration и Schedule Operations

Выберите раздел, для которого Вы хотите зарограммировать динамическую реконфигурацию (в этом примере – VIO\_client2), и щелкните OK; в результате откроется окно Customize Scheduled Operations, как показано на рис. 7-24, из которого открывается окно Add a Scheduled Operation window, как показано на рис. 7-25.

В окне Add a Scheduled Operation представлены три различные операции НМС, которые можно выполнить над разделом:

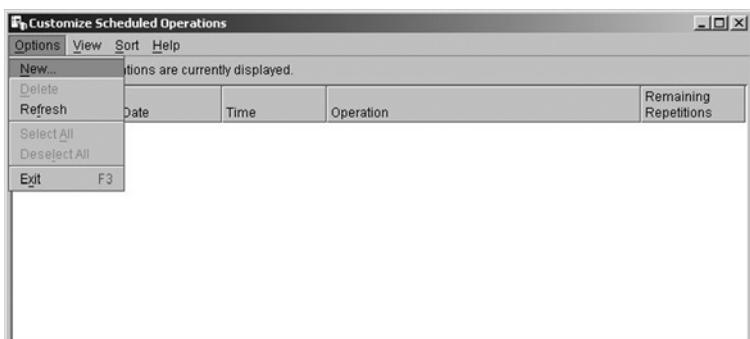


Рис. 7-24. Окно Customize Scheduled Operations

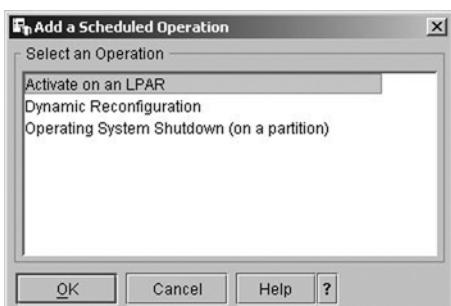


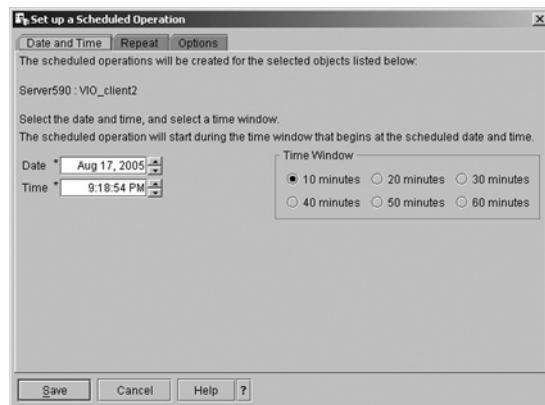
Рис. 7-25. Окно dd a Scheduled Operation

- ▶ Активизация раздела.
- ▶ Реконфигурация системных ресурсов
- ▶ Завершение раздела, используя завершение работы ОС (Operating System Shutdown).

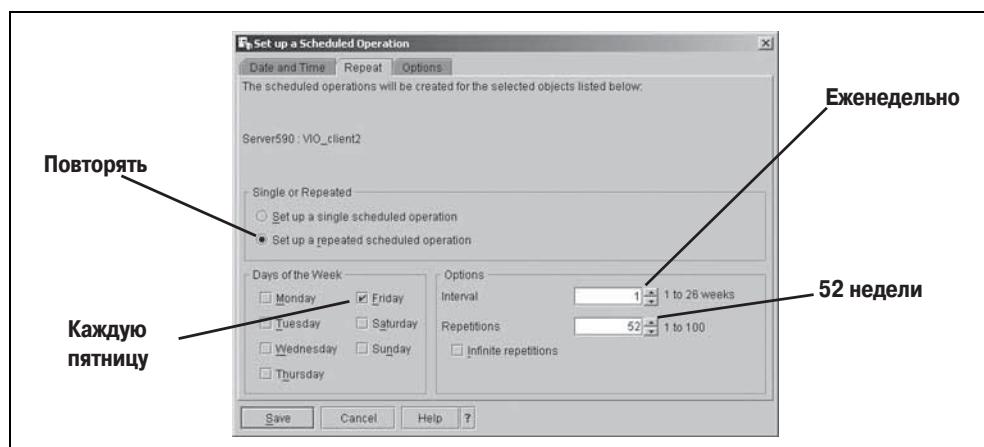
Выберите тип операции, которую Вы желаете выполнить, и щелкните OK. Окно Setup Scheduled Operation, появляющееся после этого, имеет схожий формат для всех операций. Оно имеет две или три закладки, называющиеся Date and Time, Repeat и Options (для операции Operating System Shutdown закладки Options нет). Окно Dynamic Reconfiguration показано на рис. 7-26. Поля дата и время обязательны для заполнения. Вы можете указать временное окно (time window), во время которого должна начаться операция. Минимальное окно времени – десять минут.

Если Вы желаете периодически изменять конфигурацию, щелкните на закладку Repeat, показанную на рис. 7-27. Частота повторов может быть ежедневной или еженедельной. В этом примере конфигурация будет меняться каждую пятницу в течение года.

Закладка Options, показанная на рис. 7-28, используется для указания того, какие ресурсы должны быть добавлены в раздел или удалены из него в запрограммированное время. В этом примере 0.4 процессорные единицы будут перемещаться



**Рис. 7-26.** Закладка Date and Time окна Setup a Scheduled Operation



**Рис. 7-27.** Закладка Repeat окна Set up a Scheduled Operation

из раздела VIO\_client2 в раздел VIO\_client1. Щелкните на кнопку Save, в результате задача добавится к списку запрограммированных операций.

**Важно.** Если Вы планируете периодическую динамическую реконфигурацию LPAR, как в этом примере, то Вы должны запрограммировать другие операции реконфигурации для перемещения такого же количества ресурсов в раздел или из раздела; в противном случае, возможно, у раздела будет полное истощение ресурсов.

Операции для настройки старта или завершения работы раздела схожи с произведенными для реконфигурации.

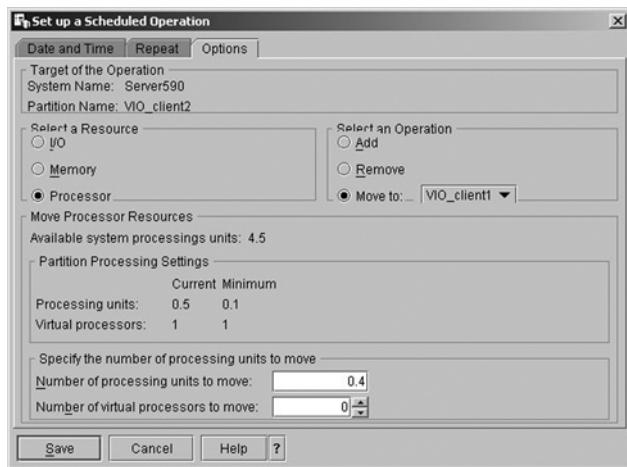


Рис. 7-28. Закладка Options окна Set up a Scheduled Operation

### 7.3.2. Реконфигурация политики PLM

Для периодической смены файла политики PLM, Вы должны использовать пла-нировщик в интерфейсе командной строки PLM, т.е. команды AIX 5L cron или at. Используйте команду `xlplm -M`, как показано в главе «Модификация сервера PLM». Обратитесь к документации по AIX 5L за информацией о командах cron и at.

## 7.4. Советы и поиск неисправностей PLM

В этой главе представлены дополнительные советы и информация о поиске не-исправностей при настройке Partition Load Manager.

### 7.4.1. Поиск неисправностей соединения SSH

Для проверки настройки SSH запустите команду `ssh`. Убедитесь в том, что HMC не спрашивает пароль; в противном случае проверьте настройку, описанную ранее, снова.

```
ssh hscroot@192.168.255.69 date
Fri Aug 19 01:29:19 CDT 2005
```

Если Ваше соединение SSH от сервера PLM к HMC не работает, убедитесь в том, что вы отметили пункт `Enable remote command execution` в окне конфигурации HMC. Проверьте также, позволяет ли функция HMC firewall входящий трафик SSH. В Web-based System Manager выберите `HMC Configuration` в меню `HMC Management` и выберите `Customize Network Settings`. Выберите закладку `LAN Adapter` в окне `Customized Network Settings`, как показано на рис. 7-29.

Выберите настроенный сетевой адаптер, подключенный к внешней сети, кото-рую Вы используете для команды `ssh`, и щелкните кнопку `Details`. Выберите за-кладку `Firewall` в окне `LAN Adapter Details`, показанном на рис. 7-30.

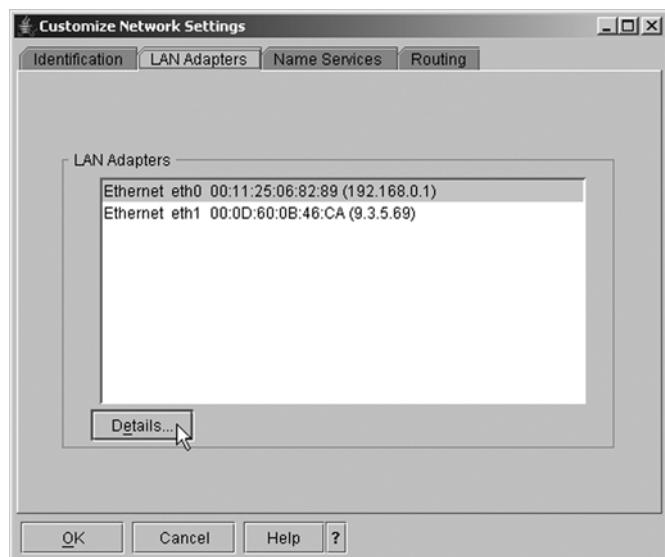


Рис. 7-29. Меню Customize Network Setting: выбор закладки LAN Adapters

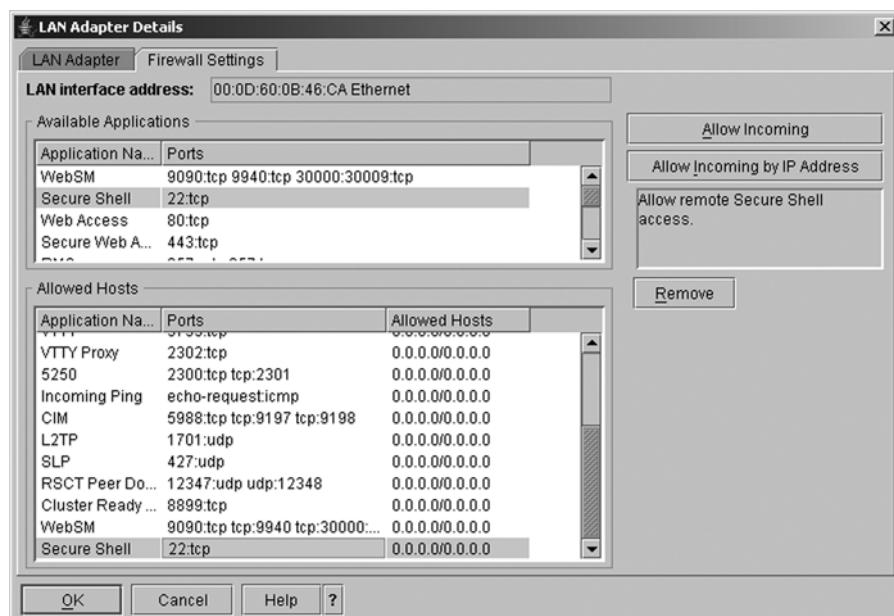


Рис. 7-30. Окно Firewall Settings

Проверьте, что приложение Secure Shell добавлено в секцию Allowed Host во второй половине окна. Если оно не добавлено, выберите Secure Shell из секции Available Applications выше и щелкните кнопку Allow Incoming или Allow Incoming by IP

**Address**, если Вы хотите ограничить доступ конкретным IP-адресам или диапазонам адресов.

Некоторые изменения в конфигурации сети потребуют перезагрузки НМС.

Если соединение ssh работает без запроса пароля, но PLM не может стартовать с выдачей следующего сообщения об ошибке, то Вы пропустили шаг обмена ключами для имени НМС, указанного в Вашем профайле:

```
1498-057 Failed to execute ssh command for hscroot@192.168.255.69. Verify the path,
permissions, and user authorization for this command.
```

The version number 1 for is not valid.

**Замечание.** Обмен ключами – это важный шаг. Если Вы не произвели обмен ключами, то PLM не стартует. Обмен ключами зависит от используемого имени хоста. Если Вы используете короткое имя хоста для НМС в профайле PLM, обменивайтесь ключами с использованием короткого имени хоста. Если Вы используете полностью квалифицированное имя хоста для НМС в профайле PLM, обменивайтесь ключами с использованием полностью квалифицированного имени хоста. Для обмена ключами используйте команду ssh.

Если Вы хотите проверить, для какого имени НМС был проведен обмен ключами, посмотрите файл `./ssh/known_hosts` на сервере PLM:

```
cat ./ssh/known_hosts
590hmc, 192.168.255.69 ssh-rsa
AAAAB3NzaC1yc2EAAABIwAAIAEAzLNs1AT5xqQMqwPXEc9cTMiIae01ytHNvkH7Qf+e8244jFemEdL
QIjY1BoVihuQrgeMgsFiv1Nvzpfqtw4Gxccr8J1E0T/7VKDp+2uJtUB40EEC/9Tt6fYIKam2fSv6YWU
4PtDbAWBeM3aKYZyRLLfShIzYDAk4BP56PVY1yLic=
590hmc.mydomain.com ssh-rsa
AAAAB3NzaC1yc2EAAABIwAAIAEAzLNs1AT5xqQMqwPXEc9cTMiIae01ytHNvkH7Qf+e8244jFemEdL
QIjY1BoVihuQrgeMgsFiv1Nvzpfqtw4Gxccr8J1E0T/7VKDp+2uJtUB40EEC/9Tt6fYIKam2fSv6YWU
4PtDbAWBeM3aKYZyRLLfShIzYDAk4BP56PVY1yLic=
```

#### 7.4.2. Поиск неисправностей соединения RMC

При настройке RMC требует обмена открытыми ключами. Это может быть выполнено или через Web-based System Manager, или через скрипт shell. В обоих случаях управляющая система должна иметь право доступа через rsh ко всем управляемым разделам.

Проверьте или создайте файл `.rhosts`. Файл `.rhosts` необходим только во время настройки. После настройки его можно удалить. Детально шаги конфигурации описаны в главе 7.2.3, «Настройка RMC для PLM».

После настройки коммуникаций RMC проверьте, успешно ли выполняется команда `CT_CONTACT`. Выполните команду на сервере PLM, используя имя хоста одного из управляемых разделов.

Для проверки коммуникаций RMC введите следующую команду на сервере PLM для всех управляемых клиентских разделов:

```
CT_CONTACT=dbsrv lsrsrc IBM.LPAR
Resource Persistent Attributes for IBM.LPAR
```

```

resource 1:
 Name = "DB_Server"
 LPARFlags = 7
 MaxCPU = 10
 MinCPU = 1
 CurrentCPUs = 2
 MinEntCapacity = 0.2
 MaxEntCapacity = 1
 CurEntCapacity = 0.5
 MinEntPerVP = 0.1
 SharedPoolCount = 0
 MaxMemory = 1024
 MinMemory = 128
 CurrentMemory = 512
 CapacityIncrement = 0.01
 LMBSIZE = 16
 VarWeight = 128
 CPUIntvl = 0
 MemIntvl = 0
 CPULoadMax = 0
 CPULoadMin = 0
 MemLoadMax = 0
 MemLoadMin = 0
 MemPgStealMax = 0
 ActivePeerDomain = ""
 NodeNameList = {"dbsrv"}

```

Если вывод команды похож на показанный здесь, то настройка RMC была выполнена успешно.

Если Вы столкнулись с проблемами, похожими на показанные в следующем примере, Вы должны проверить настройку RMC:

```
CT_CONTACT=vio_client2 lsrsrc IBM.LPAR
/usr/sbin/rsct/bin/lsrsrc-api: 2612-024 Could not authenticate user.
```

Вы можете проверить настройку конфигурации RMC командами `ctsvhbal` и `ctsth1`.

Команда `ctsvhbal` показывает идентичность (identity) локальной системы, использующейся в механизме Host Based Authentication (HBA). Аутентификация PLM основывается на первой записи Identity, показанной в выводе команды `ctsvhbal`.

Запись Identity базируется или на результате команды `nslookup` с параметром – именем хоста раздела, или на имени хоста, указанном в файле `/etc/hosts`, если DNS не настроен. Учтите, что механизм HBA выбирает первое имя, указанное после IP-адреса в файле `/etc/hosts`. Если Вы поменяете эту запись, Вы должны будете аутентифицировать новый Identity.

В нашем примере мы используем локальный файл `/etc/hosts` для разрешения имени хоста. Хотя команда `hostname` возвращает короткое имя хоста `plmserver`, команда `ctsvhbal` возвращает полностью квалифицированное имя, указанное в первой записи, соответствующей IP-адресу 192.168.255.85 в файле `/etc/hosts`:

```
hostname
```

```

plmserver
cat /etc/hosts | grep 192.168.255.85
192.168.255.85 plmserver.mydomain.com plmserver nimmaster1
vio_Client1.mydomain.com vio_client1
/usr/sbin/rsct/bin/ctsvhbal
ctsvhbal: The Host Based Authentication (HBA) mechanism identities for
the local system are:
 Identity: plmserver.mydomain.com
 Identity: 192.168.255.85
ctsvhbal: In order for remote authentication to be successful, at least one
of the above identities for the local system must appear in the trusted host
list on the remote node where a service application resides. Ensure that at
least one host name and one network address identity from the above list
appears in the trusted host list on any remote systems that act as servers
for applications executing on this local system.

```

Для того чтобы удаленная аутентификация PLM была успешной, первый Identity для локальной системы должен быть в списке доверенных хостов. Проверьте список доверенных хостов, используя команду ctsth1 с опцией -l:

- ▶ На сервере PLM должны быть записи Identity для каждого управляемого раздела, с которым Вы установили соединение RMC, и для самого сервера PLM.
- ▶ На управляемом разделе должны быть записи Identity только для сервера PLM и самого раздела.

В следующем примере показан вывод команды ctsth1 на сервере PLM:

```

/usr/sbin/rsct/bin/ctsth1 -l
ctsth1: Contents of trusted host list file:

Host Identity: vio_client2.mydomain.com
Identifier Generation Method: rsa512
Chapter 7. Partition Load Manager 381
Identifier Value:
120200d1af82e0530f9e80ddd291814cccd6339b41f0731cb1c65046797875e81c1c7f83a2d23288
a10cb9a44d75727e212442ef45c789801cc50242c98bc03f7b41a9d0103

Host Identity: plmserver.mydomain.com
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148a64546f8
9a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103

Host Identity: db_server.mydomain.com
Identifier Generation Method: rsa512
Identifier Value:
120200dc43ff154996991db8845b3d57b93cf37d674dc9bdab00f0b17923c9bc2a9f69bf88cc577
7e5af67d7d31d8e876aad2b0fac3f0f808db2ff9e286c143c8d97eb0103

Host Identity: app_server.mydomain.com
Identifier Generation Method: rsa512

```

```
Identifier Value:
120200b04353a9afea953e25846b6fe65b5062a7c2591ea52112427b699287cae1d669f9e27bcf1
26b4ab9c5e42326d1a278e1810e322fc0ec002b4febfa9f1b69c3630103

Host Identity: nimmaster1
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148a64546f8
9a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103

Host Identity: nimmaster1.mydomain.com
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148a64546f8
9a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103

Host Identity: 192.168.255.69
Identifier Generation Method: rsa512
Identifier Value:
120200c10d3ad7a20951523b7c01bd31f421be4ec9f98b1bf2d3051edd044ad19250be8c9db350f
f6fdeb0ddb46bc1b3365d33bd194382bb71b4784d88d8e7d740d78d0103

Host Identity: loopback
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148a64546f8
9a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103

Host Identity: 127.0.0.1
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148a64546f8
9a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103

Host Identity: 192.168.255.85
Identifier Generation Method: rsa512
Identifier Value:
120200e98f622904ebc0c1a7553ff2caf8daf74dc2473d4f03b7e62b5feecc699517148a64546f8
9a1f238dafddabe1187ade720bf04bf3a952709bad11cf64198d78f0103
```

Если Вы хотите удалить неправильные записи из Вашего списка доверенных хостов, используйте команду **ctsthl**:

```
/usr/sbin/rsct/bin/ctsthl -d -n vio_client2.mydomain.com
ctsthl: The following host was removed from the trusted host list:
vio_client2.mydomain.com
```

После того как Вы удалили Host Identity, Вы можете либо запустить настройку Вашего раздела снова для создания Host Identity, либо использовать команду **ctsthl** для корректного добавления Host Identity.

При использовании команды `ctsth1` Вы должны указать Host Identity, Identifier Generation Method (метод создания идентификатора) и Identifier Value (значение идентификатора).

```
/usr/sbin/rsct/bin/ctsth1 -a -n IDENTITY -m METHOD -p ID_VALUE
```

Более простой способ пересоздания (добавления заново) идентификаторов в список доверенных хостов предполагает использование интерфейса Web-based System Manager и повторного использования окна Management of Logical Partitions, как Вы уже делали во время начальной настройки. Вы можете также использовать скрипт `plmsetup` на сервере PLM, как показано ниже:

```
/etc/plm/setup/plmsetup vio_client2.mydomain.com root
```

На рис. 7-31 показан обзор рабочей конфигурации. В этом примере Вы можете использовать разрешение имен через `/etc/hosts` или DNS.

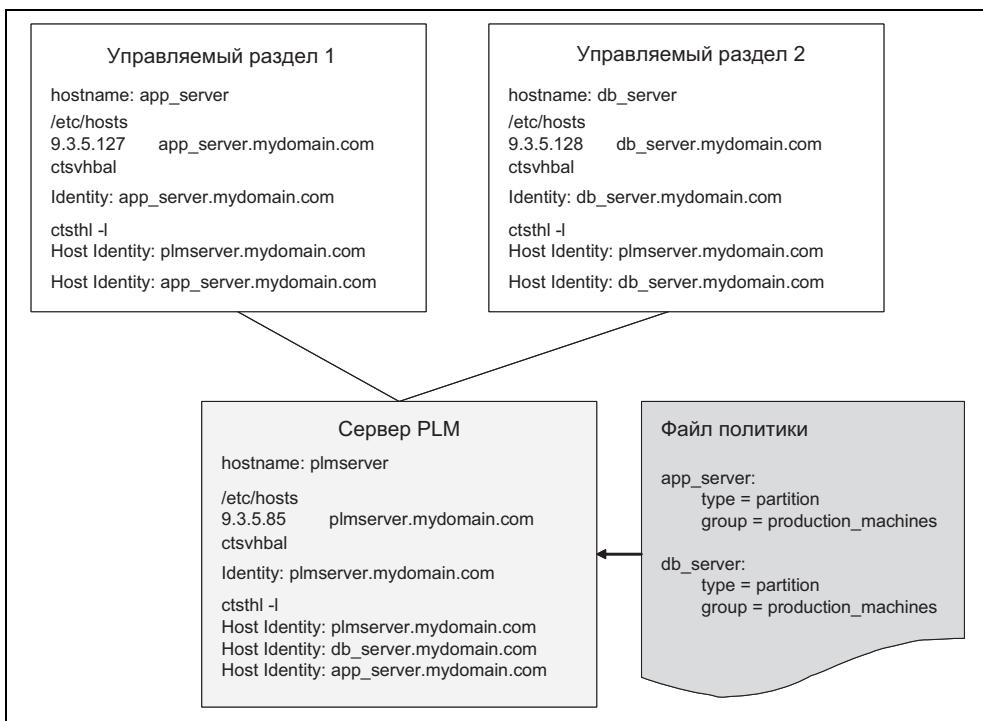


Рис. 7-31. Пример конфигурации для аутентификации PLM RMC

**Совет.** Для более легкой настройки и отладки используйте единую конвенцию именования. Используйте либо короткие, либо полностью квалифицированные имена хостов для всех управляемых разделов и сервера PLM. Если проблема все равно остается, проверьте `/etc/hosts` или DNS еще раз.

### 7.4.3. Поиск неисправностей на сервере PLM

В этой главе обсуждаются различные проблемы PLM, возникающие как при старте сервера PLM, так и после его старта.

- ▶ Если Вы сталкиваетесь с проблемами при старте сервера PLM (возникают ошибки, похожие на следующую), проверьте имя управляемой системы на HMC:

Status: Finished. Failed (5)

1498-031 Syntax error in policy, line 4.

PLM не указывает явно на проблему с именем управляемой системы, но сообщает о проблеме в строке 4. Проверьте, что имя управляемой системы не содержит пробелов, или попробуйте использовать более короткое имя управляемой системы.

- ▶ Если Ваш сервер PLM запускается успешно, проверьте файл журнала на наличие любых дополнительных сообщений об ошибках. Обновленная версия PLM предоставляет больше информации о проблемах, которые могут возникнуть, и передает сообщения об ошибках на HMC. В файле журнала PLM существует три типа сообщений:

**Trace** Информационные сообщения, показывающие действия, выполняемые PLM.

**Error** Ошибки на сервере PLM или агентах.

**External** Ошибки во внешнем окружении PLM, например сообщения об ошибках, поступающие от HMC.

- ▶ Для того чтобы определить Ваши управляемые разделы без значений Resource Entitlement, PLM запрашивает значения у HMC и отображает их в окне Show LPAR Statistic.

Если Вы сталкиваетесь с тем, что HMC не показывает значения (например, maximum, minimum и guaranteed), то может быть проблема с коммуникацией с HMC. Проверьте файл журнала на наличие дополнительной информации, похожей на следующую:

```
<08/18/05 12:33:21> <PLM_TRC> Cannot get the active profile for db_server.
```

```
lssyscfg returned 1.
```

Одна из возможных проблем состоит в том, что Ваш управляемый раздел сообщает неправильное имя раздела (не имя хоста) серверу PLM. Сервер PLM запрашивает имя раздела (partition name) у раздела, используя команду `uname -L`. Для проверки имени раздела (partition name) управляемого раздела используйте следующую команду:

```
uname -L
5 db_server
```

**Замечание.** Эта команда возвращает имя раздела, определенное на HMC, а не имя хоста раздела. Имя хоста (host name) раздела и имя раздела (partition name) на HMC могут быть различными.

Если Вы используете HMC для смены имени раздела, оно не обновляется в разделе до его перезагрузки. PLM может получить старое имя раздела, и команды HMC не будут выполняться, так как будет использоваться неправильное имя раздела.

- ▶ Если Вы видите в файле журнала сообщения, похожие на показанное ниже, проверьте настройки RMC, как описано в главах 7.2.3 «Настройка RMC для PLM» и 7.4.2 «Поиск неисправностей соединения RMC»:

```
<08/18/05 18:38:21> <PLM_TRC> Cannot establish an RMC session with
vio_client2. System call returned 39.
<08/18/05 18:38:21> <PLM_ERR> 2610-639 The user could not be authenticated
by the RMC subsystem.
```

## 7.5. Соглашения и ограничения PLM

Вы должны учитывать следующие ограничения при управлении системой с помощью Partition Load Manager:

- ▶ Partition Load Manager может использоваться в разделах под управлением AIX 5L Version 5.2 ML4 или AIX 5L Version 5.3. Поддержка Linux или i5/OS недоступна.
- ▶ Один экземпляр Partition Load Manager может управлять только одним сервером. Однако несколько экземпляров Partition Load Manager могут выполняться на одной системе, при этом каждый может управлять отдельным сервером или отдельной группой разделов на одном сервере.
- ▶ PLM не может перемещать ресурсы ввода-вывода между разделами. Partition Load Manager может управлять только ресурсами процессора и памяти.
- ▶ PLM не поддерживает управляемые разделы с выделенными процессорами и в общем пуле в одной группе. Вам нужно создать группу для разделов с выделенными процессорами и группу для разделов в общем пуле.
- ▶ Управление сервером Virtual I/O Server не поддерживается.
- ▶ Поддерживается управление разделом, в котором установлен Partition Load Manager.
- ▶ Partition Load Manager не поддерживает резервные НМС. Вы можете создать второй профайл с информацией о второй НМС. В случае возникновения проблем с основной НМС вы можете переключить профайл вручную.
- ▶ Partition Load Manager не поддерживает системы, управляемые IVM.

## 7.6. Управление ресурсами

Владельцы серверов IBM System p5, использующие AIX 5L V5.3 с опцией Advanced POWER Virtualization, имеют возможность выбора из трех механизмов управления рабочей нагрузкой:

**Общие процессоры, разделы без ограничений (uncapped)**

Гипервизор POWER передает неиспользованные процессорные такты тем разделам без ограничений, которые могут получить от них пользу.

**Workload Manager (WLM)**

Устанавливает приоритеты приложений на доступ к системным ресурсам: ЦП, памяти и подсистеме ввода-вывода в пределах раздела.

### **Partition Load Manager (PLM)**

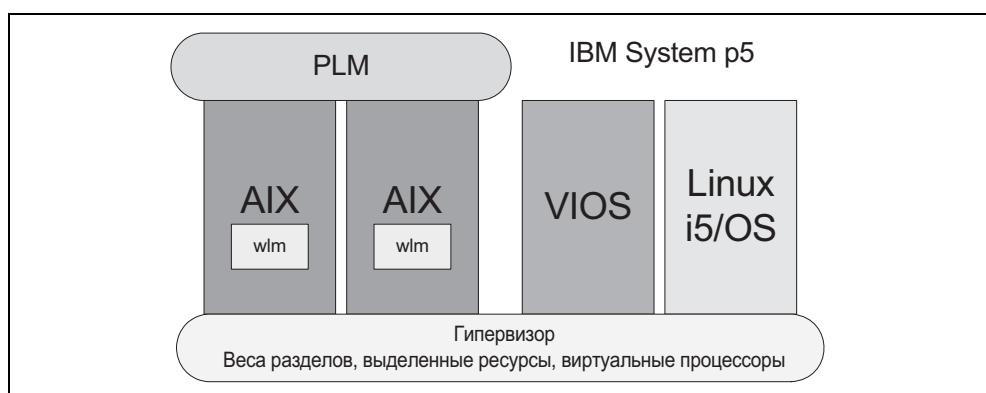
Добавляет в раздел и перемещает между разделами ресурсы ЦП и памяти, используя динамические операции LPAR.

**Замечание.** AIX 5L (планировщик с его механизмом приоритетов процессов/нитей и диспетчер виртуальной памяти (VMM)) также управляет ресурсами, но здесь это не обсуждается.

PLM и WLM поддерживаются только в AIX 5L; в i5/OS или Linux они не поддерживаются.

В этой главе обсуждается взаимодействие между всеми этими механизмами и то, как они могут быть использованы вместе для оптимизации использования ресурсов и производительности. Предполагается понимание концепций PLM, WLM и общих процессоров. Сравнение этих и других технологий предоставления ресурсов можно найти в главе 4 «*pSeries provisioning tools overview*», руководства *Introduction to pSeries Provisioning*, SG24-6389.

На рис. 7-32 показана область действия каждого из вышеуказанных механизмов управления загрузкой и ресурсами на серверах IBM System p5.



**Рис. 7-32. Механизмы управления ресурсами и рабочей нагрузкой**

#### **7.6.1. Управление ресурсами и рабочей нагрузкой**

Существуют два аспекта управления ресурсами и рабочей нагрузкой, общие для всех вышеупомянутых механизмов.

Во-первых, механизмы управления ресурсами и рабочей нагрузкой дают эффект только при *нехватке* системных ресурсов и конкурентной борьбе за эти ресурсы, т.е., в том случае, если системных ресурсов недостаточно для удовлетворения требований нагрузок (приложений) и существует две или более активных нагрузок. Если система загружена *нормально* или если есть только один потребитель ресурсов, то диспетчеры ресурсов не дают существенного эффекта, даже если они вмешиваются в работу.

Во-вторых, все механизмы управления ресурсами и рабочей нагрузкой требуют наличия точной политики, описывающей (относительные) приоритеты управляемых компонентов, что подразумевает идентификацию тех компонентов, которые будут оштрафованы, и тех, которые получат преимущество при нехватке ресурсов.

**Замечание.** Если существует пул свободных ресурсов (неиспользуемая память или неиспользуемые процессоры), то PLM будет использовать эти ресурсы до отбиения ресурсов из других разделов. Пул состоит из тех ресурсов, за которые нет конкурентной борьбы.

### Особенности WLM, PLM и общего процессорного пула

Каждый из доступных механизмов управления ресурсами имеет свои особенности, которые отличают его от остальных.

#### WLM

WLM контролирует и управляет использованием ЦП, памяти и дискового ввода-вывода в отдельной системе AIX 5L или в разделе. Он может ограничить использование этих ресурсов отдельным процессом или группой процессов. WLM помещает каждый процесс в класс (класс может состоять из нескольких процессов). Каждому классу выдается определенное количество общих ресурсов. Соотношение ресурсов, принадлежащих классу, к сумме общих ресурсов, принадлежащих всем активным классам, дает пропорциональное количество ресурсов, которые может получить класс. Такой механизм дает самонастраивающийся приоритет.

В виртуальной среде WLM способен управлять конкурентными запросами на память, выделенные и виртуальные процессоры и дисковый ввод-вывод в пределах одного экземпляра AIX 5L.

В разделах с выделенными процессорами WLM может привязывать классы к конкретным ЦП, используя наборы процессорных ресурсов.

WLM периодически оценивает использование ресурсов.

WLM periodically evaluates resource usage. Он выполняет необходимое регулирование использования ресурсов путем смены приоритетов процессов. WLM может управлять конфликтами использования ресурсов, если они происходят, в течение всего периода времени, пока сохраняется конфликтная ситуация.

**Замечание:**

- ▶ WLM видит только использование ресурсов в пределах раздела. Он не видит то, что происходит в других разделах, и не может изменить приоритет раздела относительно других разделов.
- ▶ WLM поставляется как часть AIX 5L.

#### PLM

PLM управляет памятью, разделами с выделенными процессорами и разделами в общем процессорном пуле.

PLM управляет разделами. Он не знает о важности любой из рабочих нагрузок в разделе и, таким образом, не может переназначать приоритеты, основываясь на смене типов нагрузок.

PLM принимает решения о выделении ресурсов, основываясь на файле политики, определенном системным администратором, и делает запросы к НМС на выполнение соответствующих динамических операций над LPAR.

PLM характеризуется относительно большим временем задержки (порядка минут). Это большое время задержки делает PLM подходящим решением только для среднесрочных и долговременных изменений в использовании ресурсов и неэффективным при управлении кратковременными пиками.

### **Общие процессоры**

При использовании разделов, работающих в общем процессорном пуле, гипервизор POWER управляет пулом процессоров, разделяемых между набором разделов. Неиспользуемые процессорные такты могут быть перемещены в неограниченные (uncapped) виртуальные процессоры, которые нуждаются в них, давая им больше ресурсов, чем им позволено в нормальном режиме.

Гипервизор POWER характеризуется относительно небольшим временем задержки. Он имеет квант диспетчеризации 10 мс и может производить изменения в диспетчеризации посредине кванта. Динамические операции LPAR по увеличению мощности виртуальных процессоров занимают несколько секунд.

Виртуальные процессоры должны диспетчеризоваться на физических процессорах. Наличие очень большого количества виртуальных процессоров (во всех разделах) относительно количества физических разделов в пуле может снизить общую производительность системы. В AIX 5L Version 5.3 Maintenance Level 3 побочные эффекты использования большого количества виртуальных процессоров значительно уменьшены (см. «Свертывание виртуальных процессоров»).

### **7.6.2. Как оценивается загрузка**

Все диспетчеры управления ресурсами и рабочей нагрузкой основываются на измерении использования ресурсов. Это может быть измерение в конкретный момент времени или среднее значение за период времени. Понимание того, как PLM, WLM и гипервизор POWER измеряют использование ресурсов, необходимо для понимания того, как они будут взаимодействовать.

#### **Оценка загрузки памяти**

Точное определение того, сколько физической памяти активно используется всеми приложениями в AIX 5L, – сложная задача из-за стратегии AIX 5L обеспечивать наилучшее использование всех ресурсов путем выделения их при необходимости и оставления страниц в физической памяти, даже если они больше не требуются.

Как альтернатива измерения использования физической памяти, показываемого командой `vmstat`, возможно косвенно оценить загруженность памяти по количеству (и скорости) выгрузки страниц (paging rate).

WLM использует стандартную статистику загрузки памяти, схожую с предоставляемой командой `vmstat`. PLM использует обе метрики для оценки загрузки памяти в разделе.

### Оценка загрузки ЦП

Традиционно оценка загрузки ЦП была простой задачей. Достаточно измерить количество времени, в течение которого ЦП был загружен в этот временной интервал. В многопроцессорных системах бралось среднее значение загрузки всех процессоров для получения одной величины.

При использовании общих ресурсов требуется другой подход, поэтому процессор POWER5 имеет новый регистр PURR для измерения загрузки ЦП в виртуальной среде. Использование регистра PURR для мониторинга системы с общими процессорами обсуждается в главе 6.5 «Мониторинг в виртуальном окружении». Только то, что загрузка процессора близка к или превышает 100% в неограниченном разделе с общими процессорами, не обязательно означает нехватку процессорных ресурсов. Она просто показывает, что раздел получает больше, чем его выделенная мощность (гарантированный минимум). PLM и WLM используют разные стратегии для оценки загрузки ЦП в разделе.

WLM измеряет использование ресурсов для определения того, необходимо ли изменение приоритетов. PLM использует загрузку, т.к. использование ресурсов не говорит о том, что нужны дополнительные ресурсы.

### PLM

PLM использует величину, называемую *load average* (*средняя загрузка*). Она показывает среднюю длину очереди на выполнение в AIX 5L за интервал времени (схожую с величинами, показываемыми в первой строке вывода команд `w` и `prttime` для 1-, 5- и 15-минутных интервалов). PLM использует средневзвешенное окно, которое позволяет обнаружить как кратковременные пики, так и долговременные тенденции, и нормализует значение к количеству настроенных логических процессоров. Команда `lsrsrc -Ad IBM.LPAR` показывает значение средней загрузки, используемое PLM.

### WLM

WLM периодически оценивает реальные ресурсы всех систем и все свои классы. Если ресурсы ЦП не используются полностью, то WLM не будет вмешиваться. Если занятость ЦП начинает приближаться к 100%, WLM начинает менять приоритеты всех процессов в классах так, чтобы реальное использование ЦП каждым классом приближалось к указанному целевому значению.

В случае использования неограниченных разделов в общем пуле потребление классом ЦП вычисляется, основываясь на процессорном времени, выданном разделу:

$$\text{consumation(CLASS)} = \frac{\text{chu\_time(CLASS)}}{\text{chu\_time(PARTITION)}}$$

Величина использования ресурсов (с точки зрения WLM) никогда не будет превышать 100%.

## **WLM и динамическая реконфигурация**

WLM знает о динамических LPAR. Он будет автоматически пересчитывать целевые значения использования ресурсов при добавлении или убиании ресурсов раздела.

### **7.6.3. Управление ресурсами ЦП**

В этой главе объясняется, как PLM и WLM обрабатывают виртуальные процессоры в неограниченных разделах, работающих в общем процессорном пуле. Нет специальных соглашений для выделенных разделов и ограниченных разделов, работающих в общем пуле, так как последние функционируют в основном так же, как разделы с выделенными процессорами.

#### **PLM и неограниченные виртуальные процессоры**

Когда средняя загрузка ЦП в разделе, управляемом PLM, поднимается выше порогового значения PLM, то PLM попытается увеличить мощность раздела и будет постепенно увеличивать количество виртуальных разделов, основываясь на активной политике.

Существует два параметра настройки в политике, влияющих на добавление и удаление виртуальных процессоров:

- ▶ Минимальная мощность на виртуальный процессор
- ▶ Максимальная мощность на виртуальный процессор

Каждый раз, когда изменяется мощность, PLM добавляет или убирает процессорные единицы, указанные в параметре настройки entitled capacity delta. При добавлении мощности, если новая мощность на процессор превышает максимальную на процессор, PLM будет добавлять в раздел один или несколько виртуальных процессоров. При удалении мощности, если новая мощность ниже, чем минимальная на процессор, PLM будет убирать из раздела один или несколько виртуальных процессоров.

#### **WLM и виртуальные процессоры**

Наборы ресурсов, вместе с командой и системным вызовом *bindprocessor*, позволяют процессам и классам WLM (приложениям) быть привязанными к процессорам. Эта *жесткая локальная связь* обычно делается для того, чтобы убедиться в том, что процессорные кэши *наполнены* данными приложения, что увеличивает производительность.

Разделы, работающие в общем пуле, поддерживают привязку к процессорам (команды и системные вызовы не будут возвращать ошибку), но влияние на производительность приложений будет не таким, как на разделах с выделенными процессорами:

- ▶ Хотя гипервизор и пытается поддерживать «сходство» (affinity), нет гарантированной фиксированной связи между виртуальными и физическими процессорами.
- ▶ Если выделенная мощность виртуального процессора меньше, чем 1.0 (100 процессорных единиц, PU), то гипервизор будет использовать тот же физический процессор и для других виртуальных процессоров (потенциально и из других разделов).

- Если виртуальный процессор использует менее 100% физического процессора, избыточные такты физического процессора отдаются в общий пул для использования другими виртуальными процессорами.

#### **7.6.4. Управление ресурсами памяти**

PLM управляет памятью всех управляемых разделов с активизированным управлением памятью. PLM перемещает ресурсы только в пределах группы разделов; он *не* перемещает ресурсы, память или ЦП между разделами в двух разных группах разделов.

#### **7.6.5. Какой инструмент управления ресурсами использовать?**

WLM управляет конфликтами использования ресурсов в пределах раздела. Если у раздела есть достаточное количество ресурсов для рабочей нагрузки или в разделе работает только одно приложение, то конкурентной борьбы за ресурсы не будет и наличие WLM не будет играть роли. В большинстве случаев WLM используется при консолидации нескольких различных приложений на одном сервере или разделе под управлением AIX 5L.

Разделы в общем пуле и гипервизор POWER могут перемещать ресурсы ЦП из одного раздела в другой практически моментально. Использование разделов, работающих в общем пуле, адекватно в том случае, если отмечаются кратковременные флуктуации рабочей нагрузки, а консолидация в один раздел AIX 5L не подходит.

Разделы в общем пуле могут использоваться в том случае, если им требуется часть процессора POWER5, например, два раздела с мощностью 1.5 каждый выполняются на трех процессорах POWER5 в общем пуле.

Partition Load Manager характеризуется относительно большим временем задержки и не может реагировать на кратковременные пиковые нагрузки. PLM обрабатывает среднесрочные и долговременные тенденции; он может управлять необходимой миграцией ресурсов, например, при переходе от дневных транзакций к ночных пакетным работам и наоборот.

PLM может быть использован для оптимальной настройки серверов со стабильными рабочими нагрузками. Установив начальную конфигурацию раздела с минимальными ресурсами и оставив остальные ресурсы в свободном пуле (память и процессоры), PLM будет перемещать в каждый раздел только необходимые ему для удовлетворения требованиям рабочей нагрузки ресурсы (если они доступны). Это уменьшает необходимость в выполнении точной оценки, необходимой для распределения аппаратных ресурсов сервера между разделами.

Для получения большей информации обратитесь к руководству *Introduction to pSeries Provisioning*, SG24-6389.



A

## **Характеристики надежности, готовности и ремонтопригодности (RAS) System p5**

В следующих таблицах представлена сводка характеристик RAS для различных систем IBM @server p5.

**Таблица А-1.** Сводные данные функций RAS для различных моделей System p5

| Функция RAS                                                   | p505/<br>p510 | p520  | p550/<br>p550Q | p570  | p590/<br>p595 |
|---------------------------------------------------------------|---------------|-------|----------------|-------|---------------|
| Сервисный процессор                                           | X             | X     | X              | X     | X             |
| Резервный серверный процессор                                 | Н/Д           | Н/Д   | Н/Д            | Опция | X             |
| Резервные часы                                                | Н/Д           | Н/Д   | Н/Д            | Н/Д   | X             |
| Высвобождение системой сбойных компонентов                    | X             | X     | X              | X     | X             |
| Функция повышения готовности памяти и L3-кэша                 | X             | X     | X              | X     | X             |
| Заем памяти из CUoD во время начальной загрузки               | Н/Д           | Н/Д   | Н/Д            | X     | X             |
| Функции повышения готовности для процессора и L1/L2-кэшей     | X             | X     | X              | X     | X             |
| Динамический заем процессора из CUoD                          | Н/Д           | Н/Д   | X              | X     | X             |
| Блоки ввода-вывода 7311-D10 и 7311-D11                        | Н/Д           | Н/Д   | Н/Д            | X     | Н/Д           |
| Блок ввода-вывода 7311-D20                                    | Н/Д           | X     | X              | X     | Н/Д           |
| Блок ввода-вывода 7040-61D (только PCI-X)                     | Н/Д           | Н/Д   | Н/Д            | Н/Д   | X             |
| Резервированные подключения блоков ввода-вывода               | Н/Д           | X     | X              | X     | X             |
| Резервированные или N+1 источники питания и вентиляторы       | Опция         | Опция | Опция          | X     | X             |
| Интегрированная резервная батарея (Integrated Battery Backup) | Н/Д           | Н/Д   | Н/Д            | Н/Д   | Опция         |

**Таблица А-2.** Поддержка операционными системами отдельных функций RAS

| Функция RAS                                         | AIX<br>5L<br>V5.2 | AIX<br>5L<br>V5.3 | i5/OS | RHEL<br>AS V3 | RHEL<br>AS V4 | SLES<br>V9        |
|-----------------------------------------------------|-------------------|-------------------|-------|---------------|---------------|-------------------|
| Высвобождение системой сбойных компонентов          |                   |                   |       |               |               |                   |
| Динамическое высвобождение процессора               | X                 | X                 | X     | -             | X             | X                 |
| Динамическая замена процессора во время работы      | X                 | X                 | X     | X             | X             | X                 |
| Использование процессоров CUoD                      | X                 | X                 | X     | X             | X             | X                 |
| Использование мощности из активного резервного пула | X                 | X                 | X     | X             | X             | X                 |
| Постоянное высвобождение процессора                 | X                 | X                 | X     | X             | X             | X                 |
| Ручное резервирование памяти из активного пула      | X                 | X                 | X     |               |               |                   |
| Заем памяти из CUoD во время начальной загрузки     | X                 | X                 | X     | X             | X             | X                 |
| Постоянное высвобождение шины GX+ bus               | X                 | X                 | X     | -             | -             | -                 |
| Расширенное обнаружение ошибок шины PCI             | X                 | X                 | X     | X             | X             | X                 |
| Расширенное восстановление после ошибок шины PCI    | X                 | X                 | X     | -             | -             | Огра-<br>ниченено |
| Расширенная обработка ошибок PCI-PCI bridge         | X                 | X                 | X     | -             | -             | -                 |
| Резервный канал RIO                                 | X                 | X                 | X     | X             | X             | X                 |

*Продолжение табл.*

| Функция RAS                                                               | AIX<br>5L<br>V5.2 | AIX<br>5L<br>V5.3 | i5/OS | RHEL<br>AS V3        | RHEL<br>AS V4        | SLES<br>V9        |
|---------------------------------------------------------------------------|-------------------|-------------------|-------|----------------------|----------------------|-------------------|
| Диски с горячей заменой (внутренние/ внешние)                             | X                 | X                 | X     | X                    | X                    | X                 |
| Горячая замена карты PCI                                                  | X                 | X                 | X     | -                    | -                    | X                 |
| Переключение на резерв сервисного процессора во время начальной загрузки  | X                 | X                 | X     | X                    | X                    | X                 |
| Переключение на резерв системных часов во время начальной загрузки        | X                 | X                 | X     | X                    | X                    | X                 |
| Функции доступности памяти                                                |                   |                   |       |                      |                      |                   |
| Память, L2- и L3-кэши с поддержкой ECC                                    | X                 | X                 | X     | X                    | X                    | X                 |
| Проверка четности L1 и повторное обращение                                | X                 | X                 | X     | X                    | X                    | X                 |
| Dynamic bit-steering (резерв памяти в основном модуле)                    | X                 | X                 | X     | X                    | X                    | X                 |
| Память Chipkill™                                                          | X                 | X                 | X     | X                    | X                    | X                 |
| Memory scrubbing (очистка памяти)                                         | X                 | X                 | X     | X                    | X                    | X                 |
| Улучшенное удаление строки L3-кэша                                        | X                 | X                 | X     | X                    | X                    | X                 |
| Массивное восстановление L1-, L2- и L3-кэшней                             | X                 | X                 | X     | X                    | X                    | X                 |
| Специальная обработка неисправимых ошибок                                 | X                 | X                 | X     |                      |                      |                   |
| Обнаружение и изоляция сбоев                                              |                   |                   |       |                      |                      |                   |
| Диагностика с использованием технологии FFDC (First Failure Data Capture) | X                 | X                 | X     | X                    | X                    | X                 |
| Диагностика подсистемы с использованием ввода-вывода FFDC                 | X                 | X                 | X     | -                    | -                    | X                 |
| Диагностика во время работы                                               | X                 | X                 | X     | Огра-<br>ниче-<br>но | Огра-<br>ниче-<br>но | Огра-<br>ниченено |
| Анализ журнала ошибок                                                     | X                 | X                 | X     | X                    | X                    | X                 |
| Поддержка сервисного процессора для:                                      | X                 | X                 | X     | X                    | X                    | X                 |
| - встроенной самодиагностики (BIST) для логики и массивов                 | X                 | X                 | X     | X                    | X                    | X                 |
| - тестирования соединений                                                 | X                 | X                 | X     | X                    | X                    | X                 |
| - инициализации компонентов                                               | X                 | X                 | X     | X                    | X                    | X                 |
| Ремонтопригодность                                                        |                   |                   |       |                      |                      |                   |
| Индикатор процесса загрузки                                               | X                 | X                 | X     | Огра-<br>ниче-<br>но | Огра-<br>ниче-<br>но | Огра-<br>ниченено |
| Коды ошибок микрокода                                                     | X                 | X                 | X     | X                    | X                    | X                 |

*Продолжение табл.*

| Функция RAS                                                              | AIX<br>5L<br>V5.2 | AIX<br>5L<br>V5.3 | i5/OS | RHEL<br>AS V3        | RHEL<br>AS V4        | SLES<br>V9      |
|--------------------------------------------------------------------------|-------------------|-------------------|-------|----------------------|----------------------|-----------------|
| Коды ошибок операционной системы                                         | X                 | X                 | X     | Огра-<br>ниче-<br>но | Огра-<br>ниче-<br>но | Огра-<br>ничено |
| Сбор инвентарных данных                                                  | X                 | X                 | X     | X                    | X                    | X               |
| Предупреждения об условиях внешней среды и питания                       | X                 | X                 | X     | X                    | X                    | X               |
| Вентиляторы, источники питания и регуляторы напряжения с горячей заменой | X                 | X                 | X     | X                    | X                    | X               |
| Расширенный сбор данных об ошибках                                       | X                 | X                 | X     | X                    | X                    | X               |
| Функция «call home» сервисного процессора без аппаратной консоли (HMC)   | X                 | X                 | X     | X                    | X                    | X               |
| Отключение кабелей RIO во время работы                                   | X                 | X                 | X     | X                    | X                    | X               |
| Взаимная диагностика сервисного процессора и гипервизора POWER           | X                 | X                 | X     | X                    | X                    | X               |
| Динамическое обновление микрокода с HMC                                  |                   |                   |       |                      |                      |                 |
| Сервисный агент                                                          | X                 | X                 | X     | -                    | -                    | X               |
| Системные дампы для памяти, сервисного процессора и гипервизора          | X                 | X                 | X     | X                    | X                    | X               |
| Передача ошибок операционной системы в приложение HMC SFP                | X                 | X                 | X     | X                    | X                    | X               |
| Подсистема безопасной передачи сообщений об ошибках RMC                  | X                 | X                 | X     | X                    | X                    | X               |
| Планирование операций проверки работоспособности с HMC                   | X                 | X                 | -     | X                    | X                    | X               |
| Автоматическое восстановление/ старт сервера                             | X                 | X                 | X     | X                    | X                    | X               |
| Кластеризация                                                            |                   |                   |       |                      |                      |                 |
| Кластер высокой доступности HACMP с прямым вводом-выводом                | X                 | X                 | -     | -                    | -                    | -               |
| Кластер высокой доступности HACMP с VIOS                                 |                   | X                 | -     | -                    | -                    | -               |



B

## Системные вызовы confer и cede гипервизора POWER

Для оптимизации использования физического процессора виртуальный процессор будет уступать физический процессор, если у него нет работ на выполнение или если он вошел в состояние ожидания (wait-state), например ожидание блокировки или завершения операции ввода-вывода. Виртуальный процессор может уступить физический процессор, используя системный вызов гипервизора. Есть два различных семейства системный вызовов (hcalls) для освобождения виртуального процессора:

confer<sup>a</sup>

Системный вызов confer используется для передачи оставшихся в текущем интервале диспетчеризации процессорных тактов другому виртуальному процессору в том же разделе. Он используется в том случае, когда один виртуальный процессор не может продолжать обрабатывать задачу из-за того, что он ожидает завершения события на другом виртуальном процессоре, например завершения блокировки (lock miss).

cede<sup>b</sup>

Системный вызов cede используется в том случае, когда виртуальный процессор простояивает, например, из-за ожидания завершения операции ввода-вывода. Системный вызов cede позволяет гипервизору POWER диспетчеризировать другой виртуальный процессор из другого раздела.

<sup>a</sup> To confer – даровать, присуждать (англ.) Прим. науч. ред.

<sup>b</sup> To cede – уступать (англ.) Прим. науч. ред.

Системный вызов confer имеет три формы:

- |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>h_confer_to_self</b>   | Неиспользованные процессорные единицы (processing units) сделавшего вызов виртуального процессора равномерно распределяются между остальными виртуальными процессорами в том же разделе. Сделавший вызов виртуальный процессор остается неактивным до того момента, как он будет разбужен системным вызовом <b>h_prod</b> . Владеющий процессором раздел не будет диспетчеризовать задачи на этот виртуальный процессор, пока он заморожен, и следующие проходы диспетчера гипервизора не будут диспетчеризовать этот виртуальный процессор на физический процессор. |
| <b>h_confer_to_target</b> | Неиспользуемые процессорные единицы сделавшего системный вызов виртуального процессора передаются конкретному виртуальному процессору. Системный вызов <b>h_confer_to_target</b> выполняется в том случае, когда виртуальный процессор ждет блокировки и способен определить, какой из остальных виртуальных процессоров в разделе обслуживает нить, держащую блокировку. Сделавший системный вызов виртуальный процессор получит свою нормальную мощность на следующем проходе гипервизора.                                                                         |
| <b>h_confer_to_all</b>    | Этот системный вызов похож на <b>h_confer_to_self</b> , с отличием в том, что виртуальный процессор остается активным и будет диспетчеризоваться во время следующего прохода гипервизора.                                                                                                                                                                                                                                                                                                                                                                            |
| <b>h_prod</b>             | Активизирует виртуальный процессор, уступивший ранее свои процессорные такты.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

# Аббревиатуры и акронимы

|        |                                                                                   |
|--------|-----------------------------------------------------------------------------------|
| ABI    | Application Binary Interface, бинарный интерфейс приложений                       |
| AC     | Alternating Current, переменный ток                                               |
| ACL    | Access Control List, список управления доступом                                   |
| AFPA   | Adaptive Fast Path Architecture                                                   |
| AIO    | Asynchronous I/O, асинхронный ввод-вывод                                          |
| AIX    | Advanced Interactive Executive                                                    |
| APAR   | Authorized Program Analysis Report                                                |
| API    | Application Programming Interface, программный интерфейс приложений               |
| ARP    | Address Resolution Protocol, протокол разрешения адресов                          |
| ASMI   | Advanced System Management Interface, расширенный интерфейс системного управления |
| BFF    | Backup File Format, формат архивного файла                                        |
| BIND   | Berkeley Internet Name Domain, встроенная самодиагностика                         |
| BIST   | Built-In Self-Test                                                                |
| BLV    | Boot Logical Volume, загрузочный логический том                                   |
| BOOTP  | Boot Protocol                                                                     |
| BOS    | Base Operating System, базовая операционная система                               |
| BSD    | Berkeley Software Distribution                                                    |
| CA     | Certificate Authority                                                             |
| CATE   | Certified Advanced Technical Expert                                               |
| CD     | Compact Disk, компакт-диск                                                        |
| CDE    | Common Desktop Environment                                                        |
| CD-R   | CD Recordable                                                                     |
| CD-ROM | Compact Disk-Read Only Memory                                                     |
| CEC    | Central Electronics Complex, блок ЦП и памяти                                     |
| CHRP   | Common Hardware Reference Platform                                                |
| CLI    | Command Line Interface, интерфейс командной строки                                |
| CLVM   | Concurrent LVM                                                                    |
| CPU    | Central Processing Unit, центральный процессор                                    |
| CRC    | Cyclic Redundancy Check, циклическая контрольная сумма                            |
| CSM    | Cluster Systems Management                                                        |
| CUoD   | Capacity Upgrade on Demand                                                        |
| DCM    | Dual Chip Module                                                                  |
| DES    | Data Encryption Standard                                                          |
| DGD    | Dead Gateway Detection                                                            |
| DHCP   | Dynamic Host Configuration Protocol, протокол динамической конфигурации хоста     |
| DLPAR  | Dynamic LPAR, динамический LPAR                                                   |
| DMA    | Direct Memory Access, прямой доступ к памяти                                      |

|       |                                                                             |
|-------|-----------------------------------------------------------------------------|
| DNS   | Domain Naming System, доменная система имен                                 |
| DRM   | Dynamic Reconfiguration Manager, диспетчер динамической реконфигурации      |
| DR    | Dynamic Reconfiguration, динамическая реконфигурация                        |
| DVD   | Digital Versatile Disk                                                      |
| EC    | EtherChannel                                                                |
| ECC   | Error Checking and Correcting                                               |
| EOF   | End of File, конец файла                                                    |
| EPOW  | Environmental and Power Warning                                             |
| ERRM  | Event Response resource manager                                             |
| ESS   | Enterprise Storage Server                                                   |
| F/C   | Feature Code                                                                |
| FC    | Fibre Channel                                                               |
| FCAL  | Fibre Channel Arbitrated Loop                                               |
| FDX   | Full Duplex, полный дуплекс                                                 |
| FLOP  | Floating Point Operation, операции с плавающей точкой                       |
| FRU   | Field Replaceable Unit, типовой элемент замены                              |
| FTP   | File Transfer Protocol, протокол FTP                                        |
| GDPS  | Geographically Dispersed Parallel Sysplex                                   |
| GID   | Group ID, идентификатор группы                                              |
| GPFS  | General Parallel File System, параллельная файловая система                 |
| GUI   | Graphical User Interface, графический интерфейс пользователя                |
| HACMP | High Availability Cluster Multiprocessing                                   |
| HBA   | Host Bus Adapters                                                           |
| HMC   | Hardware Management Console, консоль управления оборудованием               |
| HTML  | Hypertext Markup Language, язык разметки гипертекста                        |
| HTTP  | Hypertext Transfer Protocol, протокол передачи гипертекста                  |
| Hz    | Hertz, Герц                                                                 |
| I/O   | Input/Output, ввод-вывод                                                    |
| IBM   | International Business Machines                                             |
| ID    | Identification, идентификация (идентификатор)                               |
| IDE   | Integrated Device Electronics                                               |
| IEEE  | Institute of Electrical and Electronics Engineers                           |
| IP    | Internet Protocol, протокол IP                                              |
| IPAT  | IP Address Takeover, передача IP-адреса                                     |
| IPL   | Initial Program Load, начальная загрузка                                    |
| IPMP  | IP Multipathing                                                             |
| ISV   | Independent Software Vendor, независимый поставщик программного обеспечения |
| ITSO  | International Technical Support Organization                                |
| IVM   | Integrated Virtualization Manager, интегрированный менеджер виртуализации   |
| JFS   | Journalized File System, журнальная файловая система                        |
| L1    | Level 1                                                                     |
| L2    | Level 2                                                                     |
| L3    | Level 3                                                                     |
| LA    | Link Aggregation, агрегирование каналов                                     |

|       |                                                                                  |
|-------|----------------------------------------------------------------------------------|
| LACP  | Link Aggregation Control Protocol, протокол агрегирования каналов                |
| LAN   | Local Area Network, локальная сеть                                               |
| LDAP  | Lightweight Directory Access Protocol                                            |
| LED   | Light Emitting Diode, светодиод                                                  |
| LMB   | Logical Memory Block                                                             |
| LPAR  | Logical Partition, логический раздел                                             |
| LPP   | Licensed Program Product, лицензионный программный продукт                       |
| LUN   | Logical Unit Number, номер логического устройства                                |
| LV    | Logical Volume, логический том                                                   |
| LVCB  | Logical Volume Control Block, контрольный блок логического тома                  |
| LVM   | Logical Volume Manager, менеджер логических томов                                |
| MAC   | Media Access Control                                                             |
| Mbps  | Megabits Per Second, мегабиты в секунду                                          |
| MBps  | Megabytes Per Second, мегабайты в секунду                                        |
| MCM   | Multichip Module                                                                 |
| ML    | Maintenance Level                                                                |
| MP    | Multiprocessor                                                                   |
| MPIO  | Multipath I/O                                                                    |
| MTU   | Maximum Transmission Unit                                                        |
| NFS   | Network File System, сетевая файловая система                                    |
| NIB   | Network Interface Backup, резервирование сетевого интерфейса                     |
| NIM   | Network Installation Management, управление сетевой инсталляцией (установкой)    |
| NIMOL | NIM on Linux                                                                     |
| NVRAM | Non-Volatile Random Access Memory, энергонезависимая память                      |
| ODM   | Object Data Manager, диспетчер объектных данных                                  |
| OSPF  | Open Shortest Path First                                                         |
| PCI   | Peripheral Component Interconnect                                                |
| PIC   | Pool Idle Count                                                                  |
| PID   | Process ID, идентификатор процесса                                               |
| PKI   | Public Key Infrastructure, инфраструктура открытых ключей                        |
| PLM   | Partition Load Manager, диспетчер управления нагрузкой в разделах                |
| POST  | Power-On Self-test, начальная самодиагностика                                    |
| POWER | Performance Optimization with Enhanced Risc (архитектура)                        |
| PPC   | Physical Processor Consumption, использование ресурсов физического процессора    |
| PPFC  | Physical Processor Fraction Consumed, использованная доля физического процессора |
| PTF   | Program Temporary Fix                                                            |
| PTX   | Performance Toolbox                                                              |
| PURR  | Processor Utilization Resource Register                                          |
| PV    | Physical Volume, физический том                                                  |
| PVID  | Physical Volume Identifier, идентификатор физического тома                       |
| PVID  | Port Virtual LAN Identifier, идентификатор порта VLAN                            |
| QoS   | Quality of Service                                                               |
| RAID  | Redundant Array of Independent Disks                                             |
| RAM   | Random Access Memory, ОЗУ                                                        |

|        |                                                                                            |
|--------|--------------------------------------------------------------------------------------------|
| RAS    | Reliability, Availability, and Serviceability, Надежность, Доступность, Ремонтопригодность |
| RCP    | Remote Copy                                                                                |
| RDAC   | Redundant Disk Array Controller                                                            |
| RIO    | Remote I/O                                                                                 |
| RIP    | Routing Information Protocol                                                               |
| RISC   | Reduced Instruction-Set Computer                                                           |
| RMC    | Resource Monitoring and Control                                                            |
| RPC    | Remote Procedure Call, удаленный вызов процедур                                            |
| RPL    | Remote Program Loader, удаленный загрузчик                                                 |
| RPM    | Red Hat Package Manager                                                                    |
| RSA    | Rivet, Shamir, Adelman                                                                     |
| RSCT   | Reliable Scalable Cluster Technology                                                       |
| RSH    | Remote Shell                                                                               |
| SAN    | Storage Area Network, сеть хранения данных                                                 |
| SCSI   | Small Computer System Interface                                                            |
| SDD    | Subsystem Device Driver                                                                    |
| SMIT   | System Management Interface Tool                                                           |
| SMP    | Symmetric Multiprocessor, симметричное мультипроцессорование                               |
| SMS    | System Management Services, сервисы системного управления                                  |
| SMT    | Simultaneous Multithreading, одновременная многопоточность                                 |
| SP     | Service Processor, сервисный процессор                                                     |
| SPOT   | Shared Product Object Tree                                                                 |
| SRC    | System Resource Controller, контроллер системных ресурсов                                  |
| SRN    | Service Request Number, номер запроса на обслуживание                                      |
| SSA    | Serial Storage Architecture                                                                |
| SSH    | Secure Shell                                                                               |
| SSL    | Secure Socket Layer                                                                        |
| SUID   | Set User ID                                                                                |
| SVC    | SAN Virtualization Controller, контроллер виртуализации сети хранения данных               |
| TCP/IP | Transmission Control Protocol/Internet Protocol, протокол TCP/IP                           |
| TSA    | Tivoli System Automation                                                                   |
| UDF    | Universal Disk Format, универсальный формат дисков                                         |
| UDID   | Universal Disk Identification, универсальная идентификация диска                           |
| VIPA   | Virtual IP Address, виртуальный IP-адрес                                                   |
| VG     | Volume Group, группа томов                                                                 |
| VGDA   |                                                                                            |