МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

В. Ю. МЕЛЬЦОВ, О. В. КАРАВАЕВА

КЛАСТЕРНАЯ СИСТЕМА ВЯТГУ
HP ENIGMA X000

Учебное пособие

Киров
2019

УДК 004.75

К21

Рекомендовано к изданию методическим советом ВятГУ

Допущено редакционно-издательской комиссией методического совета ВятГУ в качестве учебного пособия для обучающихся по направлениям 09.03.01 «Информатика и вычислительная техника» и 09.03.03 «Прикладная информатика»

Рецензенты:

кандидат технических наук,
доцент кафедры автоматики и телемеханики ВятГУ
В. И. Семеновых

кандидат технических наук, доцент кафедры математических
и естественно-научных дисциплин КФ МФЮА
Т. А. Анисимова

Мельцов В.Ю., Караваева, О. В.

К21 Кластерная система ВятГУ HP ENIGMA X000 / В. Ю. Мельцов, О. В. Караваева. — Киров: ВятГУ, 2019. — 45 с.

УДК 004.75
ББК Ч51.4

Учебное пособие предназначено для обучающихся по направлениям 09.03.01 - Информатика и вычислительная техника и 09.03.03 - Прикладная информатика, всех профилей подготовки, всех форм обучения, для углублённой подготовки по дисциплинам, связанным с изучением архитектуры и организации функционирования ЭВМ и систем, параллельной и распределённой обработкой информации.

Авторская редакция

Тех. редактор Е. В. Кайгородцева

Учебное издание

Мельцов Василий Юрьевич, Караваева Ольга Владимировна

Кластерная система ВятГУ HP ENIGMA X000

Учебное пособие

Авторская редакция
Технический редактор А. Е. Свинина

Подписано к использованию XX.XX.2019. Заказ № XXX
Подписано в печать 16.10.2018. Печать цифровая. Бумага для офисной техники.
Усл. печ. л. 5,06. Тираж 5 экз. Заказ № 5358.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет».

610000, г. Киров, ул. Московская, 36, тел.: (8332) 74-25-63, http://vyatsu.ru

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

За всю историю вычислительной техники не было момента, чтобы уровня развития вычислительной техники было достаточно для решения всех стоящих перед человечеством задач. Постоянно ставятся новые, все более сложные задачи, требующие все более мощных вычислительных ресурсов для своего решения. И современные технологии создания вычислительной техники подошли к рубежу, когда дальнейшее наращивание скорости работы индивидуальных устройств становится практически невозможным. В связи с этим развитие вычислительной техники пошло по экстенсивному пути, основанному на дублировании вычислительных устройств, которые в параллели могут работать над общей задачей. Вместе с этим родилось параллельное программирование, призванное дать возможность эффективно использовать параллельные архитектуры. И сегодня разработчики программных систем используют параллелизм на всех уровнях, начиная от нескольких конвейеров суперскалярных процессоров, и заканчивая параллельно работающими вычислительными узлами в GRID.

Отдельный класс параллельных архитектур представляют кластерные системы. Кластер – это совокупность вычислительных узлов, объединенных сетью. Параллельное приложение для кластерной системы представляет собой несколько процессов, которые общаются друг с другом по сети. Таким образом, если пользователь сумеет эффективно распределить свою задачу между несколькими процессорами на узлах кластера, то он может получить выигрыш в скорости работы, пропорциональный числу процессоров.

Кластеры стали применяться в сфере выскопроизводительных вычислений сравнительно недавно. До конца 80-х практически все суперкомпьютеры представляли собой большой массив соединенных между собой процессоров. Подобные разработки чаще всего были уникальными и имели огромную стоимость не только приобретения, но и

поддержки. Поэтому в 90-х годах все более широкое распространение стали получать кластерные системы, которые в качестве основы используют недорогие однотипные вычислительные узлы.

Основными достоинствами кластерного подхода являются именно дешевизна и легкая расширяемость. Цены на системы кластерного типа стремительно падают, а некоторые модели уже сейчас доступны для одиночных исследователей. К недостаткам же можно отнести сложность создания и отладки эффективных параллельных программ для систем с распределённой памятью. Однако в последнее время развиваются инструменты, облегчающие написание параллельных программ, что способствует все большему распространению кластеров.

Типичное параллельное приложение представляет собой совокупность нескольких процессов, исполняемых на разных вычислительных узлах и взаимодействующих по сети. В принципе, разработчик может полностью взять на себя программирование распределенного приложения и самостоятельно реализовать общение по сети, например, на основе сокетов. Однако в настоящее время существует довольно большое число технологий, упрощающих создание параллельных приложений для кластеров: MPI, PVM, HPF и другие. Эти технологии существуют уже достаточно продолжительное время, за которое они доказали свою состоятельность и легли в основу огромного числа параллельных приложений.

Сегодня кластерные системы крайне востребованы предприятиями и организациями для проведения специализированных параллельных вычислений. Чтобы эффективно использовать мощное вычислительное оборудование, необходимо научиться правильно распределять нагрузку по вычислительным узлам кластера. Данный вопрос приобретает еще большую важность в случае, если кластер имеет неоднородную структуру: различается мощность центральных процессоров, объем оперативной памяти, скорость участков локальной сети. Если не учитывать особенности

аппаратуры, то можно наблюдать, как параллельное приложение простаивает, дожидаясь процесса, который был распределен на самый медленный вычислительный узел. Помимо эффективного планирования запуска задач на кластере, необходимо также автоматизировать процессы приема пользовательских задач, постановки их в очередь, запуска и сбора результатов. Важно обеспечить безопасность использования кластера, его отказоустойчивость, сделав при этом работу с кластером максимально простой.

Кластеры стали фактическим стандартом в области высокопроизводительных вычислений, и можно с большой долей уверенности сказать, что этот подход будет актуален всегда: сколь бы совершенен не был один компьютер, кластер из узлов такого типа справится с любой задачей гораздо быстрее.

1. ЧТО ТАКОЕ КЛАСТЕРНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА?

Кластерная архитектура является одним из способов создания аппаратно объединенного массива параллельно функционирующих ЭВМ. Данный способ организации ЭВМ позволяет создавать не только системы с предельно высокой производительностью, но и отказоустойчивые системы высокой готовности. Кластер представляет собой блок из N вычислительных узлов (вычислительных машин, серверов и т.п.), соединенных коммутирующими устройствами со специальными сетями связи, N копий используемой операционной системы и общую внешнюю память. Кластер для операционной системы и прикладных программ пользователей представляется единым целым.

Первая модель кластера, признанная образцовой, была реализована в 1983 году компанией DEC (Digital Equipment Corporation) на основе VAX компьютеров. Кластер на основе VAX компьютеров представляет собой многомашинную систему, имеющую общую внешнюю память.

При проектировании современных кластеров используется набор конструктивно воплощенных компонентов, поставляемых промышленностью по известной фиксированной цене.

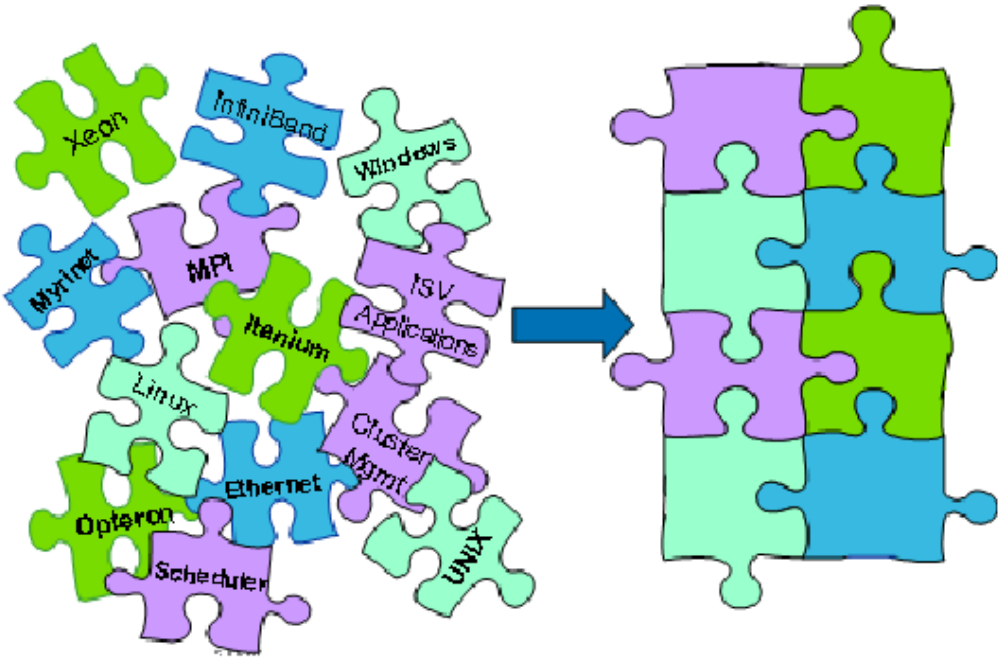


Рисунок 1 – Сборка кластерной системы из стандартных компонентов

Компоненты поставляются в модульном исполнении с соблюдением требований стандарта VME к размерам каркаса, который служит для размещения компонента. В стандартной стойке (шкафу), имеющей размеры по высоте – 6 футов, по ширине – 19 дюймов и по глубине – 30 дюймов, может поместиться 44 модуля, размер каждого из которых соответствует стандарту VME.

Двумя основными проблемами построения вычислительных систем для критически важных приложений, связанных с обработкой транзакций, управлением базами данных и обслуживанием телекоммуникаций, являются обеспечение высокой производительности и продолжительного функционирования систем. Наиболее эффективный способ достижения заданного уровня производительности - применение параллельных масштабируемых архитектур.

1.1 Высокопроизводительные кластерные системы (HPC)

Высокопроизводительные кластерные системы (HPC – High Performance Cluster) используются для задач, которые требуют значительной вычислительной мощности. Классическими областями, в которых используются подобные системы, являются:

– физика;

– газо- и гидродинамика (рис.2а);

– медицина (рис.2б);

– микробиология и фармокология (рис. 3а, 3б);

– биоинформатика и биохимия;

– авиа- и автомобилестроение (рис. 4а, 4б);

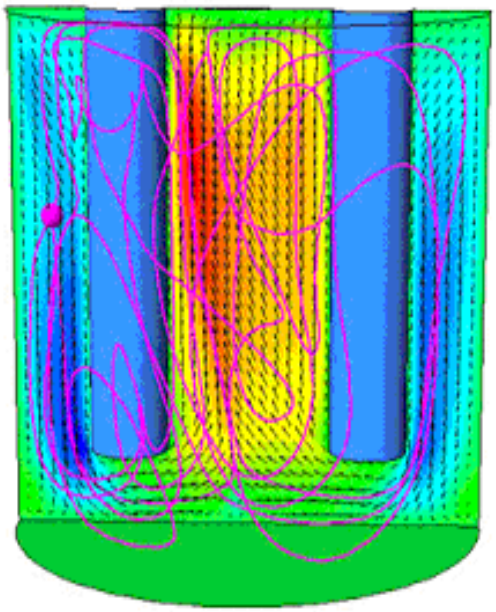
– обработка изображений: рендеринг, распознавание образов;

– геоинформационные задачи;

– нанотехнологии;

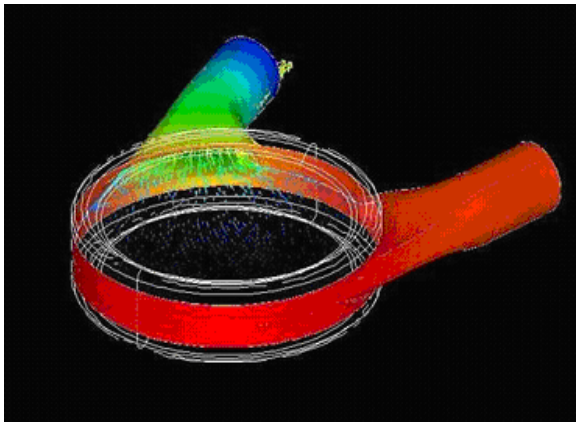
– компьютерная безопасность и т.п.

Магнитогидродинамика



а) Движение жидких смесей

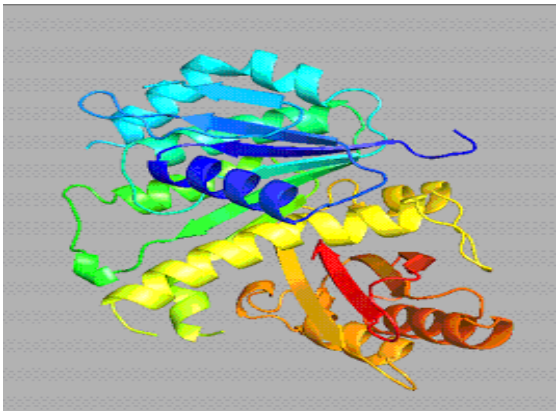
Медицина

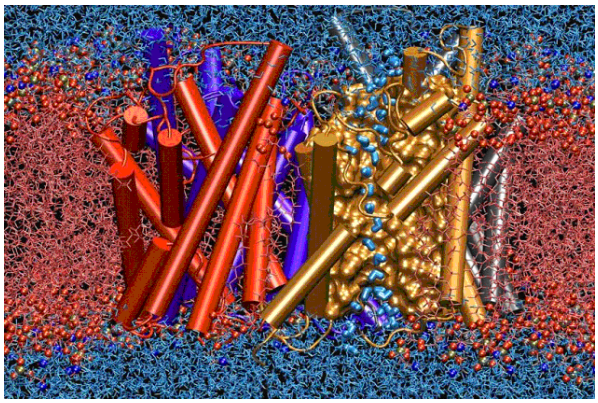


б) Движение крови в сердечном клапане

Рисунок 2 – Задачи моделирования движения жидкости

Микробиология **Фармакология**



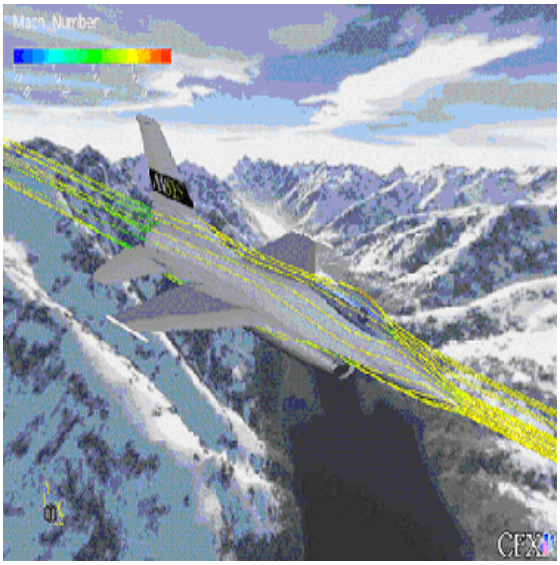


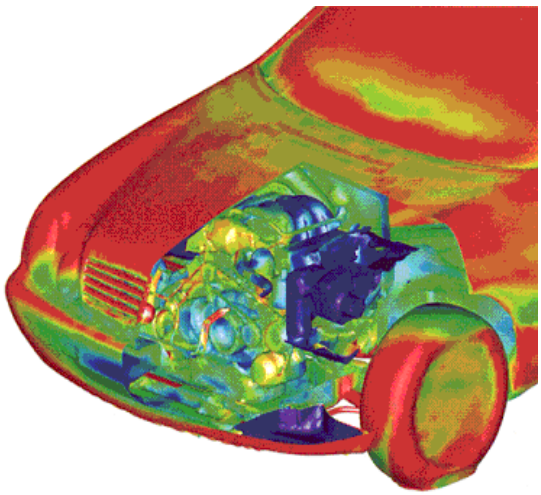
а) Структура кристалла Mycobacterium tuberculosis FtsZ

б) Взаимодействие двух реагентов

Рисунок 3 – Задачи моделирования в микробиологии

Авиастроение **Автомобилестроение**





а) Внешняя аэродинамика самолёта

б) Крэш-тест автомобиля

Рисунок 4 – Задачи моделирования в авиа- и автомобилестроении

Можно отметить следующие примеры пакетов прикладного программного обеспечения, устанавливаемых на кластерные системы:

– *Quantum Pharmaceuticals* (молекулярное моделирование лекарств).

– *AMBER, GROMACS, NAMD* (моделирование динамики биологических молекул).

– *BLAST* (сравнение полученных последовательностей с имеющимися в банке данных, например ДНК).

– *WRF-chem* (моделирование распространения атмосферных примесей и химических реакций).

– *CFX, Fluent* (анализ химической кинетики, горения, теплообмена).

– *ABAQUS* (прочностной анализ).

– *FlowVision* (моделирования трехмерных турбулентных течений жидкости и газа).

– *STAR-CD* – моделирование сложных геометрических задач.

– *LS-DYNA* – анализ высоконелинейных задач механики твердого и жидкого тел.

– *MSC/MARC* – задачи высоконелинейного поведения конструкций и теплопередачи.

– *SolidWorks* – моделирование и проектирование изделий.

– *ASG* – биометрическая аутенификация.

– *Autodeck mental ray* – многоплатформная система визуализации.

– *Maplesoft HPC-GRID* – Maple в режиме параллельных вычислений.

1.2 Кластеры высокой готовности (HAC)

Кластеры высокой готовности (HAC – High Availability Cluster), используются везде, где стоимость возможного простоя превышает стоимость затрат, необходимых для построения HPC, например: биллинговые системы, банковские системы, электронная коммерция, управление предприятием, стратегическое планирование, управление боем.

Задача обеспечения продолжительного функционирования системы имеет три составляющих: надежность, готовность и удобство

обслуживания. Все эти три составляющих предполагают, в первую очередь, борьбу с неисправностями системы, порождаемыми отказами и сбоями в ее работе. Эта борьба ведется по всем трем направлениям, которые взаимосвязаны и применяются совместно (табл. 1).

Таблица 1. Классификация систем высокой готовности

Название класса	**Основные характеристики**
Системы высокой готовности	Некоторые блоки и узлы резервируются. Время простоя от нескольких минут до нескольких часов в год
Системы эластичные к отказам	Используются как аппаратные, так и программные средства повышения готовности. Время простоя от нескольких секунд до нескольких минут в год
Системы устойчивые к отказам	Все блоки и узлы дублируются. Время восстановления менее секунды. Время простоя несколько секунд в год
Системы непрерывной готовности	Готовность системы 100%. Время простоя отсутствует, ремонт проходит в горячем режиме
Системы устойчивые к стихийным бедствиям	1. Локальный уровень 2. Корпоративный уровень 3. Городской уровень 4. Глобальный уровень

Повышение надежности основано на принципе предотвращения неисправностей путем снижения интенсивности отказов и сбоев за счет применения электронных схем и компонентов с высокой и сверхвысокой степенью интеграции, снижения уровня помех, облегченных режимов работы схем, обеспечение тепловых режимов их работы, а также за счет совершенствования методов сборки аппаратуры. Повышение уровня

готовности предполагает подавление в определенных пределах влияния отказов и сбоев на работу системы с помощью средств контроля и коррекции ошибок, а также средств автоматического восстановления вычислительного процесса после проявления неисправности, включая аппаратурную и программную избыточность, на основе которой реализуются различные варианты отказоустойчивых архитектур. Повышение готовности есть способ борьбы за снижение времени простоя системы.

Основные эксплуатационные характеристики системы существенно зависят от удобства ее обслуживания, в частности от ремонтопригодности, контроле пригодности и т.д.

Истинно кластерные системы, такие как VAX-Cluster компании DEC или LifeKeeper Cluster отделения NCR компании AT&T, являются примерами намного более сложного управления по сравнению с простыми процедурами начальной установки при переключении на резерв, и полностью используют все доступные процессоры. Однако организация таких систем влечет за собой и большие накладные расходы, которые увеличиваются с ростом числа узлов в кластере.

1.3 VAX-кластер компании DEC

Компания DEC первой анонсировала концепцию кластерной системы в 1983 году, определив ее как группу объединенных между собой вычислительных машин, представляющих собой единый узел обработки информации. По существу VAX-кластер представляет собой слабосвязанную многомашинную систему с общей внешней памятью, обеспечивающую единый механизм управления и администрирования.

VAX-кластер обладает следующими свойствами:

– *Разделение ресурсов*. Компьютеры VAX в кластере могут разделять доступ к общим ленточным и дисковым накопителям. Все компьютеры в кластере могут обращаться к отдельным файлам данных как к локальным.

– *Высокая готовность*. Если происходит отказ одного из VAX-компьютеров, задания его пользователей автоматически могут быть перенесены на другой компьютер кластера. Если в системе имеется несколько контроллеров HSC и один из них отказывает, другие контроллеры HSC автоматически подхватывают его работу.

– *Высокая пропускная способность*. Ряд прикладных систем могут пользоваться возможностью параллельного выполнения заданий на нескольких компьютерах кластера.

– *Удобство обслуживания системы*. Общие базы данных могут обслуживаться с единственного места. Прикладные программы могут инсталлироваться только однажды на общих дисках кластера и разделяться между всеми компьютерами кластера.

– *Расширяемость*. Увеличение вычислительной мощности кластера достигается подключением к нему дополнительных VAX-компьютеров. Дополнительные накопители на магнитных дисках и магнитных лентах становятся доступными для всех компьютеров, входящих в кластер.

Работа VAX-кластера определяется двумя главными компонентами. Первым компонентом является высокоскоростной механизм связи, а вторым - системное программное обеспечение, которое обеспечивает клиентам прозрачный доступ к системному сервису. Физически связи внутри кластера реализуются с помощью трех различных шинных технологий с различными характеристиками производительности (рис. 5).

Программируемый коммутатор позволяет устанавливать шинную связь между всеми компонентами кластера. При организации внешней памяти кластера обычно используется уровень RAID 1.

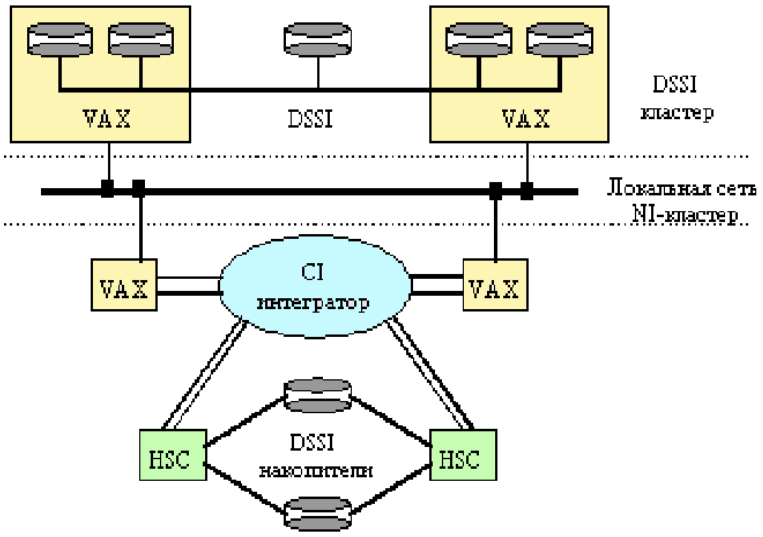


Рисунок 5 – Основные методы связи в VAX-кластере

Основная проблема всех HPC - низкая производительность сети.

Для решения данной проблемы применяют несколько методов.

1) Кластер разделяется на несколько сегментов, в пределах которых узлы соединены высокопроизводительной шиной типа InfiniBand, GB-Ethernet, Myrinet, а связь между узлами разных сегментов осуществляется низкопроизводительными сетями типа Ethernet/Fast Ethernet.

2) Применение так называемого «транкинга», т.е. объединение нескольких каналов Fast Ethernet в один общий скоростной канал, соединяющий несколько коммутаторов. Очевидным недостатком такого подхода является «потеря» части портов, задействованных в межсоединении коммутаторов

3) Создание специальных протоколов обмена информацией по таким сетям – Ultra320 SCSI, Gigabit Ethernet, Fiber Channel, которые позволяют более эффективно использовать пропускную способность каналов и снимают некоторые ограничения, накладываемые стандартными протоколами (TCP/IP, IPX).

Анализ списка самых высокопроизводительных вычислительных систем мира (TOP-500 – www.top500.org) показывает, что наиболее

популярными сетями для высокопроизводительных кластерных и MPP-систем являются InfiniBand и 10G (Ethernet).

1.4 Коммутационная сеть InfiniBand

Архитектура InfiniBand появилась в результате слияния двух параллельных разработок: Next Generation I/O (NGIO) и Future I/O (FIO). Корни NGIO находятся в корпорации Intel, где группа инженеров начала работу над технологией ввода/вывода на основе архитектуры виртуального интерфейса (VI – Virtual Interface). Перед ними стояла двоякая задача. Во-первых, построить последовательный интерфейс, отделенный от связки процессор — оперативная память. Во-вторых, обеспечить посредством этого интерфейса взаимодействие процессов, принадлежащих приложениям, работающим на разных серверах.

Началось все с неудачной попытки просто сериализовать PCI, но вскоре пришло понимание того требуется не просто улучшение, а радикально новое решение. В итоге возник проект NGIO, поддержанный Dell Computer, Hitachi, Intel, NEC, Fujitsu Siemens и Sun Microsystems и еще доброй сотней компаний.

Параллельно с этим еще одной группой компаний (в их число входили IBM, Compaq, Adaptec, 3Com, Cisco и Hewlett-Packard) велась работа над аналогичным проектом FIO. Некоторые производители даже входили в оба лагеря, поскольку проекты NGIO и FIO имели между собой много общего, но были ориентированы на разные применения. NGIO предназначался для рынка стандартных серверов, выпускаемых в больших количествах, а FIO — для платформ более высокого корпоративного класса. Оба проекта заимствовали многое из известных технологий: использовали коммутаторы, аналоги каналов из мэйнфреймов, архитектуру виртуального интерфейса, взаимодействие параллельных процессов (Inter-Process Communication). В начале 1999 года обе спецификации были обнародованы, а уже осенью произошло слияние. В октябре возникла

ассоциация InfiniBand Trade Association. Результаты слияния оказались лучше, чем можно было предположить – получилась не просто сумма, а скорее выборка лучшего из альтернативных компонентов. В октябре 2000 года была опубликована InfiniBand Architecture Specification r1.0, в подготовке которой участвовало 150 специалистов из трех десятков компаний.

Принципиальное отличие InfiniBand от традиционного интерфейса PCI заключается в замене общей шины коммутирующей решеткой (switching fabric). Работой коммутатора управляет менеджер, который обнаруживает физическую топологию подключения, назначает локальные идентификаторы и осуществляет маршрутизацию между узлами. Он же управляет и изменениями, такими, например, как добавление или отключение узлов. Главное же в том, что архитектура InfiniBand обеспечивает организацию передачи данных между сервером и периферией не по общей шине, а по выделенным каналам. Это обеспечивает ей совершенно новое качество. Вместо жестко организованного доступа посредством DMA в фиксированные области разделяемой памяти, здесь используется надежный механизм передачи сообщений, который «отвязывает» сервер от периферии. Такое решение открывает неограниченную возможность кластеризации и масштабирования. Основными преимуществами InfiniBand по сравнению с PCI являются:

– *Надежность.* Меньшее число контактов (в минимальной версии их всего четыре) уменьшает вероятность неисправности.

– *Готовность.* Как любая сетевая структура, архитектура InfiniBand допускает организацию альтернативных маршрутов; поэтому выход из строя канала не приводит к выходу из строя всей системы.

– *Обслуживаемость.* Устройства, предназначенные для использования в среде InfiniBand, проектируются в расчете на горячую замену.

В итоге архитектуры, построенные на основе InfiniBand, отличаются большей гибкостью, масштабируемостью и лучшими показателями RAS.

1.5 «Лезвия», серверы и не только

С появлением центров обработки данных критически важными характеристиками стали занимаемые помещения и потребляемая энергия. Возникло стремление размещать как можно больше серверов в пересчете на единицу объема или площади. В итоге появились тонкие и ультратонкие серверы и серверные приставки, толщина которых обычно измеряется в «юнитах» (например, 1U, 2U или 4U; U = 1,75 дюйма). По существу, эти серверы — миниатюрные исполнения обычного стоечного сервера, полностью повторяющие его архитектуру; все они имеют собственную дисковую память, поэтому их называют серверами, имеющими состояние (stateful). При упаковке их в стойку получаются довольно необычные сооружения, содержащие десятки процессоров и сотни жестких дисков, они сложны в управлении, выделяют много тепла. Инженерный опыт подсказывает, использование устройств без собственной внешней памяти, не обладающих хранимым состоянием (stateless), может оказаться эффективнее с точки зрения централизации управления и разделяемого использования общих ресурсов (различные типы внешней памяти и другие устройства ввода/вывода). Первыми образцами серверов без собственной внешней памяти стали «лезвия» (blade). Появились, но еще не получили широкого распространения серверы в формате «кирпича» (brick). Подобный сервер представляет собой небольшую плату, содержащую только процессор и оперативную память. В стандартном шасси высотой 2U удается разместить 16 или даже 24 подобных серверов. В редких случаях на лезвиях размещают и диски; такова, например, продукция компании RLX Technologies, но это исключение, подтверждающее общее правило.

Назвать бездисковое лезвие сервером в полном смысле этого слова нельзя, поскольку в отличие от ультратонких серверов он не обладает автономной функциональностью. Законченным устройством может быть сборка таких серверов, включающая системы питания, охлаждения, диагностики, но, прежде всего, обладающая средствами доступа к внешней памяти. Насколько привлекательной ни была бы идея выделения вычислительных компонентов в легкозаменяемые серверы, не имеющие состояния, она останется непродуктивной без соответствующих средств доступа к внешней памяти. Рассматривались возможные решения этой задачи, в том числе, шинные варианты VME и Compact PCI, а также Gigabit Ethernet. Два первых вскоре отпали, поскольку не обеспечивали достаточной производительности и имели ограничения по масштабированию и надежности. В последнем случае может быть использован стандарт 10 Gigabit Ethernet с аппаратной реализацией механизма TCP Offload Engine, но на рынке технология этого типа появится не ранее конца 2003 года. В итоге прекрасная по замыслу идея серверов-лезвий, возможно, оставалась бы на уровне бумажных проектов или мелкосерийных изделий, если бы не появление архитектуры InfiniBand, которая, как оказалось, на 100% подходит для создания функционально полных систем из простых лезвий.

Архитектура InfiniBand позволяет построить вычислительную систему на принципах System Area Network. Такое устройство действительно очень красиво. К сожалению, когда сейчас говорят и пишут о подобном подходе к построению вычислительной машины, почему-то забывают, что один из самых надежных в мире компьютеров Tandem строился именно по такому принципу. Разумеется, в нем были собственные уникальные средства для построения системной сети, однако смысл объединения лезвий остается тем же самым — создать сетевой кластер из однородных вычислительных модулей.

2. КЛАСТЕРНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА ВЯТГУ

В 2006 году в рамках Приоритетного национального проекта “Образование” Вятский государственный университет выиграл грант на выполнение инновационной программы «Научно-образовательный центр биотехнологии, аэробиологии, общей и промышленной микробиологии» (Объём гранта -242 млн. рублей)

Программа ВятГУ состояла из трёх проектов.

Проект 1: Создание Центра компетенций в области биотехнологии, аэробиологии, общей и промышленной микробиологии.

Проект 2: Создание инфраструктурной платформы физико-химического анализа функционирования Центра компетенций.

Проект 3: Создание аппаратно-программной IT-платформы функционирования Центра компетенций. (*Объем финансирования 104 млн. рублей).*

Одной из приоритетных задач Проекта 3 и всей инновационной программы ВятГУ было создание научно-образовательного Центра «Супервычислительных Технологий и Систем» для выполнения фундаментальных и прикладных исследований по приоритетным направлениям развития науки, техники и критическим технологиям.

Основная часть выделенных по гранту средств была потрачена на закупку оборудования и программного обеспечения создаваемого в ВятГУ вычислительного центра на базе Кластерной вычислительной системы **HP ENIGMA X000.** Конкурс на поставку аппаратных комплектующих и специализированного ПО выиграла фирма Hewlett-Packard (HP) — одна из крупнейших компаний в сфере информационных технологий, мировой лидер по количеству разработанных Супер-ЭВМ, попавшим в список TOP-500 (список 500 самых высокопроизводительных вычислительных систем мира).

Главным «строительным элементом» кластера ВятГУ служат блэйд-серверы **HP ProLiant c-Class.**

Семейство продуктов HP ProLiant включает в себя серверы, построенные на базе процессоров производства компаний AMD и Intel, использующих архитектуру x86 и последние инновации в ее развитии: 64-битность и многоядерность.

В серверах данного семейства применяются следующие технологические решения:

– *ECC* (Error Correction Code) — механизм обнаружения и коррекции однобитных ошибок в микросхеме памяти.

– *Advanced ECC* — механизм обнаружения и коррекции, обеспечивающий защиту от выхода из строя одной микросхемы памяти в модуле DIMM.

– *Горячее резервирование* (Online Spare) — переключение с потенциально сбойного модуля памяти на резервный (spare) без прерывания функционирования. Происходит при превышении порогового значения ошибок в модуле памяти. Требуется избыточный максимальный модуль памяти.

– *Зеркалирование данных на одной плате* — обеспечивает защиту от выхода из строя DIMM-модулей памяти. Замена сбойного DIMM происходит с отключением системы. Требуется избыточный объем памяти, равный основному.

– *Зеркалирование данных с горячей заменой/добавлением на 2 платах* (Hot Plug Mirrored) — обеспечивает защиту от выхода из строя DIMM-модулей памяти. Замена сбойного DIMM происходит без отключения системы. При поддержке операционной системой (например, Microsoft® Windows® 2003) возможно добавление памяти без остановки работающих приложений. Требуется избыточный объем памяти, равный основному.

– *RAID-память с горячей заменой* (hot-plug RAID Memory) инновационная технология, обеспечивающая защиту от выхода из строя DIMM-модулей памяти и восстановление без отключения системы. Модули памяти устанавливаются в четыре независимых картриджа,

последний из которых содержит информацию о четности в предыдущих. Любой из картриджей может быть извлечен из системы в горячем режиме для замены и добавления модулей. Требуется избыточный объем памяти, равный 33% от основного.

– *Механизм ускорения доступа к памяти* (interleaving) предполагает размещение последовательного набора данных в разных банках. В этом случае активная фаза цикла обращения к памяти одного банка чередуется с фазой восстановления другого, что позволяет сократить время цикла почти в два раза. В серверах HP ProLiant допускается использовать совместно память как с чередованием, так и без него:

– механизмы универсальных жёстких дисков с возможностью горячей замены;

– механизм встроенного удалённого управления (Integrated Lights-Out).

2.1 Серверы семейства HP BladeSystem c-Class

Сверхплотные серверы HP BladeSystem c-Class основываются на базе процессоров архитектуры x86. Архитектура HP BladeSystem c-class с самого начала разрабатывалась без привязки к существующим уже в течение многих лет конструктивам и форм-факторам блейд-серверов HP BladeSystem p-класса, что позволило в максимальной степени реализовать в ней новые идеи и новые технологии для центров обработки данных. Применительно к HP BladeSystem c-class компания Hewlett-Packard разработала и предлагает пользователям фирменные технологии для решения задач, связанных с повышением гибкости и управляемости серверной инфраструктуры, понижением энергопотребления и тепловыделения, сокращением стоимости закупки и владения ИТ-инфраструктурой:

– Технология Virtual Connect виртуализирует адаптеры ввода-вывода блейд-серверов, что позволяет заменять, добавлять и переразвертывать

серверы без какого-либо влияния на взаимодействующие с серверной инфраструктурой домены сетей SAN и LAN.

– Технология снижения общего энергопотребления и тепловыделения HP Thermal Logic основана на использовании возможностей регулирования энергопотребления процессоров, интеллектуальных систем охлаждения и обдува компонент в серверной полке.

– Обновленный набор средств мониторинга и интеллектуального управления для блейд-систем Hewlett-Packard стал более интегрированным, функциональным и получил новое название HP Insight Control.

– Экономия затрат на этапе приобретения инфраструктуры достигается за счет дизайна блейд-системы как применительно к серверам, так и, особенно, с точки зрения стоимости подключений к внешним сетям (LAN и SAN).

Важным фактором при использовании блейд-систем также является существенное повышение плотности компоновки серверной инфраструктуры. Так, HP BladeSystem c-class позволяет установить в одно шасси высотой 10U 16 двухпроцессорных или 8 четырехпроцессорных серверов, со всей необходимой инфраструктурой питания и охлаждения, плюс 8 встроенных коммутационных модулей – коммутаторов SAN, Gigabit Ethernet, Infiniband. Важно отметить, что плотность компоновки ничуть не сказывается на характеристиках серверов: даже самый компактный сервер оснащается двумя самыми современными процессорами от Intel® или AMD, расширенным объёмом оперативной памяти, двумя дисками SAS с возможностью горячей замены, и поддерживает до 6 внешних интерфейсов для подключения к интегрированным коммутационным модулям. Во всех серверах используются последние серверные технологии ProLiant, такие как обновленная подсистема оперативной памяти, новые жесткие диски SAS SFF, новый процессор удаленного управления iLO2,

многофункциональные сетевые адаптеры с аппаратной поддержкой iSCSI, TOE, RDMA. Для обеспечения интеграции HP BladeSystem в инфраструктуру центра обработки данных и обеспечения взаимодействия серверов между собой задействуются коммутационные модули — до 8 на серверную полку. В качестве коммутационных модулей могут выступать как патч-панели, выводящие сигналы от серверных адаптеров в соотношении «1-к-1», так и интеллектуальные коммутирующие устройства — коммутаторы Gigabit Ethernet производства Cisco или Nortel, Fiber Channel, Infiniband, а также модули виртуализации ввода/вывода – Virtual Connect GigE и FC.

Использование блейд-инфраструктуры HP BladeSystem подразумевает по умолчанию высокий уровень избыточности компонент. Так, питание блейд-системы зарезервировано по схеме «3+3», используются избыточные вентиляторы системы охлаждения, все коммутационные модули устанавливаются парами, все используемые серверные адаптеры ввода-вывода — двухпортовые. Как следствие, для блейд-системы характерен более высокий уровень доступности серверов и приложений, как правило, за более приемлемую цену по сравнению с аналогичными решениями для стоечных серверов.

Для эффективного управления HP BladeSystem в состав полки включены средства, позволяющие визуализировать серверные полки и получить наглядное представление о состоянии и конфигурации блейд-системы, контролировать параметры окружающей среды, настраивать подсистемы охлаждения и электропитания. Для управления блейд-инфраструктурой на всех этапах жизненного цикла используется интегрированный пакет программного обеспечения HP Insight Control, в состав которого входит Systems Insight Manager, а также дополнительные средства управления: Rapid Deployment Pack, Vulnerability and Patch Management Pack, Performance Management Pack.

Применение средств Insight Control, интегрирующих Rapid Deployment Pack с системами мониторинга работы серверной инфраструктуры, позволяет автоматизировать процессы восстановления в случае аппаратных и программных сбоев путем переноса функций сбойного сервера на сервер из пула запасных, обеспечив, таким образом, более высокий уровень доступности приложений.

2.2 Блэйд-серверы HP ProLiant BL460c и HP ProLiant DL360

В вычислительных узлах кластера размещены блэйд-серверы HP ProLiant BL460c и HP ProLiant DL360.

Линейка BL (Blade Line) - HP Blade System серверы с максимальной плотностью монтажа компонентов, не имеющие собственных разъемов ввода-вывода и предполагающие постановку в специальные корпуса-полки, позволяющие строить серверные инфраструктуры сверхвысокой плотности с пониженным потреблением энергии. Серверы Proliant BL460c, выпускаемые с двух-, четырёх- и шестиядерными процессорами Intel Xeon, модулями памяти DIMM с полной буферизацией DDR3, жёсткими дисками SAS или SATA, поддержкой многофункциональных контроллеров сетевого интерфейса и несколькими картами ввода/вывода, представляют собой высокопроизводительные системы, идеально подходящие для работы с полным спектром масштабируемых приложений. В компактном сервере Proliant BL460c имеется больше возможностей для обеспечения высокой доступности, таких как жесткие диски с горячим подключением, зеркалированная память, дополнительное резервирование памяти в режиме онлайн, память с чередованием адресов, встроенная возможность построения RAID, а также улучшенное дистанционное управление Lights-Out.

Блэйд-сервер HP ProLiant BL460c для кластерной системы ВятГУ – это сервер с двумя четырёхядерными процессорами, не уступающий по функциям стандартным монтируемым в стойки 1U серверам, сочетает

огромную вычислительную мощность и высокую плотность монтажа с расширенной памятью и максимальной производительностью ввода/вывода. Корпус BladeSystem c7000 поддерживает до 16 блейд-серверов Proliant BL460c, на 2 сервера больше, чем IBM BladeCenter, при этом каждый сервер Proliant BL460c даже без платы расширения поддерживает вдвое больший объем памяти, чем сервер HS21, имеющий такую же плату.

Линейка DL (Density Line). HP Blade System компактные серверы в стоечном исполнении с повышенной плотностью монтажа и максимально интегрированными компонентами. Ориентированы на работу с внешними системами хранения данных.

Блэйд-сервер HP ProLiant DL360 для кластерной системы ВятГУ – это серьёзная вычислительная мощь, сконцентрированная в корпусе высотой 1U, с технологией дистанционного управления Integrated Lights-Out (рис. 6).

Рисунок 6 – Внешний вид блэйд-сервера HP ProLiant DL360

Серверы DL360 обеспечивают высокую отказоустойчивость и подходят для установки в ограниченном пространстве. Благодаря четырёхъядерным процессорам Xeon, памяти DDR3 с полной буферизацией DIMM и технологиям Serial Attached SCSI (SAS) и PCI Express, данные серверы обеспечивают высокую производительность и идеально подходят для всех масштабируемых приложений. Кроме того, сервер DL360 G5 гарантирует повышенную отказоустойчивость на платформе со сверхвысокой плотностью размещения благодаря резервному блоку питания, резервным вентиляторам, зеркалированной памяти или резервному банку памяти, встроенной технологии RAID и комплексной системе дистанционного управления Lights-Out (рис. 7).

Рисунок 7 – Внутренние компоненты блэйд-сервера HP ProLiant DL360

Основные технические характеристики блэйд-сервера HP ProLiant DL360:

- Тип процессора – 2xQuad-Core Intel Xeon@ 5345 EM64T.
- Тактовая частота – 2,33ГГц.

- Число ядер – 8 (2x4 в каждом процессоре).
- Объем кэш - 8 МБ.
- Частота шины – 1333 МГц.
- Чипсет - Intel@ 5000P.
- Сетевые адаптеры – два 1 Гб NC373i + один 10/100.
- Оперативная память – 2xHP 2GB FBD PC2-5300.
- Дисковый накопитель – 2xHP 10K SAS 2,5 Hot Plug Hard Drive 36 GB.
- Корпус – блейд ½ для шасси HP Blade-System c7000.

Основное системное программное обеспечение блэйд-сервера HP ProLiant DL360:

- Операционная система 1.1 RedHat Enterprise Linux Advanced Server 4 update 4 (EM-64T)
- Параллельное окружение (MPI) (mpiapich-0/0/9, mpich-ch_p4 v.1.2.7)
- Средства разработки (Intel C/C++/Fortran 9.1/9.0, C/C++/Fortran GNU GCC v.4.1.1, Java JDK 1.5.0)
- HP OpenView Operations, HP OpenView Networks Node Menager
- HP OpenView Servise Desk, HP TeMIP
- Система управления кластерами OSCAR v.5 (Oscar Wizard, system Installlation Suite, kernel-picker,PXE, tftpboot, Netbootmgr, SystemImager, Systemconfigurator, Ssysteminstaller-oscsr,MySQL)

В целом, Кластерная система ВятГУ HP ENIGMA X000 (рис. 8) содержит 288 узлов, два из которых являются управляющими. Именно на них операционная система RedHat Enterprise Linux развёрнута полностью. Основные характеристики вычислительного кластера:

- Производительность – до 20 TFlops.
- Емкость электронного хранилища данных – 50 TB.

– Возможность подключения дополнительных ресурсов за счет использования технологии GRID.

– Высокоскоростное подключение к точке национального пиринга M9X.

Рисунок 8 – Кластерная система ВятГУ HP ENIGMA X000

Центр супервычислительных технологий и систем ВятГУ обеспечивает возможность выполнять на мировом уровне проекты по высокоточному математическому моделированию сложных объектов и процессов, системному анализу и прогнозированию поведения сложных многопараметрических систем

3. ПОДКЛЮЧЕНИЕ И РАБОТА НА КЛАСТЕРЕ HP ENIGMA X000

Для выполнения на кластерной системе ВятГУ практического задания вам доступны 3-4 блейд-сервера, содержащие по два 4-ядерных процессора. Для подключения к HPC HP ENIGMA X000 и запуска на ней пользовательской программы необходимо выполнить следующую последовательность действий.

1. Подключиться к удаленному рабочему столу: нажать комбинацию клавиш WIN+R, ввести «mstsc /v:10.128.1.168» и нажать OK (рис. 9).

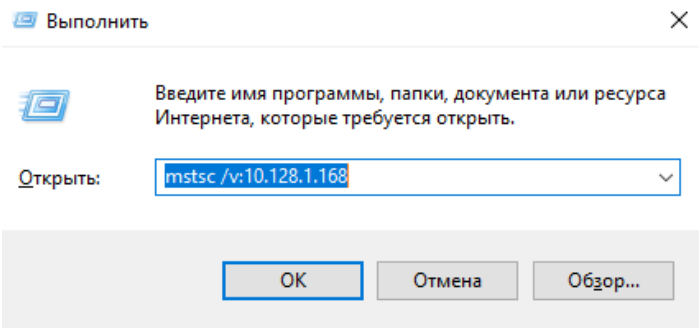


Рисунок 9 – Подключение к удаленному рабочему столу

2. В строке «Компьютер» окна «Подключение к удаленному рабочему столу» задать IP **10.128.1.168** (рис. 10).

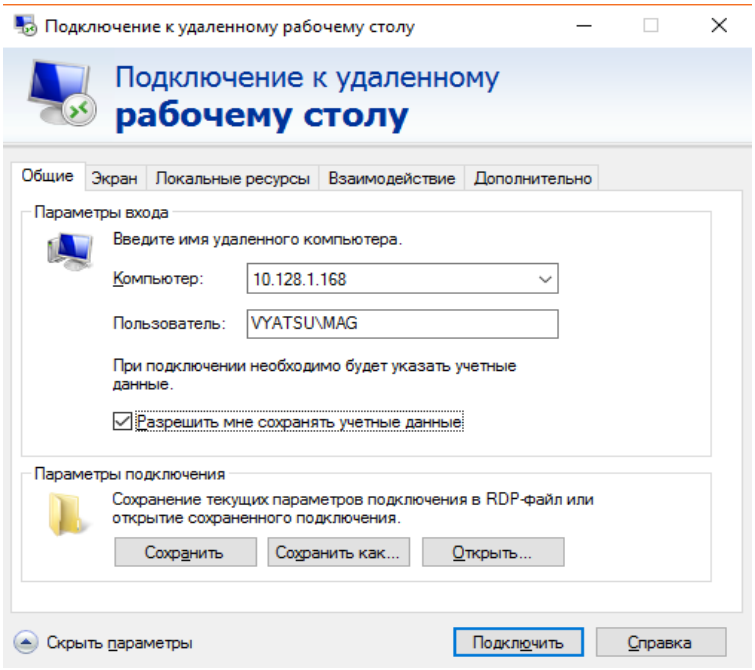


Рисунок 10 – Ввод IP-адреса

Примечание: также можно задать другие IP: 10.128.1.167 10.128.1.169 *(*10.128.1.166 – *подключается по дополнительному запросу*)

3. В открывшемся окне «Удаленный рабочий стол» выбрать пользователя **VYATSU\mag** (рис.11). Ввести пароль: **Pa$$w0rd** (*0* - *нуль*).

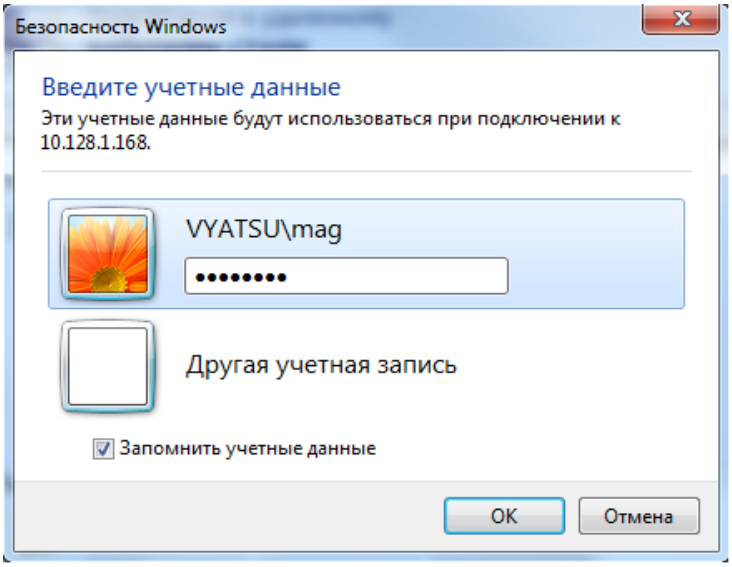


Рисунок 11 – Ввод IP-адреса

4. Установить (подтвердить) соединение, игнорируя ошибки сертификата (рис.12).

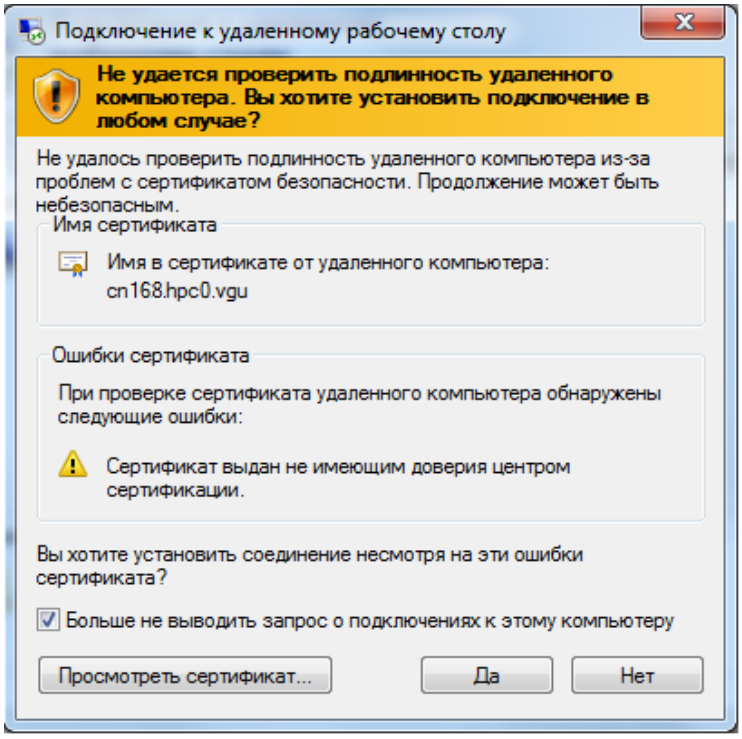


Рисунок 12 – Установить соединение

5. При верном вводе откроется удаленный рабочий стол (рис.13).

Рисунок 13 – Удаленный рабочий стол

6. Открыть проводник (**ПУСК > Компьютер**). Диск **(Z)** является общим ресурсом и доступен всем трем блейдам, которые вы будете использовать (рис.14). Скопировать (CTRL+C) необходимые файлы с вашего компьютера и вставить их (CTRL+V) в корень диска **(Z)** (рис.15).

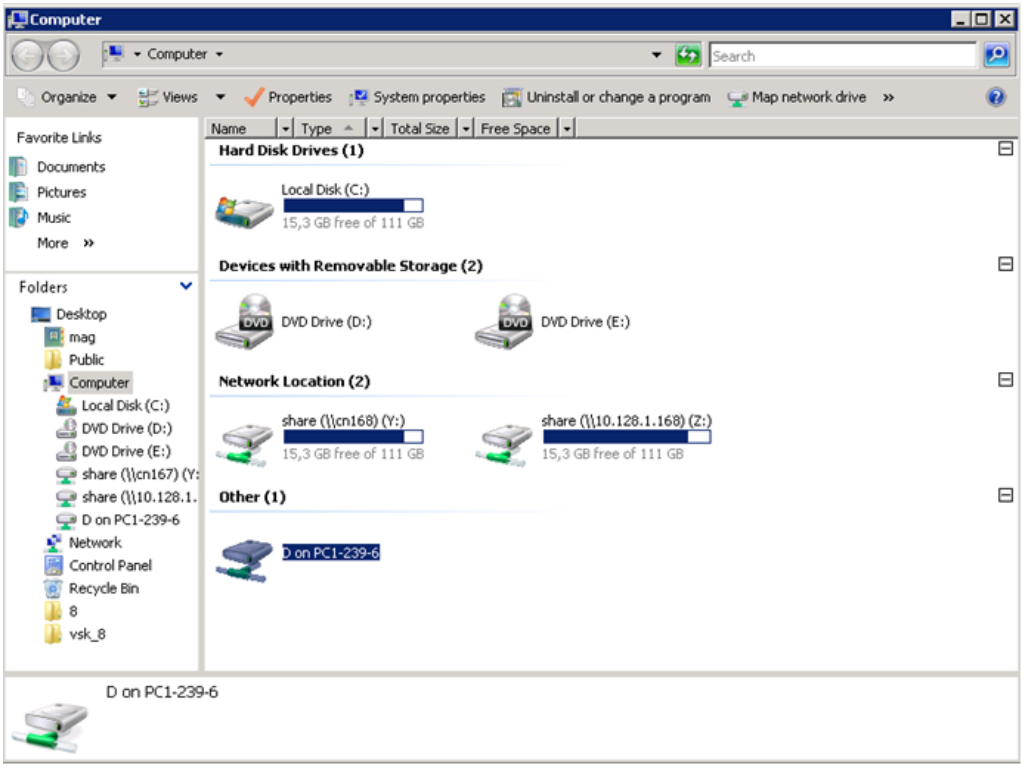


Рисунок 14 – Проводник

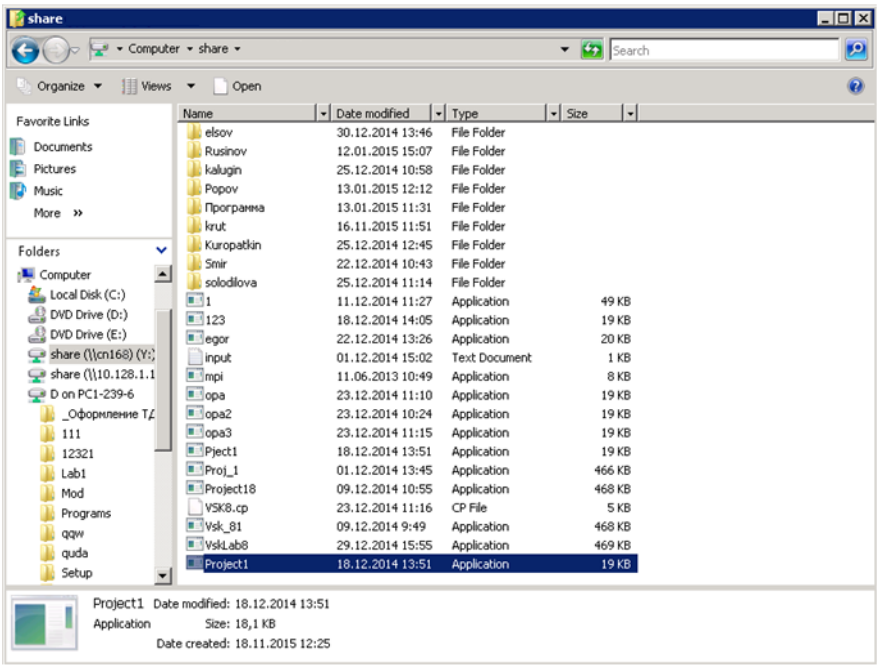


Рисунок 15 – Копирование выполняемых файлов

7. Перейти в ПУСК > MPICH2 и запустить **wmpiconfig.exe** (рис.16).

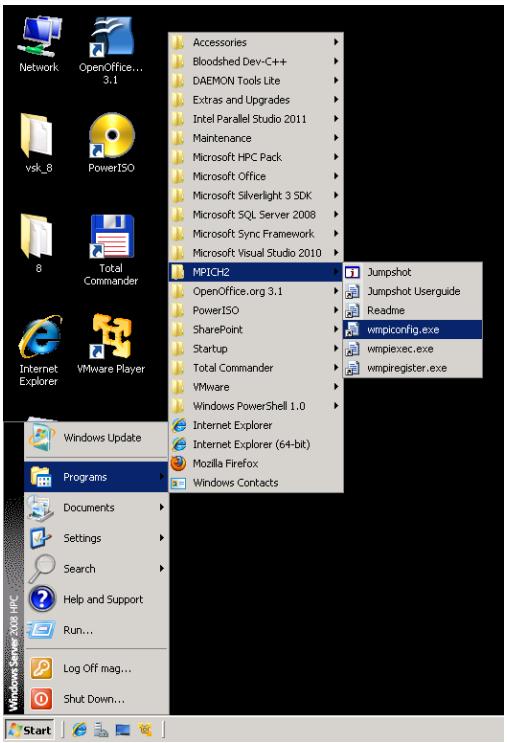


Рисунок 16 – Запуск wmpiconfig.exe

8. Должна запуститься утилита конфигурации MPICH2. Необходимо проверить работоспособность всех трех блейд-серверов, требующихся вам

для работы. В строке «HOST» последовательно вводим названия блейдов **cn167, cn168, cn169** и нажимаем «Get Settings». Если слева этот блейд подсвечен зеленым, то на нём можно запускать приложение , если серым – то блейд недоступен. Необходимо чтобы все три блейд-сервера были доступны, т.е. были подсвечены зеленым (рис.17).

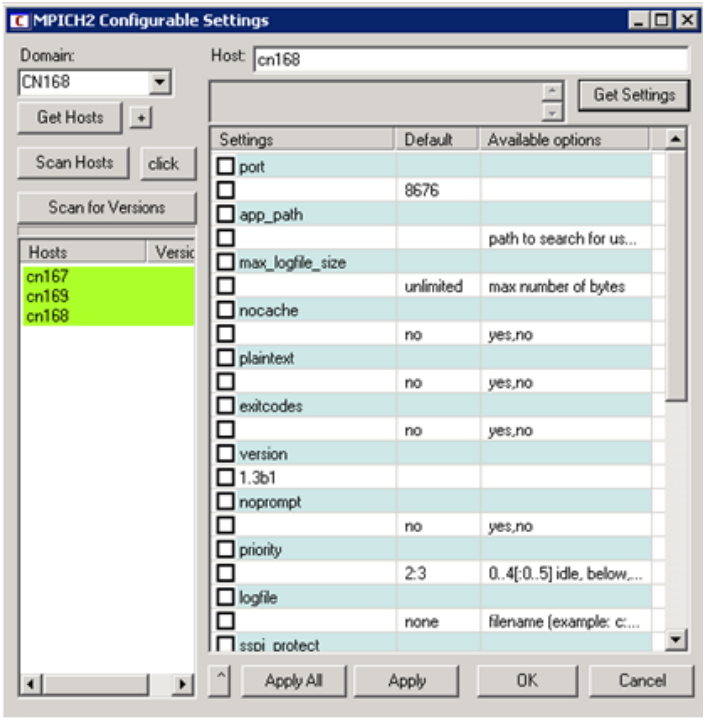


Рисунок 17 – Проверка работоспособности блэйдов

9. Перейти в меню **ПУСК > MPICH2** и запустить **wmpiexec.exe**. Это программа для выполнения вашего приложения на указанных блейдах кластерной системы (рис. 18).

10. Необходимо выбрать ваш исполняемый файл, размещённый на «общем» диске **(Z)**. Если вы выберете файл из другого диска, то программа запустится только на одном блейде, на том, к которому вы подключились.

11. Если поставить отметку («галочку») **run in an separate window**, то программа будет запускаться (выводить результаты работы) в отдельном окне (рис. 19). Если отметку не ставить, то программа будет выводить сообщения в область, расположенную по центру окна программы **wmpiexec.exe**.

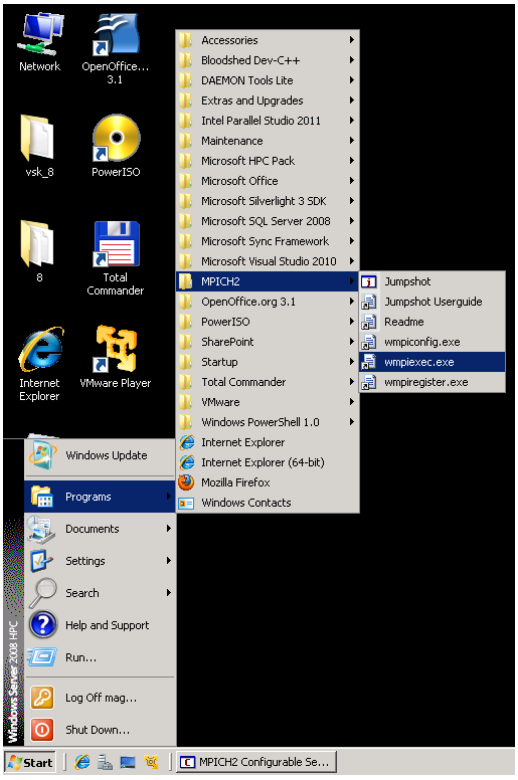


Рисунок 18 – Запуск wmpiconfig.exe

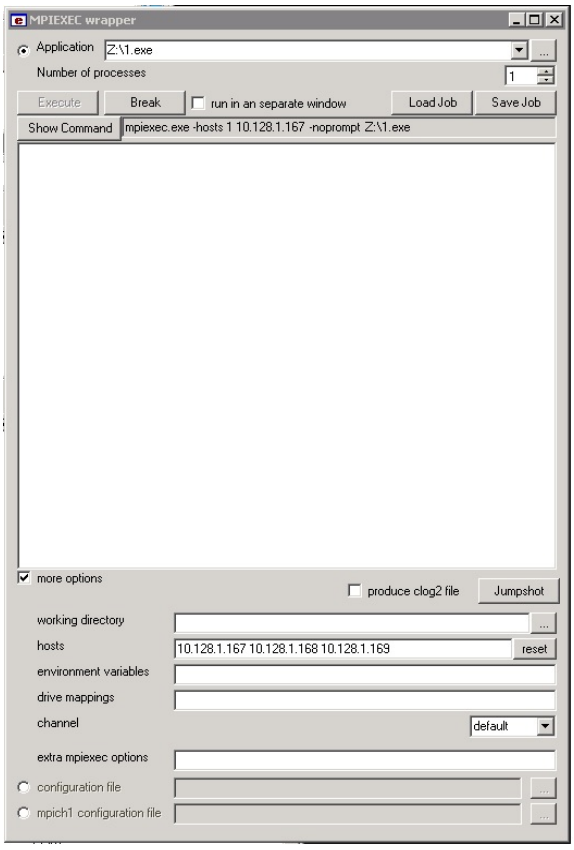


Рисунок 19 – Запуск программы в отдельном окне

12. В окне **wmpiexec.exe** можно изменять количество запускаемых процессов счётчиком **Number of processes**. По умолчанию каждый процесс запускается на отдельном ядре, если количество доступных ядер больше или равно требуемому количеству процессов. Один блейд-сервер HP ProLiant 360C содержит два четырехъядерных процессора, поэтому для параллельного выполнения более чем 8 процессов требуется подключение дополнительных блейдов. На рис. 20 показаны результаты выполнения программы пользователя на одном ядре («ноде»), то есть «параллельно обрабатывается 1 процесс».

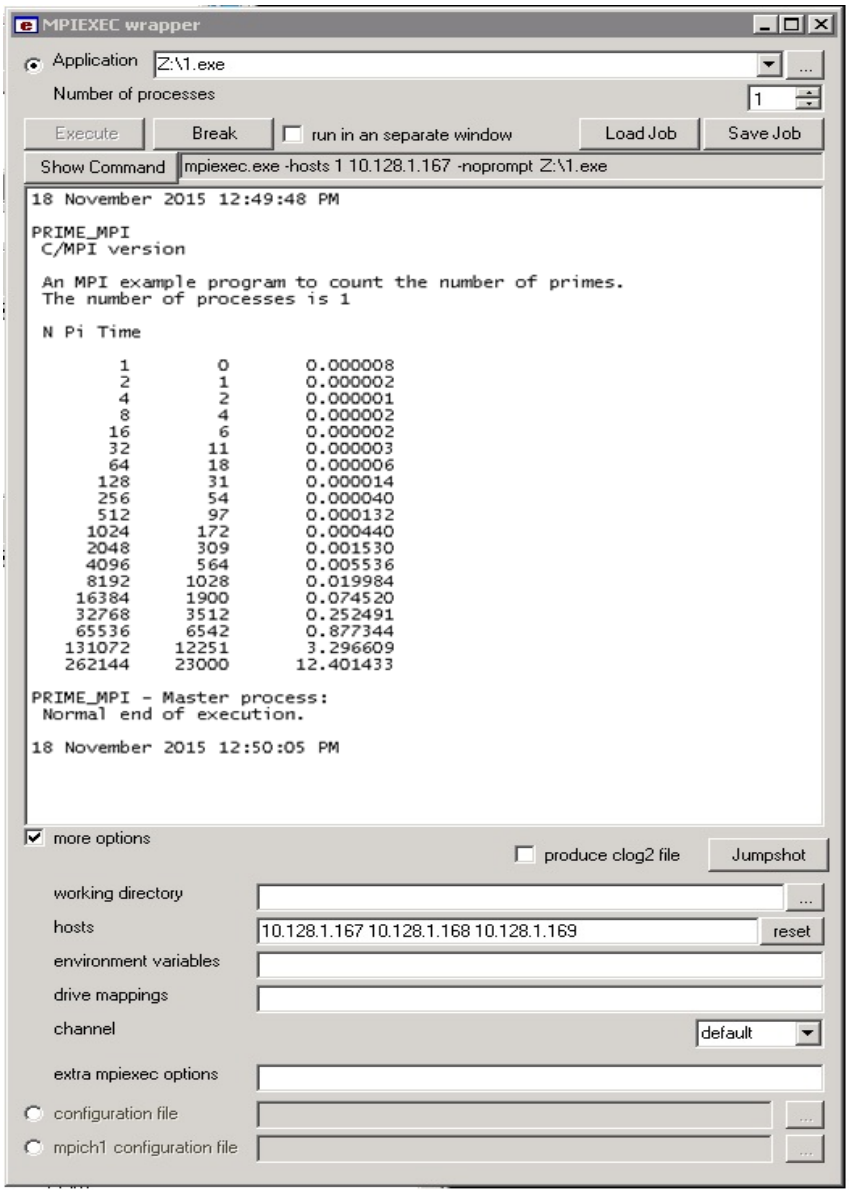


Рисунок 20 – Пример выполнения программы на одном ядре (n=1)

На рис. 21 показаны результаты выполнения программы пользователя на шестнадцати ядрах, то есть «параллельно обрабатываются 16 процессов». В ходе исследований на трёх блэйд-серверах можно увеличивать количество ядер до 24.

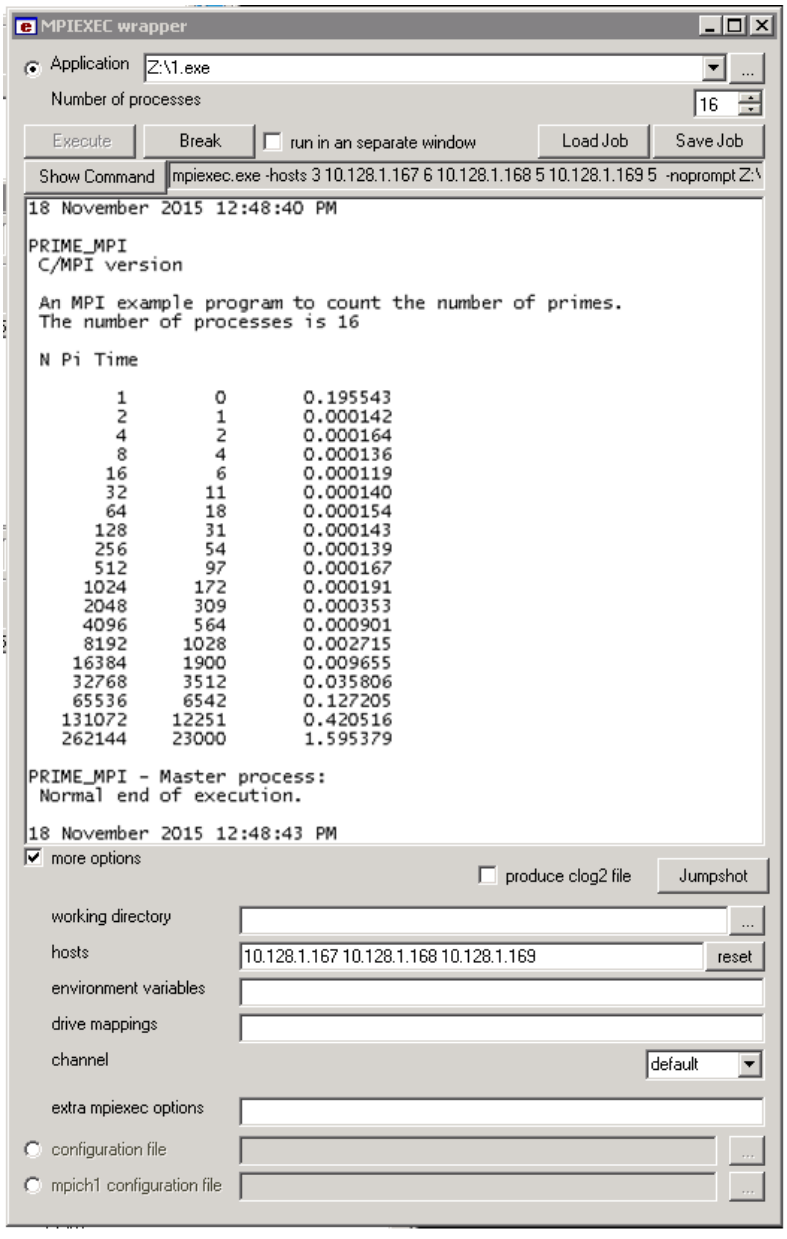


Рисунок 21 – Пример выполнения программы на шестнадцати ядре (n=16)

13. Для увеличения количества используемых блейдов необходимо в строке **hosts** через пробел записать IP этих блейдов: **10.128.1.167 10.128.1.168 10.128.1.169**. Два блейд-сервера эффективно использовать, когда при запуске программы будет организовано от 9 до 16 параллельно работающих процессов. Три блейд-сервера эффективно использовать для запуска от 17 до 24 процессов. Если запускаемая программа была расположена не на диске **(Z)**, будет выведено сообщение об ошибке, так как только диск **(Z)** доступен всем блейдам.

14. Для получения корректного времени выполнения вашей программы необходимо осуществить несколько запусков программы с одними и теми же исходными данными и параметрами запуска, так время выполнения может меняться из-за различных задержек на передачу сообщений между процессорами и блэйдами, а также из-за возможного использования одних и тех же ресурсов несколькими студентами одновременно. Для отчёта лучше использовать среднее время выполнения программы на данном количестве ядер и размерности данных.

4. ПРИМЕРЫ ЗАДАЧ ДЛЯ КЛАСТЕРНОЙ СИСТЕМЫ

Итак, практическое задание состоит в следующем.

1. Вам необходимо написать программу с использованием MPI (MPICH).

2. Многократно запустить программу на кластере с различным количеством ядер и при различных размерностях исходных данных.

3. По результатам экспериментов построить графики зависимостей времени от количества используемых ядер и от размерности данных: размерности матрицы, количества элементов массива, количества итераций при вычислении рядов и т.д.

4. Внимательно проанализировать графики и сделать выводы об эффективности выполнения программы на различных конфигурациях кластерной системы. Объяснить возможные спады производительности и возможное наличие этапов суперлинейного ускорения.

В качестве примеров практических заданий можно выделить такие алгоритмы: умножение матриц и векторов, транспонирование матриц, вычисление рядов или интегралов, параллельная сортировка массива и т.д.

Если ваша программа должна выводить время выполнения вычислений, то для этого, например, может быть использован следующий код:

```
        {
  double starttime, endtime;
  starttime = MPI_Wtime();   // запоминаем время начала вычислений

  ......  XXX  ……..              // ваша программа – основные операции

  endtime = MPI_Wtime();  // запоминаем время окончания вычислений
  printf("Work time %f sec\n", endtime-starttime);  // вывод времени
                                                    // выполнения как разницы
                                                    // времени окончания и
                                                    // времени старта программы
   }
```

Вариант 1. В табл. 2 приведены результаты работы программы, предназначенной для нахождения значения функции косинуса, полученные с использованием приближённых вычислений с помощью рядов. Время определялось как среднее арифметическое из 30 запусков при одних и тех же исходных параметрах. Количество ядер увеличивалось от 2 до 16.

Таблица 2 – Время вычислений функции косинуса

Количество итераций	Количество используемых ядер						
	2	4	5	8	9	15	16
20000	18,34207	9,45534	7,42576	4,92309	4,92113	2,90112	2,36866
25000	28,59231	14,81608	11,64575	7,57906	7,45099	4,70747	3,78564
30000	41,16907	21,20340	16,72628	10,82907	10,88108	5,71894	5,42498
35000	56,3366	28,80747	22,82034	14,44789	14,74146	9,07164	7,19415

При изменении количества ядер до 32 интересными следует считать случаи, когда добавляется ядро процессора из нового блейда, то есть следует обратить внимание на разницу времени вычислений при 8 и 9 ядрах, при 16 и 17 ядрах.

На рис. 22 представлена зависимость времени выполнения программы от количества используемых ядер. Количество итераций оставалось неизменным.

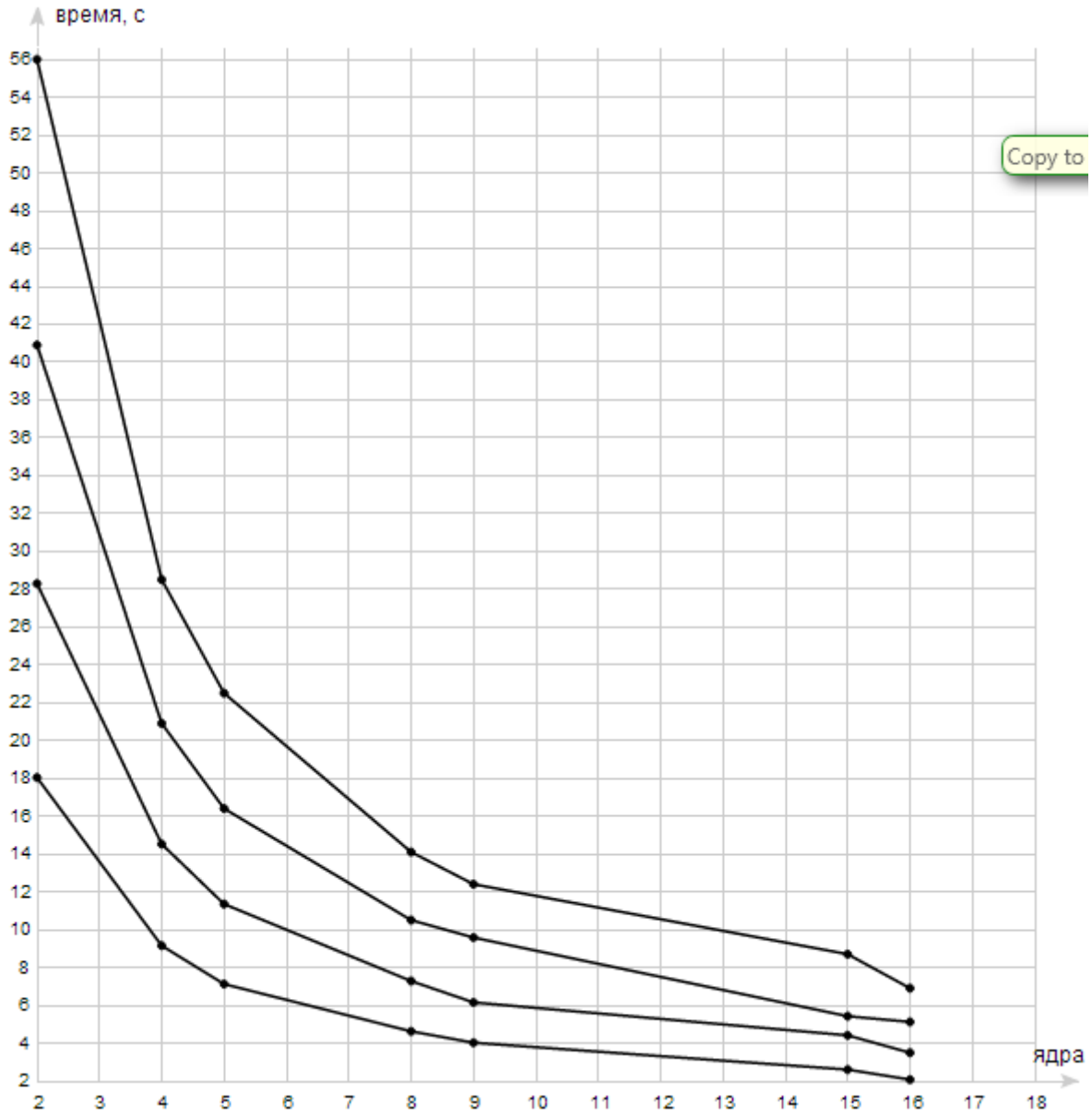


Рисунок А22 – Зависимость времени выполнения программы от количества используемых ядер

На рисунке 23 представлена зависимость времени выполнения задачи от количества итераций. Количество итераций оставалось неизменным – 2 ядра.

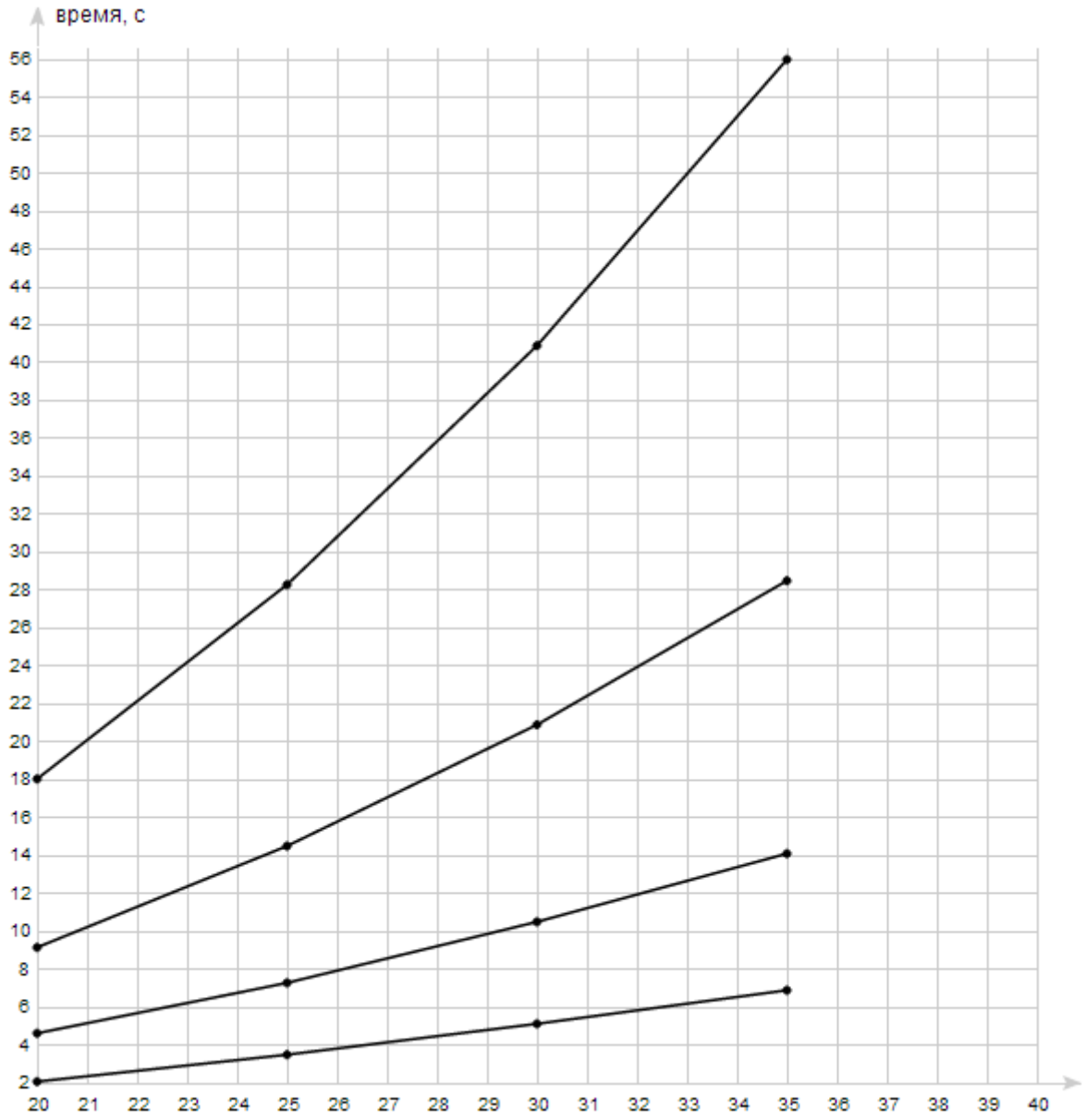


Рисунок 23 – Зависимость времени выполнения программы от количества итераций

Вариант 2. Используя технологию MPI реализовать алгоритм Штрассена для умножения матриц.

Время выполнения программы для различной размерности матриц с использованием различного числа ядер представлено в табл. 3.

Таблица 3 – Время выполнения алгоритма Штрассена

Количество ядер (увеличение аппаратурных затрат, %)	Размерность матриц			
	4096	**2048**	**1024**	**512**
1	334.175725	46.622249	6.62721	0.962019
1 (0%)	287.05 (-14.1%)	40.17 (-13.83%)	5.75 (-13.22%)	0.86 (-10.8%)
2 (100%) (0%)	193.77 (-32.49%)	26.98 (-32.83%)	3.89 (-32.36%)	0.6 (-30.12%)
3 (50%) (0%)	97.87 (-49.49%)	13.81 (-48.83%)	2.01 (-48.45%)	0.33 (-45.44%)
4 (33.33%) (0%)	67.39 (-31.14%)	9.5 (-31.19%)	1.38 (-31.1%)	0.23 (-31.18%)
6 (50%) (0%)	49.72 (-26.22%)	7.01 (-26.23%)	1.02 (-26.21%)	0.17 (-26.33%)
8 (33.33%) (0%)	38.07 (-23.43%)	5.35 (-23.68%)	0.77 (-24.34%)	0.13 (-23.66%)
9 (12.5%) (0%)	29.8 (-21.74%)	4.2 (-21.52%)	0.6 (-22%)	0.1 (-20.93%)
10 (11.11%) (0%)	25.08 (-15.82%)	3.52 (-16.16%)	0.51 (-15.6%)	0.08 (-15.94%)
12 (20%) (0%)	18.75 (-25.26%)	2.63 (-25.18%)	0.38 (-25.17%)	0.06 (-25.18%)
16 (33.33%) (0%)	14.1 (-24.81%)	1.99 (-24.33%)	0.29 (-24.84%)	0.05 (-24.81%)
17 (6.25%) (0%)	10.65 (-24.42%)	1.47 (-26.28%)	0.21 (-26.28%)	0.04 (-25.41%)
20 (17.65%) (0%)	9.14 (-14.22%)	1.26 (-14.33%)	0.18 (-14.36%)	0.03 (-15.49%)

На рис. 24 показаны графики зависимости времени выполнения от размерности матриц при использовании от одного до четырёх ядер.

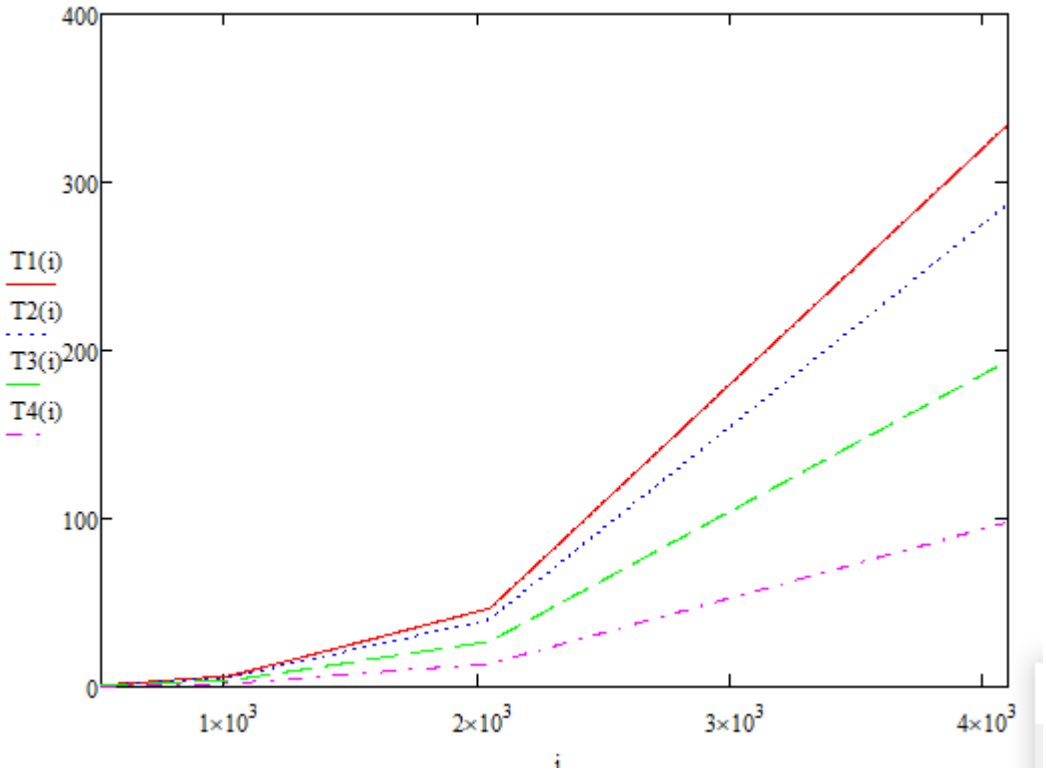


Рисунок 24 – Зависимость времени выполнения от размерности

На рис.25 показан график зависимости времени выполнения от количества ядер при умножении матриц размерностью 4096.

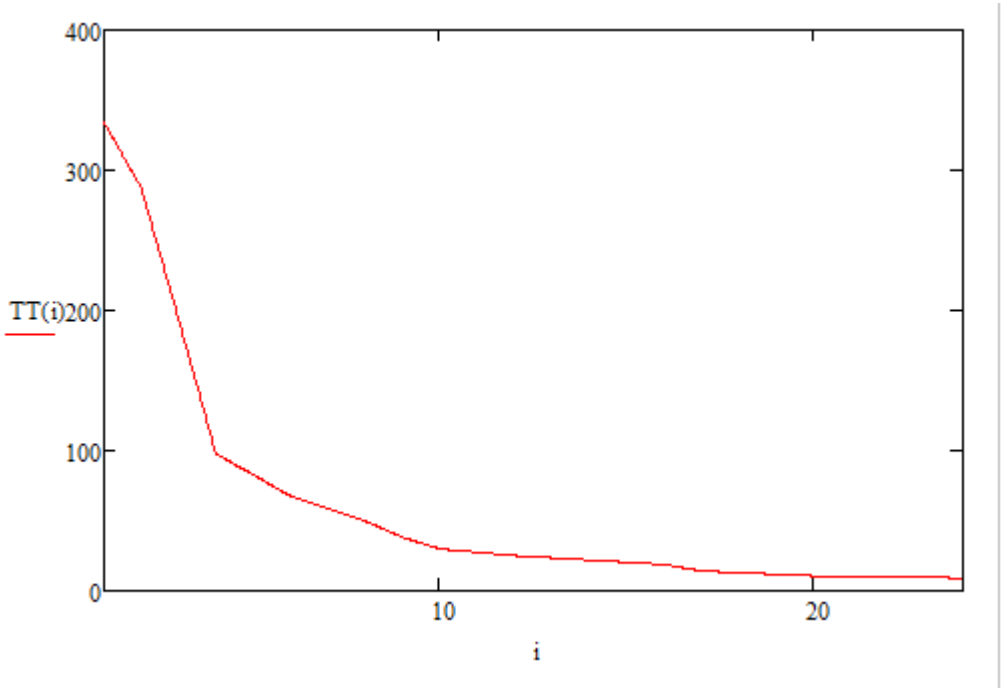


Рисунок 25 – Зависимость времени от числа ядер

Алгоритм Штрассена имеет полиномиальную ассимптотическую сложность, что подтверждает вид графика на рисунке 1. При увеличении размерности матриц время решения задачи увеличивается нелинейным образом.

Увеличение числа ядер при сохранении размерности массива приводит к уменьшению времени выполнения программы. Однако зависимость времени выполнения от количества ядер имеет обратную экспоненциальную зависимость. Это связано с накладными расходами на передачу сообщений и организацию вычислений, поэтому добавление дополнительных ядер уменьшает время решения задачи все менее значительно.

К недостаткам данного алгоритма можно отнести большие затраты памяти. По этой причине технические характеристики системы не позволили вычислить произведение матриц размерности, большей, чем 4096.

ЗАКЛЮЧЕНИЕ

Архитектура кластерных систем во многом похожа на архитектуру MPP-систем. Тот же принцип распределенной памяти, использование в качестве вычислительных узлов законченных вычислительных машин, большой потенциал для масштабирования системы и целый ряд других особенностей. В первом приближении кластерную технологию можно рассматривать как развитие идей массовых параллельных вычислений. С другой стороны, многие черты кластерной архитектуры дают основание считать ее самостоятельным направлением в области MIMD-систем.

В качестве узла кластера может выступать как однопроцессорная ВМ, так и ВС типа SMP (логически SMP-система представляется как единственная ВМ). Как правило, это не специализированные устройства, приспособленные под использование в вычислительной системе, как в MPP, а серийно выпускаемые вычислительные машины и системы. Еще одна особенность кластерной архитектуры состоит в том, что в единую систему объединяются узлы разного типа, от персональных компьютеров до мощных ВС. Кластерные системы с одинаковыми узлами называют гомогенными кластерами, а с разнотипными узлами - гетерогенными кластерами.

Использование машин массового производства существенно снижает стоимость ВС, а возможность варьирования различных по типу узлов позволяет получить необходимую производительность за приемлемую цену. Важно и то, что узлы могут функционировать самостоятельно и отдельно от кластера. Для этого каждый узел работает под управлением своей операционной системы. Чаще всего используются стандартные ОС: Linux, FreeBSD, Solaris и версии Windows, продолжающие на-правление Windows NT.

Узлы в кластерной системе объединены высокоскоростной сетью. Решения могут быть простыми, основанными на аппаратуре Ethernet, или сложными с высокоскоростными сетями пропускной способности в сотни

мегабайтов в секунду (Мбит/с). К последней категории относятся сети SCI компании Scali Computer (≈100 Мбит/с) и Mirynet (≈120 Мбит/с). В принципе, за основу кластерной системы может быть взята стандартная локальная сеть (или сеть большего масштаба), с сохранением принятых в ней протоколов (правил взаимодействия). Аппаратурные изменения могут не потребоваться или, в худшем случае, сводятся к замене коммуникационного оборудования на более производительное. При соединении машин в кластер почти всегда поддерживаются прямые межмашинные связи.

Вычислительные машины (системы) в кластере взаимодействуют в соответствии с одним из двух транспортных протоколов. Первый из них, протокол TCP (Transmission Control Protocol), оперирует потоками байтов, гарантируя надежность доставки сообщения. Второй - UDP (User Datagram Protocol) пытается посылать пакеты данных без гарантии их доставки. В последнее время применяют специальные протоколы, которые работают намного лучше, например Virtual Interface Architecture (VIA).

При обмене информацией используются два программных метода: передачи сообщений и распределенной, совместно используемой памяти. Первый опирается на явную передачу информационных сообщений между узлами кластера. В альтернативном варианте также происходит пересылка сообщений, но движение данных между узлами кластера скрыто от программиста.

Подключение узлов к сети осуществляется посредством сетевых адаптеров. Учитывая роль коммуникаций, для связи ядра вычислительного узла с сетью используется выделенная шина ввода/вывода. К этой шине также подключаются локальные магнитные диски. Наличие таких дисков типично для кластерных систем, но не характерно для MPP-систем. Элементы вычислительного ядра объединяются посредством локальной системной шины. Связь между этой шиной и шиной ввода/ вывода обеспечивает мост.

Неотъемлемая часть кластера - специализированное программное обеспечение (ПО), организующее бесперебойную работу при отказе одного или нескольких узлов. Такое ПО должно быть установлено на каждый узел кластера. Оно реализует механизм передачи сообщений над стандартными сетевыми протоколами и может рассматриваться как часть операционной системы. Именно благодаря специализированному ПО группа ВМ, объединенных сетью, превращается в кластерную вычислительную систему. Кластерное ПО перераспределяет вычислительную нагрузку при отказе одного или нескольких узлов кластера, а также восстанавливает вычисления при сбое в узле. ПО каждого узла постоянно контролирует работоспособность всех остальных узлов. Этот контроль основан на периодической рассылке каждым узлом сигнала, известного как keepalive («пока жив») или heartbeat («сердцебиение»). Если сигнал от некоторого узла не поступает, то узел считается вышедшим из строя; ему не дается возможность выполнять ввод/ вывод, его диски и другие ресурсы (включая сетевые адреса) переназначаются другим узлам, а выполнявшиеся им программы перезапускаются в других узлах. При наличии в кластере совместно используемых дисков, кластерное ПО поддерживает единую файловую систему.

Кластеры обеспечивают высокий уровень доступности - в них отсутствуют единая операционная система и совместно используемая память, то есть нет проблемы когерентности кэшей. При создании кластерных систем используется один из двух подходов. Первый подход применяется для построения небольших кластерных систем, например, на базе небольших локальных сетей организаций или их подразделений. В таком кластере каждая ВМ продолжает работать как самостоятельная единица, одновременно выполняя функции узла кластерной системы.

Второй подход ориентирован на использование кластерной системы в роли мощного вычислительного ресурса. Узлами кластера служат только

системные блоки вычислительных машин, компактно размещаемые в специальных стойках. Управление системой и запуск задач осуществляет полнофункциональный хост-компьютер. Он же поддерживает дисковую подсистему кластера и разнообразное периферийное оборудование. Отсутствие у узлов собственной периферии существенно удешевляет систему.

Кластеры хорошо масштабируются путем добавления узлов, что позволяет достичь высочайших показателей производительности. Благодаря этой особенности архитектуры кластеры с сотнями и тысячами узлов положительно зарекомендовали себя на практике. До недавнего времени именно кластерная ВС занимала первую позицию в этом списке самых производительных систем. Речь идет о кластерной системе Roadrunner BladeCenter QS22, созданной компанией IBM. Теоретическая пиковая производительность системы составляет 1376 TFLOPS, состоит она из 122 400 узлов на базе процессоров Opteron и PowerXCell.

Четыре преимущества, достигаемые с помощью кластеризации:

Абсолютная масштабируемость. Возможно создание больших кластеров, превосходящих по вычислительной мощности даже самые производительные одиночные ВМ. Кластер в состоянии содержать десятки узлов, каждый из которых представляет собой мультипроцессор.

Наращиваемая масштабируемость. Кластер строится так, что его можно наращивать, добавляя новые узлы небольшими порциями. Таким образом, пользователь может начать с умеренной системы, расширяя ее по мере необходимости.

Высокий коэффициент готовности. Поскольку каждый узел кластера - самостоятельная ВМ или ВС, отказ одного из узлов не приводит к потере работоспособности кластера. Во многих системах отказоустойчивость автоматически поддерживается программным обеспечением.

Превосходное соотношение цена/производительность. Кластер любой производительности можно создать, соединяя стандартные

«строительные блоки», при этом его стоимость будет ниже, чем у одиночной ВМ с эквивалентной вычислительной мощностью.

В то же время нужно упомянуть и основной недостаток, свойственный кластерным системам, - взаимодействие между узлами кластера занимает гораздо больше времени, чем в других типах ВС. Кластеры больших SMP-систем Огромный потенциал масштабирования, свойственный кластерной архитектуре, делает ее очень перспективным направлением в области создания высокопроизводительных вычислительных систем. Масштабирование возможно как за счет увеличения числа узлов, так и путем применения в качестве узлов не одиночных ВМ, а также хорошо масштабируемых вычислительных систем, обычно SMP-типа. Это направление получило настолько широкое развитие, что было выделено в отдельную группу MIMD-устройств - Constellations. Термин, обусловленный названием одной из первых ВС данного типа - Sun «Constellation», - можно перевести как «созвездие».

Constellation-система - это кластер, узлами которого служат SMP-системы. В качестве отличительного признака выступает число процессорных элементов в узле. Изначально принималось, что система относится к Constellation-системам, если число узлов в кластере из SMP-систем меньше или равно количеству процессорных элементов в SMP-системе узла. Для современных систем с большим количеством узлов такое условие не всегда соблюдается, поэтому в настоящее время условием причисления ВС к Constellation-системам служит число ПЭ в узле - оно должно быть больше (равно) 16. Системы, не отвечающие данному условию, считаются классическими кластерными системами.

Перспективность Constellation-систем обусловлена удачным сочетанием преимуществ распределенной памяти (возможности наращивания количества узлов) и разделяемой памяти (эффективного доступа множества ПЭ к памяти).

Формально структура Constellation-системы полностью соответствует кластерной архитектуре, однако явная направленность на высокую производительность может отражаться в некоторых конструктивных решениях.

В качестве примера Constellation-системы можно привести систему Tera 10 фирмы Bull с теоретической пиковой производительностью 58,8 TFLOPS. Система представляет собой кластер из 544 узлов, в котором каждый узел - это SMP-система, образованная 8 2-ядерными процессорами Itanium 2.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Жуматий С. А. , Воеводин В. В. Вычислительное дело и кластерные системы. М.: Интернет Университет Информационных Технологий, 2008. –125 с.
2. Сердюк Ю. П. Кластерные вычисления: учебный курс - М.: Интернет Университет Информационных Технологий, 2008. – 178 с.
3. Гергель В. П. Технологии построения и использования кластерных систем. - М.: Интернет Университет Информационных Технологий, 2009. – 470 с.
4. Левин М. П. Параллельное программирование с использованием OpenMP: учебное пособие - М.: Интернет Университет Информационных Технологий, 2008. – 120 с.
5. Богданов А.В., Корхов В.В., Мареев В.В., Станкова Е.Н. Архитектуры и топологии многопроцессорных вычислительных систем. Интернет-университет информационных технологий - ИНТУИТ.ру, 2004.
6. Варфоломеев В.А., Лецкий Э.К., Шамров М.И., Яковлев В.В. Архитектура и технологии IBM eServer zSeries Интернет-университет информационных технологий - ИНТУИТ.ру, 2005.
7. Новиков Ю.В., Скоробогатов П.К. Основы микропроцессорной техники Интернет-университет информационных технологий - ИНТУИТ.ру, 2004
8. Гуров В.В., Чуканов В.О. Основы теории и организации ЭВМ Интернет-университет информационных технологий - ИНТУИТ.ру, 2006.
9. Пятибратов А.П. Вычислительные системы, сети и телекоммуникации : учебник. [Электронный ресурс]. - М.: Финансы и статистика, 2013: Точка доступа/ http://biblioclub.ru.
10. Чекмарев Ю.В. Вычислительные системы, сети и коммуникации - М.: ДМК Пресс, 2009.

11. Лавров Д.Н. Сети и системы телекоммуникаций: учебное пособие. - Омск: Омский государственный университет, 2009.
12. Архитектура вычислительных систем: учебное пособие для вузов.- М.: МГТУ им. Н.Э. Баумана, 2009.
13. Мелехин В.Ф. Вычислительные машины, системы и сети: учебник для вузов.- М.: Академия, 2010.
14. Исаченко О.В. Программное обеспечение компьютерных сетей. – М.: Академия, 2014.
15. А.В.Сенкевич Архитектура ЭВМ и вычислительные системы: учебник: – М.: Academia, 2014.
16. Горец Н.Н. ЭВМ и периферийные устройства. Устройства ввода и вывода -М: Издательский центр «Академия», 2013.
17. Баула В.Г. Архитектура ЭВАМ и операционные среды-М: Издательский центр «Академия», 2012.
18. Афанасьев К. Е. , Григорьева И. В. , Рейн Т. С. Основы высокопроизводительных вычислений: учебное пособие. Т. 3. Параллельные вычислительные алгоритмы - Кемерово: Кемеровский государственный университет, 2012. – 185 с.
19. Афанасьев К. Е. , Стуколов С. В. , Малышенко В. В. , Карабцев С. Н. , Андреев Н. Е. Основы высокопроизводительных вычислений: учебное пособие. Т. 2. Технологии параллельного программирования - Кемерово: Кемеровский государственный университет, 2012 – 412с.
20. Антонов А. С. Параллельное программирование с использованием технологии MPI. -М.: Интернет Университет Информационных Технологий, 2008. – 71 с.