## Схема тестовых серверов

## Общие характеристики

|  |  |  |
| --- | --- | --- |
|  | **IBM02** | **SV084** |
| OS | AIX 7 | Oracle Linux 7.4 |
| * Login | oracle/oracle  root/1234 | oracle/oSposOra  root/oSposStand |
| * System monitor | nmon | nmon |
| * Editor | vi | nano |
| * Bash profile config | ~/.profile | ~/.bash\_profile |
| Oracle |  |  |
| * Database | 12.1.0.1 (multitenant) | 12.1.0.1, 12.2.0.1 (multitenant) |
| * Oracle Grid | 12.1.0.1 (standalone) | 12.2.0.1 (standalone) |
| * Oracle home | /ora1/12\_1\_0  /ora1/grid | /u01/app/oracle/product/12.1.0/db\_1  /u01/app/oracle/product/12.2.0/grid |
| * Set env scripts | ~/.cdb\_env  ~/.grid\_env | ~/.cdb\_env  ~/.grid\_env |
| * Login | sys/oSposC.c  system/passwd | sys/oSposC.c  system/passwd |
| Places |  |  |
| * Database files | /ora1/dat | /u01/app/oracle/oradata |
| * Backup files | /ora1/sav/bkup | /u01/app/oracle/backup/aix\_files |
| * Datapump | /ora1/buf/datapump/ | /u01/app/oracle/buf/datapump |
| * Duplicate scripts | /home/oracle/scripts/duplicate | /home/oracle/scripts/duplicate |
| * Restart scripts | /home/oracle/scripts/maintenant | /home/oracle/scripts/maintenant |
| * PDB\_DAEMON |  | /home/oracle/scripts/pdb\_daemon |

## Типы БД

* **TSTCDB** – контейнерная БД
* **TSTDB** – копия FONDB, полученная командой RMAN DUPLICATE из BACKUP-файлов и конвертированная в PDB. Используется для подготовки БД к транспортировке в Linux и доступная для работы в остальное время.
* **PDB\_ROOT** – PDB для создания клонов и управления структурой тестовой БД в целом. Содержит одну единственную, одноименную схему PDB\_ROOT (см. раздел [PDB\_ROOT](#_Клонирование))
* **DEV\_NODE, WEEKLY\_NODE** –полные копии TSTDB, полученные методом Transportable Tablespaces. Эти PDB используются только как основа для создания клонов первого уровня после обновления, в остальное время не доступны для использования.
* **DEV\_CLONE, WEEKLY\_CLONE** – клоны первого уровня, полученные методом «тонкого» клонирования отDEV\_NODE, WEEKLY\_NODE. Предназначены только для создания клонов второго уровня, доступны только на чтение.
* **DEV, WEEKLY, PDB\_RELEASE** – клоны второго уровня, предназначенные для совместного использования разработчиками, тестировщиками.
* **Others** – все остальные клоны, второго или более уровня, созданные для индивидуального или совместного использования (см. раздел [PDB\_ROOT](#_Клонирование)).

## ACFS

На SV084, для организации «тонкого» клонирования БД, используется файловая система ACFS, поддерживающая технологию снапшотов.

Для хранения файлов БД на SV084 используются две файловые системы ACFS (по 1,5 Tb для каждой ноды):

[oracle@SV084 oradata]$ df -h | grep vol

/dev/asm/vol\_dev-420 1.5T 1.1T 484G 68% /u01/app/oracle/oradata/dev

/dev/asm/vol\_weekly-82 1.5T 434G 1.1T 29% /u01/app/oracle/oradata/weekly

Для вывода информации о состоянии подключенных ACFS настроен алиас **snapinfo** (см. ~/.bash\_profile):

[oracle@SV084 ~]$ **snapinfo**

SNAP INFO /u01/app/oracle/oradata/dev

snapshot name: 73C6B81E135544CFE0530100007F6737

snapshot location: /u01/app/oracle/oradata/dev/.ACFS/snaps/73C6B81E135544CFE0530100007F6737

RO snapshot or RW snapshot: RW

parent name: /u01/app/oracle/oradata/dev

snapshot creation time: Sun Aug 19 13:42:48 2018

storage added to snapshot: 524308480 ( 500.02 MB )

snapshot name: 73C6B81E135F44CFE0530100007F6737

snapshot location: /u01/app/oracle/oradata/dev/.ACFS/snaps/73C6B81E135F44CFE0530100007F6737

RO snapshot or RW snapshot: RW

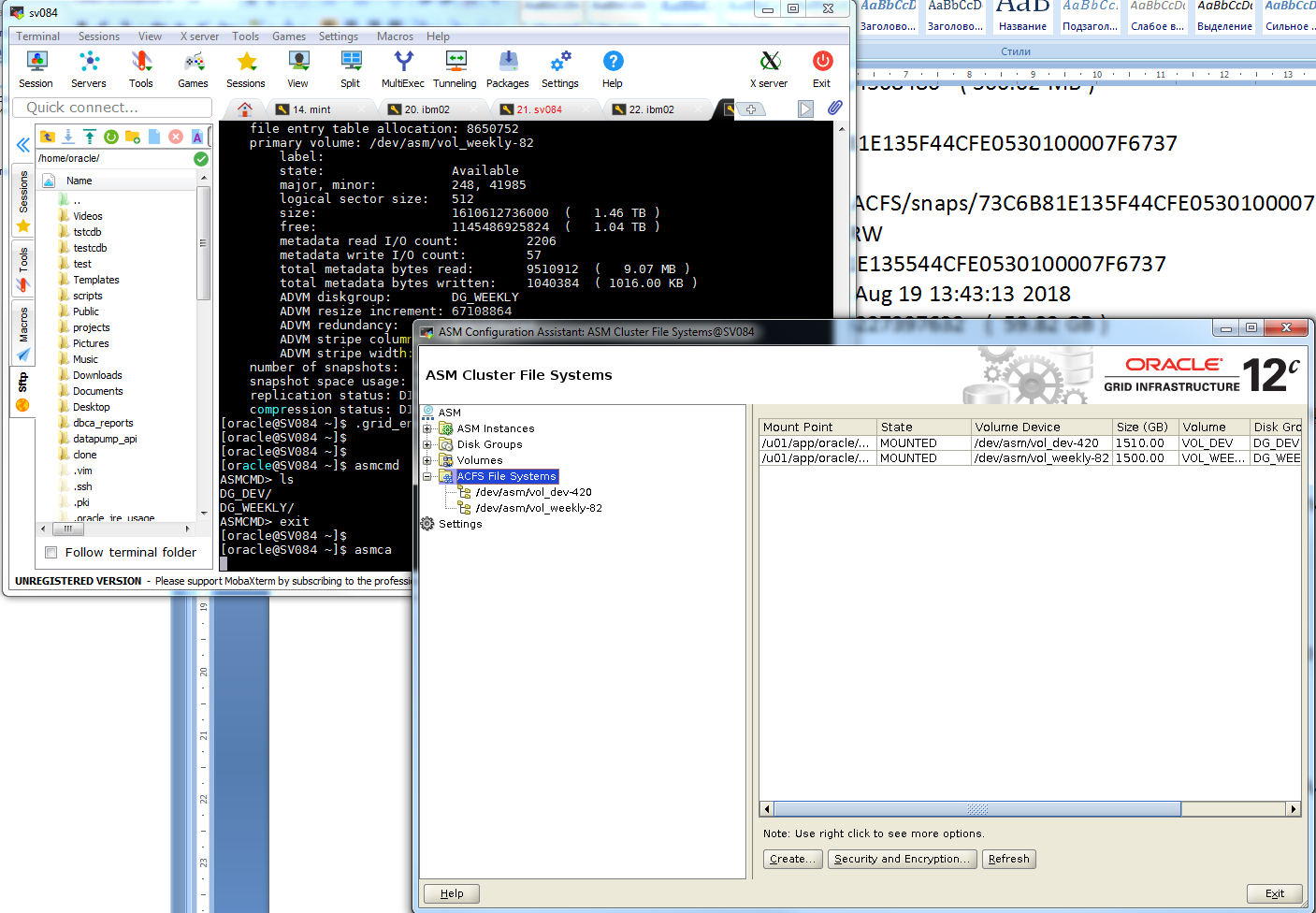
parent name: 73C6B81E135544CFE0530100007F6737

snapshot creation time: Sun Aug 19 13:43:13 2018

storage added to snapshot: 64227397632 ( 59.82 GB )

Информация выводится в разрезе подключенных ACFS, при этом сначала идут данные по имеющимся снапшотам, затем данные в целом по файловой системе.

Для управления и конфигурации ACFS, можно использовать либо консольную утилиту asmcmd, либо графическое приложение asmca (терминал должен поддерживать X Window):



Полное описание команд для управления и мониторнига см. в документации по Oracle Grid Infrastructure.

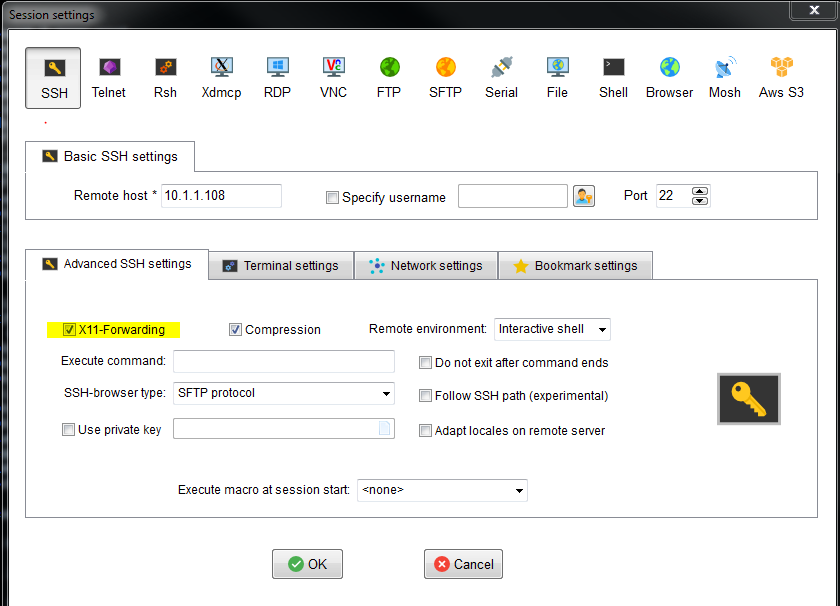
### Проблемы

После аварийной остановки или после обслуживания сервера, сталкивался с невозможностью подключения дисков ACFS.

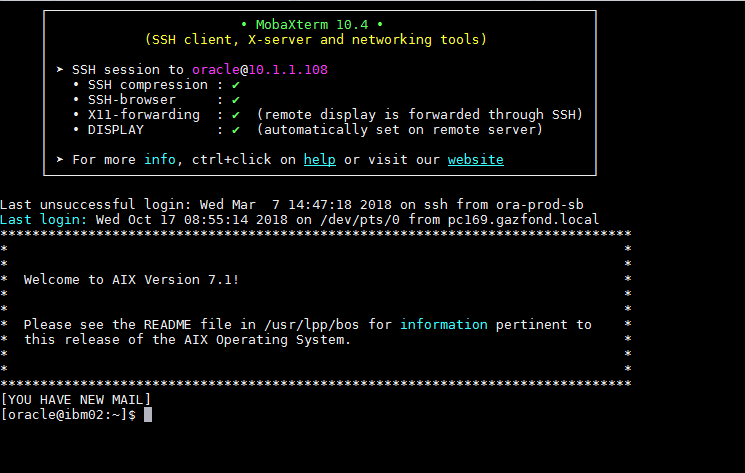
На IBM02 Grid установлен, но не используется. При желании можно задействовать.

## Подключение

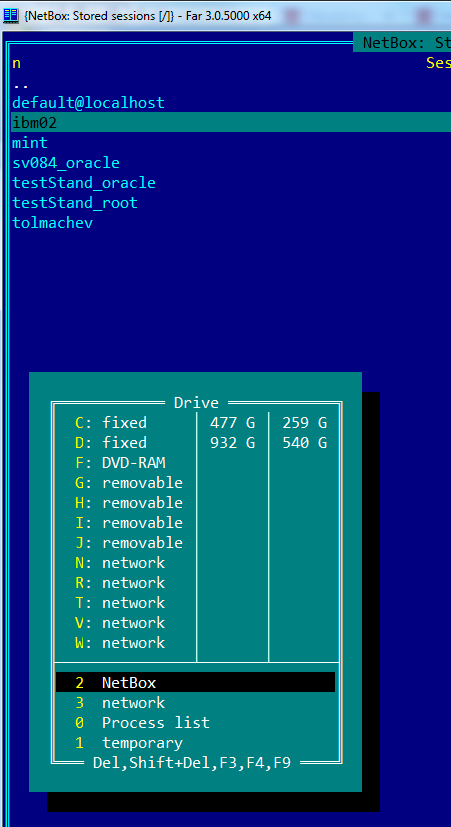
Подключение к серверам выполняется любым терминалом по ssh. Для примера, настройка MobaXterm:

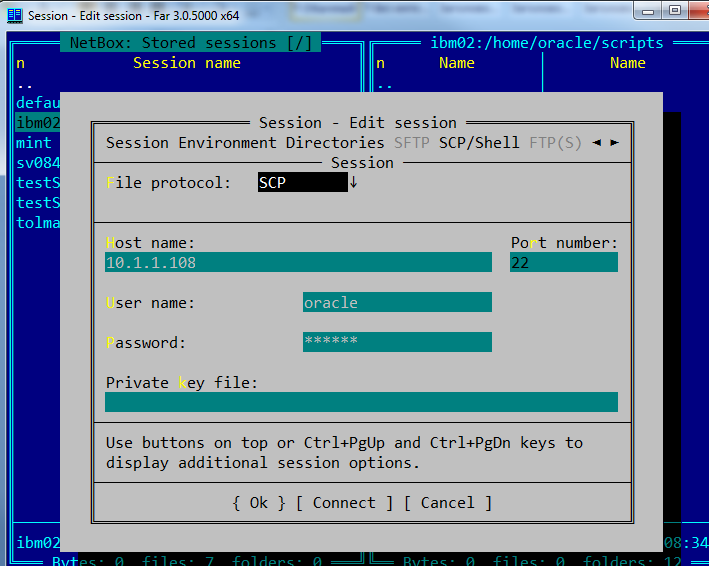


Подключение:



Для доступа к файловой системе, удобно использовать плагин FAR`а NetBox:





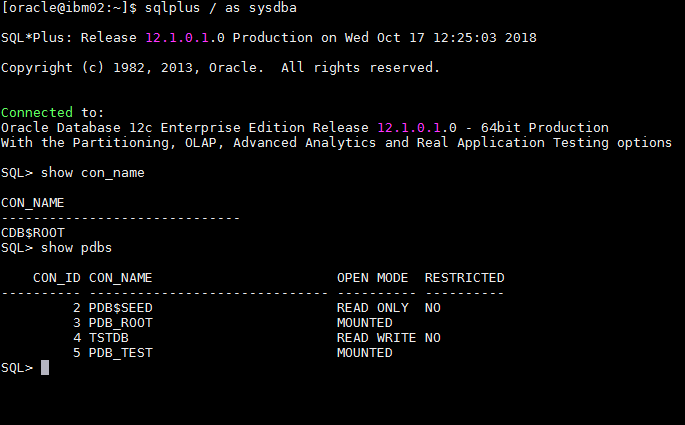
### Enterprise Manager

IBM02: <https://10.1.1.108:5501/em>

На SV084 тоже настроен, но коннект не проходит. В чем дело – не разбирался.

### SQLPLUS

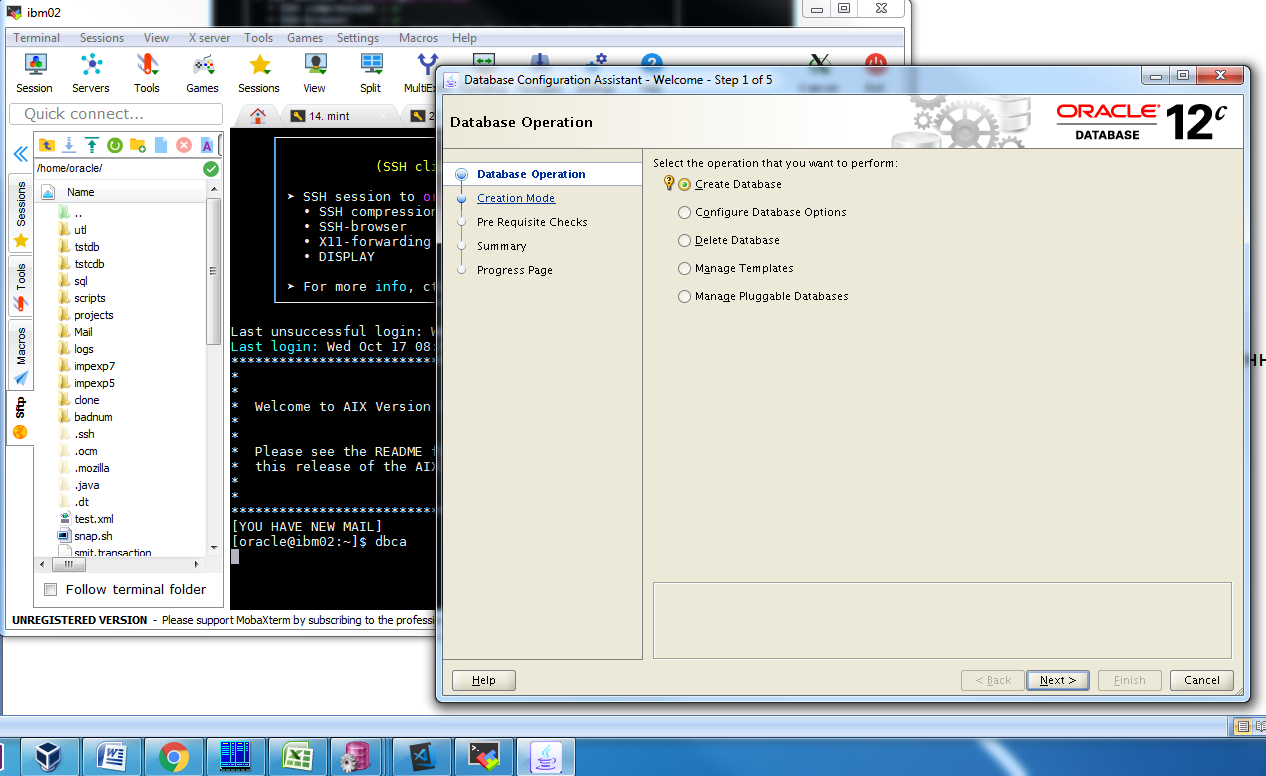
По умолчанию, переменные окружения настроены для подключения к БД, поэтому просто запускаем:



На sv084 sqlplus запускается через rlwrap (настроен алиас, см. ~/.bash\_profile), т.е. работают клавиши навигации, в т.ч. для истории команд (вверх и вниз).

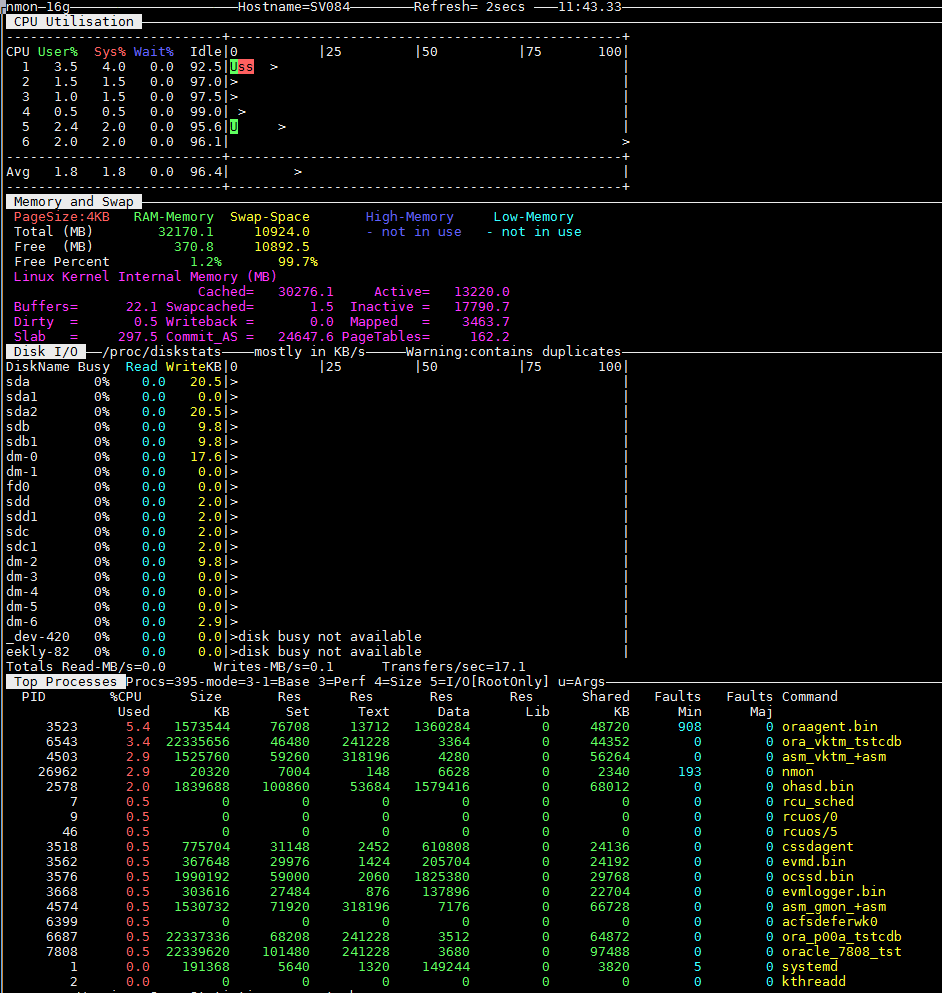
### X Window

На обоих серверах настроен X Window, позволяющий вывести GUI интерфейс непосредственно на подключенном компьютере. Например, запуск конфигуратора баз Oracle dbca:



### Мониторинг загрузки сервера

Консольная утилита nmon, пример (после запуска последовательно нажал клавиши m-d-c-t)



## Перезагрузка

Для перезагрузки сервера, подготовлены скрипты, расположенные в /home/oracle/scripts/maintenant

Остановка Oracle и отправка сервера в перезагрузку:

$ cd /home/oracle/scripts/maintenant/

$ ./reboot.sh

Запуск Oracle после перезагрузки

$ cd /home/oracle/scripts/maintenant/

$ ./start.sh

Если с ACFS возникают проблемы, например, не выполняется монтирование дисков, проще разобраться через приложение asmca, предварительно выставив окружение скриптом ~/.grid\_env.

## Обновление тестовых БД

Под обновлением БД Oracle подразумевается пересоздание БД (из бэкапа FONDDB или другой PDB).

Из-за различий в платформах IBM (big endian) и SV084 (small endian), на последнем не возможно развернуть БД непосредственно из бэкапа FONDDB. Поэтому, обновление БД на SV084 происходит следующим образом:

1. Создание БД на IBM02 из бэкапа FONDDB
2. Подготовка БД к транспортировке табличных пространств (transport tablespace)
3. Копирование метаданных и файлов данных с IBM на SV084
4. Создание БД на SV084

Репозиторий: <http://teamcity:3000/v.zhuravov/TransportDB.git>

Далее пути к скриптам указаны от корня репозитория. Эти же скрипты можно найти в домашней папке oracle на серверах: ~/scripts/duplicate

### Обновление TSTDB (IBM02)

Обновление TSTDB выполняется путем создания дубликата FONDDB на основе бэкап-файлов последнего. Т.к. FONDDB не контейнерная БД, после создания TSTDB конвертируется в PDB и подключается в TSTCDB.

Скрипты: @repo:\\scripts\recreate\ibm\duplicate

Диаграмма обновления TSTDB:

Сигнальный файл – признак готовности файлов данных и метаданных к транспортировке. Наличие этого файл проверяется перед началом обновления БД на SV084.

Сигнальный файл создается так:

echo "TTS Ready" > /ora1/buf/datapump/tts\_ready\_`date +%y%m%d`

Запуск процесса обновления выполняется crontab’ом:

$ crontab –e

30 08 \* \* 6 /home/oracle/scripts/duplicate/cron\_start.sh

Т.е. запуск обновления каждую субботу в 8:30

Продолжительность обновления ~5 часов, т.е. вполне подходит для ежедневного пересоздания.

После завершения подготовки, БД остается доступной только на чтение, т.к. все пользовательские табл.пространства доступны только на чтение. После копирования файлов данных на целевую систему (SV084) нужно подключиться к TSTDB и перевести табличные пространства в read write. Запуск этого процесса не автоматизирован и выполняется вручную. Пример скрипта: sql/online.sql

Вообще, переключение режима доступа табл.пространств, лучше добавить в скрипты на базе приемника (SV084), например, после копирования файлов данных, запустить удаленное подключение к TSTDB по SYSTEM и переключить.

P.s.: во второй версии скриптов обновления на SV084 (duplicate2) добавил скрипт переключения табл.пространств на TSTDB.

### Подготовка к транспортировке

Перенос БД методом транспортировки табл.пространств имеет ряд ограничений. Часть из них можно увидеть так:

begin

dbms\_tts.transport\_set\_check( ‘ARHDATA,CTXDATA,DWHDATA,ETL\_DATA,ETL\_INDX,FIASDATA,FONDDATA,FONDINDX,GFNDDAT2,GFNDINDX,GFPNDATA,GFPNINDX,INFDATA,MDMDAT2,MDMDATA,OPSDATA,SMPPDATA,USERDATA,WEBDATA',

true

);

end;

/

select \*

from sys.transport\_set\_violations

/

Последний запрос не должен возвращать записей. Если записи есть – надо разбираться с проблемами

К сожалению, это не дает гарантии, что транспортировка пройдет успешно. Я столкнулся со следующей проблемой. При импорте DATAPUMP’ом метаданных на SV084, процесс импорта просто отваливался, без каких-либо внятных сообщениях об ошибке, просто разрыв соединения при импорте PACKAGE BODIES. Просмотр alert.xml показал на ошибку ORA-07445 – критическая ошибка на уровне ОС. Т.е. по сути тоже ничего не показал. При этом не поддерживается перезапуск импорта DATAPUMP при транспортировке. Опытным путем была локализована следующая ошибка: BUG 17404511

Oracle 12.1.0.1 на Linux Oracle не смог обработать использование переменных в оконных выражениях запросов: fetch first и fetch next и offset. Т.е. если используется константа – норм, если переменная – обрыв соединения. Выход: обернуть переменные функцией TO\_NUMBER.

На текущий момент все переменные в оконных выражениях обвернуты в to\_number, но …

### SV084

На SV084 созданы две физические копии БД: WEEKLY\_NODE и DEV\_NODE. Каждая из которых, имеет множество «тонких» клонов, различного уровня вложенности. Перед обновлением физической копии БД, выполняется удаление всех ее клонов, с сохранением существующей структуры в PDB\_ROOT. После обновления физ.копии, выполняется воссоздание всех клонов, входящих в существующую структуру.

#### WEEKLY

Еженедельное обновление.

Скрипты: scripts\recreate\sv084\duplicate

#### DEV

Обновление по запросу. Обновление может быть выполнено двумя способами:

1. Обновление транспортировкой базы TSTDB (IBM02), аналогично WEEKLY.
2. Обновление на основе WEEKLY, т.е. создание полновесного клона WEEKLY.

Первый способ описан выше.

Второй способ. Скрипты: scripts\recreate\sv084\update\_dev

Скрипты использовались единожды. Требуют доработки

## Управление клонами

Для управления клонами создана отдельная разработка PDB\_ROOT. Ее основное назначение – обеспечение безопасного управления жизнедеятельностью «тонких» клонов БД: создание, изменение состояния, удаление.

PDB\_ROOT работает асинхронно, задачи ставятся в очередь, которую обрабатывает фоновый процесс.

«Тонкий» клон (“thin” clone) – это клон БД, созданный на базе «снапшота» исходной БД.

Состав PDB\_ROOT:

1. Common user C##PDBROOT
2. Подключаемая база PDB\_ROOT
3. Процесс PDB\_DAEMON

Репозиторий: http://teamcity:3000/v.zhuravov/pdb\_root.git

### C##PDBROOT

Пользователь C##PDBROOT (пароль c##pdbroot) создан для запуска команд создания, изменения состояния и удаления «тонких» клонов, требующих SYS привилегий на уровне CDB. На уровне этого пользователя хранятся только метаданные: пакет PDB\_DAEMON\_API, несколько представлений и синонимов для объектов PDB\_ROOT, а также DBLINK на PDB\_ROOT.

Пакет PDB\_DAEMON\_API – содержит логику работы процесса-демона PDB\_DAEMON: обработка очереди команд на создание, изменение и удаление клонов.

ВНИМАНИЕ: у пользователя SYS привилегия

### PDB\_ROOT

Отдельная база PDB\_ROOT. В PDB\_ROOT имеется только один пользователь PDB\_ROOT, с оригинальным паролем: pdb\_root.

В схеме PDB\_ROOT имеется два пакета:

1. PDB\_API – API для управления клонами
2. PDB\_PUB – wrapper для PDB\_API

Все действия по управлению клонами, лучше всего выполнять через PDB\_PUB.

В схеме есть несколько представлений, для вывода текущего состояния очереди заданий, структуры клонов, процесса PDB\_DAEMON и лога выполнения заданий, scripts\:

* edit\_actions.sql
* edit\_clones.sql
* edit\_daemon.sql
* show\ show\_daemon\_log.sql

### PDB\_DAEMON

Постоянно работающий фоновый процесс PDB\_DAEMON. Этот процесс запускается непосредственно на сервере, скрипты для запуска и остановки:

* репозиторий: scripts\ibm\_shell
* SV084: /home/oracle/scripts/pdb\_daemon

Процесс PDB\_DAEMON периодически сканирует таблицу заданий PDB\_ROOT.ACTIONS\_T и обрабатывает новые задания. Т.е. по сути простой аналог шедулера или очереди.

Остановить процесс можно в PDB\_ROOT, изменив в таблице PDB\_DAEMON\_T статус запущенного процесса (STATUS) на STOP и дождавшись заполнения поля STOP\_TIME.

Для процесса не предусмотрен автозапуск. Поэтому запуск демона включен в скрипты перезагрузки и обновления.