

Системы управления базами данных

Лектор — Цопа Е.А.
2016/17 уч. год

1. Введение

(C) Wikipedia:

СУБД — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Основные функции СУБД:

- Управление данными на дисках.
- Управление данными в ОП (в т.ч., кэширование).
- Журналирование изменений, резервное копирование, восстановление после сбоев.
- Поддержка языков БД (DML + DDL).

Состав СУБД

- *Ядро* — отвечает за управление данными.
- *Процессор языка БД* — транслирует запросы с высокоуровневого языка на низкоуровневый.
- *Подсистема поддержки времени исполнения.*
- *Сервисные программы.*

По модели данных:

- *Иерархические* — данные представляются в виде дерева. Пример — LDAP / AD, реестр Windows.
- *Сетевые* — используют сетевую модель данных. Частный случай — графовые СУБД. Примеры — HypergraphDB, OrientDB.
- *Объектно-ориентированные* — используют ОО-модель данных. Пример — InterSystems Caché.
- *Реляционные и объектно-реляционные* — используют реляционную модель данных (возможно, с частичной поддержкой ООП). Примеры — Oracle, MySQL, PostgreSQL.

По степени распределённости:

- *локальные;*
- *распределённые.*

По способу доступа к БД.

- *Файл-серверные* — данные находятся на файл-сервере, СУБД — на каждом клиентском компьютере. Примеры — MS Access, dBase, FoxPro.
- *Клиент-серверные* — СУБД находятся на сервере вместе с данными. Примеры — Oracle, MS SQL Server, Caché.
- *Встраиваемые* — СУБД встраивается в приложение, хранит только его данные и не требует отдельной установки. Примеры — SQLite, BerkeleyDB.

Архитектура ANSI-SPARC

Предложена в 1975 г. подкомитетом SPARC (Standards Planning And Requirements Committee) ANSI.

Архитектура СУБД включает в себя 3 уровня:

- Внешний (пользовательский).
- Промежуточный (концептуальный).
- Внутренний (физический).

Почти все современные СУБД соответствуют принципам ANSI-SPARC.

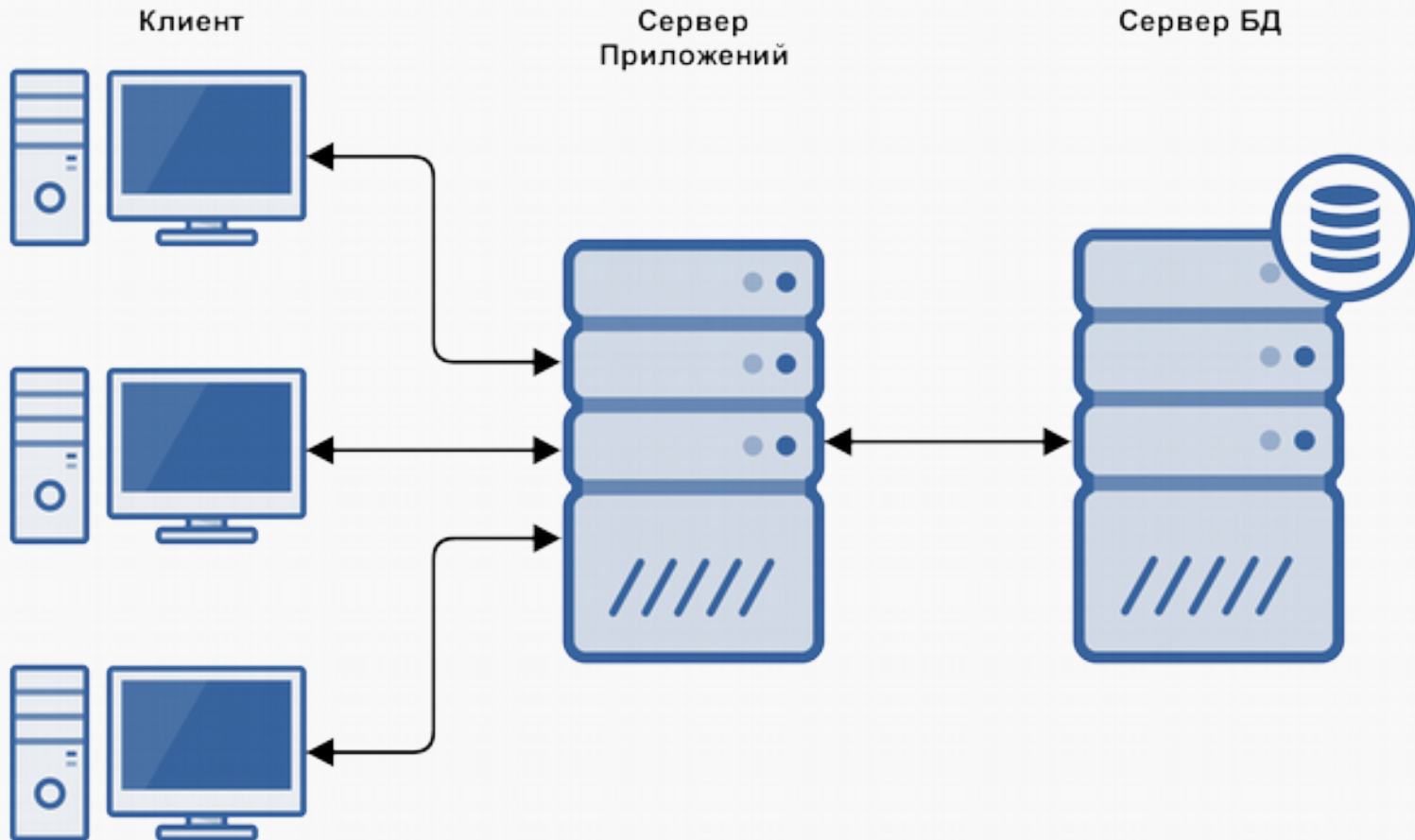
СУБД Oracle

- Исторически первая и наиболее распространённая коммерческая СУБД на основе языка SQL.
- По классификации — *объектно-реляционная распределённая клиент-серверная СУБД.*
- Очень показательный пример архитектуры Enterprise-level решения.
- Первая версия (v2) была выпущена в 1979 г.
- Актуальная версия — 12с («cloud» — «облако»).
- Начиная с версии 3 (1983 г.) реализована поддержка транзакций.
- В версии 7 (1992 г.) появилась поддержка PL/SQL.
- В версии 8 (1997 г.) реализована поддержка ООП.
- В версии 9 (2001 г.) реализована технология RAC (Real Application Cluster) — появилась возможность реализации кластерных БД.

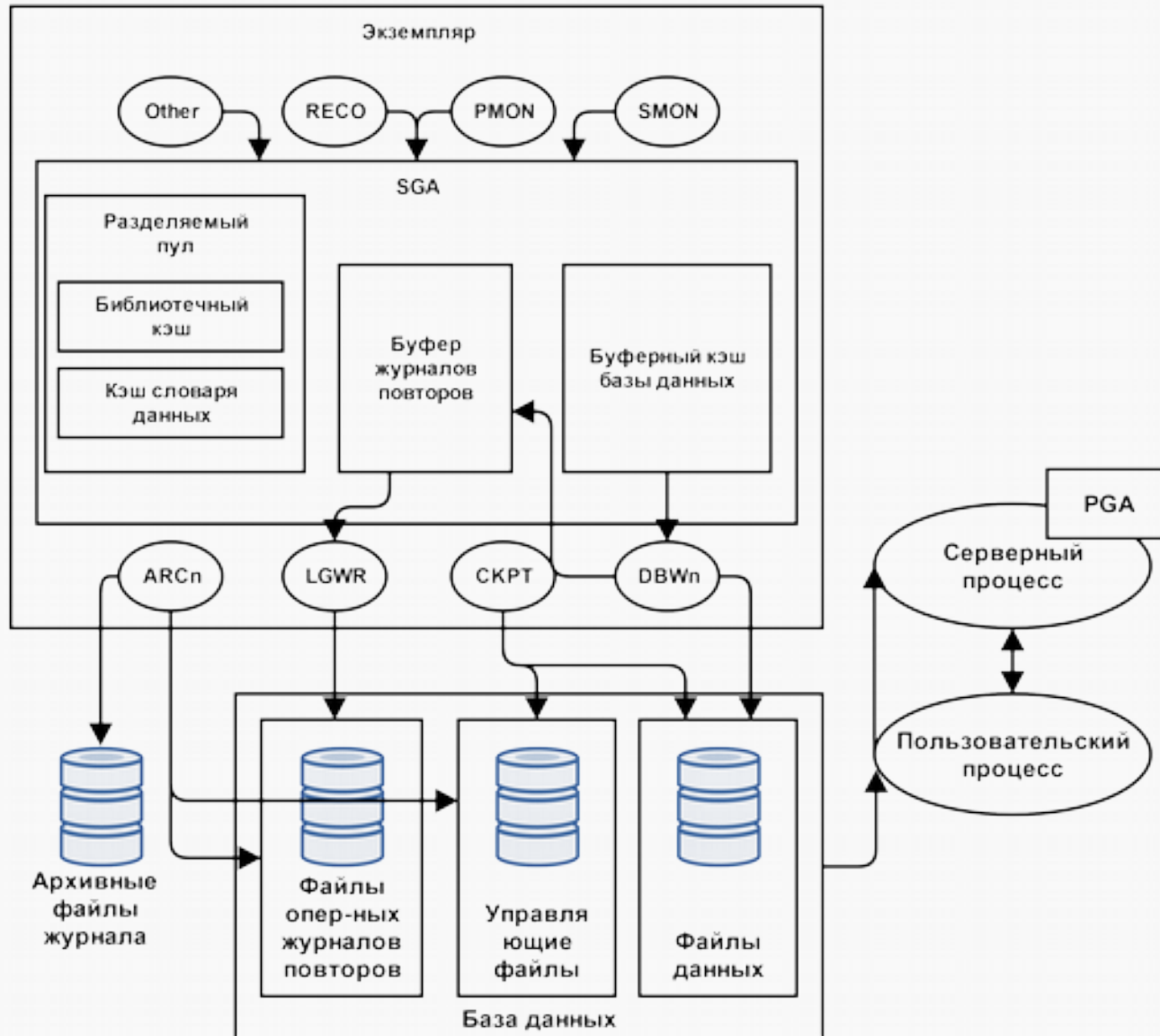
2. Архитектура СУБД Oracle

Многоуровневая архитектура

Обычно ИС на базе СУБД Oracle включает в себя 3 уровня:



Архитектура Oracle

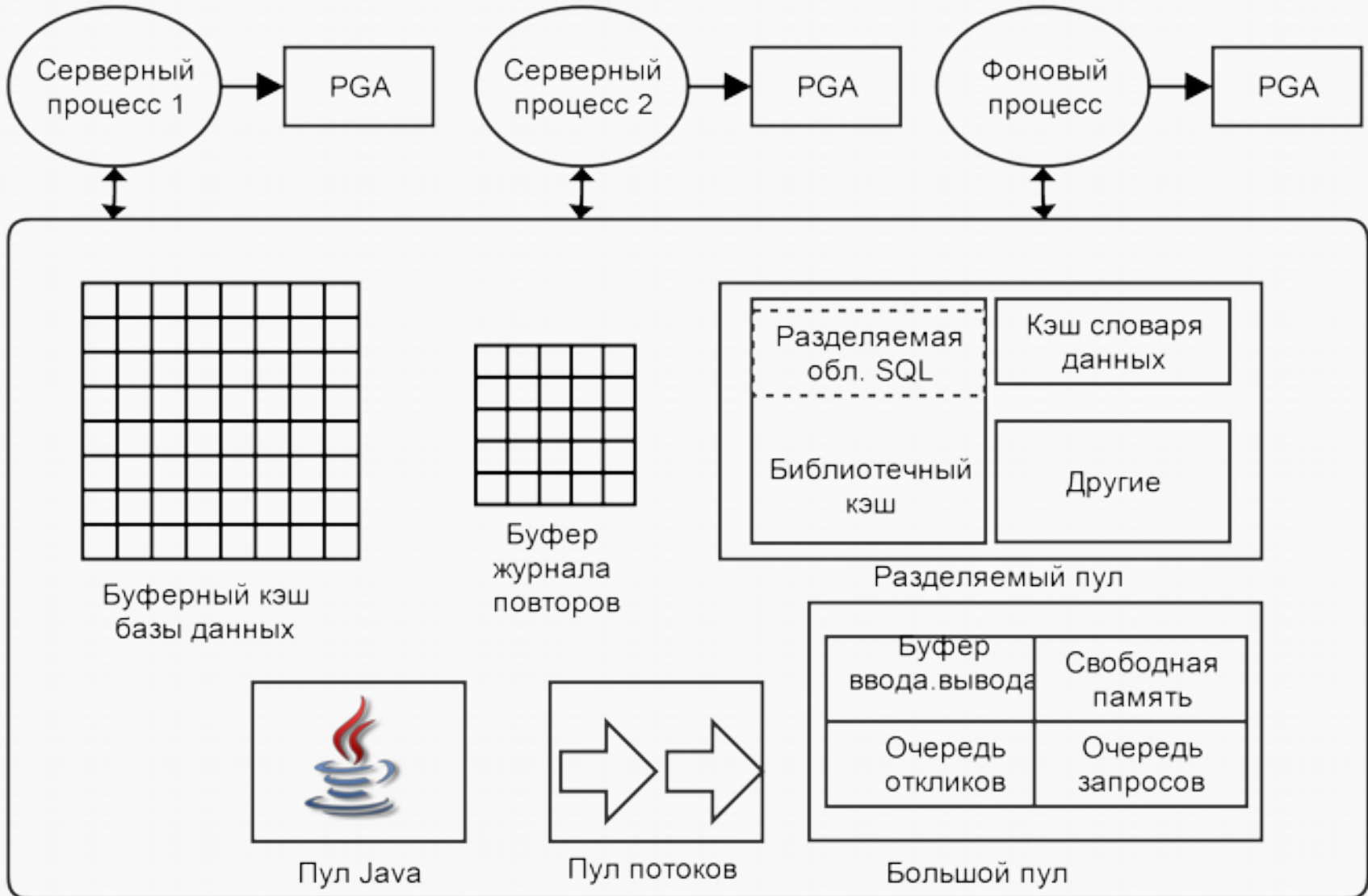


Структуры БД

Структуру БД можно рассматривать по-разному:

- На уровне структур в основной памяти ЭВМ.
- На уровне процессов в ОС.
- На уровне структуры хранилища данных в ФС.

Структуры памяти БД Oracle





Структуры памяти — буферный кэш БД (Cache Buffer)

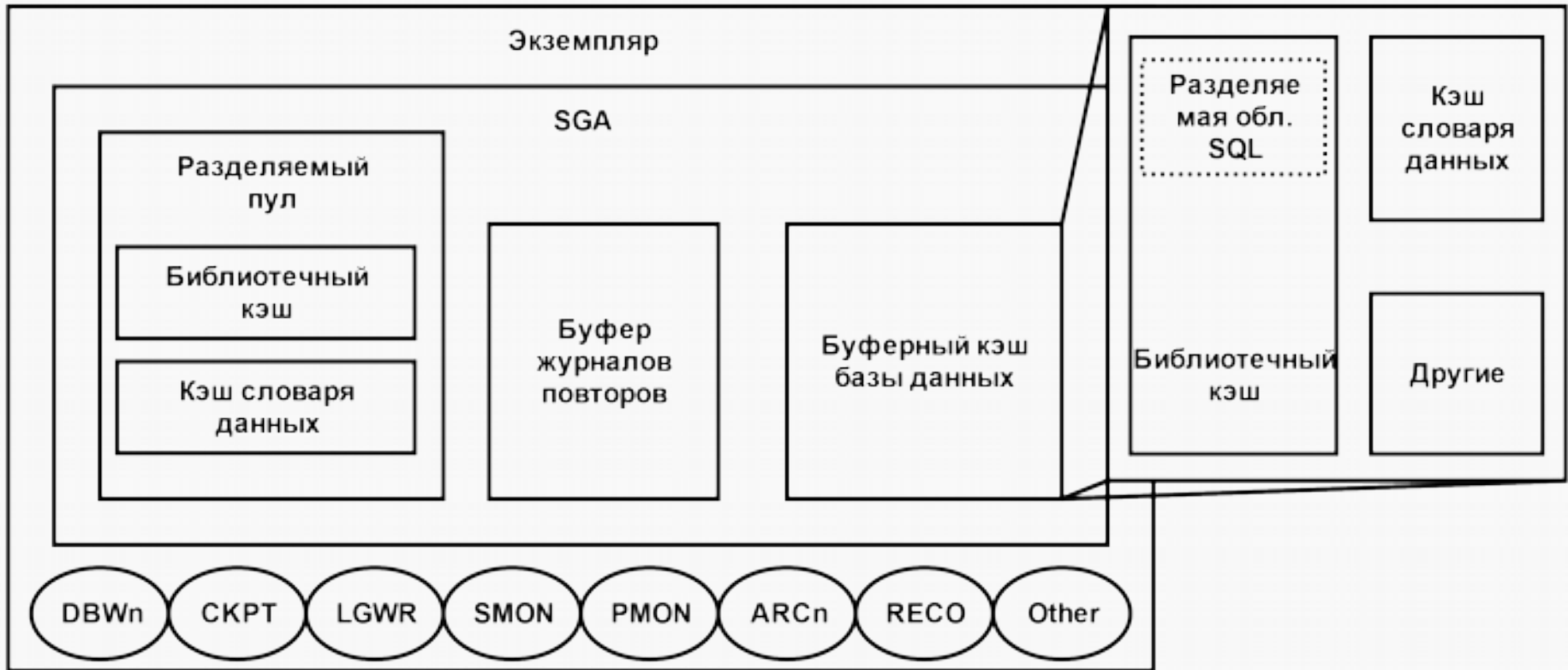
- Является частью SGA.
- Хранит копии блоков данных, считанных из файлов данных.
- Если нужного блока данных нет в кэше, он читается с диска и помещается в кэш.
- Совместно используется всеми параллельно работающими пользователями.
- Управляется сложным алгоритмом, основанным на LRU.

Структуры памяти — буфер журнала повторов (Redo Log Buffer)

- Циклический буфер в SGA.
- Хранит информацию об изменениях в БД.
- Содержит записи повторов, в которых хранится информация для повторного применения изменений, внесенных операциями DML и DDL.
- Записи повторов используются для восстановления базы данных в случае необходимости.
- Фоновый процесс LGWR производит запись буфера журнала повторов на диск.


Структуры памяти — разделяемый пул (Shared Pool)

- Область SGA.
- Структура пула:



Выделение памяти в разделяемом пуле

- Данные вытесняются из пула по алгоритму LRU.
- Серверный процесс проверяет разделяемый пул на предмет наличия разделяемой области SQL для идентичного оператора.
- Серверный процесс выделяет частную область SQL по запросу сеанса.
- В некоторых случаях разделяемая область SQL сбрасывается целиком:
`ALTER SYSTEM FLUSH SHARED_POOL;`



Структуры памяти — большой пул (Large Pool)

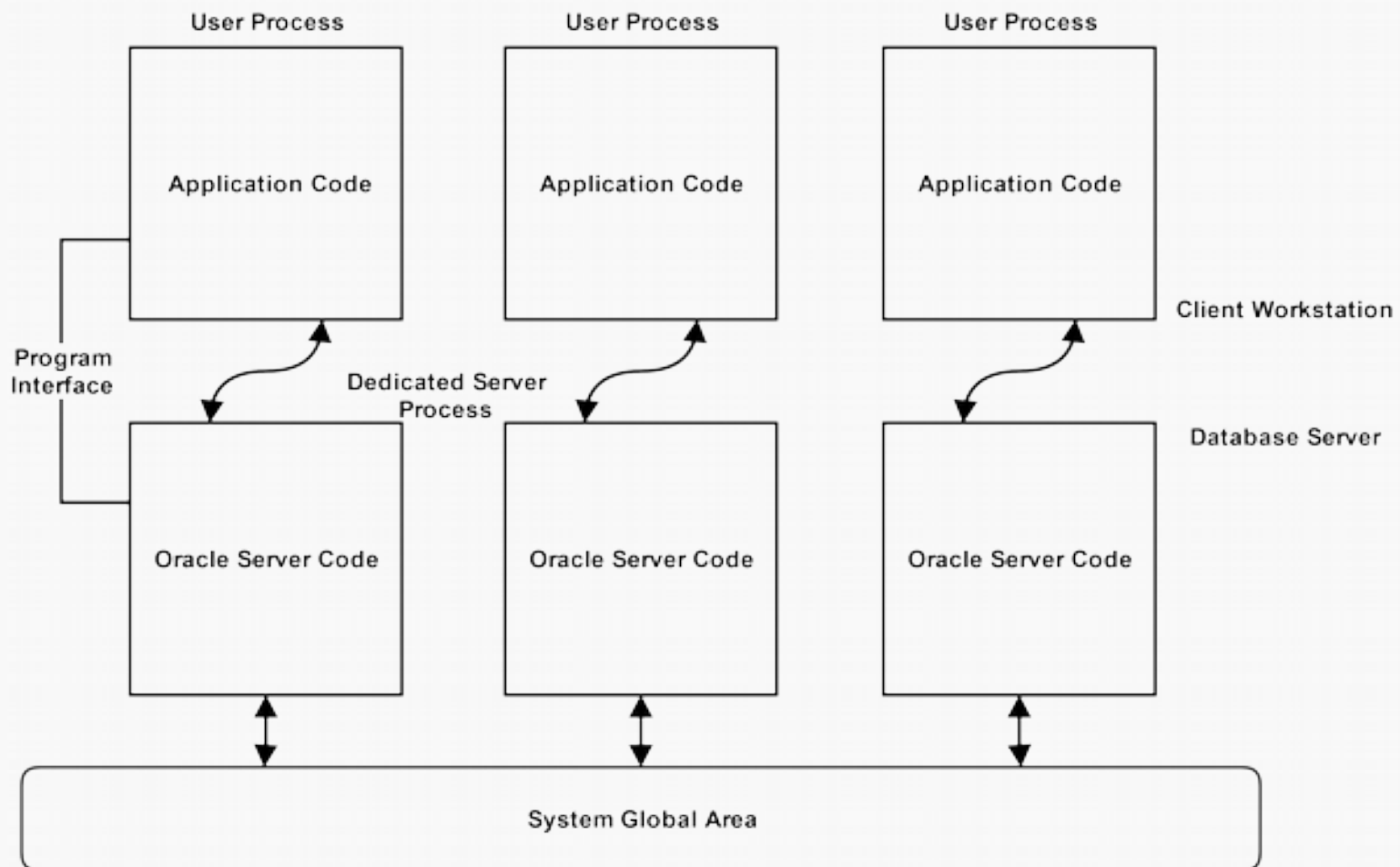
- Необязательная область SGA.
- Выделяется вручную администратором БД.
- В отличие от разделяемого пула, нет автоматического освобождения памяти по LRU.
- Может быть использован:
 - Для операций передачи большого объёма данных между разными БД.
 - Для операций резервного копирования / восстановления.
- Размер задаётся параметром инициализации `LARGE_POOL_SIZE`.

2 вида процессов:

- Пользовательские процессы. Запускаются в момент подключения пользователя к БД.
- Процессы базы данных:
 - Серверный процесс: подключается к экземпляру Oracle и запускается при установлении сеанса пользователем.
 - Фоновые процессы: запускаются при запуске экземпляра Oracle.

Архитектура процессов — выделенный сервер (Dedicated Server)

В режиме выделенного сервера каждому пользовательскому процессу создаётся свой «персональный» серверный.

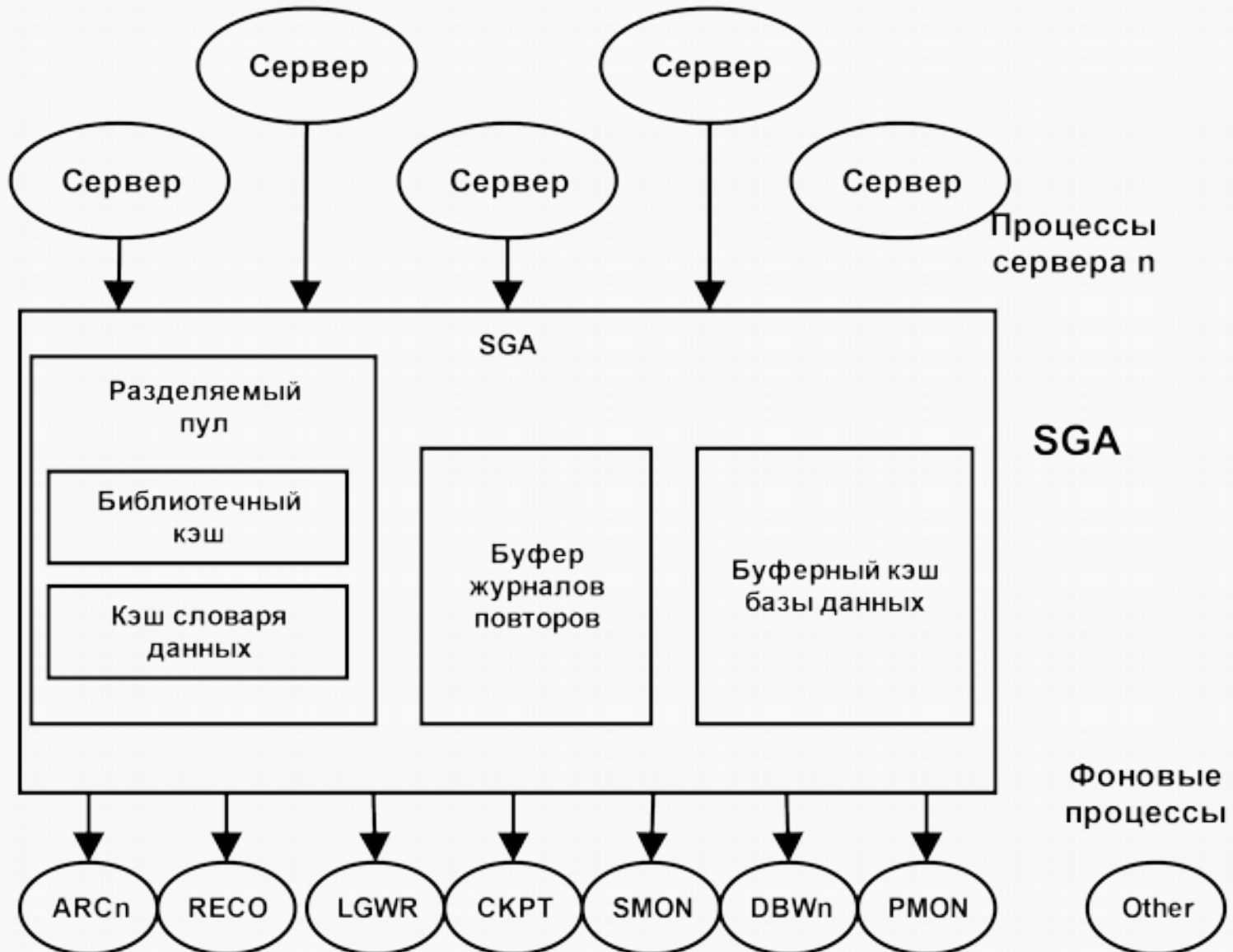


Архитектура процессов — разделяемый сервер (Shared Server)

В режиме разделяемого сервера каждому серверные процессы выделяются диспетчером из специального пула.



Структуры процессов



Процесс записи в БД (DBWn)

Записывает измененные (заполненные) буферы из буферного кэша базы данных на диск:

- асинхронно во время выполнения другой обработки;
- периодически для перехода к следующей контрольной точке.

Администратор может сконфигурировать СУБД на использование до 20 параллельных процессов DBW с помощью параметра инициализации `DB_WRITER_PROCESSES`.

Процесс записи в БД (DBW_л, продолжение)

- Изменения записываются в файлы в том порядке, в котором они были сделаны (согласно SCN — System Change Number).
- Для этого используется LRUW (LRU-Write) — список заполненных буферов в кэше, отсортированный по SCN.
- При записи данных в файл DBW одновременно «перемещает» указатель на контрольную точку, с которой будет начато восстановление в случае сбоя (инкрементальная установка контрольных точек).

Процесс LogWriter (LGWR)

Записывает буфер журнала повторов в файл журнала повторов на диске.

Запись осуществляется:

- когда пользовательский процесс фиксирует транзакцию;
- когда буфер журнала повторов заполняется на одну треть;
- перед тем как процесс DBWn записывает измененные буферы на диск.

После того, как данные из буфера журнала повторов записаны на диск, серверные процессы могут записать на их место новые данные.

Процесс LogWriter (LGWR, продолжение)

- Данные в файл журнала повторов записываются *сразу же* после того, как пользователь вызвал оператор COMMIT. Т.е., данные в журнал повторов обычно записываются *раньше*, чем в файлы данных.
- Это называется механизмом *быстрой фиксации транзакции*.



Процесс создания контрольной точки (СКРТ)

- *Контрольная точка* – это структура данных, определяющая SCN в потоке повторов базы данных.
- Контрольные точки записываются в управляющий файл и в заголовок каждого из файлов данных. Эту операцию выполняет *процесс СКРТ*.
- Процесс СКРТ не записывает блоки на диск, эту операцию выполняет процесс DBWn.
- Запись номеров SCN в заголовки файлов гарантирует, что все изменения, внесенные в блоки базы данных до фиксации данного номера SCN уже записаны на диск.

Процесс системного монитора (SMON)

- Выполняет восстановление при запуске экземпляра (если в этом есть необходимость).
- Выполняет очистку временных сегментов, которые больше не используются.
- Если во время восстановления экземпляра какая-либо из завершенных транзакций была пропущена (из-за ошибки чтения файла или ошибки автономного режима), SMON восстановит их при переводе табличного пространства или файла обратно в оперативный режим.

Процесс монитора процессов (PMON)

- Выполняет восстановление пользовательского процесса при сбое:
 - Выполняет очистку буферного кэша базы данных.
 - Освобождает ресурсы, использовавшиеся пользовательским процессом.
- Выполняет мониторинг времени ожидания при отсутствии действий в сеансах.
- Динамически регистрирует службы базы данных в процессах-слушателях (Network Listeners).
- Периодически проверяет состояние процессов диспетчера и сервера и перезапускает любой из этих процессов в случае его остановки.

Процесс восстановления (RECO)

- Используется при распределенной конфигурации БД.
- Автоматически подключается к другим БД, задействованным в сомнительных распределенных транзакциях.
- Автоматически разрешает все сомнительные транзакции.
- Удаляет все строки, соответствующие сомнительным транзакциям.

Процессы архиваторов (ARCn)

- Копируют файлы журнала повторов на указанное устройство хранения после заполнения журнала.
- Могут собирать данные для восстановления транзакций.
- Функционируют, только если БД работает в режиме ARCHIVELOG.
- Можно изменить максимальное количество процессов архиваторов при помощи параметра инициализации LOG_ARCHIVE_MAX_PROCESSES.



Серверный процесс и буферный кэш БД

Каждый буфер может находиться в одном из четырёх возможных состояний:

- **Закрепленный** (за сеансом). Другим сеансам запрещено одновременно выполнять запись в один и тот же блок — они должны ждать, пока он освободится.
- **Очищенный**. Буфер является незакрепленным и становится кандидатом на немедленное устаревание, если его текущее содержимое (блок данных) не будет запрошено еще раз.
- **Свободный / неиспользуемый**. Буфер пуст, так как экземпляр был только что запущен. Отличие от очищенного в том, что буфер еще не использовался.
- **Заполненный**. Буфер больше не закреплен, однако его содержимое (блок данных) изменилось, и процессу DBWn необходимо сбросить его на диск, прежде буфер можно будет объявить устаревшим.

Архитектура хранения БД

Файлы, из которых состоит БД, делятся на следующие категории:

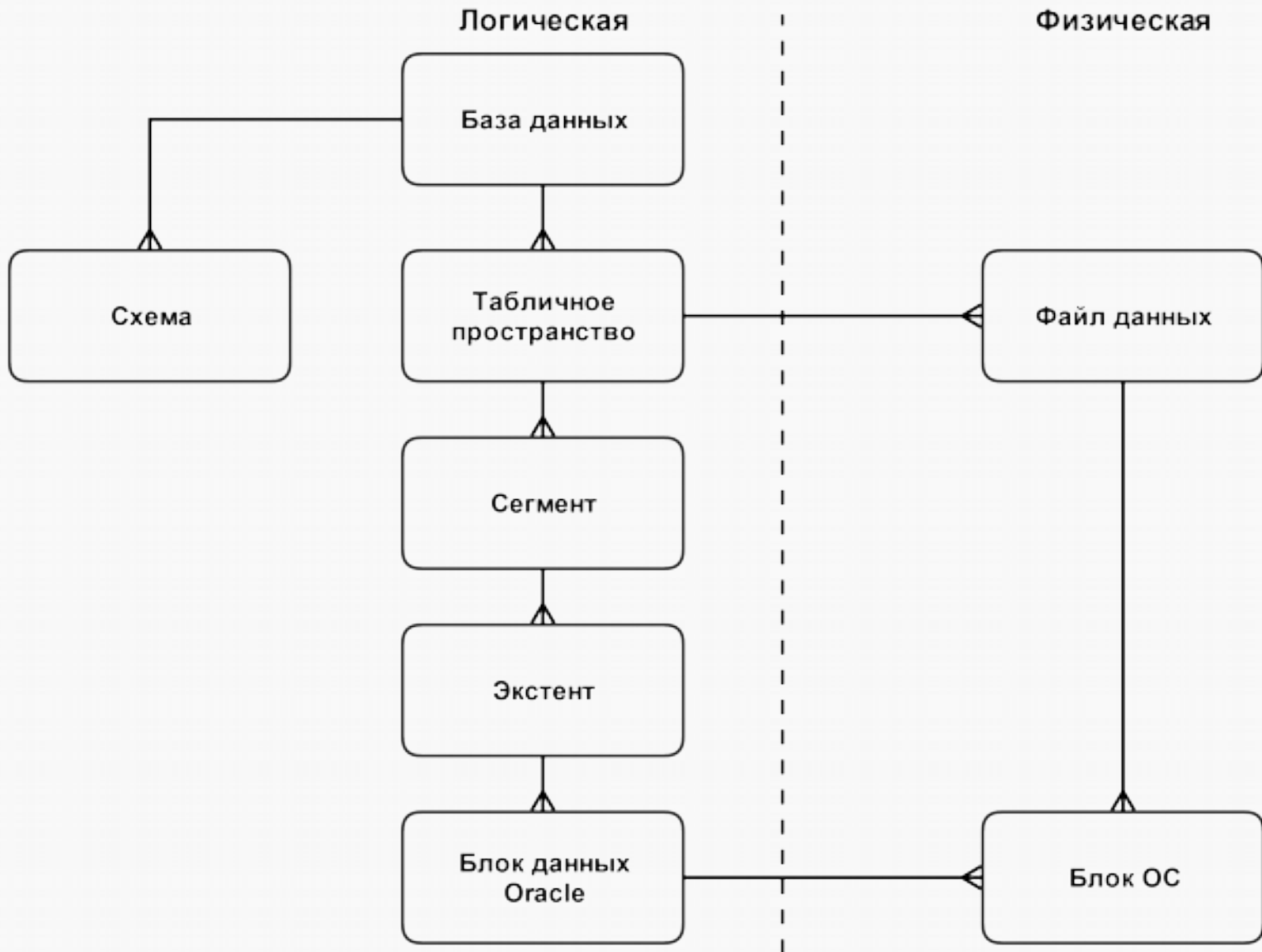
- **Управляющие файлы.** Содержат данные о самой базе данных (то есть информацию о физической структуре базы данных).
- **Файлы данных.** Содержат данные пользователя или приложения БД, а также метаданные и словарь данных.
- **Оперативные файлы журнала повторов.** Если происходит сбой сервера БД, при котором не теряются файлы данных, экземпляр позволит восстановить базу данных с помощью информации, содержащейся в этих файлах.

Архитектура хранения БД (продолжение)

Помимо перечисленных, экземпляр БД использует следующие файлы:

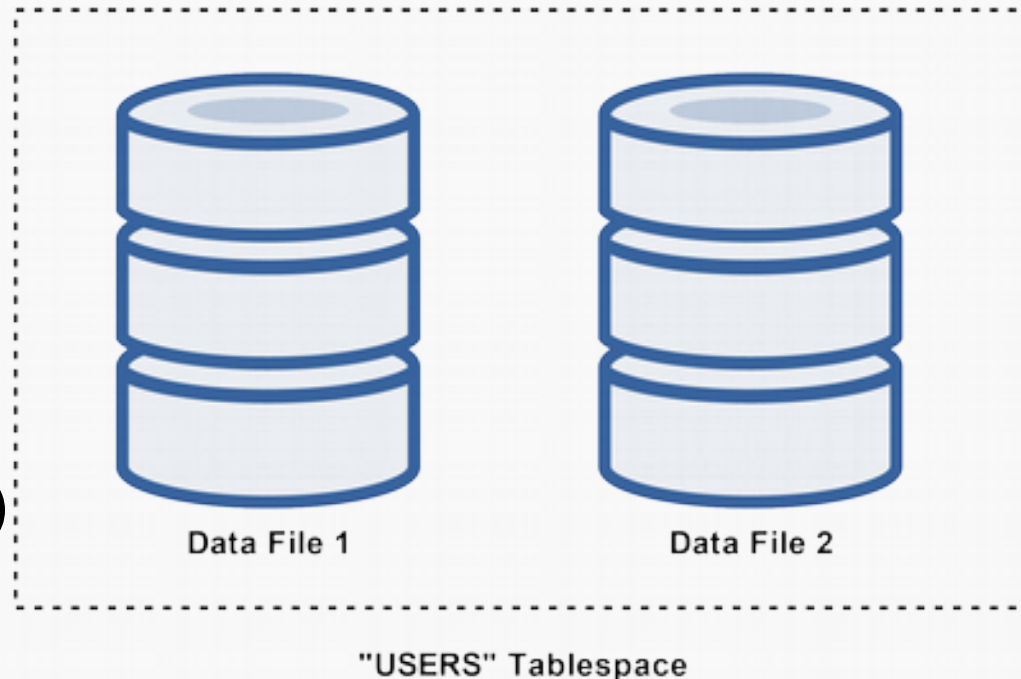
- **Файл параметров.** Используется для определения конфигурации экземпляра для запуска.
- **Файл паролей.** Позволяет удаленно подключаться к БД пользователям `sysdba`, `sysoper` и `sysasm`.
- **Резервные копии файлов.** Используются для восстановления БД.
- **Архивные файлы журнала повторов.** Содержат непрерывную историю изменений данных (повторных операций), которую создает экземпляр. С помощью этих файлов и резервной копии БД можно восстановить утраченные файлы данных.
- **Файлы трассировки.** Любой серверный или фоновый процесс может выполнять запись в определенный файл трассировки. При обнаружении процессом внутренней ошибки он записывает дамп информации об ошибке в свой файл трассировки.
- **Файлы журнала предупреждений.** Журнал предупреждений БД – это хронологический журнал сообщений и ошибок. Каждый экземпляр использует один файл журнала предупреждений.

Логические и физические структуры БД



Табличные пространства и файлы данных

- Каждая БД логически состоит из одного или более табличных пространств, являющихся логическими блоками хранения.
- Табличные пространства состоят из одного или нескольких файлов данных.
- Файл данных может принадлежать только одному табличному пространству.
- Табличное пространство может находиться в оперативном (доступном) и автономном (недоступном) режимах.

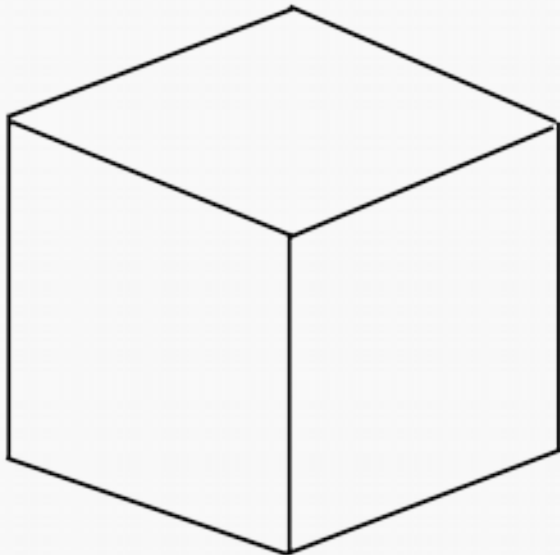


Табличные пространства SYSTEM и SYSAUX

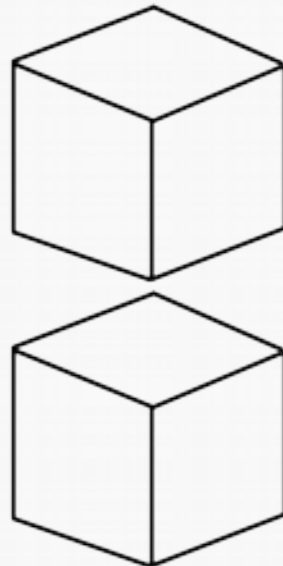
- Любая БД Oracle должна содержать табличные пространства SYSTEM и SYSAUX. Они автоматически создаются вместе с БД и всегда должны находиться в оперативном режиме.
- В **табличном пространстве SYSTEM** хранятся таблицы, обеспечивающие основные функции базы данных, например, таблицы словаря данных.
- **Табличное пространство SYSAUX** является вспомогательным. В нем хранится множество компонентов БД, например, репозиторий Enterprise Manager.

Сегменты, экстененты и блоки

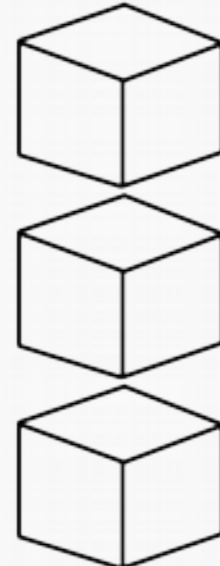
- Сегменты существуют в табличном пространстве.
- Сегмент – это набор экстенентов.
- Экстенент – это набор блоков данных.
- Блоки данных связаны с дисковыми блоками.



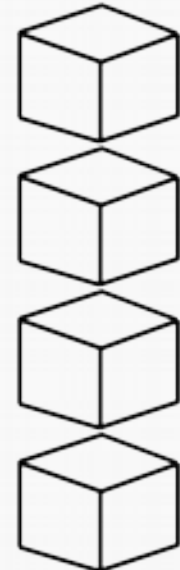
Сегмент



Экстененты



Блоки данных



Дисковые блоки

3. Установка Oracle и создание БД

Переменные окружения

Для корректной работы Oracle настоятельно рекомендуется перед установкой задать значения ряду переменных окружения:

- ORACLE_BASE. Устанавливает путь к «корню» иерархии каталогов Oracle. Например:
`export ORACLE_BASE=/u01/app/oracle`
- ORACLE_HOME. Устанавливает путь к «корневому» каталогу БД. Этот путь свой для каждого экземпляра БД. Пример:
`export ORACLE_HOME=$ORACLE_BASE/product/11.1.0/db_1`
- ORACLE_SID. Задаёт имя экземпляра Oracle. Значение по умолчанию - ORCL. Формат — строка, состоящая из цифр и букв и начинающаяся с буквы.
- NLS_LANG. Устанавливает язык и кодировку БД. Формат - язык_местность.набор символов, например:
`export NLS_LANG=RUSSIAN_CIS.AL32UTF8`

Способы установки Oracle

- С помощью Oracle Universal Installer — в интерактивном режиме с помощью графической утилиты (написанной на Java):
`./runInstaller`
- «Silent Mode» — с помощью файла конфигурации, (Response File) заданного в ходе одной из предыдущих установок:
`./runInstaller -record -responseFile
./runInstaller -silent -responseFile
responsefilename`

После завершения работы OUI необходимо выполнить ряд скриптов из-под суперпользователя:

```
$ su
# password:
# cd /u01/app/oracle/oraInventory
# ./oraInstRoot.sh
# cd /u01/app/oracle/product/11.1.0/db_1
# ./root.sh
```

- Универсальный подход (шаблон) к конфигурированию СУБД (не только Oracle).
- Описывает структуру каталогов БД и других ресурсов в ФС.
- Как и остальные шаблоны, предназначен для построения максимально гибкой структуры экземпляра БД и избежания возможных «типовых» проблем.
- Не является обязательным.

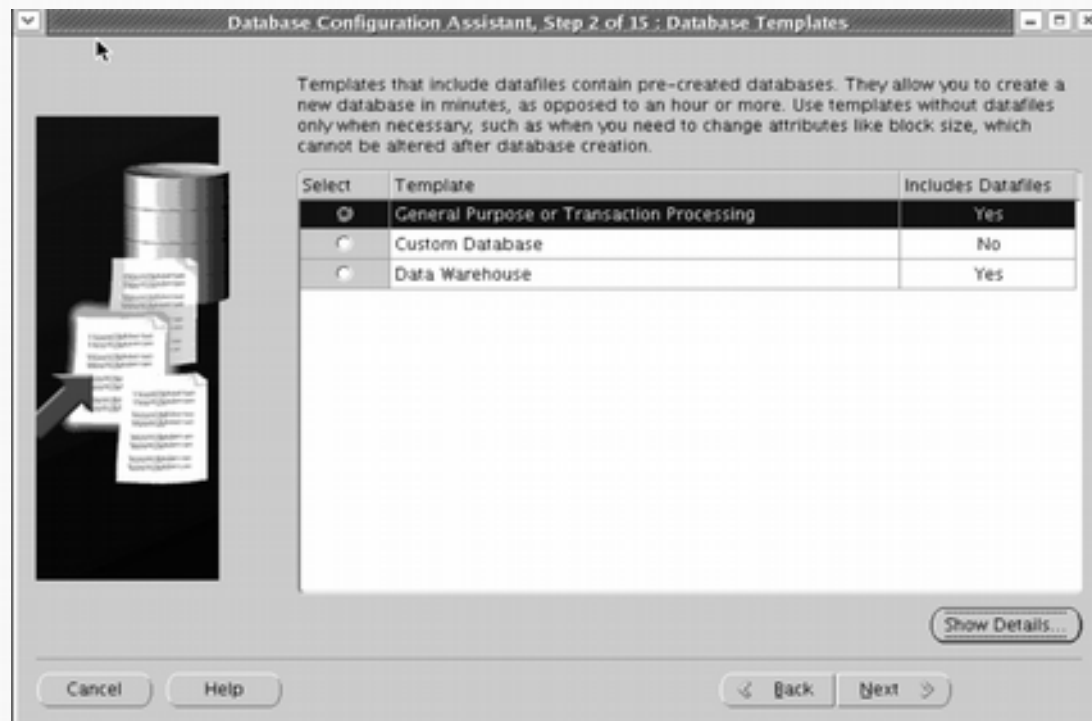
Имена каталогов в соответствии с OFA



- Точки монтирования — /рм (например, /u01 или /u02).
- Домашние каталоги — /рм/н/и (например, /u01/app/oracle или /u02/home/oracle).
- Каталоги бинарных файлов СУБД — /рм/н/и/product/v (например, /u01/app/oracle/product/11.1.0/db_1).
- Каталоги с конфигурационными и другими административными файлами — /рм/н/и/admin/d/a (например, /u01/app/oracle/admin/db01/arch).
- Файлы БД:
 - Управляющие файлы: /рм/q/d/controlnctl.
 - Файлы журнала повторов: /рм/q/d/redon.log.
 - Файлы данных: /рм/q/d/tn.dbf.

Способы создания БД

- «Вручную» — в SQL*Plus, с помощью команды CREATE DATABASE.
- С помощью графической утилиты Database Configuration Assistant (DBCA):



1. Задаём Oracle System Identifier (SID):

```
ORACLE_SID=mynewdb  
export ORACLE_SID
```

2. Ещё раз проверяем, задана ли переменная окружения ORACLE_HOME:

```
echo $ORACLE_HOME  
/u01/app/oracle/product/11.1.0/db_1
```

3. Выбираем метод аутентификации, который будет использоваться в БД:

- с помощью файла паролей;
- на уровне пользователей ОС.

Файл паролей Oracle

Конфигурируется с помощью утилиты ORAPWD:

```
ORAPWD FILE=filename [ENTRIES=numusers] [FORCE={Y|N}]  
[IGNORECASE={Y|N}]
```

Аргументы:

- FILE — имя файла паролей.
- ENTRIES — максимальное количество аккаунтов пользователей (если аргумент не задан, количество не ограничено).
- FORCE — если Y, то существующий файл паролей будет перезаписан.
- IGNORECASE — если Y, то при проверке правильности ввода пароля будет игнорироваться регистр символов.

Файлы параметров инициализации БД

- Считываются экземпляром Oracle при его запуске.
- Существует 2 типа файлов параметров:
 - Файлы параметров сервера (SPFILE) — двоичный файл, чтение и запись в который осуществляет сервер БД. Самостоятельно изменять этот файл нельзя. Имя по умолчанию — `spfile<SID>.ora`.
 - Текстовый файл параметров инициализации — может быть только считан сервером, но не записан. Настройки параметров инициализации необходимо задавать и изменять вручную (с помощью текстового редактора). Имя по умолчанию — `init<SID>.ora`. При наличии SPFILE этот файл игнорируется.

Создание БД (продолжение)

4. Создаём текстовый файл с параметрами инициализации БД. Он должен содержать как минимум 3 параметра:

- `DB_NAME` — имя БД (максимум 8 символов).
- `CONTROL_FILES` — список управляющих файлов БД.
- `MEMORY_TARGET` — общее количество памяти, которое будет выделено экземпляру БД.

Параметры инициализации БД

Существует два типа параметров инициализации:

- Статические параметры — могут быть изменены только путем редактирования файлов `init.ora` и `SPFILE`. Чтобы изменения вступили в силу, необходимо остановить и перезапустить БД.
- Динамические параметры — могут изменяться при работающей БД. Существует два типа динамических параметров:
 - Параметры уровня сеанса — оказывают влияние только на текущий сеанс. Пример — параметры поддержки национальных языков (NLS).
 - Параметры уровня системы — оказывают влияние на всю базу данных и все сеансы. Пример — параметры, отвечающие за сброс данных разделяемого пула или параметры расположения архивного журнала. Действие данных параметров зависит от настройки `SCOPE.x`.

Динамические параметры можно изменять с помощью команд `ALTER SESSION` и `ALTER SYSTEM`:

```
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'mon dd yyyy';
```

Параметры инициализации БД (продолжение)

Некоторые дополнительные параметры инициализации БД:

- `DB_FILES` — максимальное количество файлов БД.
- `PROCESSES` — максимальное количество параллельных пользовательских процессов.
- `DB_BLOCK_SIZE` — размер блока данных БД (в байтах; по умолчанию — 8 КБ).
- `DB_CACHE_SIZE` — размер блока буферного кэша БД (в байтах; по умолчанию — 48 МБ для однопроцессорной системы).
- `SGA_TARGET` — общий размер SGA (в байтах).



Пример файла параметров инициализации

```
db_name='ORCL'  
memory_target=1G  
processes = 150  
db_block_size=8192  
db_domain=cs.ifmo.ru  
db_recovery_file_dest='<ORACLE_BASE>/flash_recovery_area'  
db_recovery_file_dest_size=2G  
diagnostic_dest='<ORACLE_BASE>'  
dispatchers='(PROTOCOL=TCP) (SERVICE=ORCLXDB)'  
open_cursors=300  
remote_login_passwordfile='EXCLUSIVE'  
undo_tablespace='UNDOTBS1'  
  
# You may want to ensure that control files are created on  
# separate physical  
# devices  
control_files = (ora_control1, ora_control2)  
compatible = '12.0.0'
```

Создание БД (продолжение)

5. Создаём экземпляр Oracle (только для Windows, в *NIX экземпляр создаётся автоматически):

```
oradim -NEW -SID sid -STARTMODE MANUAL -PFILE file
```

6. Запускаем SQL*Plus без подключения к БД:

```
$ sqlplus /nolog
```

7. Подключаемся к экземпляру с привилегиями SYSDBA:

```
SQL> CONNECT SYS AS SYSDBA
```

или

```
SQL> CONNECT / AS SYSDBA
```

8. Создаём бинарный файл параметров инициализации сервера на основании созданного ранее текстового:

```
SQL> CREATE SPFILE FROM PFILE;
```

Создание БД (продолжение)

9. Запускаем экземпляр Oracle без монтирования БД:

```
SQL> STARTUP NOMOUNT;
```

10. Вызываем команду CREATE DATABASE. Возможны 2 варианта вызова:

- Перечисляем все параметры конфигурации БД в аргументах;
- Все параметры конфигурации БД читаются из файла.

Команда CREATE DATABASE

Пример вызова:

```
CREATE DATABASE mynewdb
  USER SYS IDENTIFIED BY sys_password
  USER SYSTEM IDENTIFIED BY system_password
  LOGFILE GROUP 1 ('/u01/logs/my/redo01a.log', '/u02/logs/my/redo01b.log')
                SIZE 100M BLOCKSIZE 512,
  GROUP 2 ('/u01/logs/my/redo02a.log', '/u02/logs/my/redo02b.log')
                SIZE 100M BLOCKSIZE 512,
  GROUP 3 ('/u01/logs/my/redo03a.log', '/u02/logs/my/redo03b.log')
                SIZE 100M BLOCKSIZE 512

  MAXLOGHISTORY 1
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 1024
  CHARACTER SET AL32UTF8
  NATIONAL CHARACTER SET AL16UTF16
  EXTENT MANAGEMENT LOCAL
```

Команда CREATE DATABASE (продолжение)

```
DATAFILE '/u01/app/oracle/oradata/mynewdb/system01.dbf'  
  SIZE 700M REUSE AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED  
SYSAUX DATAFILE '/u01/app/oracle/oradata/mynewdb/sysaux01.dbf'  
  SIZE 550M REUSE AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED  
DEFAULT TABLESPACE users  
  DATAFILE '/u01/app/oracle/oradata/mynewdb/users01.dbf'  
  SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED  
DEFAULT TEMPORARY TABLESPACE tempts1  
  TEMPFILE '/u01/app/oracle/oradata/mynewdb/temp01.dbf'  
  SIZE 20M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE UNLIMITED  
UNDO TABLESPACE undotbs1  
  DATAFILE '/u01/app/oracle/oradata/mynewdb/undotbs01.dbf'  
  SIZE 200M REUSE AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED  
USER_DATA TABLESPACE usertbs  
  DATAFILE '/u01/app/oracle/oradata/mynewdb/usertbs01.dbf'  
  SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
```




Команда CREATE DATABASE (продолжение)

Большую часть параметров конфигурации БД можно сохранить в файле:

```
DB_CREATE_FILE_DEST='/u01/app/oracle/oradata'
```

В этом случае команда создания БД выглядит проще:

```
CREATE DATABASE mynewdb  
USER SYS IDENTIFIED BY sys_password  
USER SYSTEM IDENTIFIED BY system_password  
EXTENT MANAGEMENT LOCAL  
DEFAULT TEMPORARY TABLESPACE temp  
UNDO TABLESPACE undotbs1  
DEFAULT TABLESPACE users;
```

11. Создаём пользовательские и дополнительные табличные пространства:

```
CREATE TABLESPACE apps_tbs LOGGING
  DATAFILE
    '/u01/app/oracle/oradata/mynewdb/apps01.dbf'
  SIZE 500M REUSE AUTOEXTEND ON NEXT 1280K
  MAXSIZE UNLIMITED
  EXTENT MANAGEMENT LOCAL;
CREATE TABLESPACE indx_tbs LOGGING
  DATAFILE
    '/u01/app/oracle/oradata/mynewdb/indx01.dbf'
  SIZE 100M REUSE AUTOEXTEND ON NEXT 1280K
  MAXSIZE UNLIMITED
  EXTENT MANAGEMENT LOCAL;
```

Создание БД (продолжение)

12. Заполняем первичными данными представления словаря данных:

```
@?/rdbms/admin/catalog.sql  
@?/rdbms/admin/catproc.sql  
@?/sqlplus/admin/pupbld.sql
```

13 (...). Настраиваем резервное копирование БД, автозапуск БД при рестарте сервера, и т.д.

4. Словарь данных Oracle

Что такое словарь данных?

Словарь данных (Data Dictionary, DD) — это набор доступных только для чтения таблиц и представлений, которые содержат различную информацию о БД:

- Информацию обо всех объектах схемы БД (таблицах, представлениях, индексах, кластерах, синонимах, последовательностях, процедурах, функциях, пакетах, триггерах и т. д.).
- Информацию о том, сколько дискового пространства выделено объектам схемы, и какой процент этого пространства уже использован.
- Информацию о значениях полей таблиц по умолчанию.
- Информацию о существующих в БД ограничениях целостности.
- Сведения обо всех пользователях БД, их ролях и выданных привилегиях.
- Результаты текущего аудита — кто в данный момент имеет обращается и/или модифицирует объекты схемы.

Структура словаря данных

Словарь данных состоит из двух «уровней»:

- Системные таблицы — содержат информацию в логичном с точки зрения архитектуры системы виде.
- Представления — содержат ту же самую информацию в более удобном для чтения и обработки формате.

Представления словаря данных

Каждое представление имеет префикс, характеризующий, что за информация в нём содержится:

DBA_XXX — все объекты во всех схемах БД

ALL_XXX — объекты доступные текущему пользователю

USER_XXX -
объекты принадлежащие текущему пользователю

Популярные представления:

- CAT, USER_CATALOG, ALL_CATALOG, DBA_CATALOG.
- TAB, TABS, USER_TABLES, ALL_TABLES, DBA_TABLES.
- DICT, DICTIONARY — описание таблиц и представлений словаря.
- IND, USER_INDEXES, ALL_INDEXES, DBA_INDEXES.
- XXX_SNAPSHOTS, XXX_DB_LINKS, XXX_CLUSTERS, XXX_ERRORS, XXX_SOURCES, XXX_DEPENDENCIES...
- Подробный список
http://citforum.ru/database/oraclepr/oraclepr_15.shtml

Таблица DUAL

- Таблица в словаре данных, которая состоит из одного столбца с именем DUMMY и одной строки со значением x.
- Используется в случаях, когда нужно проверить работоспособность БД — результат запроса к этой таблице всегда заранее известен:

```
SQL> desc dual
```

Name	Null?	Type

DUMMY		VARCHAR2(1)

```
SQL> select * from dual;
```

```
D  
-  
X
```

- Также может использоваться в качестве таблицы-«заглушки», т. к. она всегда существует:

```
SQL> select user from dual;
```

```
USER  
-----  
SCOTT
```

Dynamic Performance Views

Представление производительности:

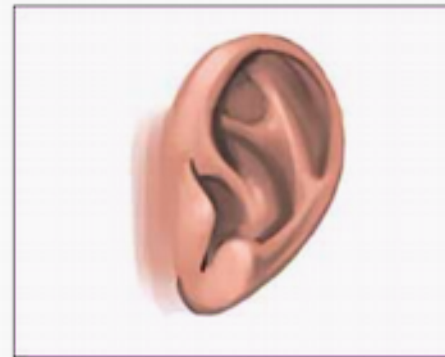
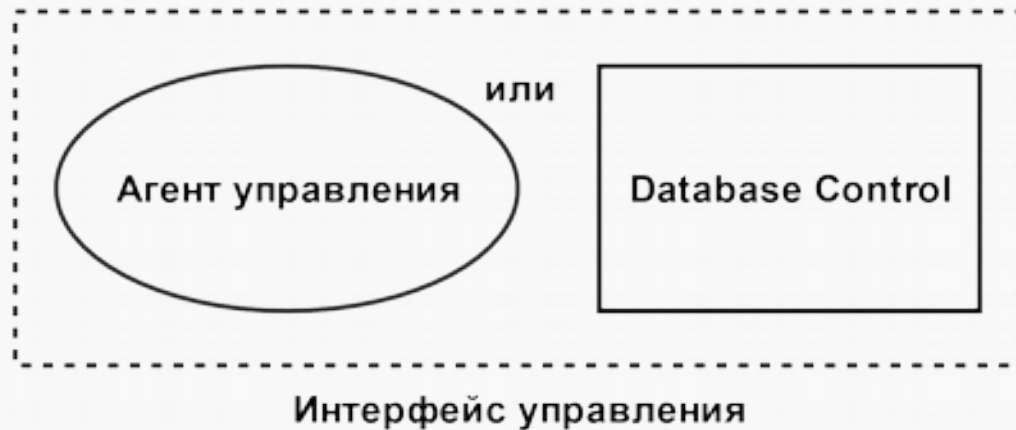
- Отражают текущее состояние экземпляра БД.
- Информация динамически генерируется в зависимости от состояния.
- Префикс V\$ - представления производительности экземпляра БД.
- Префикс GV\$ - глобальные представления узлов кластера RAC.
- Подробный список:
http://citforum.ru/database/oraclepr/oraclepr_15.shtml

Dynamic Performance Views

Полезные представления:

- V\$SYSTEM_EVENT — содержит общесистемную информацию о ресурсах, которых ждет весь экземпляр.
- V\$SESSION_EVENT — список событий, которые приходилось ждать в каждом сеансе.
- V\$SESSION_WAIT — детальная посеансовая информация о ресурсах, которые сеанс ожидает в данный момент или ждал в последний раз.
- V\$SESSION — информация о сеансе, в том числе о событии, которое ожидает сеанс в данный момент или ждал в последний раз
- Подробный список:
http://citforum.ru/database/oraclepr/oraclepr_15.shtml

5. Управление экземпляром Oracle



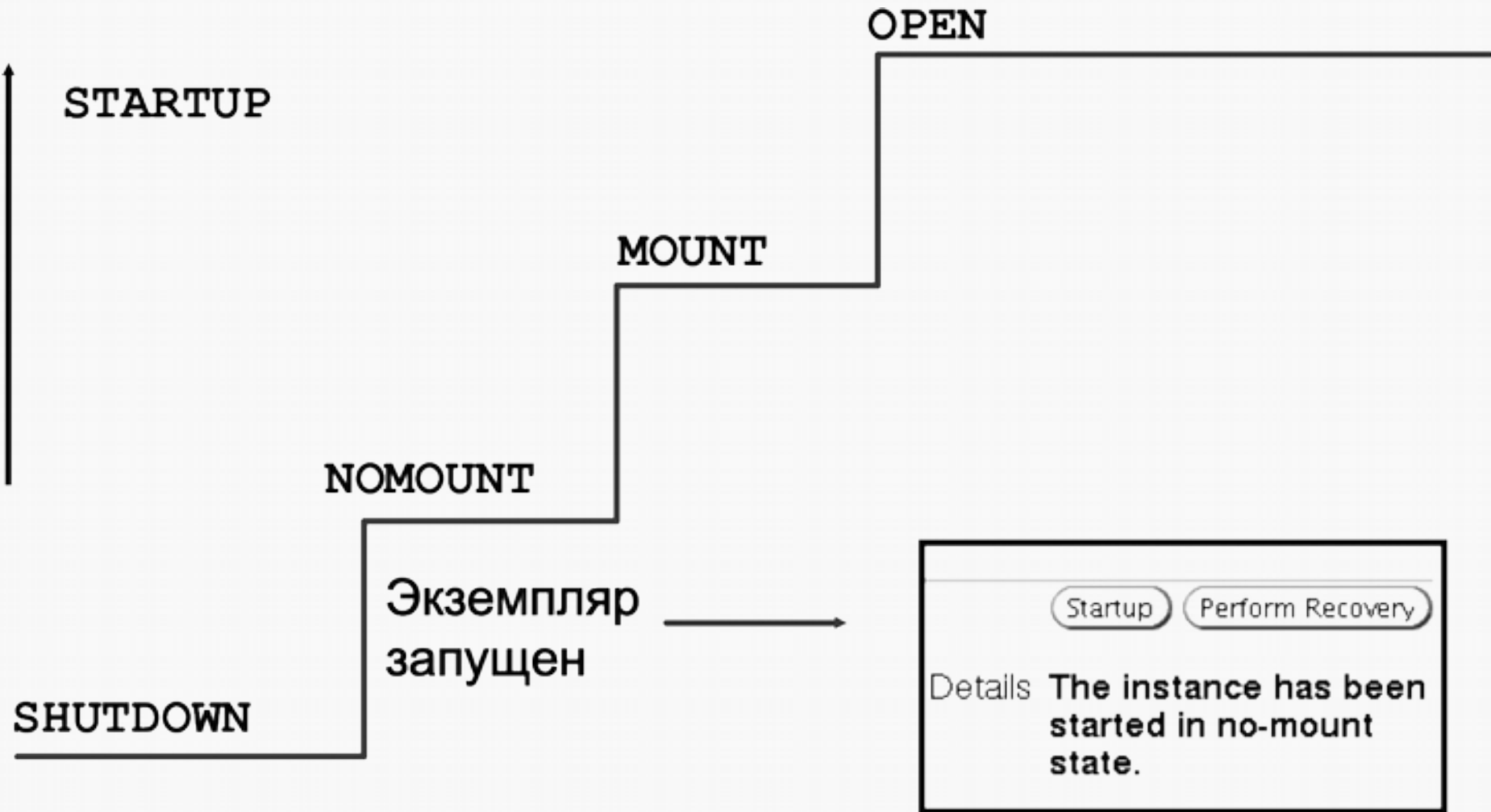
Прослушиватель



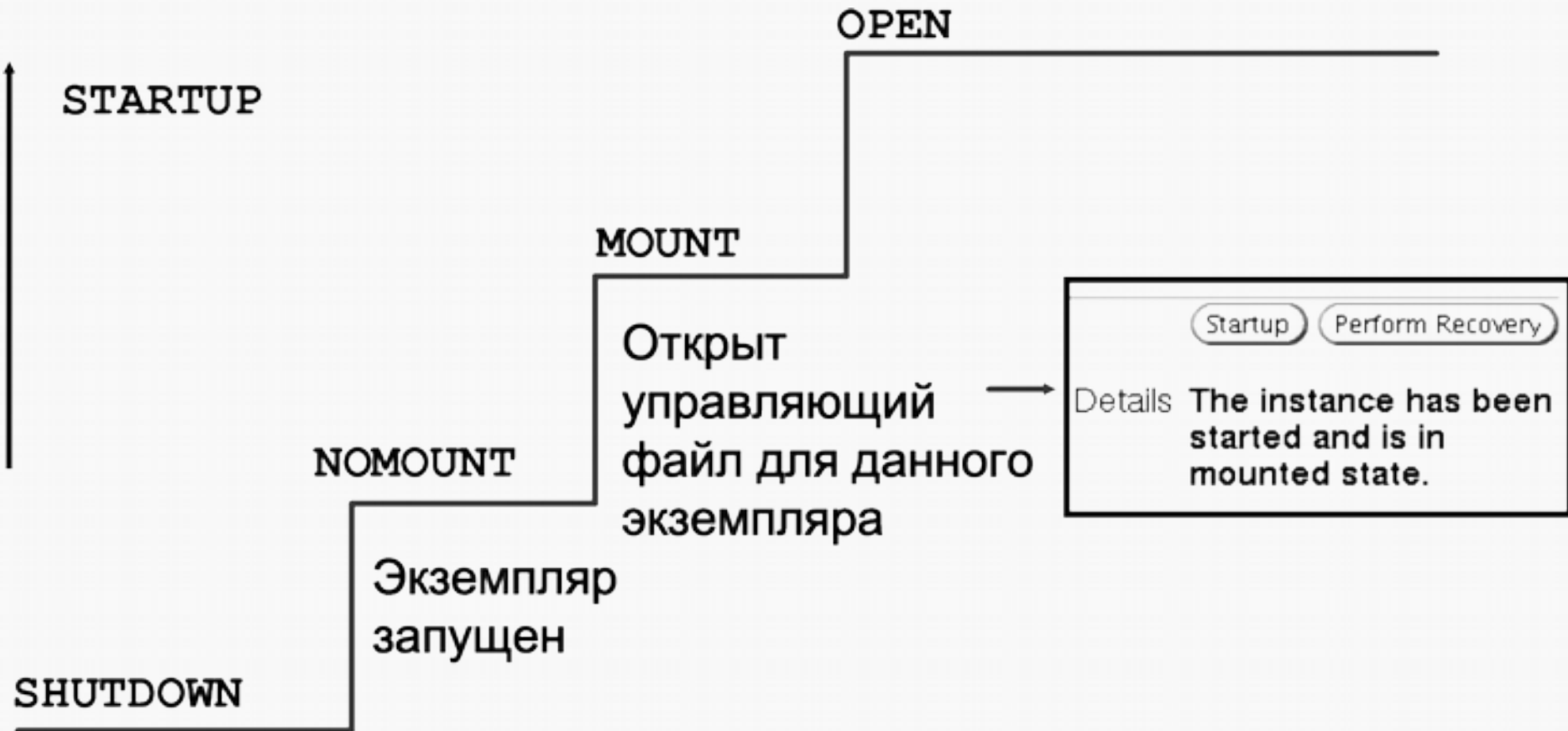
Интерфейс управления:

- Oracle Enterprise Manager — веб-интерфейс.
- SQL*Plus — утилита командной строки.

Запуск экземпляра Oracle - NOMOUNT



Запуск экземпляра Oracle - MOUNT

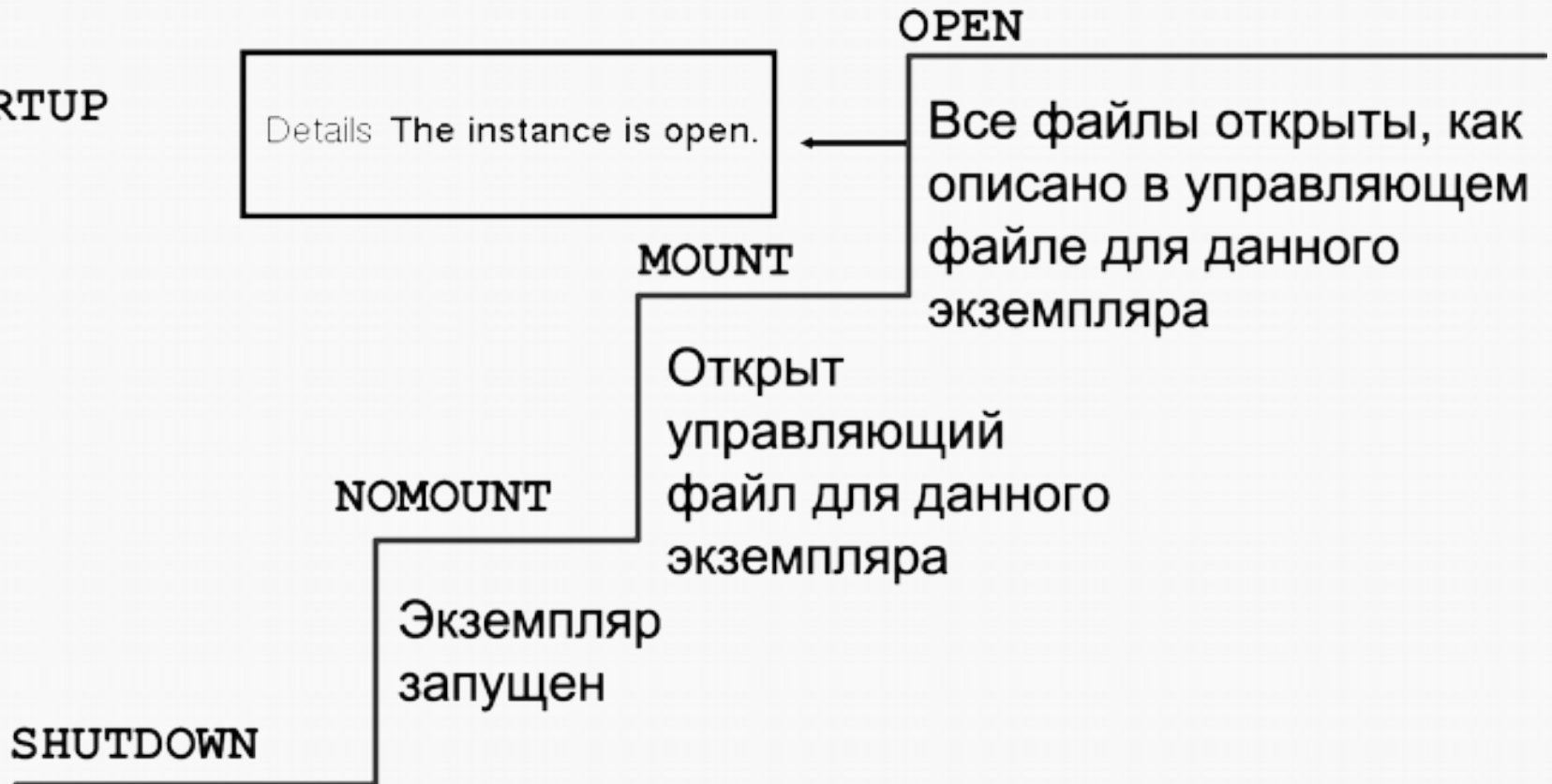


Монтирование БД

При монтировании БД выполняются следующие операции:

- БД ассоциируется с ранее запущенным экземпляром.
- Выполняется поиск и открытие управляющих файлов, указанных в файле параметров.
- Производится чтение управляющих файлов для получения сведений об именах и состоянии файлов данных и интерактивных файлах журналов повтора.

Запуск экземпляра Oracle - OPEN



Открытие БД

При открытии БД выполняются следующие операции:

- Открытие файлов данных.
- Открытие оперативных файлов журнала повторов.

Команды запуска экземпляра Oracle

- SQL> startup
Запускает экземпляр, ассоциирует с ним файлы БД, а затем монтирует и открывает БД.
- SQL> startup nomount
Запускает экземпляр без монтирования БД.
- SQL> alter database mount;
Монтирует и открывает БД, находящуюся в состоянии NOMOUNT.

2 способа:

- Через веб-интерфейс Enterprise Manager.
- «Вручную» — с помощью SQL*Plus:
SQL> shutdown
SQL> shutdown transactional
SQL> shutdown immediate
SQL> shutdown abort

Остановка экземпляра Oracle (продолжение)

Режим остановки	A	I	T	N
Разрешены новые подключения	No	No	No	No
Ожидание завершения текущего сеанса	No	No	No	Yes
Ожидание завершения текущей транзакции	No	No	Yes	Yes
Принудительное создание контрольной точки и закрытие файлов	No	Yes	Yes	Yes

Режимы остановки:

A - Abort

I - Immediate

T - Transactional

N - Normal



Мониторинг состояния экземпляра Oracle

Есть несколько способов:

- Графические утилиты в составе Enterprise Manager.
- Журнал предупреждений (Alert Log):
`$ORACLE_BASE/diag/rdbms/
<db_name>/<SID>/trace/alert_<SID>.log`
- Файлы трассировки.
- Представления словаря данных (Dynamic Performance Views).

Хронологический журнал сообщений о:

- Использованных при загрузке нестандартных параметрах инициализации.
- Всех случившихся внутренних ошибках (ORA-600), ошибках о повреждениях блоков (ORA-1578), а также ошибках взаимных блокировок (ORA-60).
- Операциях администрирования (SQL-операторы CREATE, ALTER, DROP DATABASE / TABLESPACE, операторы Enterprise Manager или SQL*Plus STARTUP, SHUTDOWN, ARCHIVE LOG и RECOVER).
- Сообщениях и ошибках, связанных с функциями разделяемого сервера процессов диспетчера.
- Ошибках, возникших при автоматическом обновлении материализованного представления.



Журнал предупреждений БД (продолжение)

- Журнал можно просматривать через интерфейс Enterprise Manager, либо напрямую в файле.
- Журнал сохраняется в двух форматах — XML и plain text.
- Путь к файлам журнала можно узнать, используя DPV V\$DIAG_INFO:

```
SQL> SELECT * FROM V$DIAG_INFO;
INST_ID NAME                                VALUE
-----
1 Diag Enabled                             TRUE
1 ADR Base                                  /u01/oracle
1 ADR Home                                  /u01/oracle/diag/rdbms/orclbi/orclbi
1 Diag Trace                               /u01/oracle/diag/rdbms/orclbi/orclbi/trace
1 Diag Alert                               /u01/oracle/diag/rdbms/orclbi/orclbi/alert
1 Diag Incident                             /u01/oracle/diag/rdbms
                                           /orclbi/orclbi/incident
1 Diag Cdump                                /u01/oracle/diag/rdbms/orclbi/orclbi/cdump
1 Health Monitor                            /u01/oracle/diag/rdbms/orclbi/orclbi/hm
1 Default Trace File                        /u01/oracle/diag/rdbms
                                           /orclbi/orclbi/trace/orcl_ora_22769.trc

1 Active Problem Count 8
1 Active Incident Count 20
```


Файлы трассировки

- Каждый процесс в составе экземпляра Oracle формирует свой файл трассировки.
- Сведения об ошибке записываются в общий журнал предупреждений БД и в файл трассировки конкретного процесса.
- Имя файла трассировки содержит SID и идентификатор породившего его процесса:
mydb_ora_3532.trc

SID PID
- Найти идентификатор текущего процесса можно, обратившись к DPV V\$PROCESS и V\$SESSION:
SQL> SELECT p.spid FROM V\$PROCESS p, V\$SESSION s
WHERE p.addr = s.paddr AND s.audsid =
userenv('SESSIONID');

Automatic Diagnostic Repository

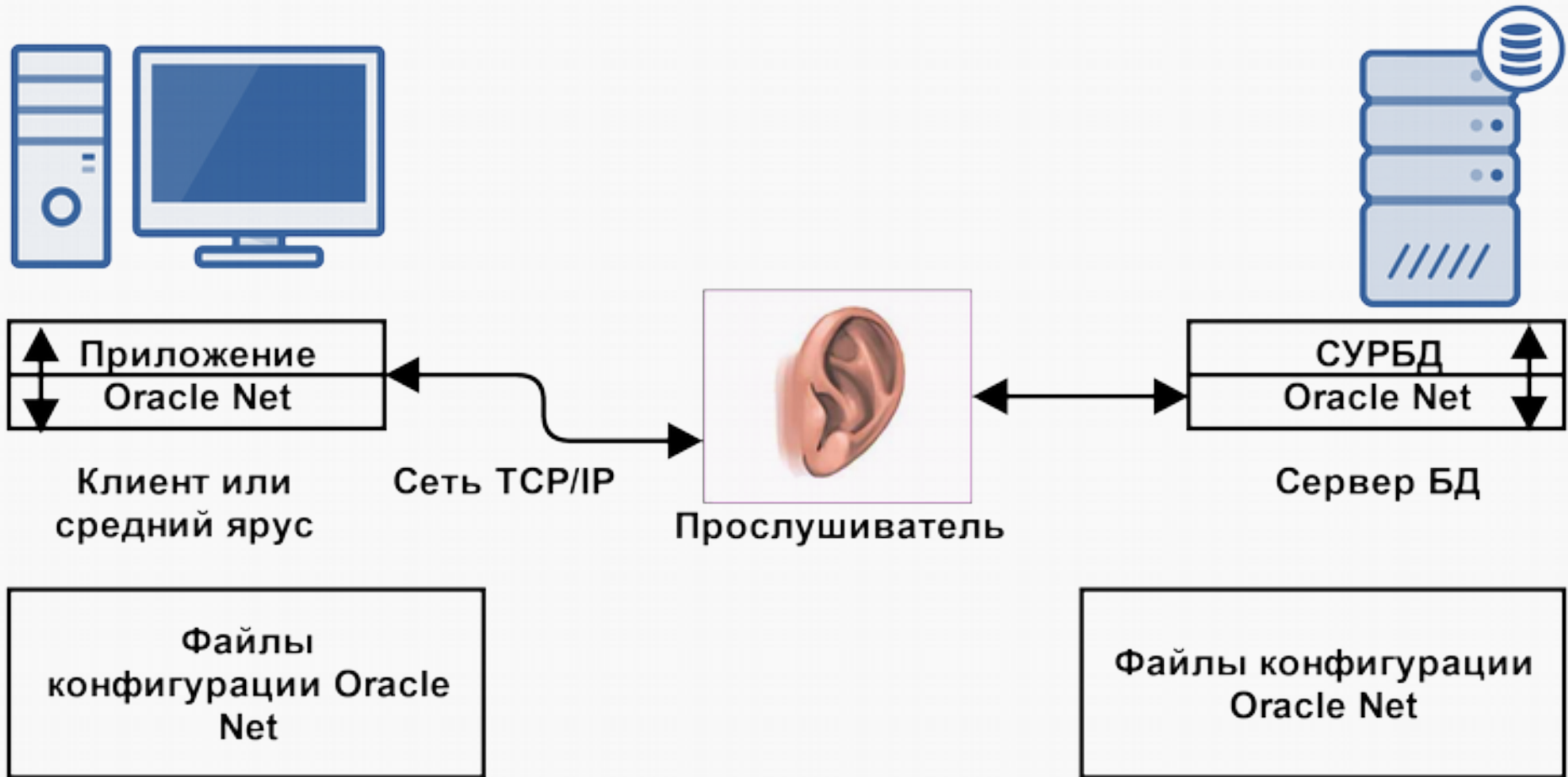
- Все файлы трассировки содержатся в каталоге ADR — Automatic Diagnostic Repository.
- Путь к каталогу ADR задаётся параметром инициализации ADR_BASE и ADR_HOME.
- Текущий путь к каталогу ADR можно посмотреть в DPV V\$DIAG_INFO:

```
SQL> SELECT * FROM V$DIAG_INFO;
```

INST_ID	NAME	VALUE
1	Diag Enabled	TRUE
1	ADR Base	/u01/oracle
1	ADR Home	/u01/oracle/diag/rdbms/orclbi/orclbi
1	Diag Trace	/u01/oracle/diag/rdbms/orclbi/orclbi/trace
1	Diag Alert	/u01/oracle/diag/rdbms/orclbi/orclbi/alert
1	Diag Incident	/u01/oracle/diag/rdbms /orclbi/orclbi/incident
1	Diag Cdump	/u01/oracle/diag/rdbms/orclbi/orclbi/cdump
1	Health Monitor	/u01/oracle/diag/rdbms/orclbi/orclbi/hm
1	Default Trace File	/u01/oracle/diag/rdbms /orclbi/orclbi/trace/orcl_ora_22769.trc
1	Active Problem Count	8
1	Active Incident Count	20

6. Сетевая среда Oracle

Общая схема Oracle Net



Oracle Net Listener

- *Слушатель* (или *прослушиватель*, Oracle Net Listener) – это шлюз экземпляра Oracle, который обрабатывает все внешние соединения пользователей.
- Один слушатель может одновременно обслуживать несколько экземпляров базы данных и тысячи клиентских соединений.
- Слушатели можно конфигурировать с помощью Enterprise Manager или «вручную» — путём правки конфигурационных файлов.

Конфигурация Oracle Net Listener

- Хранится в файле `$ORACLE_HOME/network/admin/listener.ora`.
- Конфигурация всех слушателей хранится в одном и том же файле.
- Пример конфигурации:

```
LISTENER=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=sale-server)(PORT=1521))
      (ADDRESS=(PROTOCOL=ipc)(KEY=extproc))))
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (GLOBAL_DBNAME=sales.us.example.com)
      (ORACLE_HOME=/oracle11g)
      (SID_NAME=sales))
    (SID_DESC=
      (SID_NAME=plsextproc)
      (ORACLE_HOME=/oracle11g)
      (PROGRAM=extproc)))
```

Чтобы установить соединение, клиенту должны быть известны:

- Имя хоста, на котором работает слушатель (например, `helios.cs.ifmo.ru`).
- Имя протокола, за которым следит слушатель (например, TCP).
- Протокол, по которому осуществляется обмен сообщениями со слушателем.
- Имя службы, подключение к которой осуществляет слушатель.

Алгоритм сетевого взаимодействия



Алгоритм сетевого взаимодействия (продолжение)

- Слушатель принимает пакет CONNECT и проверяет запрашиваемое имя службы Oracle Net.
- Если имя службы не запрашивается (например, в случае запроса tnsping), слушатель подтверждает запрос соединения и больше операций не выполняет.
- При запросе неверного имени службы слушатель передает пользовательскому процессу код ошибки.
- Если в пакете CONNECT запрашивается корректное имя службы, слушатель связывает пользовательский процесс с серверным процессом.
- Слушатель может создавать новый серверный процесс или использовать существующий — это зависит от режима работы сервера.
- После этого слушатель уже не обрабатывает соединение, этим занимается серверный процесс.

Управление слушателями

Для управления слушателями используется утилита lsnrctl:

```
[oracle@edrsr17p1 ~]$ lsnrctl
LSNRCTL for Linux: Version 11.1.0.3.0 - Beta on 30-MAY-
2007 22:38:19
Copyright (c) 1991, 2006, Oracle. All rights reserved.
Welcome to LSNRCTL, type "help" for information.
LSNRCTL> help
The following operations are available
An asterisk (*) denotes a modifier or extended command:
start                stop                 status
services            version              reload
save_config         trace                spawn
change_password    quit                 exit
set*                 show*
```

Разрешение имён

Oracle Net поддерживает несколько методов разрешения информации о соединении:

- Разрешение имен Easy Connect: используется строка соединения TSP/IP.
- Локальное разрешение имен: используется локальный файл конфигурации.
- Разрешение имен на сервере каталогов: используется центральный сервер каталогов с поддержкой LDAP.
- Внешнее разрешение имен: используется внешняя служба разрешения имен.

Подробнее — см. главу 5 учебника.

Проверка связности сети

`tnsping` — утилита, предназначенная для проверки псевдонимов служб Oracle Net (аналог `ping`):

- Проверяет соединение между клиентом и Oracle Net Listener.
- Не проверяет доступность самой запрошенной службы — только слушателя.
- Поддерживает разрешение имен Easy Connect:
`tnsping db.us.oracle.com:1521/dba11g`
- Поддерживает локальное разрешение имен и разрешение имен на сервере каталогов:
`tnsping orcl`

Выделенный сервер (Dedicated Server)

Главный недостаток — плохая масштабируемость.



Разделяемые серверы (Shared Servers)

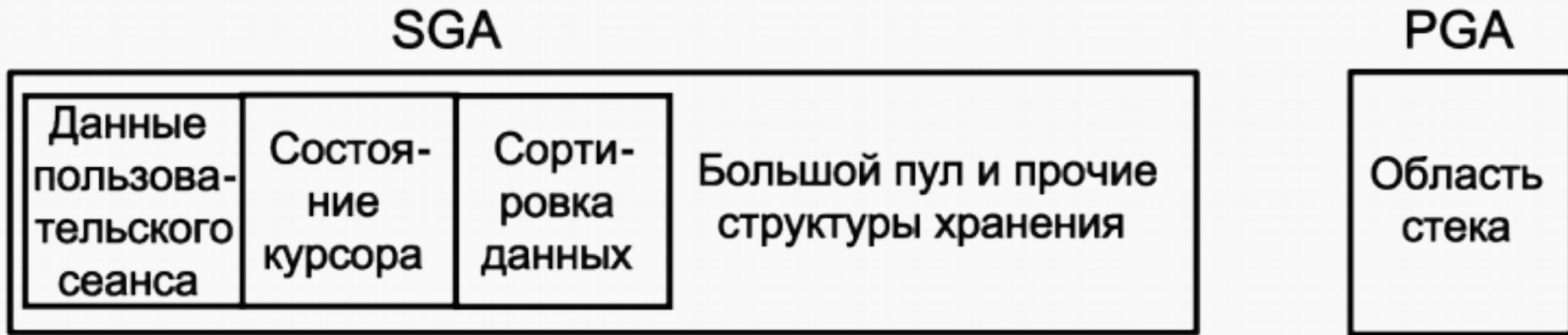


Разделяемые серверы (Shared Servers, продолжение)

- Для каждой службы в архитектуре разделяемого сервера используется как минимум один процесс *диспетчера* (обычно больше).
- Слушатель хранит список доступных диспетчеров для каждого имени службы, а также информацию о загрузке соединения (количество одновременных соединений) для каждого диспетчера.
- Запрос перенаправляется к наименее загруженному диспетчеру, обслуживающему службу с запрошенным именем.
- Во время сеанса пользователь поддерживает соединение с одним и тем же диспетчером (но запросы могут обрабатывать разные серверные процессы).
- Один диспетчер может обслуживать сотни сеансов пользователей.
- Диспетчеры направляют запросы пользователей в общую очередь, которая размещена в области SGA, выделенной для разделяемого пула.

Разделяемые серверы (Shared Servers, продолжение)

- Т.к. запросы одного пользовательского процесса могут обрабатывать *разные* серверные, большая часть данных из PGA переносится в SGA:



- Это нужно учитывать при конфигурации размера SGA.

Переключение между режимами сервера

- Текущее состояние можно проверить в DPV v\$SESSION:
SQL> SELECT server FROM v\$session;
- Переключение из dedicated в shared:
SQL> ALTER SYSTEM SET SHARED_SERVERS = 2;
- Установка количества диспетчеров:
SQL> ALTER SYSTEM
SET DISPATCHERS =
'(PROTOCOL=TCP)(DISPATCHERS=5) (INDEX=0)',
'(PROTOCOL=TCPS)(DISPATCHERS=2) (INDEX=1)';
- Переключение из shared в dedicated:
ALTER SYSTEM SET SHARED_SERVERS = 0 scope = both;
или
ALTER SYSTEM SET MAX_SHARED_SERVERS = 0 scope = both;

Разделяемые серверы — пул соединений



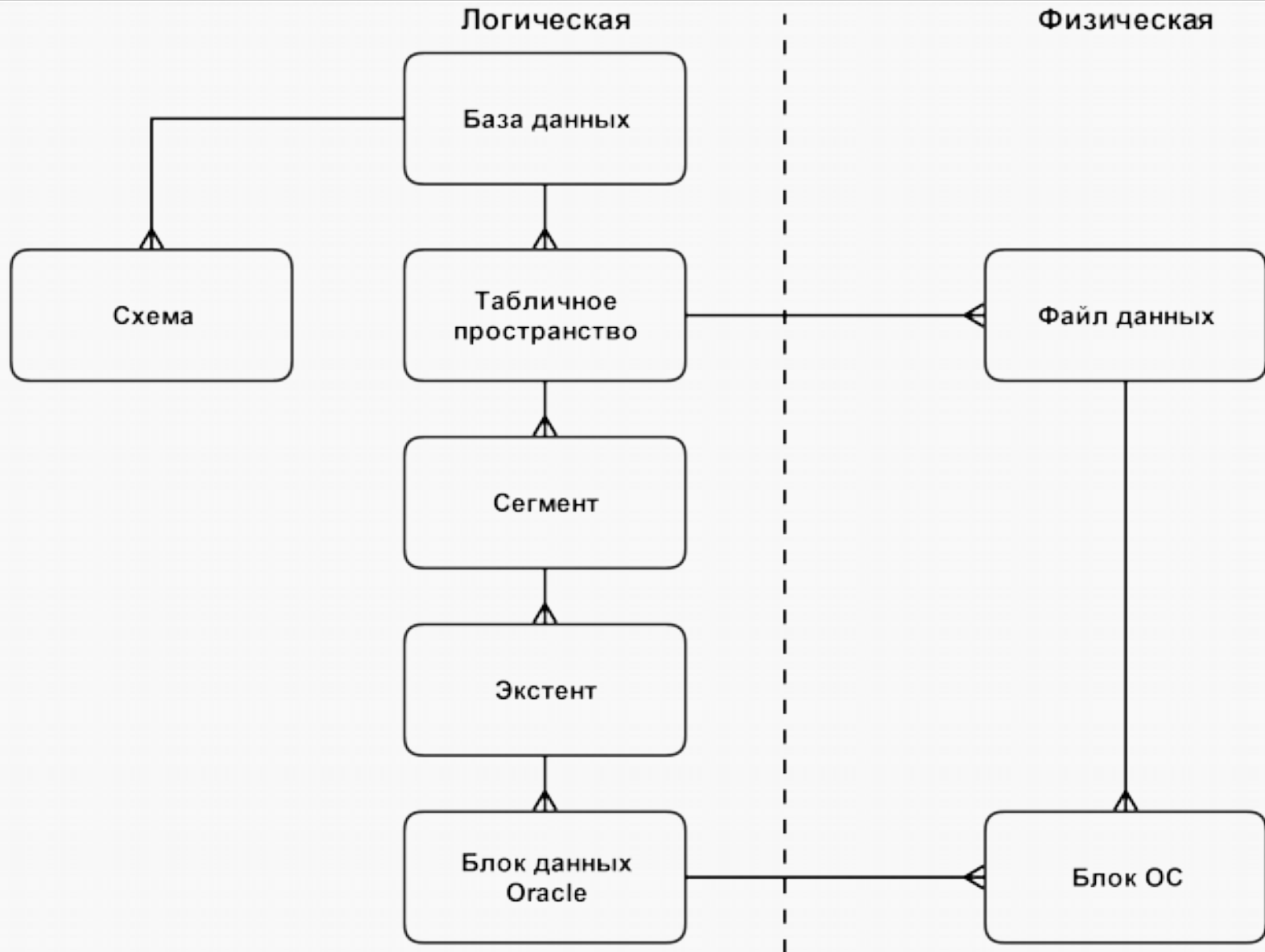
Когда не стоит использовать разделяемые серверы

Определённые операции с БД не стоит выполнять с помощью разделяемых серверов:

- Администрирование БД.
- Операции резервного копирования и восстановления.
- Пакетную обработку и операции с массовой загрузкой.
- Операции с хранилищами данных.

7. Структуры хранения БД

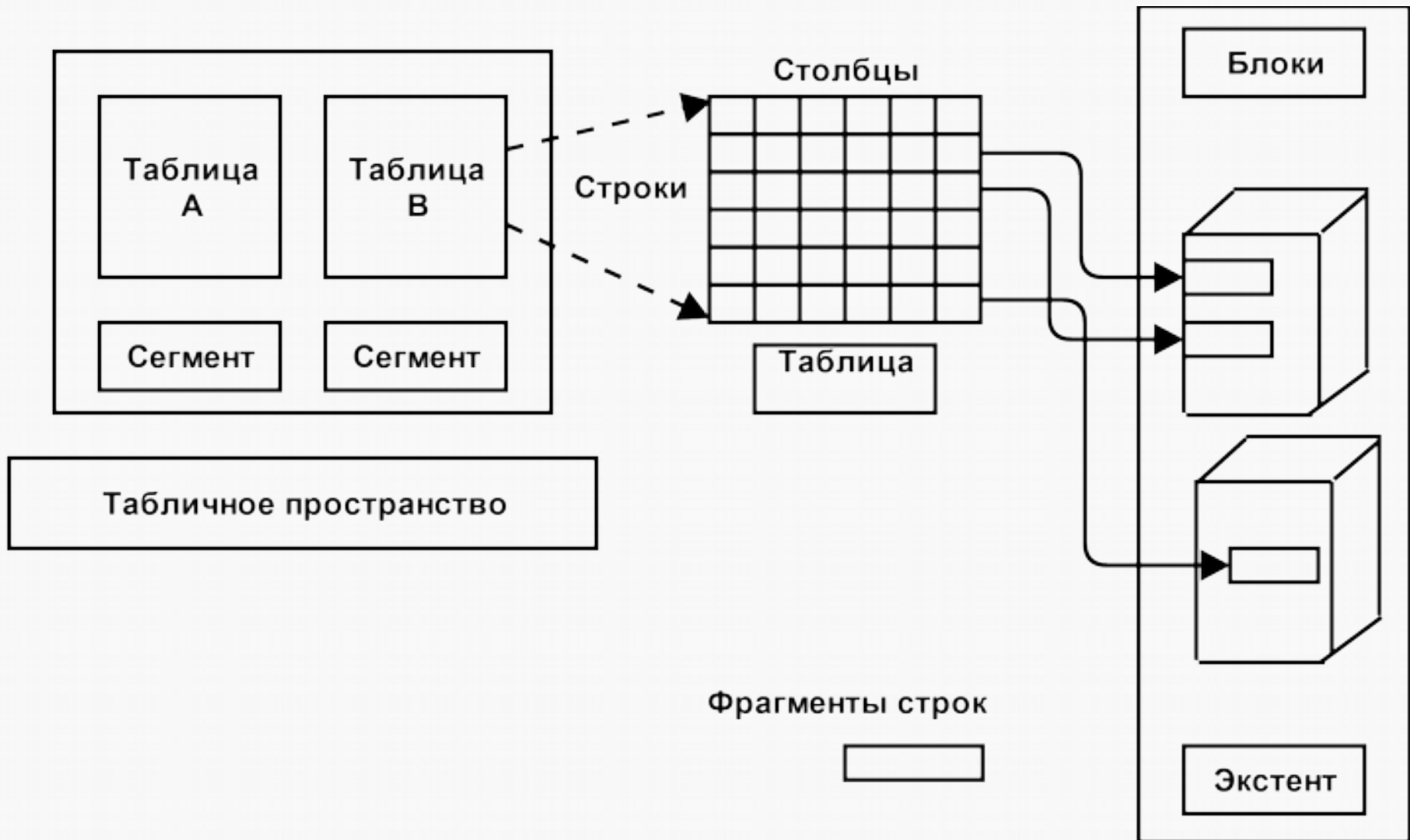
Логическая и физическая структуры хранения БД



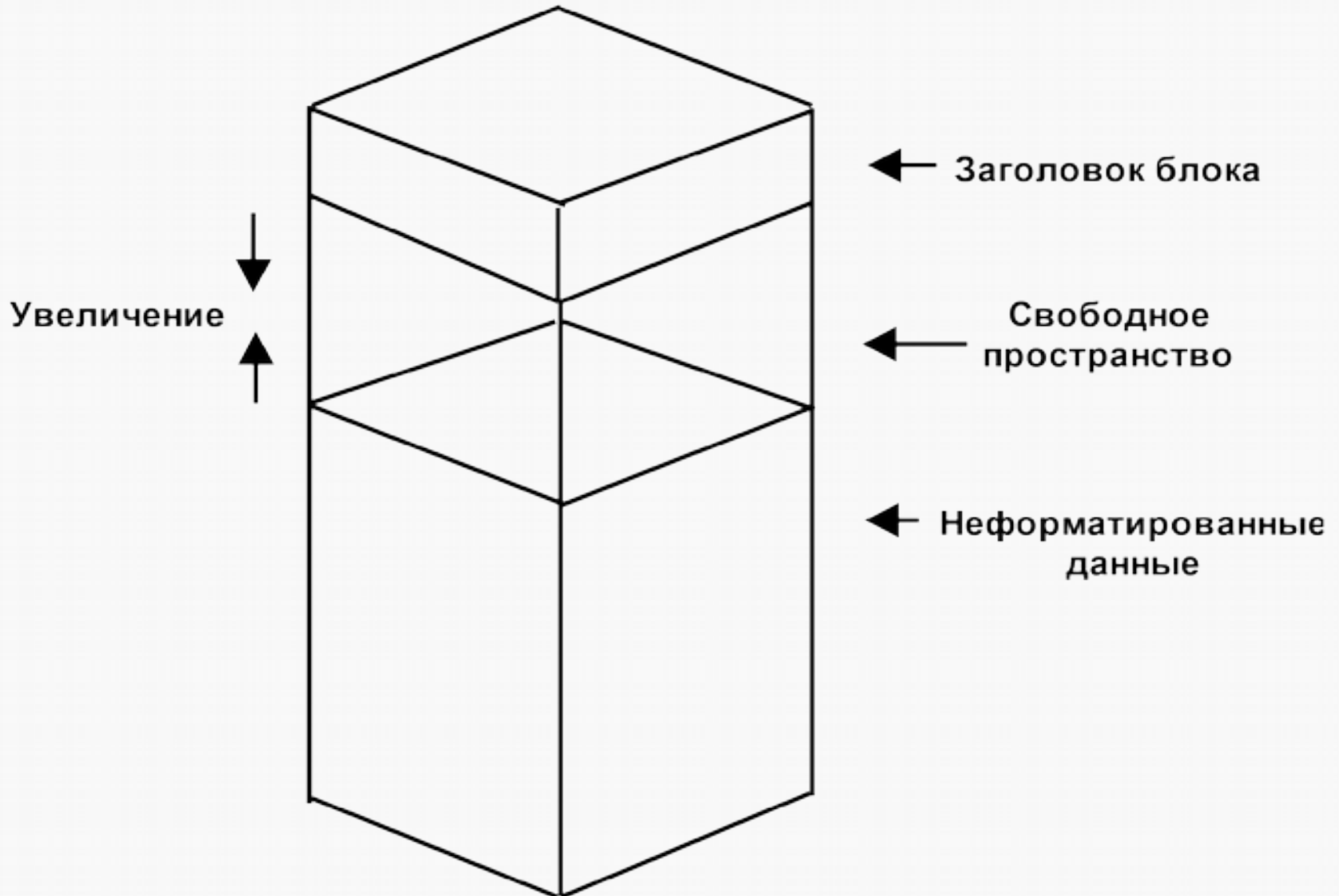
Размер блока данных

- Определяется параметром инициализации `DB_BLOCK_SIZE`.
- Размер — от 2 до 32 КБ.
- По умолчанию — 8 КБ.
- Изменить размер блока данных можно только путём повторного создания БД.

Как хранятся табличные данные




Содержимое блока данных



Содержимое блока данных (продолжение)

- Заголовок блока — содержит сведения о типе сегмента (табличный или индексный), адресе блока данных, каталоге таблицы, каталоге строки и транзакционных слотах размером по 24 байта каждый, которые используются при внесении изменений в строки блока. Заполняется сверху вниз.
- Неформатированные данные — данные для строк блока. Заполняются снизу вверх.
- Свободное пространство внутри блока, за счет которого могут увеличиваться заголовок и пространство неформатированных данных.
- События, которые могут увеличить размер заголовка:
 - Требуется больше неформатированных записей.
 - Требуется большее количество транзакционных слотов.
- Изначально свободное пространство является непрерывным, однако в ходе выполнения операций удаления и обновления может стать фрагментированным.
- Свободное пространство может быть дефрагментировано сервером Oracle.



Табличные пространства и файлы данных

- БД состоит из одного или нескольких логических табличных пространств.
- Каждое табличное пространство базы данных Oracle состоит из одного или нескольких файлов данных.
- БД должна содержать минимум два табличных пространства — SYSTEM и SYSAUX, каждое из которых представлено минимум одним файлом данных.
- Одна БД может иметь до 65534 файлов данных.
- Если табличное пространство в течение всего жизненного цикла представлено одним (и только одним) файлом данных, оно называется табличным пространством типа BIGFILE.
- Временный файл – это файл, который принадлежит временному табличному пространству:
 - Создается с параметром TEMPFILE:

```
alter tablespace temp
add tempfile 'c:\oracle\oradata\temp3\temp02.dbf'
size 50m
reuse
autoextend on
next 1m
maxsize 500m;
```
 - Временные табличные пространства используются для операций сортировки и не могут содержать постоянных объектов базы данных, таких как таблицы.

Управление табличными пространствами

- Управляемое локально:
 - управление свободными экстентами осуществляется в табличном пространстве;
 - для записи свободных экстентов используется битовый образ;
 - каждый бит соответствует блоку или группе блоков;
 - значение бита описывает экстент: свободен или занят.
- Управляемое словарём:
 - свободными экстентами управляет Oracle через словарь данных;
 - при выделении и освобождении экстентов обновляются соответствующие представления

Табличные пространства, управляемые локально

- Экстенты могут выделяться одним из двух способов:
 - Автоматически — размерами экстентов управляет система (он всегда кратен 64 КБ). Способ неприменим ко временным табличным пространствам.
 - Унифицированно — табличные пространства используют унифицированный размер экстентов указываемый администратором (по умолчанию — 1 МБ). Режим нельзя использовать для табличных пространств отмены операций.
- Управление пространством сегментов можно осуществлять:
 - Автоматически — используются битовые образы. Битовый образ описывает состояние каждого блока данных в сегменте в соответствии с объемом пространства блока, которое доступно для вставки строк.
 - Вручную — используются списки свободных сегментов (списки блоков данных, в которых имеется свободное пространство). Требуется вручную задавать и настраивать значения параметров хранения PCTUSED, FREELISTS и FREELIST GROUPS для объектов схемы.

Состояния табличных пространств

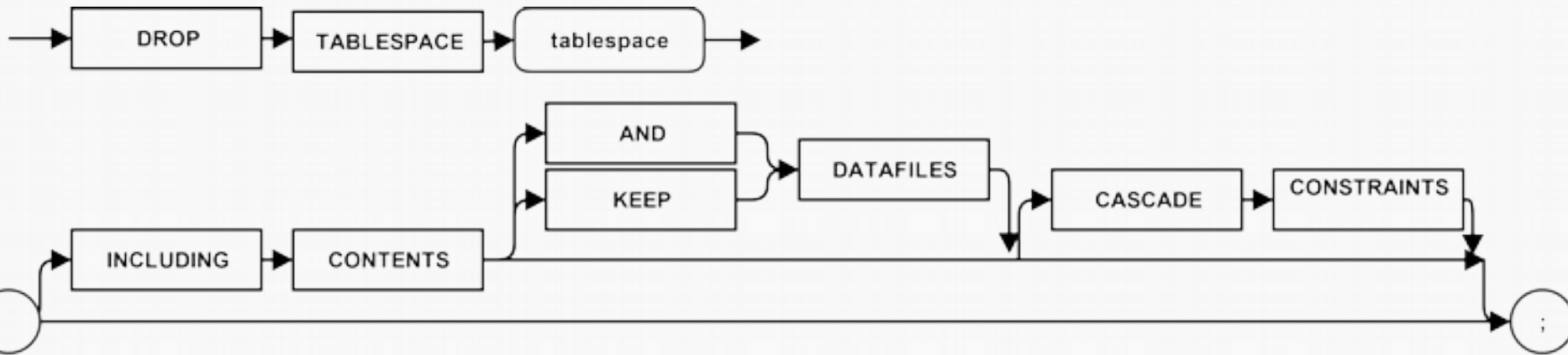
- **(Оперативное) Чтение-запись:** доступно как для чтения, так и для записи.
- **(Оперативное) Только чтение:** доступно только для чтения. После перевода в это состояние текущие транзакции будут завершены (с фиксацией или откатом), новые операции DML будут запрещены. Табличные пространства SYSTEM и SYSAUX нельзя перевести в этот режим.
- **Автономное:** эта часть БД становится временно недоступной для использования (остальная БД будет оставаться открытой и доступной). При переводе в автономный режим можно использовать следующие параметры:
 - Normal: режим можно использовать, если во всех файлах данных табличного пространства отсутствуют ошибки. СУБД установит контрольную точку для всех файлов данных перед переводом их в автономный режим.
 - Temporgu: табличное пространство временно переводится в автономный режим даже при наличии ошибок в файлах данных. При «обратном» переводе в оперативный режим *может* потребоваться восстановление данных из журнала повторов.
 - Immediate: табличное пространство можно перевести в автономный режим немедленно, без установки контрольной точки для файлов данных БД Oracle. При «обратном» переводе в оперативный режим *обязательно* потребуются восстановление данных из журнала повторов.

Просмотр сведений о табличных пространствах

- С помощью GUI Enterprise Manager.
- С помощью DPV:
 - Сведения о табличных пространствах:
 - DBA_TABLESPACES
 - V\$TABLESPACE
 - Сведения о файлах данных:
 - DBA_DATA_FILES
 - V\$DATAFILE
 - Сведения о временных файлах:
 - DBA_TEMP_FILES
 - V\$TEMPFILE

Удаление табличных пространств

- Используется команда DROP TABLESPACE:



- Нельзя удалить табличное пространство SYSTEM.
- Табличное пространство SYSAUX может удалить только DBA и только если БД запущена в режиме MIGRATE.
- Нельзя удалить табличное пространство отката если оно используется и в нём есть незавершённые транзакции.

Oracle Managed Files (OMF)

- Файлы, которыми управляет экземпляр Oracle.
- Операции над ними указываются в терминах объектов базы данных, а не имён файлов.
- Oracle умеет управлять файлами следующих структур БД:
 - табличные пространства;
 - файлы журнала повторов;
 - управляющие файлы;
 - архивные журналы;
 - файлы отслеживания изменений в блоках;
 - журналы моментального отката;
 - резервные копии RMAN.
- БД может содержать как управляемые Oracle файлы, так и неуправляемые.
- Каталог файловой системы, в котором хранятся OMF, должен быть создан заранее — БД автоматически его не создаст. Также каталог должен иметь все необходимые разрешения, чтобы БД могла создавать в нем файлы.

Способы увеличения размера БД:

- создать новое табличное пространство;
- добавить файл данных в существующее табличное пространство;
- увеличить размер файла данных;
- разрешить динамический рост файла данных.

8. Резервное копирование и восстановление

Задачи администратора

- Защита базы данных от сбоев, когда это возможно.
- Увеличение среднего промежутка времени между сбоями (MTBF).
- Повышение надёжности БД путём избыточности.
- Снижение среднего времени восстановления (MTTR).
- Минимизация потери данных.

Категории сбоев

- **Сбой оператора:** ошибка отдельной операции базы данных (выборка, вставка, обновление или удаление).
- **Сбой пользовательского процесса:** ошибка отдельного сеанса базы данных.
- **Сбой сети:** потеря соединения с базой данных.
- **Ошибка пользователя:** операция выполняется успешно, но сама операция неверна (удаление таблицы или ввод неверных данных).
- **Сбой экземпляра:** непредвиденное завершение работы экземпляра базы данных.
- **Сбой носителя:** потеря одного или нескольких файлов базы данных (то есть, удаление файлов или сбой в работе диска).

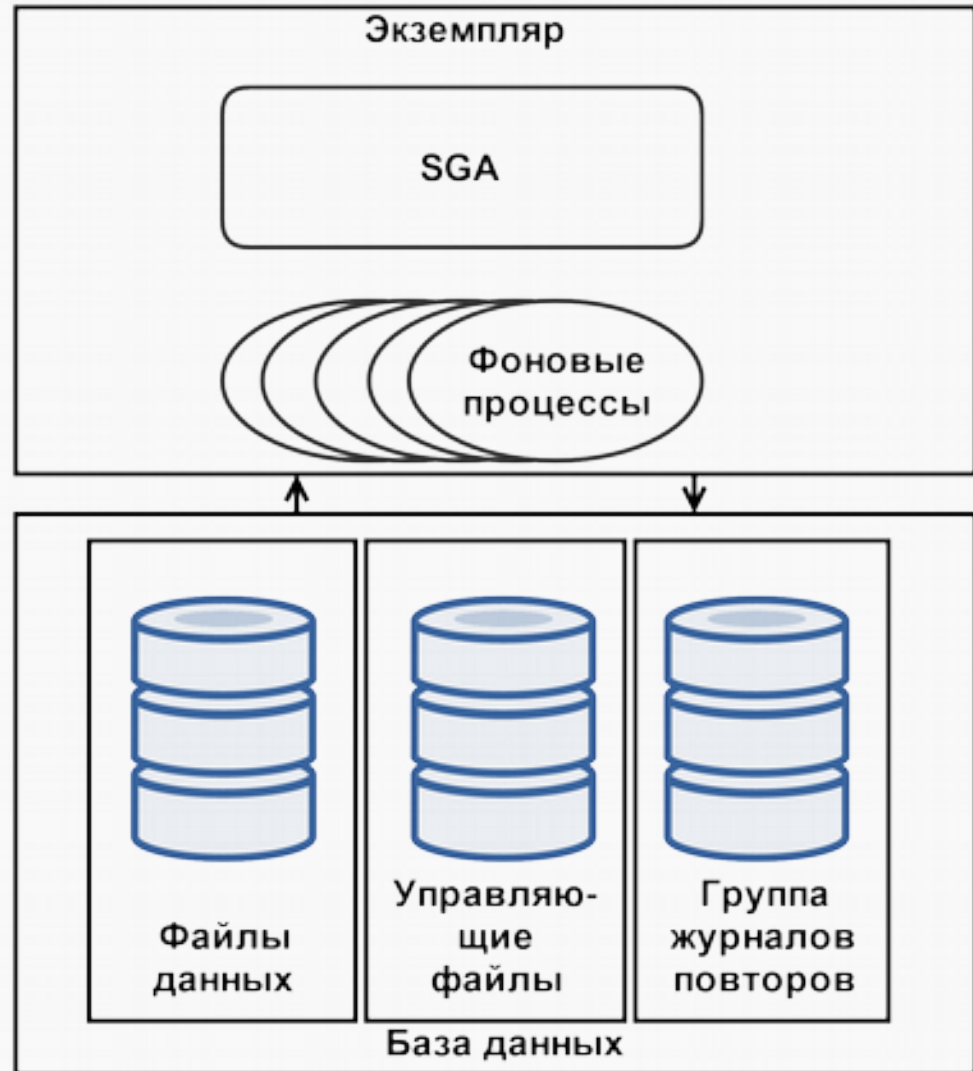
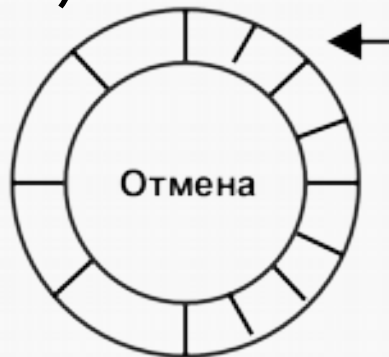
Автоматическое восстановление экземпляра

Автоматическое восстановление *после сбоя экземпляра*:

- Производится вследствие попыток открыть базу данных, файлы которой не были синхронизированы при завершении работы.
- Не требует от администратора никаких действий, за исключением запуска экземпляра БД.
- Использует информацию, хранящуюся в группах журналов повторов, для синхронизации файлов.
- Состоит из двух отдельных операций:
 - накат: восстанавливается состояние файлов данных до момента сбоя экземпляра;
 - откат: внесенные, но не зафиксированные изменения возвращаются к исходному состоянию.

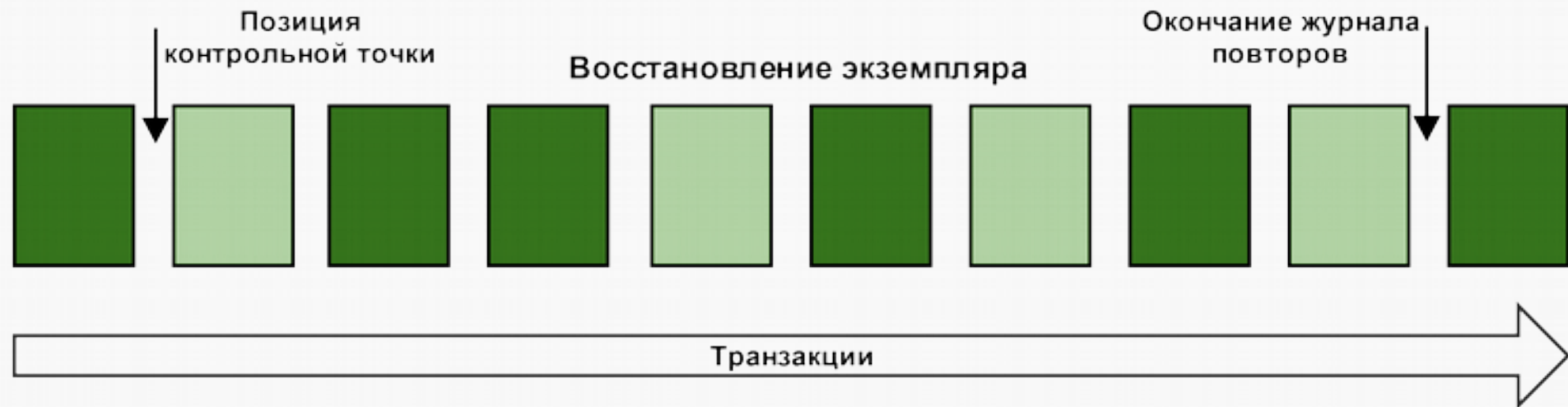
Этапы восстановления экземпляра

- 1) Файлы данных не синхронизированы.
- 2) Накат (повтор).
- 3) Зафиксированные и незафиксированные данные в файлах.
- 4) Открытие БД.
- 5) Откат (отмена).
- 6) Зафиксированные данные в файлах.



Настройка восстановления экземпляра

- Во время восстановления экземпляра к файлам данных необходимо применить все транзакции между позицией контрольной точки и окончанием журнала повторов.
- Настройка восстановления экземпляра производится путем управления интервалом между позицией контрольной точки и окончанием журнала повторов:
`SQL> ALTER SYSTEM SET FAST_START_MTTR_TARGET=30;`



Сбой носителя

- *Сбой носителя* — это любой сбой, который приводит к потере или повреждению одного или нескольких файлов базы данных (файла данных, управляющего файла или файла журнала повторов).
- Восстановление после сбоя носителя требует восстановления отсутствующих файлов.



Настройка возможности восстановления после сбоя носителя

Настройка БД для обеспечения максимальных возможностей восстановления включает в себя:

- Планирование регулярных операций резервного копирования.
- Мультиплексирование управляющих файлов.
- Мультиплексирование групп журналов повторов.
- Хранение архивных копий журналов повторов.



Область мгновенного восстановления (Flash Recovery Area)

Область мгновенного восстановления:

- Упрощает управление хранением резервных копий.
- Использует отдельное пространство на диске (отдельно от рабочих файлов БД); рекомендуется использовать отдельный накопитель.
- Расположение задается параметром `USE_DB_RECOVERY_FILE_DEST`.
- Должна быть достаточного размера.
- Управляется автоматически.

Настройка области мгновенного восстановления предполагает определение ее расположения, размера и методики сохранения.



Мультиплексирование управляющих файлов

Общие рекомендации:

- Нужно создавать как минимум две копии каждого управляющего файла (рекомендуется три копии).
- Каждая копия должна размещаться на отдельном диске.
- Как минимум одна копия должна размещаться на отдельном контроллере диска.

Чтобы добавить управляющий файл вручную:

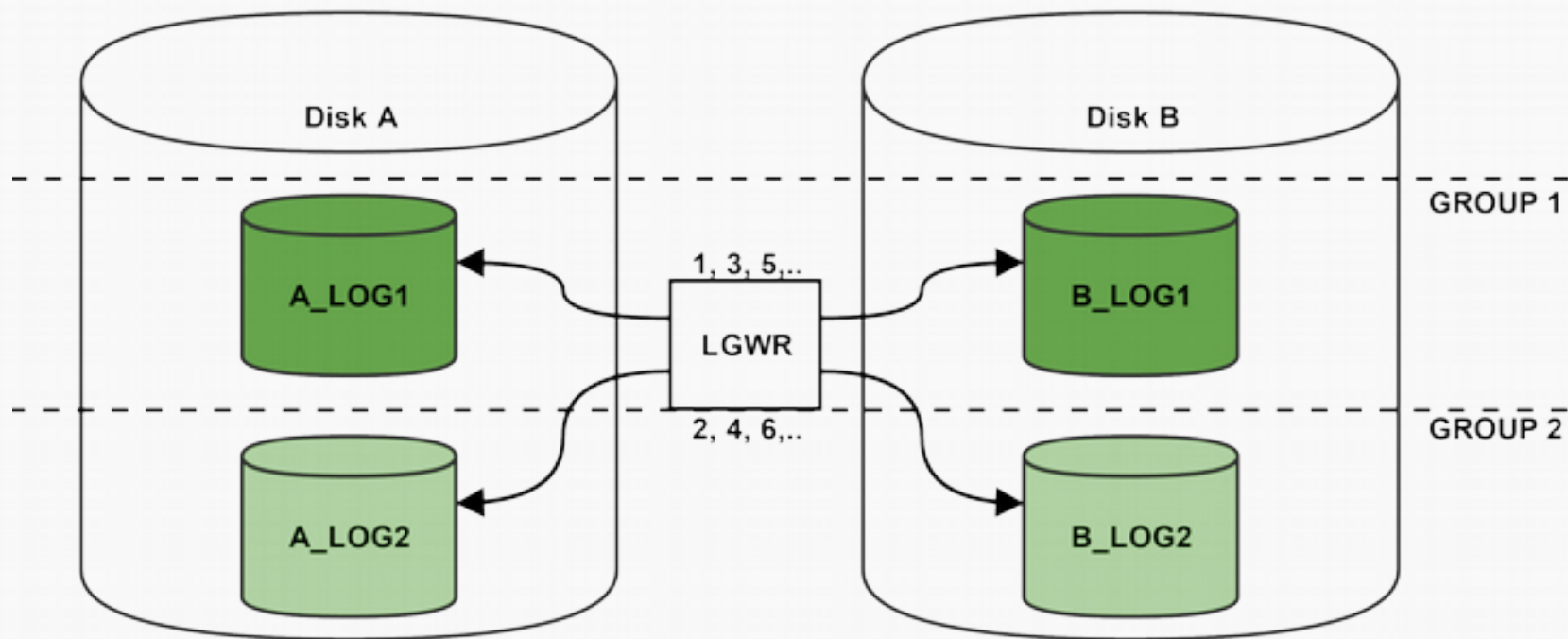
- 1) Измените файл SPFILE с помощью команды `ALTER SYSTEM SET control_files`.
- 2) Завершите работу базы данных.
- 3) Перенесите в новое расположение копию файла, созданную средствами ОС.
- 4) Откройте базу данных.

Мультиплексирование файлов журнала повторов

Проще всего настраивается в Enterprise Manager.

Рекомендуется:

- Наличие не менее двух файлов на каждую группу журналов повторов.
- Размещение каждого файла на отдельном диске.
- Размещение каждого файла на отдельном контроллере диска.



Архивные файлы журнала повторов

Чтобы сохранить информацию о повторах, необходимо «заставить» Oracle создавать архивные копии файлов журналов повторов. Для этого нужно:

1) Указать правило именования архивных файлов журналов:

- %s: включает в имя файла порядковый номер журнала;
- %t: включает в имя файла номер потока;
- %г: включает идентификатор журнала сбросов, чтобы гарантировать уникальность имени архивного файла журнала;
- %d: включает в имя файла идентификатор базы данных.

Формат *обязательно* должен включать символы %s, %t и %г.

2) Указать один или несколько каталогов для хранения архивных файлов журналов (параметр инициализации DB_RECOVERY_FILE_DEST).

3) Переключить базу данных в режим ARCHIVELOG.

БД в режиме ARCHIVELOG

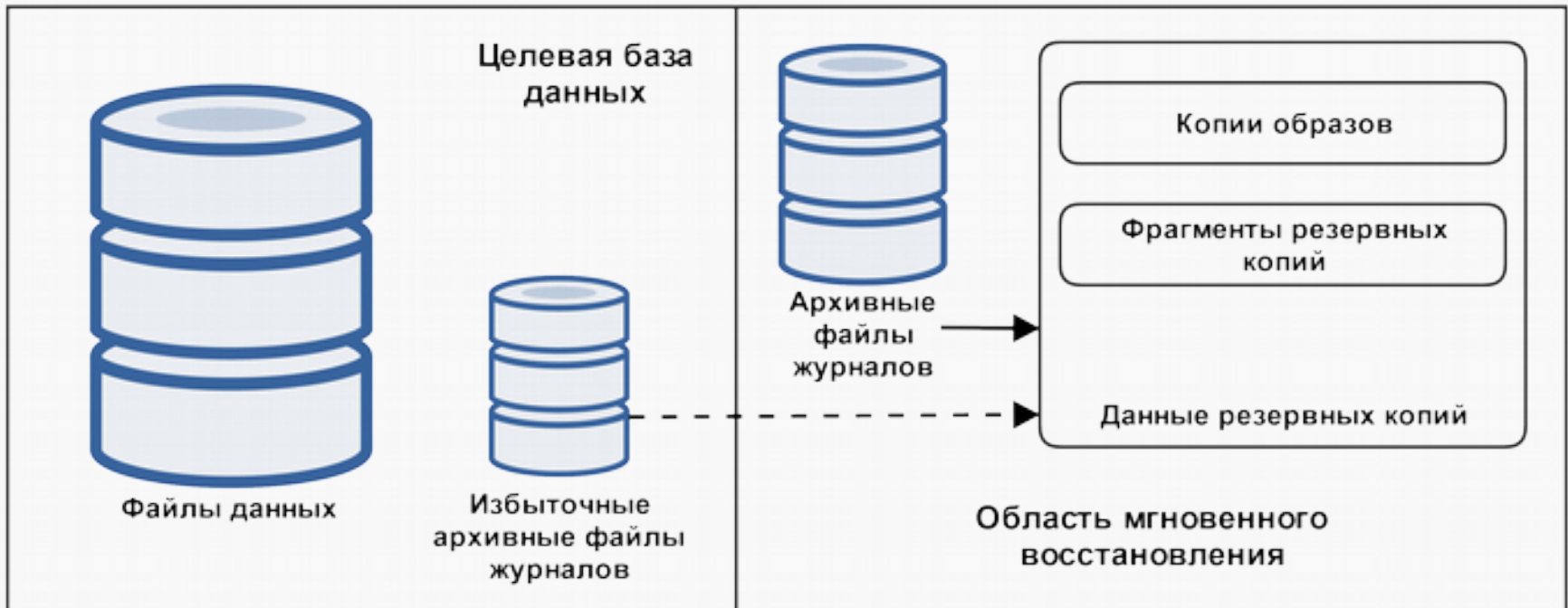
- Если БД находится в режиме NOARCHIVELOG (по умолчанию), восстановление возможно только на момент создания последней резервной копии.
- В режиме ARCHIVELOG состояние базы данных можно восстановить до момента последней фиксации.
- Переключение в режим ARCHIVELOG:

```
sqlplus / as sysdba  
shutdown immediate  
startup mount  
alter database archivelog;  
alter database open;  
archive log list
```

Решения для резервного копирования

Можно использовать:

- Утилиты `exp/imp` (устарели) для схем пользователей;
- `Data pump` — архитектура резервного копирования, включающая новые утилиты `expdp/impdp`
- Диспетчер восстановления (`RMAN` — `Recovery MANager`);
- Утилиту `Oracle Secure Backup`;



- Стратегия резервного копирования может охватывать:
 - всю базу данных (полная);
 - часть базы данных (частичная).
- В зависимости от типа резервная копия может содержать:
 - все блоки данных в пределах выбранных файлов (полная);
 - только ту информацию, которая изменилась с момента последнего резервного копирования (инкрементная):
 - кумулятивная (изменения вплоть до последнего уровня 0);
 - дифференциальная (изменения вплоть до последнего инкрементного резервного копирования).
- Существует два режима резервного копирования:
 - автономное (согласованное, «холодное»);
 - оперативное (несогласованное, «горячее»).

Терминология (продолжение)



Копии образов
(повторяющиеся данные и
файлы журналов в
формате ОС).

Файл данных 1	Файл данных 2
Файл данных 3	Файл данных 4
Файл данных 5	Файл данных 6

Набор резервирования
(двоичные сжатые файлы
в собственном формате
Oracle).

- Резервные копии можно хранить как:
 - копии образов;
 - наборы резервирования.

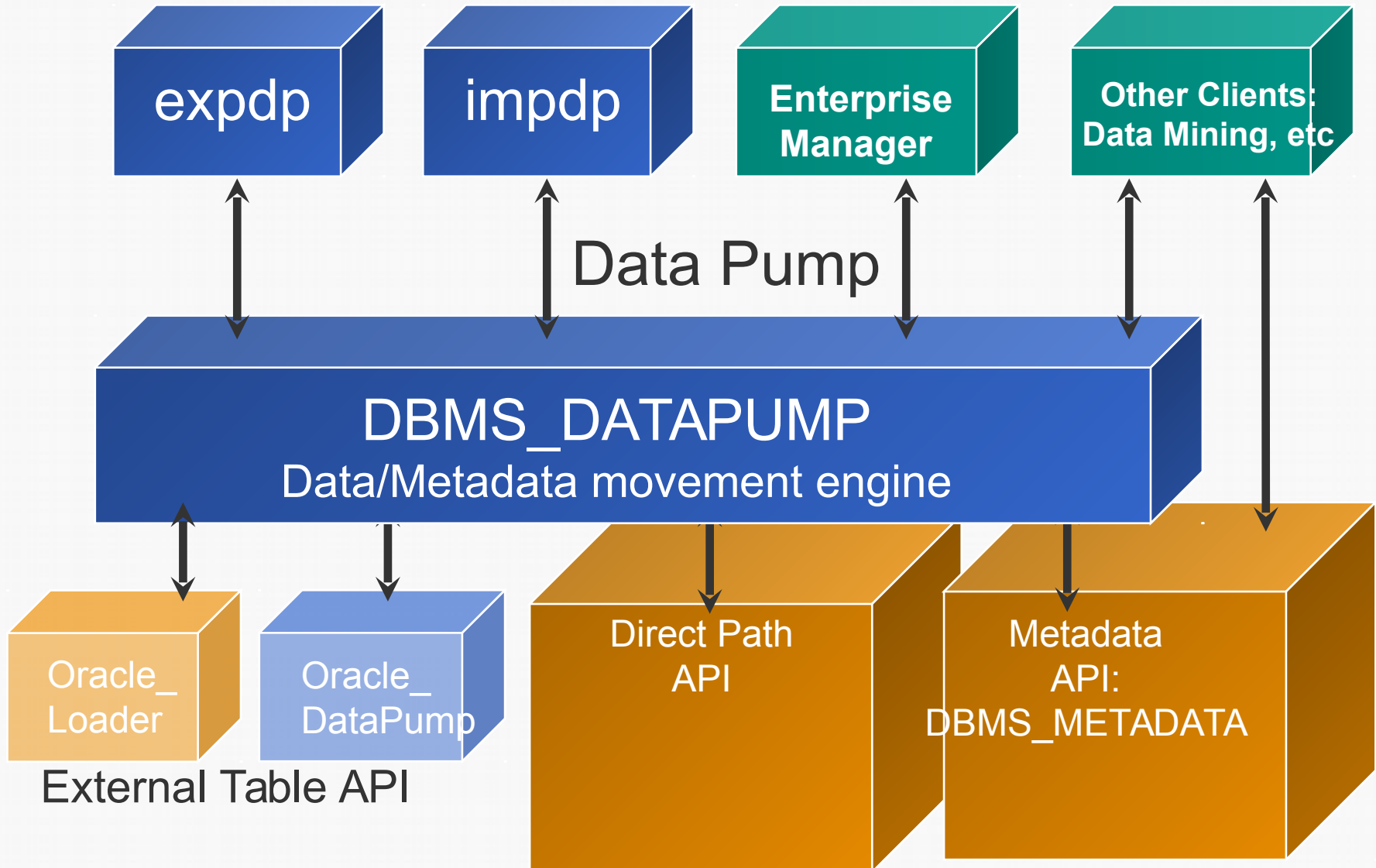
- Включены в состав СУБД с ранних версий
- Копируют выбранную схему БД в командном режиме на стороне клиента
- Можно копировать отдельно схему и данные
- Загрузка данных в один поток, следовательно невысокая производительность
- Основная проблема — слишком долгое восстановление больших БД.

```
$ EXP SCOTT/TIGER GRANTS=Y TABLES=(EMP,DEPT,MGR)
```

Data Pump - «насос данных»

- Высокопроизводительная архитектура на стороне сервера БД
- Загрузка и выгрузка данных и метаданных
- Реализована в пакете DBMS_DATAPUMP.
- Данные передаются в формате потока. Метаданные в виде XML
- Обновленные клиенты expdp and impdp. Синтаксис **похож** на exp/imp

Data Pump — архитектура





Data Pump — производительность!

- Автоматический двухуровневый параллелизм
 - Простое задание параллелизма:
`parallel=<number of active threads>`
 - Динамическое добавление и удаление заданий в Enterprise Edition
 - Параллельное построение индексов
 - Одновременная выгрузка данных и метаданных
- Скорость выгрузки в один поток: 1.5-2X exp
- Скорость загрузки в один поток: 15X-40X imp
- Скорость вместе с построением индексов : 4-10X imp

Data Pump — Checkpoint / Restart

- Прогресс задания учитывается в “Мастер-таблице”.
- Возможность приостановить и возобновить задание.
- Прерванные из-за ошибки задания можно запустить заново.
- DVA может подключиться к любому процессу резервного копирования или восстановления.
- Командный режим `expdp/impdp` по Ctrl-C (следующий слайд).

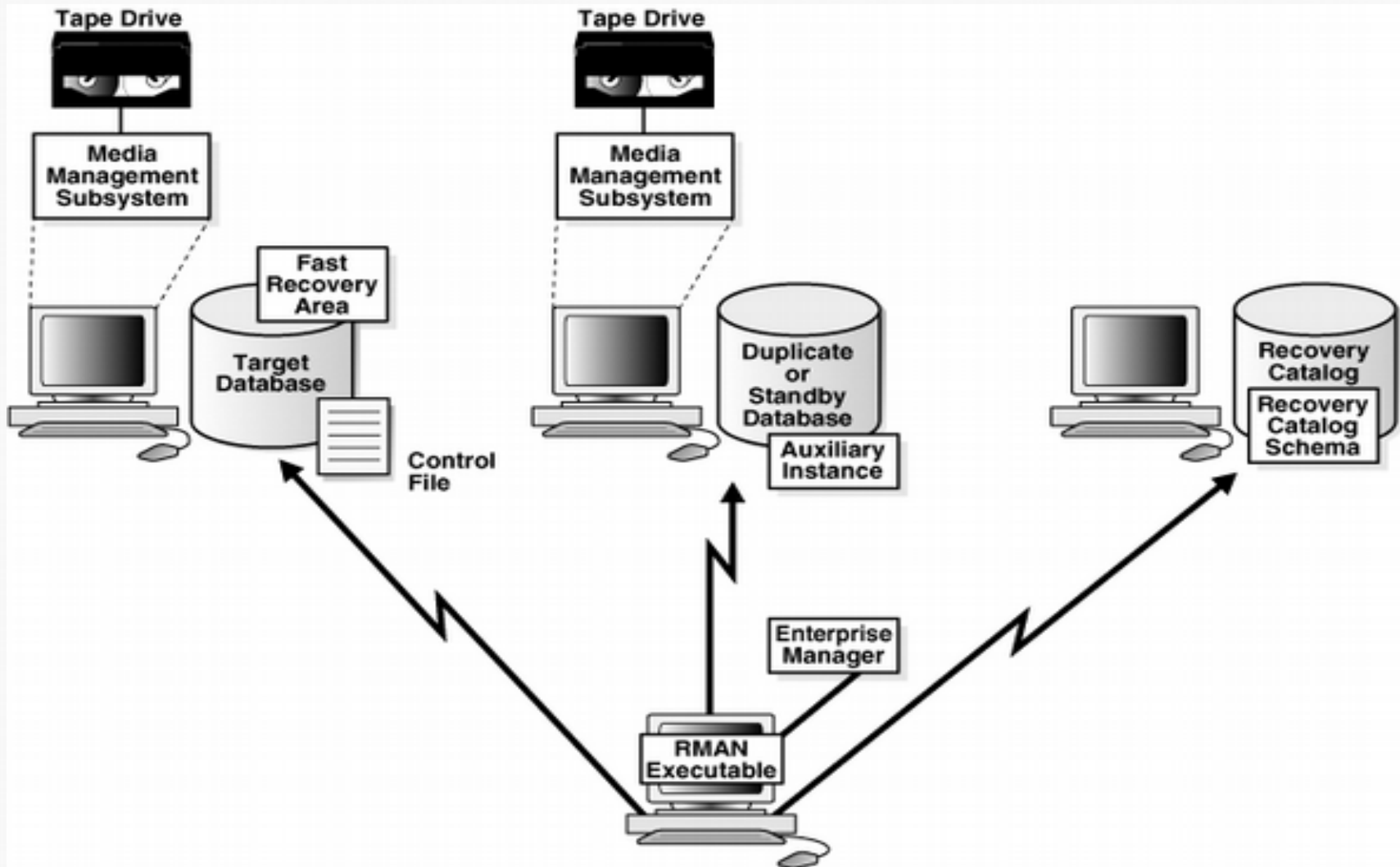
Data Pump — командный режим

- `ADD_FILE`: Добавление файлов к заданию.
- `PARALLEL`: добавление и удаление обработчиков (workers).
- `STATUS`: получение статуса обработчиков.
- `STOP_JOB{=IMMEDIATE}`: Остановка задания с возможностью возобновления работы.
- `START_JOB`: Рестарт задания.
- `KILL_JOB`: Уничтожение задания без возможности возобновления.
- `CONTINUE`: выход из интерактивного режима, продолжение задания.
- `EXIT`: выход из клиента — обработчики продолжают задание.

RMAN — Recovery MANager

- Отдельная утилита для организации сохранения и восстановления БД в автономном или оперативном режимах.
- Использует свой собственный «язык».
- Состоит из следующих компонент:
 - Канал(ы) - серверный процесс, возникающий при установлении связи с устройством ввода/вывода.
 - Целевая БД (ключевой параметр TARGET).
 - Клиент — утилита rman, может выполняться на отличной от БД системе.
 - Media manager — приложение для управления роботами ленточных библиотек.
 - Каталог восстановления — отдельная схема БД для хранения информации о резервных копиях.

RMAN (2)



Вызов RMAN

- Пример вызова утилиты из командной строки и соединения с целевой БД:

```
$ rman
```

```
RMAN> CONNECT TARGET SYS@prod
```

```
target database Password: password
```

```
connected to target database: PROD  
(DBID=39525561)
```

- Пример подключения к целевой БД используя аутентификацию ОС и добавления информации к журналу:

```
$ rman TARGET / LOG /tmp/msglog.log APPEND
```



RMAN — резервное копирование БД

```
$ RMAN NOCATALOG
```

```
RMAN> CONNECT TARGET /
```

```
RMAN> SHUTDOWN IMMEDIATE
```

```
RMAN> STARTUP MOUNT
```

```
RMAN> RUN {
```

```
2> ALLOCATE CHANNEL d1 TYPE DISK;
```

```
3> BACKUP FULL FORMAT
```

```
' /oracle/oradata/teacher/rman-backup\rman_%d_  
%U.bus' DATABASE;
```

```
4> }
```

```
RMAN>
```

- Полное восстановление БД и перевод ее в открытое состояние

```
RMAN> RUN {  
2> ALLOCATE CHANNEL d1 TYPE DISK;  
3> RESTORE DATABASE;  
4> RECOVER DATABASE;  
5> ALTER DATABASE OPEN;  
6> }
```

Oracle Secure Backup

Комплексное решение, функционирующее «поверх» RMAN.

Особенности:

- Вместо NFS используется NDMP (Network Data Management Protocol).
- Управление резервным копированием на основе политик (Policy-based Backup Management).
- Поддержка шифрования резервных копий.

9. Управление доступом пользователей к БД

ОСНОВНЫЕ ПОНЯТИЯ

Формуляр (account) пользователя БД — это способ организации принадлежности и доступа к объектам БД.

Пароль необходим для аутентификации в БД Oracle.

Полномочие (privilege) — это право на выполнение определенного типа SQL-оператора или на доступ к объекту пользователя.

Роль (role) — это именованная группа связанных полномочий, которая предоставляется пользователям или другим ролям.

Профили (profiles) представляют собой именованные наборы ограничений на использование ресурсов БД и экземпляра.

Квота (quota) — это допустимый объем пространства в заданном табличном пространстве.

Формуляр пользователя БД

У каждого пользователя БД есть свой уникальный формуляр БД.

У каждого формуляра пользователя есть:

- Уникальное имя пользователя. Не может превышать 30 байт, не может содержать специальные символы, должно начинаться с буквы.
- Метод аутентификации. По умолчанию — пароль.
- Табличное пространство по умолчанию.
- Временное табличное пространство.
- Профиль пользователя. Набор ресурсов и ограничений с помощью паролей, присвоенных пользователю.
- Состояние формуляра. Пользователям доступны только «открытые» формуляры.



Предопределённые формуляры SYS и SYSTEM

Формуляр SYS:

- получает роль администратора БД;
- обладает всеми полномочиями с параметром ADMIN OPTION;
- необходим для запуска, остановки и выполнения некоторых служебных команд;
- является владельцем словаря данных;
- является владельцем репозитория автоматической рабочей нагрузки (AWR).

Формуляру SYSTEM предоставляется роль администратора БД.

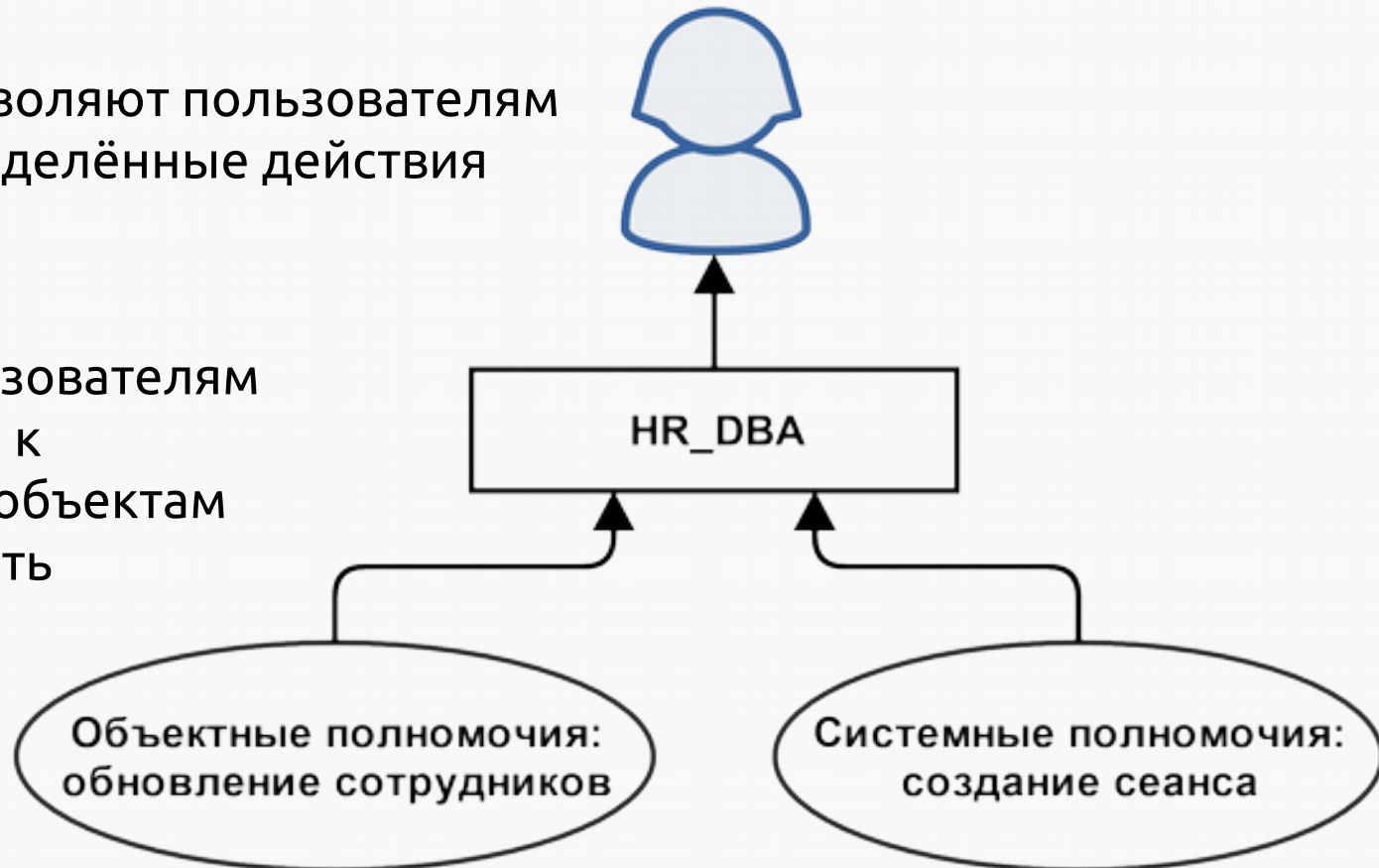
Оба этих формуляра не должны использоваться для стандартных операций.

Полномочия

Полномочие (privilege) — это право на выполнение определенного типа SQL-оператора или на доступ к объекту пользователя.

Существует два типа полномочий пользователя:

- Системные: позволяют пользователям выполнять определённые действия с базой данных.
- Объектные: позволяют пользователям получать доступ к определенным объектам и манипулировать ими.



Предоставление и аннулирование полномочий

- С помощью Enterprise Manager:



- С помощью SQL-операторов GRANT и REVOKE:

GRANT privileges ON object TO user;

REVOKE privileges ON object FROM user;

Например:

```
SQL> GRANT SELECT, INSERT, UPDATE ON suppliers TO smithj;
```

```
SQL> REVOKE ALL ON suppliers FROM anderson;
```

Системные полномочия

- Предоставление полномочий, которые содержат фразу ANY, означает выход за пределы схемы. Например, обладая полномочиями CREATE TABLE, можно создать таблицу, однако только в собственной схеме. А полномочия SELECT ANY TABLE позволяют выбирать среди таблиц других пользователей.
- Пользователь SYS и пользователи, владеющие ролью DBA, обладают всеми полномочиями ANY.
- SYSDBA и SYSOPER. Эти полномочия позволяют выполнять административные задачи в БД. SYSOPER позволяет пользователю выполнять оперативные задачи, но без возможности просмотра пользовательских данных. В него входят следующие системные полномочия:
 - STARTUP и SHUTDOWN
 - CREATE SPFILE
 - ALTER DATABASE OPEN/MOUNT/BACKUP
 - ALTER DATABASE ARCHIVELOG
 - ALTER DATABASE RECOVER (только полное восстановление).
 - RESTRICTED SESSION

Системное полномочие SYSDBA дополнительно дает разрешение на неполное восстановление и удаление базы данных.

Системные полномочия (продолжение)

- **SYSASM.** Позволяет запустить, остановить экземпляр ASM и управлять им.
- **DROP ANY объект.** Позволяет удалять объекты других пользователей схемы.
- **CREATE, MANAGE, DROP и ALTER TABLESPACE.** Позволяют управлять табличными пространствами.
- **CREATE LIBRARY.** В БД Oracle имеется возможность создавать и вызывать внешний код (например библиотеку C) с помощью PL/SQL. Такой библиотеке должно быть присвоено имя объектом LIBRARY в базе данных.
- **CREATE ANY DIRECTORY.** В качестве меры безопасности каталог ОС, в котором хранится код, необходимо связать с виртуальным объектом каталога Oracle. Владелец полномочия CREATE ANY DIRECTORY потенциально может вызвать небезопасные объекты кода.
- **GRANT ANY OBJECT PRIVILEGE.** Позволяет предоставлять разрешения для объектов, которыми его обладатель не владеет.
- **ALTER DATABASE и ALTER SYSTEM.** Позволяют изменять базу данных и экземпляр Oracle (например, можно переименовать файл данных или очистить кэш буфера).

Объектные полномочия

`GRANT privileges ON object TO user;`

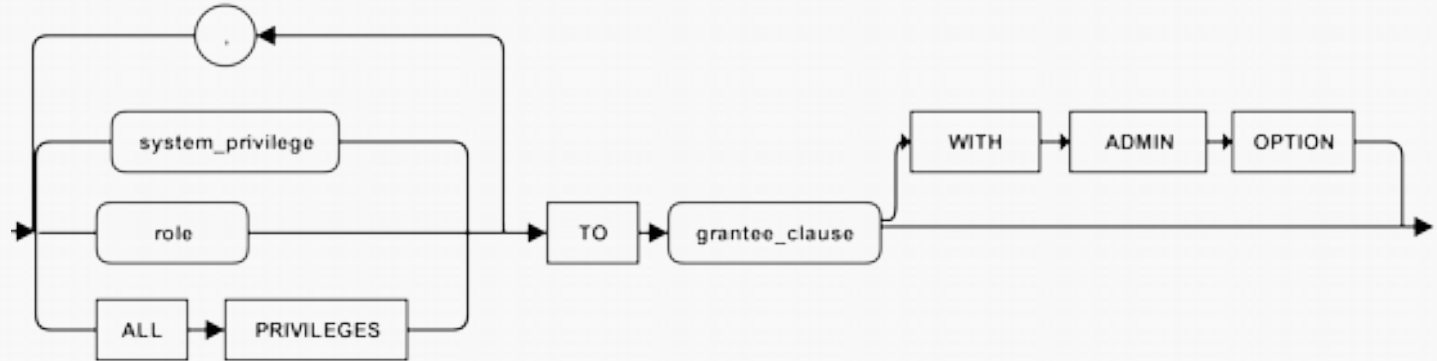
`REVOKE privileges ON object FROM user;`

Виды полномочий:

- `SELECT` — возможность выполнять выборку данных из таблицы.
- `INSERT` — возможность добавлять новые строки в таблицу.
- `UPDATE` — возможность изменять данные в таблице.
- `DELETE` — возможность удалять строки из таблицы.
- `REFERENCES` — возможность создавать ограничения целостности для выбранной таблицы.
- `ALTER` — возможность выполнять оператор `ALTER TABLE` применительно к выбранной таблице.
- `INDEX` — возможность создавать индексы на столбцы выбранной таблицы.
- `ALL` — все перечисленные выше полномочия.

Параметры ADMIN OPTION и GRANT OPTION

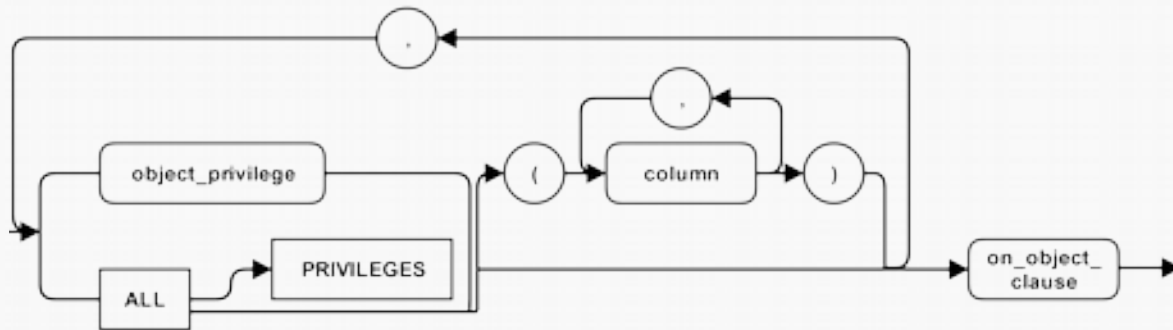
ADMIN OPTION:



Позволяет:

- Выдать эти системные полномочия другому формуляру или роли.
- Аннулировать эти полномочия у другого формуляра или роли.

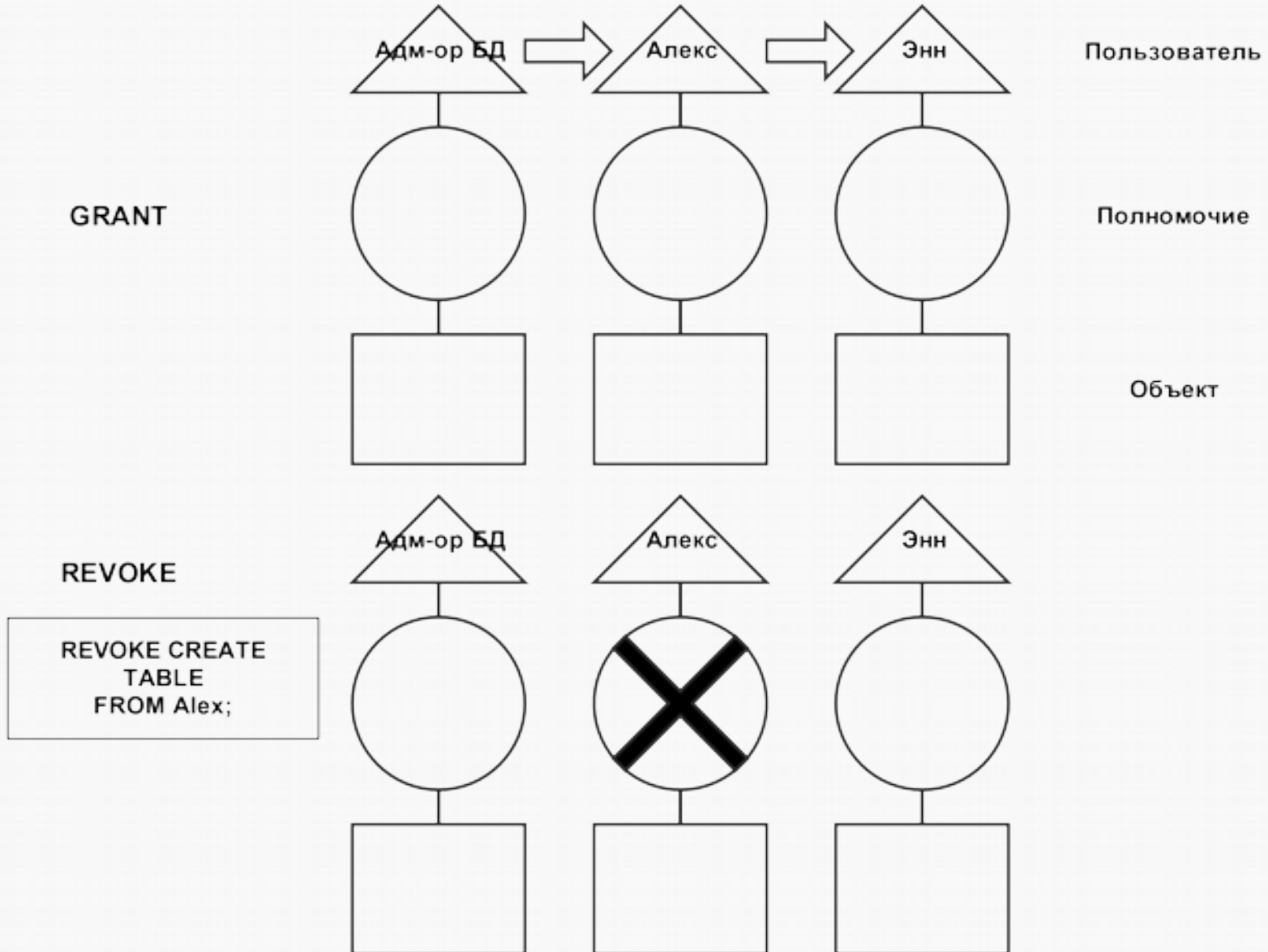
GRANT OPTION:



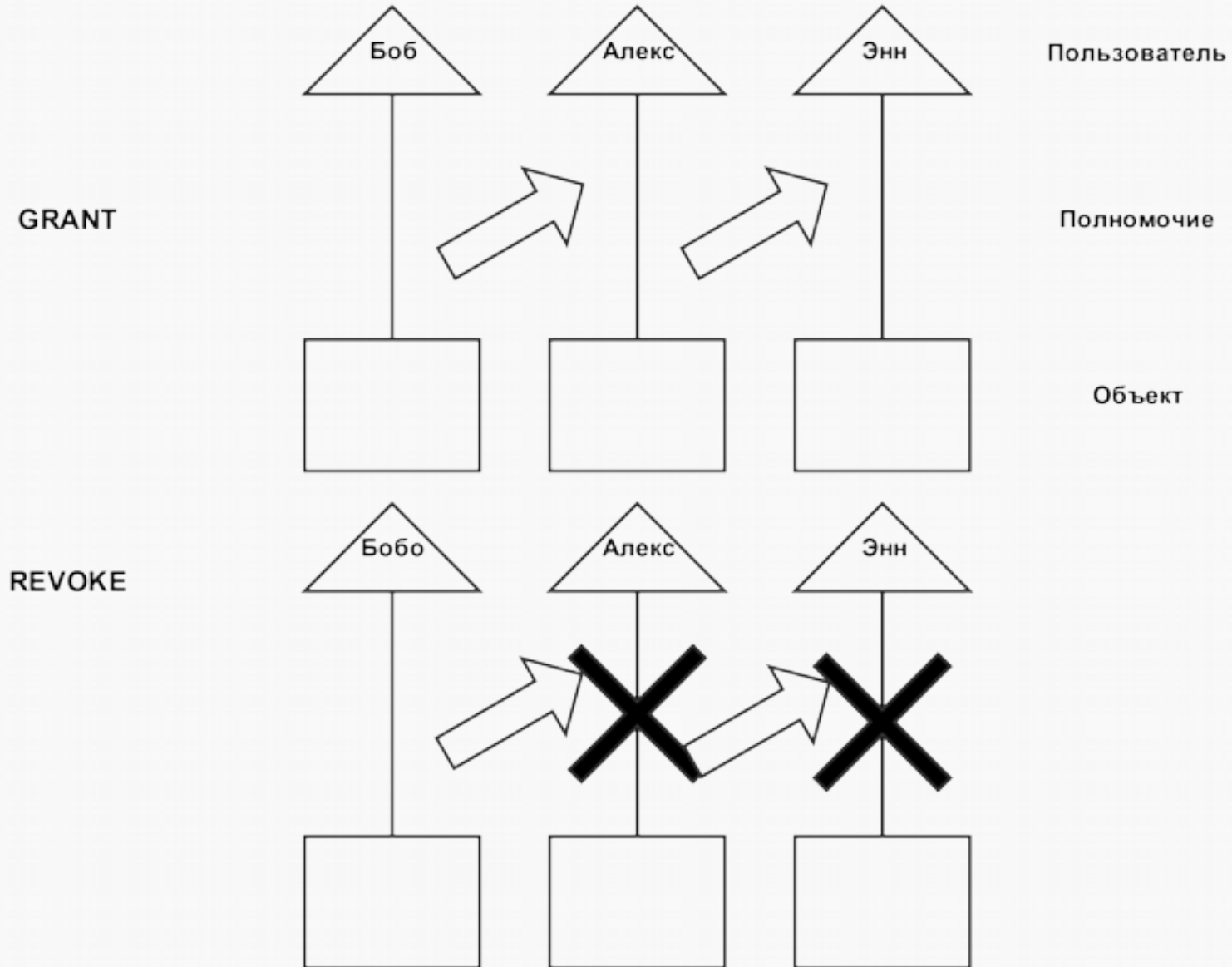
Позволяет:

- Выдать эти объектные полномочия другому формуляру или роли.

Аннулирование системных полномочий с параметром ADMIN OPTION



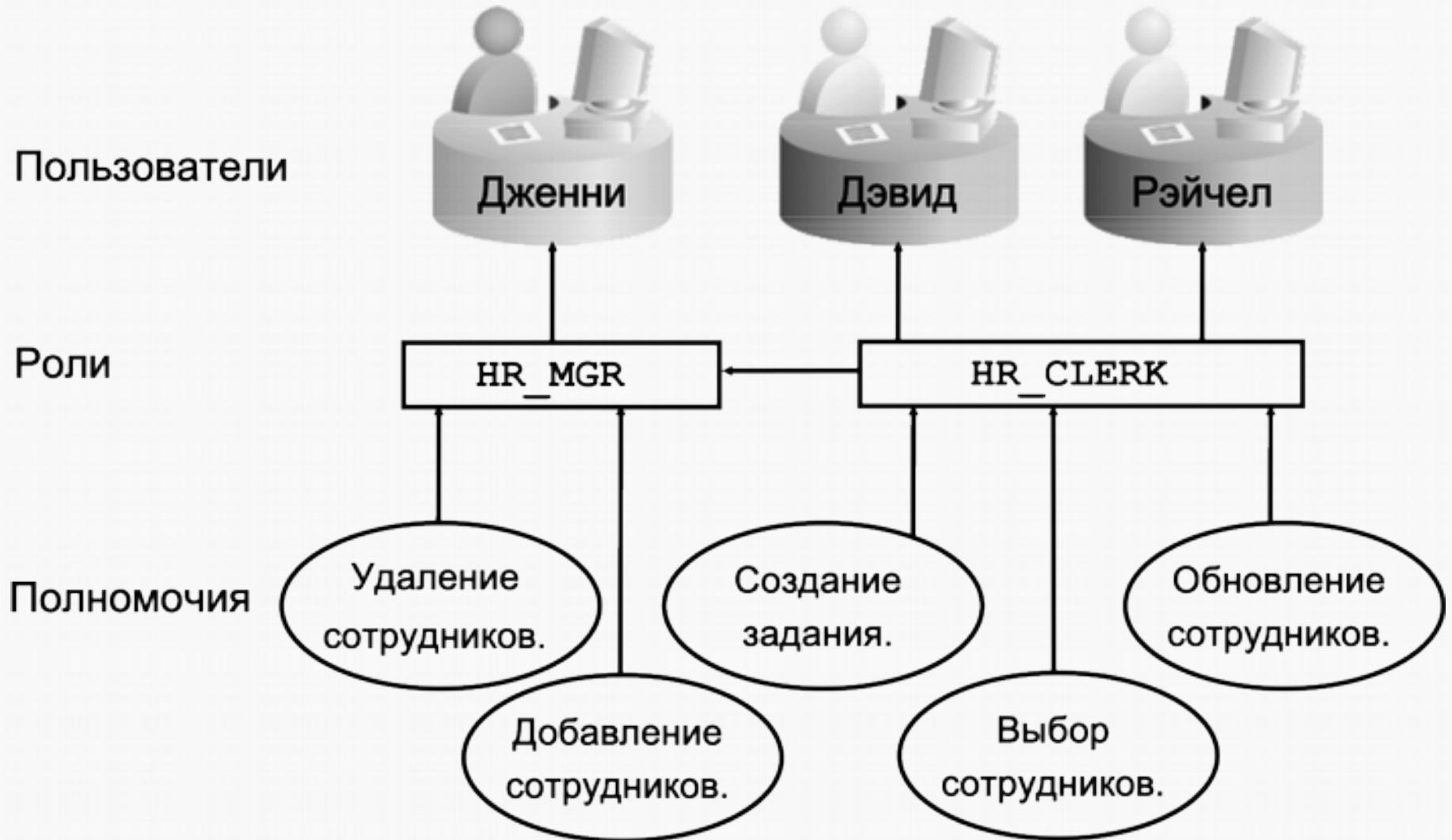
Аннулирование объектных полномочий с параметром GRANT OPTION



Преимущества использования ролей

- **Упрощенное управление полномочиями.** Роли используются для упрощенного управления полномочиями. Вместо того чтобы предоставлять нескольким пользователям один и тот же набор полномочий, можно предоставить эти полномочия роли, а затем предоставить эту роль пользователям.
- **Динамическое управление полномочиями.** Если полномочия, связанные с ролью, изменяются, все пользователи, которым предоставлена эта роль, немедленно автоматически получают обновленные полномочия.
- **Выбор доступных полномочий.** Роли можно включать и отключать, чтобы временно активировать или деактивировать полномочия. Таким образом можно контролировать полномочия пользователя в той или иной ситуации.

Полномочия, роли и пользователи



Предопределённые роли

Роль	Полномочия
CONNECT	CREATE SESSION
RESOURCE	CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
SCHEDULER_ADMIN	CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
DBA	Большая часть системных полномочий; несколько других ролей. Следует предоставлять только администраторам.
SELECT_CATALOG_ROLE	Системных полномочий нет; HS_ADMIN_ROLE и более 1700 объектных полномочий в словаре данных

Назначение ролей пользователям

Используются те же самые операторы GRANT и REVOKE:

```
SQL> CREATE ROLE cust_serv_mgr;  
SQL> GRANT cust_serv_clerk TO cust_serv_mgr;  
SQL> GRANT insert, update ON customers TO cust_serv_mgr;  
SQL> GRANT delete ON issue_track TO cust_serv_mgr;  
SQL> GRANT cust_serv_mgr TO mary;  
  
SQL> REVOKE insert ON customers FROM cust_serv_mgr;  
SQL> REVOKE cust_serv_mgr FROM mary;
```

Профили

- *Профили (profiles)* — это именованные наборы ограничений на использование ресурсов БД и экземпляра.
- Каждому пользователю назначается профиль, причем одновременно только один.
- Если при изменении профиля пользователь уже находился в системе, изменения вступят в силу со следующего сеанса работы пользователя.
- Профиль DEFAULT является базовым для всех остальных профилей.
- Профили не накладывают ограничения на использование ресурсов пользователями, если параметру инициализации RESOURCE_LIMIT не присвоено значение TRUE.
- Если параметру RESOURCE_LIMIT присвоено стандартное значение FALSE, ограничения на использование ресурсов, заданные профилем, игнорируются. Настройки пароля для профиля всегда применяются принудительно.

Профили (продолжение)

С помощью профилей администратор может контролировать следующие системные ресурсы:

- **Процессор.** Ресурсы процессора можно ограничивать на основе сеансов или вызовов:
 - Ограничение процессор/сеанс, равное 1000, означает, что если сеанс потребляет более 10 секунд процессорного времени, то будет вызвана ошибка и выполнен выход:
ORA-02392: exceeded session limit on CPU usage, you are being logged off
 - При ограничении вызовов место всего сеанса пользователя контролируется процессорное время, потребляемое отдельными командами. Если команда пользователя превышает его, её выполнение прерывается и пользователь получает сообщение об ошибке:
ORA-02393: exceeded call limit on CPU usage
- **Сеть/память.** Можно задать следующие параметры:
 - Время соединения: указывает интервал времени в минутах, в течение которого пользователь может быть соединен до того, как он будет автоматически отключен;
 - Время простоя: указывает интервал времени в минутах, в течение которого сеанс пользователя может бездействовать до того, как он будет автоматически отключен.
 - Параллельные сеансы: определяет, сколько параллельных сеансов может быть создано с помощью формуляра пользователя БД.
- **Личная область SGA.** Ограничивает размер области, выделяемой в глобальной системной области (SGA) для сортировки, слияния растровых изображений и т.д.
- **Дисковый ввод/вывод.** Ограничивает объем данных, которые пользователь может считать за сеанс или за вызов.
- Кроме того, для профиля можно задать **смешанные ограничения**.

Если пользователю не предоставлено системное полномочие `UNLIMITED TABLESPACE`, для создания объектов в табличном пространстве такому пользователю следует задать *квоту*.

Квота может иметь:

- значение в мегабайтах или килобайтах;
- неограниченное значение.

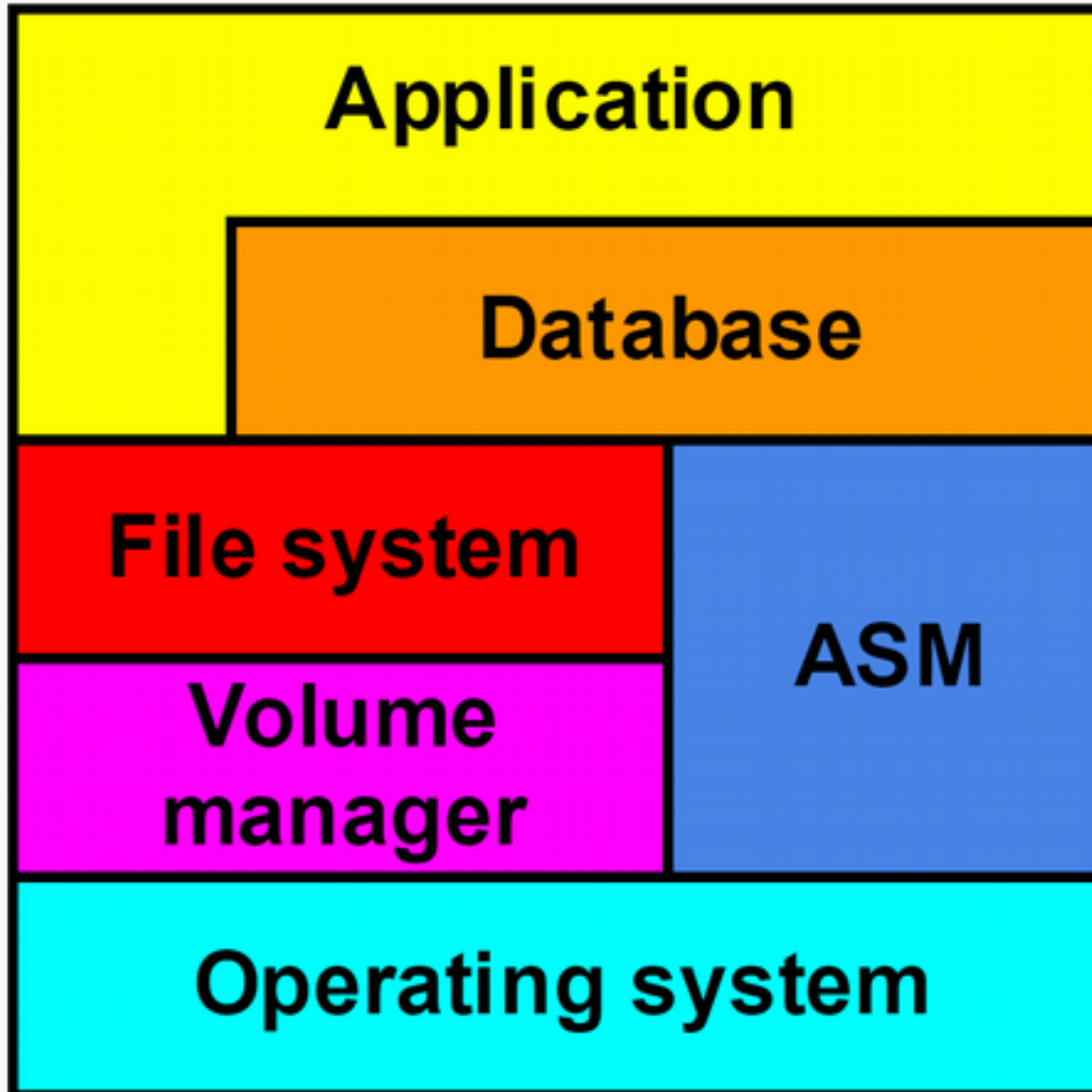
10. ASM — Automatic Storage Management

Automatic Storage Management (ASM) — технология в составе СУБД Oracle, реализующая автоматическое управление хранением данных на уровне менеджера томов.

Особенности ASM:

- Управление хранением осуществляется с помощью специальных экземпляров Oracle — ASM Instances.
- ASM может управлять хранением данных как на отдельной машине, так и на уровне кластера RAC в целом.
- Один экземпляр ASM может управлять хранением данных сразу нескольких БД.

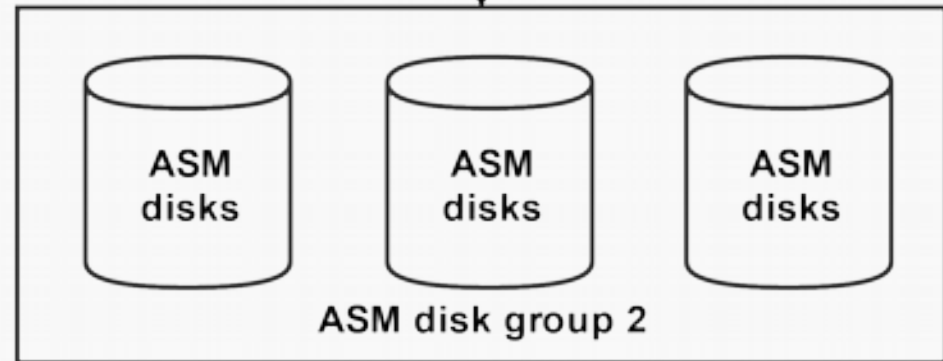
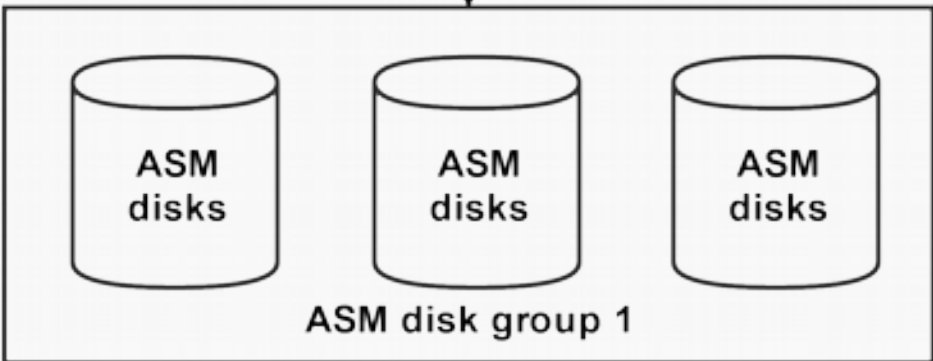
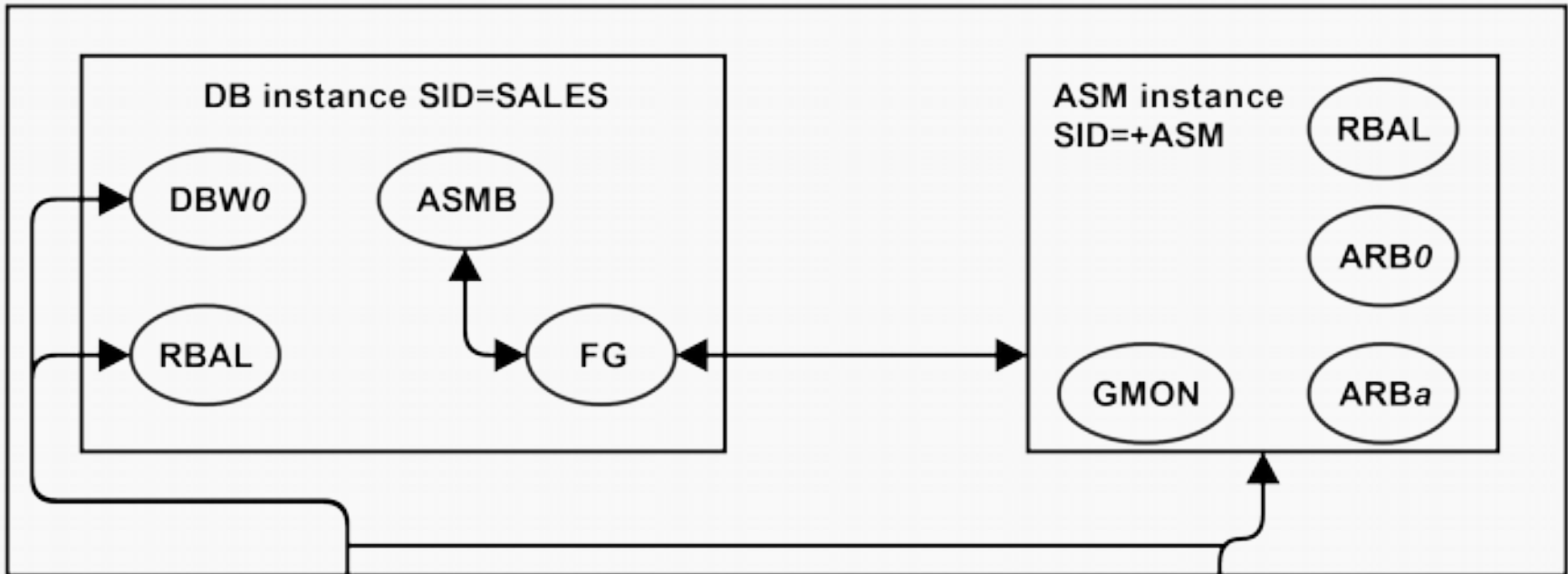
Обзор технологии (продолжение)



Другие особенности ASM:

- Управление операциями ввода/вывода осуществляется автоматически — «прозрачно» как для приложения, так и для администратора БД.
- Можно расширять хранилище, добавляя в него дополнительные накопители, без остановки БД.
- ASM может управлять созданием резервных копий данных самостоятельно, либо использовать для этого механизмы на уровне ОС и / или системы хранения данных.
- Файлы БД разделены на «блоки распределения» (Allocation Unit — AU). Для того, чтобы определить, где физически находятся файлы блока распределения, используются специальные индексы.
- При изменении ёмкости хранилища (например, при добавлении в него нового диска), файлы внутри AU автоматически перераспределяются пропорционально произошедшему изменению.
- ASM обеспечивает свой механизм зеркалирования, независимый от используемого менеджера томов.
- ASM реализует распределённое хранилище для всех основных файлов в составе экземпляра Oracle — файлов данных, файлов журнала повторов (как оперативных, так и архивных), управляющих файлов и т. д.
- ASM обеспечивает полную поддержку RAC.

Архитектура ASM



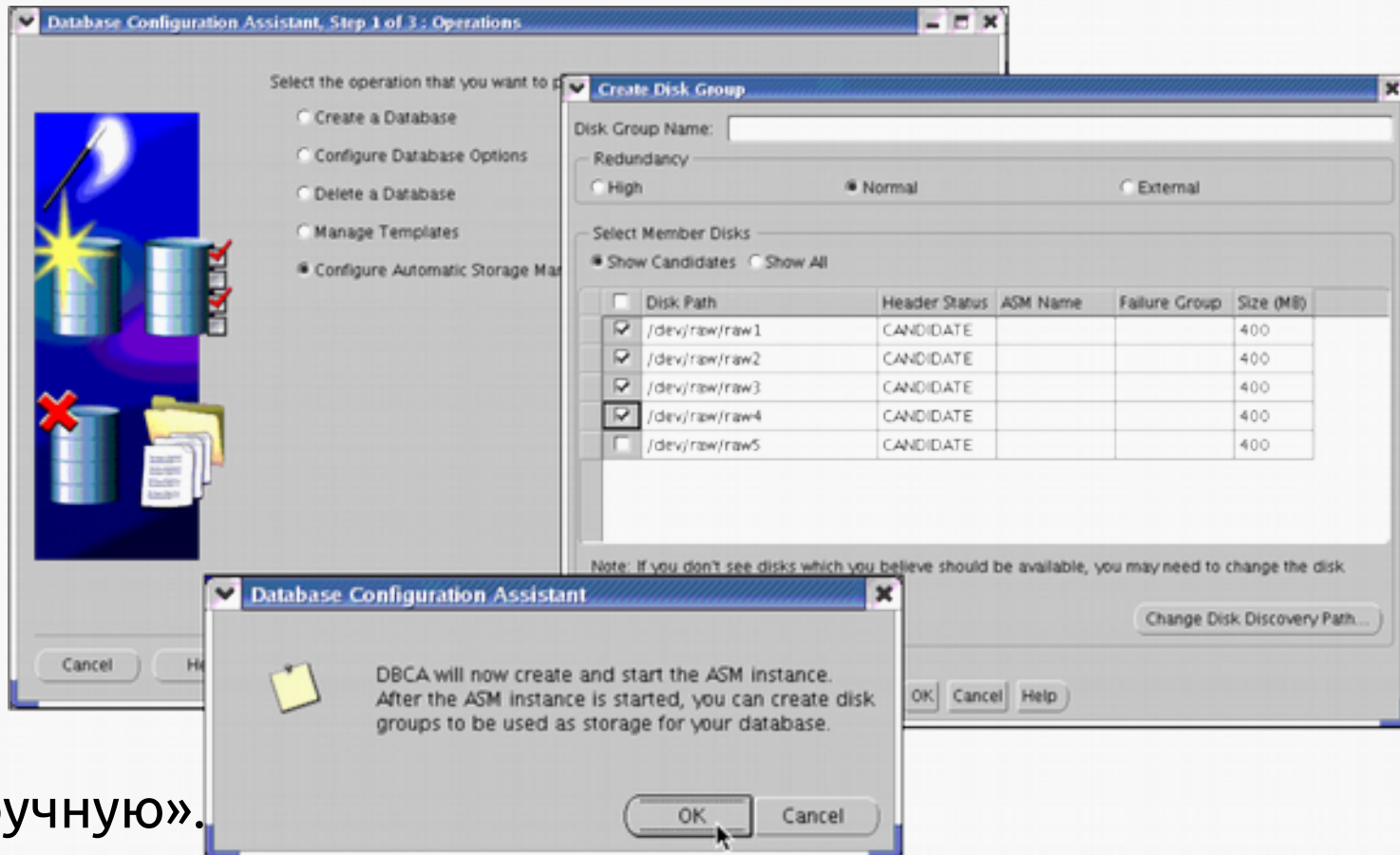
Архитектура ASM (продолжение)

- Распределённым хранилищем управляет отдельный экземпляр Oracle (ASM Instance), независимый от «основного» экземпляра БД. Этот экземпляр должен быть запущен до запуска экземпляра БД.
- Экземпляр БД получает у экземпляра ASM информацию о расположении необходимых ему файлов, после чего работает с ними напрямую, без участия экземпляра ASM.
- Диски, на которых хранятся данные, объединяются в логические группы — *disk groups*.
- Экземпляр ASM содержит ряд дополнительных фоновых процессов, которых нет в «обычном» экземпляре БД:
 - *RBAL* (Rebalance) — управляет перераспределением ресурсов при изменениях в дисковых группах.
 - *ARBn* (Asm Rebalance Process) — пул процессов, непосредственно осуществляющих перемещение данных AU между дисками.
 - *GMON* (Group Monitor) — осуществляет мониторинг состояния дисков в группах.
- В случае использования ASM, в экземпляре БД также появляются «дополнительные» процессы:
 - *RBAL* — управляет доступом к дискам в группах.
 - *ASMB* (ASM Background Process) — осуществляет взаимодействие с экземпляром ASM.

Создание экземпляра ASM

2 стандартных способа:

- С помощью DBCA:



- «Вручную».

Параметры инициализации экземпляра ASM

При создании экземпляра ASM используется файл параметров — такой же, как и при создании «обычного» экземпляра Oracle. Тем не менее, он содержит ряд специфичных для ASM параметров:

- `INSTANCE_TYPE` — должен быть задан как ASM.
- `DB_UNIQUE_NAME` — имя сервиса ASM.
- `ASM_POWER_LIMIT` — определяет количество ресурсов, которые может использовать ASM при «перевыравнивании» БД. Чем выше это значение, тем быстрее ASM выполняет перевыравнивание, но тем больше он при этом потребляет ресурсов. Принимает значения от 1 до 11.
- `ASM_DISKSTRING` — параметр, определяющий набор дисков, которые «видит» ASM.
- `ASM_DISKGROUPS` — список имён дисковых групп, которые «видит» ASM в момент запуска.

Запуск экземпляра ASM

```
$ export ORACLE_SID='+ASM'  
$ sqlplus /nolog  
SQL> CONNECT / AS sysasm  
Connected to an idle instance.  
SQL> STARTUP;  
Total System Global Area      284565504 bytes  
Fixed Size                     1299428 bytes  
Variable Size                  258100252 bytes  
ASM Cache                       25165824 bytes  
ASM diskgroups mounted
```

SYSASM — администратор экземпляра ASM:

```
SQL> CONNECT / AS SYSASM
```

```
SQL> CREATE USER ossysasmusername IDENTIFIED by passwd;
```

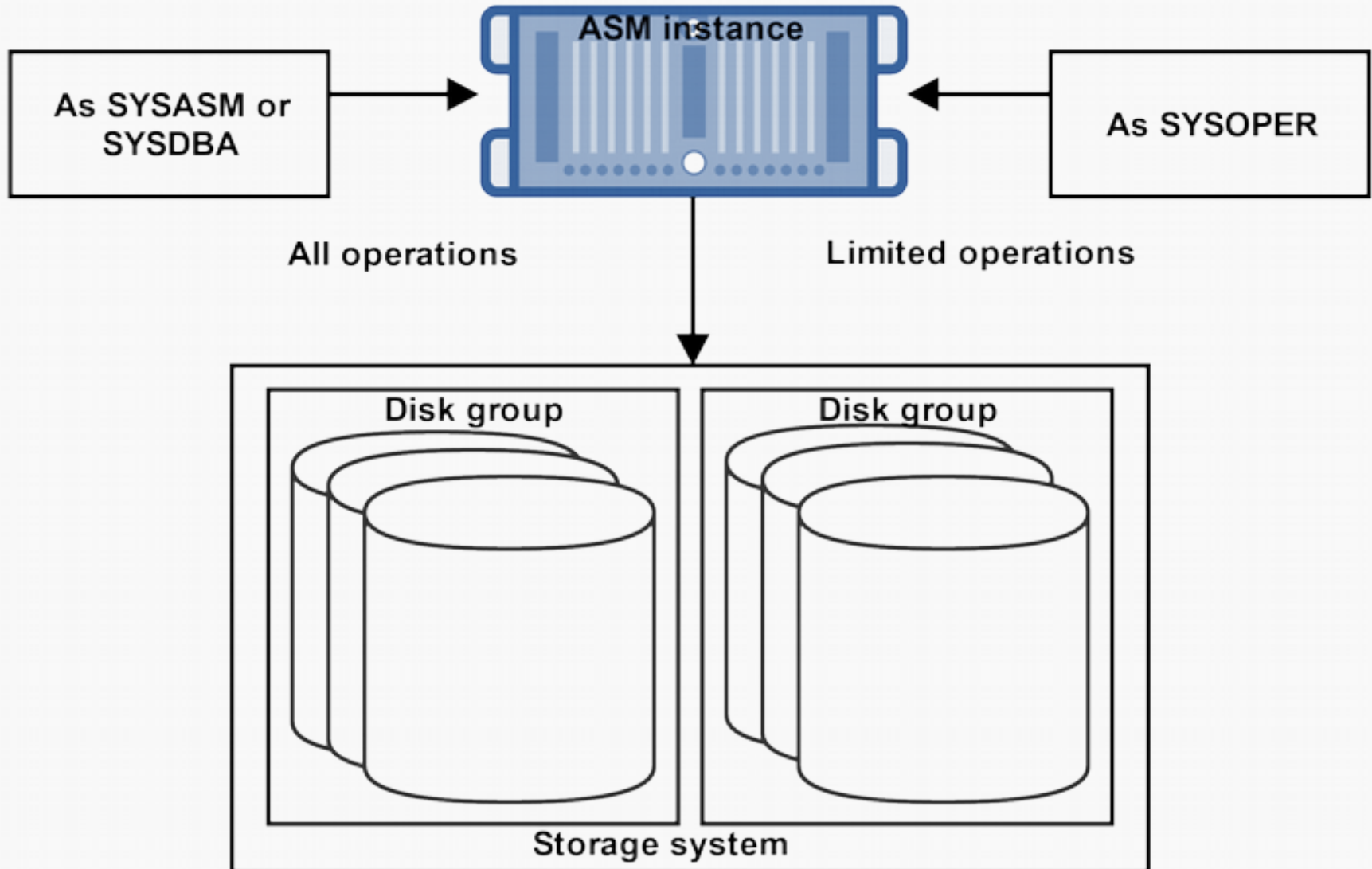
```
SQL> GRANT SYSASM TO ossysasmusername;
```

```
SQL> CONNECT ossysasmusername / passwd AS SYSASM;
```

```
SQL> DROP USER ossysasmusername;
```

По своим возможностям аналогична SYSDBA. В будущих версиях Oracle роль SYSDBA в экземплярах ASM будет недоступна.

Доступ к экземпляру ASM

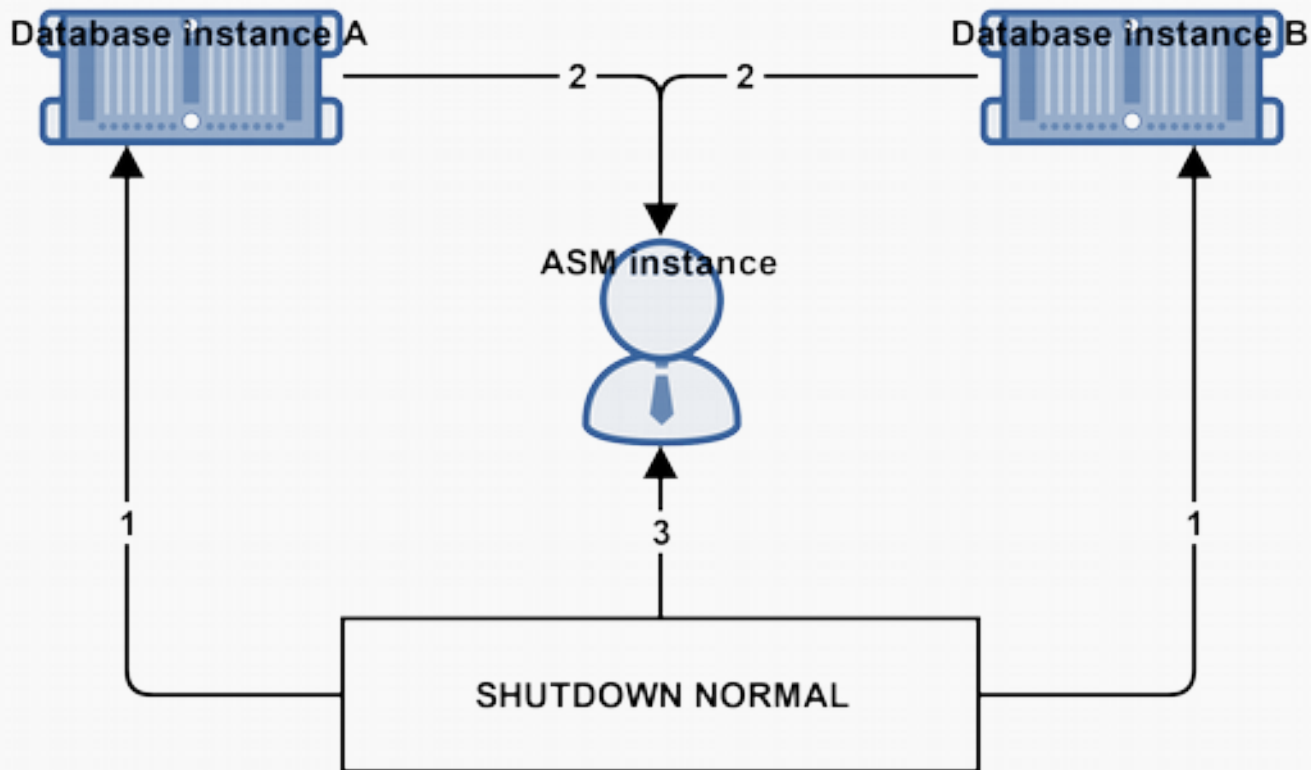


Доступ к экземпляру ASM (продолжение)

- У экземпляров ASM нет своего словаря данных, поэтому к ним можно подключаться только с использованием аутентификации на уровне ОС или файла паролей.
- Существует 3 роли, под которыми пользователь может подключиться к ASM:
 - SYSASM и SYSDBA — административный доступ без каких-либо ограничений.
 - SYSOPER — набор доступных SQL-команд ограничен минимально необходимым для обслуживания уже сконфигурированной системы:
 - STARTUP/SHUTDOWN
 - ALTER DISKGROUP MOUNT/DISMOUNT
 - ALTER DISKGROUP ONLINE/OFFLINE DISK
 - ALTER DISKGROUP REBALANCE
 - ALTER DISKGROUP CHECK
 - SELECT all V\$ASM_* views

Все остальные команды, в частности, CREATE DISKGROUP, ADD/DROP/RESIZE DISK и т. д., требуют наличия привилегий SYSASM или SYSDBA.

Остановка экземпляра ASM

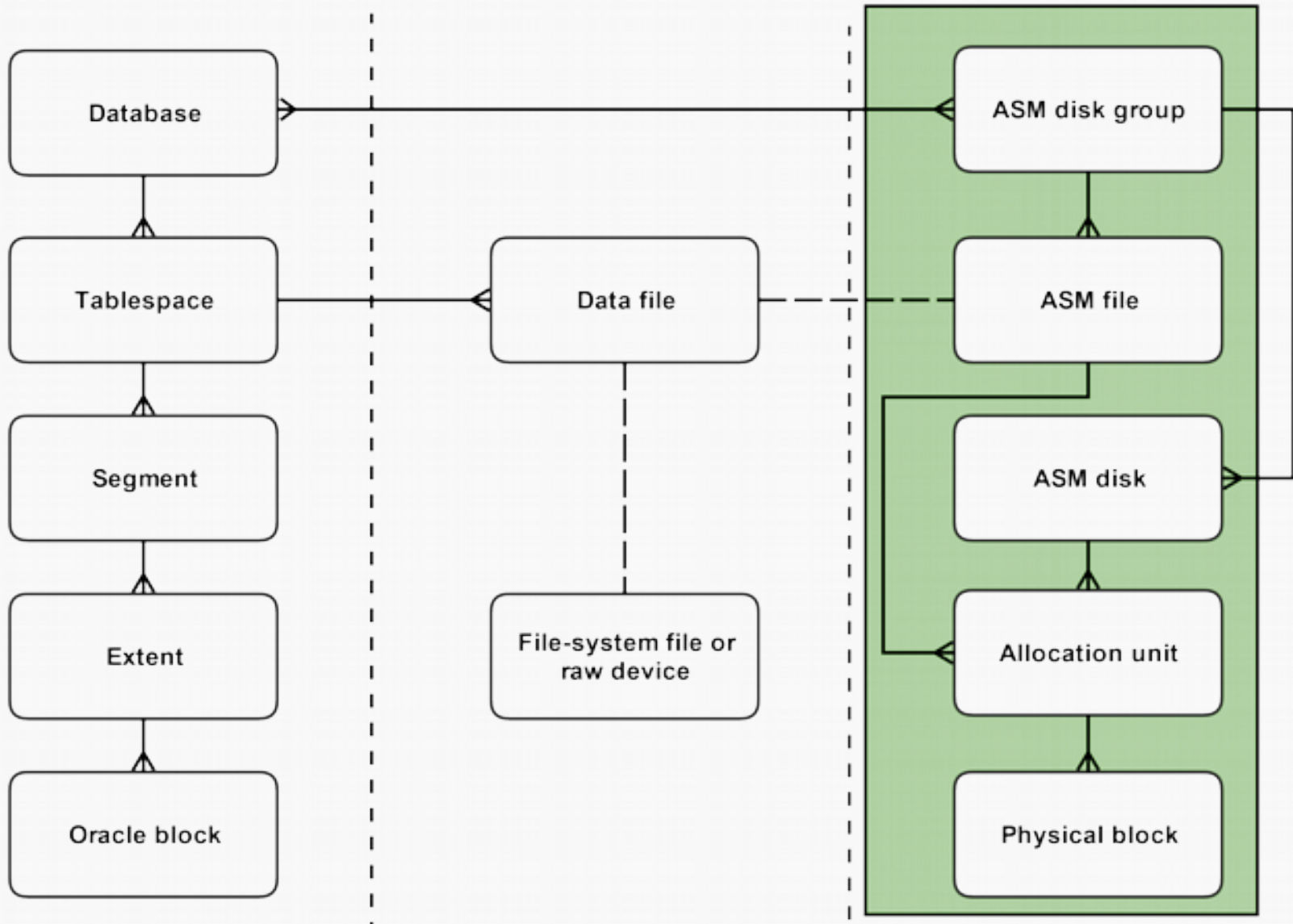


- Если попытаться остановить экземпляр ASM с помощью команд SHUTDOWN NORMAL, IMMEDIATE или TRANSACTIONAL, и к этому экземпляру подключен как минимум 1 работающий экземпляр БД, то будет получена ошибка:

ORA-15097: cannot SHUTDOWN ASM instance with connected RDBMS instance

- Если попытаться остановить его с помощью команды SHUTDOWN ABORT, то экземпляр ASM будет остановлен, но при следующем запуске потребуются его восстановление.

Хранилище данных ASM

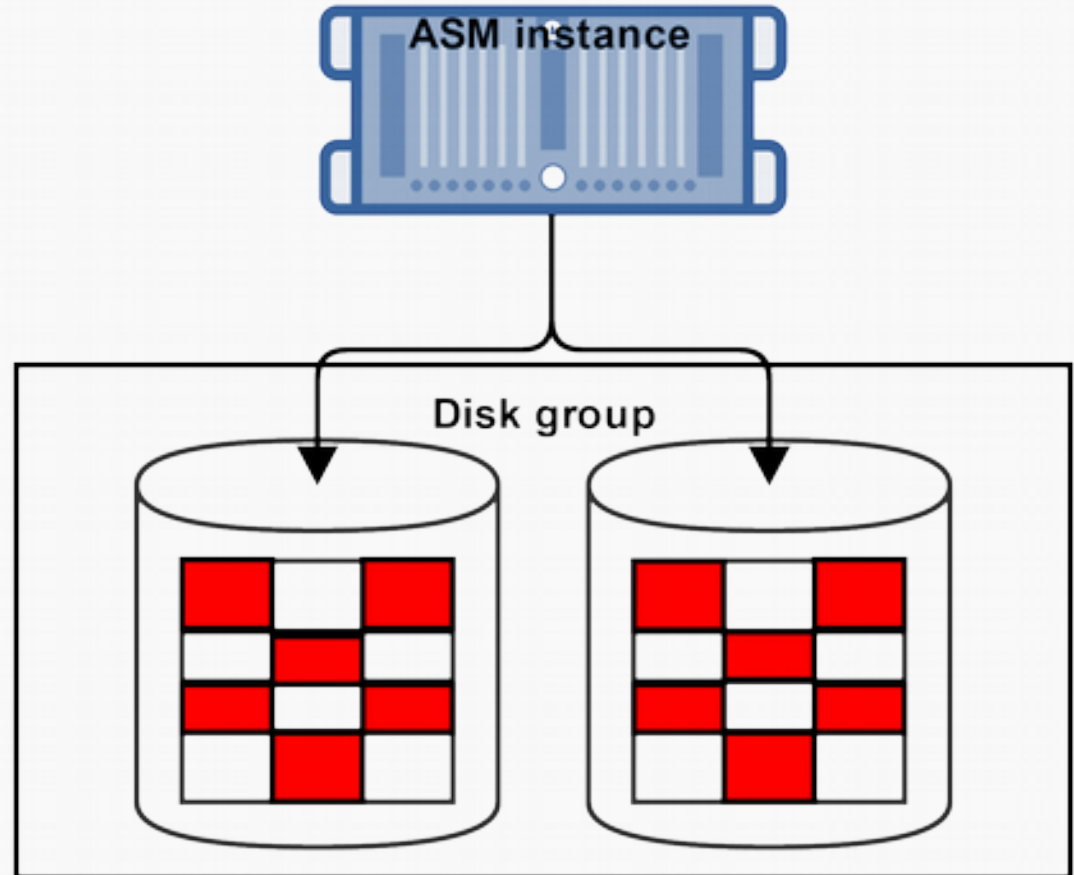


Хранилище данных ASM (продолжение)

- ASM никак не влияет на работу БД с ресурсами, существовавшими до конфигурации ASM.
- Новые файлы могут создаваться как файлы ASM, а существующие — продолжать администрироваться «по-старому», или быть перенесены под управление ASM.
- Верхний уровень иерархии — *дисковая группа ASM* (ASM disk group).
- Диски ASM разбиты на *блоки распределения* (allocation units — Aus):
 - Размер AU по умолчанию — 1 МБ.
 - AU — минимальный объём дискового пространства, которым может оперировать ASM.
 - Один блок данных может храниться только в одном конкретном AU.

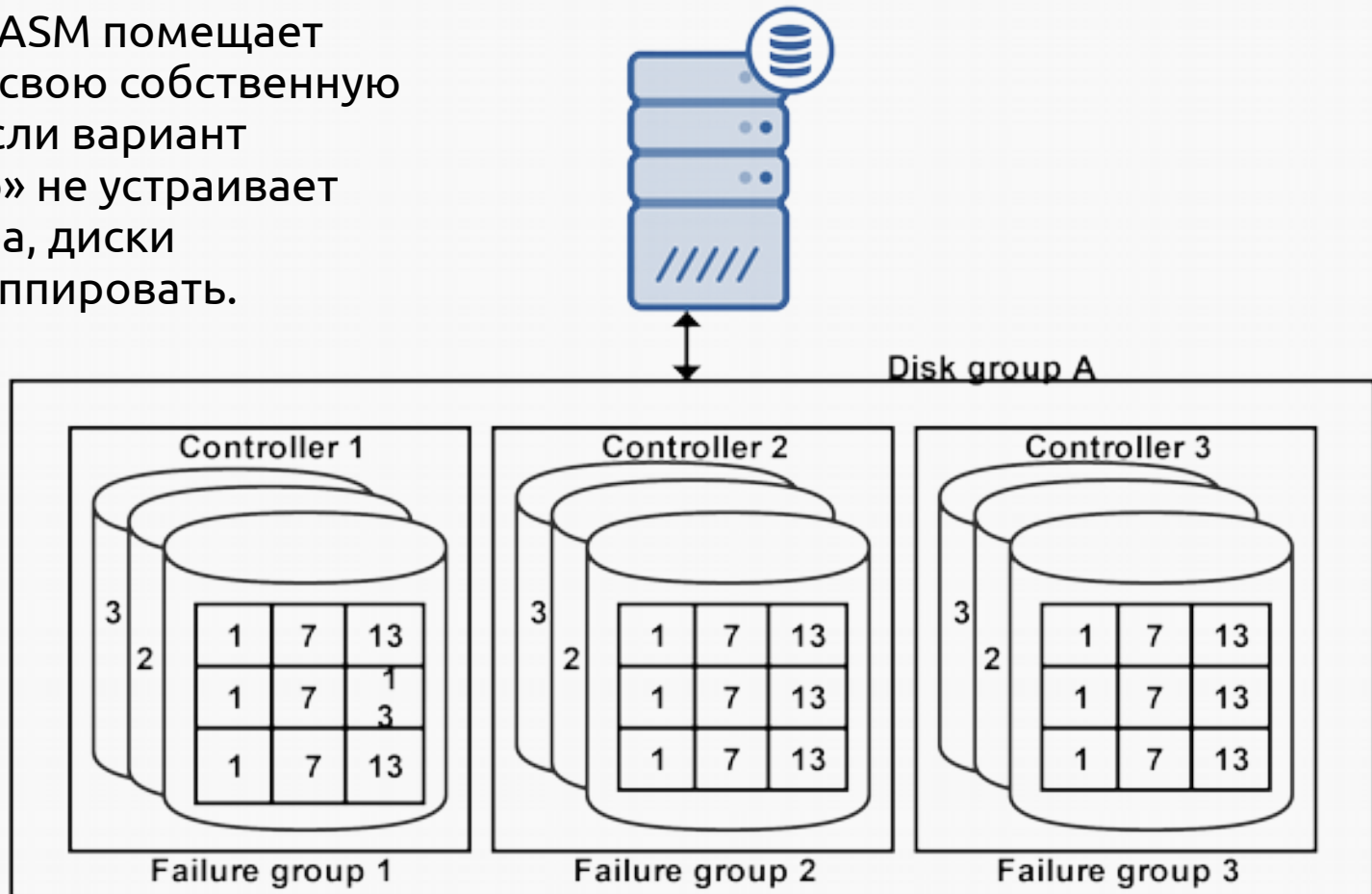
Дисковая группа ASM

- Дисковая группа — логическое объединение нескольких дисков ASM.
- Может хранить данные нескольких БД.
- Одна БД может хранить свои данные в нескольких дисковых группах.
- Диск может принадлежать только одной дисковой группе.
- Файл ASM может быть сохранён только на одной дисковой группе.
- Файлы хранятся распределённо — сразу на всех дисках, входящих в соответствующую группу.



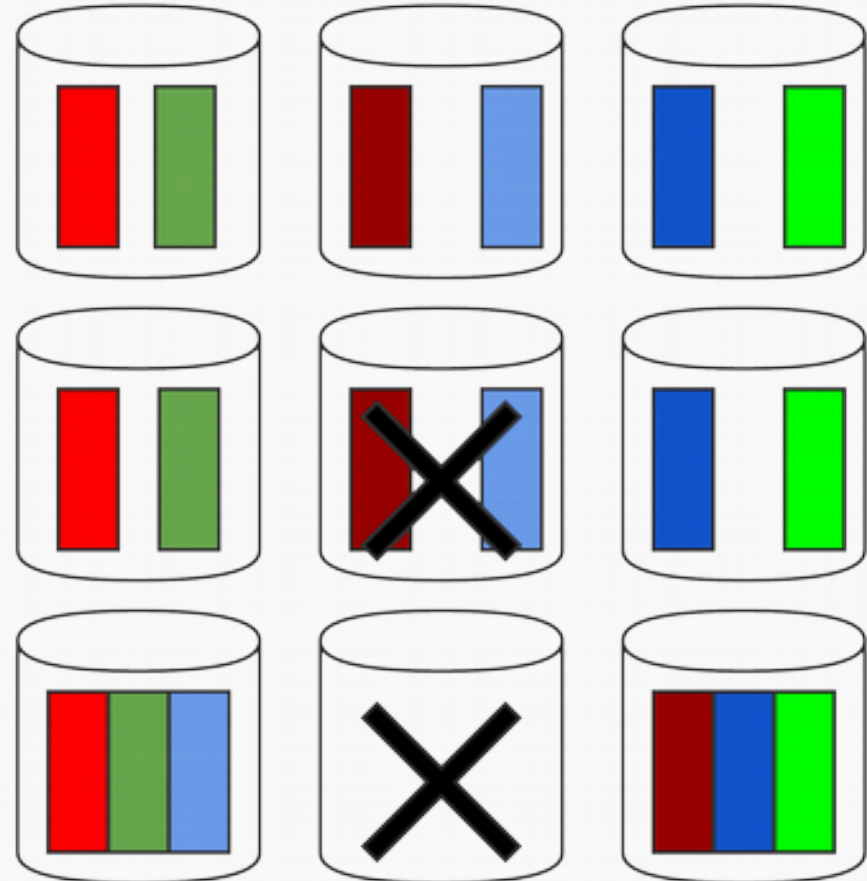
ASM Failure Group

- Набор дисков внутри конкретной группы, использующий общий ресурс (например, контроллер), от отказа которого должна быть обеспечена защита.
- По умолчанию ASM помещает каждый диск в свою собственную failure group. Если вариант «по умолчанию» не устраивает администратора, диски можно перегруппировать.
- ASM автоматически располагает данные так, чтобы отказ разделяемого ресурса failure group не привёл к потере данных.



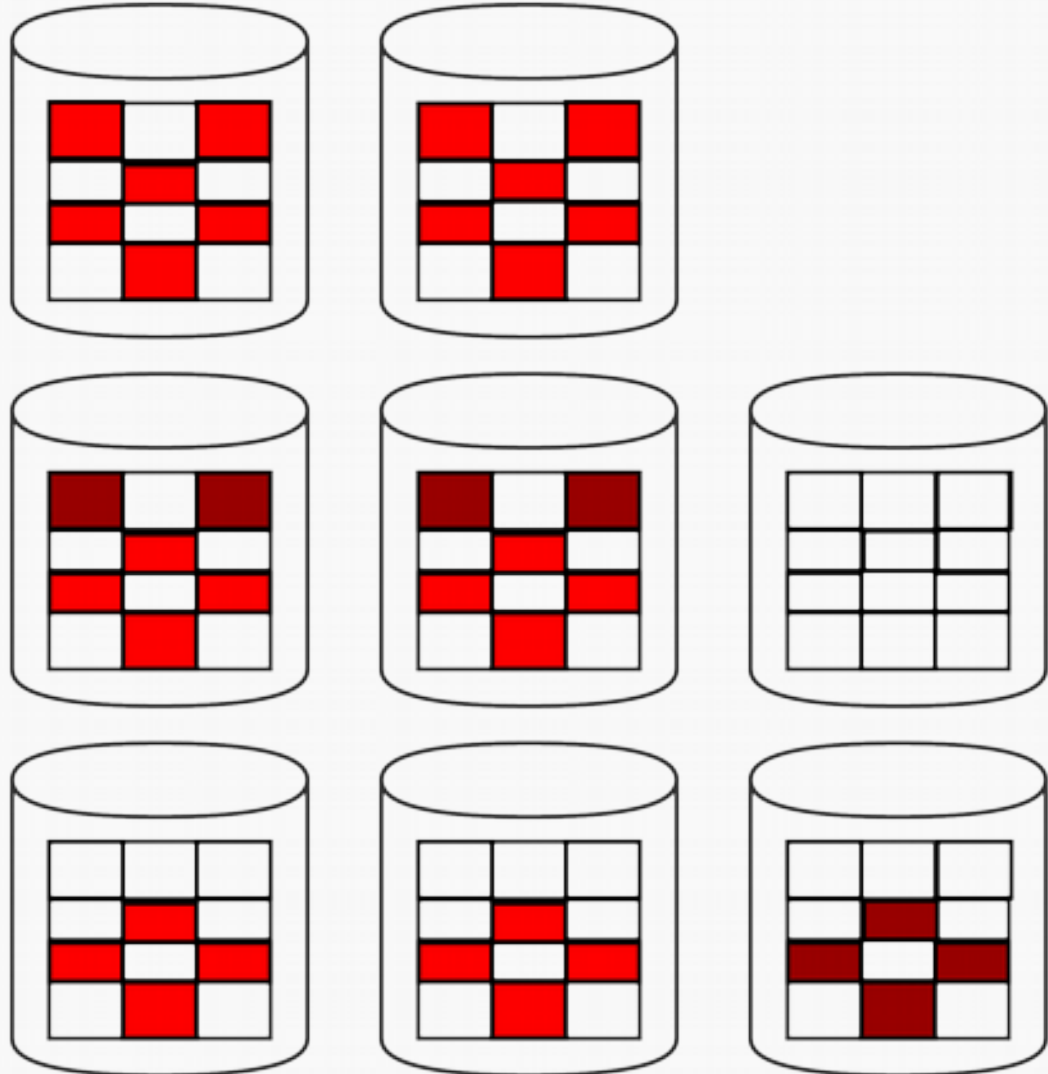
Зеркалирование дисковых групп

- Реализовано на уровне AU (а не на уровне дисков).
- На одном и том же диске могут храниться оригиналы и резервные копии различных AU.
- Если «оригинал» AU хранится на одном диске, то его «зеркало» будет храниться на другом диске в рамках той же группы.
- 3 режима организации избыточности данных:
 - «Внешняя» (External Redundancy) — используется аппаратная реализация зеркалирования.
 - «Стандартная» (Normal Redundancy):
 - Двукратное зеркалирование.
 - Как минимум, 2 failure groups.
 - «Высокая» (High Redundancy):
 - Тройное зеркалирование.
 - Как минимум, 3 failure groups.

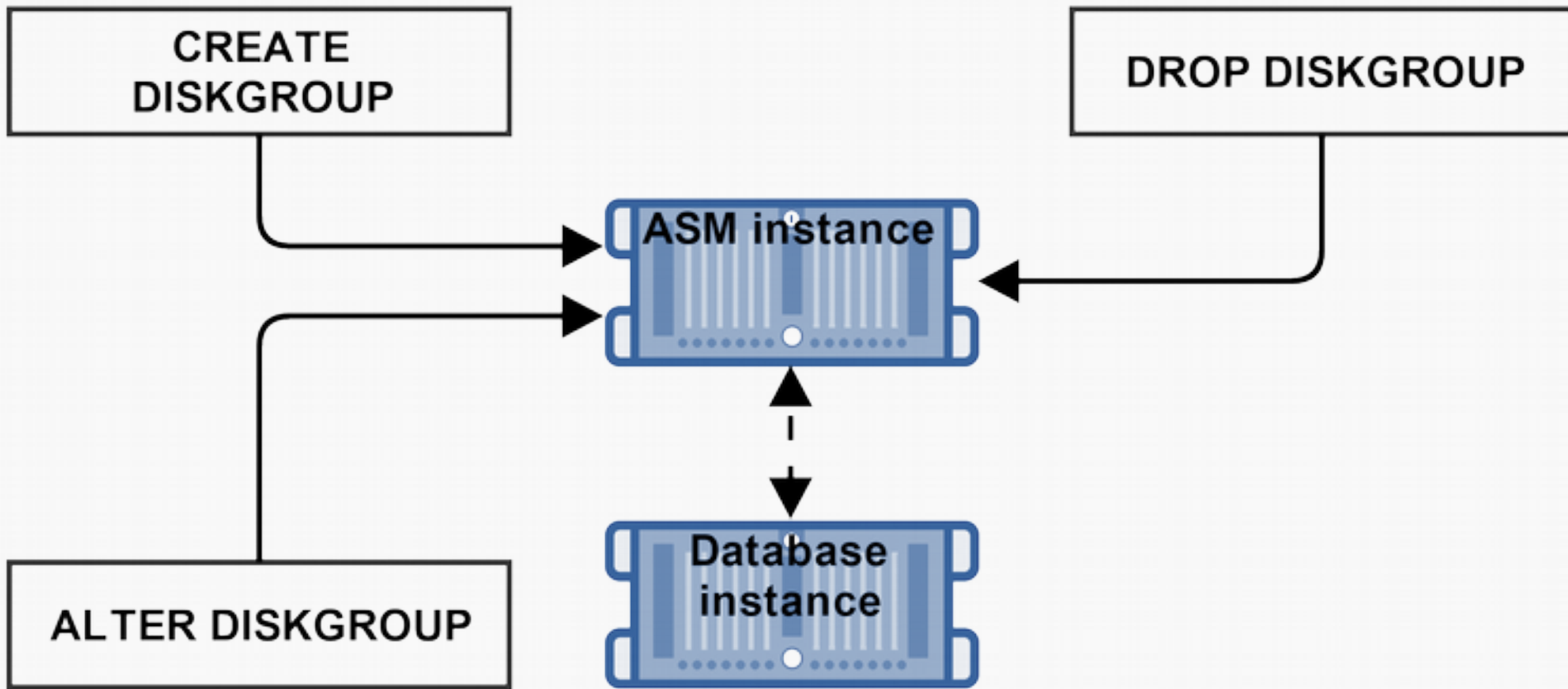


Динамическая «перевыравнивание» ДИСКОВЫХ ГРУПП

- Автоматически осуществляется при каждом изменении конфигурации хранилища (например, при добавлении в него нового диска).
- Не требует перезапуска БД или какого-либо ограничения доступа к ней.
- Перерасмещает данные в соответствии с новой ёмкостью хранилища.
- Осуществляется автоматически, никакой дополнительной конфигурации не требуется.
- Нагрузкой на систему можно управлять с помощью параметра `ASM_POWER_LIMIT`.



Управление дисковыми группами



Для всех операций требуются полномочия SYSDBA или SYSASM.

«А» и «В» - разные SCSI-контроллеры — создаём 2 fail groups:

```
CREATE DISKGROUP dgroupA NORMAL REDUNDANCY
FAILGROUP controller1 DISK
    '/devices/A1' NAME diskA1 SIZE 120G FORCE,
    '/devices/A2',
    '/devices/A3'
FAILGROUP controller2 DISK
    '/devices/B1',
    '/devices/B2',
    '/devices/B3';

DROP DISKGROUP dgroupA INCLUDING CONTENTS;
```

Добавление дисков в группы

```
ALTER DISKGROUP dgroupA ADD DISK  
  '/dev/rdisk/c0t4d0s2' NAME A5,  
  '/dev/rdisk/c0t5d0s2' NAME A6,  
  '/dev/rdisk/c0t6d0s2' NAME A7,  
  '/dev/rdisk/c0t7d0s2' NAME A8;
```

```
ALTER DISKGROUP dgroupA ADD DISK '/devices/A*';
```

Форматирование диска



Перебалансировка дисковой группы

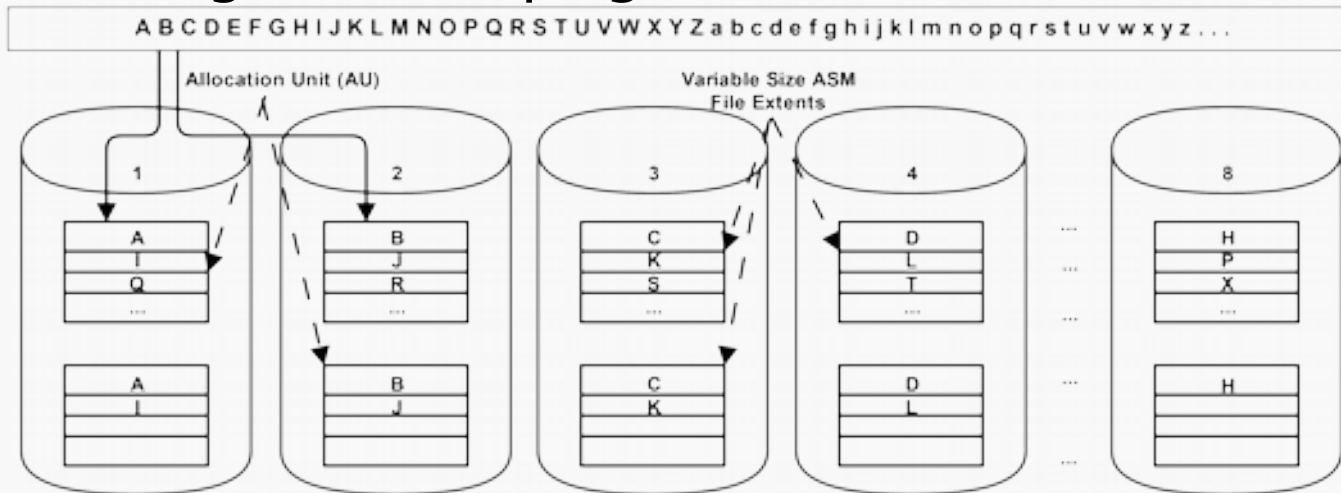
Атрибуты дисковых групп

Свойство	Create / Alter	Допустимые значения	Что определяет
<code>au_size</code>	C	1 2 4 8 16 32 64 MB	Размер AU
<code>compatible.rdbms</code>	AC	Валидная версия СУБД	Формат сообщений, которыми обмениваются БД и ASM
<code>compatible.asm</code>	AC	Валидная версия ASM	Формат, в котором хранит свои данные ASM
<code>disk_repair_time</code>	A	От 0 М до 2^{32} D	Время на восстановление, после которого диск переводится в состояние OFFLINE
<code>template.tname.redundancy</code>	A	UNPROTECT MIRROR HIGH	Способ организации зеркалирования
<code>template.tname.stripe</code>	A	COARSE FINE	Способ размещения данных на дисках

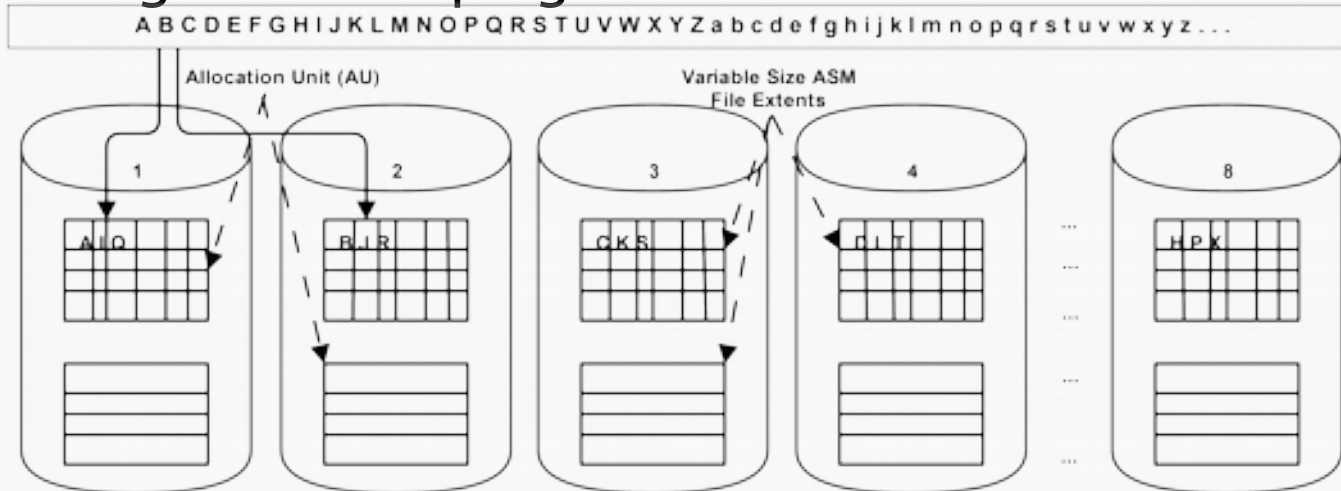
```
CREATE DISKGROUP DATA NORMAL REDUNDANCY
DISK '/dev/raw/raw1', '/dev/raw/raw2'
ATTRIBUTE 'compatible.asm'='11.1';
```

Coarse- & Fine-Grained Striping

Coarse-grained striping:



Fine-grained striping:



Удаление диска из dgroupA:

```
ALTER DISKGROUP dgroupA DROP DISK A5;
```

Удаление и добавление дисков одной командой:

```
ALTER DISKGROUP dgroupA  
    DROP DISK A6  
    ADD FAILGROUP fred  
DISK '/dev/rdisk/c0t8d0s2' NAME A9;
```

Отмена операции удаления диска:

```
ALTER DISKGROUP dgroupA UNDROP DISKS;
```

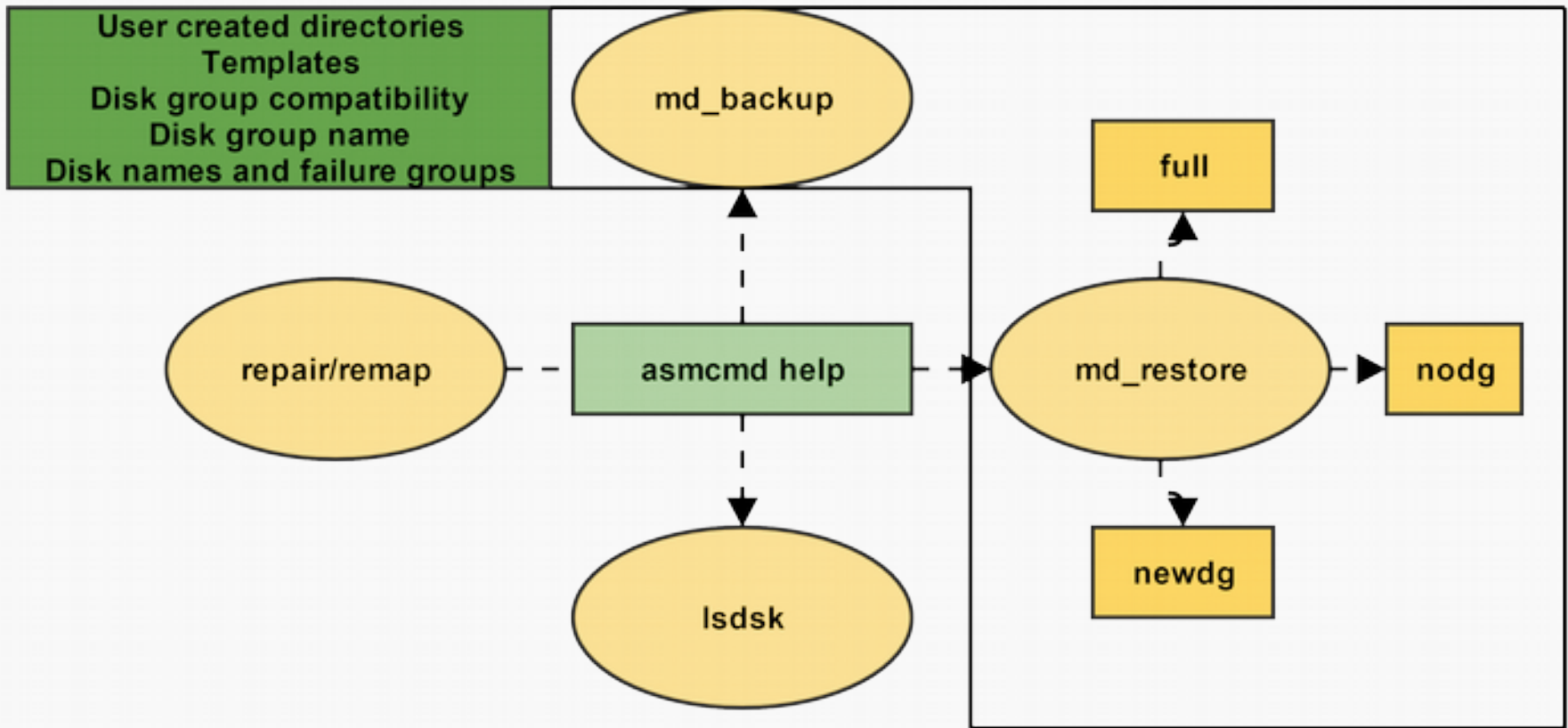
```
SQL> CREATE TABLESPACE tbsasm DATAFILE '+DGROUP1' SIZE 100M;  
Tablespace created.  
SQL> CREATE TABLESPACE hrapps DATAFILE '+DGROUP1' SIZE 10M;  
Tablespace created.
```

```
$ export ORACLE_SID=+ASM  
$ asmcmd
```

```
ASMCMO> ls -l DGROUP1/ORCL/DATAFILE
```

Type	Redund	Striped	Time	Sys	Name
DATAFILE	MIRROR	COARSE	OCT 05 21:00:00	Y	HRAPPS.257.570923611
DATAFILE	MIRROR	COARSE	OCT 05 21:00:00	Y	TBSASM.256.570922917

Утилита ASMCMD (продолжение)



```

ASMCMD> md_backup -b /tmp/dgbackup070222 -g admsk1 -g asmsk2
ASMCMD> md_restore -t full -g asmsk1 -i backup_file
ASMCMD> lsdsk -k DATA *_0001
  
```

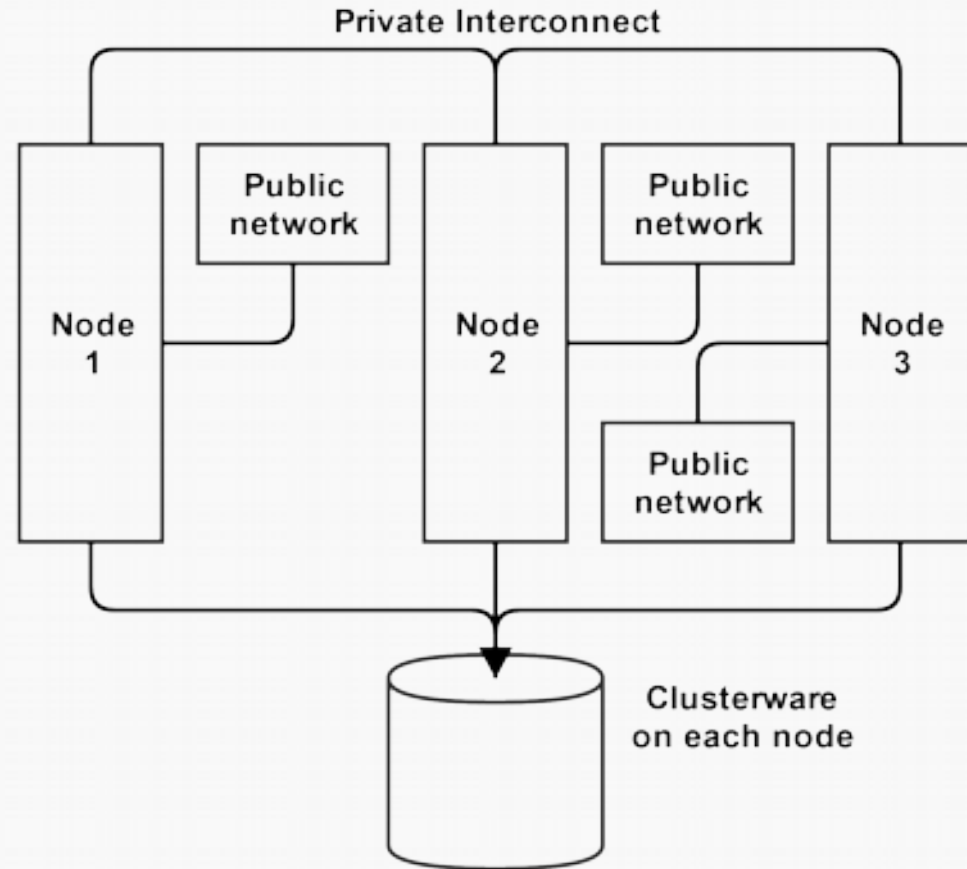
Масштабируемость и производительность ASM

- Размер экстента «растёт» автоматически пропорционально «росту» размера файла данных.
- Поддерживается переменный размер экстента, что позволяет:
 - Повысить максимальный объём файлов данных.
 - Снизить потребление памяти в разделяемом пуле.
- Некоторые лимиты и ограничения ASM:
 - До 63 дисковых групп.
 - До 10 000 дисков (суммарно во всех группах).
 - Максимальный размер каждого диска — 4 петабайта.
 - Максимальный размер всей системы хранения — 40 эксабайт.
 - Максимальное количество файлов в каждой дисковой группе — 1 млн.

11. RAC — Real Application Clusters

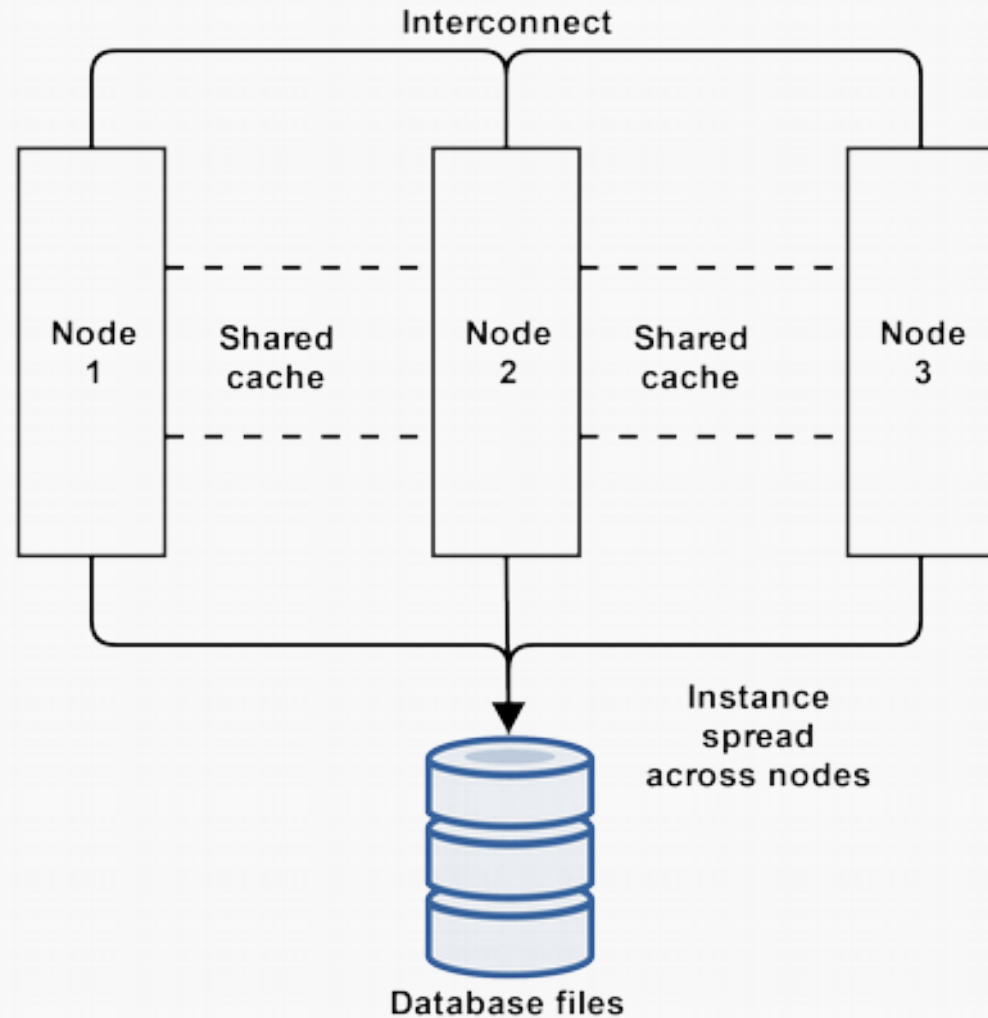
Кластер БД

- Кластер состоит из двух или более взаимосвязанных машин — узлов кластера.
- Узлы кластера «видны» снаружи как единое целое.
- Внутренняя структура кластера «скрывается» кластерным ПО — все кластеры «выглядят» снаружи как обычные серверы БД.
- Все диски доступны для чтения и записи всем узлам кластера.
- На всех узлах кластера используется одна и та же версия ОС.



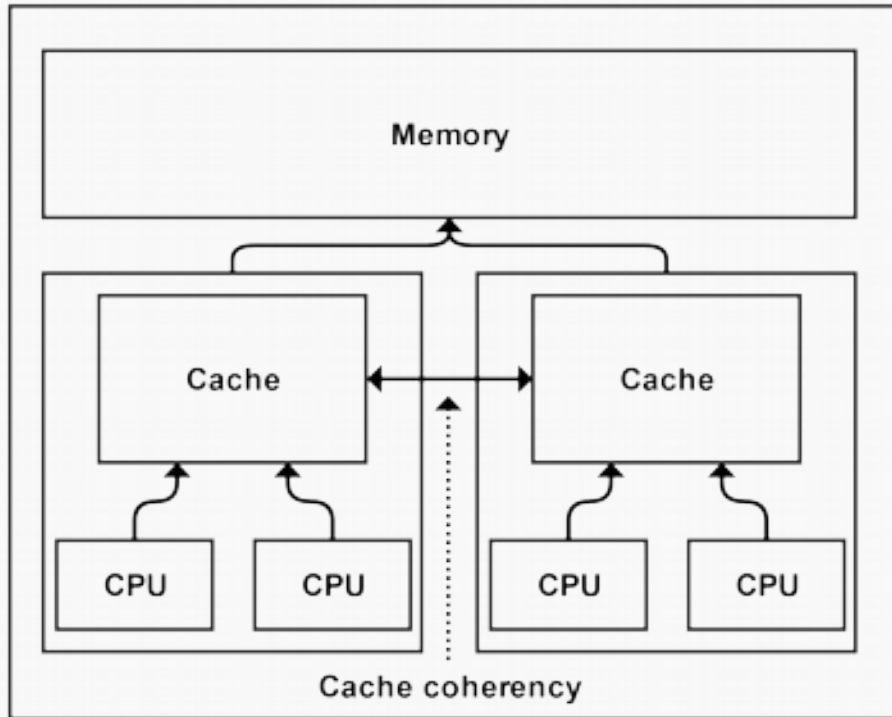
Oracle RAC

- Множество экземпляров Oracle составляют единую БД.
- На каждой машине запущено по одному экземпляру Oracle.
- Файлы БД — «общие» для всех экземпляров Oracle внутри кластера.
- Доступом к данным и взаимодействием между экземплярами Oracle управляет специальное инфраструктурное ПО.

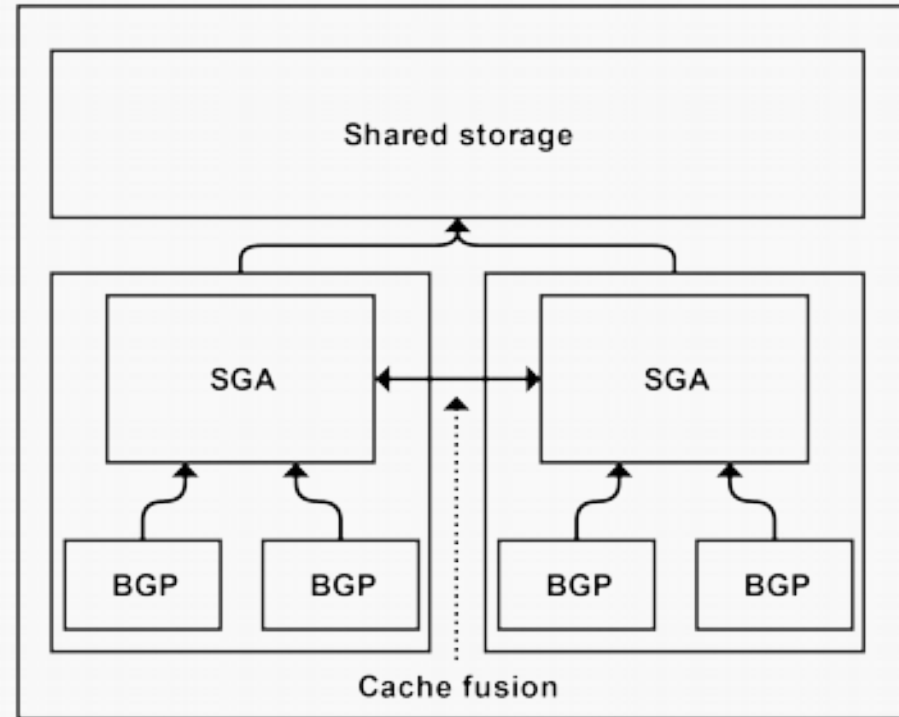


Кластеры vs SMP

SMP model



RAC model



Концепции очень похожи, т. е., если приложение хорошо масштабируется на SMP, то оно будет хорошо масштабироваться и на кластере.

Уровни масштабируемости

Для того, чтобы использование кластера было эффективным, требуется обеспечить масштабируемость на всех уровнях:

- Аппаратный — скорость чтения / записи на диски.
- Взаимодействие между узлами — пропускная способность сети и время отклика.
- Операционная система — возможность работы на многопроцессорных машинах.
- СУБД — синхронизация при параллельном доступе к данным.
- Приложение — особенности архитектуры.

Масштабируемость приложений: Scaleup & Speedup

Cluster system
scaleup

Original system

Hardware

Time

Cluster system
speedup

Hardware

Time

Up to 200% of task

Hardware

Time

Up to 300% of task

Hardware

Time

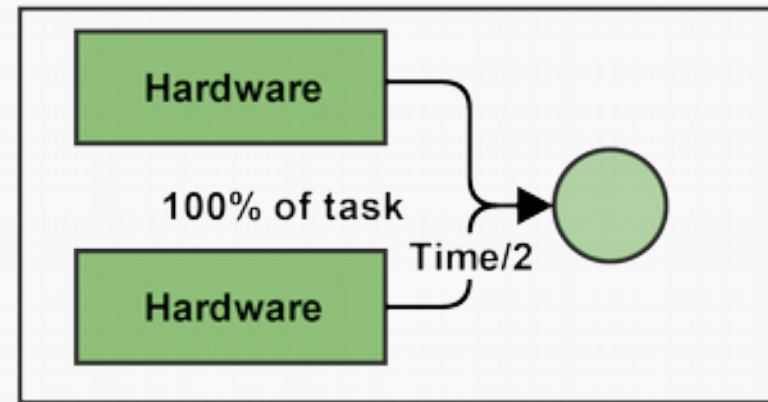
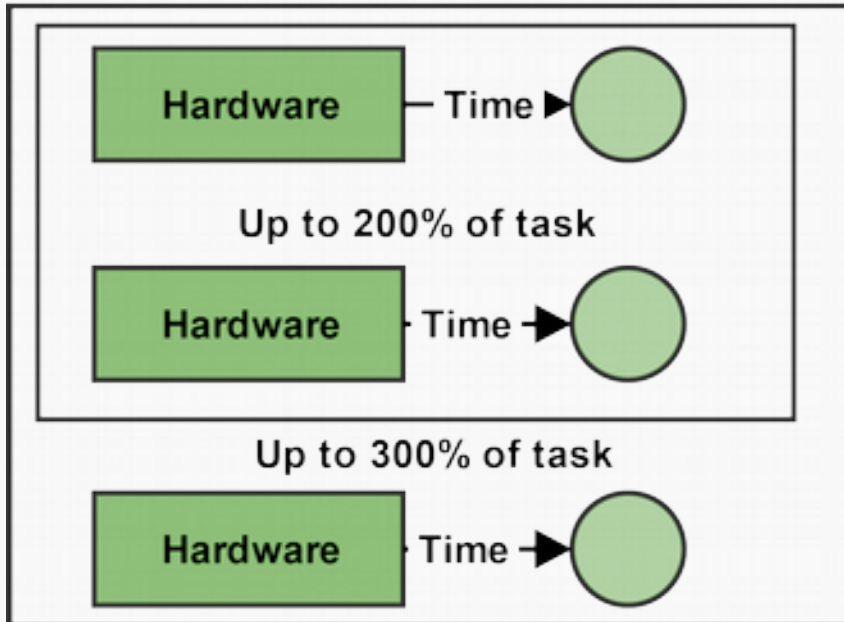
100% of task

Hardware

100% of task

Hardware

Time/2



Масштабируемость ТИПОВЫХ ПРИЛОЖЕНИЙ

<i>Категория приложений</i>	<i>Scaleup</i>	<i>Speedup</i>
Интернет-приложения, OLTP (Online Transaction Processing)	Да	Нет
Системы принятия решений с распараллеливаемой обработкой запросов	Да	Да
Комплексные системы («всё сразу»)	Да	Зависит от архитектуры

Пример: хватает ли аппаратных ресурсов для построения кластера?

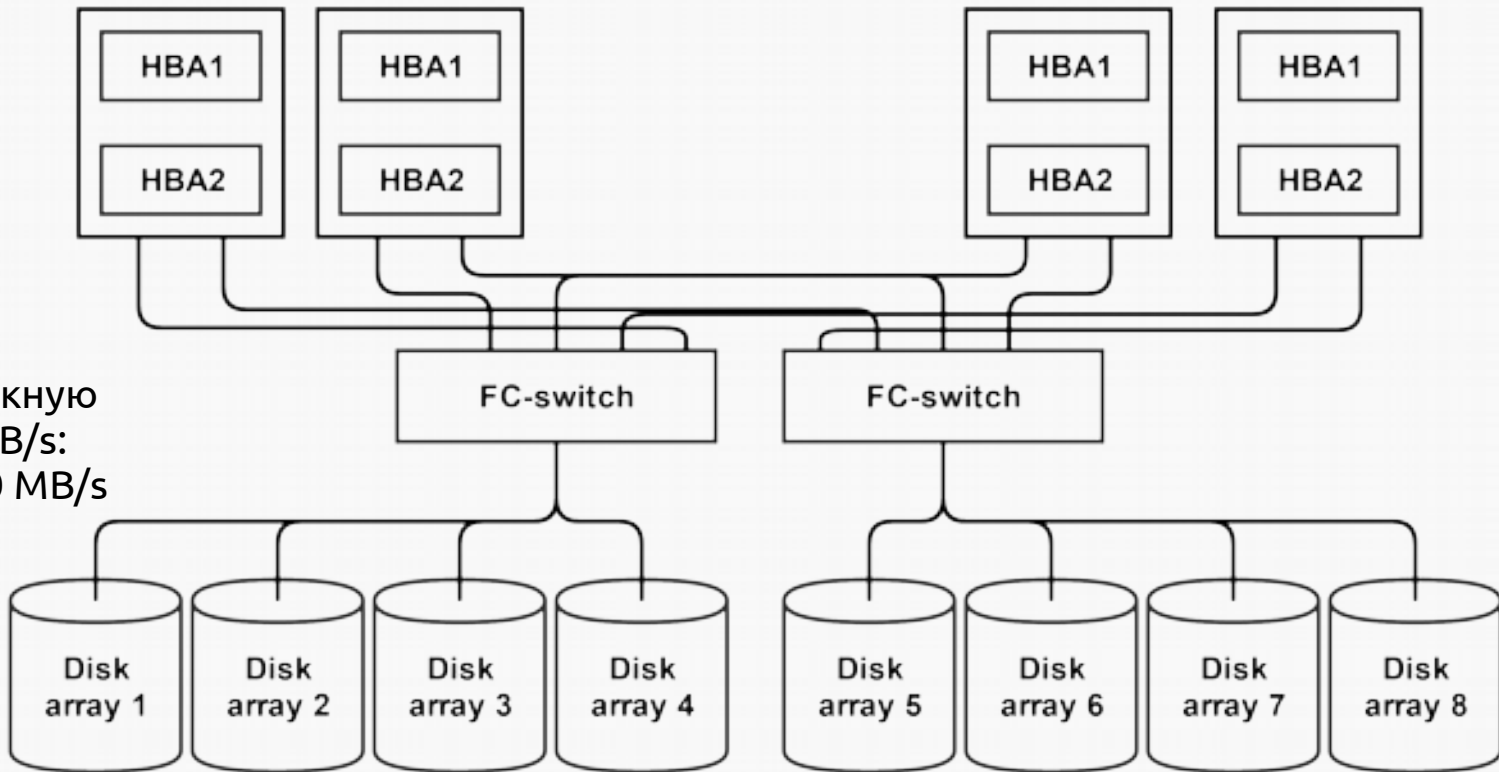
4 машины

В каждой машине по 2 CPU: $2 \times 200 \text{ MB/s} \times 4 = 1600 \text{ MB/s}$

В каждой машине по 2 HBA: $8 \times 200 \text{ MB/s} = 1600 \text{ MB/s}$

2 коммутатора

Каждый должен обеспечить пропускную способность 800 MB/s:
 $2 \times 800 \text{ MB/s} = 1600 \text{ MB/s}$

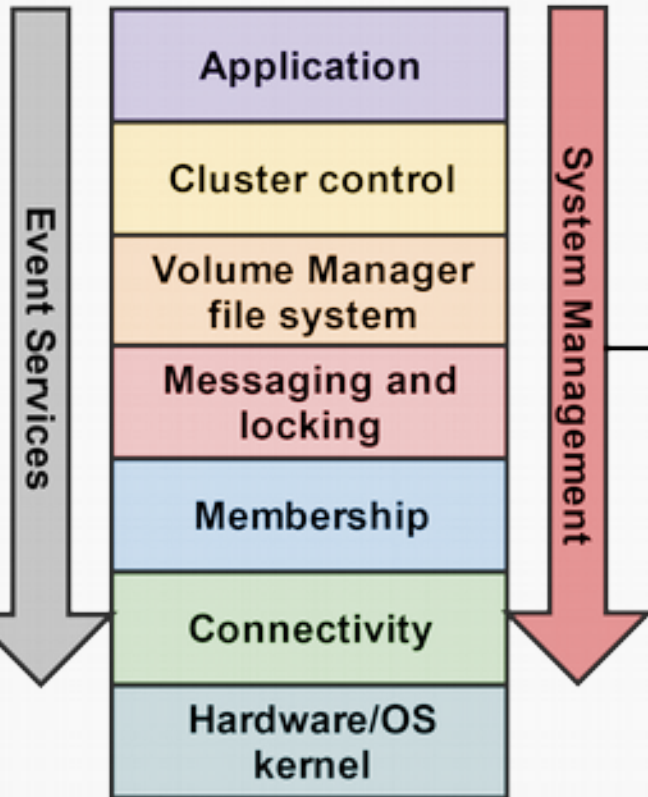


8 дисковых массивов

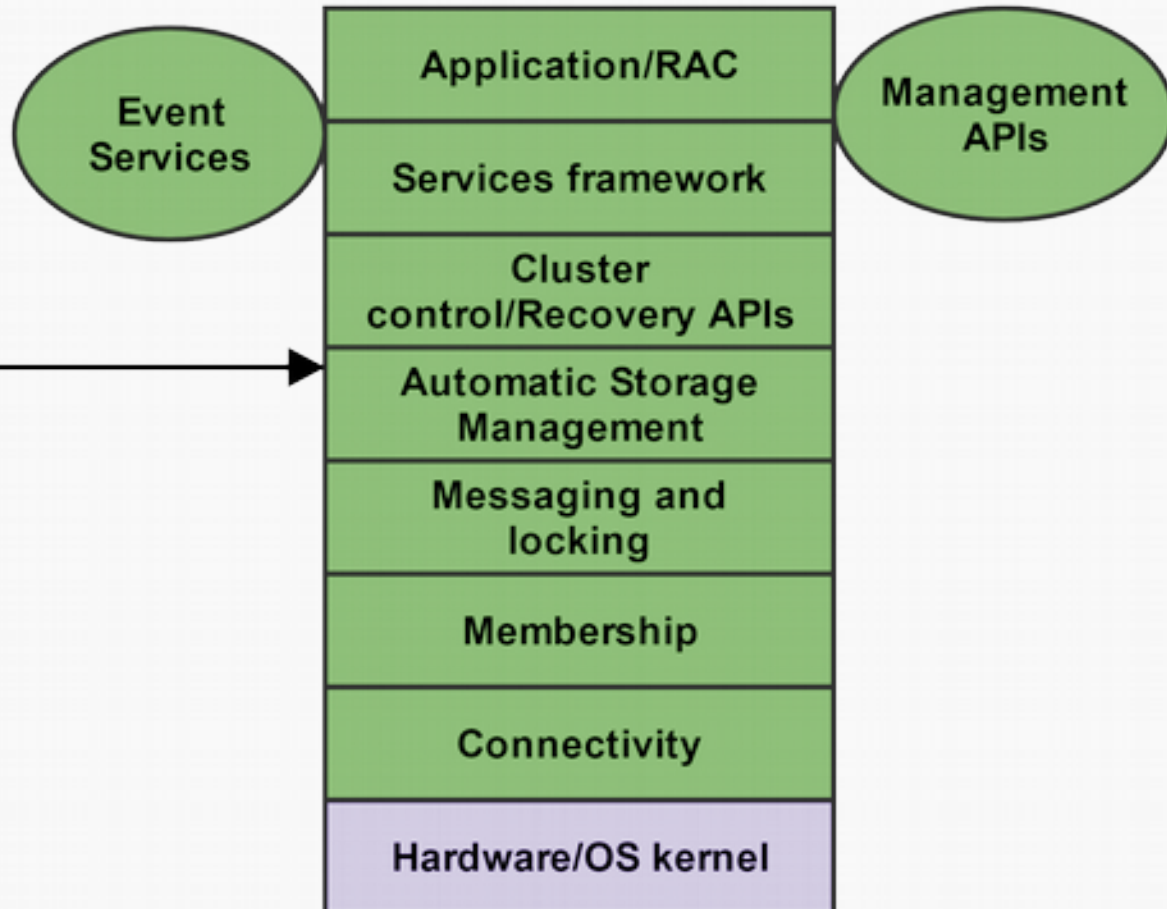
Каждый использует один 2 Гбитный контроллер: $8 \times 200 \text{ MB/s} = 1600 \text{ MB/s}$

Архитектура Oracle RAC

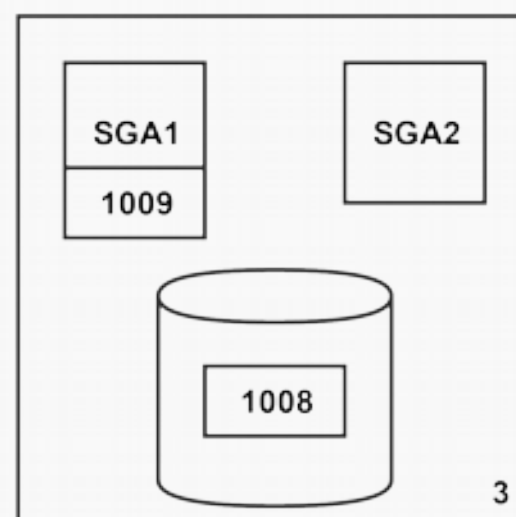
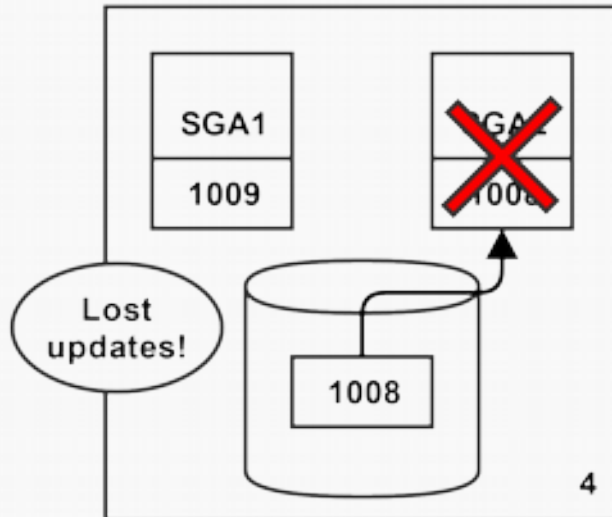
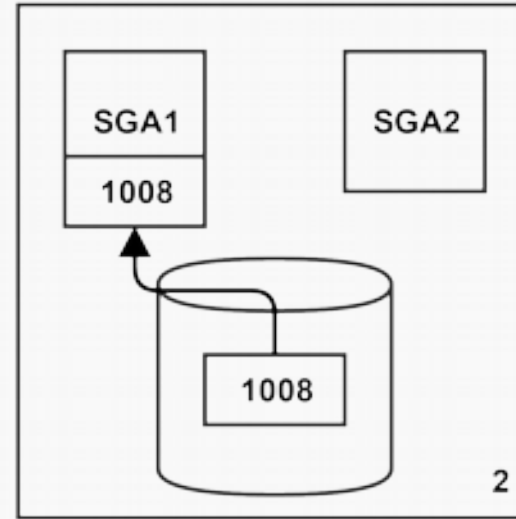
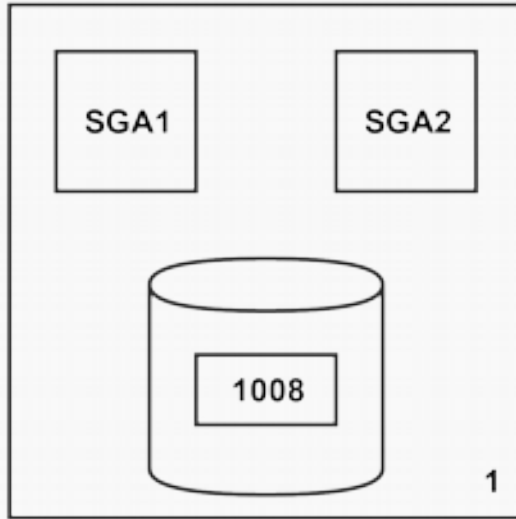
9i RAC



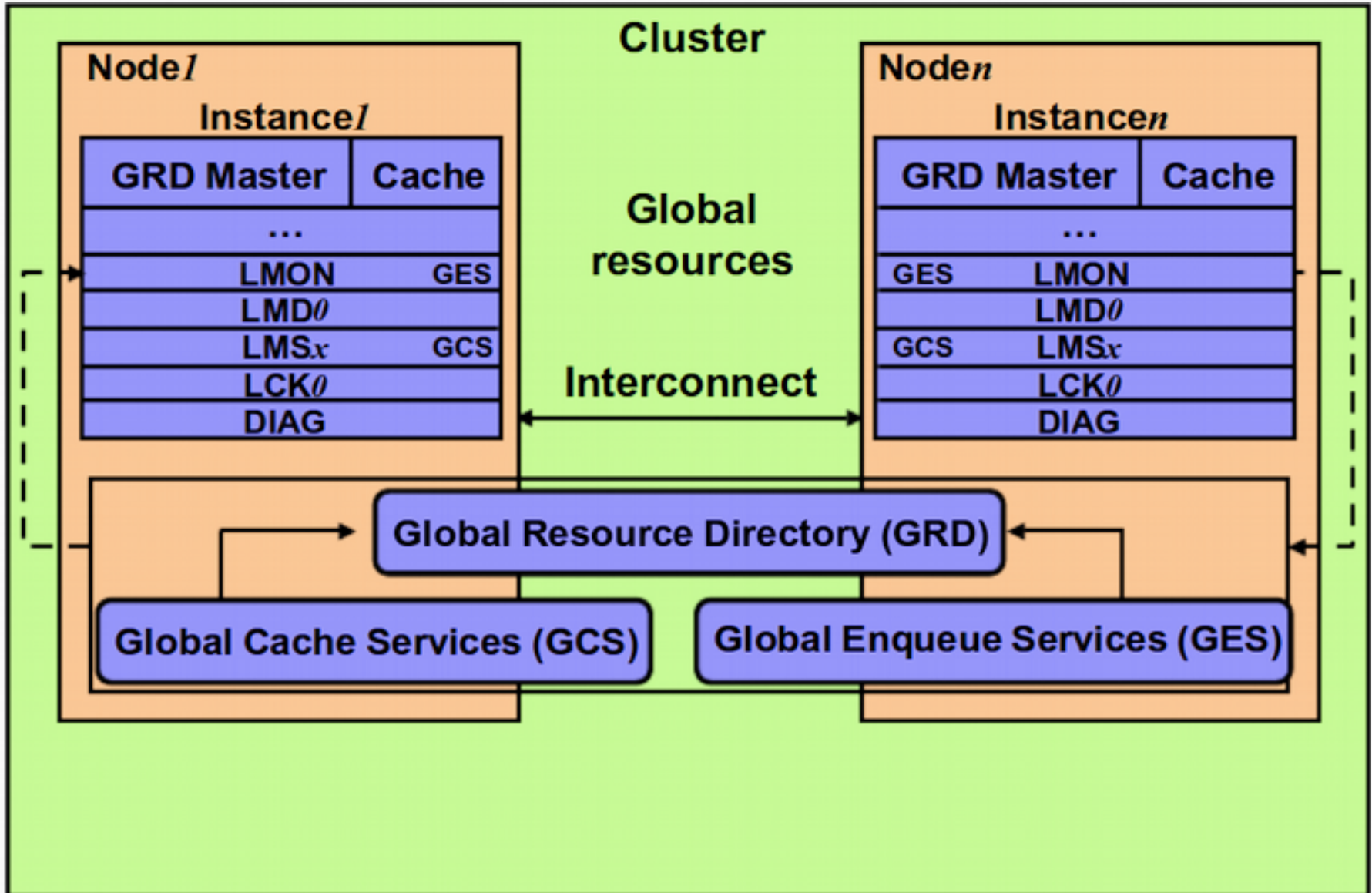
11g Oracle Clusterware



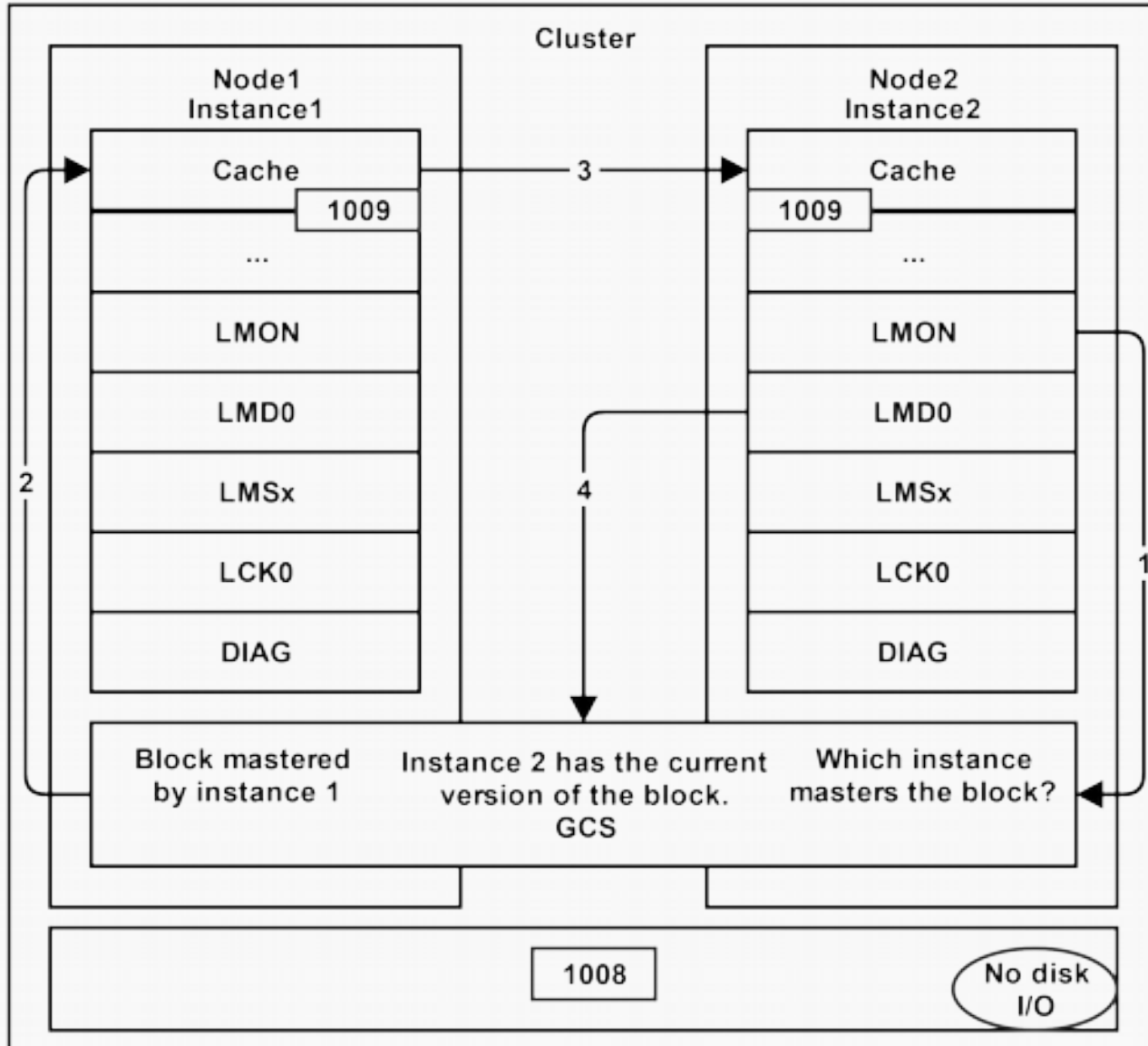
Пример: зачем нужны «глобальные» ресурсы?



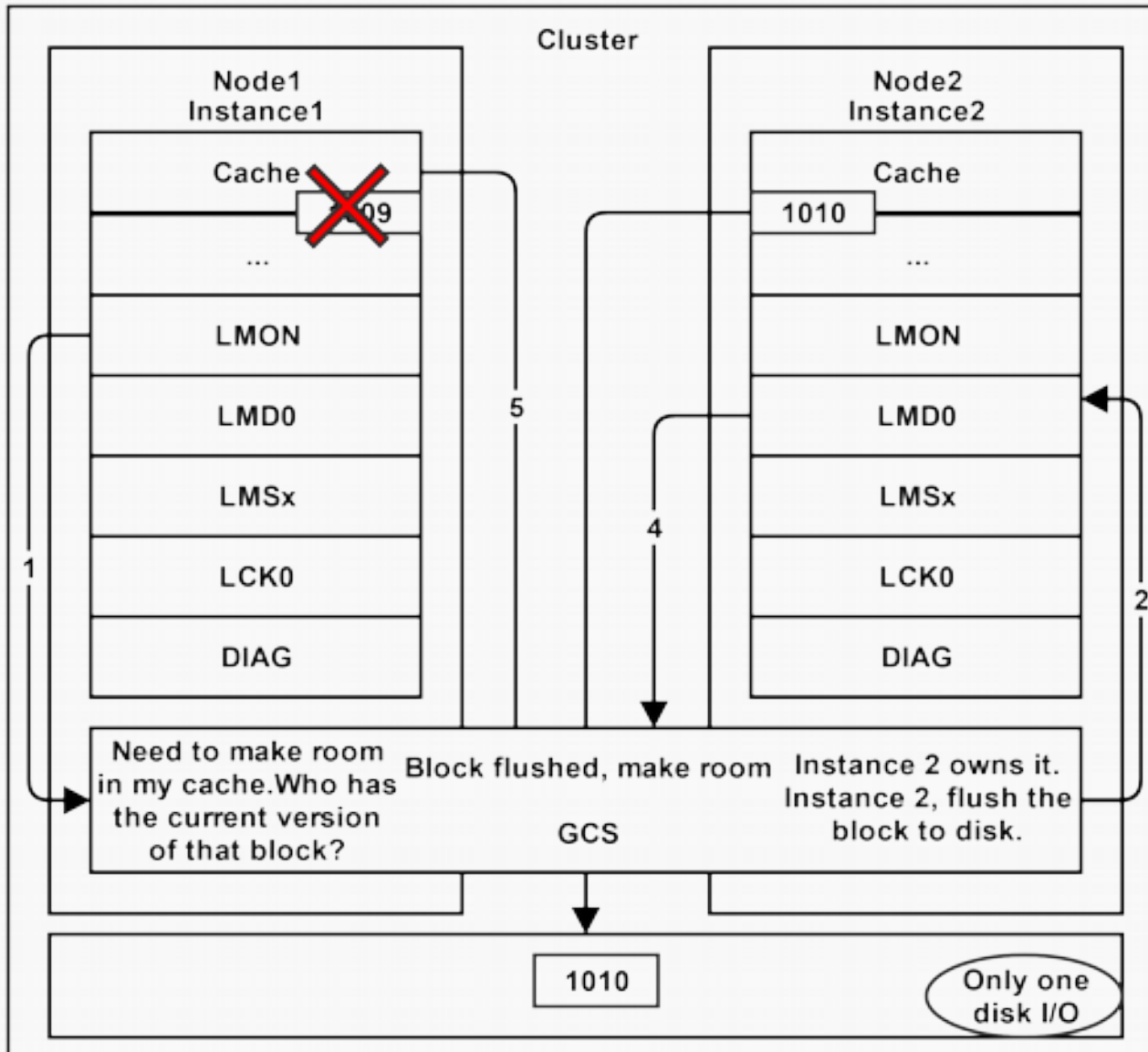
Управление глобальными ресурсами



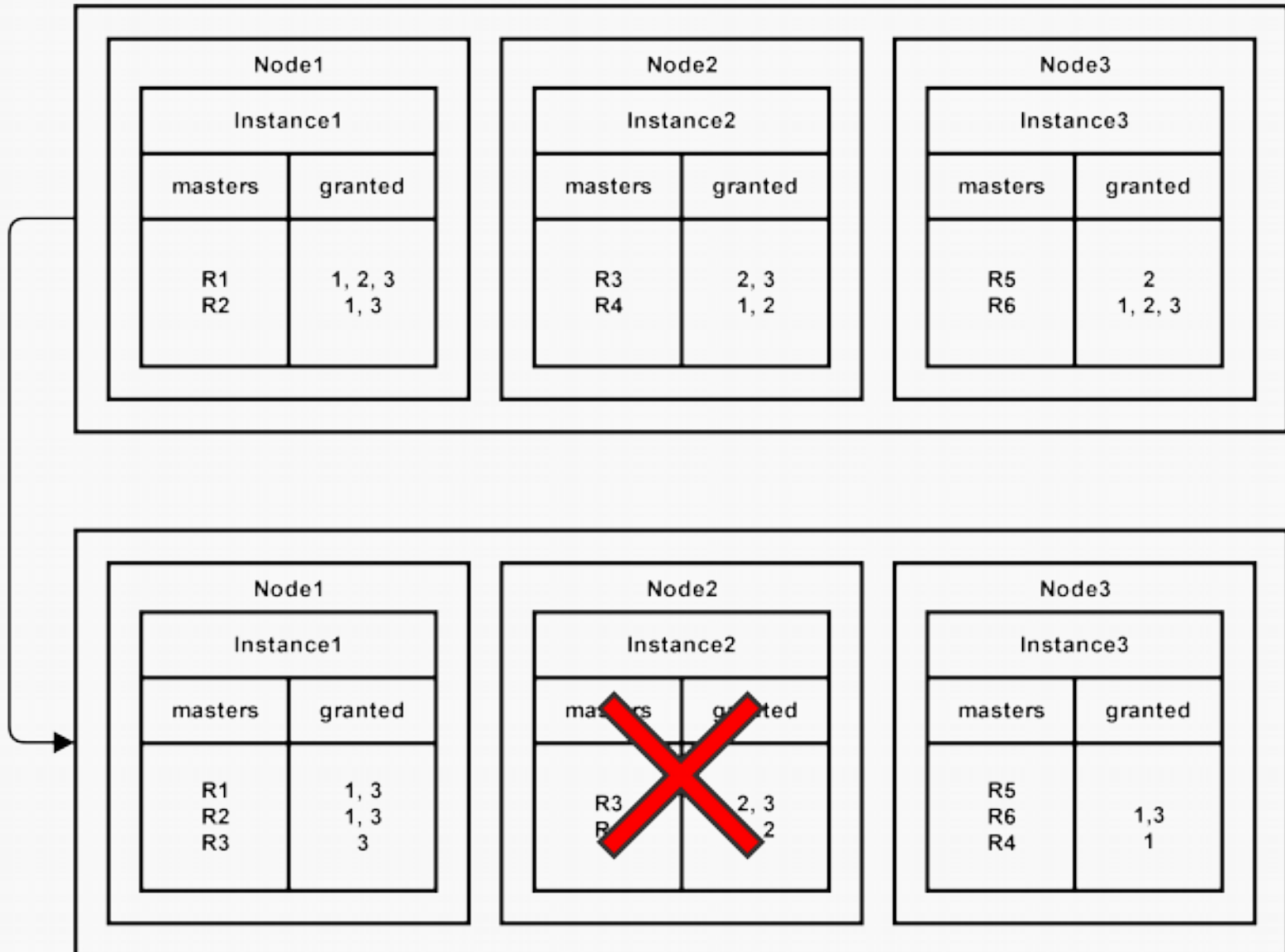
Пример: синхронизация глобального кэша



Пример: КООРДИНАЦИЯ ЗАПИСИ НА ДИСК

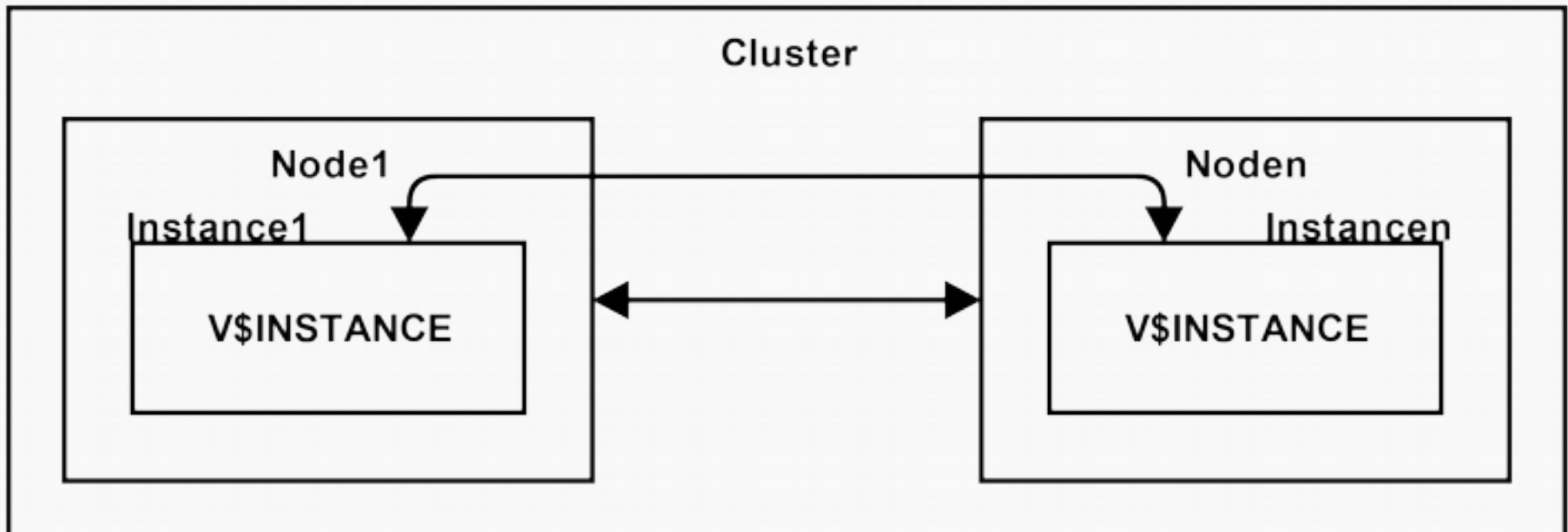


Динамическая реконфигурация кластера



Глобальные DPV

- Содержат информацию обо всех запущенных экземплярах в составе кластера.
- У каждого локального представления (V\$) есть соответствующее ему глобальное представление (GV\$).
- Исполняются параллельно на всех узлах кластера — «ведущий» запрос на узле, к которому осуществляется обращение и «ведомые» запросы к V\$ на остальных узлах.
- Параллелизмом управляет специальный сервис — *координатор параллельного исполнения* (Parallel Execution Coordinator, PEC).



Дополнительные требования к памяти

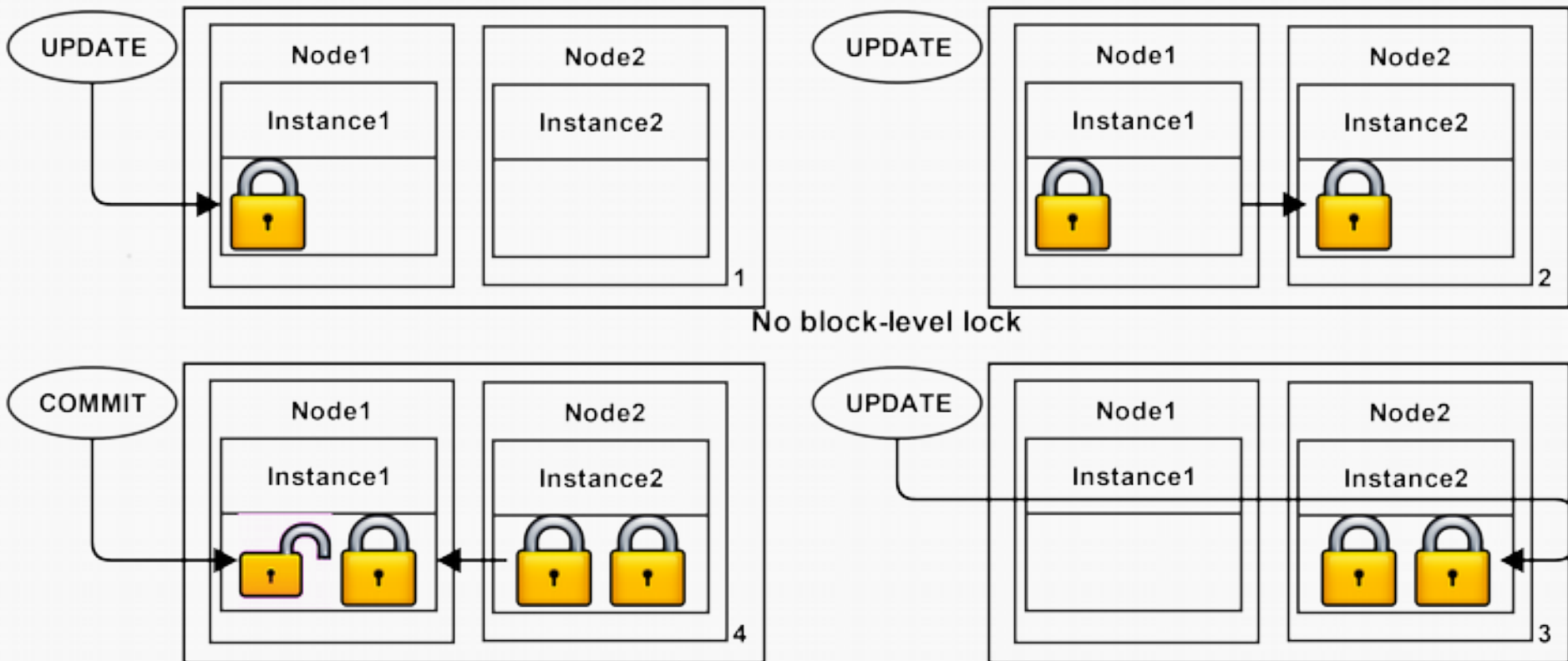
- Oracle RAC потребляет больше памяти по сравнению с «обычным» экземпляром БД:
 - На 15% больше для разделяемого пула.
 - На 10% больше для буферного кэша.
- Тем не менее, размер буферного кэша на каждом конкретном узле уменьшается, т. к. не все данные пользовательской сессии хранятся в нём.
- Команды для проверки текущего потребления памяти:

```
SELECT resource_name,  
       current_utilization, max_utilization  
FROM   v$resource_limit  
WHERE  resource_name like 'g_s%';
```

```
SELECT * FROM v$sgastat  
WHERE name like 'g_s%' or name like 'KCL%';
```

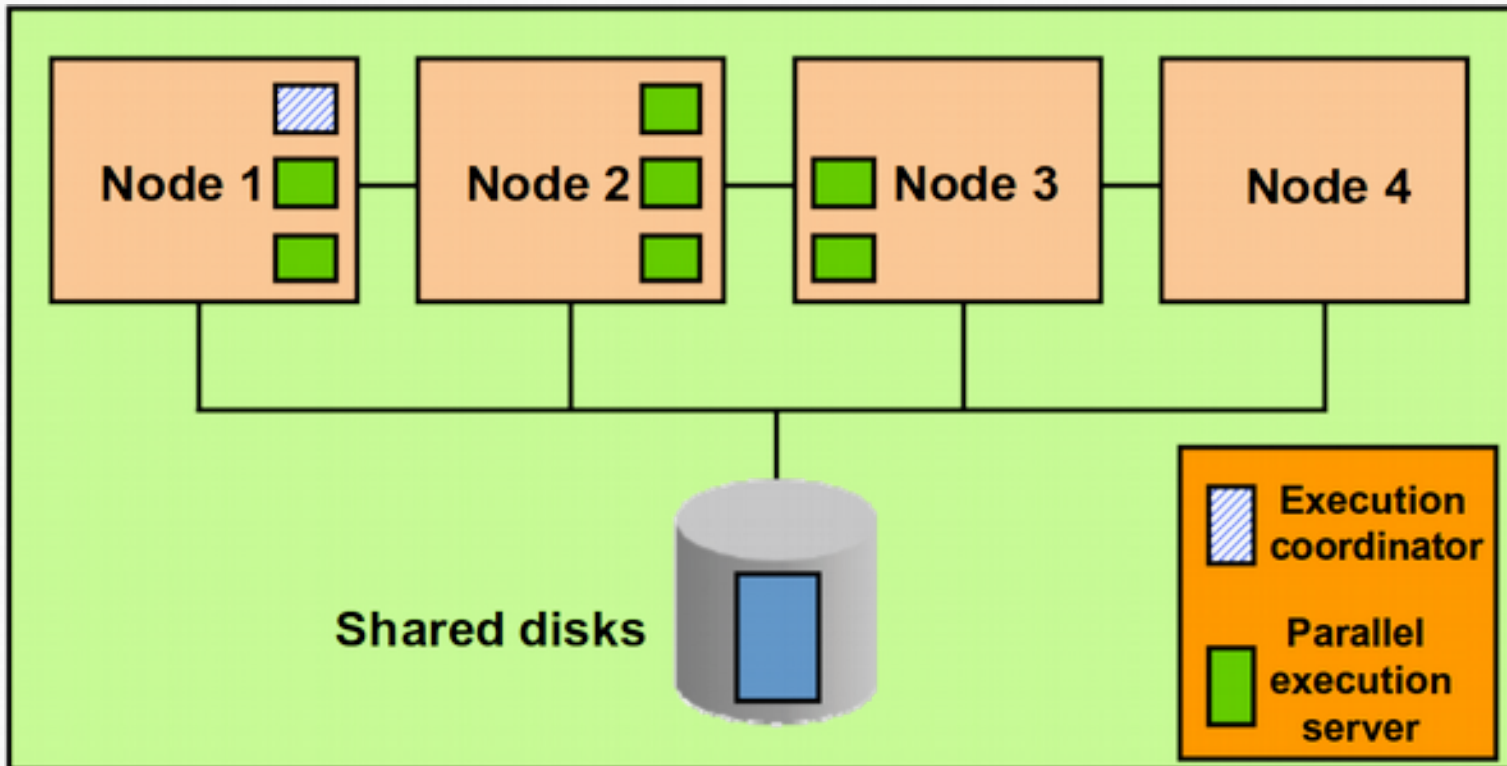
Блокировки на уровне строк

Если транзакции на разных узлах затрагивают непересекающиеся множества строк, они могут осуществляться параллельно:

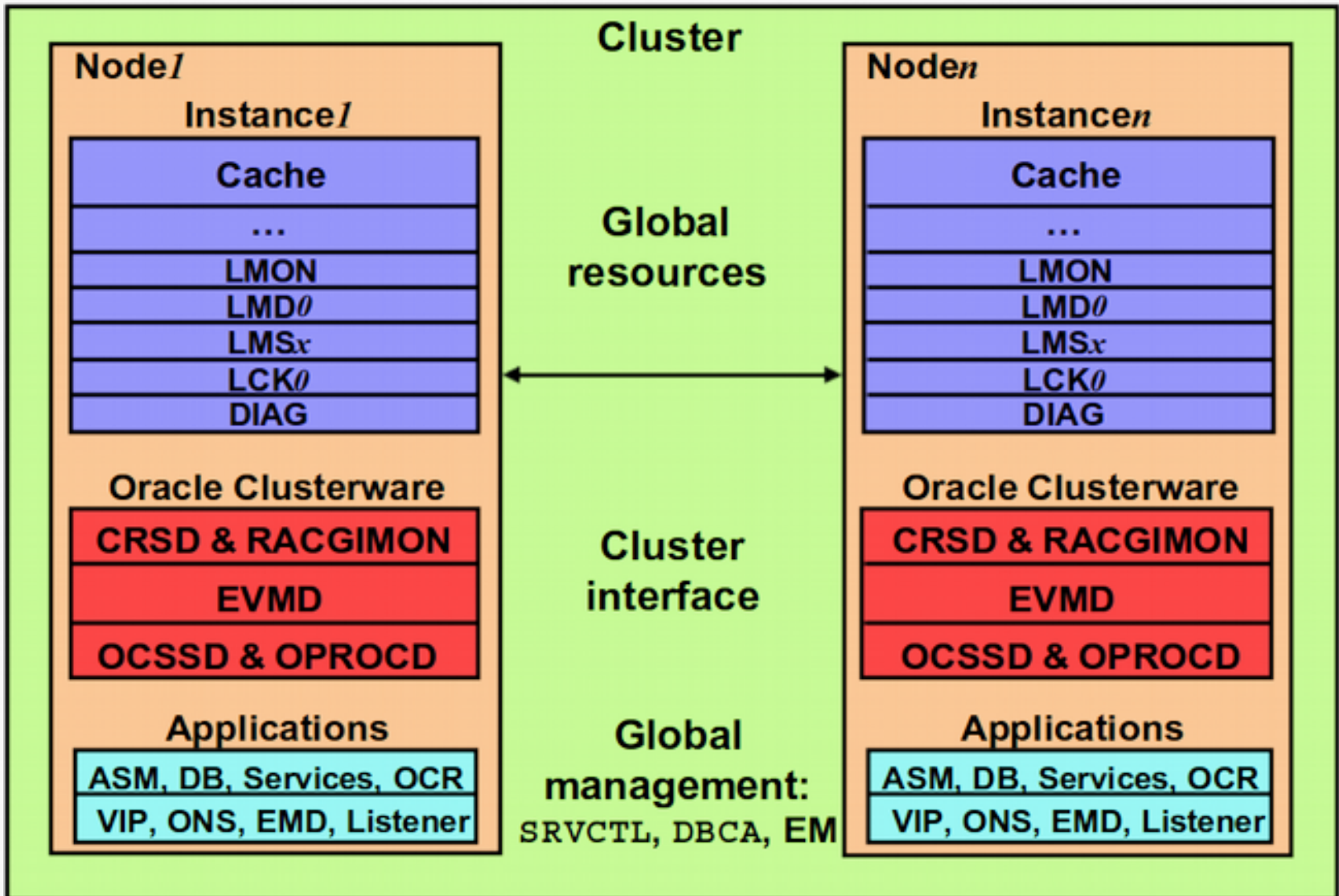


Параллельное исполнение запросов

- РЕС использует интеллектуальный алгоритм параллелизма выполнения запросов.
- Запросы распараллеливаются как по процессам в составе одного экземпляра, так и по разным экземплярам в составе кластера.

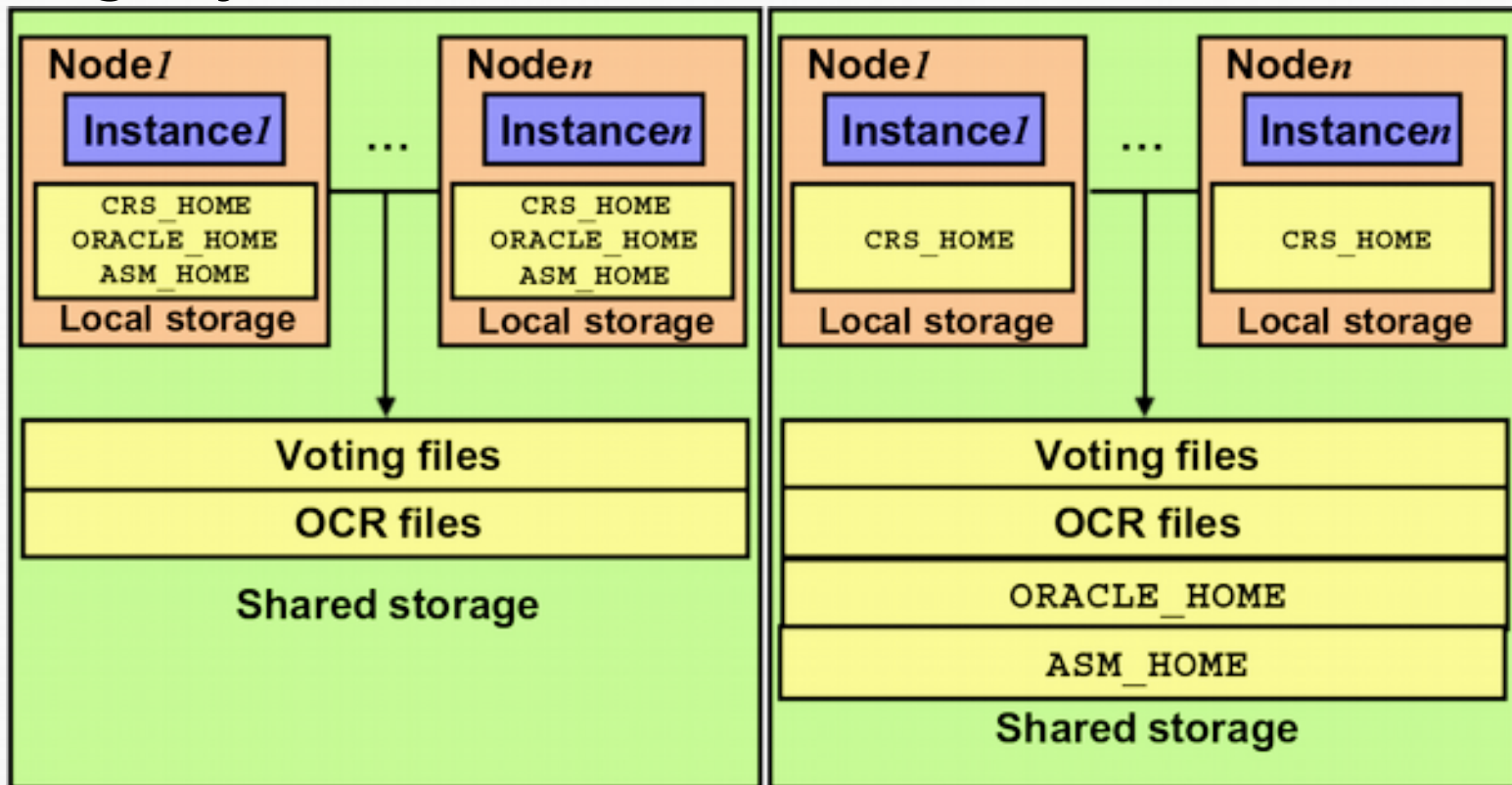


Архитектура RAC: процессы

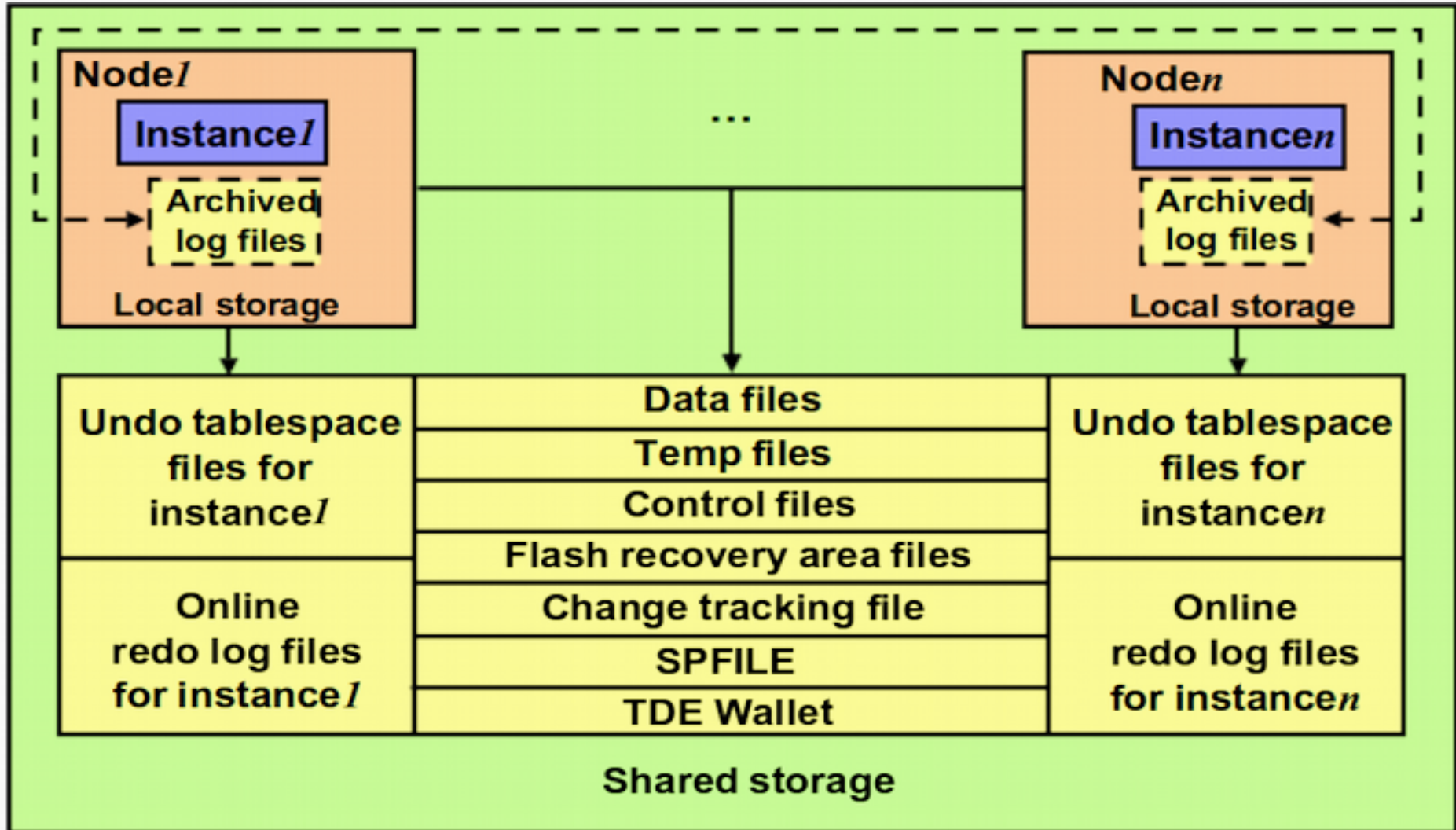


Архитектура RAC: конфигурационные файлы

- Появляются 2 новых категории файлов
 - Файлы с результатами мониторинга состояния кластера.
 - Файлы OCR и OLR — Oracle Cluster Registry и Oracle Local Registry.



Архитектура RAC: файлы БД





Oracle Cluster File System (OCFS)

Файловая система, специально разработанная для хранения разделяемых ресурсов RAC.

Особенности:

- Позволяет всем узлам кластера использовать общий ORACLE_HOME.
- Каждый том OCFS может размещаться на одном или нескольких физических дисках.

В файловом хранилище на базе OCFS можно разместить:

- Установленные бинарные файлы Oracle.
- Файлы экземпляра Oracle (конфигурационные, файлы данных и т. д.).
- Файлы параметров инициализации (spfile).
- Временные файлы, созданные экземпляром Oracle в время работы.
- Voting & OCR files.

CFS vs Raw

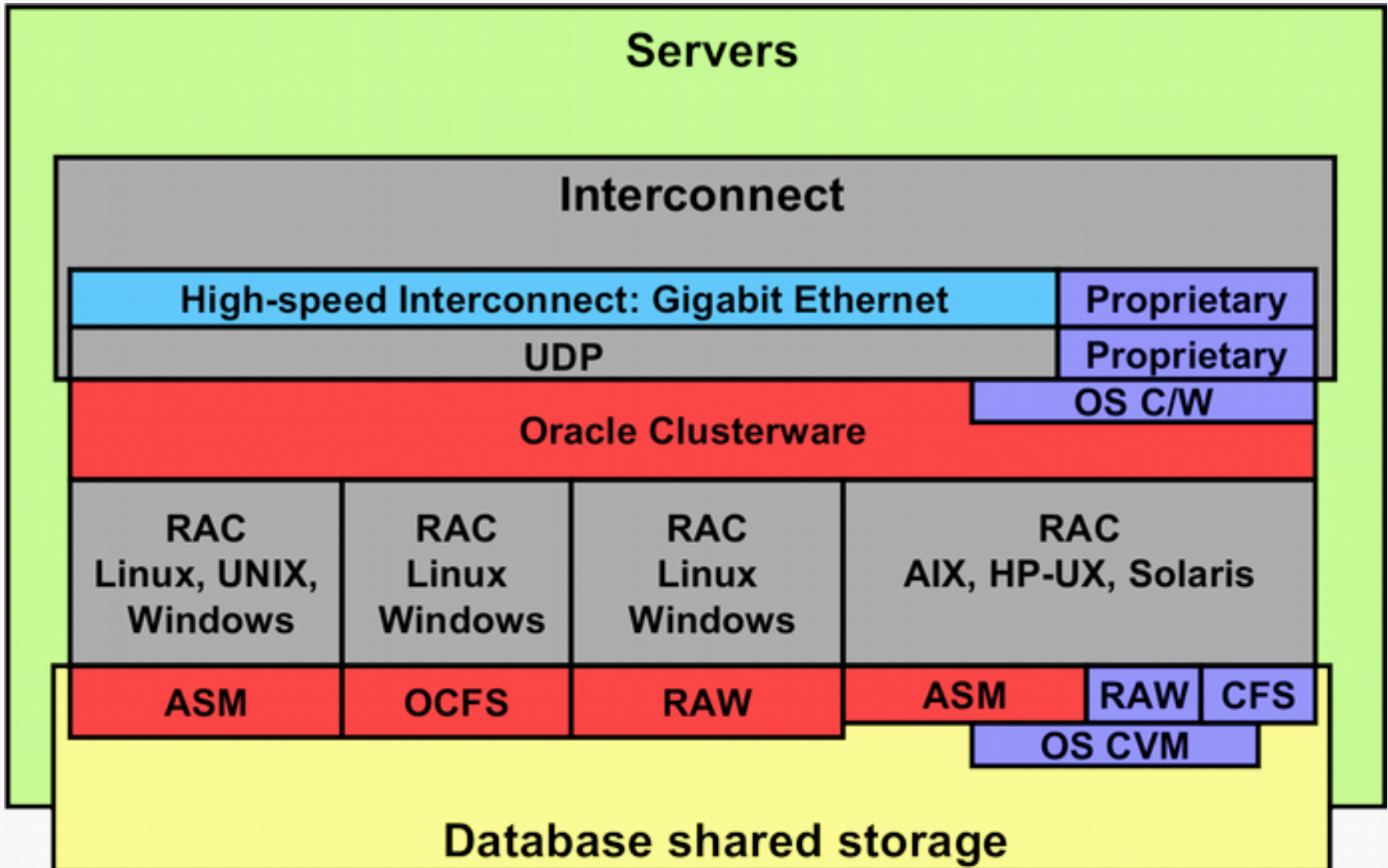
Преимущества CFS:

- Проще администрировать.
- Ставится вместе с Oracle, не нужна дополнительная конфигурация.
- Автоматически расширяется по мере возрастания объёма данных.
- Можно использовать для хранения файлов архива журнала повторов.

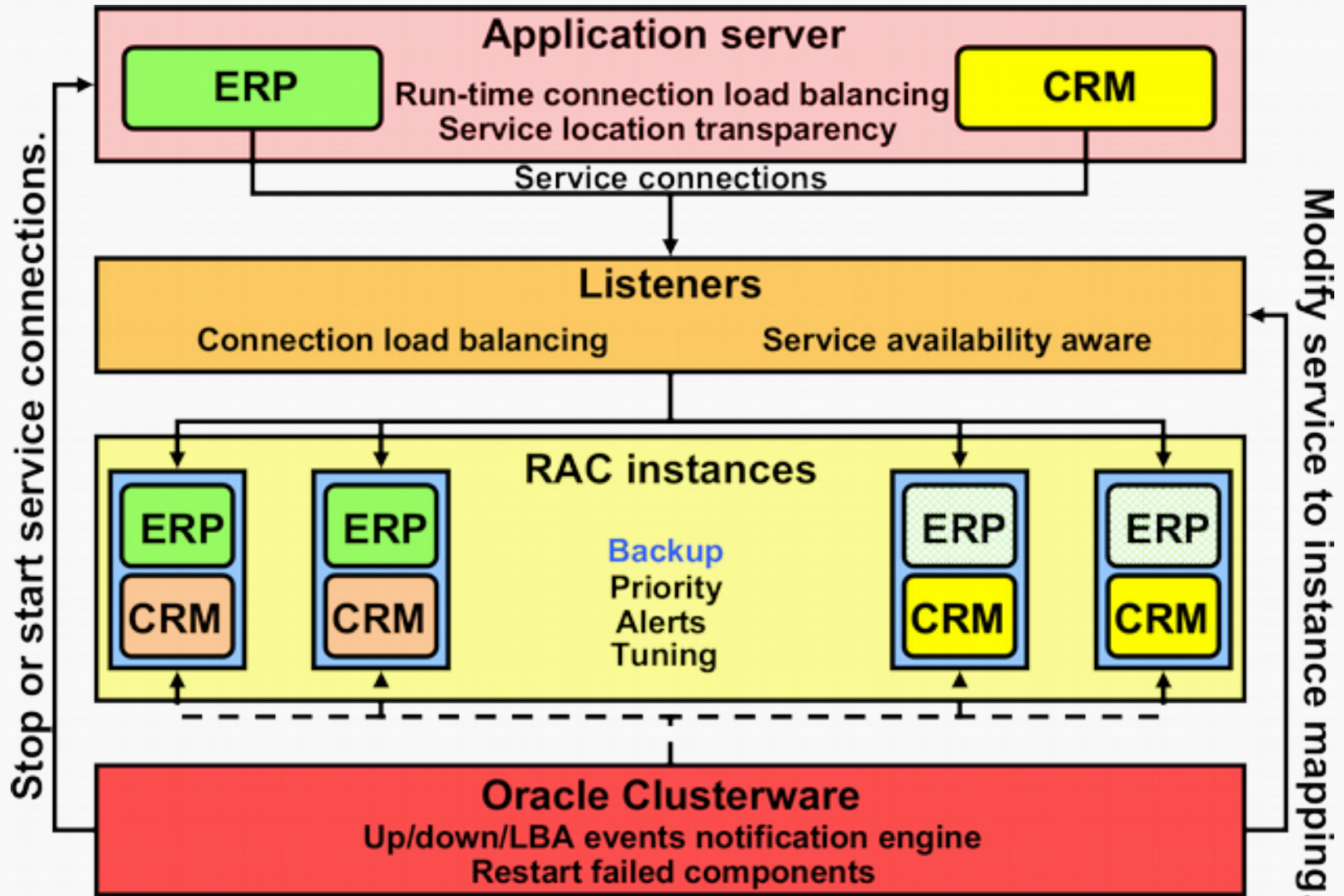
Преимущества raw:

- Потенциально быстрее.
- Ниже требования к инфраструктуре.
- Можно использовать ASM для расширения возможностей.

Пример: кластер на базе Oracle RAC



Сервисы RAC

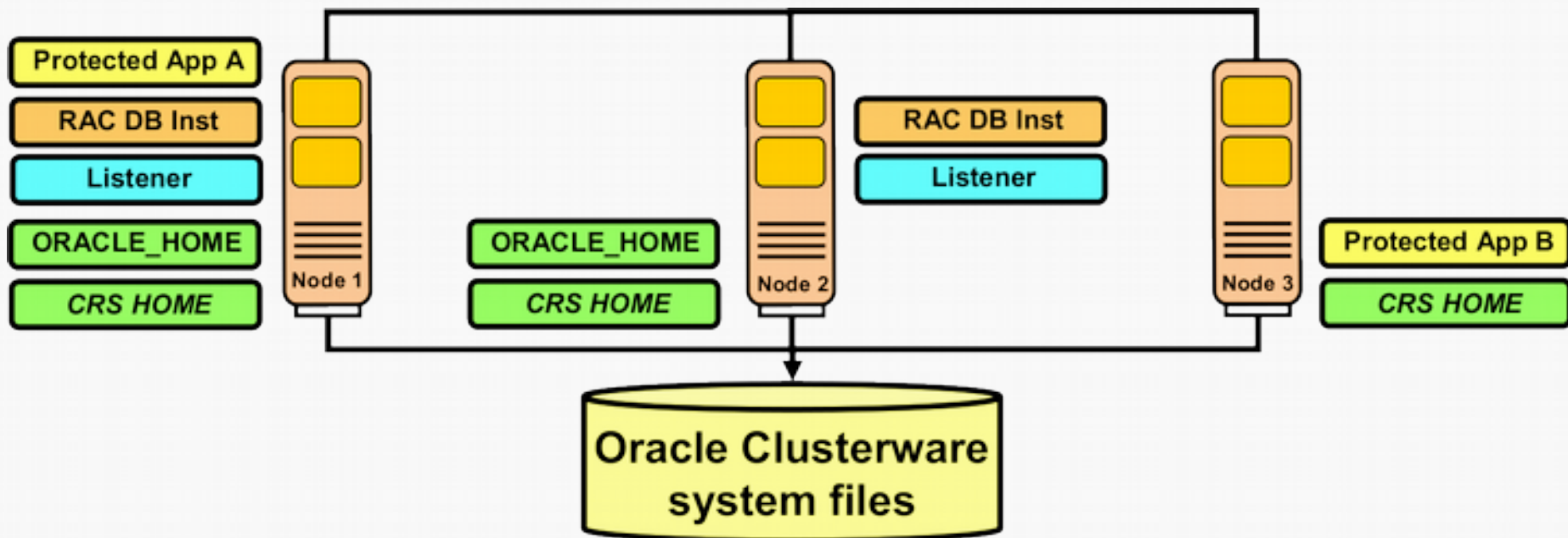


Oracle Clusterware

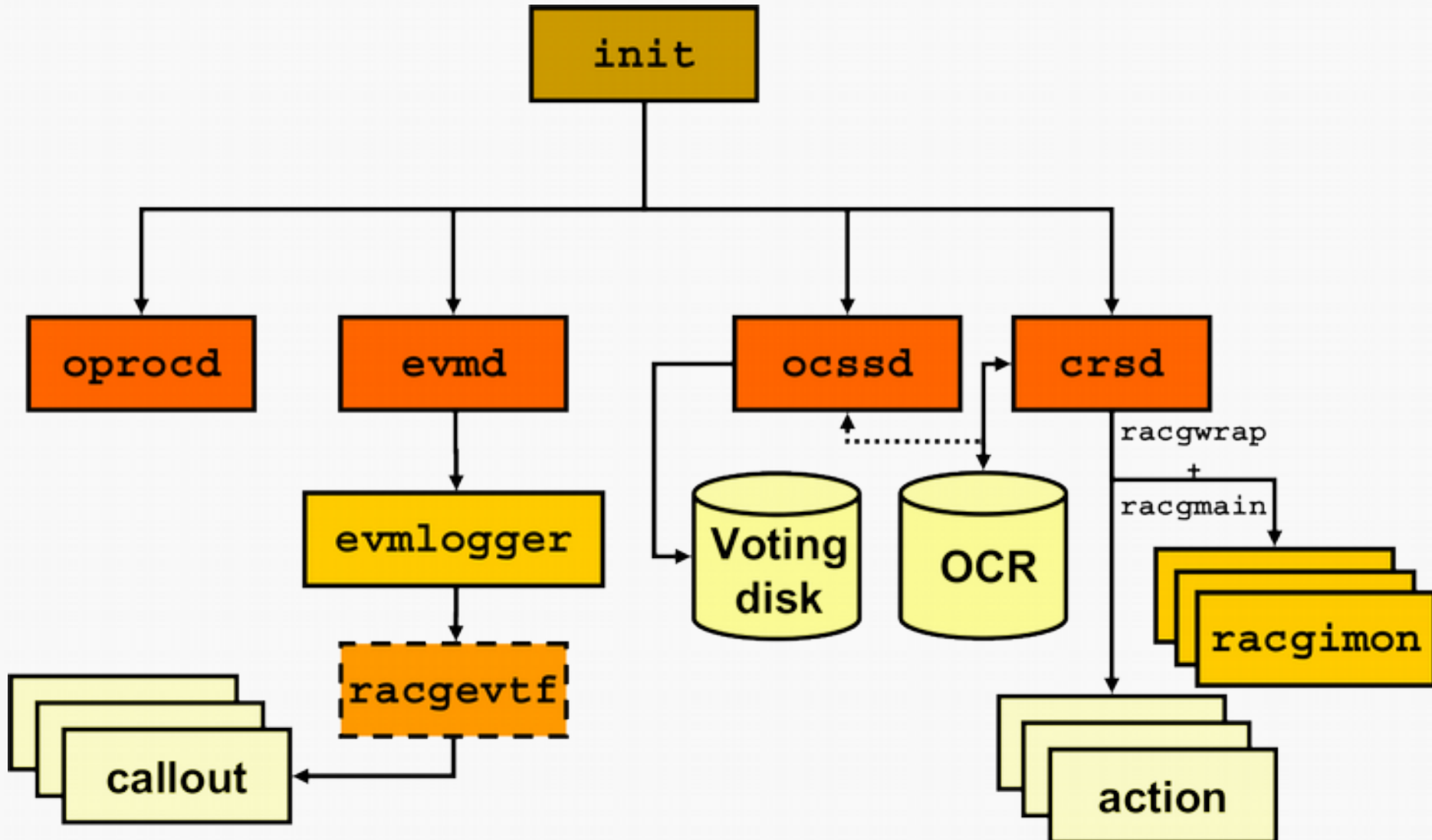
Инфраструктурное ПО, которое добавляет поддержку High Availability в кластерную БД.

Решает следующие задачи:

- Мониторинг состояния приложений.
- Автоматический перезапуск приложений в случае их «падения».
- Failover приложений в случае «падения» узла кластера.



Oracle Clusterware Runtime



Процессы Oracle Clusterware

Под *nix все процессы «прописываются» в `/etc/inittab`, под Windows — как сервисы.

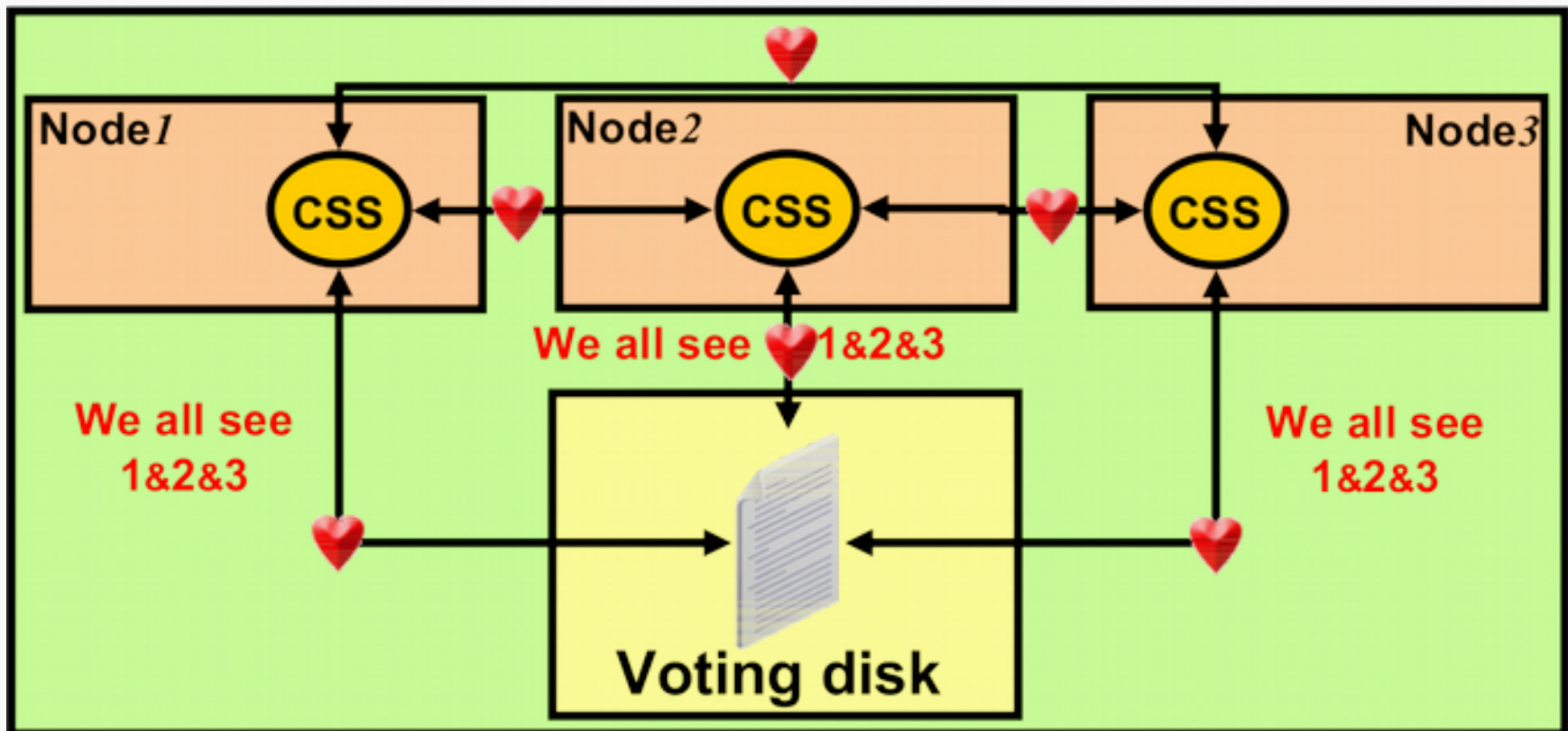
Oracle Clusterware включает в себя следующие процессы:

- **Cluster Synchronization Services Daemon (OCSSD)** — осуществляет мониторинг состояния кластера и сервисов ASM. Запускается из-под пользователя Oracle; «падение» этого процесса приводит к перезапуску всей машины во избежание повреждения и / или рассинхронизации состояния БД.
- **Process Monitor Daemon (OPROCD)** — запускается на каждом узле из-под суперпользователя. В случае обнаружения каких либо проблем на аппаратном уровне или на уровне драйверов устройств он аварийно останавливает узел кластера.
- **Cluster Ready Services Daemon (CRSD)** — «инфраструктурный» процесс для всех сервисов HA. Помимо всего прочего, управляет состоянием ресурсов кластера в OCR.
- **Event Management Daemon (EVMD)** — управляет передачей информацией о событиях жизненного цикла кластера.

Voting Disk

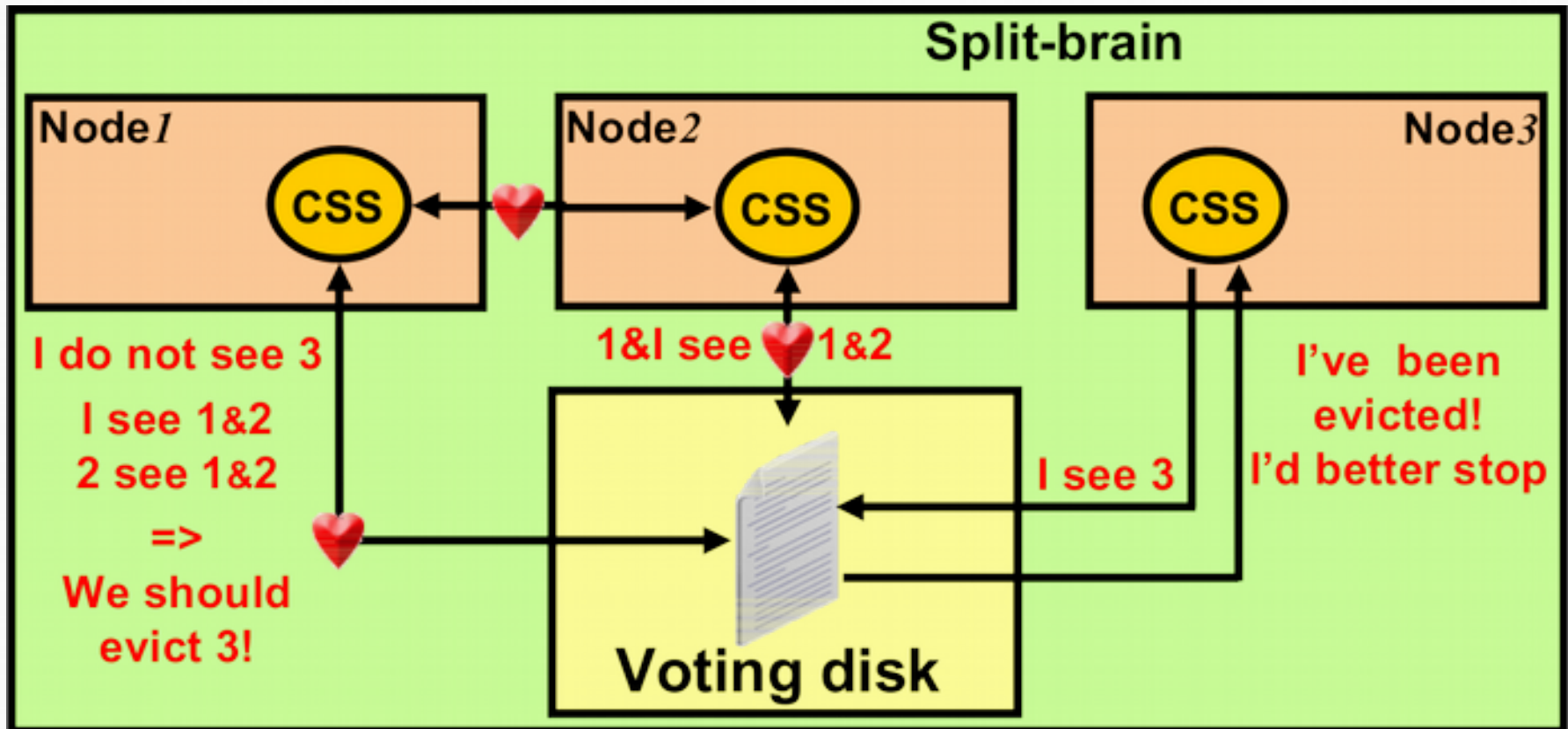
- Нужен для мониторинга состояния узлов кластера.
- Может быть реализован в виде файла (Voting Disk File).

Все узлы кластера «видят» друг друга:



Voting Disk (продолжение)

Узел Node3 «потерял» связь с другими узлами, нагрузка перераспределяется:



Oracle Cluster Registry (OCR)

- *Реестр кластера* хранит информацию о конфигурации Oracle Clusterware и кластерной БД (список узлов, расположение узлов, параметры установки ПО и т.д.).
- Хранится в файлах OCR:

```
[oracle@racnode1 ~]$ ls -l /u02/oradata/racdb
total 16608
-rw-r--r-- 1 oracle oinstall 10240000 Aug 26 22:43
CSSFile
drwxr-xr-x 2 oracle oinstall      3896 Aug 26 23:45 dbs/
-rw-r----- 1 root    oinstall  6836224 Sep  3 23:47
OCRFile
```

- Файлы OCR могут храниться в локальной файловой системе, а также под управлением ASM или CFS.

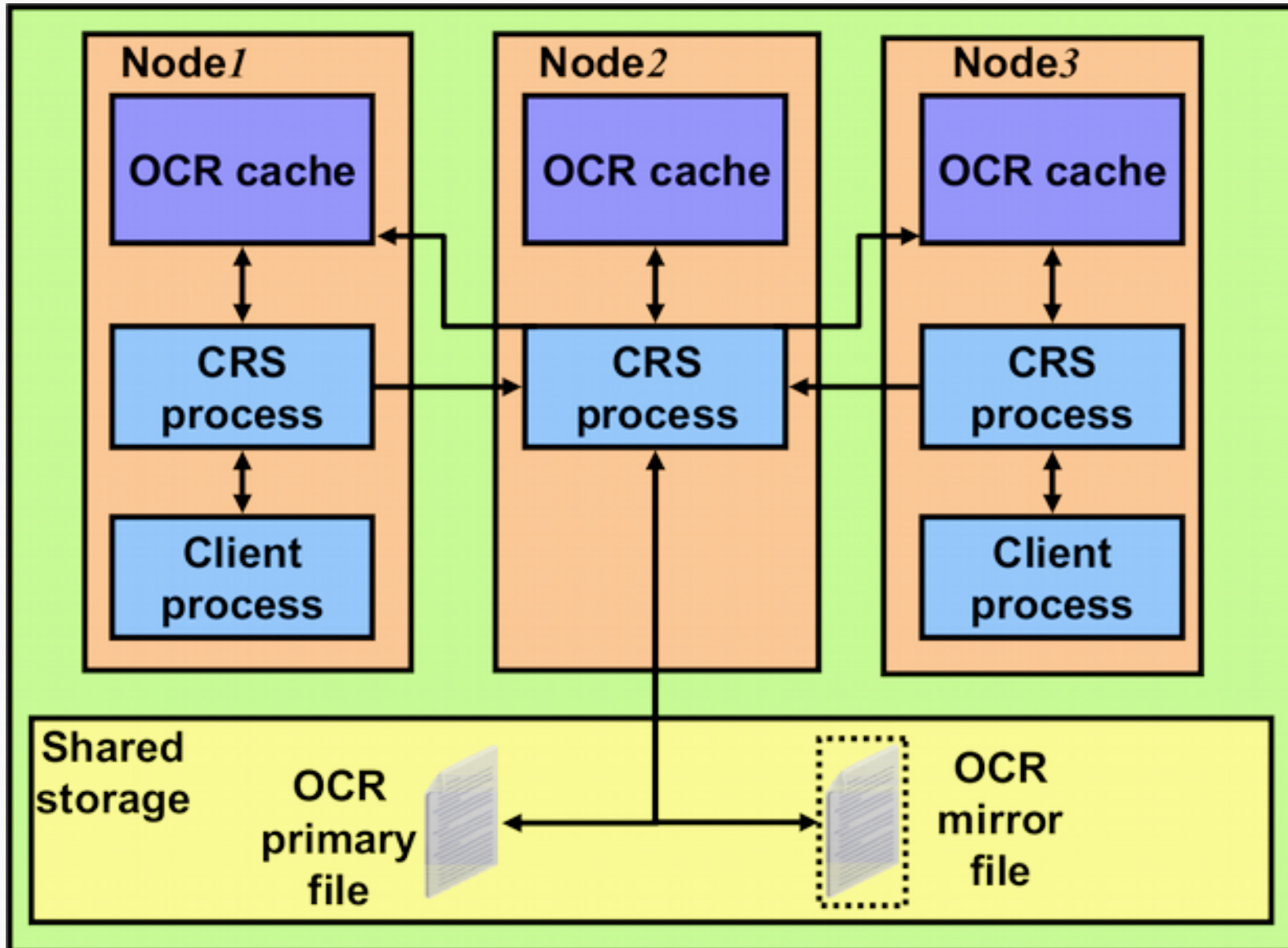
Oracle Local Registry (OLR)

- Копия OCR — Oracle Local Registry (OLR) располагается на каждом узле кластера.
- OLR управляет конфигурацией Oracle Clusterware на каждом конкретном узле.
- Файл OLR хранится в локальной файловой системе конкретного узла:

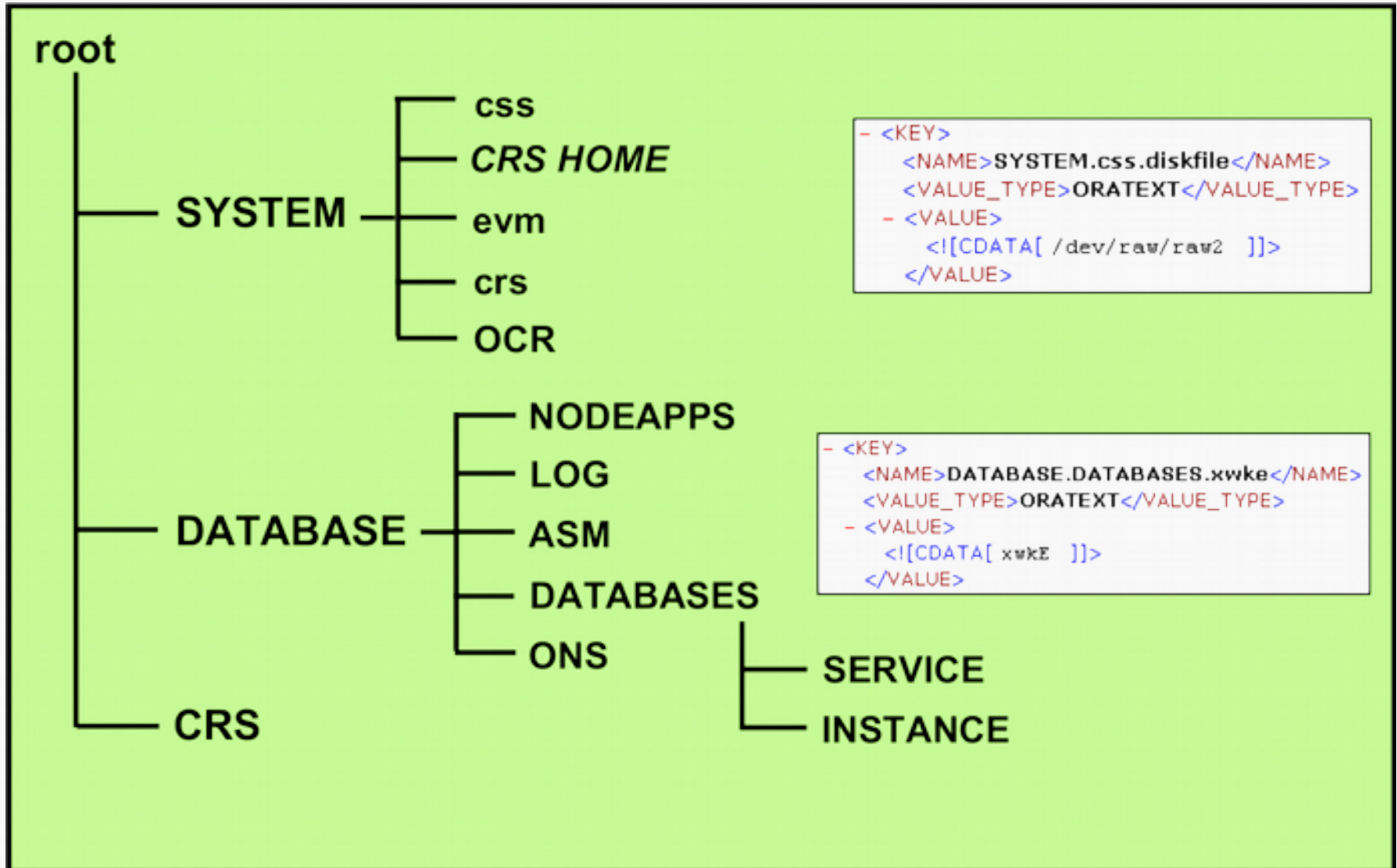
```
$ pwd
/etc/oracle
$ ls -lrt olr*
-rw-r--r--  1 root dba 96 Feb 24 07:34 olr.loc
```

- Удаление или повреждение OLR приводит к невозможности использования данного узла в составе кластера.

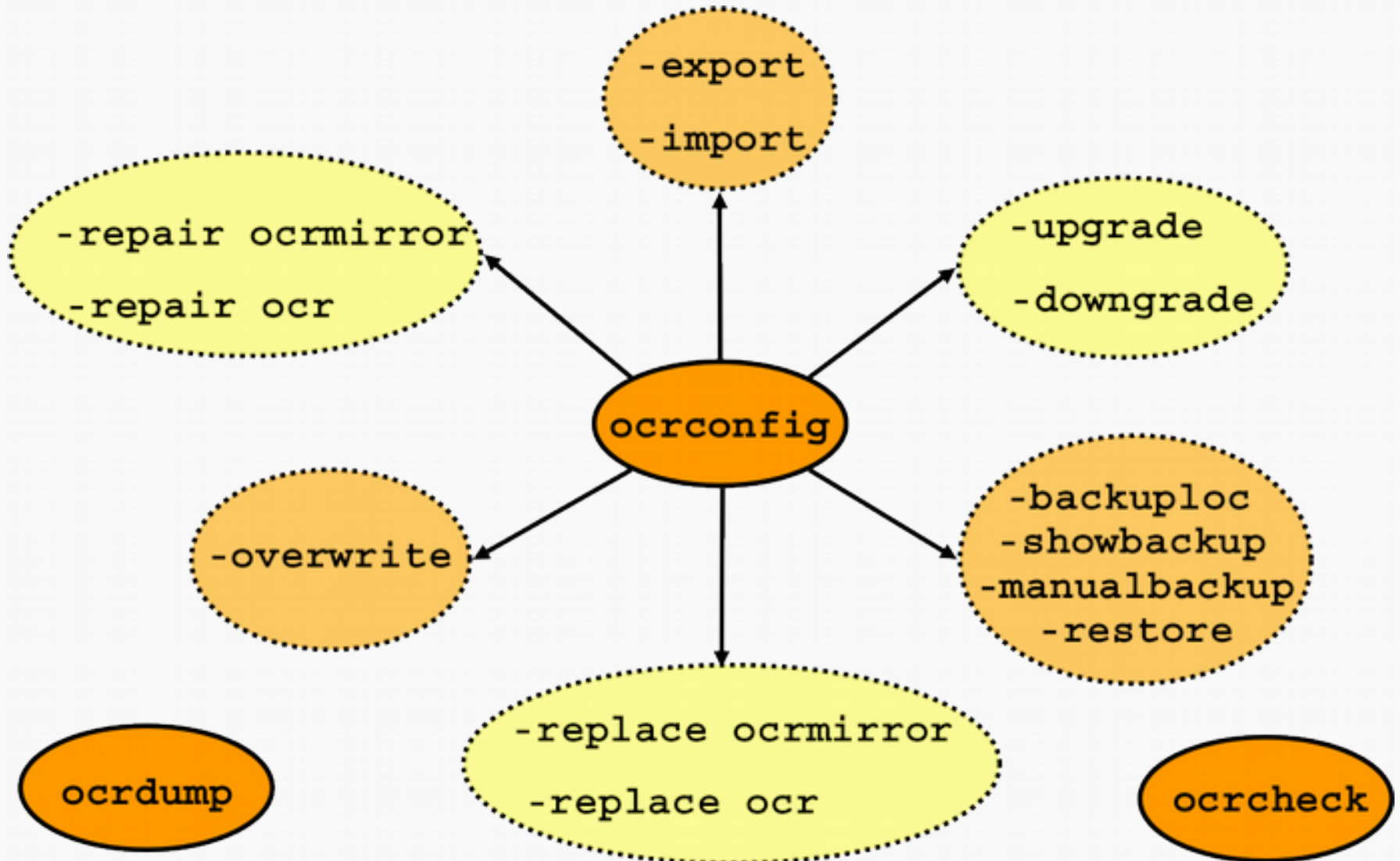
Архитектура OCR



Структура реестра OCR



Управление ресурсами OCR



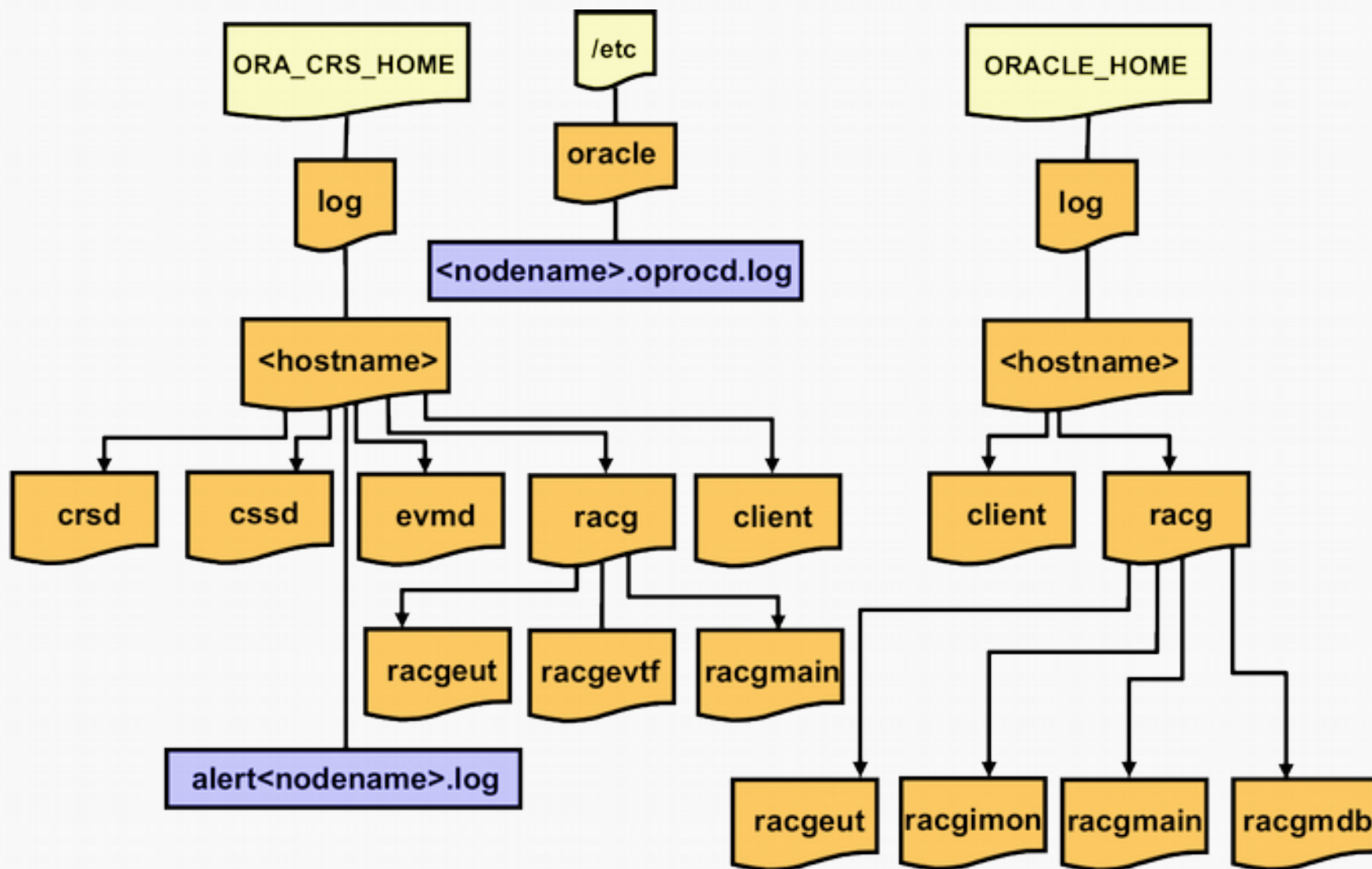
Управление ресурсами OCR (продолжение)

- Утилиты ocrconfig и ocrcheck:

```
# ocrconfig -add +new_disk_group
# ocrconfig -delete /dev/raw/raw2
# ocrconfig -delete /dev/raw/raw1
[oracle@racnode1 ~]$ ocrcheck
Status of Oracle Cluster Registry is as follows :
    Version                :                2
    Total space (kbytes)    :           262120
    Used space (kbytes)     :             4660
    Available space (kbytes) :           257460
    ID                      :           1331197
    Device/File Name        : /u02/oradata/racdb/OCRFile
                           Device/File integrity check succeeded
                           Device/File not configured
    Cluster registry integrity check succeeded
```

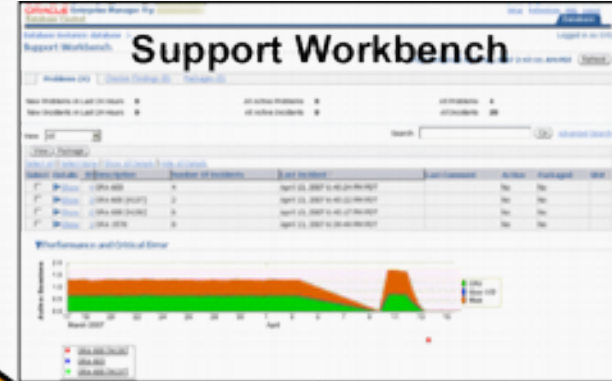
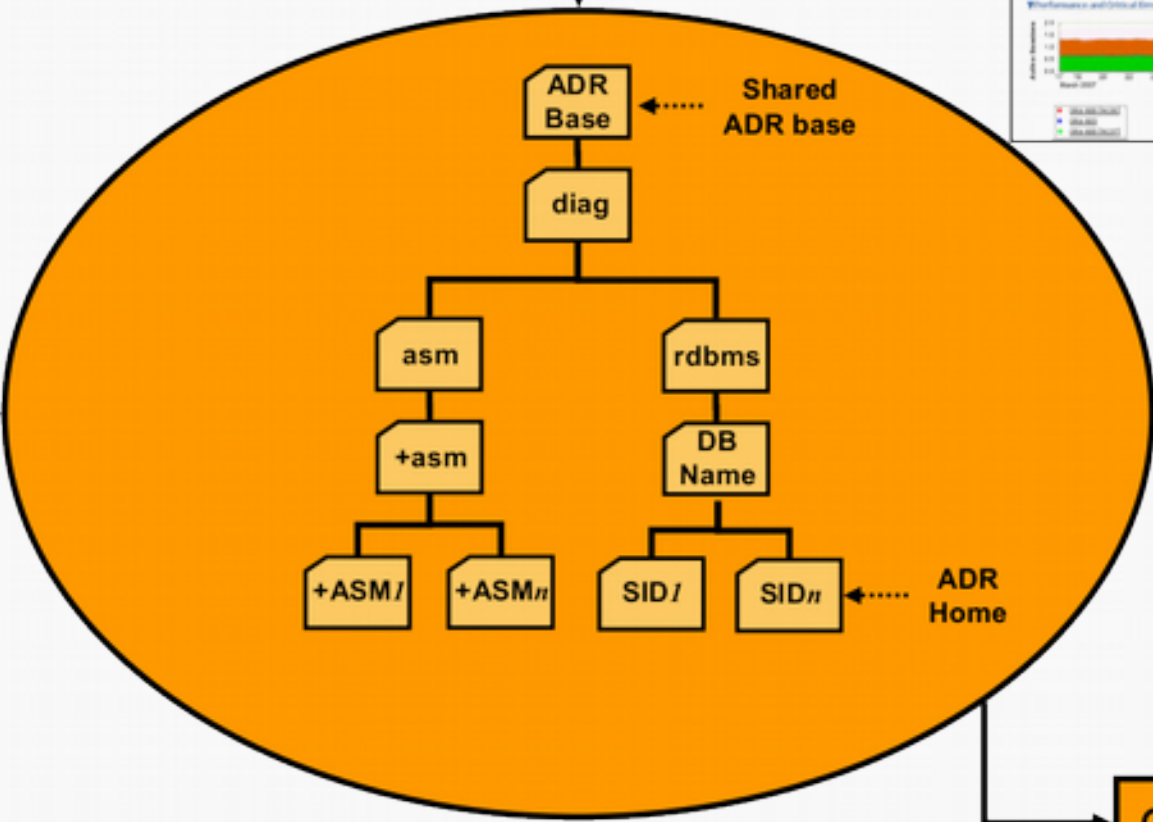
- Если файл OCR хранится под управлением ASM, то рекомендуется создать его резервную копию на диске из другой дисковой группы.
- Oracle не поддерживает возможность параллельного хранения файлов OCR в разных типах хранилищ (например, ASM и NFS), но возможна миграция файлов OCR между разными типами хранилищ.
- Если ASM «падает» на каком-либо узле, OCR становится недоступным для этого узла.

Файлы логов Oracle Clusterware



Диагностика состояния кластера

DIAGNOSTIC_DEST



ADRCI

GV\$DIAG_INFO

«Пошаговая» верификация конфигурации кластера

