

Пример построения автоматизированного управления дисками (ASM)

Владимир Пржиялковский

Преподаватель технологий Oracle

prz@yandex.ru

<http://www.ccas.ru/prz>

Декабрь 2007 г.

Есть вещи, о которых солдатам надлежит ведать, и есть вещи которых им лучше не знать.

Отон у Тацита в «Истории»

Аннотация

Automatic Storage Management (ASM) – средство автоматического управления дисковым пространством БД, появившееся в версии 10 и развитое в версии 11. Два главных свойства ASM – контролируемая избыточность данных и автоматическая балансировка загрузки дисков. В статье рассматривается доступный пример построения ASM.

Введение

До версии 10 в Oracle имелось два способа отображения данных БД (табличных пространств, журнальных файлов, контрольного файла) на диск: в файлы ОС и на неформатированные файловой системой разделы. В версии 10 появился третий способ, в определенном смысле промежуточный между первыми двумя, называемый *автоматическим управлением дисковой памятью*, Automatic Storage Management (ASM). Он предполагает, что из неформатированных разделов диска формируются *дисковые группы*, внутри которых формируется своего рода облегченный специализированный вариант файловой системы для нужд БД. Управление «файлами» внутри дисковых групп берет на себя облегченный специализированный вариант экземпляра СУБД (*экземпляр ASM*). Отличительными свойствами такого управления дисковым пространством являются, помимо прочего:

- контролируемая избыточность во имя отказоустойчивости;
- автоматическое перераспределение нагрузки на дисковые устройства.

Файлы, размещением которых можно управлять таким образом, могут быть:

- файлами БД,
- файлами резервирования и восстановления,
- *SPFILE.ORA*
- файлы программ *expdp/impdp*.

Может показаться, что ASM развивает собой, или даже заменяет введенную в версии 9 возможность OMF. Это не так: оба средства самостоятельны в отношениях друг с другом, и ASM может при желании использоваться совместно с OMF. В то же время ASM дает администратору БД намного больше.

Фирма Oracle прочит ASM в перспективу, и в версии 11, помимо усовершенствований (хотя без радикальных новшеств), добавила в документацию по СУБД отдельную книгу, *Storage Administrator's Guide*. Основную ценность ASM фирма, судя по заявлениям, видит в применении этого механизма к большим установкам, в первую очередь кластерным (RAC), использующим «сотни» дисковых устройств. Не исключено, что в будущем пользователям Oracle придется иметь дело с ASM как с основным механизмом. Поэтому, несмотря на то, что (а)

в сегодняшней практике ASM редкость, и что (б) будучи новой, эта техника, как водится, требует выдержки временем¹, разумно присматриваться к ней уже сейчас.

Ниже рассматривается пример организации ASM, доступный для воспроизведения на обычной, не кластерной платформе. Не потребуется даже иметь дело с неформатированными разделами диска, так как в данном случае допускается их имитация обычными файлами ОС. Для определенности, пример относится к Windows. Многие необходимые подправки для Unix очевидны. Более полно материал для Unix превосходно изложен в статье http://www.idevelopment.info/data/Oracle/DBA_tips/Automatic_Storage_Management/ASM_10.shtml (далее [1]), из которой я много позаимствовал².

Пример приводится для версии 10.2.0.3.0 СУБД.

Подготовка и запуск службы синхронизации для кластера (CSS)

CSS (Cluster Synchronization Service), как следует из названия, рассчитана на употребление кластерной разновидности БД Oracle (RAC), но эта служба используется и экземпляром ASM на обычном компьютере. Хотя установка CSS на конфигурации RAC выполняется и сложнее, в нашем случае единственного компьютера она проста. В версии 10.1 служба устанавливалась автоматически при установке ПО СУБД (независимо, нужна она пользователю, или не нужна), а с версии 10.2 ее требуется устанавливать самостоятельно.

Выдаем в ОС:

```
>localconfig reset
```

На Windows CSS оформлена как служба ОС, но запускать и останавливать ее можно и из командной строки, например:

```
>ocssd start
```

О других действиях можно узнать, выдав:

```
>ocssd --help
```

Обратите внимание, чтобы имя пользователя ОС не содержало русских букв.

На Unix CSS оформлена как демон *cssd*. При ближайшем рассмотрении `[ORACLE_HOME]/bin/localconfig` оказывается сценарием для командной оболочки ОС (*.bat* или *.sh*).

Подготовка места для дисковых групп

Если на дисках рабочего компьютера нет неформатированных файловой системой разделов, на которых положено размещать файловые группы для ASM, такие разделы можно промоделировать обычными файлами. Создавать их можно как угодно, но в статьях из интернета с поразительным постоянством приводится метод с использованием Perl. Воспользоваться им нетрудно, так как с версии 10 Perl включен в *ORACLE_HOME* «на законных основаниях» в подкаталог *perl* (а до этого он тоже присутствовал в *ORACLE_HOME*, но более завуалировано). Можно составить примерно следующую программу в файле *Createtextfiles.pl*:

```
my $s='0' x 2**20;

open (DF1, ">C:/asmdisks/_file_disk1") || die "Cannot create file - $!\n";
open (DF2, ">C:/asmdisks/_file_disk2") || die "Cannot create file - $!\n";
open (DF3, ">D:/asmdisks/_file_disk3") || die "Cannot create file - $!\n";
open (DF4, ">D:/asmdisks/_file_disk4") || die "Cannot create file - $!\n";

for ( my $i = 1; $i < 100; $i++ ) {
    print DF1 $s;
    print DF2 $s;
    print DF3 $s;
}
```

1 Отзывы первопроходцев не все были благоприятными, но уже в версии 10.2 картина улучшилась.

2 В интернете есть и другие статьи на эту тему, все вместе удивительно перекликающиеся общими моментами. В то же время [1] содержит ряд обозначений их документации по Oracle. Однако статья [1] является наиболее ясным и полным описанием, встреченным мною.

```
    print DF4 $s;
}

exit
```

Желающие могут отнестись к такому файлу утилитарно, но нетрудно понять, что здесь создаются четыре файла, содержащие 100 раз по 1024*1024 знаков '0', то есть по 100 мегабайтов. Заставить программу проработать можно следующими действиями:

```
>mkdir c:\asmdisks
>mkdir d:\asmdisks
>set ORACLE_HOME=c:\oracle\product\10.2.0\db_1
>set PERL_HOME=%ORACLE_HOME%\perl\5.8.3\bin\MSWin32-x86-multi-thread
>%PERL_HOME%\perl Createtextfiles.pl
```

Если на тренировочной машине одно дисковое устройство, все файлы придется разместить на нем.

Подготовка экземпляра +ASM

Для использования обычной СУБД средства ASM требуется дополнительное наличие на компьютере (в случае RAC – на каждом узле кластера) хотя бы одного специального экземпляра СУБД. Такие экземпляры могут носить разные имена. Ниже предполагается использование имени +ASM, которое считается умолчательным для этой техники. Вообще, экземпляр можно назвать как угодно, но принято начинать его имя со знака '+'.

Подготовить экземпляр +ASM можно, во-первых, средствами DBCA (для этого в DBCA версии 10 были введены несколько специальных форм), а во-вторых, вручную. Здесь рассматривается именно второй вариант, который в большей степени способен дать понимание происходящему.

Подготовка инфраструктуры

Как и для всякого экземпляра СУБД, для работы +ASM потребуется обеспечить место рабочим файлам. Выдадим в ОС:

```
>mkdir c:\oracle\product\10.2.0\admin\+asm\bdump
>mkdir c:\oracle\product\10.2.0\admin\+asm\cdump
>mkdir c:\oracle\product\10.2.0\admin\+asm\udump
```

Все же +ASM – вырожденный экземпляр СУБД, создаваемый по техническим причинам, а не для обслуживания доступа к какой-либо БД, а потому файл с параметрами для него может выглядеть необычно просто, например:

```
background_dump_dest = c:\oracle\product\10.2.0\admin\+ASM\bdump
core_dump_dest        = c:\oracle\product\10.2.0\admin\+ASM\cdump
user_dump_dest        = c:\oracle\product\10.2.0\admin\+ASM\udump

instance_type = ASM
compatible = 10.2.0.3.0      # для определенности

large_pool_size = 12M       # можно и больше, например, 64М, если потребуется

remote_login_passwordfile = exclusive

_asm_allow_only_raw_disks = FALSE      # этот пример без "сырых" устройств
asm_diskstring = 'c:\asmdisks\_file*', 'd:\asmdisks\_file*'
                                           # возможно групповое указание файлов
```

Файл размещается по правилам *INIT.ORA*. Назовем его *init+ASM.ora* и разместим в *%ORACLE_HOME%\database* (в Unix – в *\$ORACLE_HOME/dbs*).

Неочевидны специальная забота о *LARGE_POOL_SIZE* и обратное, необязательность *SHARED_POOL_SIZE* (в нашем случае легко проверить, что *SGA_TARGET* окажется = 0).

Параметр *_ASM_ALLOW_ONLY_RAW_DISKS* является недокументированным и неофициальным.

Параметр ASM_DISKSTRING ссылается на созданные предварительно файлы, имитирующие неформатированные устройства для размещения дисковых групп.

Другие возможные параметры, здесь не указанные:

- ASM_DISKGROUPS: явно задает дисковые группы, что будут автоматически монтироваться при запуске +ASM или монтировании командой ALTER DISKGROUP ALL MOUNT;
- ASM_POWER_LIMIT: регулирует автоматическую балансировку загрузки дисковых групп;
- ASM_PREFERRED_READ_FAILURE_GROUPS (версия 11.1): указывает предпочтительные группы отказа в дисковых группах;
- DB_UNIQUE_NAME: задает, если необходимо, имя, отличное от умолчательного +ASM.

Подготовка и запуск экземпляра СУБД

Поскольку здесь речь идет о Windows, требуется создать службу ОС:

```
>oradim -new -asmsid +ASM -startmode manual
>oradim -edit -asmsid +ASM -startmode auto
```

Перевод службы в режим автозапуска обсуждаем; главное – так, или иначе обеспечивать запуск +ASM ранее экземпляров СУБД, использующих +ASM на компьютере (но после запуска демона *cssd*).

Далее как обычно:

```
>set ORACLE_SID=+ASM
>sqlplus / AS SYSDBA
```

Версия 11 отнеслась более серьезно к разграничению доступа и ввела специальную привилегию SYSASM, так что подключаться начиная с нее следует так:

```
>sqlplus / AS SYSASM
```

В SQL*Plus набираем, как обычно:

```
SQL> STARTUP
```

Необычным будет результат: легко удостовериться, что СУБД перейдет в состояние STARTED, а не OPEN.

Кроме того, в ответ мы, скорее всего, увидим ошибку. Она вызвана тем, что +ASM не воспринимает наши файлы, как «дисковые группы»; не знает, как их использовать.

Создание дисковых групп

Работающий экземпляр +ASM по наводке параметра СУБД ASM_DISKSTRING распознает разделы, выделенные для дисковых групп:

```
COLUMN path FORMAT A30
SELECT
  group_number
, disk_number
, mount_status
, header_status
, state
, path
FROM v$asm_disk
.
SAVE asm_disks
@asm_disks
```

Чтобы превратить эти разделы в дисковые группы, можно выдать:

```
CREATE DISKGROUP prima_data1 NORMAL REDUNDANCY
```

```
FAILGROUP controller1 DISK 'c:\asmdisks\_file_disk1', 'c:\asmdisks\_file_disk2'  
FAILGROUP controller2 DISK 'd:\asmdisks\_file_disk3', 'd:\asmdisks\_file_disk4'  
;
```

Создана дисковая группа со свойством «нормальной избыточности» (NORMAL REDUNDANCY), запрещающей наличие менее двух групп отказа. Можно было указать HIGH REDUNDANCY (не менее трех групп отказа) или EXTERNAL REDUNDANCY (достаточно одной). Названия групп отказа можно было явно не задавать. Результаты выполненной команды можно наблюдать такими запросами:

```
COLUMN name FORMAT A15  
SELECT  
  group_number  
, name  
, total_mb  
, free_mb, state  
, type  
FROM v$asm_diskgroup  
.  
SAVE asm_diskgroup  
@asm_diskgroup  
@asm_disks
```

Другие сведения о конфигурации ASM можно посмотреть в нескольких прочих таблицах с именами LIKE 'V\$ASM%'.

Использование дисковых групп

Вот пример того, как созданную дисковую группу можно употребить для размещения в ней файла табличного пространства:

```
CONNECT /@prima.class AS SYSDBA  
CREATE TABLESPACE users2 DATAFILE '+prima_data1' SIZE 5M;
```

Проверка:

```
SQL> COLUMN name FORMAT A60  
SQL> SELECT file#, name FROM v$datafile;
```

```
FILE# NAME  
-----  
1 D:\ORACLE\PRODUCT\10.2.0\ORADATA\PRIMA\SYSTEM01.DBF  
2 D:\ORACLE\PRODUCT\10.2.0\ORADATA\PRIMA\UNDOTBS01.DBF  
3 D:\ORACLE\PRODUCT\10.2.0\ORADATA\PRIMA\SYSAUX01.DBF  
4 D:\ORACLE\PRODUCT\10.2.0\ORADATA\PRIMA\USERS01.DBF  
5 +PRIMA_DATA1/prima/datafile/users2.256.639862169
```

К новому «файлу» можно обращаться как к обычному, например:

```
ALTER DATABASE  
DATAFILE '+PRIMA_DATA1/prima/datafile/users2.256.639862169'  
RESIZE 10M  
;
```

Но поскольку это может показаться неудобным, для таких названий можно заводить псевдоним (alias). Подробности имеются в документации по Oracle.

«Файл» вполне работоспособен:

```
CONNECT scott/tiger@prima.class  
CREATE TABLE emp2 TABLESPACE users2 AS SELECT * FROM emp;
```

Проверяем отказоустойчивость

Приведем пример последовательности действий, доказывающей устойчивость такого «файла» к потерям в файловой системе:

```
CONNECT /@prima.class AS SYSDBA
SHUTDOWN IMMEDIATE
CONNECT / AS SYSDBA
SHUTDOWN
HOST del d:\asmdisks\_file_disk3
STARTUP
@asm_diskgroup
@asm_disks
ALTER DISKGROUP prima_data1 MOUNT;
@asm_diskgroup
@asm_disks
CONNECT /@prima.class AS SYSDBA
STARTUP
SELECT COUNT ( * ) FROM scott.emp2;
```

Сценарии *asm_diskgroup.sql* и *asm_disks.sql* были выданы для диагностики.

Восстановить утерянный файл помогут следующие действия. Воссоздадим файл *d:\asmdisks_file_disk3* программой на Perl указанным выше образом. Затем выполним следующую последовательность действий:

```
CONNECT /@prima.class AS SYSDBA
ALTER TABLESPACE users2 OFFLINE;
CONNECT / AS SYSDBA
ALTER DISKGROUP prima_data1 DISMOUNT;
SHUTDOWN
STARTUP
ALTER DISKGROUP prima_data1 MOUNT;
ALTER DISKGROUP prima_data1 ADD
FAILGROUP controller2 DISK 'd:\asmdisks\_file_disk3'
;
CONNECT /@prima.class AS SYSDBA
ALTER TABLESPACE users2 ONLINE;
SELECT COUNT ( * ) FROM scott.emp2;
```

При необходимости файл можно восстанавливать в любом другом надежном месте. Уточнить происходящее помогает своевременная выдача запросов в *asm_diskgroup.sql* и *asm_disks.sql*.

Упражнение. Повторить пример с потерей файлов, удалив, а затем восстановив, сразу два из них: *d:\asmdisks_file_disk3* и *d:\asmdisks_file_disk4*.