CN SOUG 中国南方ORACLE用户组

IT大咖说 知识共享平台

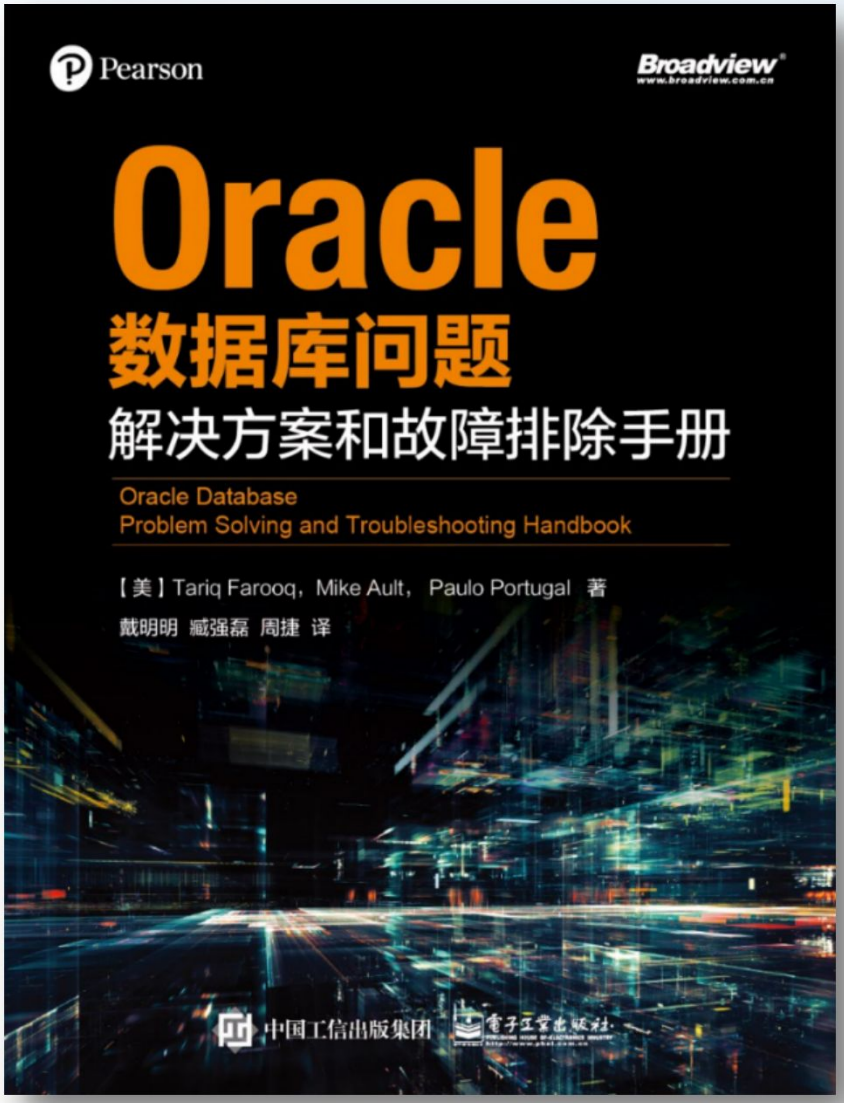
Oracle数据库云一体机性能压测和运维的探讨

戴明明/Dave

- 一体机压力测试
- 一体机的运维

个人介绍

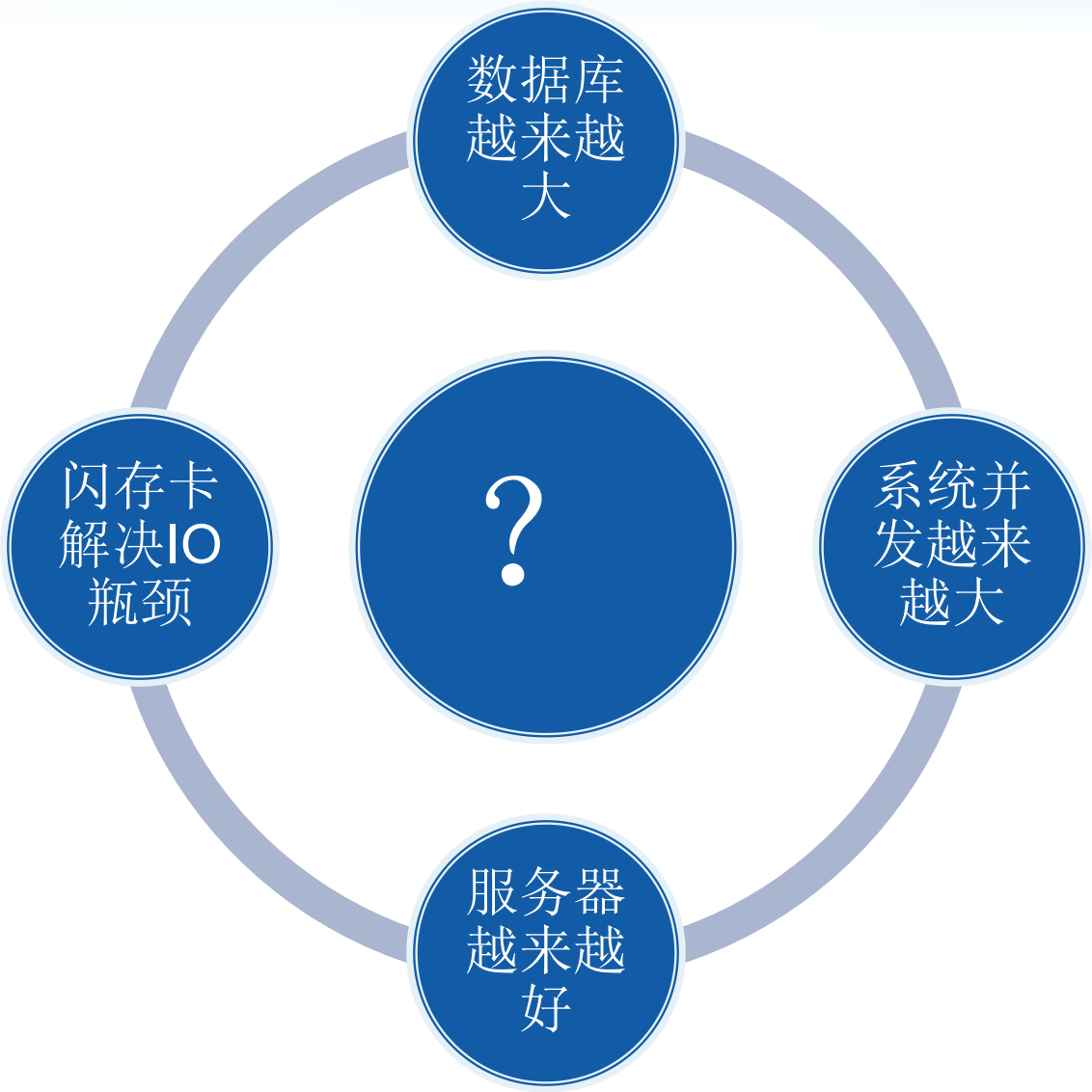
- 戴明明/Dave;
- Oracle ACE-A;
- 中国Oracle用户组核心成员;
- 中国南方Oracle用户组核心成员;
- 10年DBA经验，在Oracle 高可用方面有一定的经验积累;
- 热衷于Oracle 技术的研究与分享;
- 曾在CSDN博客撰写博客1000余篇(*http://blog.csdn.net/tianlesoftware*), 博客累积访问量超过1210万;
- 从2014年开始研究基于PCIe闪存卡的数据库高可用，高性能解决方案;
- 2014年Oracle 技术嘉年华上分享基于PCIe闪存卡的Oracle 高可用与高性能的解决方案;
- 2015年Oracle 技术嘉年华上分享Oracle 12c中海量数据的优化思路。
- 2017年翻译了《*Oracle 数据库问题解决和故障诊断手册*》一书
- 博客：http://www.cndba.cn/dave
- 邮箱：ahdba@qq.com
- 微信：tianlesoftware


Pearson
Broadview
www.broadview.com.cn
Oracle
数据库问题
解决方案和故障排除手册
Oracle Database
Problem Solving and Troubleshooting Handbook
【美】Tariq Farooq，Mike Ault，Paulo Portugal 著
戴明明 臧强磊 周捷 译
中国工信出版集团
電子工業出版社

- ***Tariq Farooq***，超过24年DBA经验，Oracle ACE和ACE Director。出版过《Oracle Exadata Expert's Handbook》和《the best-selling Expert Oracle RAC 12c》。
- ***Mike Ault***， 出版了超过25本书，其中包括《Oracle Administration and Management》和《Oracle DBA OCP ExamCram》。
- ***Paulo Portugal***，F2C DBA专家， 14年的Oracle DBA。
- ***Dr.Mohamed Houri***，Oracle独立顾问，有超过15年的Oracle经验。
- ***Syed Hussain***，Oracle ACE Director， 2011年的年度最佳DBA，有超过20年的IT 经验，其中15年是Oracle DBA，出版了多本Oracle 书籍。
- ***Jim Czuprynski***，Oracle ACE Director，有超过35年IT经验，其中15年是Oracle DBA。现在是OnX Enterprise Solutions公司的战略解决方案顾问，培训了超过2000个Oracle DBA。
- ***Guy Harrison***，戴尔研发部的执行董事，<Oracle Performance Survival Guide> 和 <MySQL Stored Procedure Programming >的作者。

一体机压力测试

压测的目的



配置选项	系统参数
处理器	最大支持8颗Intel至强E7-8800 v2系列处理器，主频最高可达3.4GHz，具备37.5MB大容量三级缓存，最多120个物理核心，240个线程，
内存	最大支持192条DIMM，最大可扩展12TB内存
硬盘数量	最大支持16个2.5寸热插拔SAS/SATA/SSD硬盘
IO扩展槽	26个标准PCI-E3.0插槽

容量	1.6TB	3.2TB	6.4TB
闪存类型	MLC	MLC	MLC
读带宽	2.6GB/s	2.6GB/s	2.6GB/s
写带宽	1.8GB/s	1.9GB/s	1.9GB/s
随机读延迟(4KB)	67us	67us	67us
随机写延迟(4KB)	9us	9us	9us
机读IOPS(4KB)	590,000	590,000	590,000
机写IOPS(4KB)	480,000	480,000	480,000
写入寿命	每天5次全盘写(5DWPD)，持续3年		

压测的方法

TPC (Transaction Processing Council：事务处理委员会）提供了量化的方法和标准。
TPC 的官网：http://www.tpc.org/

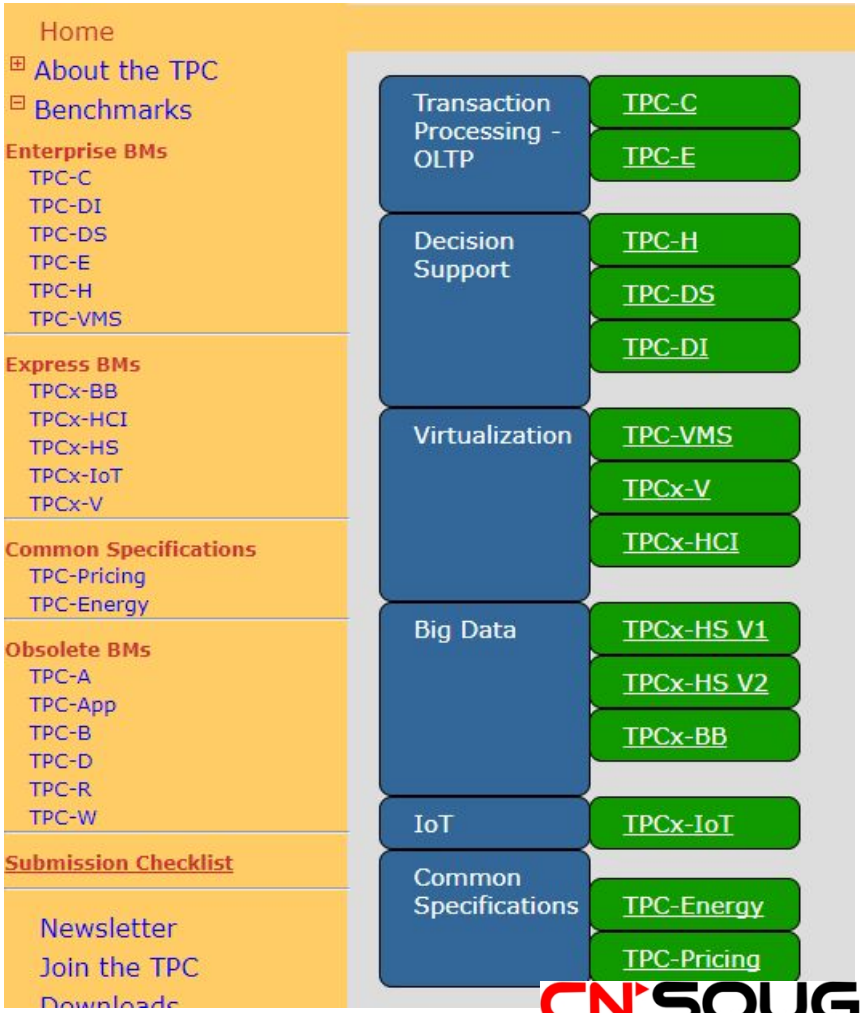
TPC-C模拟了一个订单系统，基准由交易组成，包括输入和交付订单、记录支付、检查订单状态以及监控仓库的库存水平。

最频繁的交易包括输入一个新订单，平均来说，它由10个不同的项目组成。

每个仓库都试图维护公司目录中的10万个项目的库存，并从该库存中填写订单。

TPC-C报告的性能度量指标度量了每分钟可以完全处理的订单数量。注意：TPC-C将被弃用，并被TPC-E代替

更多内容参考：
http://www.cndba.cn/dave/article/103



压测的工具

裸盘IO性能测试：

- FIO： linux平台下，http://www.cndba.cn/dave/article/104
- IOMETER: Windows平台，http://www.cndba.cn/dave/article/111

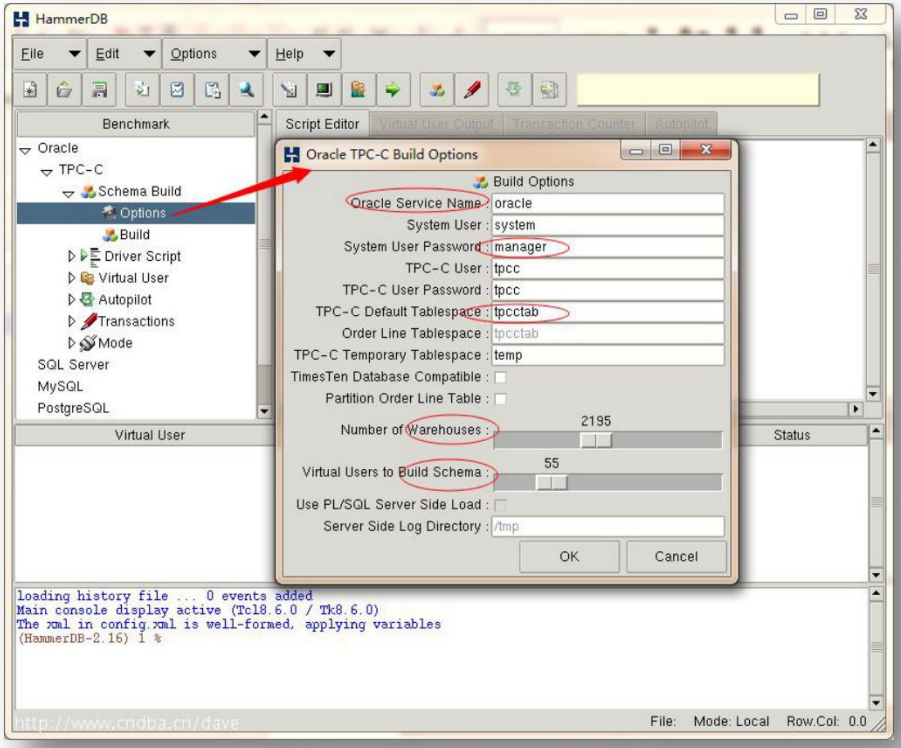
支持Oracle TPC-C的工具：

Swingbench: http://www.cndba.cn/dave/article/113

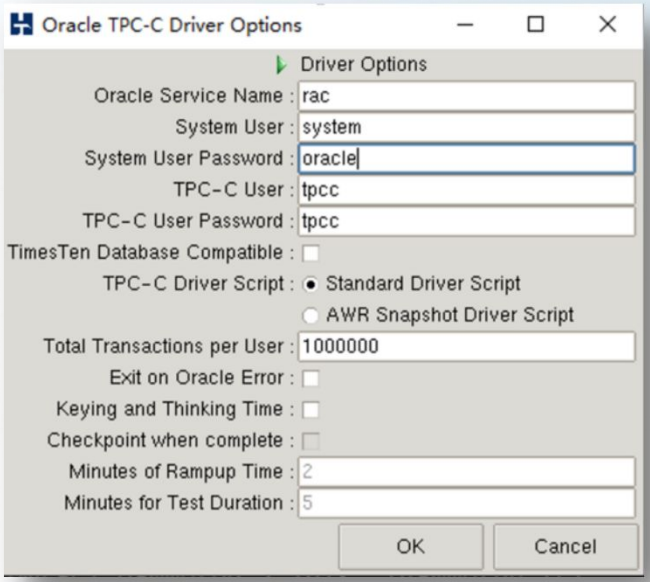
HammerDB：http://www.cndba.cn/dave/article/106

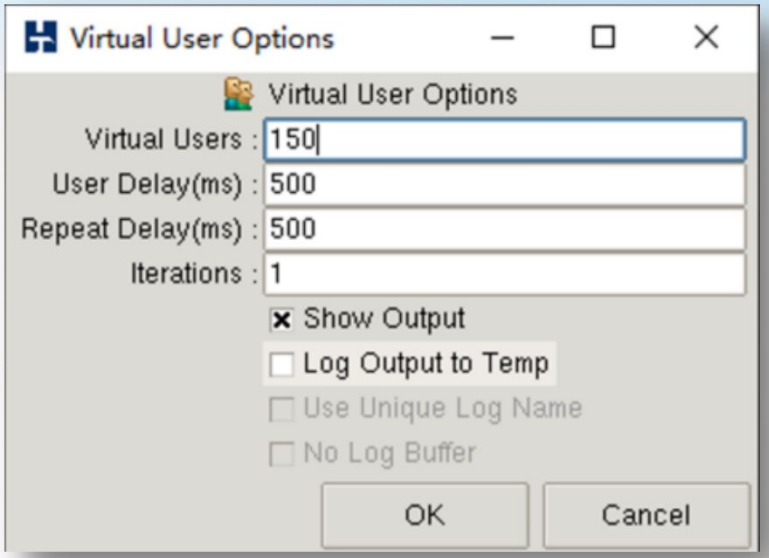
HammerDB的注意事项

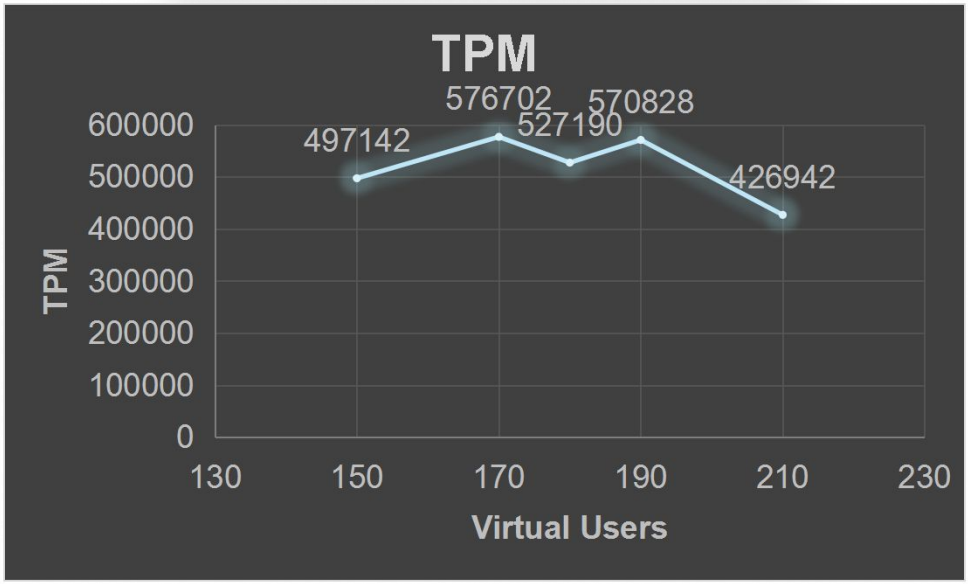
- Hammer DB 测试之前，需要先用Hammer DB 往数据库里加载测试数据。 数据单位是warehouse。
- 1000个warehouse需要80G的空间。 默认配置下，最大支持5000个warehouse，即需要500G的空间。
- Hammer DB 在加载完数据之后，会创建索引，收集统计信息，默认的脚本没有使用并行，所以这2步比较慢。
- 如果CPU 比较强劲，那么加载5000个warehouse 需要6个小时。
- warehouse: 数据的基数。 一般至少测试3000个warehouse。 如果想测试极限性能，建议压5000个warehouse。
- Build Schema： 加载warehouse的用户数，这里越多越好，也要和CPU 数匹配，一般是CPU数量的1-1.5倍。

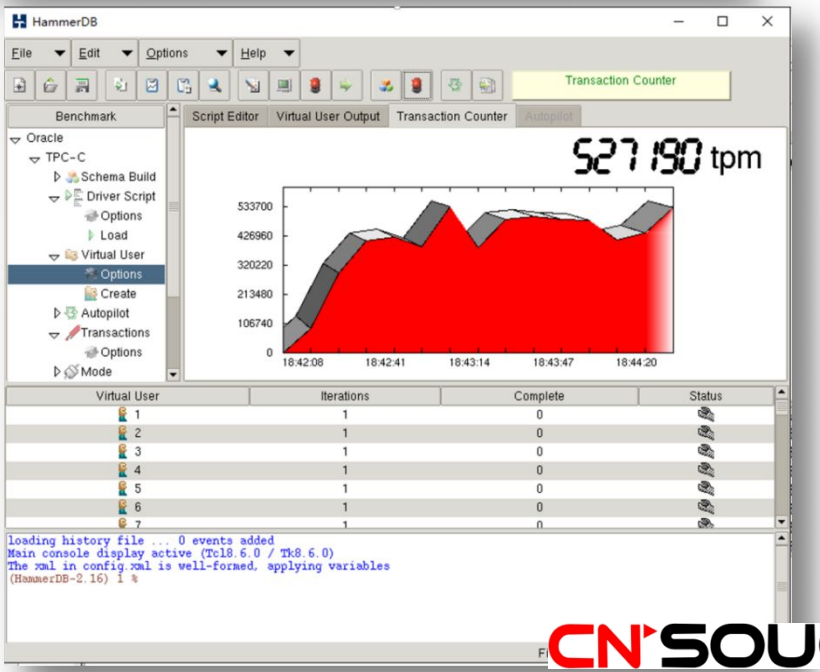


HammerDB中使用标准的测试脚本





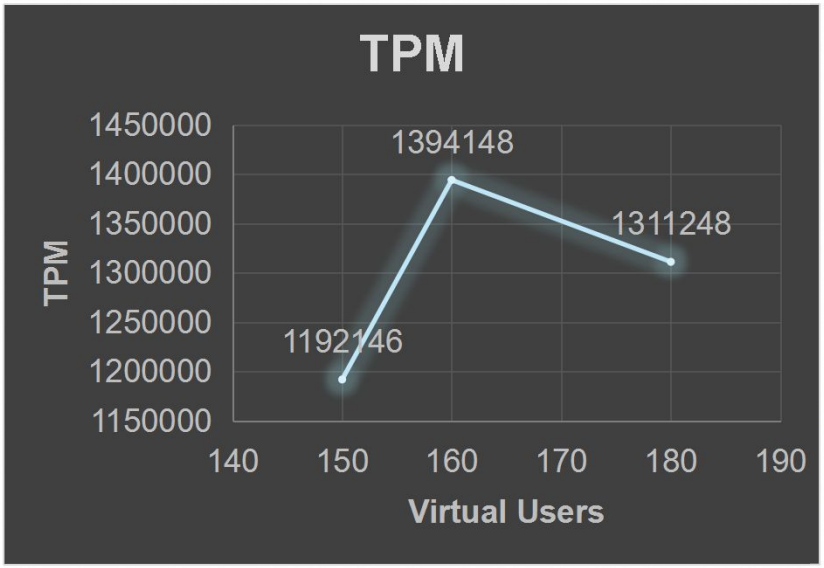


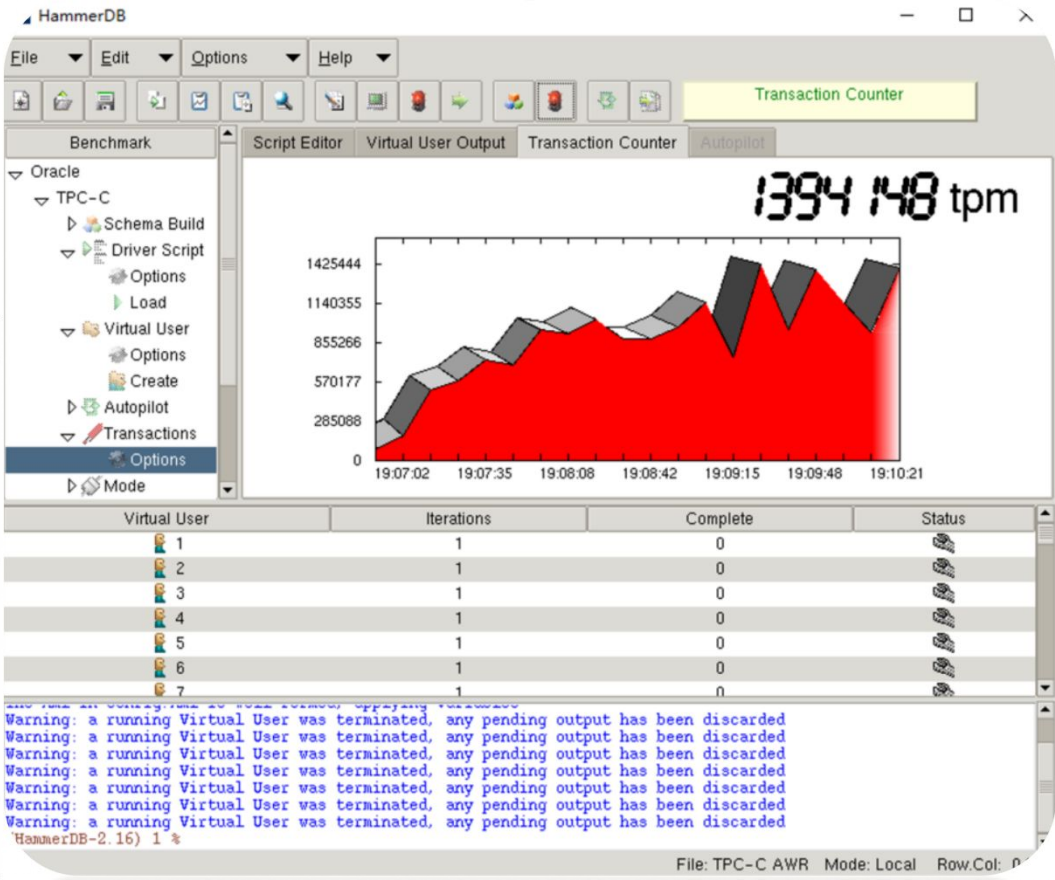


HammerDB中使用AWR快照的测试脚本

在与上节测试采用同样warehouse的情况下，使用awr快照计算出来的数据如下：

Virtual User	TPM
150	1192146
160	1394148
180	1311248





更多详细指标：

在测试的同时，可以使用Orz工具查看更多详细的指标：

```
[root@dg1 ~]# orz dbs
|--------|---load average--|-------|--------|-------IOPS------|-----MBPS----|-------|--LOGICAL--|----PHYSICAL----|
|--time--|--1m---5m---15m--|---TPS-|---QPS--|---reads/writes--|--recv/send--|-redos-|---reads---|-(reads/writes)-|
|14:23:07| 6.88 8.75 11.76 |   553 |   4466 |   1513    680   |   12M    7M |    2M | 441375    |   1377    442  |
|14:23:09| 6.88 8.75 11.76 |   557 |   4562 |    579    695   |    5M    8M |    2M | 566940    |    582    348  |
|14:23:12| 6.89 8.72 11.74 |   182 |   1579 |    592    489   |    5M    5M |1017K | 485663    |    554    306  |
|14:23:14| 6.89 8.72 11.74 |   340 |   3165 |    418    438   |    3M    2M | 820K | 674347    |    415      0  |
|14:23:17| 6.66 8.64 11.70 |   124 |   1296 |    367    290   |    3M    2M | 687K | 669638    |    265      0  |
|14:23:19| 6.45 8.56 11.66 |   296 |   3079 |   1282    453   |   10M    4M | 707K |1432690    |   1292    301  |
|14:23:22| 6.45 8.56 11.66 |   119 |   1483 |    369    674   |    3M    7M | 679K | 396599    |    357    377  |
|14:23:24| 6.33 8.51 11.62 |   181 |   2677 |    307    427   |    2M    4M | 540K | 709095    |    293    369  |
|14:23:27| 6.33 8.51 11.62 |   124 |   1612 |   3451    316   |   27M    1M | 678K | 362492    |   3380   1088  |
|14:23:29| 6.22 8.45 11.58 |   164 |   1524 |    280    244   |    2M    4M |    2M | 823902    |    283      0  |
|14:23:32| 6.22 8.45 11.58 |    87 |   1040 |    360    558   |    3M    8M |    2M | 975095    |    354    490  |
|14:23:35| 6.20 8.41 11.55 |    78 |    867 |    247    922   |    2M   17M |    1M |2030313    |    237   1531  |
|14:23:37| 6.20 8.41 11.55 |   242 |   3311 |    389    577   |    3M    5M | 654K | 597779    |    374    798  |
|14:23:40| 5.87 8.30 11.50 |   130 |   1585 |    582    328   |    5M    1M | 680K | 878510    |    541     86  |
|14:23:42| 5.87 8.30 11.50 |   223 |   2979 |    336    322   |    3M    1M | 540K | 480947    |    336      0  |
```

压测结果的进一步分析：

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
ora11g	Linux x86 64-bit	240	120	8	1007.79

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	73	20-Oct-14 10:09:39	255	5.3
End Snap:	74	20-Oct-14 10:39:42	258	
Elapsed:		30.05 (mins)		
DB Time:		5,156.93 (mins)		

性能数据很好看：
每秒生成Redo log 350M
每秒执行SQL: 139w
每秒执行事务：6.7w
每秒用户调用：5.1w

Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	171.6	0.0	0.00	0.00
DB CPU(s):	162.0	0.0	0.00	0.00
Redo size (bytes):	349,888,630.2	5,190.3		
Logical read (blocks):	8,073,773.8	119.8		
Block changes:	2,145,992.7	31.8		
Physical read (blocks):	2,319.9	0.0		
Physical write (blocks):	19,354.9	0.3		
Read IO requests:	2,094.8	0.0		
Write IO requests:	5,466.7	0.1		
Read IO (MB):	18.1	0.0		
Write IO (MB):	151.2	0.0		
User calls:	51,681.8	0.8		
Parses (SQL):	28,748.1	0.4		
Hard parses (SQL):	0.0	0.0		
SQL Work Area (MB):	14.2	0.0		
Logons:	0.0	0.0		
Executes (SQL):	1,391,944.6	20.7		
Rollbacks:	113.1	0.0		
Transactions:	67,412.3			

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
cursor: pin S	16,784,948	126K	8	72.2	Concurrency
DB CPU		41.8K		24.0	
library cache: mutex X	3,349,849	3043.6	1	1.7	Concurrency
enq: TX - row lock contention	97,609	2285.9	23	1.3	Application
latch: enqueue hash chains	250,032	570.8	2	.3	Other
log file sync	33,047	242	7	.1	Commit
db file sequential read	922,557	240.1	0	.1	User I/O
latch free	36,161	80.7	2	.0	Other
latch: In memory undo latch	571,095	65.6	0	.0	Concurrency
db file scattered read	163,298	57.3	0	.0	User I/O

等待事件:
cursor：pin S 占72%
DB CPU 占 24%

这个符合我们的测试过程：
频繁执行SQL共享解析

进一步优化措施1： 4k Online Redo Log

扇区大小

上一代存储多采用512 bytes的扇区，现在的存储则采用4k的扇区，扇区即每次最小IO的大小。

4k 扇区有两种工作模式：native mode 和 emulation mode。

- Native mode，即4k模式，物理和逻辑的block大小一样，都是4096 bytes。 Native mode 的缺点是需要操作系统和软件（如DB）的支持。 Oracle 从11gR2 开始支持4k IO操作。 Linux 内核在2.6.32 之后也开始支持4k IO操作。
- emulation mode：物理块是4k，但逻辑块是512 bytes。 在该模式下，IO操作时底层物理还是4k进行操作，所以就会导致Partial I/O 和4k 对齐的问题。

在emulation mode下，每次IO操作大小是512 bytes，但存储底层的IO操作大小必须是4k，如果要读512 bytes的数据，实际需要读4k，是原来的8倍，就是partial IO。 而在写时，也是先读4k 的物理block，然后更新其中的512 bytes的数据，在把4k 写回去。

所以在emulation mode下，增加的工作会增加延时，降低性能。

进一步优化措施1： 4k Online Redo Log

4k block size

在Oracle 数据库的文件中，默认情况下，datafile的block 是8KB，控制文件是16KB，所以都没有partial IO的问题，唯有online redo log，默认是512 bytes，存在partial IO的问题。

从Oracle 11gR2 开始，在存储支持4k扇区的情况下，可以创建Blocksize 为512，1024，4096 的redo log。如：

```
alter database add logfile group 5 size 100m blocksize 4096;
```

如果是emulation mode的4k扇区，创建4k 的redo log时可能会触发如下错误：

```
ORA-01378: The logical block size (4096) of file +DATA is not compatible with the disk sector size (media sector size is 512 and host sector size is 512)
```

只要确认存储物理是4k的扇区，可以设置_disk_sector_size_override参数为true，来覆盖扇区的设置。该参数支持动态修改，如：

```
ALTER SYSTEM SET “_DISK_SECTOR_SIZE_OVERRIDE”=”TRUE”;
```

进一步优化措施2： 修改warehouse表空间

把STOCK和CUSTOMER表改成分区表

Segments by Physical Reads

- Total Physical Reads: 4,183,024
- Captured Segments account for 97.9% of Total

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	Physical Reads	%Total
TPCC	TPCCTAB	CUSTOMER		TABLE	1,924,770	46.01
TPCC	TPCCTAB	STOCK		TABLE	1,710,149	40.88
TPCC	TPCCTAB	CUSTOMER_I2		INDEX	45,058	1.08
TPCC	TPCCTAB	HISTORY		TABLE	35,840	0.86
TPCC	TPCCTAB	IORDL	SYS_P71	INDEX PARTITION	28,692	0.69

Back to Segment Statistics
Back to Top

Segments by Physical Read Requests

- Total Physical Read Requests: 3,777,004
- Captured Segments account for 97.8% of Total

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	Phys Read Requests	%Total
TPCC	TPCCTAB	CUSTOMER		TABLE	1,783,400	47.22
TPCC	TPCCTAB	STOCK		TABLE	1,467,806	38.86
TPCC	TPCCTAB	CUSTOMER_I2		INDEX	44,255	1.17
TPCC	TPCCTAB	HISTORY		TABLE	35,544	0.94
			SYS_P71	INDEX PARTITION	28,692	0.76

Segments by DB Blocks Changes

- % of Capture shows % of DB Block Changes for each top segment compared
- with total DB Block Changes for all segments captured by the Snapshot

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	DB Block Changes	% of Capture
TPCC	TPCCTAB	STOCK		TABLE	199,046,256	38.27
TPCC	TPCCTAB	INORD		INDEX	32,506,048	6.25
TPCC	TPCCTAB	IORDL	SYS_P71	INDEX PARTITION	19,857,936	3.82
TPCC	TPCCTAB	IORDL	SYS_P60	INDEX PARTITION	15,008,576	2.89
TPCC	TPCCTAB	IORDL	SYS_P66	INDEX PARTITION	14,920,640	2.87

进一步优化措施3： 减少cursor pin S

使用dbms_shared_pool.markhot过程标记library cache 对象为 hot object.

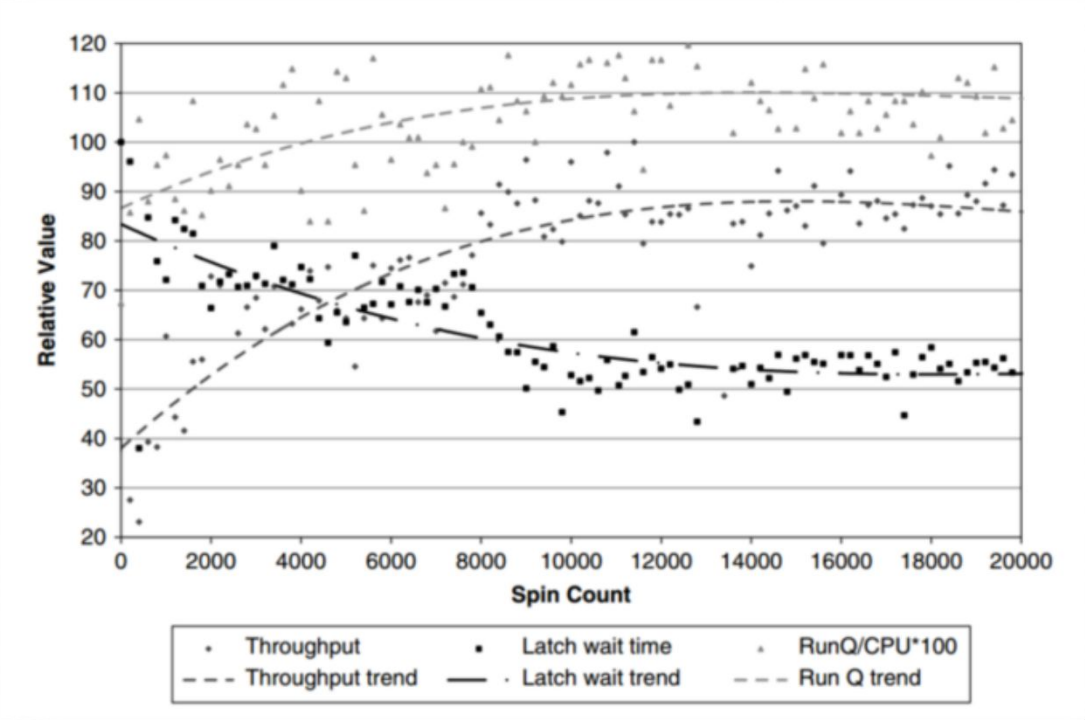
```
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_RANDOM',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_RANDOM',NAMESPACE=>2);
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_OUTPUT',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'SYS',objname=>'DBMS_OUTPUT',NAMESPACE=>2);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'NEWORD',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'PAYMENT',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'DELIVERY',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'OSTAT',NAMESPACE=>1);
exec dbms_shared_pool.markhot(schema=>'TPCC',objname=>'SLEV',NAMESPACE=>1);
```

Oracle Namespace 说明
http://www.cndba.cn/dave/article/1383

进一步优化措施4： 增加 _spin_cou...

_spin_count 参数默认值为 2000 ，对于高速 CPU 来说，这个值太低了，其可能在没有问题的情况下会导致太多的闩锁休眠。这个隐含参数依赖于 CPU 的速度，而默认值 2000 是基于Oracle 7.0 发布时可用的 CPU 速度。

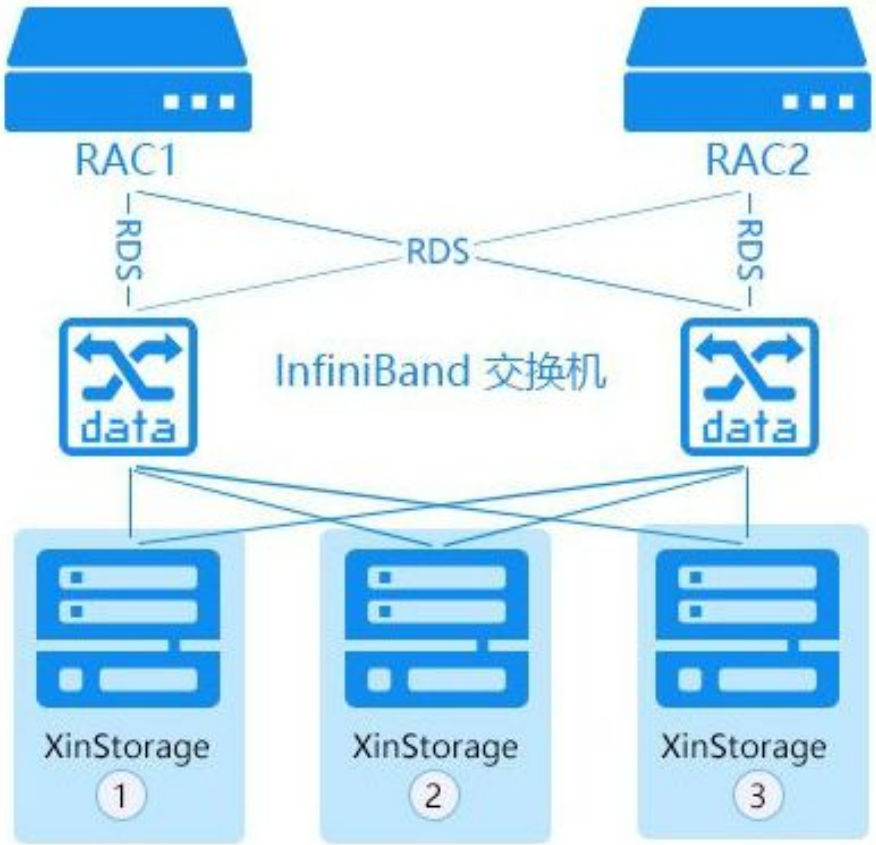
在CPU 还有空闲的情况下，可以通过增加_spin_count参数值，来提升应用程序的吞吐量。



Latch 专家 Andrey Nikolaev的测试数据(http://arxiv.org)

一体机的运维

一体机的架构



- 计算节点

基于X86 服务器的Linux系统，安装运行 Oracle 软件。

- 存储节点

基于X86服务器Linux 系统，安装PCIe 闪存卡，SSD 或者 HDD等类型的存储单元。

- InfiniBand 交换机

采用 56Gb/s 高带宽低延迟的网络传输设备。

- 存储单元管理软件

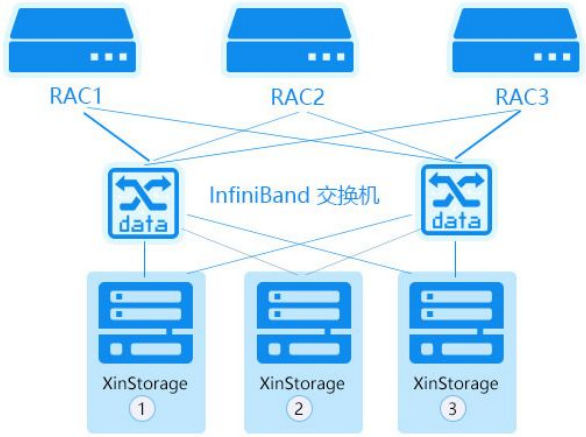
基于InfiniBand的高速网络的存储单元管理软件，它对存储单元进行统一管理并提供高并发、低延迟的IO请求。

- 智能监控平台

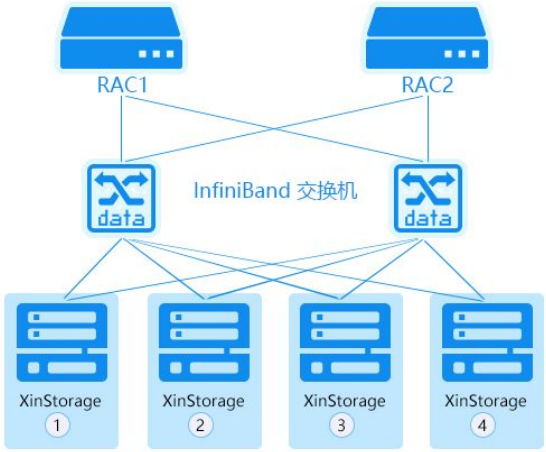
监控管理平台可以对数据库、操作系统和存储设备提供全面的可视化监控及自动化运维。

一体机的扩容

- 高性能：通过IB网络和PCIe Flash解决系统的IO瓶颈，5~10倍于传统架构的RAC环境。
- 高可靠性：核心组件均采用双机和冗余模式，高度的数据冗余与RAC保障系统的高可用性。
- 在线扩展：计算节点与存储节点均可以根据需要进行扩展。
- 按需定制：可以根据用户的需求提供定制化的计算性能和存储容量。

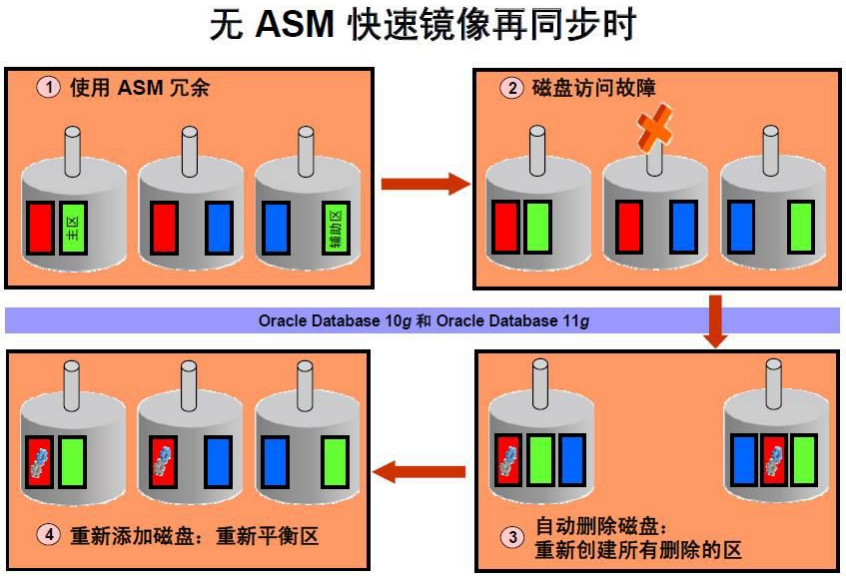


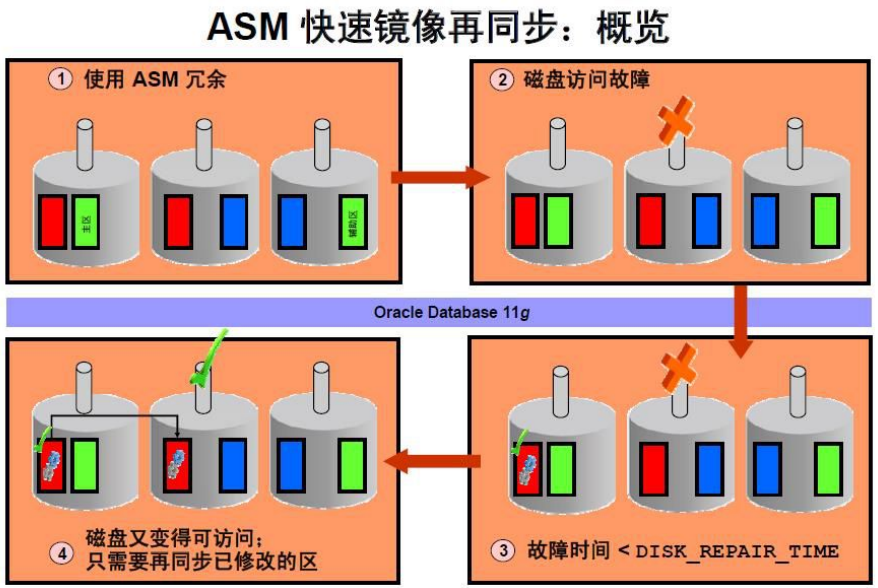
计算节点在线扩容



存储节点在线扩容

ASM 快速镜像再同步（Fast Mirror Resync）





ASM 快速镜像再同步（Fast Mirror Resync）

前提条件：
1）磁盘组是NORMAL或者HIGH。
2）compatible.asm & compatible.rdbms = 11.1.0.0.0 or higher
3） 设置磁盘组的 DISK_REPAIR_TIME 参数，默认3.6小时

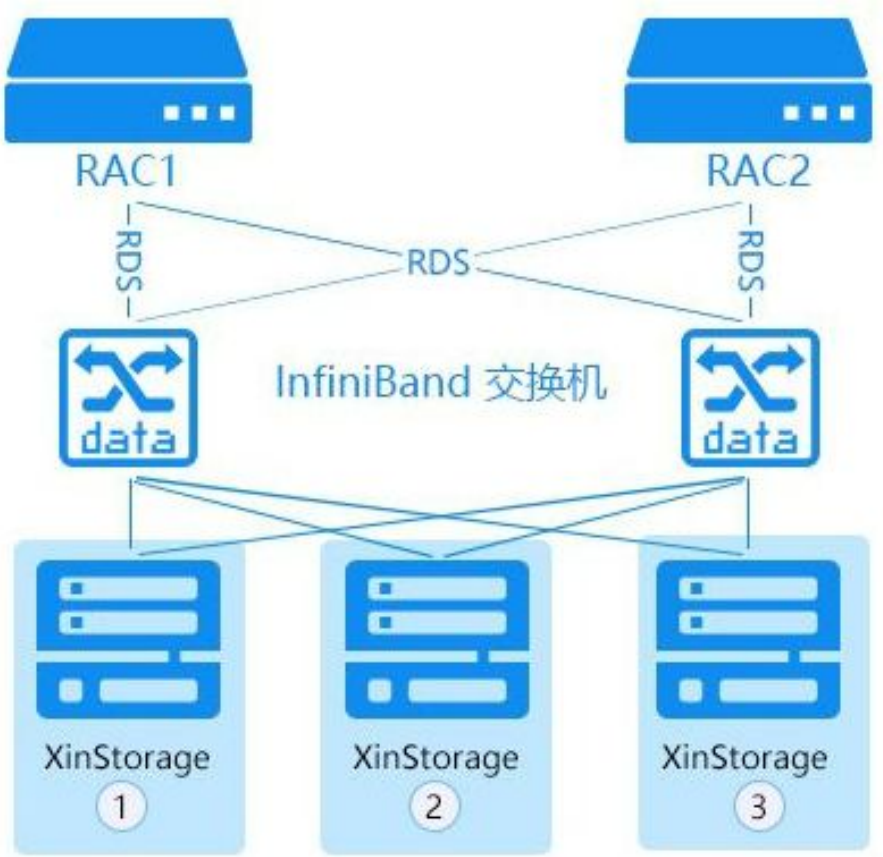
```
SQL> alter diskgroup asm set attribute 'compatible.rdbms'='11.2.0.0.0';

SQL> alter diskgroup asm set attribute 'disk_repair_time'='5h';

SQL> select name,value from v$asm_attribute where group_number=2;

NAME                                                             VALUE
---------------------------------------------------------------- --------------------
disk_repair_time                                                   3.6h
au_size                                                            1048576
sector_size                                                        512
compatible.asm                                                     11.2.0.0.0
compatible.rdbms                                                   11.2.0.0.0
```

一体机存储节点的扩容

加节点好？

还是加闪存卡好？

如果掉一个硬盘怎么处理？

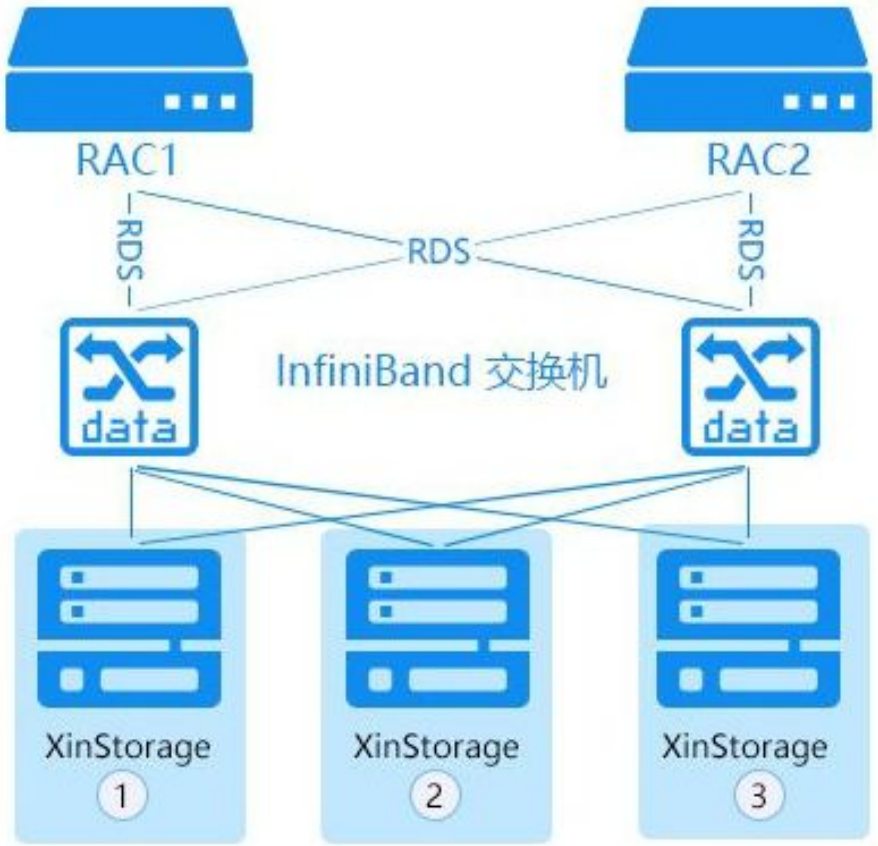


注意事项：
每操作一个节点后，务必等待Rebalance 操作结束。查询V$ASM_OPERATION试图。

一体机中数据的冷热数据分层

- 一体机利用PCIe闪存卡和IB网络来提升整体的性能。
- 数据库越来越大，存储成本就会越来越高。

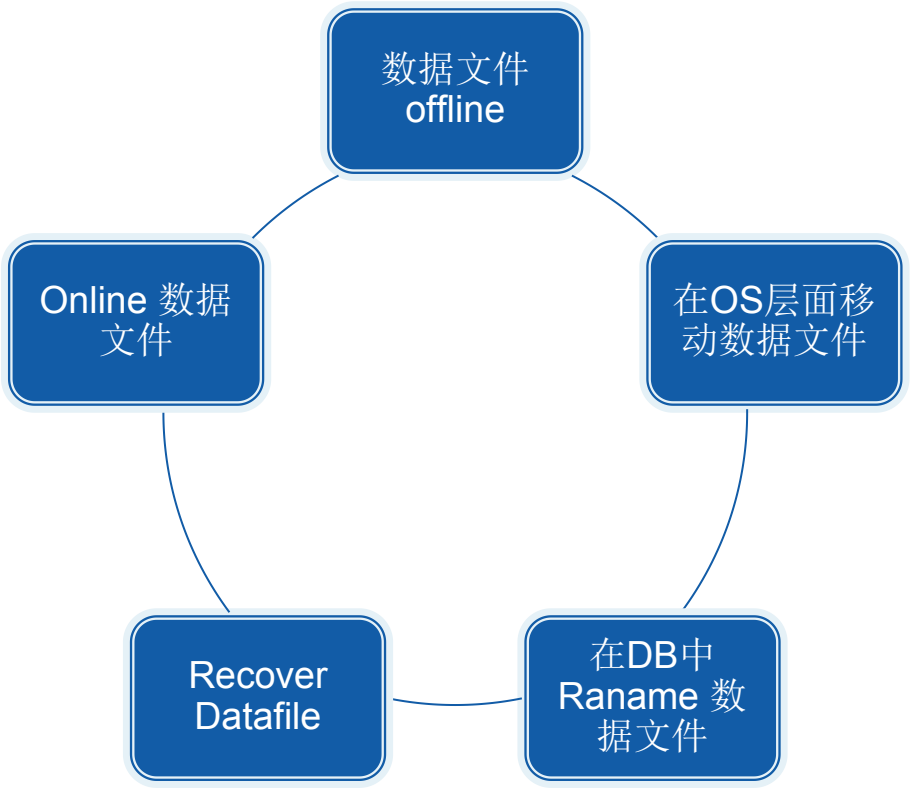
冷热数据分层



Oracle 11g – 数据转历史

- 交换分区
- 导出导入
- 对于非分区表，人工处理数据

Oracle 12c– Online Datafile Move



Oracle 12c – Online Datafile Move

在Oracle 12c中直接Move 即可：

```
SQL> alter database move datafile
'/u01/app/oracle/oradata/dave/huaining.dbf' to
'/u01/app/oracle/oradata/dave/anqing.dbf';
```

这种移动不仅会修改控制文件中的信息，也会在OS级别物理的移动。

利用这个特性可以做如下事情：

- 在线移动数据文件位置
- 在线重命名数据文件
- 在线移动数据文件从ASM到文件系统

Oracle 12c - 热图（HEAT MAP）

Heat Map 是Oracle 12c引入的新特性，用来存储系统在row 和segment 级生成的数据使用情况统计信息。

段级热图跟踪表和分区上次修改和访问的时间。行级热图显示行级的修改时间。

ADO可以利用Heat Map的信息自动执行数据压缩和移动的存储策略，以便降低存储成本、提高性能。

热图一经启用，即可自动收集段级和行级统计信息，在收集这些信息时会跳过针对系统任务执行的内部访问（自动排除统计信息收集、DDL 或表重定义）。

Oracle 12c - 热图（HEAT MAP）

默认情况下，数据库是禁用Heat Map的，启动Heat Map需要修改如下参数：

```
SQL> show parameter heat_map
NAME                 TYPE                 VALUE
-------------------- -------------------- ------------------
heat_map             string               OFF


SQL> ALTER SYSTEM SET HEAT_MAP = ON;


SQL>  show parameter heat_map
NAME                  TYPE                  VALUE
--------------------- --------------------- ---------
heat_map              string                ON
```

Oracle 12c - Automatic Data Opti

ADO 可以对表或者表分区在segment 或者row 级别创建压缩策略和移动策略，从而实现存储和压缩的分层。数据压缩的判断依据就是来自Heat Map收集的信息。

策略条件可以指定如下值：如no data access，no data modification，或者n days，months，years 没有修改。

压缩类型包括：

ROW STORE COMPRESS （Basic 压缩）
ROW STORE COMPRESS ADVANCED (Advanced Row 压缩）
COLUMN STORE COMPRESS FOR QUERY LOW/HIGH (HCC Query）
COLUMN STORE COMPRESS FOR ARCHIVE LOW/HIGH (HCC Archive）

Oracle 12c - Automatic Data Optimization

可以在创建表的时候指定ILM ADO策略，也可以在创建之后使用alter 语法来修改。

```
CREATE TABLE sales_ado
 (
PROD_ID NUMBER NOT NULL,
  CUST_ID NUMBER NOT NULL,
  TIME_ID DATE NOT NULL,
  QUANTITY_SOLD NUMBER(10,2) NOT NULL,
  AMOUNT_SOLD NUMBER(10,2) NOT NULL
)
PARTITION BY RANGE (time_id)
(
PARTITION sales_q3_2012 VALUES LESS THAN (TO_DATE('01-OCT-2012','dd-MON-yyyy')),
PARTITION sales_q4_2012 VALUES LESS THAN (TO_DATE('01-JAN-2013','dd-MON-yyyy'))
)
ILM ADD POLICY COMPRESS FOR ARCHIVE HIGH SEGMENT
AFTER 12 MONTHS OF NO ACCESS;
```

Oracle 12c - Automatic Data Optimization (ADO)

ADO 的TIER 分层语法：

```
ALTER TABLE sales_ado ILM ADD POLICY  TIER TO my_low_cost_tablespace;
```

这个策略的作用是把数据自动转移低性能的表空间，通俗一点，就是放在便宜一点的存储设备上，降低硬件成本。

触发由ILM的参数决定：

```
SQL> select * from dba_ilmparameters;

NAME                          VALUE
----------------------------  ---------------
ENABLED                       1
RETENTION TIME                30
JOB LIMIT                     2
EXECUTION MODE                2
EXECUTION INTERVAL            15
TBS PERCENT USED              85
TBS PERCENT FREE              25
POLICY TIME                   0
```

感 谢