



# 2014中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2014



大数据技术探索和价值发现

## Oracle Exadata 在数据仓库系统中的运用





# 从认识开始

□ 王科 ([wangke@shsnc.com](mailto:wangke@shsnc.com))

□ 13632429857

□ [www.shsnc.cn](http://www.shsnc.cn)

□ 8年ORACLE从业经验

□ 开发、集成、运维、架构多岗位经验

□ 现任职于国内最大的数据库服务厂商

**上海新炬**

□ DTCC2013 分享主题：

*Oracle TimesTen企业级应用实践*

□ DTCC2014 分享主题：

*Exadata在数据仓库系统中的运用*





## 5年后我们这个行业会怎么样？



世界级Oracle专家Jonathan Lewis：我很为DBA们的未来担心  
(图灵访谈 2013/11)



1 数据量迈入PB时代

2 集成式一体机时代正式来临

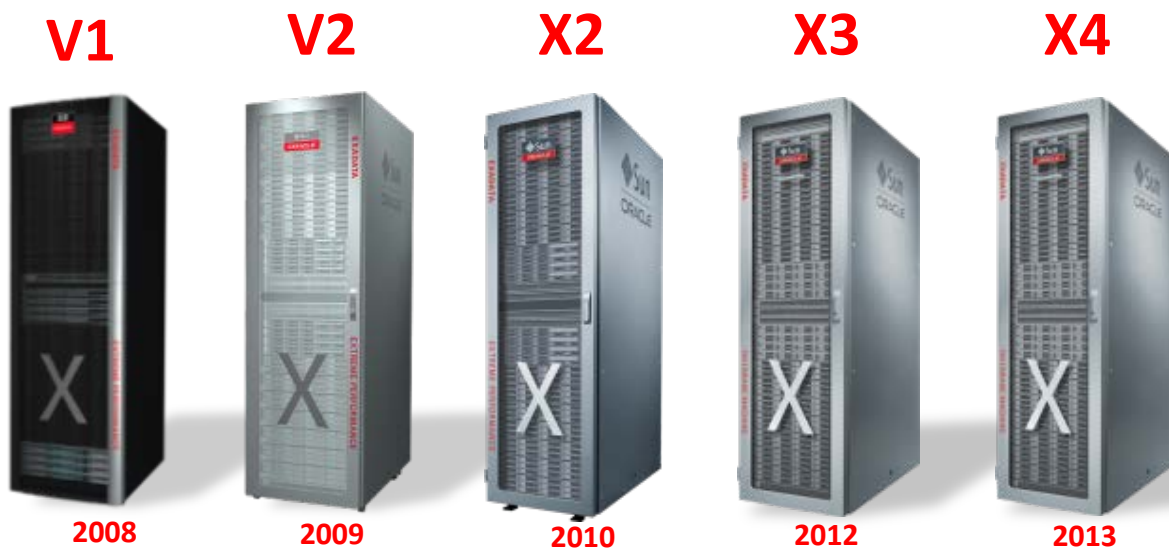
3 中小企业IT业务系统向云端转移

4 对技术架构与运维的关注度远高于某个产品

5 基础设施外包，应用向套装或自行开发转移



# Exadata的发展史，硬件能力不断提升



Storage (TB)	168	336	504	504	672	4X
Flash (TB)	0	5.3	5.3	22.4	44.8	8X
CPU (Cores)	64	64	96	128	192	3X
Memory (GB)	256	576	1152	2048	4096	16X
Connectivity (Gb/s)	8	24	184	400	400	50X

注：数据来源于官方宣传文档。





# Exadata — 分享大纲

1. 混合列  
压缩

3. 索引使用

5. 优化案例

2. 数据处理

4. 存储使用

6. EM12C  
监控

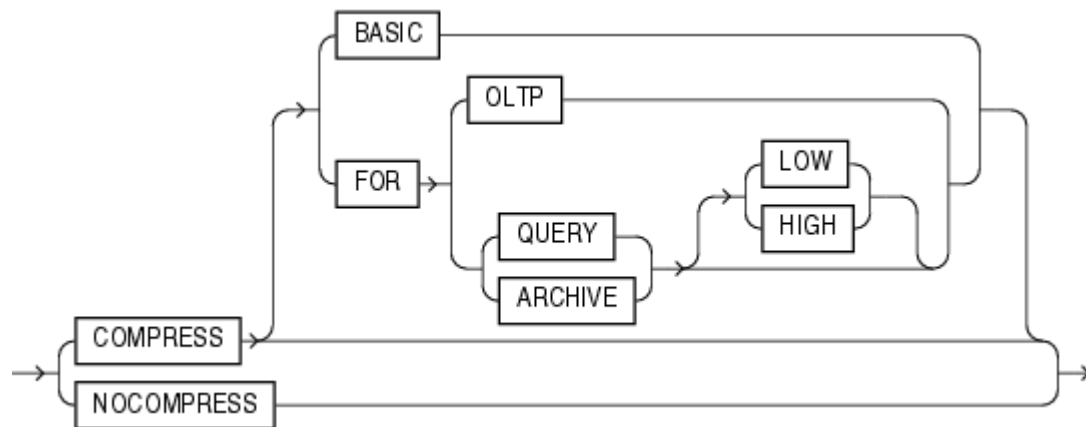


有一辆好车，就能跑起来了么？



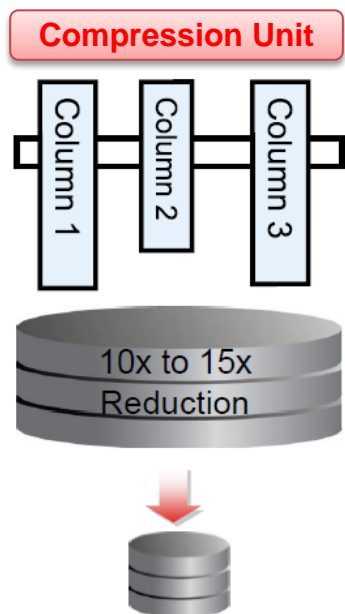
# (1/6)混合列压缩：数据仓库中不得不谈的核心功能！

*table\_compression::=*



EHCC	方式/速度	压缩比
QUERY LOW	LZO	6x
QUERY HIGH	ZLIB	10x
ARCHIVE LOW	ZLIB	12x
ARCHIVE HIGH	BZ2	16x

注：以上压缩比来自某数据仓库的实测值



## • 逻辑压缩单元

- CU是跨多个数据库块的逻辑结构
- CU大小由数据库自动确定
- 在加载数据时按列组织数据
- 每一列都分别进行压缩
- 支持智能扫描

## • 适合相对静止的数据

- 更新期间CU内的所有行会被锁定
- 更新会导致重组整个CU，压缩降级或不压缩

- ✓ CREATE ... as SELECT
- ✓ APPEND INSERT
- ✓ IMPDP
- ✓ ALTER TABLE MOVE

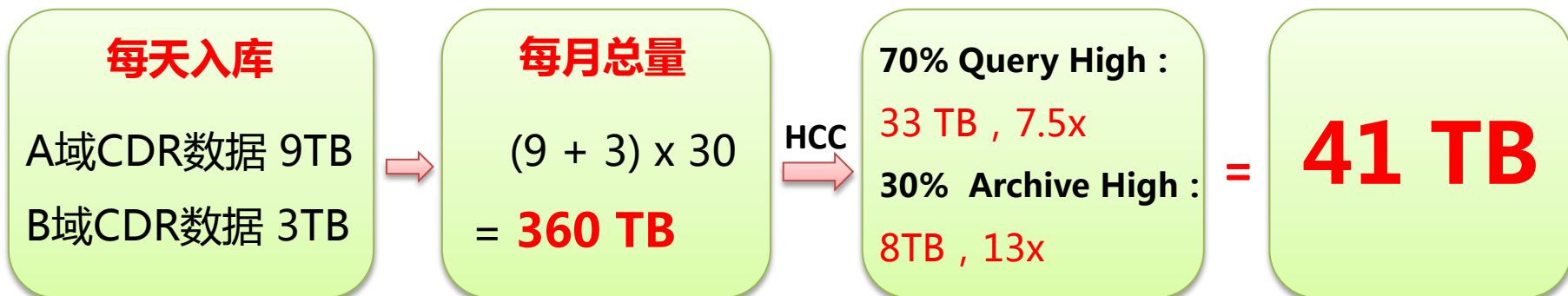
## ● 节约存储成本

## ● 减少I/O扫描



# (1/6)混合列压缩：案例分享及压缩方式建议

## ➤ 某运营商数据仓库混合列压缩案例(X2, Half Rack):



## ➤ 常用的两种压缩方式:

### ● 方式一：预定义

- CREATE TABLE BIGTABLE ..... COMPRESS FOR QUERY HIGH;
- SQL\*Loader / IMPDP / APPEND INSERT

### ● 方式二：后压缩

- CREATE TABLE BIGTABLE ..... NOCOMPRESS;
- SQL\*Loader / IMPDP / APPEND INSERT
- ALTER TABLE BIGTABLE COMPRESS FOR QUERY HIGH;
- or: ALTER TABLE BIGTABLE MOVE PARTITION P1 COMPRESS FOR QUERY HIGH;
- or: ALTER TABLE BIGTABLE MOVE PARTITION Pn COMPRESS FOR ARCHIVE HIGH;

不同分区，  
可使用不同的  
压缩类型



## (2/6)数据处理：集合数据处理

先来看一个简单的例子：

```
DECLARE
  CUR REC_CUR;
  REC TYPE_TABLE%ROWTYPE;
BEGIN
  OPEN CUR FOR SELECT * FROM TABLE1;
  LOOP
    FETCH CUR INTO ...
    IF CONDITION(REC)
      THEN
        INSERT INTO TABLE2 ...
      ELSE
        INSERT INTO TABLE3 ...
    END IF;
    COMMIT;
  END LOOP;
END;
```

逐行的数据处理方式，在大数据集合中，是否还合适？





## (2/6)数据处理：集合数据处理

集合数据处理的实现方式：(有哪些好处?)

```
INSERT /*+ append */ INTO TABLE2  
  SELECT * FROM TABLE1  
  WHERE CONDITION ...
```

```
INSERT /*+ append */ INTO TABLE3  
  SELECT * FROM TABLE1  
  WHERE NOT CONDITION ...
```

也可以这样实现：

```
INSERT /*+ append */ FIRST  
  WHEN CONDITION THEN  
    INTO TABLE2 VALUES ...  
  ELSE  
    INTO TABLE3 VALUES ...  
  SELECT * FROM TABLE1;
```

- ✓ 尽量简洁，SQL 应该告诉数据库做什么而不是怎么做
- ✓ 集合数据处理应有效的使用CPU 和 IO等 资源



## (2/6)数据处理：转变思维，DML重写 (*DELETE*)

□ 常见的DML场景： delete

如果是大量数据的delete，可能会出现什么问题？

```
ALTER SESSION ENABLE PARALLEL DML
/  
DELETE FROM table1  
  WHERE local = 'BEIJING'  
/  
COMMIT  
/
```

保留的  
记录

```
CREATE TABLE table1_new  
  NOLOGGING  
  PARALLEL  
  COMPRESS FOR ALL operations  
as  
SELECT * FROM table1  
  WHERE local != 'BEIJING'  
/  
ALTER TABLE table1 RENAME TO table1_old  
/  
ALTER TABLE table1_new RENAME TO table1  
/
```



## (2/6)数据处理：转变思维，DML重写 (*UPDATE*)

### □ 常见的DML场景：update

如果是大量数据的update，可能会出现什么问题？

```
ALTER SESSION ENABLE PARALLEL DML
/  
UPDATE table1  
    set price = 12  
    WHERE price = 10  
        and sdate > '10-Apr-14'  
/  
COMMIT  
/
```

```
CREATE TABLE table1_new  
    NOLOGGING  
    PARALLEL  
    COMPRESS FOR ALL operations  
as  
SELECT  
    ... ,  
    case  
        price = 10  
        and sdate > '10-Apr-14'  
    then  
        12  
    else  
        price  
    end,  
    ...  
FROM table1  
/  
ALTER TABLE table1 RENAME TO table1_old  
/  
ALTER TABLE table1_new RENAME TO table1  
/
```



## (2/6)数据处理：数据转换与修改两种方式的比较

影响因素	转换方式	修改方式
• 压缩	没有影响	压缩效果可能下降 因压缩可能引起性能问题
• 碎片	没有	碎片、行迁移都很有可能发生
• Logging and UNDO	没有、或很少	较多，影响性能
• 索引	需要重建	索引自动维护 产生索引碎片 位图索引需重建
• 元数据	表权限等需要重新定义	没有影响
• 空间需求	2倍的数据空间	UNDO and Logging
• 代码	需要重写代码	传统代码不变，但可能遭遇性能问题
• 其它	第三方ETL工具可能不支持	

大量数据的修改，建议通过转换和表重定义而不是传统的 OLTP DML 方式

- ✓ 这样才能发挥硬件和并行处理能力
- ✓ 最小化数据碎片问题同时最大化压缩的效果
- ✓ 最小化日志和数据恢复问题



## (3/6)索引使用：这是一个值得探讨的问题

在Exadata环境中，索引使用成为一个有争议且值得探讨的问题

- ❑ 在OLTP系统中，索引经常是提升性能的利器
- ❑ 在大数据集合下，索引维护成本很高(碎片、重建…)
- ❑ 索引对批量数据的DML操作，影响性能

索引扫描，通常会产生随机的IO读取操作(不可预估，与查询条件有关)。假设I/O带宽从200MB/S提升到40GB/S，你是否依然坚定的选择索引扫描？



# (3/6)索引使用：真的需要使用索引吗？

我们来看一个真实的案例：(注：已采集相应表的最新统计信息)

```
SQL> select count(*) from BUFBSG where datetime=to_date('2014-03-22 03:30:00');
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	8	32621 (1)	00:06:32
1	SORT AGGREGATE		1	8		
* 2	INDEX RANGE SCAN	IDX_BUFBSG	12M	93M	32621 (1)	00:06:32

Statistics

48203 consistent gets  
48203 physical reads

CBO聪明的选择了索引扫描，从执行计划来看，成本更低。

```
SQL> alter session set "_serial_direct_read" = true;
```

```
SQL> select /*+ full(BUFBSG) */ count(*) from BUFBSG where datetime=to_date('2014-03-22 03:30:00');
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	8	740K (1)	02:28:06
1	SORT AGGREGATE		1	8		
* 2	TABLE ACCESS STORAGE FULL	BUFBSG	12M	93M	740K (1)	02:28:06

Statistics

2725896 consistent gets  
2725865 physical reads





# (3/6)索引使用：真的需要使用索引吗？

但事实却是这样的：

```
SQL> select count(*) from BUFBSG where datatime=to_date('2014-03-22 03:30:00');
```

**Elapsed: 00:00:22.01**

```
SQL> alter session set "_serial_direct_read" = true;
```

```
SQL> select /*+ full(BUFBSG) */ count(*) from BUFBSG where datatime=to_date('2014-03-22 03:30:00');
```

**Elapsed: 00:00:03.49**

但实际情况，  
全表扫描更快。

Exadata智能扫描，过滤掉40多GB的数据，实际传到DB节点的数据只有474MB，再加上InfiniBand网络的优势，全表扫描完胜索引扫描。

```
SQL> SELECT IO_CELL_OFFLOAD_ELIGIBLE_BYTES / 1024 / 1024, IO_INTERCONNECT_BYTES / 1024 / 1024,  
           PHYSICAL_READ_BYTES / 1024 / 1024, IO_CELL_OFFLOAD_RETURNED_BYTES / 1024 / 1024  
FROM V$SQL  
WHERE PLAN_HASH_VALUE = 1615168263;
```

CELL_OFFLOAD_ELIGIBLE_BYTES	IO_INTERCONNECT_BYTE	PHYSICAL_READ_BYTES	IO_CELL_OFFLOAD_RETURNED_BYTES
42591.6406	474.503487	42591.6406	474.503487



## (3/6)索引使用：总结及其它的一些因素考虑

### □ 总结：

1、当无法预知多少记录会被读取时，索引扫描是否会失控？全表扫描呢？

索引的优势是在读取少量记录时，更适合OLTP系统。

2、删除不必要的索引有助于提高DML操作的性能并节省存储空间，如何判断索引存在是否合理？

技术定位：DISTINCT\_KEYS

```
ALTER INDEX <index_name> INVISIBLE;
```

业务惯性：谓词访问频率

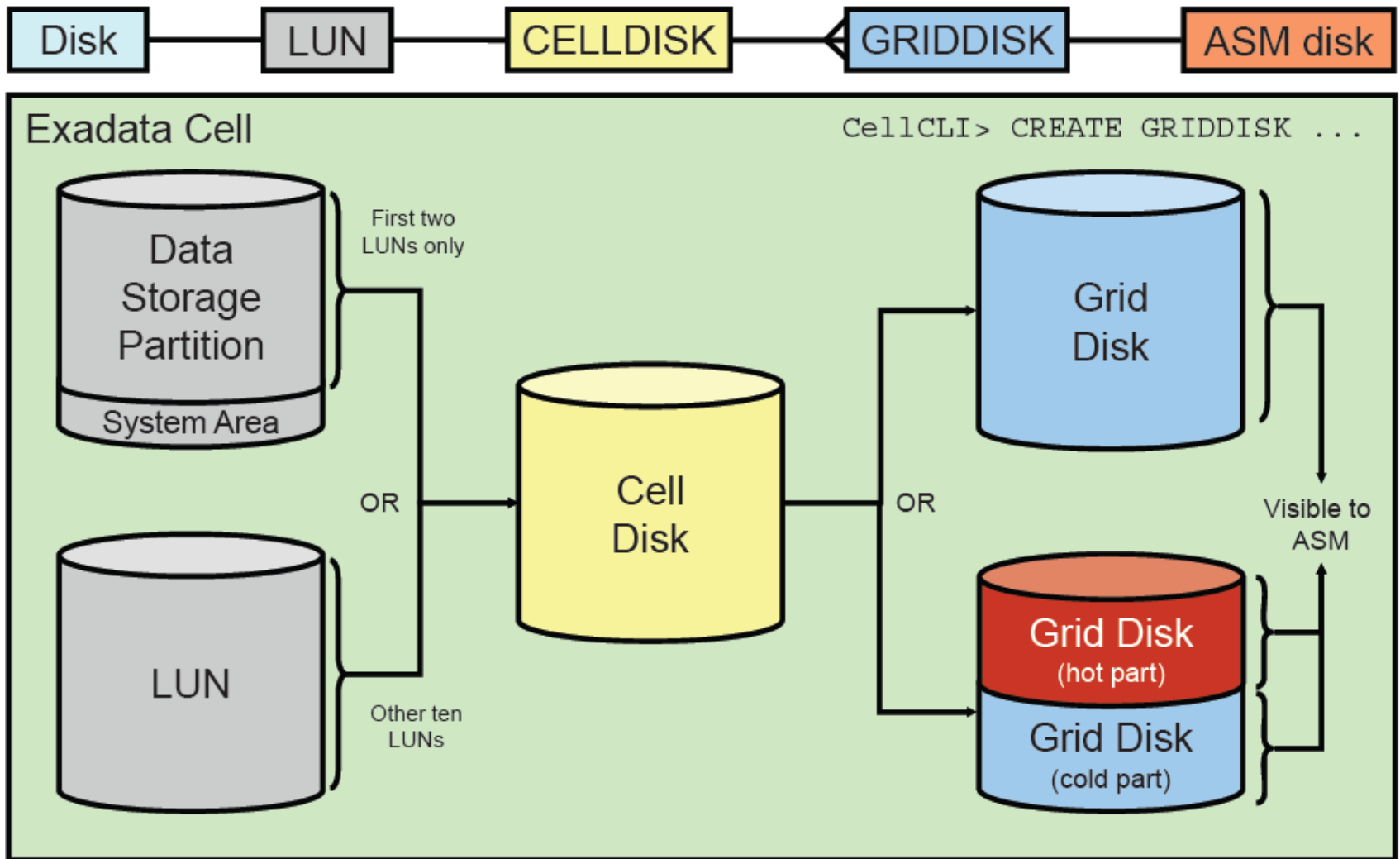
OLAP的频繁查询通常发生在小表上

### □ 其它因素考虑：

- 添加索引可加速某些场景，但可能会给别的场景带来负面影响。
- 添加索引，给优化器提供更多的可能性，可能导致选择到非最优的执行计划。



# (4/6)存储使用：Exadata存储的分配过程了解





# (4/6)存储使用：巧用DBFS，解决外部表等空间问题

- 外表部 or SQL\*Loader

- 当计算节点本地存储空间不足时，
- 当外部表需要被RAC多节点使用时，
- 外部数据文件存放到哪里？

- 数据泵的导出/导入

- 本地存储空间不足，DMP文件放哪里？

- 其它文件临时存放

- 本地存储空间不足，怎么办？？

**数据仓库中旺盛的空间需求！**

## DBFS: Database File System

存放的文件像任何Oracle数据一样受到保护：

ASM镜像、DataGuard、闪回，等等。

### □ 步骤参考：

1. 创建用于 DBFS 存储的大文件表空间(BIGFILE TABLESPACE)
2. 在数据库实例上创建一个DBFS用户帐户 (推荐使用单独数据库)
3. 把将要挂载 DBFS 文件系统的操作系统用户添加到 fuse 组
4. 以 root 身份创建 /etc/fuse.conf配置文件
5. 为 DBFS 创建一个挂载点，将所有权和组权限设置为 将要挂载 DBFS 文件系统的用户
6. @dbfs\_create\_filesystem\_advanced.sql，创建DBFS；dbfs\_client，挂载DBFS

### □ 案例分享：

为某电网客户提供和实施数据库备份方案，在Exadata上使用DBFS，巧用ASM上的剩余空间。



# (5/6)优化案例：关于数据仓库的一些优化建议

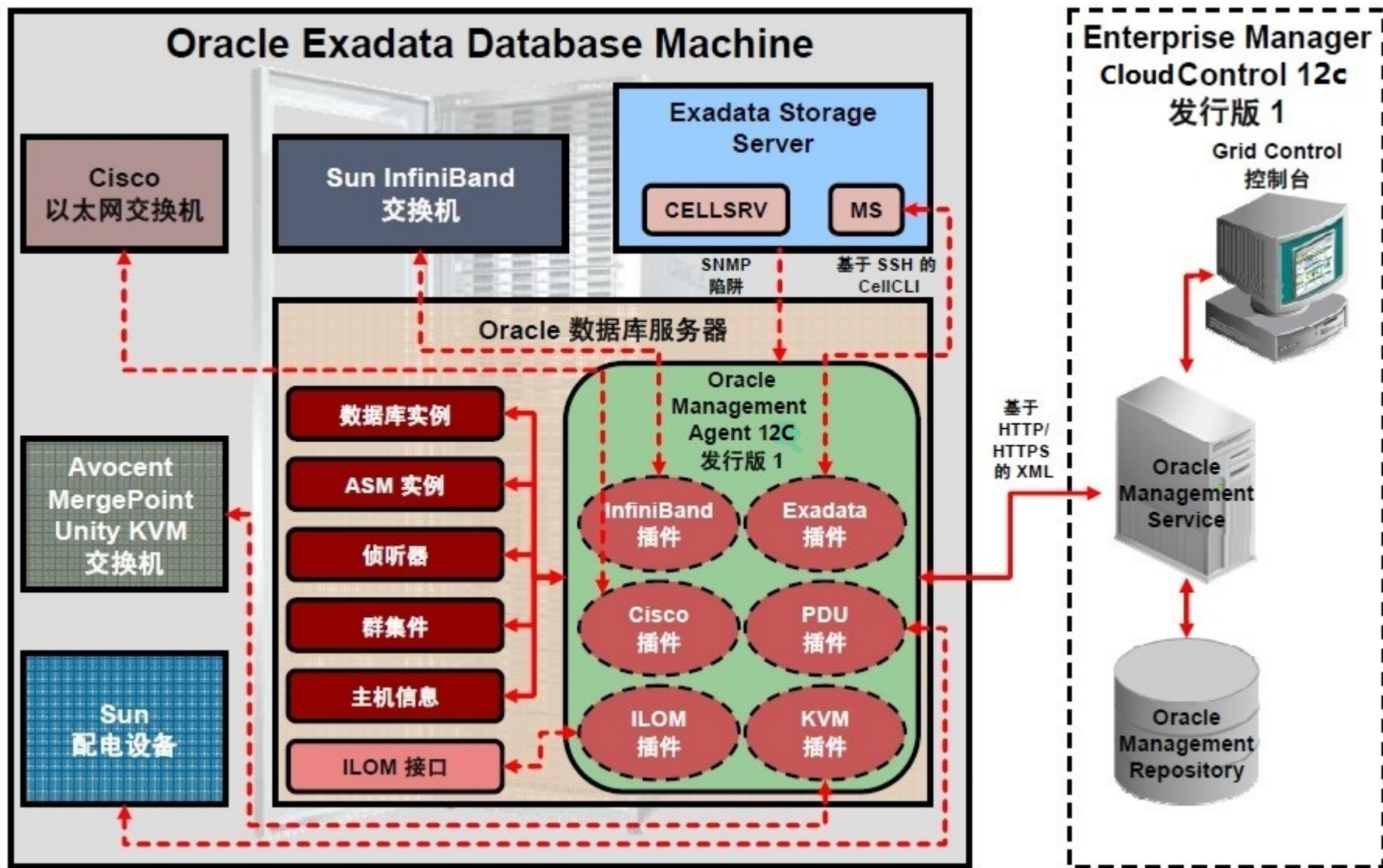
## ➤ 某省运营商数据仓库的优化案例分享

注：除压缩、分区、并行、索引优化等常见方法外

- **配置更大的PGA**
- **\_smm\_auto\_min\_io\_size/smm\_auto\_max\_io\_size** :  
*Maximum/minimum IO size (in KB) used by sort/hash-join in auto mode*  
增大每次hash的内存分配大小，提升group by的SQL性能
- **表空间使用uniform的Extent Allocation方式，避免表空间碎片**  
create tablespace tbs ... uniform size 8M ...;
- **热点表keep到FlashCache**  
alter table t storage (cell\_flash\_cache keep);



# (6/6)EM12C监控：Exadata目前最好的监控工具







# (6/6)EM12C监控：面板直观展示各组件状态、温度

半配的Exadata，  
7个存储节点，4  
个计算节点，3个  
Infiniband交换机

KVM状态正常

Cisco以太网交换机  
状态正常

7个CELL节点，状  
态正常，温度  
23°~25°

Infiniband交换机  
状态正常，温度  
24°、29°，级联的  
温度为24°

4个计算节点，其  
中2个没启动。2个  
状态正常，温度  
19°~20°

Component	Status	Temperature
dm01sw-ib3	Up	24°C
dm01sw-kvm	Up	
dm01sw-ip	Up	
dm01sw-ib2	Up	29°C
dm01db04	Up	
dm01db03	Up	
dm01db02	Up	19°C
dm01db01	Up	20°C
dm01cel07	Up	24°C
dm01cel06	Up	25°C
dm01cel05	Up	24°C
dm01cel04	Up	24°C
dm01cel03	Up	23°C
dm01cel02	Up	23°C
dm01cel01	Up	23°C
dm01sw-ib1	Up	22°C

**Legend**

- Up (Green dot)
- Down (Red dot)
- Blackout (Black)
- Exadata Cell (Dark Gray)
- Compute Node (Light Blue)
- Infiniband Switch (Medium Gray)
- Ethernet Switch (White)



# (6/6)EM12C监控：计算节点面板展示

**概要**

状态

所有者 SYSMAN

生命周期状态 -

引导时间 18-五月-2013 11:58:35

**诊断**

意外事件 0 0

配置更改 1

关键补丁程序指导 0

**配置**

**一般信息**

IP 地址 192.168.254.4

操作系统 Oracle Linux Server release 5.6

文件系统 (CFS) 402.57

内存

磁盘组

ACFS

**型号**

型号名

CPU 实

**处理器**

**作业活动**

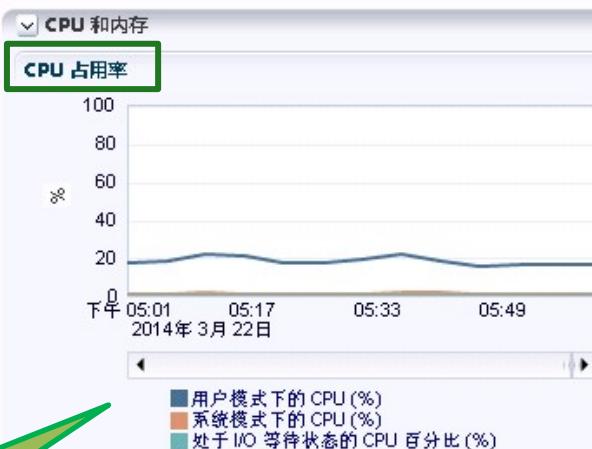
最近 7 天之内开始的作业的概要。

显示 最新运行

查看 状态 1 0 0 0

名称	状态	启动
DBM_ECM_DB_MACHI		2014-3-17 2:

实时掌握计算节点CPU、内存、文件系统使用、网络流量，是否发生过事故等情况。



表视图

表视图

表视图

拼



# (6/6)EM12C监控：存储节点面板展示

Exadata Grid dm01.

Exadata Storage Server Grid

页面刷新于：2014-3-22 18:13:55 CST

Cell性能柱状图

性能



概览

状态 ↑ (7 ↑)

健康状况 (4 ✓, 3 ✗)

IORM 状态 已启用

发行版本 11.2.3.2.0

容量

单元磁盘大小 (GB) 53510 100%

硬盘大小 (GB) 46452 100%

闪存磁盘大小 (GB) 7058 100%

闪存高速缓存大小 (GB) 7049 100%

按数据库统计的工作量分配









当前各数据库IO占用率

ASM 磁盘组概要

ASM	磁盘组	大小 (GB)	空闲空间 (GB)	网格磁盘数	脱机磁盘数	数据库名
+ASM_dm01-cluster	DATA_DM01	35532	1917.633	84	0	OLAP,ASM,MVIEW,OLTP
+ASM_dm01-cluster	DBFS_DG	2038	1614.555	70	0	DBFSDB
+ASM_dm01-cluster	RECO_DM01	8877	4934.852	84	0	OLTP,OLAP,DBM,MVIEW

ASM diskgroup容量信息

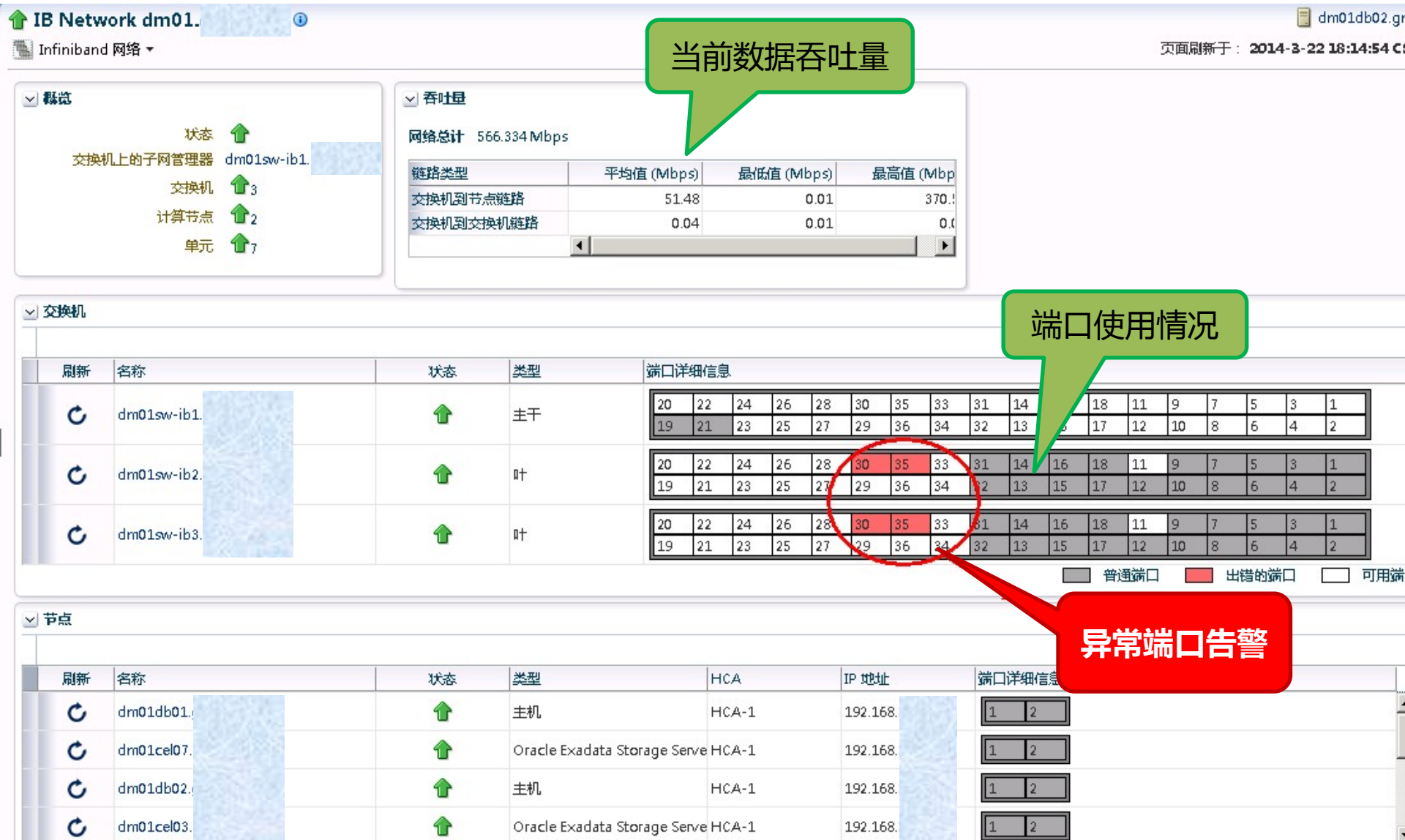
意外事件

查看	目标	本地目标和相关目标	类别	全部	0	3	0	0
概要					目标	严重性	状态	升级
File system "/" is 97% full, which is above the 80% threshold. Accelerated space reclamation has sta							新建	-
File system "/" is 97% full, which is above the 80% threshold. Accelerated space reclamation has sta							新建	-
File system "/" is 97% full, which is above the 80% threshold. Accelerated space reclamation has sta							新建	-

异常告警关注



# (6/6)EM12C监控：Infiniband交换机状态展示





# (6/6)EM12C监控：Cisco以太网交换机状态展示

dm01sw-ip-  
Cisco 以太网交换机

dm01db02.  
页面刷新于：2014-3-22 18:12:01 CST

dm01sw-ip.gmcc.net > 所有度量  
所有度量

搜索

查看

dm01sw-ip-  
CPU  
内存  
内存池使用率 (%)  
响应  
温度  
电源  
系统信息  
网络接口  
发送丢弃率 (%)  
发送通信量 (%)  
发送通信量 (Kb/秒)  
发送错误率 (%)  
接口状态  
接收丢弃率 (%)  
接收通信量 (%)  
接收通信量 (Kb/秒)  
接收错误率 (%)  
操作状态  
管理状态  
风扇

网络接口  
收集调度 每5分钟 修改  
上载间隔 每次收集  
上次上载 2014-3-22 18:04:20 CST

接口名称	网络接口别名	带宽 (Mb/秒)	操作状态	发送错误率 (%)	发送丢弃率 (%)	发送通信量 (%)	发送通信量 (Kb/秒)
GigabitEthernet1/21		1000	Down	0	0	0	0
GigabitEthernet1/22		1000	Down	0	0	0	0
GigabitEthernet1/23		1000	Down	0	0	0	0
GigabitEthernet1/24		100	Up	0	0	1	1
GigabitEthernet1/25		1000	Up	0	0	1	1
GigabitEthernet1/26		100	Up	0	0	1	1
GigabitEthernet1/27		1000	Up	0	0	1	6
GigabitEthernet1/28		100	Up	0	0	1	2
GigabitEthernet1/29		1000	Up	0	0	1	73
GigabitEthernet1/30		100	Up	0	0	1	2
GigabitEthernet1/31		1000	Up	0	0	1	2
GigabitEthernet1/32		100	Up	0	0	1	1
GigabitEthernet1/33		1000	Up	0	0	1	2
GigabitEthernet1/34		100	Up	0	0	1	1
GigabitEthernet1/35		1000	Up	0	0	1	2

上表中显示的数据是实时收集的。

网络接口状态





# (6/6)EM12C监控：数据库运行负载展示

↑ olap ⓘ

dm01db01

集群数据库 ▾ 性能 ▾ 可用性 ▾ 安全性 ▾ 方案 ▾ 管理 ▾

页面刷新于：2014-3-22 18:13:07 CST

自动刷新 禁用

概要

状态

实例 ↑ 2

运行时间 26天, 7小时

版本 11.2.0.2.0

加载 4.41 平均活动会话数

会话总数 210

上次备份 N/A

可用空间 1,613.62 GB

已用空间 7,457.32 GB

SGA 总容量 14,271.98 MB

诊断

最新的全局 ADDM 查找结果 6

兼容性概要 (简要)

兼容性标准 成员

查看趋势

名称	平均分数
没有可显示的数据	

正在运行的作业

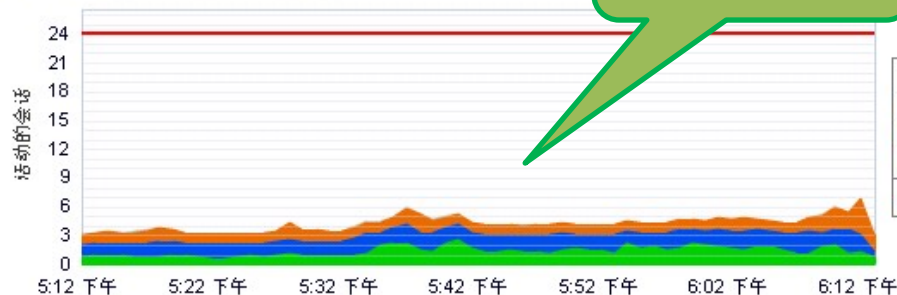
名称	持续时间 (分钟)
ORA\$AT_SA_SPC_SY_15578	733.2
ORA\$AT_OS_OPT_SY_15577	732.9

性能

活动类

服务

实例



资源

SQL 监视 - 过去一小时

状态	持续时间	SQL ID	会话 ID	实例	并行	数据库时间
🟢	1.00 秒	3rx14rd5qbwcr	1241	olap_1	2	0.4
🟢	13.00 秒	6jg7hwjt0jky1	334	olap_2		11.
🟢	2468.00 秒	0s1yrwn1byws4	1241	olap_1		224.
🟢	2.00 秒	atg33krfwr7jz	1241	olap_1	2	4.7
🟢	10.00 秒	8ydgbfjw4a60r	1241	olap_1		9.0
🟢	7.00 秒	8szmwam7fyas3	1241	olap_2		6.5
🟢	35.00 秒	g8d9qc57f4c6w	334	olap_2		34.
🟢	75.00 秒	ggt2xpa7baxb	1241	olap_1	2	85.1
🟢	6.00 秒	8szmwam7fyas3	136	olap_2		5.4
🟢	6.00 秒	8szmwam7fyas3	136	olap_2		5.8

正在运行的  
JOB

TOP SQL 监控





# (6/6)EM12C监控：部署建议

在部署EM12C的OMS和Exadata Discovery的过程中可能会遇到较多的问题

- **EM12C版本要求**

监控Exadata，EM12C最低版本要求为12.1.0.2

- **Agent部署方式**

建议Automation Kit for Exadata的方式安装Agent，可减少Exadata Discovery过程出错的概率

参考：MOS Note 1440951.1

Oracle Exadata Discovery Cookbook

- **可能遭遇的bug**

**Bug 16003324**：Cell节点Agent状态正常，但是EM12C上无数据显示，检查cell节点

CellCLI> list celldisk

CELL-02620: An unmapped CELLSRV error has occurred. The internal message is: msosscommerr#3.

触发bug 16003324，需要应用Patch 16042459、或者重启管理服务解决

CellCLI> [alter cell restart services ms](#)

**Bug 13574842**：GETTING METRIC COLLECTION ERROR “COLLECTION RESULT MAXIMUM FLOOD CONTROL LEVEL，导致ASM diskgroup信息无法显示。修改EM Agent的flood control参数([Note 1499381.1](#))：  
CollectionResults.MaximumRowsFloodControlMin  
CollectionResults.MaximumRowsFloodControlMax

# 更多精彩，与您分享

- 3月29日 上海站
- 4月26日 合肥站
- 5月上旬 福州站
- .....

**新炬** 2014

全国技术沙龙

新炬ITPUB全国巡回DBA技术沙龙

