DTCC

2016中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2016

数据定义未来

SequeMedia 盛拓传媒 IT168.com ChinaUnix ITPUB

从传统银行到互联网金融

——Oracle数据库架构设计实践

个人介绍

- 杨廷琨(yangtingkun)
  - Oracle ACE Director
  - ITPUB数据库管理区版主
  - ACOUG核心会员
  - 参与编写《Oracle数据库性能优化》、《Oracle DBA手记》和《Oracle DBA手记3》
  - 十六年的一线DBA经验
  - 个人BLOG中积累了2500篇原创技术文章
  - 云和恩墨CTO

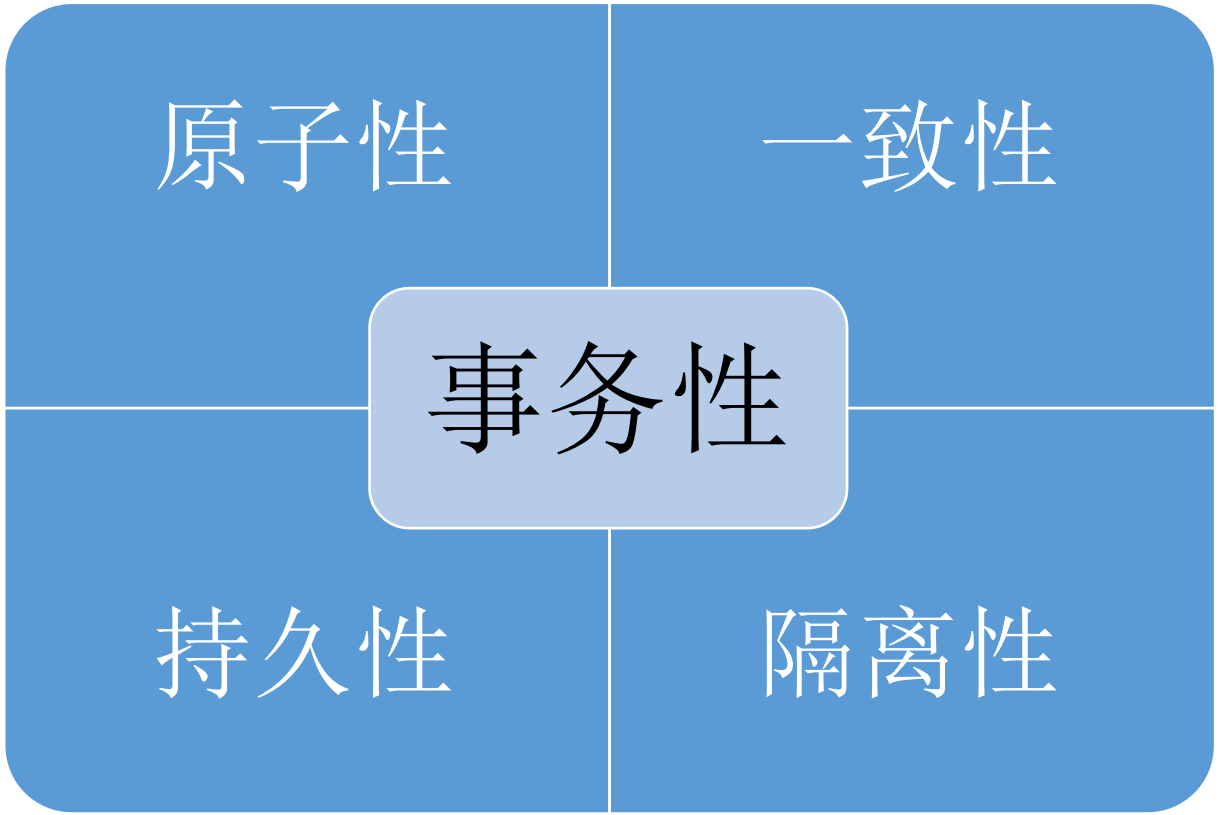
目录

- 金融行业数据库要求
- 传统金融行业
- 互联网金融
- 数据库架构设计

金融行业数据库要求

- 数据安全——事务性
- 系统可用——高可用
- 客户感知——高性能
- 容量增长——扩展性

关系型数据库



高可用系统

双机	• HA • DataGuard
集群	• RAC
容灾	• DataGuard • 存储复制

性能设计

硬件

- CPU
- MEMORY
- IO

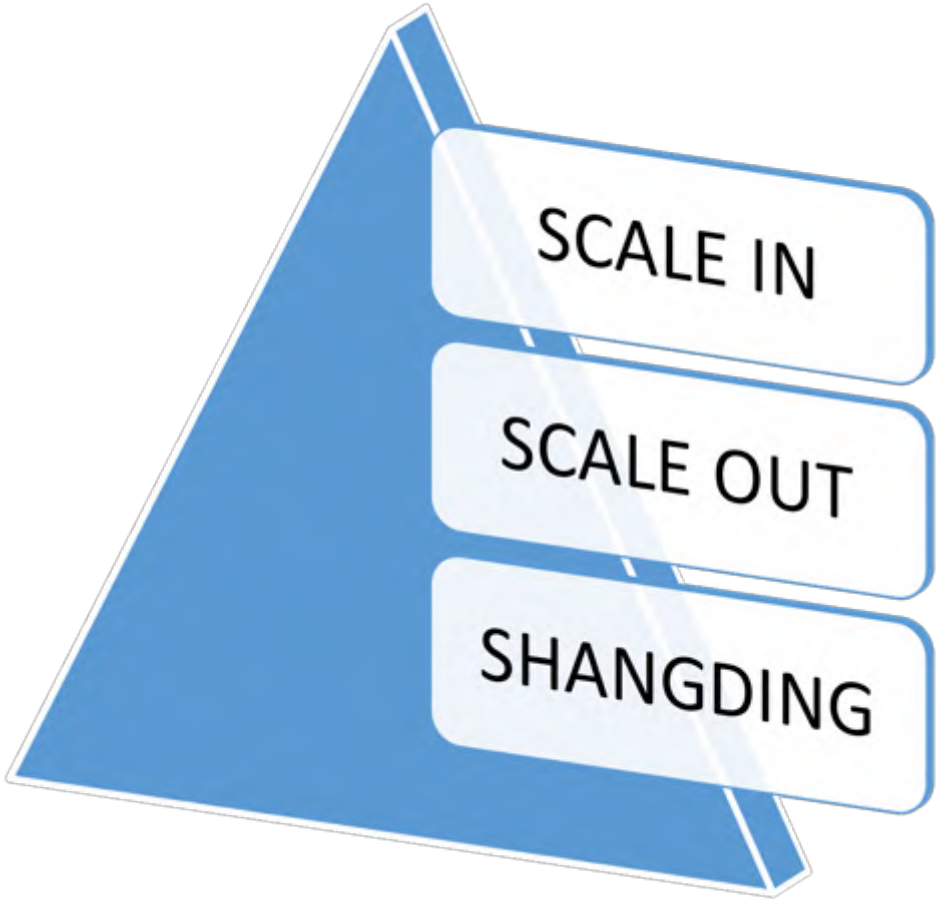
建模

- 物理
- 逻辑

开发

- 登录
- SQL

数据库扩展性



目录

- 金融行业数据库要求
- 传统金融行业
- 互联网金融
- 数据库架构设计

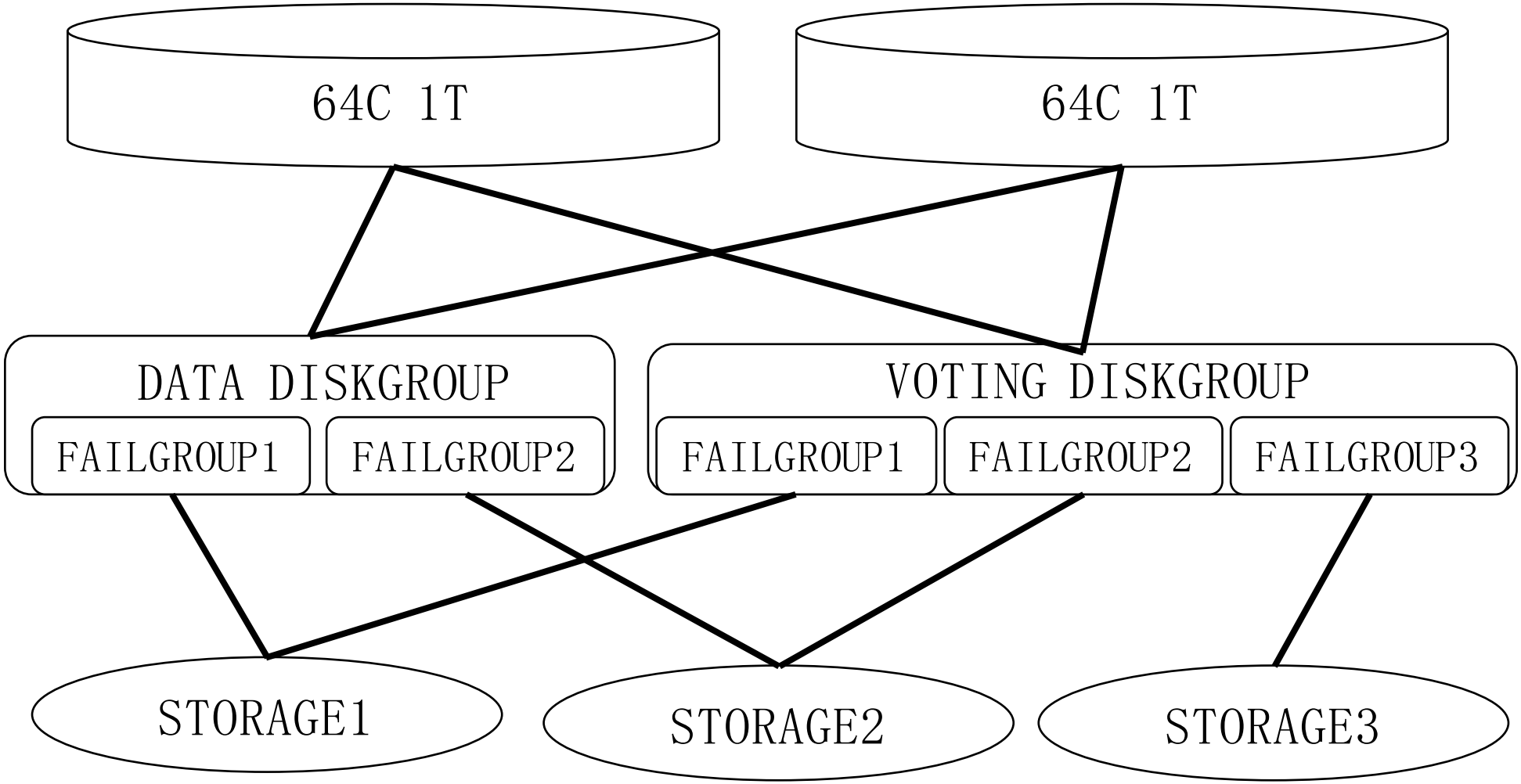
金融行业数据库要求

- 数据安全——事务性
- 系统可用——高可用
- 客户感知——高性能
- 容量增长——扩展性

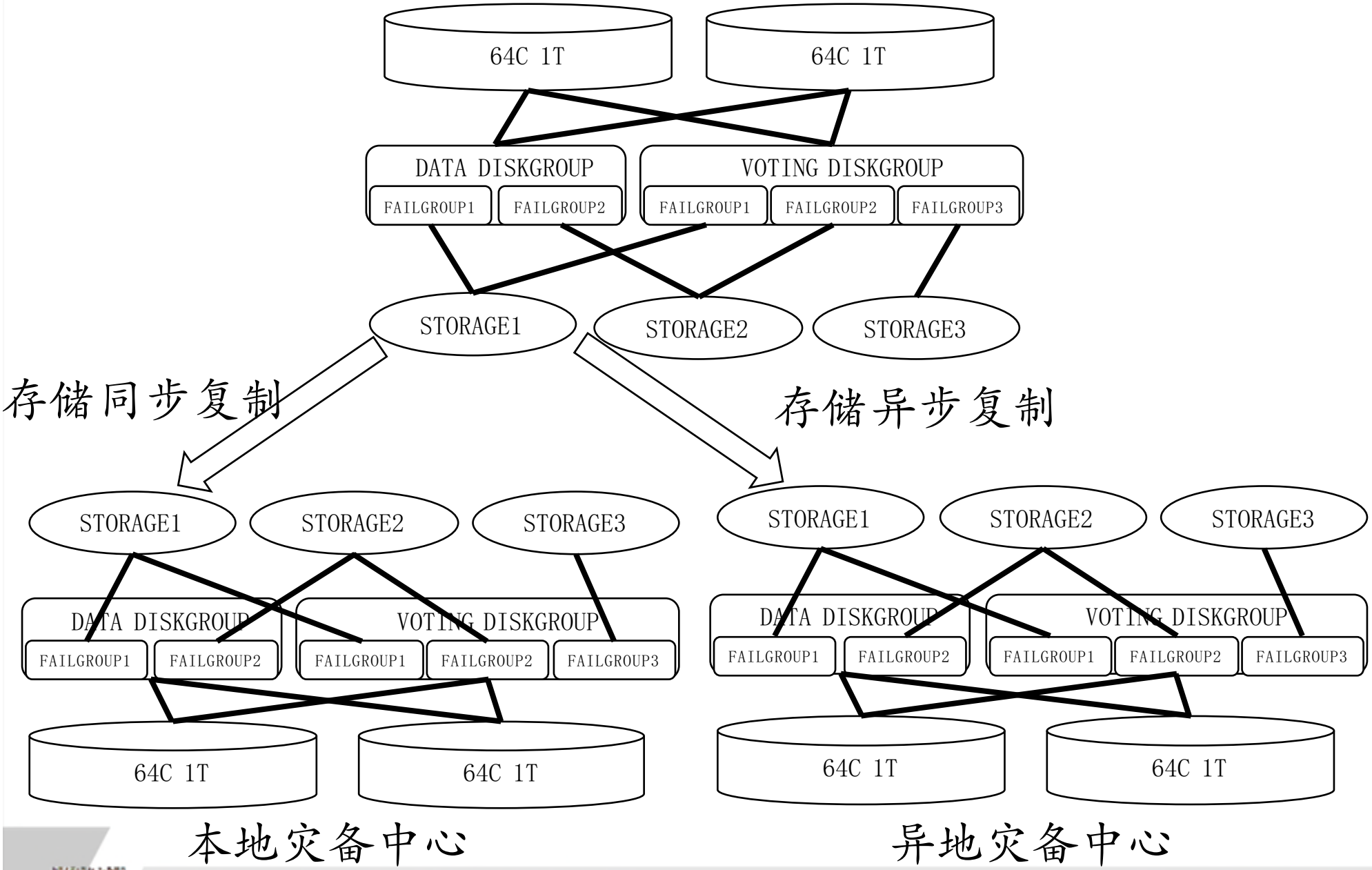
传统金融行业高可用架构案例

- 主机64核1T内存
- 11.2.0.3 两节点RAC
- ASM NORMAL冗余
- 每个FAILOVER GROUP对应独立存储
- VOTING DISK第三块盘存放在单独第三个存储上
- 存储同步复制到本地第二机房
- 存储异步复制到异地数据中心
- 2颗CPU绑定到LGWR进程
- CacheLUN技术将REDO映射到存储内存

传统金融行业高可用架构案例



传统金融行业高可用架构案例



传统金融行业高可用架构案例

Platform	CPUs	Cores	Sockets	Memory (GB)
HP-UX IA (64-bit)	62	62	16	1023.21

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	35976	02-9月 -14 07:00:22	4384	5.8
End Snap:	35991	02-9月 -14 12:00:18	4392	5.8
Elapsed:		299.94 (mins)		
DB Time:		3,425.69 (mins)		

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
log file sync	24,298,813	96,728	4	47.06	Commit
DB CPU		79,401		38.63	
gc current block 2-way	11,621,914	12,780	1	6.22	Cluster
db file sequential read	907,707	6,682	7	3.25	User I/O
gc cr block 2-way	2,278,022	2,076	1	1.01	Cluster

传统金融行业高可用架构案例

Snap Time	Load	%busy	%user	%sys	%idle	%iowait
02-9月 07:00:22	0.29					
02-9月 07:20:28	0.32	26.43	21.11	5.32	73.57	0.14
02-9月 07:40:38	0.36	29.16	23.37	5.79	70.84	0.16
02-9月 08:00:46	0.42	32.70	26.45	6.25	67.30	0.32
02-9月 08:20:53	0.50	38.35	31.38	6.97	61.65	0.42
02-9月 08:40:03	0.55	41.67	34.30	7.36	58.33	0.34
02-9月 09:00:10	0.60	45.27	37.61	7.66	54.73	0.26
02-9月 09:20:16	0.67	48.62	40.61	8.01	51.38	0.26
02-9月 09:40:24	0.70	52.86	44.30	8.56	47.14	0.26
02-9月 10:00:32	0.72	51.63	43.29	8.34	48.37	0.28
02-9月 10:20:41	0.69	52.05	43.51	8.55	47.95	0.27
02-9月 10:40:10	0.67	50.83	42.21	8.62	49.17	0.28
02-9月 11:00:21	0.68	51.93	43.19	8.73	48.07	0.27
02-9月 11:20:05	0.67	49.45	40.81	8.63	50.55	0.30
02-9月 11:40:10	0.59	46.92	38.84	8.09	53.08	0.30
02-9月 12:00:18	0.59	46.06	38.13	7.93	53.94	0.41

Buffer Nowai	99.99
Buffer Hit %:	100.00
Library Hit %	98.47
Execute to P	99.60
Parse CPU t	99.21

传统金融行业高可用架构案例

Background Wait Events

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn	% bg time
log file parallel write	7,922,482	0	15,731	2	0.33	40.56
db file parallel write	3,456,187	0	8,566	2	0.14	22.09
gcs log flush sync	1,187,491	1	1,368	1	0.05	3.53
reliable message	3,403	8	401	118	0.00	1.03
Log archive I/O	40,593	0	160	4	0.00	0.41

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	35976	02-9月 -14 07:00:22	4384	5.8
End Snap:	35991	02-9月 -14 12:00:18	4392	5.8
Elapsed:		299.94 (mins)		
DB Time:		3,425.69 (mins)		

```
SQL> select 3600*5 from dual;
     3600*5
----------
      18000
```

传统金融行业高可用架构案例

- 提高存储响应，降低单次LOG FILE PARALLEL WRITE等待时间；
- 业务端合并短小事务，降低每秒处理的事务数量；
- 初始化参数COMMIT_WAIT=nowait；
- 增加实例，分担单个实例每秒处理事务数量；
- 数据库级切分，增加新的数据库来分担每秒处理事务数量。
- 数据库版本升级到12.1
- 存储同步复制改为异步复制

传统金融行业高可用架构案例

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	36048	03-9月 -14 07:00:30	4393	5.8
End Snap:	36063	03-9月 -14 12:00:48	4402	5.8
Elapsed:		300.30 (mins)		
DB Time:		2,642.16 (mins)		

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		76,590		48.31	
log file sync	23,949,904	46,365	2	29.25	Commit
gc current block 2-way	11,665,593	15,464	1	9.75	Cluster
db file sequential read	1,611,454	8,906	6	5.62	User I/O
gc cr block 2-way	2,328,981	2,447	1	1.54	Cluster

传统金融行业高可用架构案例

Background Wait Events

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn	% bg time
log file parallel write	7,922,482	0	15,731	2	0.33	40.56
db file parallel write	3,456,187	0	8,566	2	0.14	22.09
gcs log flush sync	1,187,491	1	1,368	1	0.05	3.53
reliable message	3,403	8	401	118	0.00	1.03
Log archive I/O	40,593	0	160	4	0.00	0.41

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn	% bg time
log file parallel write	18,294,289	0	6,842	0	0.76	26.77
db file parallel write	3,500,665	0	2,799	1	0.15	10.95
gcs log flush sync	580,316	2	603	1	0.02	2.36
reliable message	3,622	12	536	148	0.00	2.10
Log archive I/O	45,838	0	181	4	0.00	0.71

目录

- 金融行业数据库要求
- 传统金融行业
- 互联网金融
- 数据库架构设计

金融行业数据库要求

- 数据安全——事务性
- 系统可用——高可用
- 客户感知——高性能
- 容量增长——扩展性

互联网金融爆发性业务增长案例

- 背景
  - 一年业务增长超过100倍
  - 预计在今后一年内，业务量仍然会有数十倍增长
- 架构
  - Exadata 1/8一体机
  - 已采购Exadata X5满配一体机
- 困惑
  - 当前数据库架构所能支撑的业务极限

互联网金融爆发性业务增长案例

CPU资源主要消费			
逻辑读	硬解析	闩锁	无效或低效SQL

现有环境CPU资源	
1/8配Exadata 2节点18CPU36核	满配Exadata 8节点36CPU72核

扩展性			
1/8配单节点可承载45倍业务	满配单节点可承载90倍业务	可通过多节点扩展CPU资源	SQL优化前提下CPU不存在瓶颈

互联网金融爆发性业务增长案例

IO资源主要消费			
物理读	数据文件写	日志文件写	备份

现有环境IO能力	
1/8配Exadata IOPS 1866	满配Exadata IOPS 24487

扩展性			
1/8配可承载10倍业务	满配可承载100倍业务	可配置FLASH提高IO性能	配置FLASH存储IO不存在瓶颈

互联网金融爆发性业务增长案例

心跳网络资源主要消费

节点间心跳	GC数据传输	数据库后台进程通信

现有环境网络带宽

Exadata Infiband 40G网络带宽

扩展性

可承载7000倍业务	多节点承载业务心跳压力会成倍增长	心跳网络不存在瓶颈

互联网金融爆发性业务增长案例

LGWR时间消耗

log file parallel write	CPU调度	远程节点通信

当前配置下资源消耗

LGWR进程目前使用率6%

扩展性

单节点可承载15倍业务	升级到满配不带来扩展性提升	可通过部署FLASH提升性能	可通过多节点业务进行扩展

互联网金融爆发性业务增长案例

1/8配

- IO为主要瓶颈
- 可承载业务压力10倍
- 通过配置FLASH可提高IO处理能力

满配

- LGWR单进程为主要瓶颈
- 可承载业务压力15倍
- 通过配置FLASH可提高LGWR处理效率

多节点

- 单个数据库的锁、热点等内部机制为主要瓶颈
- 可承载业务50到100倍
- 需要对现有业务模块改造，避免多个实例间的争用和冲突

Sharding

- 可承载业务压力超过100倍，具备良好的扩展性
- 需要对现有业务架构进行彻底重构

目录

- 金融行业数据库要求
- 传统金融行业
- 互联网金融
- 数据库架构设计

数据库架构设计

- 明确业务增长预期并转换为数据库指标
- 深入分析发现当前系统瓶颈
- 综合考虑可用性、性能和扩展性
- 根据业务预期和瓶颈进行容量预测

THANKS

SequeMedia 盛拓传媒 IT168.com ChinaUnix ITPUB