# 淘宝
# 商品库 MySQL 优化实践
# QCon 2011 Beijing

核心系统数据库组
余锋（褚霸）
http://yufeng.info
2011/04/08

# Agenda

商品库项目背景介绍以及约束
技术要求和方案
性能保证
安全性保证
运维保证
优化成果
交流时间

# 商品库(单机，测试)情况

无复杂查询，离散度高
记录数： 1 亿条键值对
记录大小： 100 字节
数据文件： 170G
访问热点情况： 20% 的键占用 55 %的访问量
键读写比例： 10 ： 1

# 硬件选择

主机： Dell; PowerEdge C2100;
处理器： physical = 2, cores = 12, virtual = 24
内存： 96 G
RAID 卡： LSI MegaSAS 9260/512MB Memory
PCI-E Flash 卡 : Fusion-io ioDrive 320GB/MLC
硬盘 : SEAGATE ST3300657SS 300G x 12

# 软件选择

发行版： Red Hat Enterprise Linux Server release 5.4
内核： Kernel | 2.6.18-164.el5
文件系统： Ext3
Flashcache: FB 内部版本
MySQL 版本： 5.1.48-log Source

# Agenda

商品库项目背景介绍以及约束
技术要求和方案
性能保证
安全性保证
运维保证
优化成果
交流时间

# 商品库技术要求

高可用，安全第一
高性能，性能平稳，性价比高
控制运维风险

# 技术方案

MySQL 数据库集群，数据水平切割，主从备份
采用高性价比 PC 服务器，大内存，强劲 CPU ，可靠性高
采用高性能 PCI-E Flash 卡作为 cache， 提高系统的 IO 性能
充分利用系统各部件的 cache，大胆采用新技术
充分考虑容灾，在各个层面考虑数据的安全性

# 系统资源规划

内存分配：
- MySQL
- InnoDB buffer pool
- OS pagecache
- 驱动程序

IO 能力分配：
- 读能力，零散读，提高 IOPS
- 写能力，集中写，提高吞吐量

Cache 分配：
- MySQL 内部 cache
- 匿名页面 / 文件页面
- Flashcache 混合存储
- Raid 卡内部 cache

# 调优指导思想

杜绝拍脑袋，理论（源码）指导 + 精确测量 + 效果验证
内存为王
数据访问规律导向，随机数据和顺序数据尽量分离
尽量提高 IO 的利用率，减少无谓的 IO 能力浪费
在安全性的前提下，尽可能的利用好系统各个层次 cache

# 调优工具

源码 +emacs+ 大脑
必备工具
- systemtap
- oprofile
- latencytop
- blktrace/btt/seekwatcher
- aspersa
- tcprstat
- sar
- gdb

自制工具
- bash 脚本
- gnuplot 脚本

# Agenda

商品库项目背景介绍以及约束
技术要求和方案
性能保证
安全性保证
运维保证
优化成果
交流时间

# MySQL 数据库

考虑因素：

主从备份带来的性能影响

复杂数据查询操作是否需要预留内存以及上限

数据备份 dump 对系统的影响，避免系统 swap

开启 binlog 带来的性能开销

限制最大链接数

```
#####################################
max_binlog_cache_size=2G
max_binlog_size = 500M
max_connections = 1020
max_user_connections=1000
query_cache_size = 30M
```

# InnoDB 引擎

考虑因素：

- 尽可能大的 BP(buffer pool)
- 日志和数据分设备存储
- 离散数据走 direct-IO ，顺序日志走 buffered-IO
- 减少脏页的同步，提高命中率
- 减少锁对多核 CPU 性能的影响
- 提高底层存储默认的 IO 能力

```
#####################################
innodb_buffer_pool_size = 72G
innodb_flush_method = O_DIRECT
innodb_sync_spin_loops=0
innodb_log_group_home_dir = /u02/
innodb_io_capacity=2000
innodb_thread_concurrency = 64
```

# 高速页缓存

考虑因素：

page 资源倾斜给数据库，尽量不浪费，兼顾临时内存申请

避免 NUMA 架构带来的 zone 内存分配不均而导致的 swap 现象

cache 大部分由 InnoDB 日志产生，适时清除，限制 page 数量

```
#####################################
# numactl --interleave=all mysqld
# sysctl vm.drop_caches = 1

vm.swappiness = 0
vm.dirty_ratio =  ?
vm.dirty_background_ratio =  ?
vm.pagecache = ?
```

# 文件系统

考虑因素 ( 选择 ) :

- Ext3/4
- Xfs

考虑因素 ( 配置 ) :

- 减少元数据变化产生的 IO
- 对混合存储系统友好
- 关闭 barrier

```
#####################################
 /dev/mapper/cachedev  (rw,noatime,nodiratime,barrier=0) /u01
 /dev/sda12            (rw,barrier=0)                    /u02
```

# IO 调度

考虑因素：

- 调度算法对减少磁头移动的效果
- 关闭预读
- 设备队列长度

##################################
sda | [deadline]  128
sdb | [deadline]  128

# 混合存储（ Flashcache ）

考虑因素

- 结合磁盘的大容量， PCI-E Flash 卡的高随机读写性能优点
- 数据尽可能多停留在 PCI-E Flash 卡上，提高读写命中率
- 减少同步次数，保留磁盘的 IO 能力
- 适时同步数据，减少安全风险

```
#####################################
dev.flashcache.dirty_thresh_pct = 90
dev.flashcache.cache_all = 0
dev.flashcache.fast_remove = 1
dev.flashcache.reclaim_policy = 1
```

# Raid 卡

考虑因素：

- 逻辑分卷
- Cache 使用写优先，读少分配（数据无相关性效果不好）
- 数据安全和 raid level
- 少预读

```
#####################################
 Controller |  LSI Logic / Symbios Logic LSI MegaSAS 9260 (rev 03)
       Model | LSI MegaRAID SAS 9260-8i, PCIE interface, 8 ports
       Cache | 512MB Memory, BBU Present
         BBU | 95% Charged, Temperature 28C, isSOHGood=

  VirtualDev Size      RAID Level Disks SpnDpth Stripe Status  Cache
  0(no name) 278.875 GB 1 (1-0-0)      2     1-1      64 Optimal WB, RA
  1(no name) 1.361 TB  1 (1-0-0)      2     5-5      64 Optimal WB, RA
```

# 存储设备驱动

考虑因素：
- 减少 IO 的抖动，提高 IOPS
- 提高寿命
- 关闭或减少预读

##################################
PCI-E Flash 卡驱动：
$cat /etc/modprobe.d/iomemory-vsl.conf

options iomemory-vsl use_workqueue=0
options iomemory-vsl disable-msi=0
options iomemory-vsl use_large_pcie_rx_buffer=1

# 性能保证小结

解决 IO 瓶颈：

- 高速 PCI-E Flash 卡做 Cache ，读写速度可达 800/500M
- 10 x SAS 300G 存放离散度高数据文件
- 2 x SAS 300G 存放顺序 binlog 和 trx 日志
- 控制数据库脏页面的刷新频率和强度
- 优化操作系统的 pagecache ，资源倾斜，杜绝 swap 发生
- 优化文件系统减少 meta 数据的产生，以及写入延迟
- 优化 IO 调度器和预读
- 开启 raid 卡的读写 cache
- 优化设备驱动，适应高强度的读写请求，减少 jitter

解决 CPU 瓶颈：

- 业务上优化掉复杂查询
- 优化自旋锁

# Agenda

商品库项目背景介绍以及约束
技术要求和方案
性能保证
安全性保证
运维保证
优化成果
交流时间

# 安全性保证概要

Raid 卡带 Flash ，掉电保护， raid level10 防止磁盘损害
PCI-E 卡自身有日志系统，恢复时间最差 10 分钟
Ext3 文件系统带日志保护
Flashcache 上的 cache 数据最多 24 小时都会同步到 SAS 盘
数据库 Innodb 引擎本身有 redo 日志，数据安全校验，高级别日志同步
MySQL 主从备份
商品库应用方有事务日志

# Agenda

商品库项目背景介绍以及约束
技术要求和方案
性能保证
安全性保证
运维保证
优化成果
交流时间

# 运维保证概要

数据预热：

- 支持热点数据每秒 150M 从磁盘直接加载到混合存储
- 数据库重新启动，无需重新预热

数据库 DDL 操作：

- 控制数据表的大小，让 DDL 时间可接受
- 减少 DDL 对性能的冲击

混合存储 cache ：

- 通过设置白名单，减少诸如备份操作对 cache 的干扰
- 混合存储 cache 可管理

# Agenda

商品库项目背景介绍以及约束
技术要求和方案
性能保证
安全性保证
运维保证
优化成果
交流时间

# 优化成果

充足的容量规划，可对抗突增业务，满足未来几年业务增长
系统总体运行平稳，系统负载 CPU util <50% ，磁盘 util <10% ， PCI-E Flash 卡 util < 20%
QPS/36000 ，其中读 /32800 ，写 /3200
请求平均延时时间： 260 微秒 ( 包括网络时间 )
掉电和操作系统失效的情况下，数据无丢失
第一次预热时间半个小时以内，之后只需几分钟

# Agenda

商品库项目背景介绍以及约束
技术要求和方案
性能保证
安全性保证
运维保证
优化成果
交流时间

# 谢谢！

# Talents wanted!

联络： chuba@taobao.com