58同城的分布式存储架构实践

58同城技术中心架构部
徐振华
2012-04-07

- 分布式存储架构介绍和现状
- 需要那些基础知识
  - 提高资源利用率,做到线性扩展
  - 常用存储架构比较
- 如何设计分布式存储的架构
  - 分析需求，做好平衡
  - 如何做到RAS(可靠，可用，可扩展);
  - 利用硬件,分级存储
  - 58的分布式存储实践(mysql,mongodb,file system,hadoop)

介绍

- 重点： 架构
- 理论： CAP： Consistency Availability Partition tolerance 只能满足其二

  BASE :Basically Available（基本可用）Soft state（柔性状态）

  Eventually consistent（最终一致）

  ACID （原子性 一致性 隔离性 持久性）

  I/O五分钟法则

  Amdahl定律和Gustafson定律，摩尔定律

现状

Amazon 2011年，Amazon S3服务增加了5000亿份存储对象和文件

技术: SimpleDB ， Dynamo

Facebook目前全球累计已经有超过1400亿张照片发布在Facebook网站上，每天平均有超过2.5亿张照片上传至Facebook,Facebook目前存储的照片和视频数据量超过100PB(1PB=1024TB,HBase每月存储1350亿条信息

技术: Facebook图片存储系统 HayStack, cassandra，hbase

Google 2008年 谷歌网页索引数量突破1万亿

技术: Google Megastore , GFS, bigtable

Zynga Draw Something在发布前几周，完成了3千5百万的下载量，每日活跃用户更是达到1千5百万以上，每秒超过3000张的图画产生

技术: Couchbase vs EA The Simpsons 下架

基础知识

- 数据结构
- 网络
- 集群
- 操作系统
- 存储领域
- 其它领域

网络

引入：
C10K问题,C500K， C**K

服务器模型

s:1 , c:1 , bio; 一个请求一个线程

s:1 , c:n , nio ;多个请求，一个线程分发

seda :Staged Event-Driven Architecture

Select (轮询) 和 epoll （事件驱动 callback）

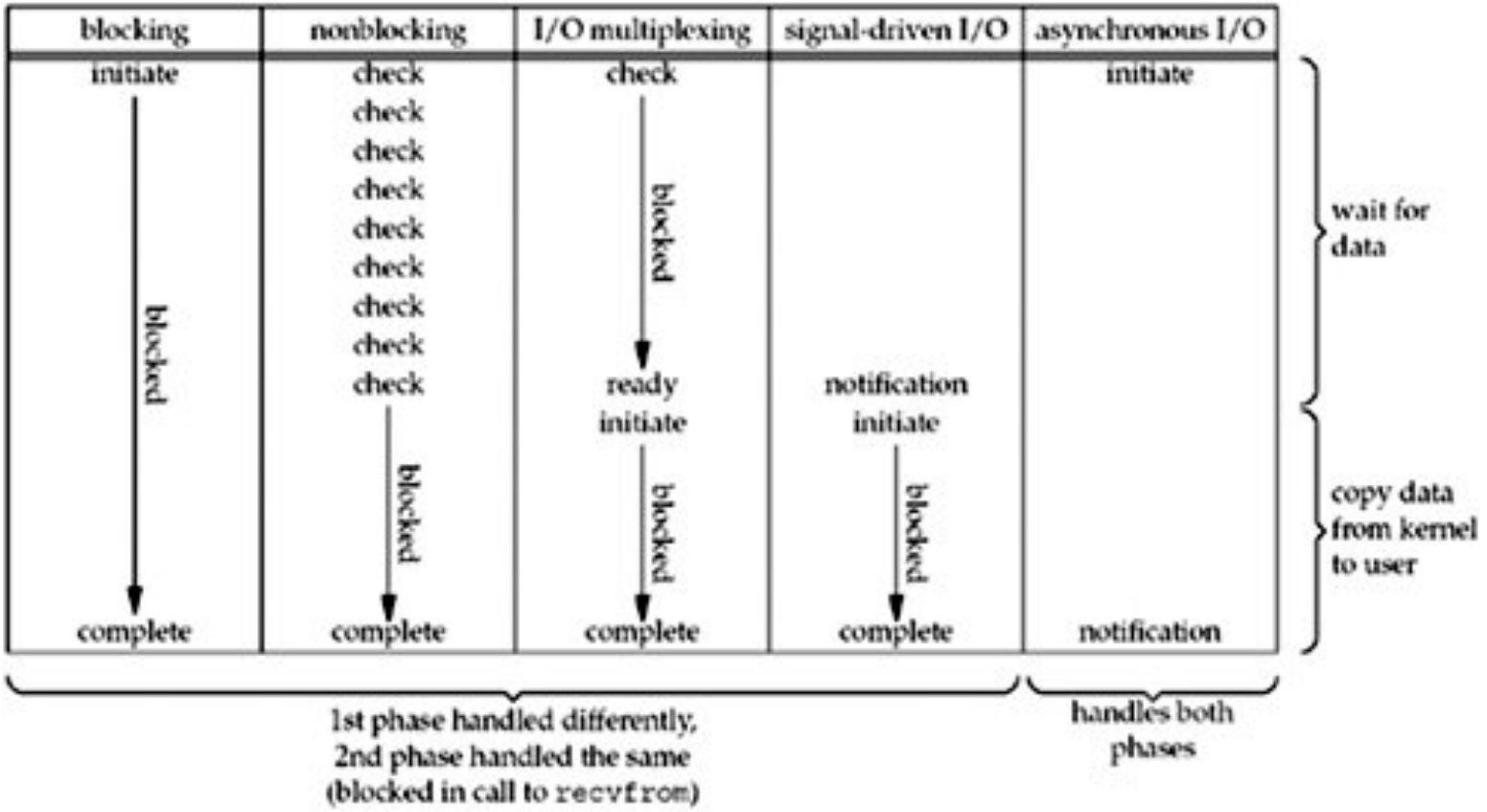
本质

方法: I/O模型 职责划分 内核和协议栈优化
目标 :尽量少占用CPU， 提高资源利用率,做到线性扩展
DRM平台(分布式存储和计算平台)的目标一致

常用网络库

Netty , Mina， libevent， libev, ACE， ASIO

各种IO模型


blocking
nonblocking
I/O multiplexing
signal-driven I/O
asynchronous I/O
initiate
check
check
check
check
check
check
check
check
check
check
initiate
blocked
blocked
ready
initiate
notification
initiate
blocked
blocked
blocked
complete
complete
complete
complete
notification
wait for
data
copy data
from kernel
to user
1st phase handled differently,
2nd phase handled the same
(blocked in call to recvfrom)
handles both
phases

数据结构

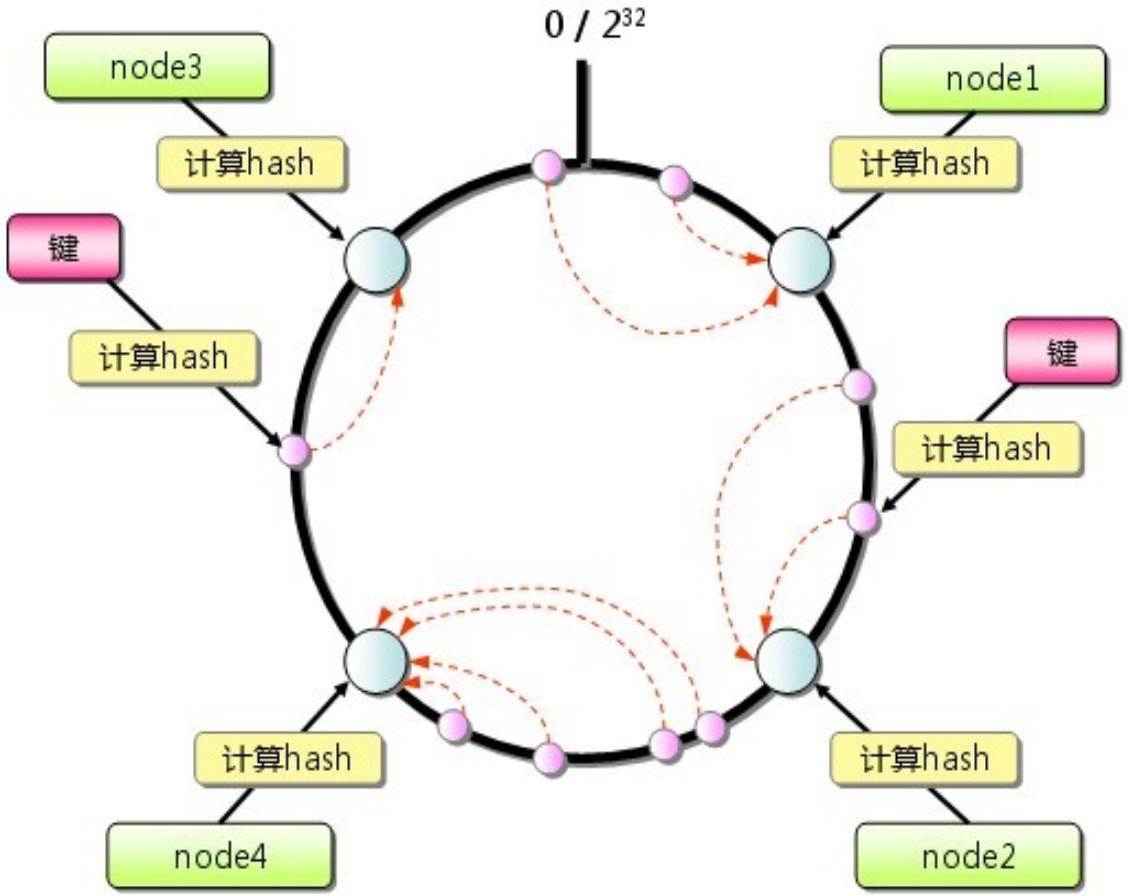
主要存储模型
Consistent hash, (去中心化)

B+ tree , (实时,随机)

LSM tree, (批量 顺序)

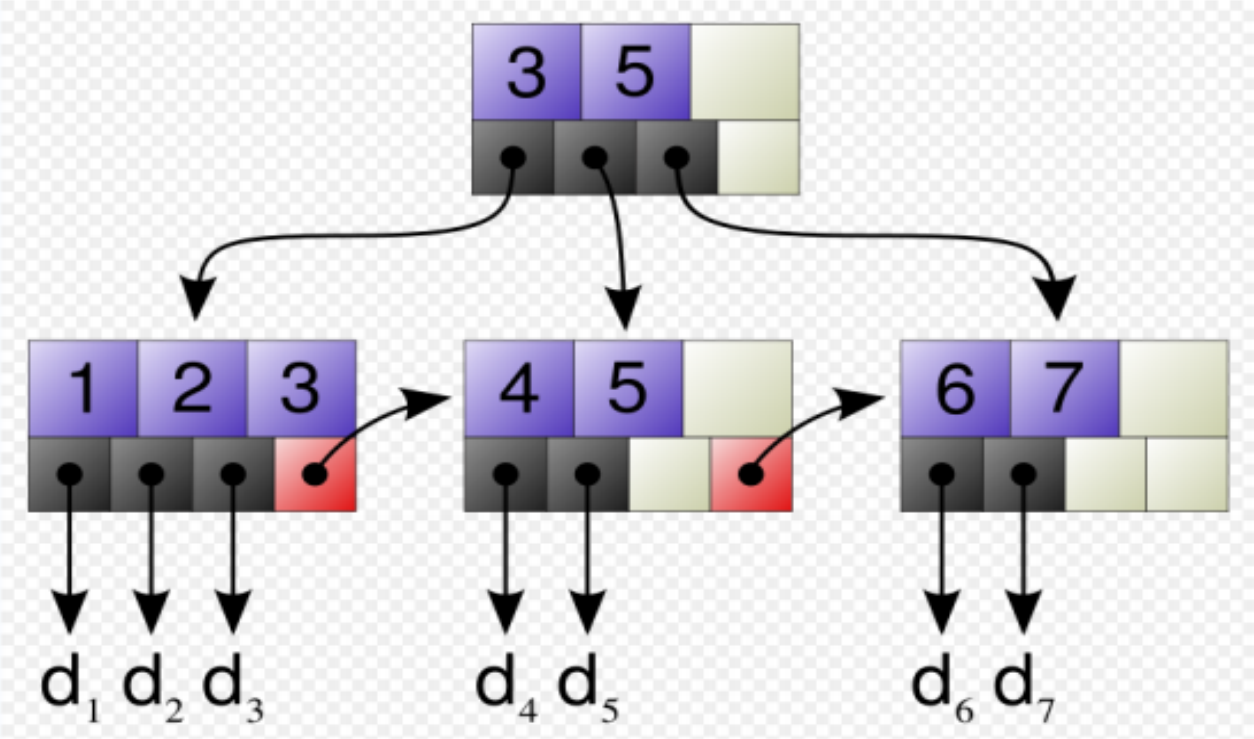
其它
Bitmap (bloom filter 缓存命中) Dynamo,hbase
Merkle Tree (一致性) Dynamo
Skip List (跳表 lsm变形) leveldb
prio_tree (优先搜索树 mmap) mognodb

consistent hash

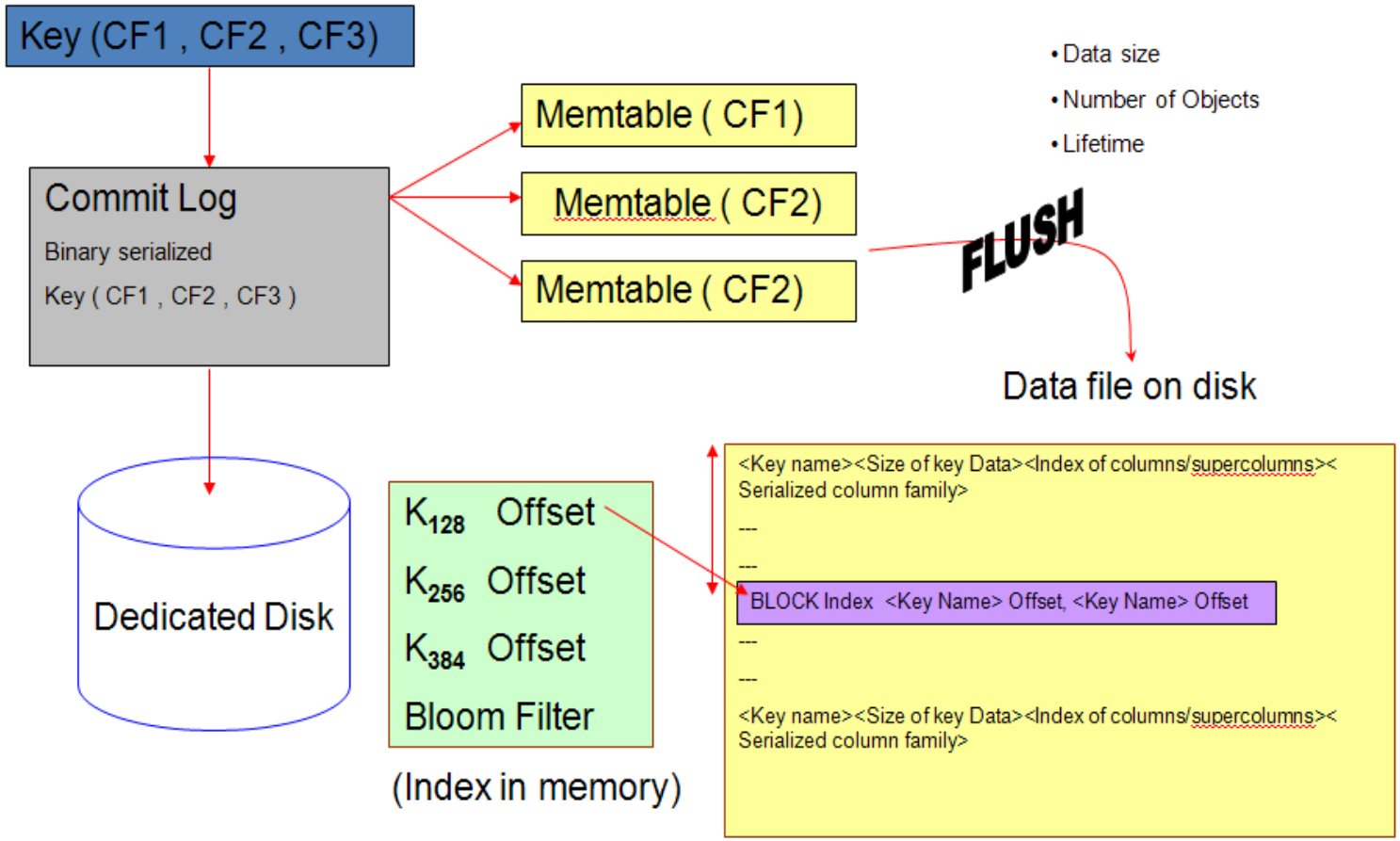


B+ tree

存储模型：B+ tree

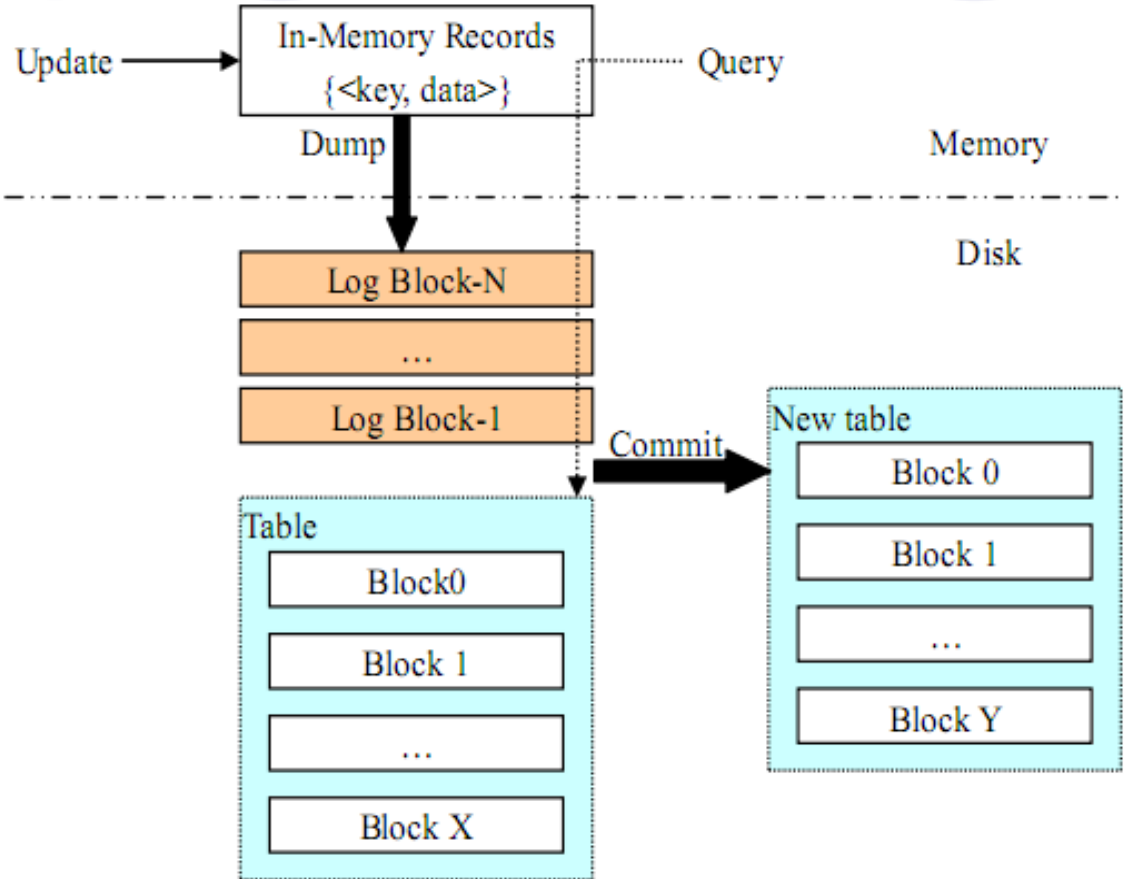


cassandra 数据存储过程

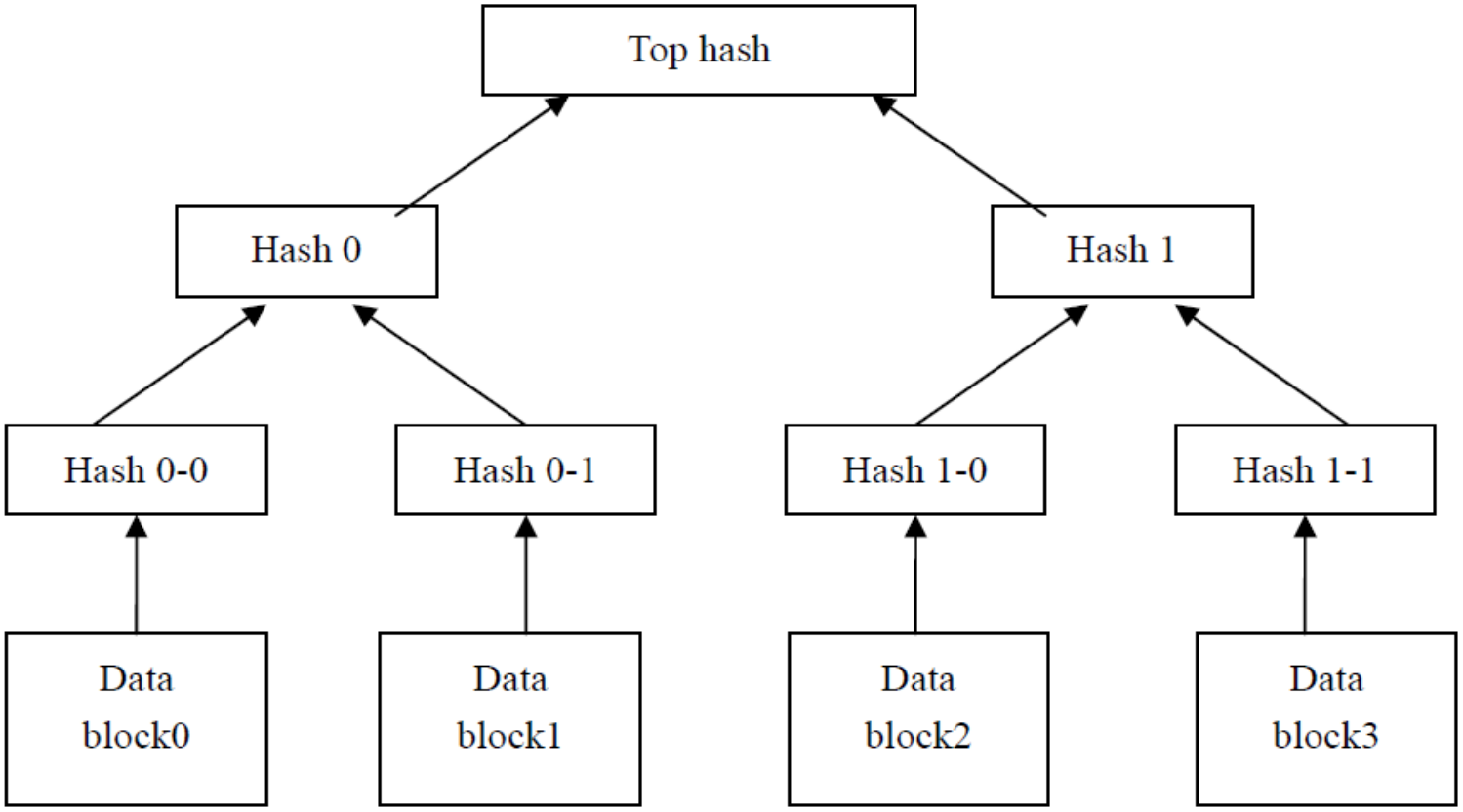


Lsm tree

存储模型：Log-based structure



Merkle Tree


Top hash
Hash 0
Hash 1
Hash 0-0
Hash 0-1
Hash 1-0
Hash 1-1
Data block0
Data block1
Data block2
Data block3

集群

1. 无主； Master_slave ; P2P(全主)

操作系统

进程调度 内存管理 文件管理：

存储领域

存储硬件
事务和锁（ MVCC 二段提交 三段提交 ,paxos）
常用数字 扇区 512, 内存页4k , 磁盘块大小 4k mtu 1500

其它领域

人工智能 : k_means

基础知识应用

- Dynamo　数据同步，　BT下载
  Merkle Tree
- memcached
  linux 内存管理 + libevent + (consisten hash)

如何设计架构

- 分析需求，做好平衡
- 使用Kiss原则 , 做到RAS
- 设计和充分利用硬件,分级存储

58的分布式存储实践（mysql,mongodb,file system,hadoop）

分析需求，做好平衡

1 分析需求

数据结构 结构化 半结构化 文件 table , object
数据特点 容量大小
访问模式 读写比例，实时读写，顺序读写
实时性

2 平衡

1 CAP理论 , BASE理论/ACID
2 选择存储模型 B+ or LSM
资源利用率和管理 高吞吐和低延迟 随机与顺序 规模与实时 B+ or LSM

3 实践

Mongodb选择

原则和目标

原则： kiss

unix 设计哲学

目标: RAS

RAS: Reliability, Availability, Scalability 高可靠，高可用，高扩展

R:过载控制 : Qos, (随机早期检测)
A: 容灾 ◊多副本 （同机柜，机房，数据中心）
S 扩容 ◊ 分片： a 取模 b 一致性hash c B+ tree 或变种

利用硬件,分级存储

数据中心

Facebook开源服务器、数据中心,将开源存储方案,

分级存储

网络延迟 局部性原理 ◊ cdn

按对象访问热点进行迁移：

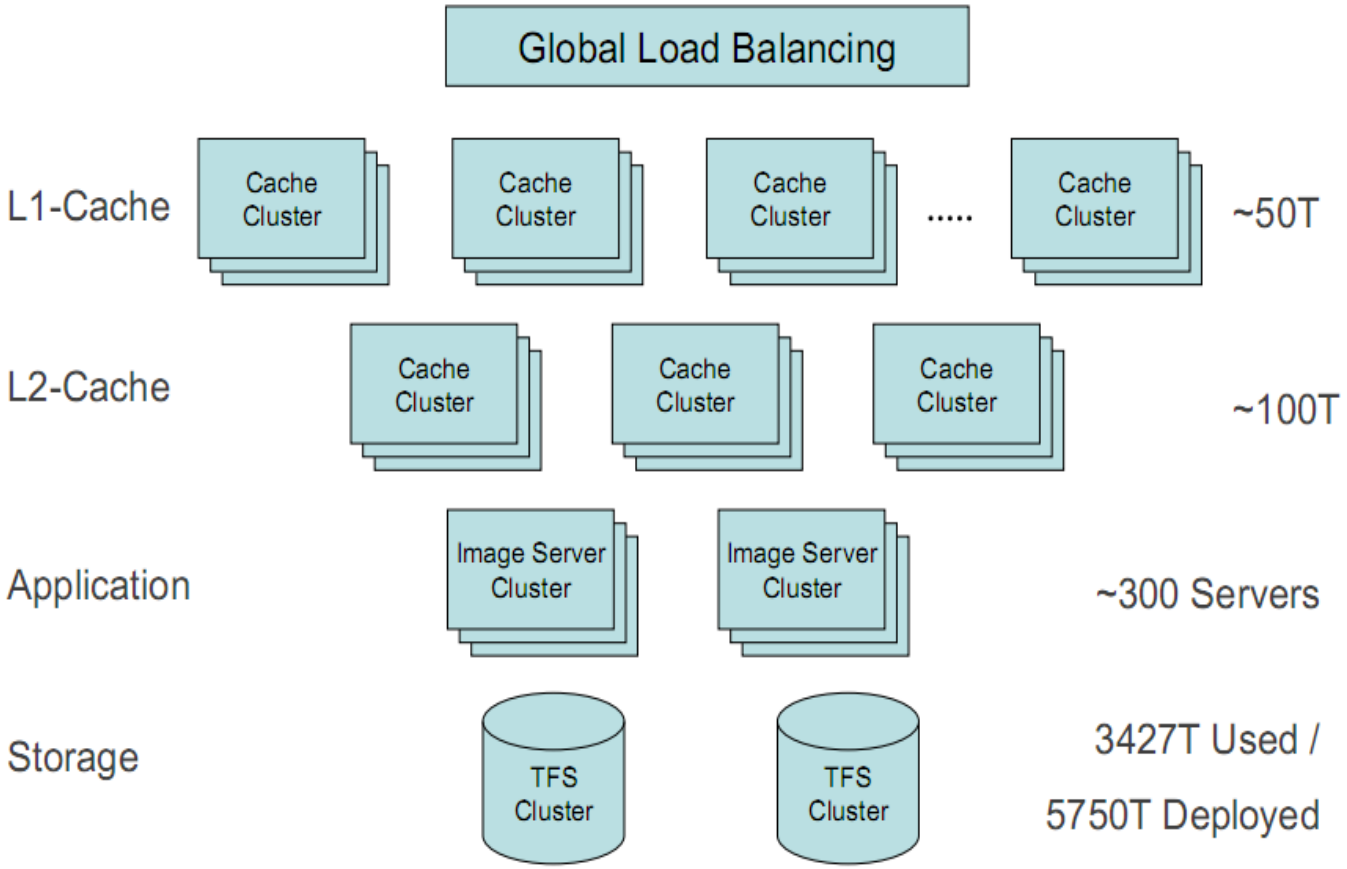
最热的进SSD，中等热度的放SAS，轻热度的存SATA

展望

硬件分离，你的CPU在一台服务器上，而内存在另外一台服务器，网卡可能在第三台服务器上

facebook数据中心电力布局

cdn分级示例


Global Load Balancing
L1-Cache
Cache Cluster
Cache Cluster
Cache Cluster
.....
Cache Cluster
~50T
L2-Cache
Cache Cluster
Cache Cluster
Cache Cluster
~100T
Application
Image Server Cluster
Image Server Cluster
~300 Servers
Storage
TFS Cluster
TFS Cluster
3427T Used /
5750T Deployed

数据分布

实践1：信息系统架构

- 架构:

  search engine(index) +Mysql (shard + M/S)+ memcached

  分库 : infoid % dbNum

  infoid 生成: local times + ip(mac) + pid

  Memcached 拦截 90%以上请求

数据量:

信息(贴子): ~10亿，20K qps ， 256 dbs

- 优点 :
- 架构简单 高可靠性（写logfile,失败重试）

  高扩展（2的倍数扩展, 备-->主）， 不用移动数据;
- 扩容时不用停服务

实践2：站内信和统计数(实时)架构

- **架构：** mongos + auto sharding（自动分片）

  实时统计数服务架构变迁: (mysql + memched) → mysql+ (应用层做缓存)-->cassandra-->redis--> mongodb(线性扩展)

  升级为通用服务:appid+appinforid == _id

Mongodb优势: 高可用，高性能,线性扩展,无模式，查询支持好

数据量： message : ~2亿

Steps : 统计数: ~10亿，10kqps, 10ktps,

实践：

1 shard key： Steps： inforid , message : userid

2 range(sql)to kv： range to kv (性能msgcount);sql to kv(兼容性)

- 3 key 尽量小 ; 4 谨慎使用自动分片 5 使用mongos(高可用)

实践3：图处存储

架构： cdn (Squid) + (lvs)+ Ngix (代理,实时生成缩略图 by GraphicsMagick) +httpServer(接入层 webdav,sso) + simple GFS(master-slave)

扩展：rest的URI层次扩展;文件名携带所有的信息

备份：三份， 主 + 实时备 +（延时备份 不同机房）

分块：Block Size 128M (option) , >8M 拆分chunk

Restful ：http://*..58control.cn/n_1817278286***.jpg

数据量: total 10t ， 100G/add ; 500w ； 20:1 (r:w); 1000iops

优点：

高可用 高扩展;

实时生成 节省空间

不足：

实践4：分布式计算的实践(附加)

基于Hadoop的统计分析平台

友好的用户界面；HBASE为主要数据源;

数据收集 js , udp接口 ,http 接口; log file ;

数据分析：只写部分业务代码;自动生成代码，部署和运行.

用户点击行为分析 （按时间段,区域，类别统计帖子点击数）

执行过程:Js(页面嵌入) → http Server -->udp Server -->原始输入数据 → hadoop -->生成统计数据

Drm云平台;

根据前端请求动态增加或加收站点和服务.

实时计算-->spark,S4,hama等;

Mpl(scala) ;ESB;分布式中间层;

微博：http://weibo.com/zhuozhe
邮箱：xuzh@58.com

Q&A

谢 谢