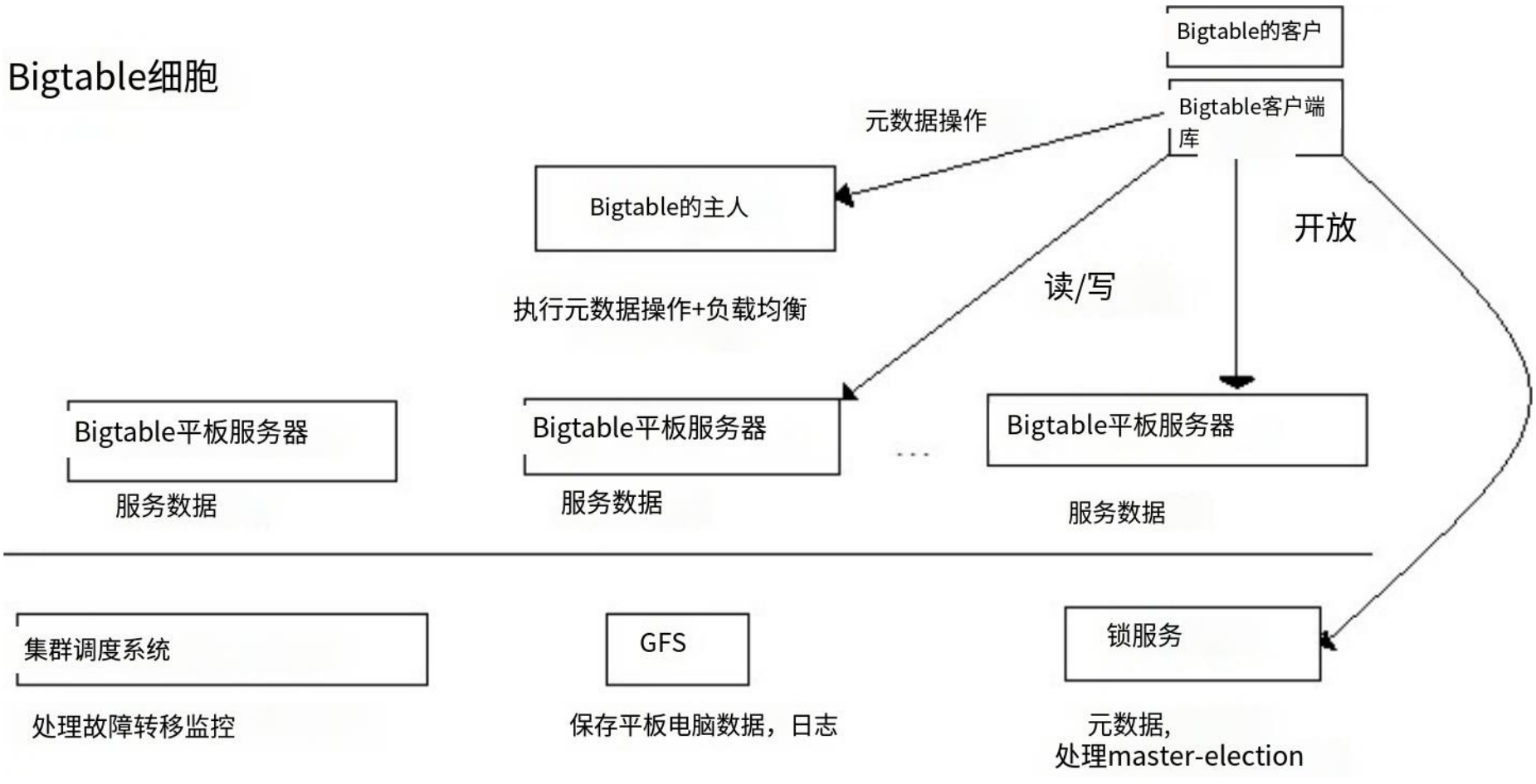
大衣串击可

编辑主字幕样式

大数据分析|何铁科
http: // hetieke.cn

南京大学
南京大学

”大地图”


Bigtable细胞
Bigtable的客户
Bigtable客户端库
元数据操作
Bigtable的主人
执行元数据操作+负载均衡
读/写
开放
Bigtable平板服务器
服务数据
Bigtable平板服务器
服务数据
...
Bigtable平板服务器
服务数据
集群调度系统
处理故障转移监控
GFS
保存平板电脑数据，日志
锁服务
元数据，
处理master-election

参考文献

’Chang, F.， Dean, J.， Ghemawat, S.， Hsieh, W.， Wallach, D.， Burrows, M.， Chandra, T.， Fikes, A.， and Gruber, R. 2008。大表:结构化数据的分布式存储系统。*ACM 反式。第一版。系统 26,2*(2008 年 6 月)，1-26

大表:结构化数据的分布式存储系统，第七届操作系统设计与实现研讨会论文集，2006 年 11 月

康奈尔大学高级分布式存储系统课程的幻灯片

路线图

1.动机
2.概述
3.数据模型
4.客户端 API 概述
5.构建块
6.大表实现的基础知识
7.细化
8.结论

动机(我)

- **大量(半)结构化数据在谷歌**
  - url:
    - 内容、抓取元数据、链接、主播、pagerank、…
  - 每用户数据:
    - 用户偏好设置，最近的查询/搜索结果，…
  - 地理位置:
    - 物理实体(商店、餐馆等)、道路、卫星图像数据、用户注释，……
- **规模大**
  - 数十亿的 url，许多版本/页面(~20K/版本)
  - 亿级用户，千 q/秒
  - 100TB+的卫星图像数据

动机(2)

- 要求 DB 具有广泛的可扩展性、广泛的适用性、高性能和高可用性

商业数据库的成本

- 在内部建立系统有助于将其用于其他低增量成本的项目
- 可以进行低级别存储优化，这有助于提高性能

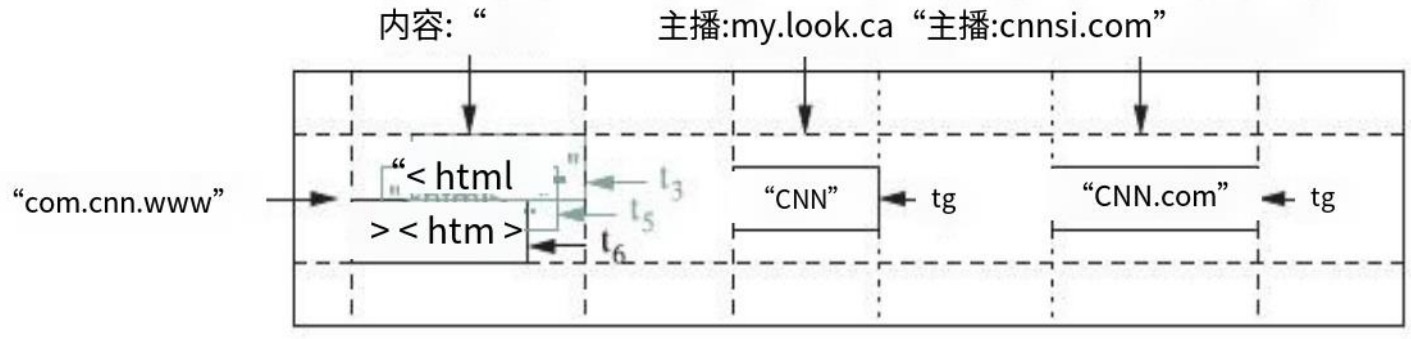
概述

- Bigtable 不支持全关系数据模型
- 支持动态控制数据布局和格式
- 客户端可以通过选择模式来控制数据的局部性
- 模式参数让客户端动态控制是否从内存/磁盘提供数据。

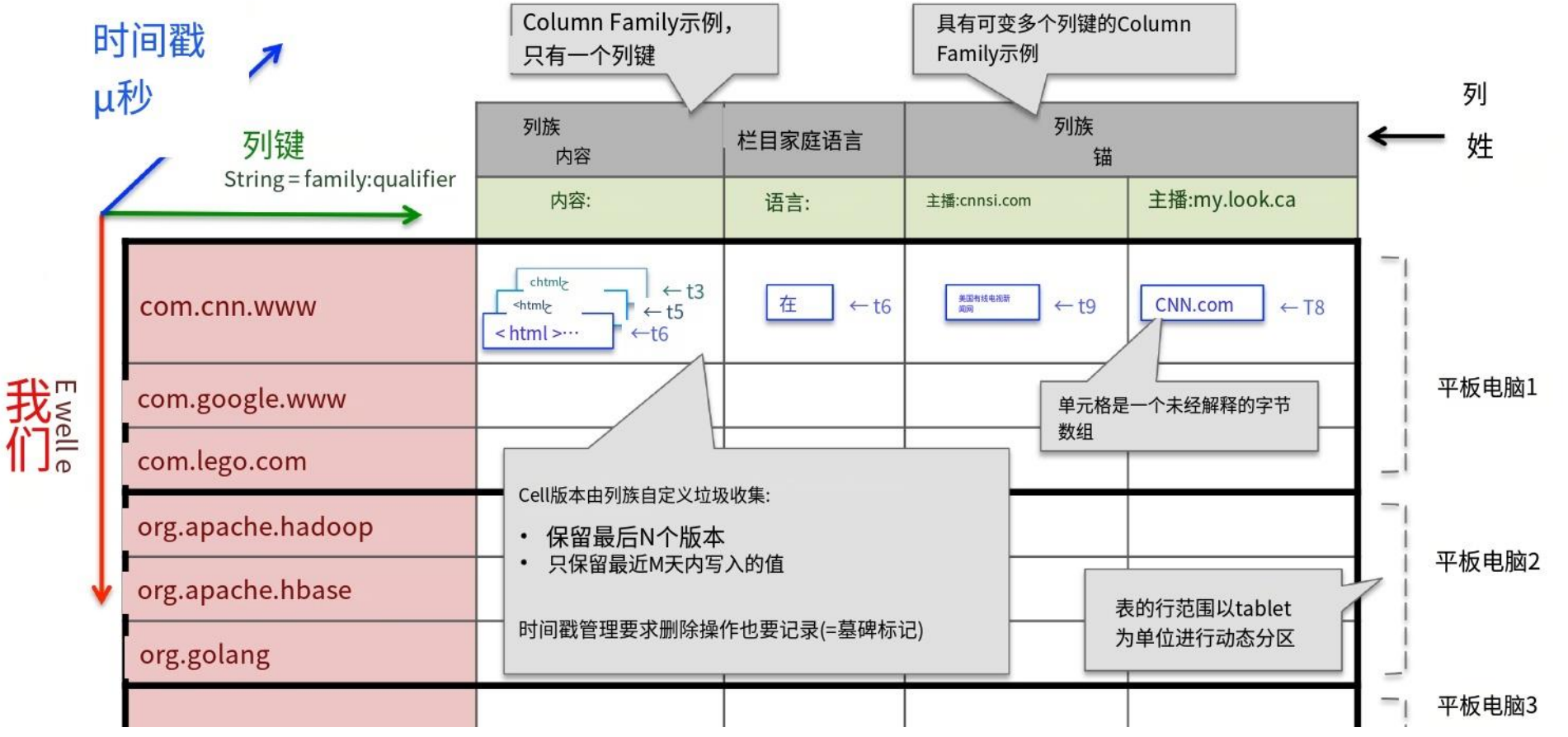
数据模型(一)

- 分布式多维稀疏图
- (行、列、时间戳)->单元格内容
- 行键是任意字符串

行是事务一致性的单位



数据模型(II)


时间戳
μ秒
列键
String = family:qualifier
Column Family示例，只有一个列键
具有可变多个列键的Column Family示例
列
姓
列族
内容
栏目家庭语言
列族
锚
内容:
语言:
主播:cnnsi.com
主播:my.look.ca
com.cnn.www
com.google.www
com.lego.com
org.apache.hadoop
org.apache.hbase
org.golang
<html>
<html>
< html >…
←t3
←t5
←t6
在
←t6
←t9
CNN.com
←T8
单元格是一个未经解释的字节数组
Cell版本由列族自定义垃圾收集:
• 保留最后N个版本
• 只保留最近M天内写入的值
时间戳管理要求删除操作也要记录(=墓碑标记)
表的行范围以tablet为单位进行动态分区
平板电脑1
平板电脑2
平板电脑3
我们
E well e

数据模型(三)

具有连续键的行被分组为“平板电脑”。

列键被分组成称为“列族”的集合，这构成了访问控制的单位。

存储在列族下的数据通常是相同类型的。

列键的命名语法如下:列族:限定符

数据模型(四)

访问控制和磁盘/内存会计在列族级别执行。

- Bigtable 中的每个单元格可以包含多个版本的数据，每个版本都通过时间戳进行索引。
- 时间戳是 64 位整数。
- 数据以时间戳递减顺序存储，以便于访问最近的数据。

客户端api(我)

n **Bigtable api 提供以下函数:**

创建/删除表、列族

- 更改集群、表和列族元数据，如访问控制权限

客户机api (2)

n **Bigtable api 提供了如下函数:**

•支持单行事务

允许单元格用作整数计数器

客户端提供的脚本可以在服务器的地址空间中执行

- “胖乎乎的”用于以下任务
  - 存储根表、模式信息、访问控制列表。
  - 同步和检测平板电脑服务器
- 什么是 **Chubby？**
  - 高可用持久锁服务。
  - 具有目录和小文件的简单文件系统

对文件的读写是原子性的。

当会话结束时，客户端解除所有锁

构建模块(II)

nGFS 来存储日志和数据文件。nSSTable 用于内部存储数据文件。什么是 SSTable ？

- **命令**
- **不可变的**
- **键到值的映射，都是任意字节数组**
- **优化存储在 GFS 和可选映射到内存。**

构建模块(III)

nBigtable 依赖于谷歌集群管理系统如下:

- 调度作业
- 管理共享机器上的资源
- 监控机器状态

处理机器故障

执行(I) -精通

- 三大组成部分
  - 库(每个客户端)
  - 一个主服务器
  - 多台平板电脑服务器
- 个单主任务:
  - 将平板电脑分配给服务器
  - 检测服务器的添加/过期
  - 平衡服务器的负载
  - GFS 中的垃圾回收
  - 处理模式更改

实现- Tablet Server

“平板电脑服务器任务:

§ 处理对加载的平板电脑的读写请求 § 拆分平板电脑

客户端直接与服务器通信 § Master 轻负载

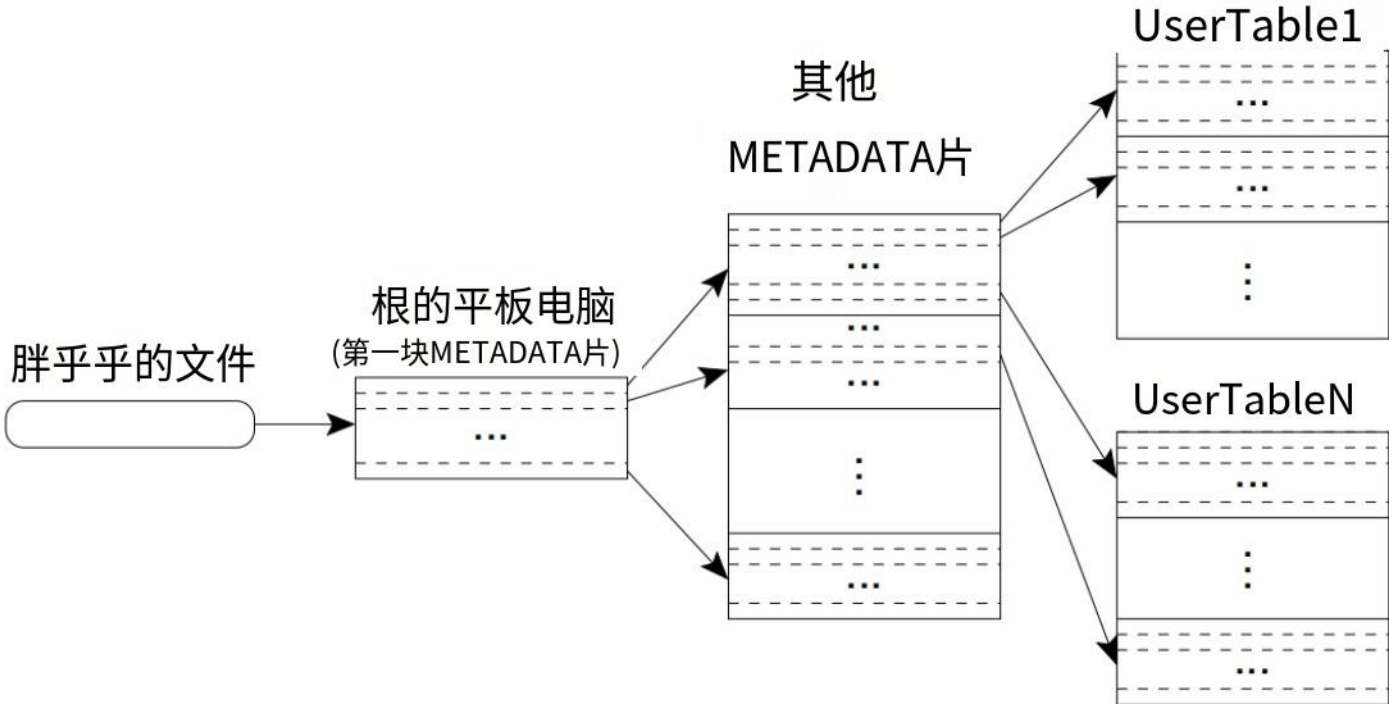
`Each 表`

§ 开始时用一块平板电脑

§ 随着增长分割，每片大小为 100- 200mb

平板电脑的位置

我们使用类似于 B+树的三级层次结构来存储片的位置信息



平板电脑的位置

§ Chubby 中的一个文件用于根片的位置 § 根片包含元数据片的位置 § 元数据表包含用户片的位置

- § 行键:[平板电脑的表 ID] +[结束行]

¡ 客户端库缓存平板电脑的位置

- § 如果位置不确定，则晋升

平板电脑作业

- 一项 Master 跟踪 Chubby 使用的分配/活动服务器
  - § 服务器创建并锁定*服务器目录*中的唯一文件 § 如果失去锁定则停止服务
  - § Master 定期检查服务器
    - § 如果 lock 丢失，master 尝试锁定文件，取消分配 tablet § master 失败不要更改 tablet 分配

平板电脑作业

- Master 重启
  - 在 chubby 中抓取唯一的主锁;扫描服务器目录;与每个 tablet 服务器通信;扫描元数据表

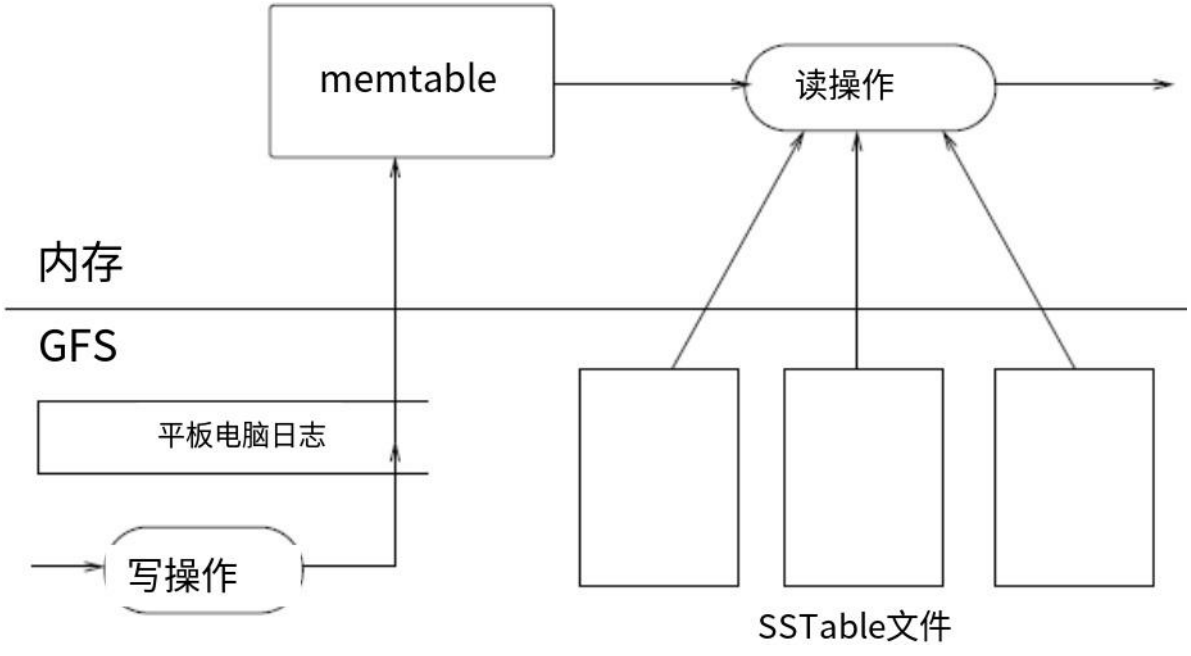
平板电脑的变化

- Tablet 创建/删除/合并 à master
- Tablet 拆分 à Tablet 服务器
  - 服务器通过在元数据中记录新平板电脑的信息来提交
  - 通知主服务器

平板电脑服务

- 平板电脑服务
  - GFS 中的片剂
  - REDO 日志
    - 在 sstable 序列中较旧的

平板电脑服务



平板电脑恢复:

•服务器从元数据表中读取 sstable 列表

•List =(包含 sstable +一组用于 REDO 提交日志的 ptrs)

服务器通过应用 REDOs 来重构状态和 memtable

R/W在平板电脑

服务器授权发送方

- 胖乎乎的文件中允许的用户阅读列表“Write”
  - 写入提交日志(memtable)的有效突变 - 使用的是组提交

只能读

- 在 sstable 和 memtable 的合并视图上执行

压实

- 小压实
  - (Memtable size > threshold)，新 Memtable，旧 Memtable 转换为 SSTable，写入 GFS
    - 减少内存使用和减少恢复中的日志长度

压实

- 合并压实

  读取和收缩少量 sstable 和 memtable

- §BT 为删除的数据回收资源§删除的数据消失(敏感数据)

改进-局部性组

客户端将多个 cols 族分组在一起，为平板电脑中的每个 LG 单独设置 SSTable，划分不能同时访问的 cols 族

- (语言和校验)VS(页面内容)
- 更高效的阅读

改进-局部性组

- 每个组的调优参数
  - 一个 LG 声明在内存中
  - 对频繁访问的小块有用
    - 例子。元数据中的 Location 列族

改进-压缩

客户端可以压缩 SSTable 为 LG 压缩格式适用于每个 SSTable 块 § 小表部分读取不完全分解通常两遍压缩

§ 在小窗口(16 KB)内快速重复查看 § 大幅减少(10-1)

§ 数据布局(单个主机的页面放在一起)

改进——读性能缓存

- 平板电脑服务器使用两级缓存
  - 扫描缓存:SSTable 接口返回给平板服务器代码的键值对的高级缓存
  - 块缓存:从 GFS 读取 sstable 块的低级缓存

改进-布隆过滤器

- 问题:读取操作必须从构成平板状态的所有 sstable 中读取
  - -大量的磁盘访问 😭
- 解决方案:对特定位置组中的每个 SSTable 使用 Bloom 过滤器
  - -布隆过滤器使用少量内存并允许知道 如果 SSTable 不包含指定的行/列对
  - -大多数不存在的行或列的查找不需要接触磁盘 😊

Refinements-Commit-log实现

- 提交日志的实现
  - 每个平板服务器都有一个提交日志
  - 恢复复杂
    - 主坐标排序日志文件<*Table, Row, log Seq*)

细化-不变性

- 加速药片恢复
- 利用不变性
  - Bigtable 系统的各个部分都被简化了，因为所有生成的 sstables 都是不可变的
  - sstable 的不变性允许快速分割片剂

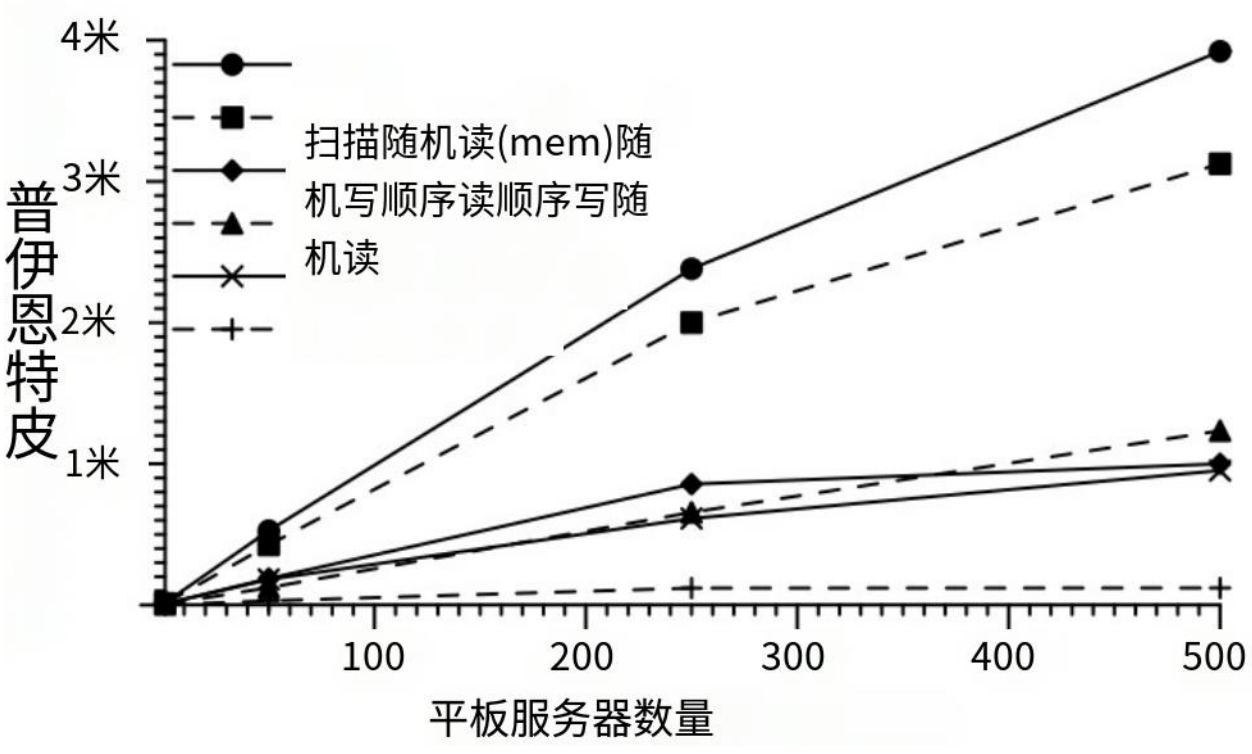
绩效评估

- 每台服务器#读/写号

实验	平板电脑服务器数量			
	1	50	250	500
随机读	1212	593	479	241
随机读(mem)	10811	8511	8000	6250
随机写	8850	3745	3425	2000
顺序读	4425	2463	2625	2469
顺序写	8547	3623	2451	1905
扫描	15385	10526	9524	7843

绩效评估

- 总#读/写号


4米
3米
2米
1米
普伊恩特皮
扫描随机读(mem)随
机写顺序读顺序写随
机读
100
200
300
400
500
平板服务器数量

结论

Bigtable 已经实现了高性能、数据可用性和可扩展性的目标。

- 它已经成功部署在真实应用中(个性化搜索，Orkut, GoogleMaps，…)

构建自己的存储系统的显著优势，如设计数据模型的灵活性，对实现的控制以及 Bigtable 所依赖的其他基础设施。