IT大咖说
知识分享平台

分布式存储系统设计方案浅析

刘迪

腾讯网
qq.com

关于我

MongoDB 中文社区

IT大咖说 知识分享平台

刘迪，腾讯运维工程师，中国传媒大学计算机专业硕士毕业，现从事数据库运维、存储系统设计等相关工作。

对数据库相关技术、负载均衡技术、分布式系统理论和工程实践有一定理解。

在腾讯参与了MongoDB技术实验性研究和数据库平台建设的相关工作，也曾参与多个存储系统重构项目的研究。

热衷于分布式存储、数据库技术等相关领域的研究和实践。

腾讯网 qq.com

目录




分布式储存基础

大规模分布式存储架构

小规模分布式存储架构

腾讯MongoDB托管平台建设历程

第一部分




分布式储存基础

大规模分布式存储架构

小规模分布式存储架构

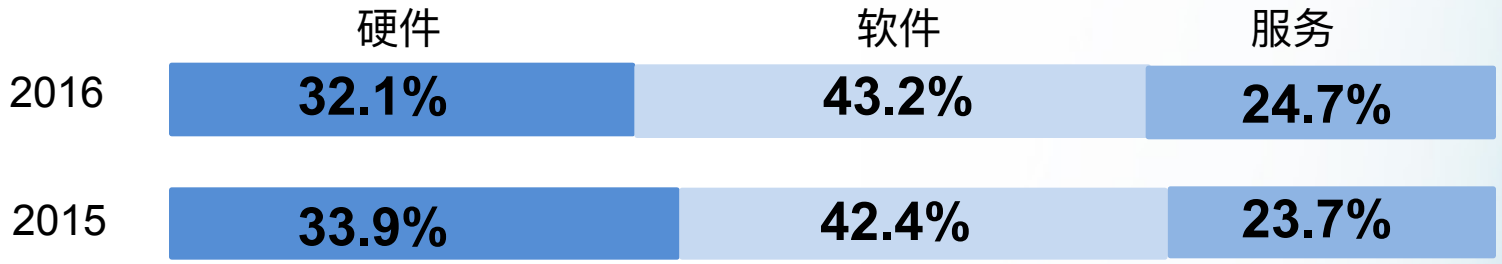
腾讯MongoDB托管平台建设历程

大数据行业发展现状

1 2016年中国大数据市场增速达到45%；预计2017至2020年保持30%以上。

2 66.1%的企业表示非结构化数据比例在70%以上，22%的企业非结构化数据比例为50%~70%。

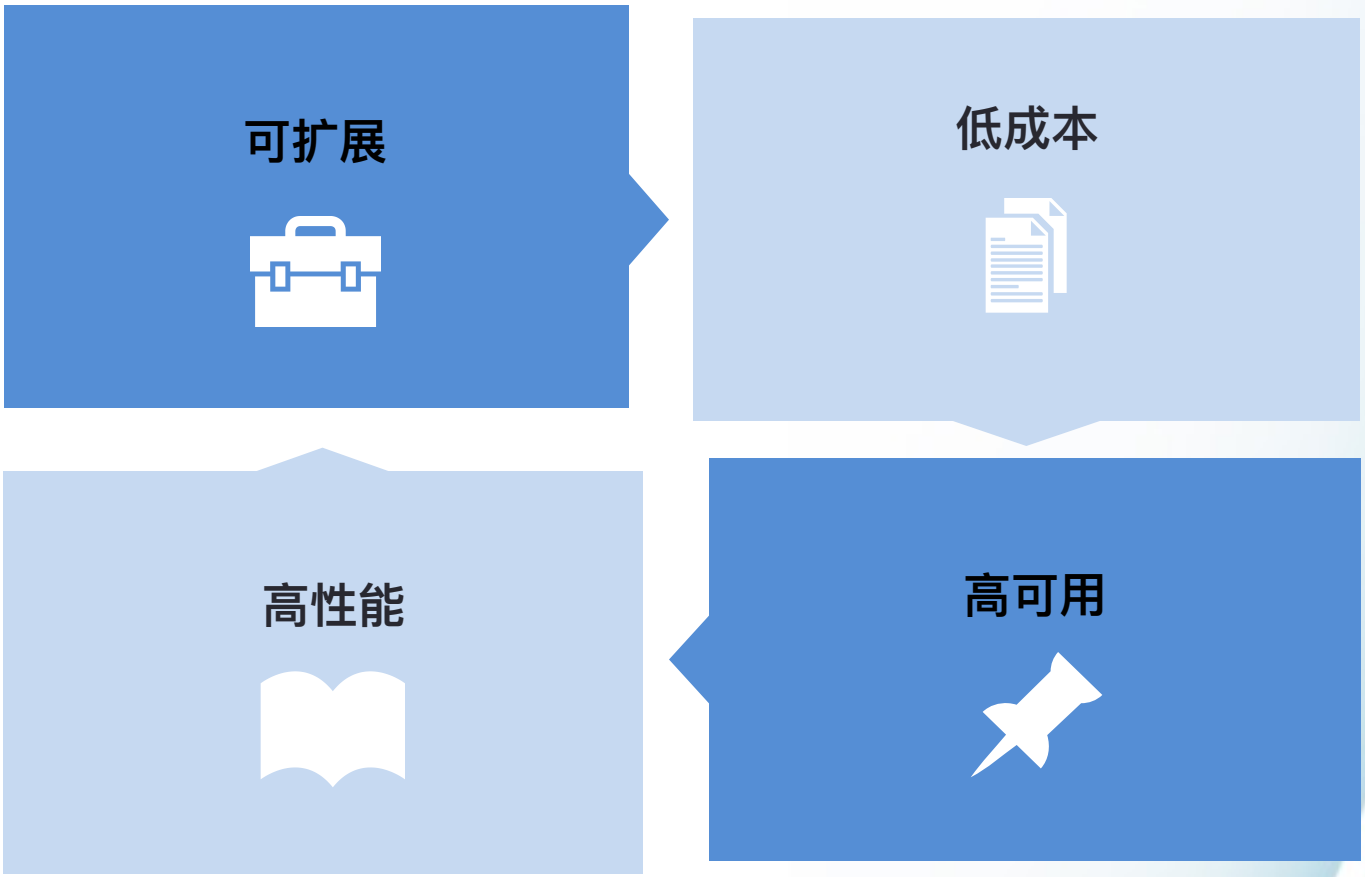
3



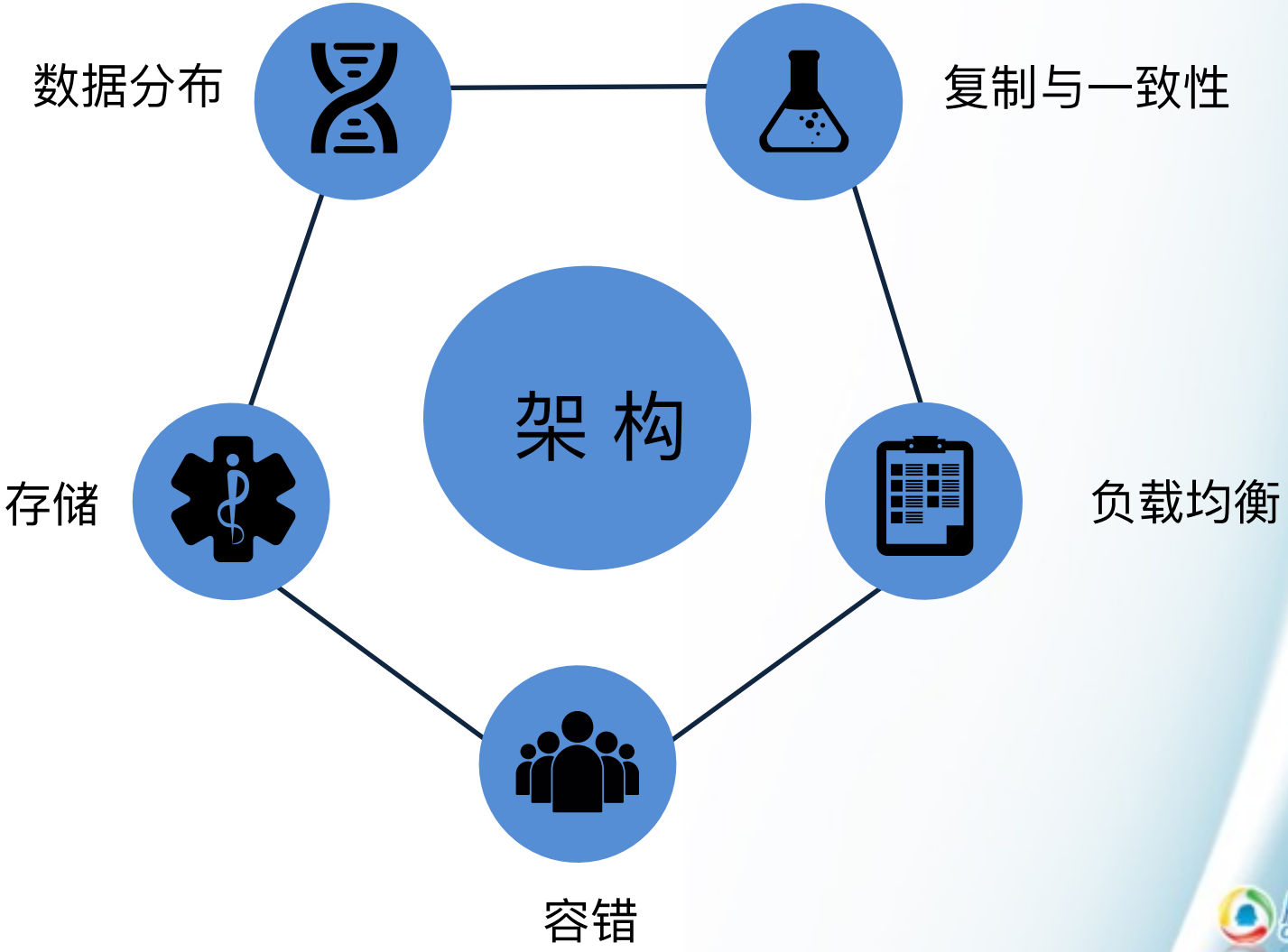
4

分布式存储系统

分布式存储系统是大量普通的PC服务器通过Internet互联，对外作为一个整体提供存储服务。



分布式存储关键技术



第二部分



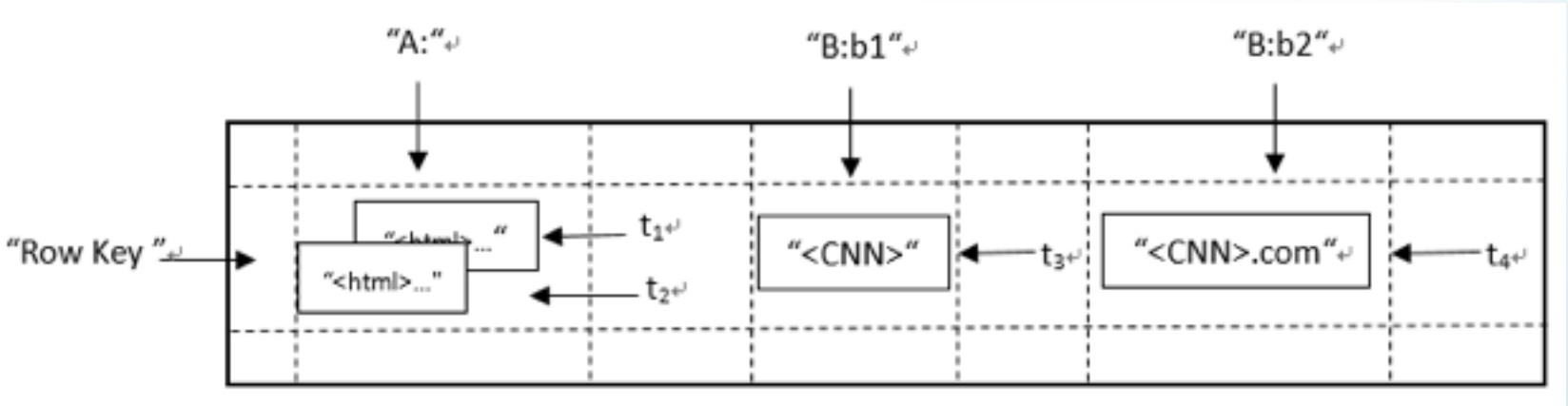

分布式储存基础

大规模分布式存储架构

小规模分布式存储架构

腾讯MongoDB托管平台建设历程

Bigtable



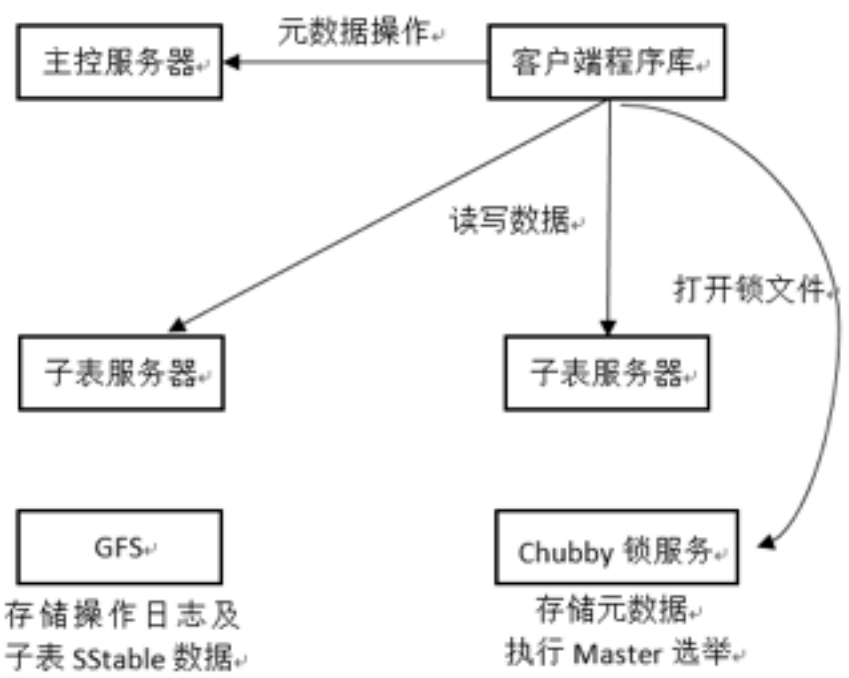
1、 架构特点：GFS+ Bigtable双层架构，Bigtable是在GFS之上的一层分布式索引。

2、设计理念：构建在廉价硬件之上，通过软件层面提供自动化容错和线性扩展能力。

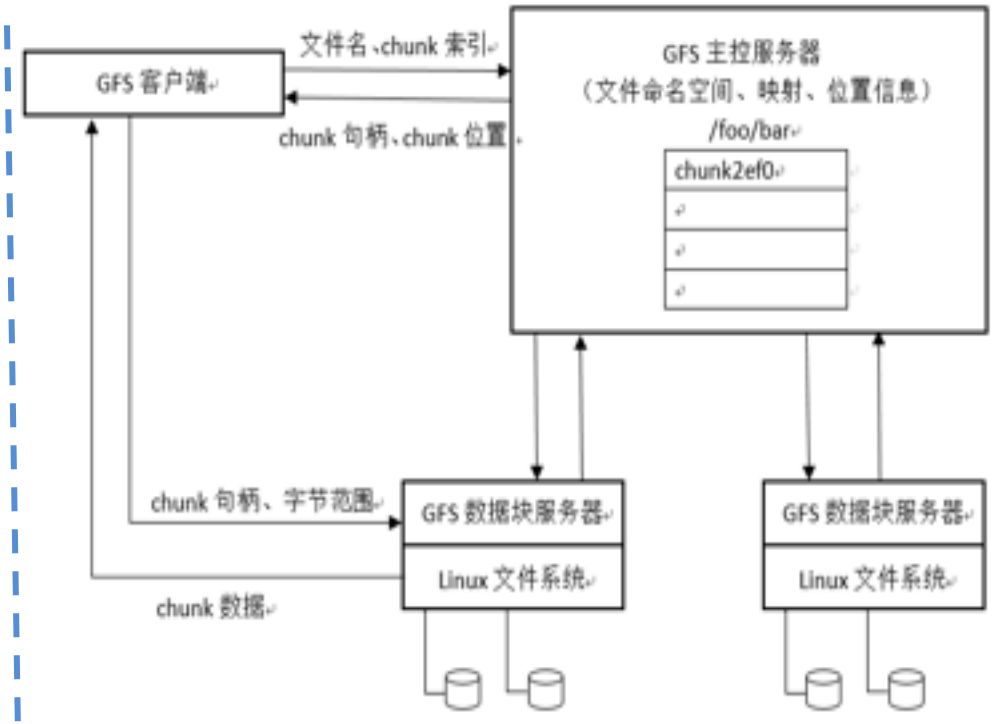
3、数据结构：Bigtable系统有很多表格组成，每个表格包括很多行，每行通过一个主键（Row Key）唯一标识，每行又包括很多列（Column）。某一行的某一列构成一个单元（Cell），每个单元包含多个版本的数据。多个列组织成一个列族，是Bigtable中访问控制的基本单元，列族创表前需预先设定。

Bigtable

架构



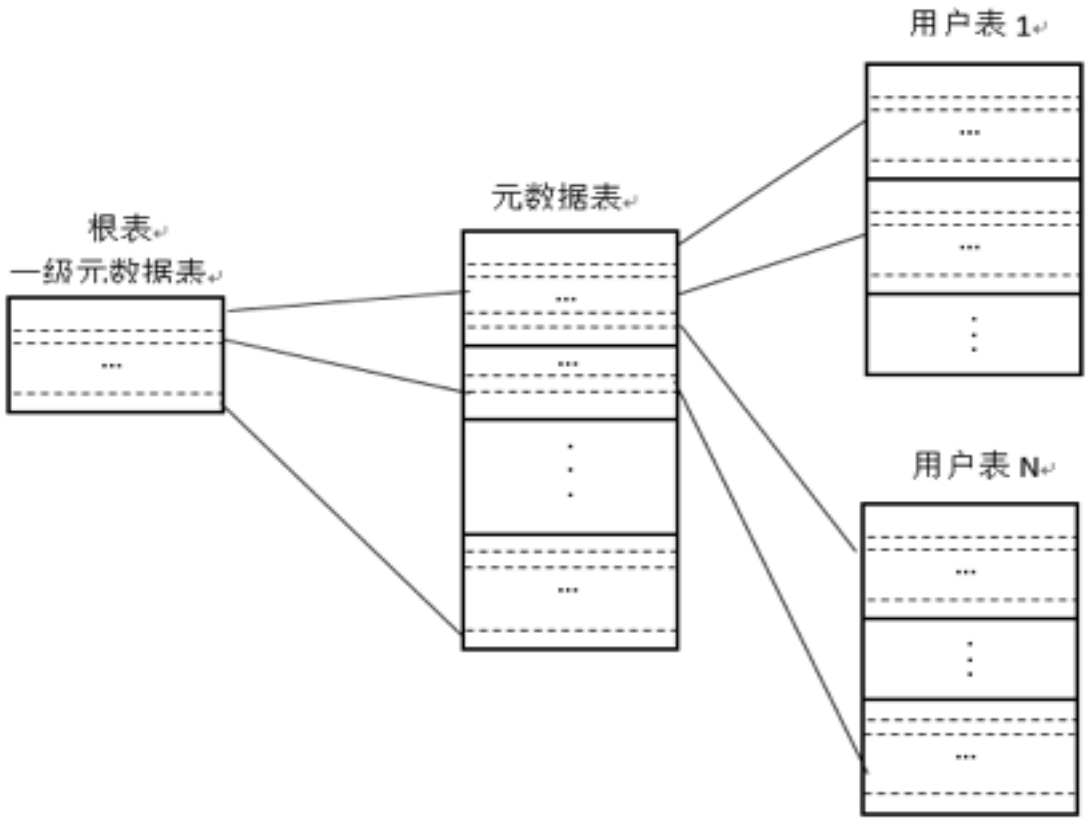
Bigtable



GFS

Bigtable

数据分布



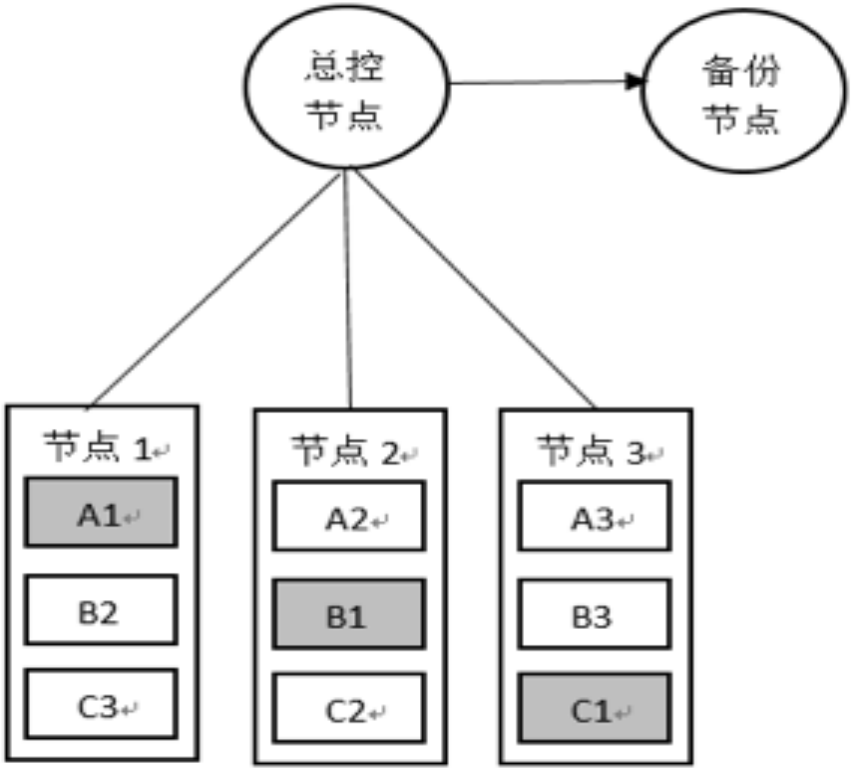
1、Bigtable中的数据被切分为大小100~200MB的子表，所有数据按照行主键全局排序。

2、根表、元数据表、用户表。

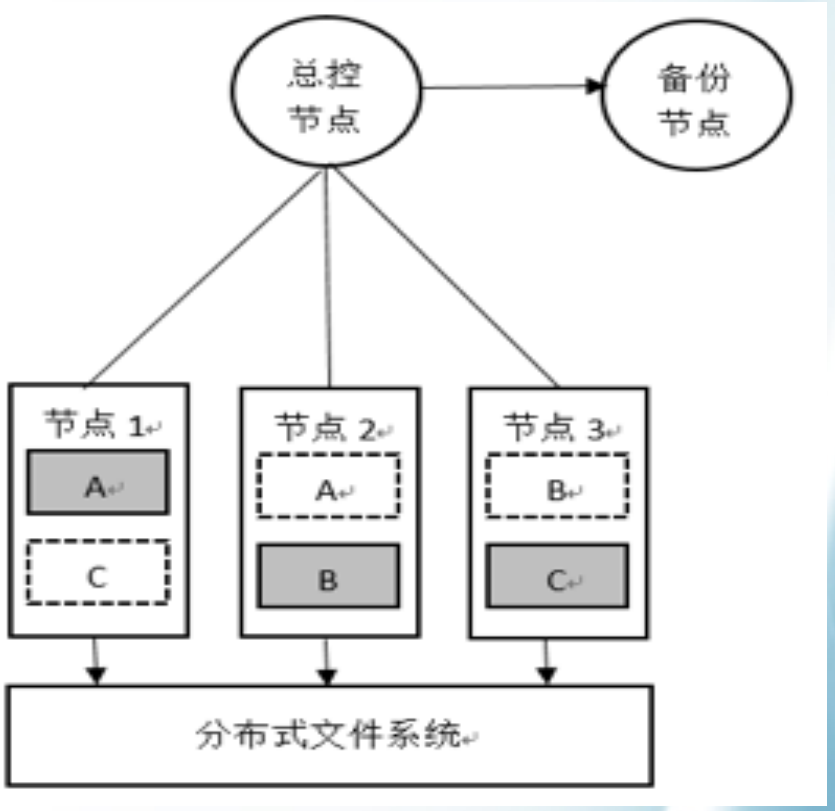
3、缓存、预取

Bigtable

容错



单层结构



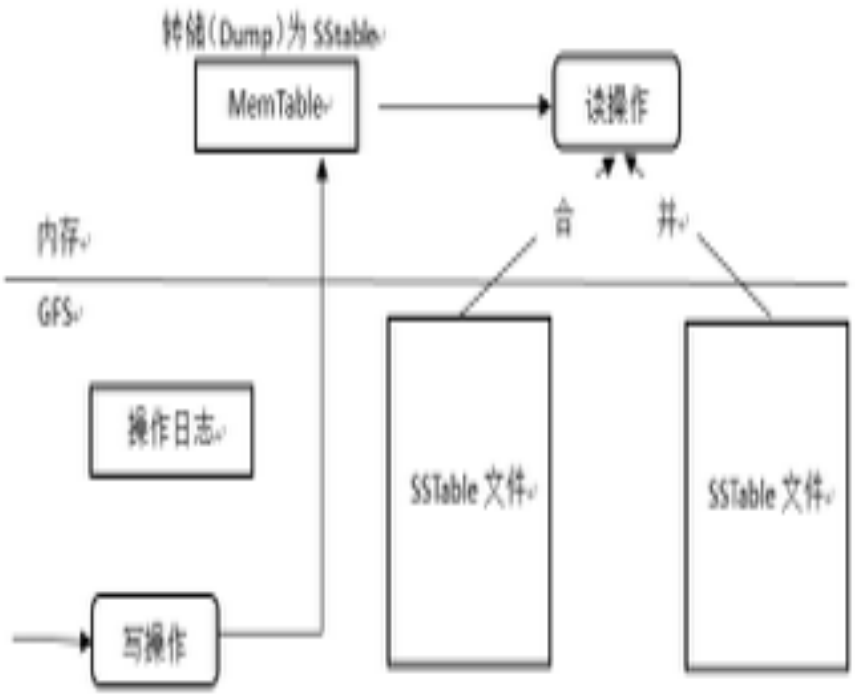
两层结构
(Bigtable)

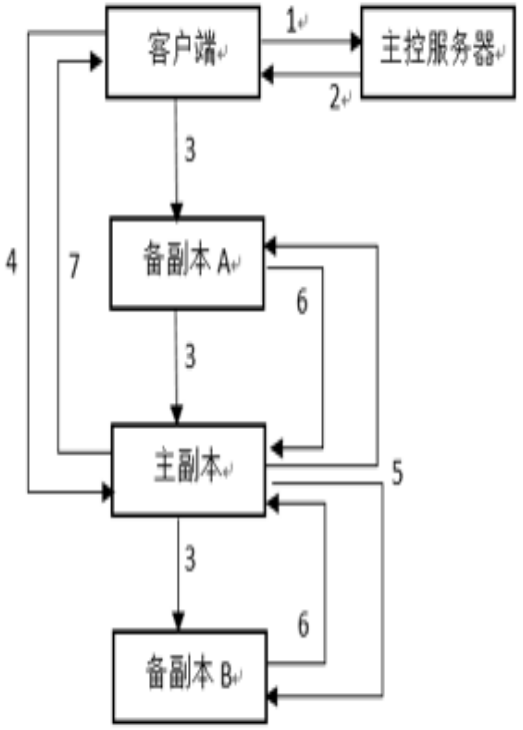
Bigtable

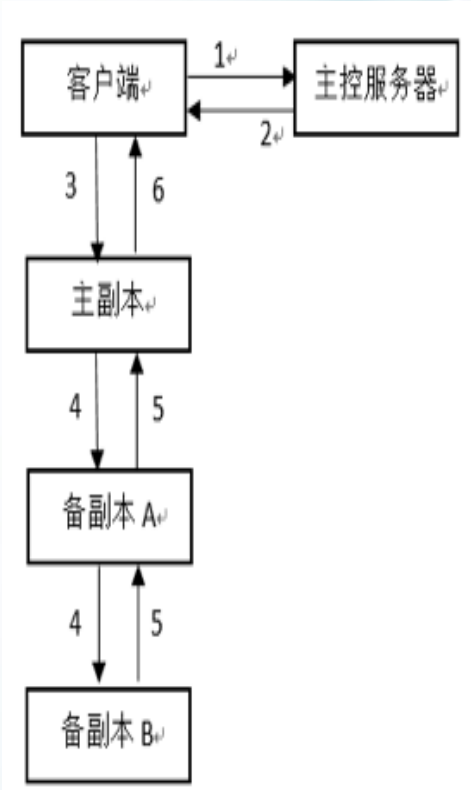
Bigtable		GFS	
强一致性	1 复制与一致性	复制与一致性	1 弱一致性
子表迁移	2 负载均衡	负载均衡	2 Chunk创建、复制、移动

Bigtable

存储







Bigtable **GFS** 传统主备复制

Tera

Tera是由百度开源的一个高性能、可伸缩的结构化数据存储系统，被设计用来管理搜索引擎万亿量级的超链与网页信息。使用多级Cache系统，充分利用新一代服务器硬件大内存、SSD盘和万兆网卡的性能优势，100PB级的数据存储量，支撑了目前百度万亿级的动态读写业务。

- 使用分布式文件系统（HDFS）持久化数据与元信息
- 使用分布式协调服务（Zookeeper）选主与协调

Tera

性能指标

①单机吞吐

顺序读写: 100MB/S　　随机读1KB: 30000QPS　　随机写1KB: 30000QPS

②延迟

在延迟敏感性不高的场合，使用延迟换吞吐策略，写操作延迟<50ms，读延迟<10ms。
在延迟敏感性高的场合，读写延迟定位在<1ms，但吞吐会有损失。

③扩展性

水平扩展至5000台机器，单机管理200个数据分片。

④稳定性指标

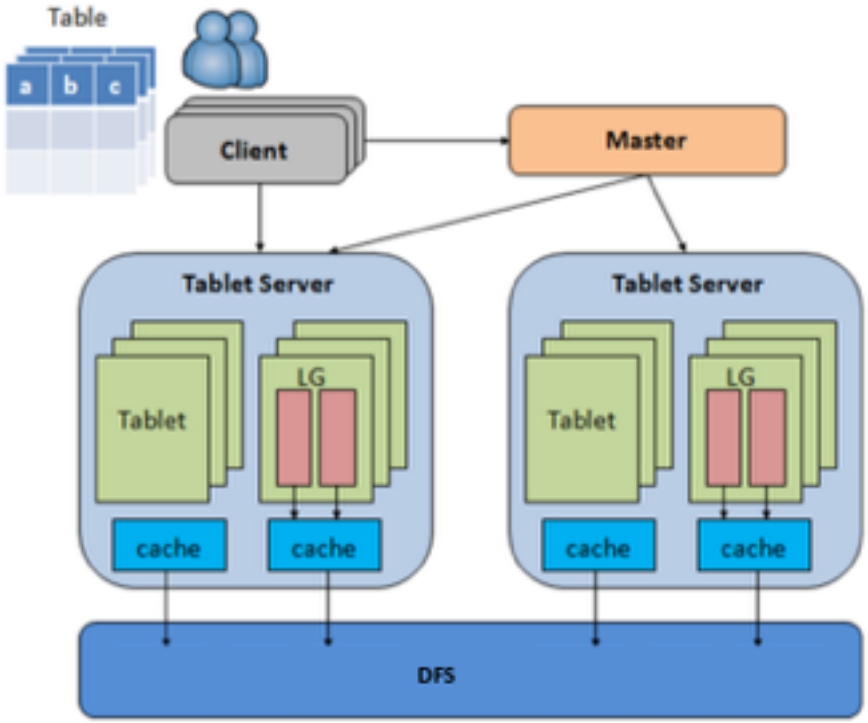
数据节故障：设计数据节点30S不可用，才进行切换。
Master故障：设定master的lease过期时间为10S，10秒无法服务才会被备份节点取代。

⑤集群均衡度

设计单机数据量超过集群平均数据量1.2倍触发负载均衡操作，单点访问频率超过集群平均

Tera

架构



数据分布

使用按行键、列名和时间戳全局排序的三维数据模型组织数据，其中RowKey、ColumnFamily、Qualifier和Value是字符串，Timestamp是一个64位整形。ColumnFamliy需要建表时指定，是访问控制、版本保留等策略的基本单位。

```
map<RowKey, map<ColumnnFamily:Qualifier, map<Timestamp, Value> > >
```

Tera

容错

为保证数据安全性，使用三副本存储，但维护副本的一致性与副本丢失恢复需要处理大量细节，基于一个分布式的文件系统构建，可以显著降低开发代价，所以选择使用HDFS。 系统的所有数据都存储在HDFS上，每次写入，保证在HDFS上三副本落地后，才返回成功。

复制与一致性

数据会按key分区存储在HDFS上，分区信息由Master统一管理，Master保证每个分区同一时间只由一个数据节点提供读写服务，保证一致性。

负载均衡

①目标：

从当前tabletnode->tablet的映射集合中通过策略选出部分需要移动的tablet，将其从原tabletnode中unload，并在负载较轻的tabletnode上load起来，并调节各tabletnode上的数据量。

②过程：

第一阶段优先进行读负载均衡，将读热点打散。

- 此tabletnode上发生读请求pending数目超过某阈值
- 只将读热点打散至可服务状态即可
- 只对cpu不足导致的pending，进行迁移
- 迁移时每次选qps第二高的tablet，保证最忙tablet优先得到服务

第二阶段进行数据量均衡，视负载不均的情况可能执行N轮。

- 最大与最小数据量tabletnode比值超过某阈值
- 迁移前校验迁移结果，防止出现数据量反转，形成迁移的死循环
- 迁移时选择最有可能将数据量均衡的tablet进行迁移，防止迁移过多小tablet

Tera

存储

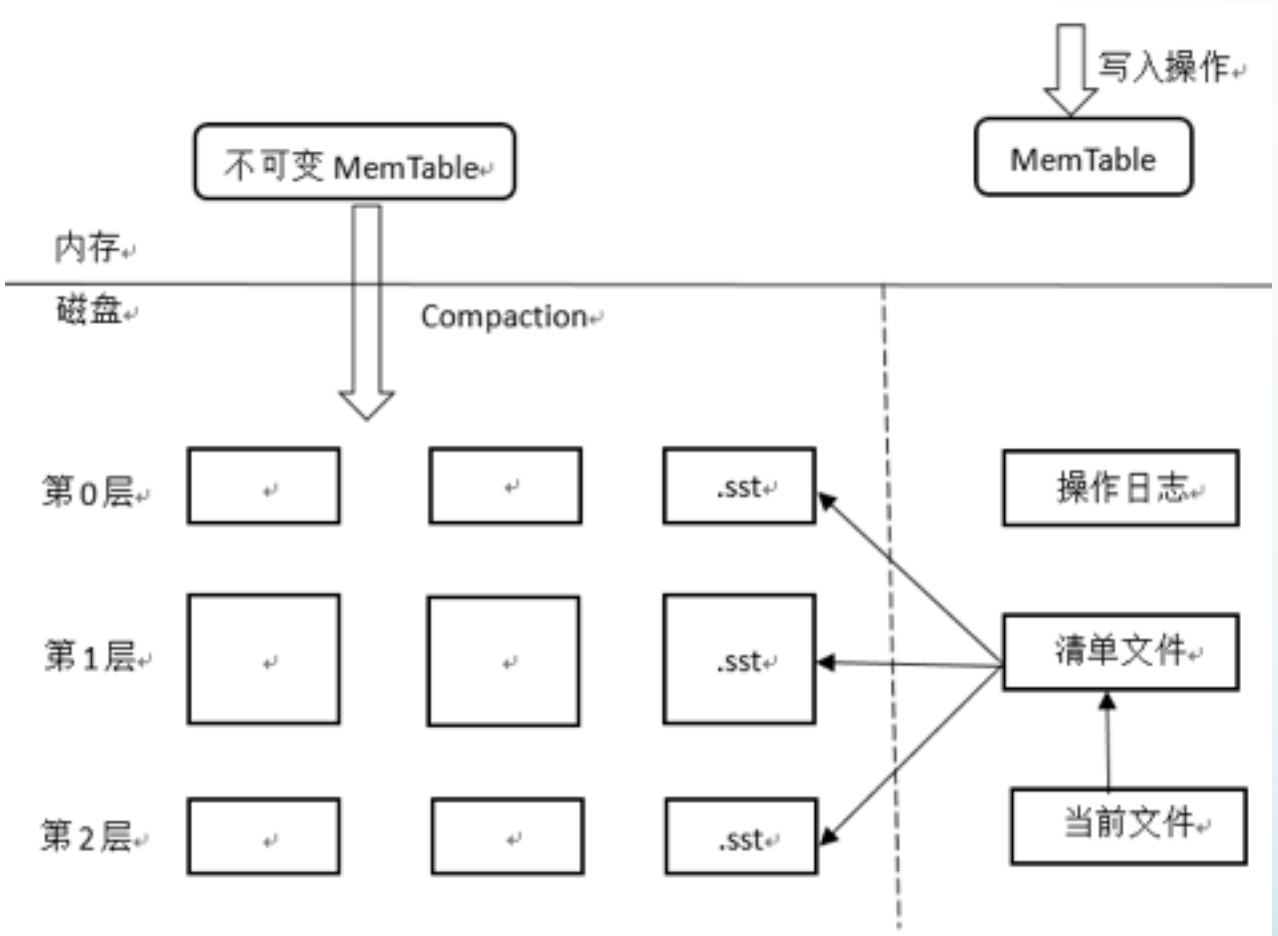
tera的底层存储引擎采用了基于leveldb优化后的key-value存储。 通过将表格内容平展为key-value结构进行存储。 表格中rowkey/columnfamily/qualifier/timestamp统一编码为一个key，结合value进行存储。平展化后，同一行的数据存储在一起，方便进行前缀压缩，一行数据不会被分配至不同的表格分片中。Tera中的数据删除采用标记删除方式，通过后台compact完成数据的物理删除

	age	weight	country	language:en	language:cn
John		54KG	USA	yes	
Lilei	17		China		yes
Toshi	19	60KG	Japan	no	

```
John:country:USA
John:language:en:yes
John:weight:54KG
Lilei:age:17
Lilei:country:China
Lilei:language:cn:yes
Toshi:age:19
Toshi:country:Japan
Toshi:language:en:no
Toshi:weight:60KG
```

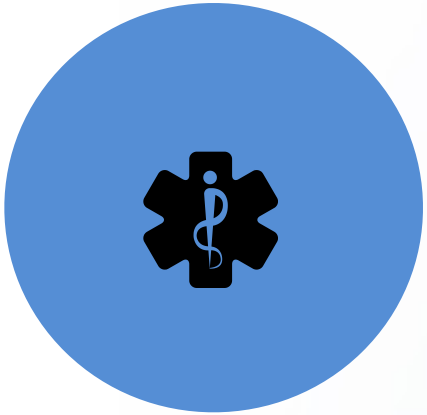
Tera

Leveldb



第三部分





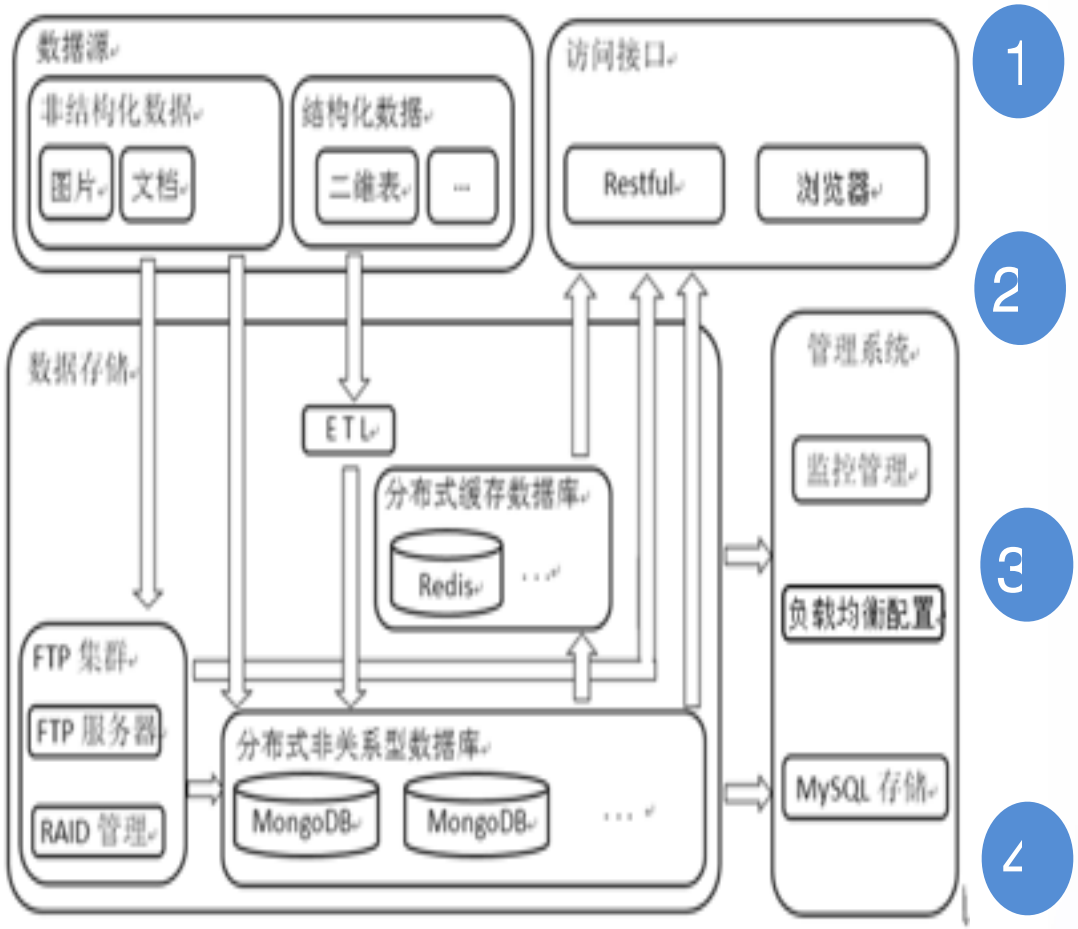

分布式储存基础

大规模分布式存储架构

小规模分布式存储架构

腾讯MongoDB托管平台建设历程

小规模分布式存储架构



数据源

提供所需要数据的器件或原始媒体，分为结构化数据和非结构化数据

访问接口

用户可以使用这组方法向分布式数据存储系统发送服务请求、信息和数据，分布式数据存储系统中的各层依次响应，最终完成数据传输。

数据存储

数据记录在计算机内部或外部存储介质上，分布式数据存储系统中由FTP集群、分布式缓存数据库、分布式非关系型数据库这三个协同工作来完成存储的任务。

管理系统

为系统的管理者提供一个便利和高效的管理平台，同时在一定程度上实现了系统负载均衡和监控管理的功能。

小规模分布式存储架构

1 Redis

高性能的、基于键值对的缓存与存储系统，其通过提供多种键值数据类型来适应不同场景下的缓存与存储需要。

2 树形结构（流水线同步）

在分布式数据存储系统中Redis集群采用树形结构，将数据同步的压力逐层分担，主数据库上的同步压力不会随着集群的不断扩展而成几何倍数的增长。

3 分片机制

通过分片机制增加了自身的可扩展性，避免了单个Redis服务器成为存储瓶颈的问题。

4 乐观复制

主数据库执行完客户端请求的命令会立即将命令在主库的执行结果返回给客户端，异步地将命令同步给从数据库，不会等待从数据库接收到该命令后再返回给客户端。

小规模分布式存储架构

Redis共有五种对象的类型：

类型常量	对象的名称
REDIS_STRING	字符串对象
REDIS_LIST	列表对象
REDIS_HASH	哈希对象
REDIS_SET	集合对象
REDIS_ZSET	有序集合对象

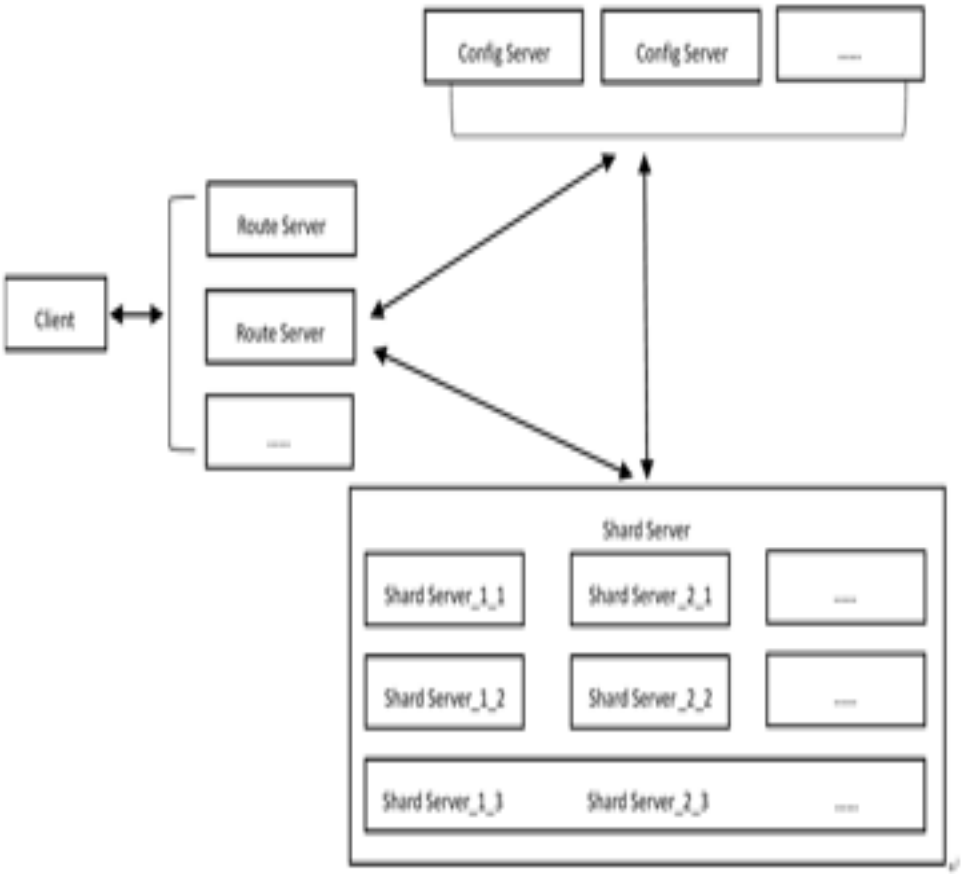
Redis中的对象的结构体：

```
/*
 * Redis 对象
 */
typedef struct redisObject {

    // 类型
    unsigned type:4;

    // 不使用(对齐位)
    unsigned notused:2;

    // 编码方式
    unsigned encoding:4;

    // LRU 时间（相对于 server.lruclock）
    unsigned lru:22;

    // 引用计数
    int refcount;

    // 指向对象的值
    void *ptr;

} robj;
```

Redis对象底层数据结构：

编码常量	编码所对应的底层数据结构
REDIS_ENCODING_INT	long 类型的整数
REDIS_ENCODING_EMBSTR	embstr 编码的简单动态字符串
REDIS_ENCODING_RAW	简单动态字符串
REDIS_ENCODING_HT	字典
REDIS_ENCODING_LINKEDLIST	双端链表
REDIS_ENCODING_ZIPLIST	压缩列表
REDIS_ENCODING_INTSET	整数集合
REDIS_ENCODING_SKIPLIST	跳跃表和字典

小规模分布式存储架构



1

MongoDB

面向文档存储的非关系型数据库。使用“文档”模型，使用记录来表现复杂的层次关系。

2

分布式架构

三副本避免产生单点故障。从节点的开放读分担负载压力。备份节点不对外开放操作，保证数据回滚和维护的安全性。多备份节点共同部署在同一设备上，节约成本的同时，增加备份的灵活性以及安全性。

3

哨兵

监控主数据库和从数据库是否正常运行、主数据库出现故障时自动将从数据库转换为主数据库。

4

Wired Tiger引擎

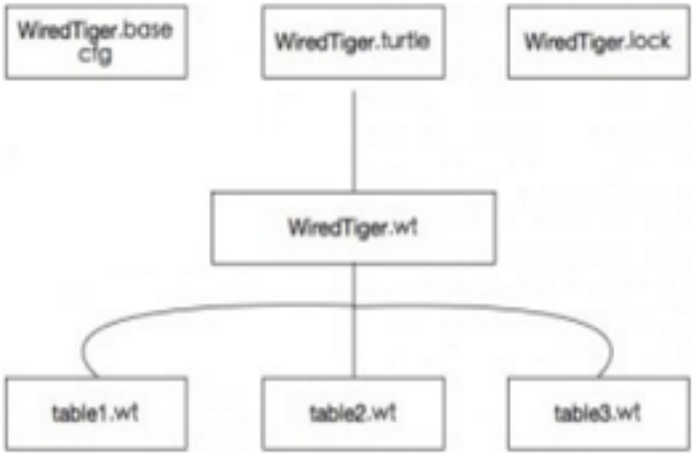
Wired Tiger存储引擎代替MMAP存储引擎，实现文档级别的锁机制，且支持对存储的所有集合和索引进行压缩。

Wired Tiger引擎

1. Checkpoint
2. Btree
3. Copy on write

4.

WiredTiger.basecfg	存储基本配置信息
WiredTiger.lock	用于防止多个进程连接同一个WiredTiger数据库
table*.wt	存储各个table（数据库中的表）的数据
WiredTiger.wt	是特殊的table，用于存储所有其他table的元数据信息
WiredTiger.turtle	存储WiredTiger.wt的元数据信息
journal	存储Write ahead log



小规模分布式存储架构

1 数据库协同

基于Redis的分布式缓存数据库和基于MongoDB的分布式非关系型数据库之间的协同工作机制是保证其性能实现“1+1大于2”的重要保证。为了降低这种多层重定向模式对请求的响应时间的额外消耗，当分布式非关系型数据库返回数据的同时，也将数据同时写入缓存数据库中，那么当再次请求同一份数据时，访问请求将在分布式缓存数据库取得该数据。故能够大大提高对热数据的响应效率，同时也降低了分布式非关系型数据库服务器的负载。

2 延时刷新机制

进行图片存储时，连接消耗的时间会较长，容易在并发高峰时，影响其他服务，对此通过构建FTP集群作为图片的中转站，采用延时刷新机制对其进行物理隔离，在不影响图片存储性能的同时，避免了对整体系统负载的严重影响。

3 重定向查询

底层数据存储采用分布式的架构设计，数据的存储方式是以类似“块”存储的方式进行的，当集群中出现节点数据量不均衡时，必然需要进行拆分大的数据块。重定向查询机制，就是针对在数据进行内部迁移（或是系统自动触发，或者人工干预）时，通过标记搬迁任务所涉及的数据块，可以保证不管数据的状态如何都可以将正确的结果返回给用户。

第四部分

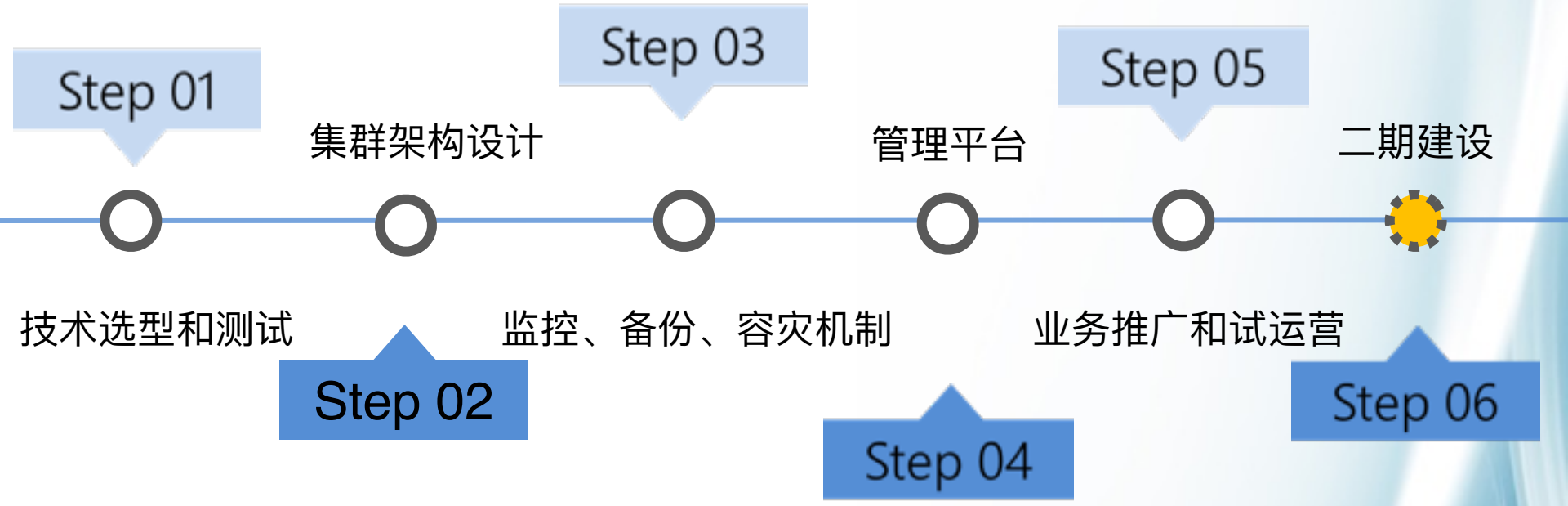
分布式储存基础

大规模分布式存储架构

小规模分布式存储架构

腾讯MongoDB托管平台建设历程

腾讯MongoDB托管平台建设历程



Thank you for watching