360搜索 so.com | IT大咖说 不止于技术

# 360搜索技术论坛长期交流群

360搜索


# 分布式存储实践

陈宗志

# 介绍

- 2013年加入360
- 主要项目
  - Bada 类Dynamo 分布式存储
  - Pika 兼容Redis 协议Nosql
  - Zeppelin 海量数据存储, 支持pb, s3协议
  - 基础库: Pink, Slash, floyd 等等

# 大纲

- 在线存储
- 了解硬件
- 基础理论
- 存储引擎
- zeppelin 设计

# 离线存储

- 离线存储
  - 吞吐
  - 成本

# 在线存储

- 在线存储
  - 性能
  - 稳定性
  - 接口

# 在线存储



- key-value
- 表格存储
- 块存储
- S3(Simple Storage Service)
- 图数据库
- new-sql
- …

# Bada

- key-value store
- 云盘场景: 所有云盘用户文件索引, 通过bada 查询文件所在存储节点 写入近百亿, 峰值QPS20万, 数据总量千亿, 单条value 64字节, 数据总量20 T
- onebox场景: 存储关键字的rank 信息在bada中, 也无需要在4个机房数据同步, 单机房亿级别访问量

# ceph

- 块存储
- 虚拟机提供块存储服务
- 线上500 台虚拟机

# pika

- 大容量, 支持redis 接口的Nosql
- 360 内部500+ 实例, 数据大小10T.
  - 数据分析业务之前7个redis 集群, 每个20G, 每次分析挨个遍历redis 集群. 迁移pika 以后非常方便
  - 手助存用户下载信息库, 历史遗留问题存在redis 30G. 担心主库挂重灌对用户影响. 迁移pika 以后故障秒切
- 外部公司大量使用

360 新浪微博 weibo.com Garena connecting the dots APUS Zol ffan.com

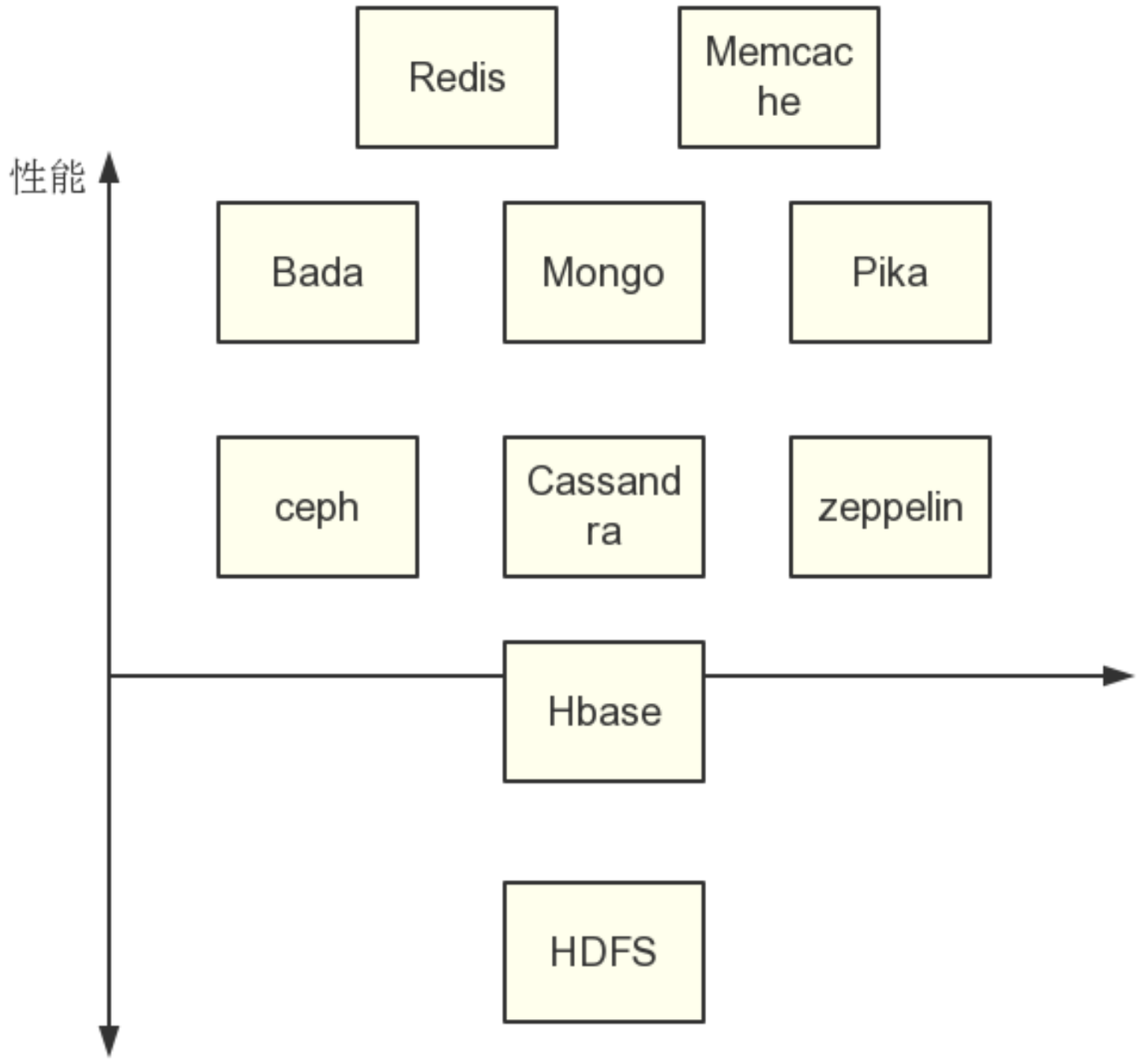
美团网 meituan.com 学而思网校 环信 easemob.com 高伟达 Global InfoTech

oiYiOa 泛微 EMOR亿玛

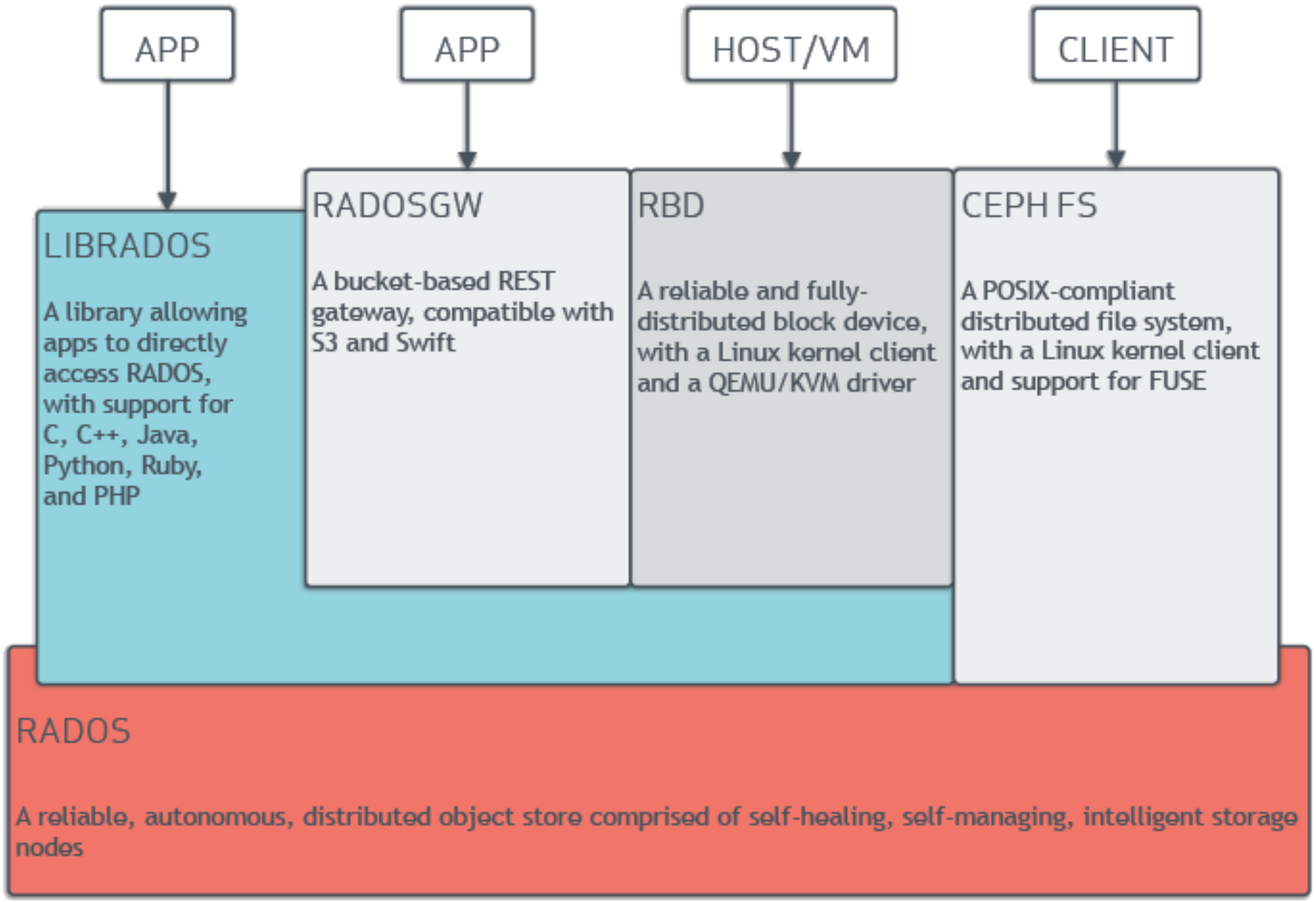
# zeppelin



- key-value store, S3(Simple Storage Service)
- 存储PB 级别数据
- 场景: docker 容器镜像存储
- 低成本


性能
Redis
Memcac
he
Bada
Mongo
Pika
ceph
Cassand
ra
zeppelin
Hbase
HDFS

# ceph


APP
APP
HOST/VM
CLIENT
LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP
RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift
RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver
CEPH FS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE
RADOS
A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

# facebook

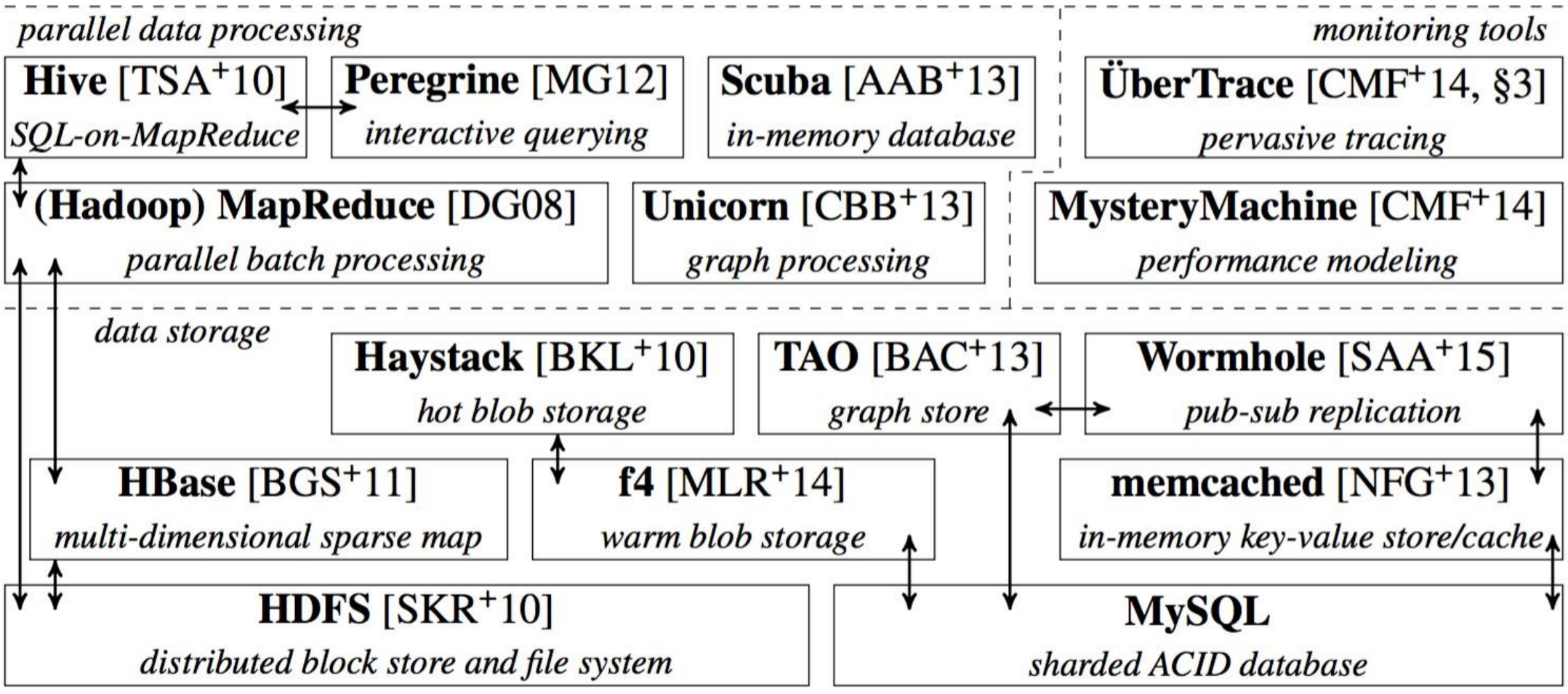



**Figure 1:** The Facebook infrastructure stack. I omit front-end serving systems about which details are unknown. Arrows indicate data exchange and dependencies between systems; simple layering does *not* imply a dependency or relation.

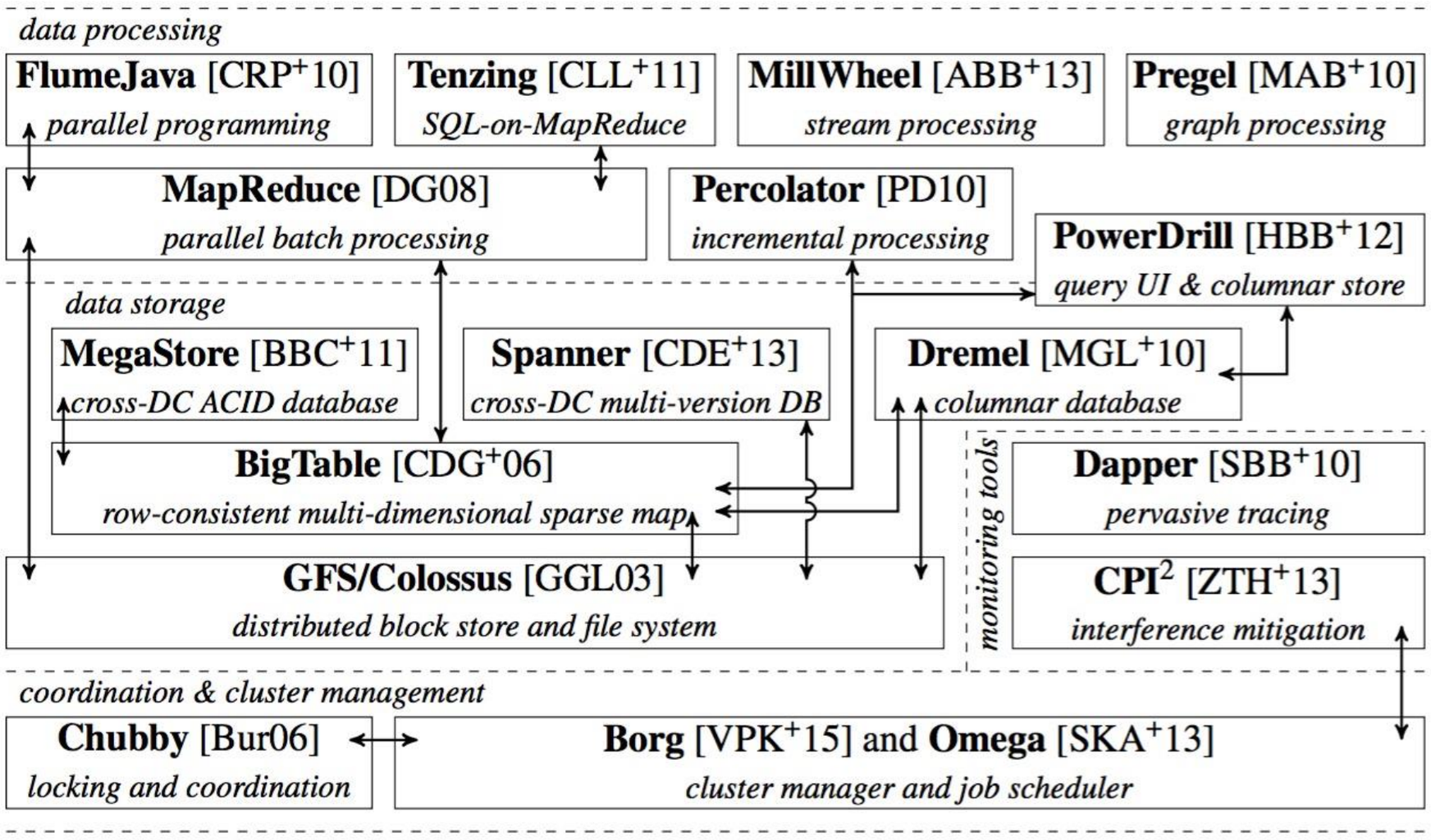



**Figure 1:** The Google infrastructure stack. I omit the F1 database [SOE+12] (the back-end of which was superseeded by Spanner), and unknown front-end serving systems. Arrows indicate data exchange and dependencies between systems; simple layering does *not* imply a dependency or relation.

# Performance

# Performance

1. Response time: 完成一个任务需要的时间
2. Throughtput: 指定时间内可以完成的任务数
3. Average VS Quantile

**1 Two Lists of Response Times (average for each is 1.000 second**

| | List A | List B |
|---|---|---|
| 1 | .924 | .796 |
| 2 | .928 | .798 |
| 3 | .954 | .802 |
| 4 | .957 | .823 |
| 5 | .961 | .919 |
| 6 | .965 | .977 |
| 7 | .972 | 1.076 |
| 8 | .979 | 1.216 |
| 9 | .987 | 1.273 |
| 10 | 1.373 | 1.320 |

# Numbers Everyone Should Know

| | |
|---|---|
| L1 cache reference | 0.5 ns |
| Branch mispredict | 5 ns |
| L2 cache reference | 7 ns |
| Mutex lock/unlock | 25 ns |
| Main memory reference | 100 ns |
| Compress 1K bytes with Zippy | 3,000 ns |
| Send 2K bytes over 1 Gbps network | 20,000 ns |
| Read 1 MB sequentially from memory | 250,000 ns |
| Round trip within same datacenter | 500,000 ns |
| Disk seek | 10,000,000 ns |
| Read 1 MB sequentially from disk | 20,000,000 ns |
| Send packet CA->Netherlands->CA | 150,000,000 ns |

**Jeff Dean** [Designs, Lessons and Advice from Building Large Distributed Systems]

| CATEGORY | REPRESENTATIVE DEVICE | SEQUENTIAL READ BAND-WIDTH | SEQUENTIAL WRITE BAND-WIDTH | 4KB READ IOPS | 4KB WRITE IOPS |
|---|---|---|---|---|---|
| Mechani-cal disk | Western Digital Black WD4001FAEX (4TB) | 130MB/s | 130MB/s | 110 | 150 |
| SATA-at-tached SSD | Samsung 850 Pro (1TB) | 550MB/s | 520MB/s | 10,000 | 36,000 |
| PCIe-at-tached SSD | Intel 750 (1.2TB) | 2,400MB/s | 1,200MB/s | 440,000 | 290,000 |
| Main memory | Skylake @ 3200MHz | 42,000MB/s | 48,000MB/s | 16,100,000 (62ns/op-eration) | |

(In the above table, all IOPS figures are reported assuming a queue depth of 1, so will tend to be worst case numbers for the SSDs.)

Max Bolingbroke **Data structures for external memory**

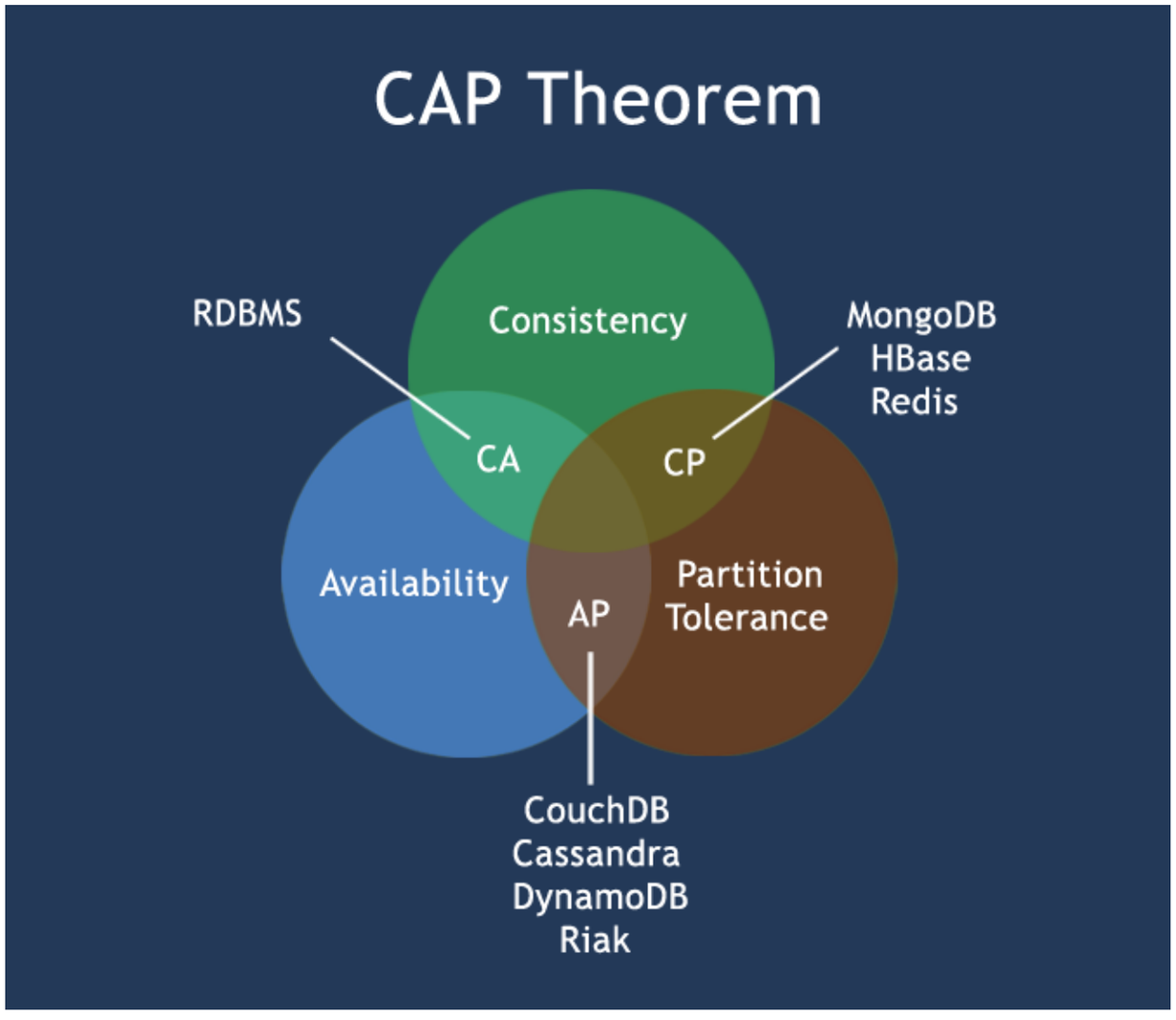
| CATEGORY | IMPLIED SEEK TIME FROM READ | IMPLIED SEEK TIME FROM WRITE | MEAN IMPLIED SEEK TIME |
|---|---|---|---|
| Mechanical Disk | 9.06ms | 6.63ms | 7.85ms |
| SATA-attached SSD | 92.8us | 20.2us | 56.5us |
| PCIe-attached SSD | 645ns | 193ns | 419ns |

# Example

- 对于SATA 盘的zeppelin, 写入一条256K大小数据延迟
  - 9.06 ms + 256k * 20 ms/MB = 14.06 ms
- bada 北京机房与上海机房之间同步延迟极限
  - 1500km / (3 * 10^8) * 2  = 10ms

# CAP

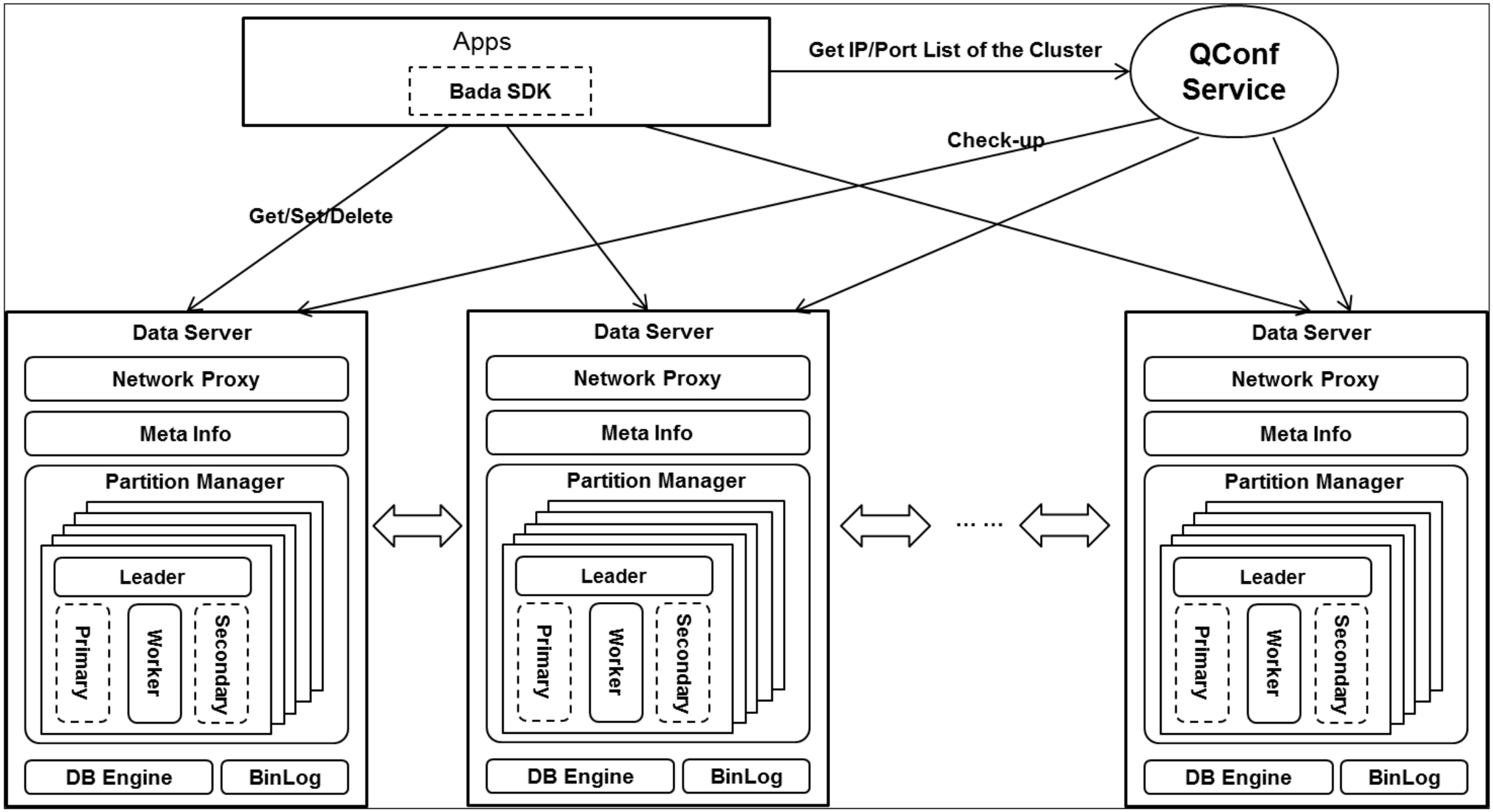
Eric Brewer 1998年提出
Consistency
Availability
Partition Tolerance



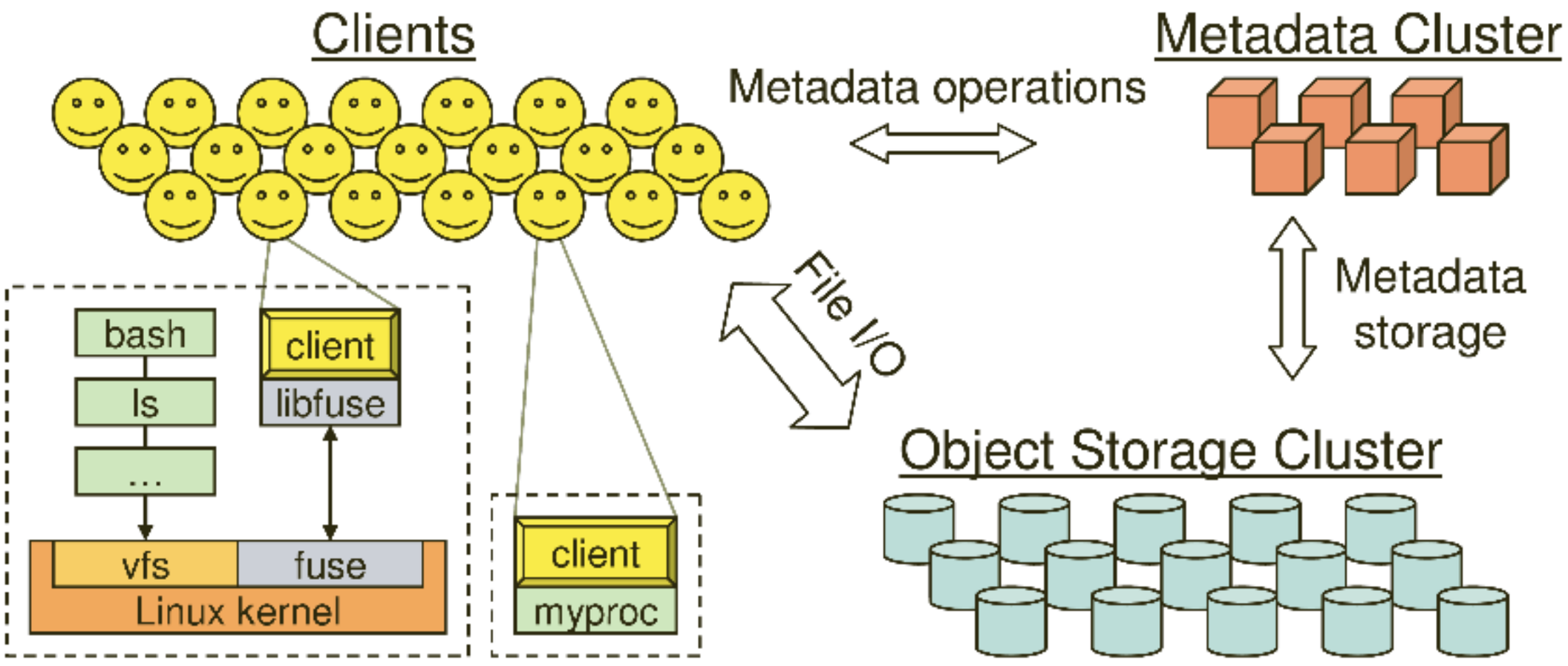
# 基础理论

- 中心节点设计
  - 有中心 VS 去中心
- 副本策略
  - 3副本主从 VS Quorum VS erasure code
- 数据分布策略
  - consistent hash vs range map
- 一致性协议
  - 弱一致, 最终一致, 强一致
  - multi-paxos vs raft

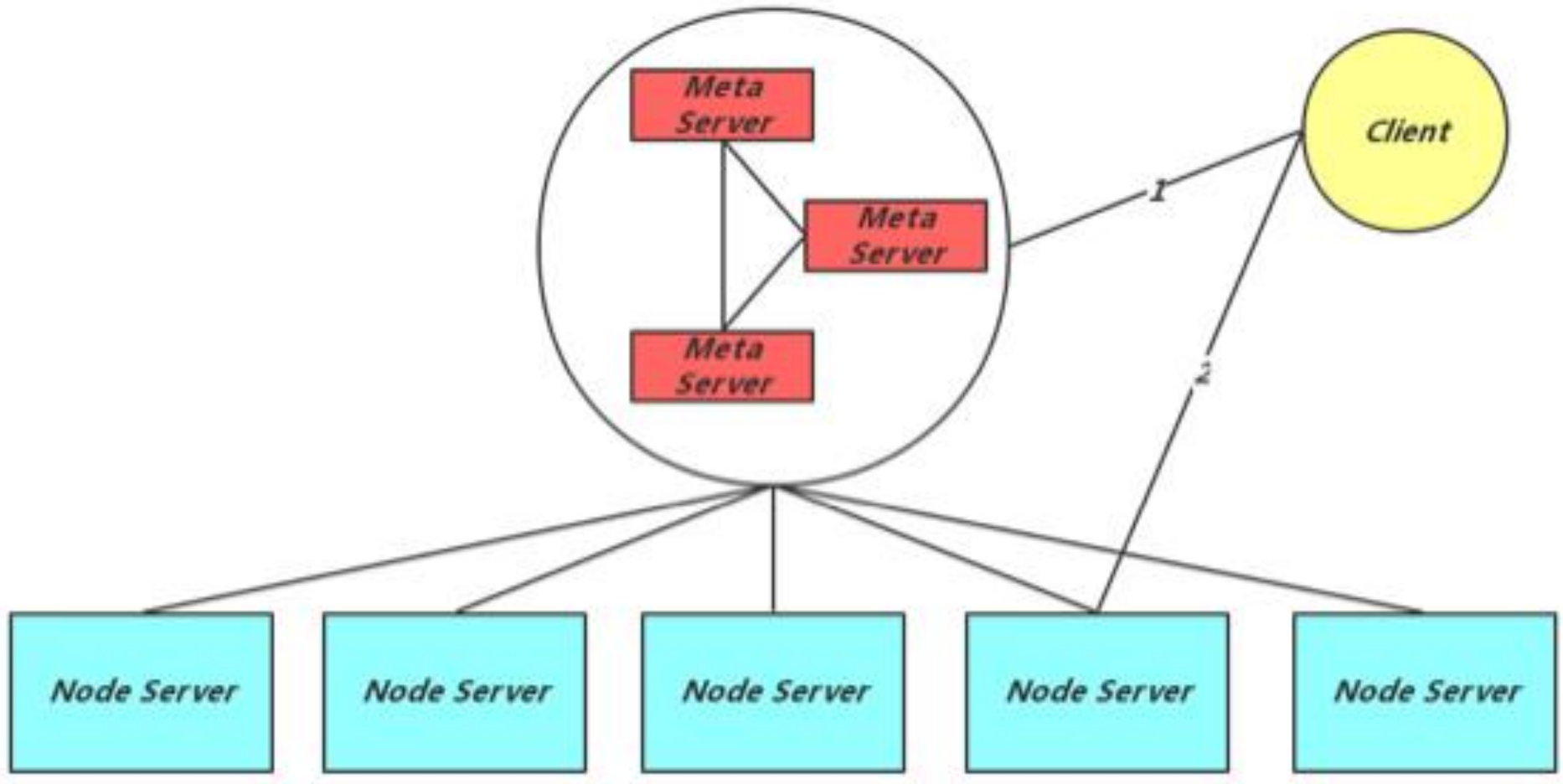
# Bada


Apps
Bada SDK
Get IP/Port List of the Cluster
QConf Service
Check-up
Get/Set/Delete
Data Server
Network Proxy
Meta Info
Partition Manager
Leader
Primary
Worker
Secondary
DB Engine
BinLog
Data Server
Network Proxy
Meta Info
Partition Manager
Leader
Primary
Worker
Secondary
DB Engine
BinLog
... ...
Data Server
Network Proxy
Meta Info
Partition Manager
Leader
Primary
Worker
Secondary
DB Engine
BinLog

# ceph

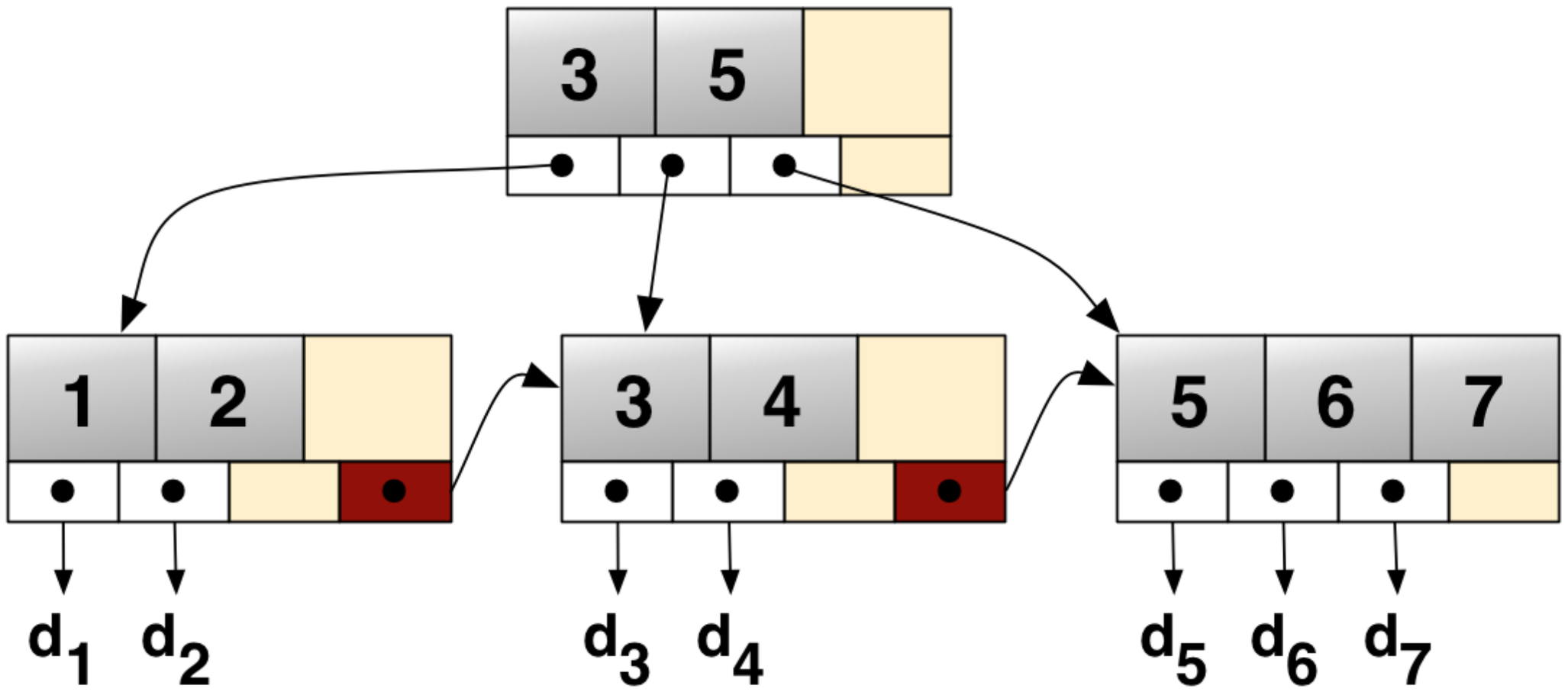

Clients
Metadata operations
Metadata Cluster
File I/O
Metadata
storage
Object Storage Cluster
bash
ls
...
client
libfuse
vfs
fuse
Linux kernel
client
myproc

# zeppelin


Meta Server
Meta Server
Meta Server
Client
1
Node Server
Node Server
Node Server
Node Server
Node Server

# 存储引擎

# B+ Tree



# LSM Tree

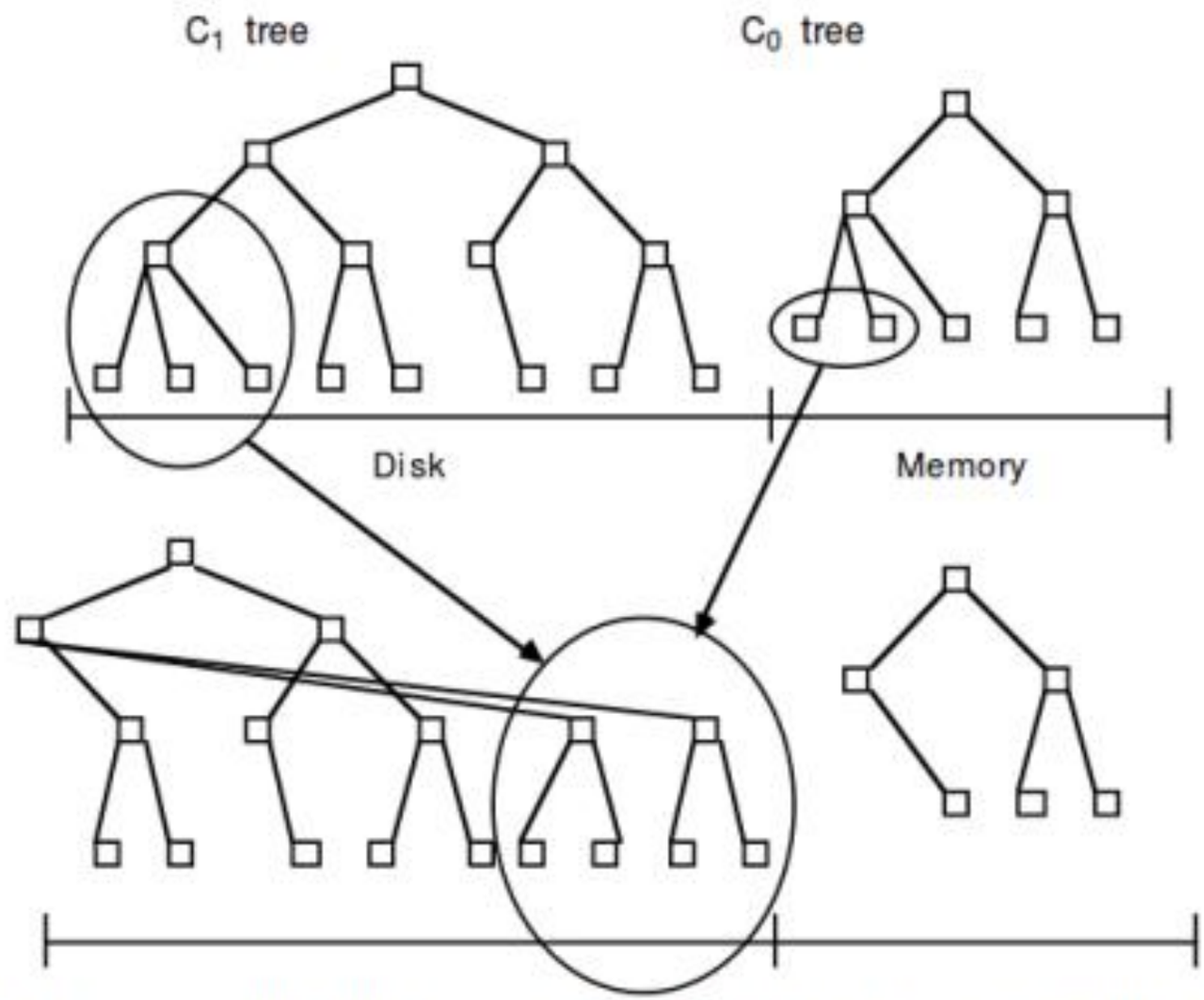
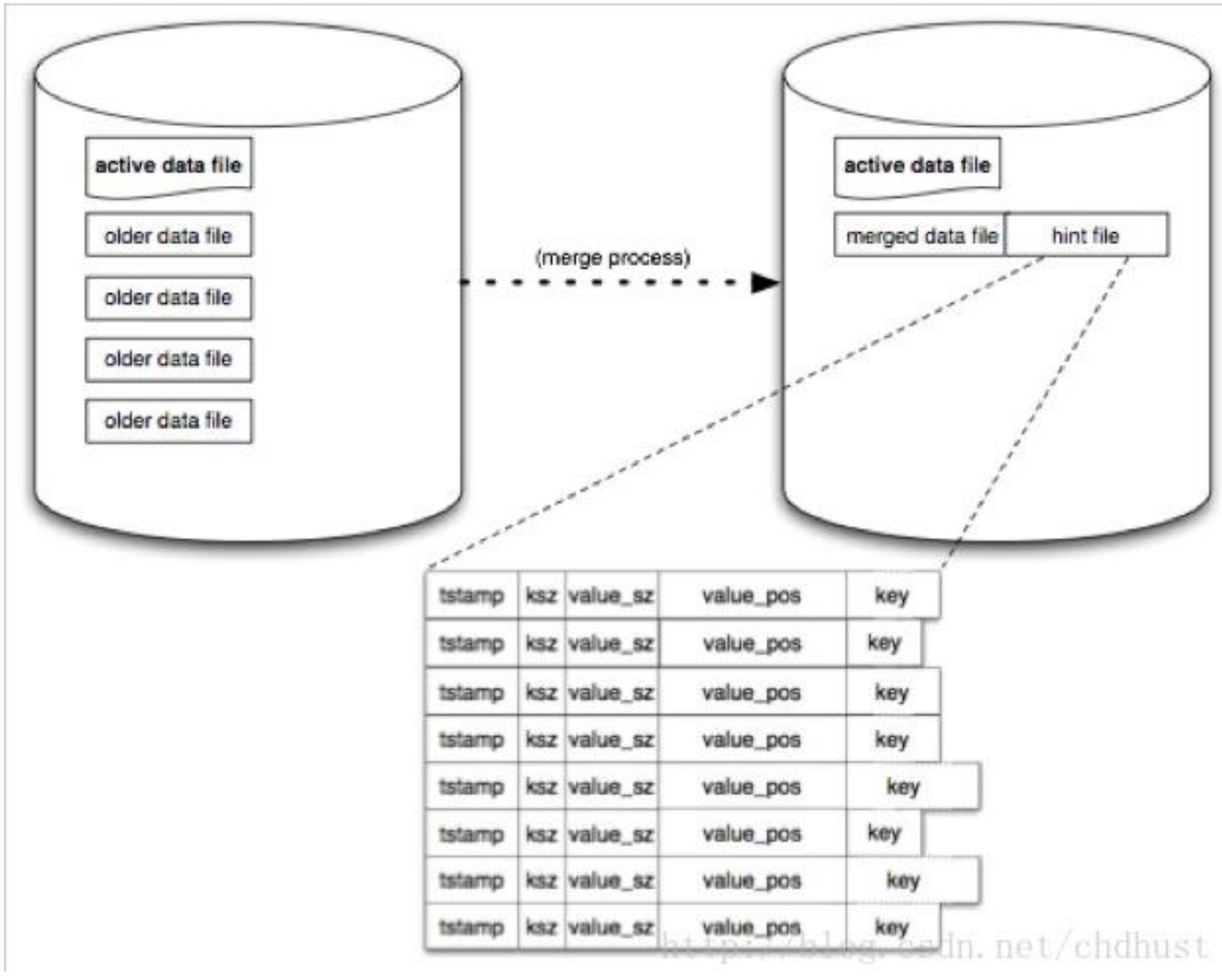



**Figure 2.2.** Conceptual picture of rolling merge steps, with result written back to disk

# bitcask


active data file
older data file
older data file
older data file
(merge process)
active data file
merged data file
hint file
tstamp ksz value_sz value_pos key
tstamp ksz value_sz value_pos key
tstamp ksz value_sz value_pos key
tstamp ksz value_sz value_pos key
tstamp ksz value_sz value_pos key
tstamp ksz value_sz value_pos key
tstamp ksz value_sz value_pos key
tstamp ksz value_sz value_pos key

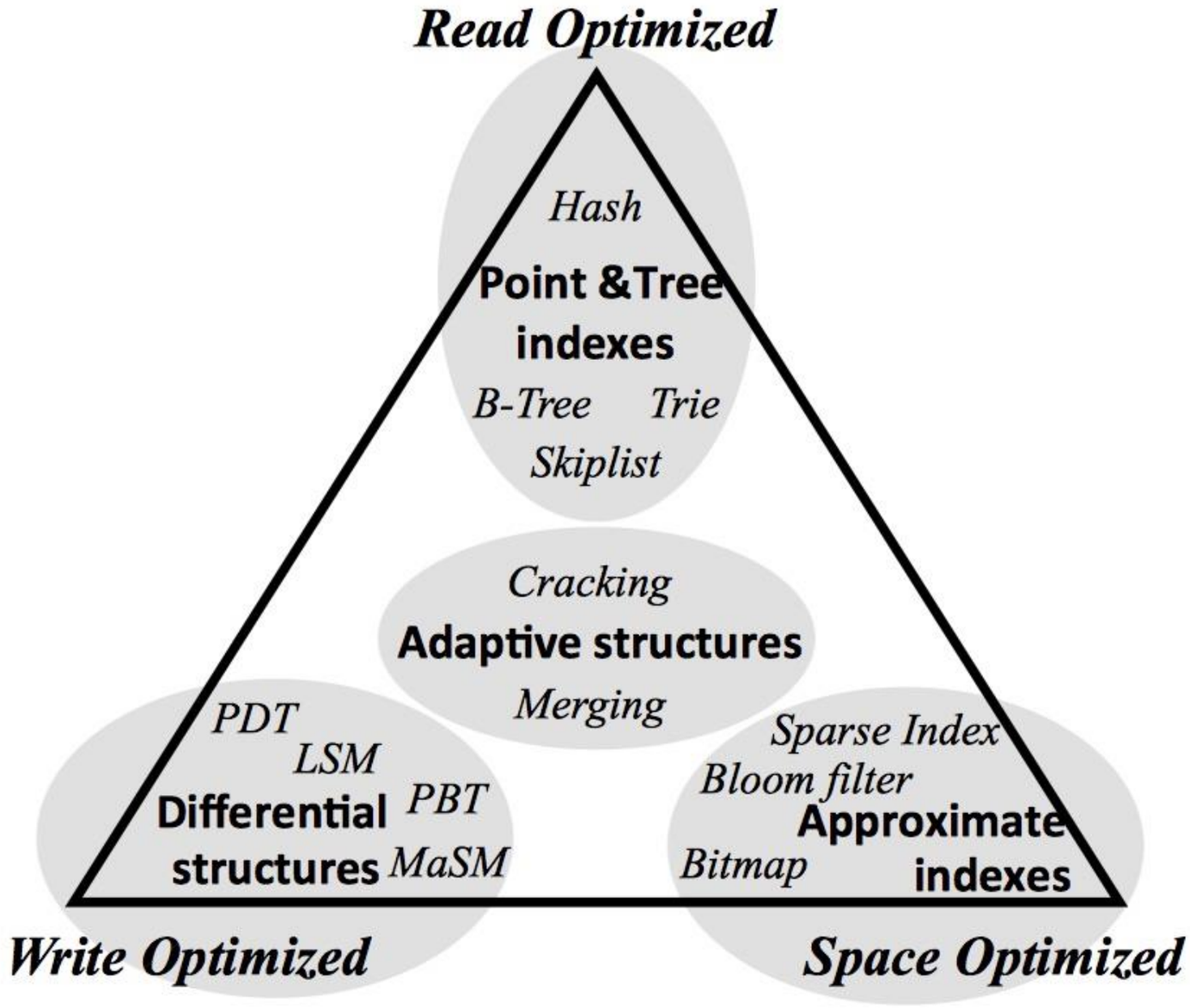



**Figure 1: Popular data structures in the RUM space.**

Manos Athanassoulis [Designing Access Methods: The RUM Conjecture]

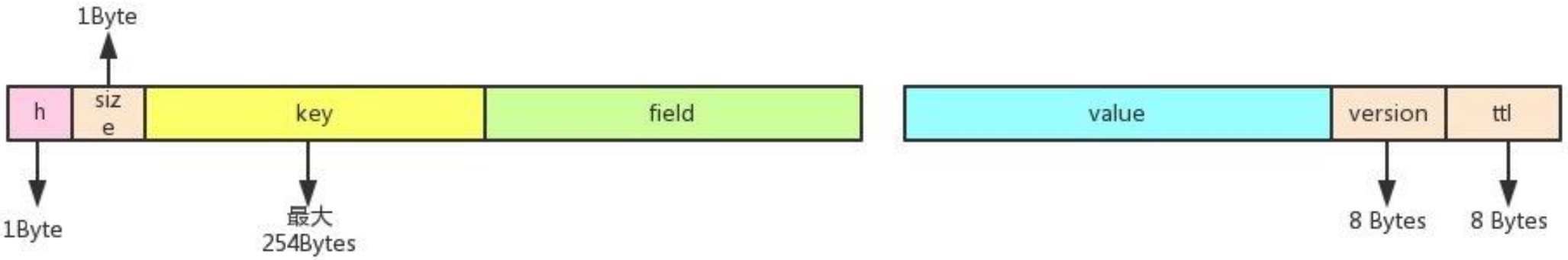
| Data Structure | Write Amp (worst case) | Read Amp (range) (cold cache) | Read Amp (range) (warm) | Space Amp |
|---|---|---|---|---|
| B Tree | $O(B)$ | $O(\log_B N/B)$ | 1 | 1.33 |
| FT index | $O(k\log_k N/B)$ | $O(\log_k N/B)$ | 1 | negligible |
| LSM leveled | $O(k\log_k N/B)$ | $O((\log^2 N/B)/\log k)$ | 3 | 2 (file-per-run)<br>1.1 (many files) |
| LSM size-tiered | $O(\log_k N/B)$ | $O(k(\log^2 N/B)/\log k)$ | 13 | 3 |

# 存储引擎

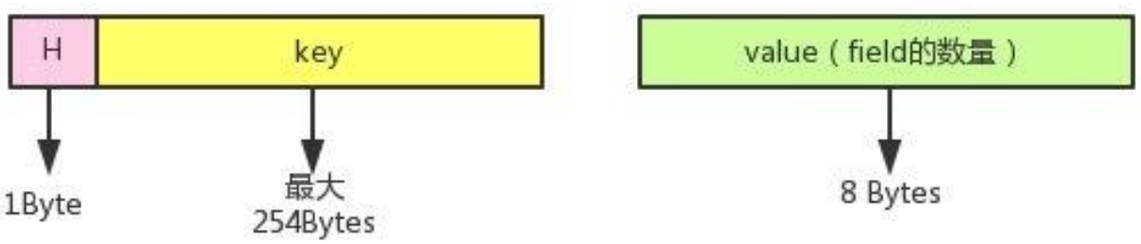
- Nemo
  - 基于rocksdb key-value 接口, 实现hash, list, set, zset 等接口
  - https://github.com/Qihoo360/nemo
- Meepo
  - Bitcask 存储引擎实现
  - https://github.com/baotiao/meepo

# nemo-Hash

a.每个hash键、field、value到落盘kv的映射转换




b.每个hash键的元信息的落盘kv的存储格式



# zeppelin

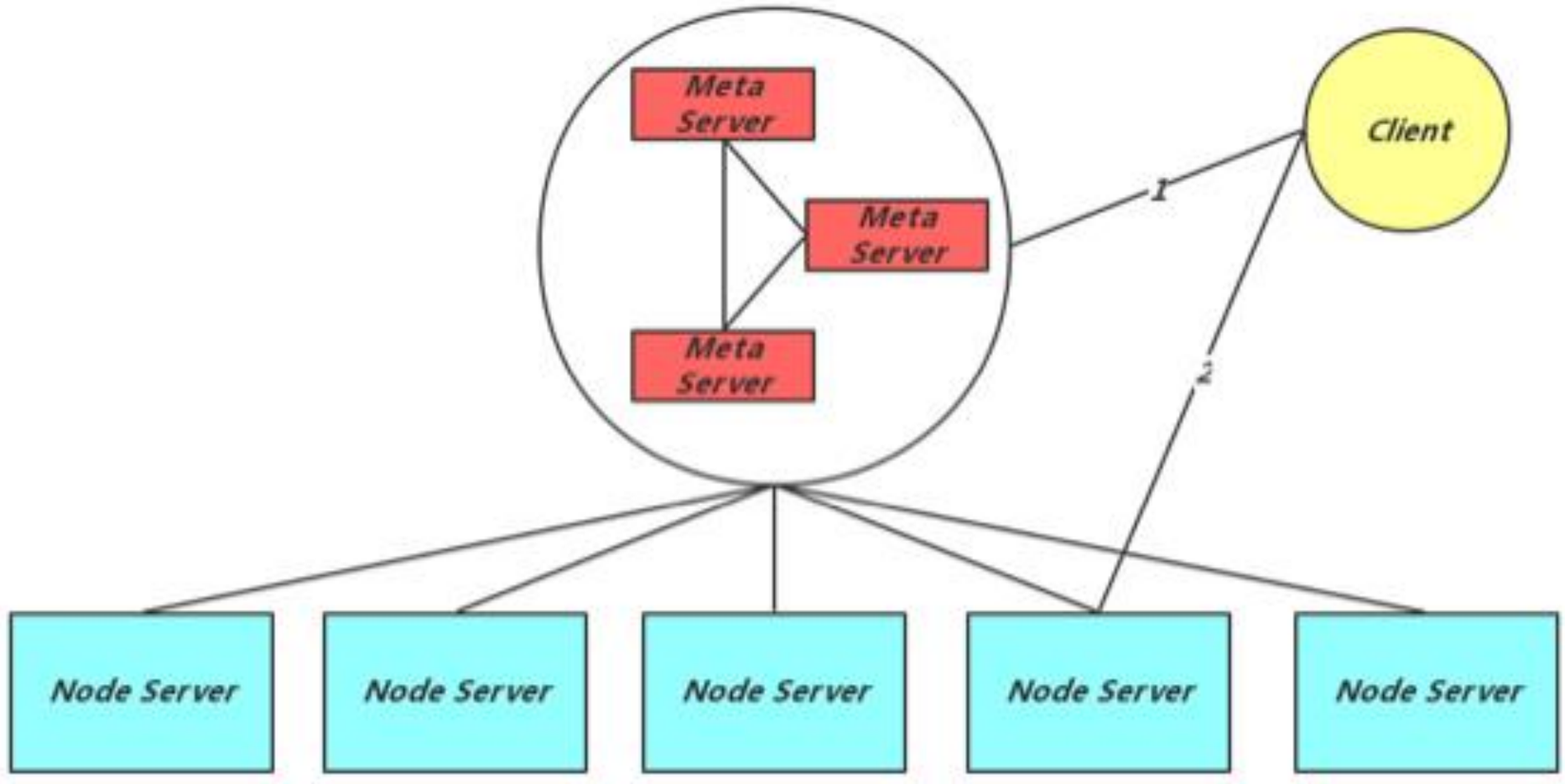
# zeppelin

- 目标支持单机群千台
- 最终一致性
- 存储半离线数据

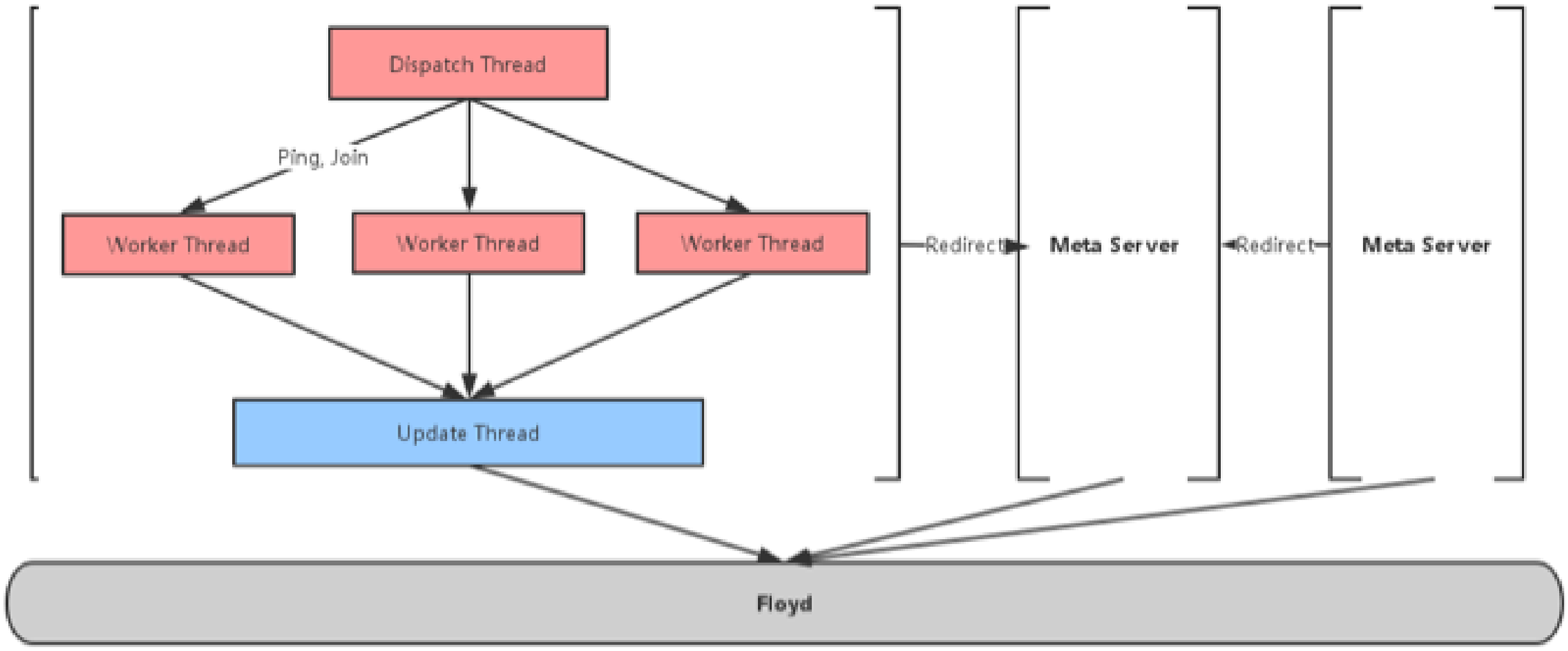
# Pink

- 基础架构团队开发网络编程框架
- 高性能
  - 单节点支持80w
- 稳定
  - pika, zeppelin, gpstall, mongosync, 稳定使用3年
- 易用
- https://github.com/Qihoo360/pink

# 整体结构


Meta Server
Meta Server
Meta Server
Client
1
Node Server
Node Server
Node Server
Node Server
Node Server

# Meta 设计


Dispatch Thread
Ping, Join
\Worker Thread
\Worker Thread
\Worker Thread
Update Thread
Redirect
Meta Server
Redirect
Meta Server
Floyd

# 分表

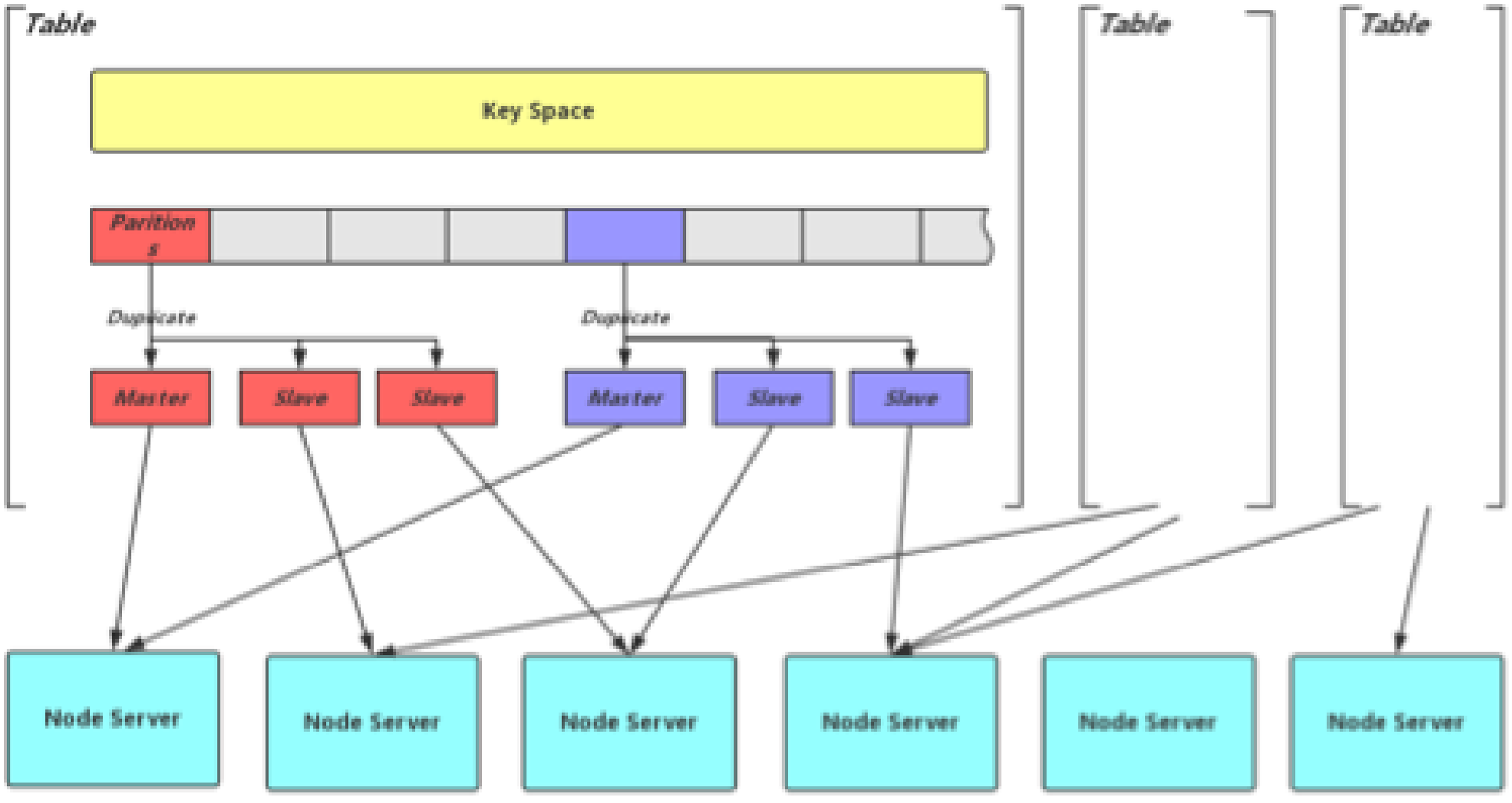
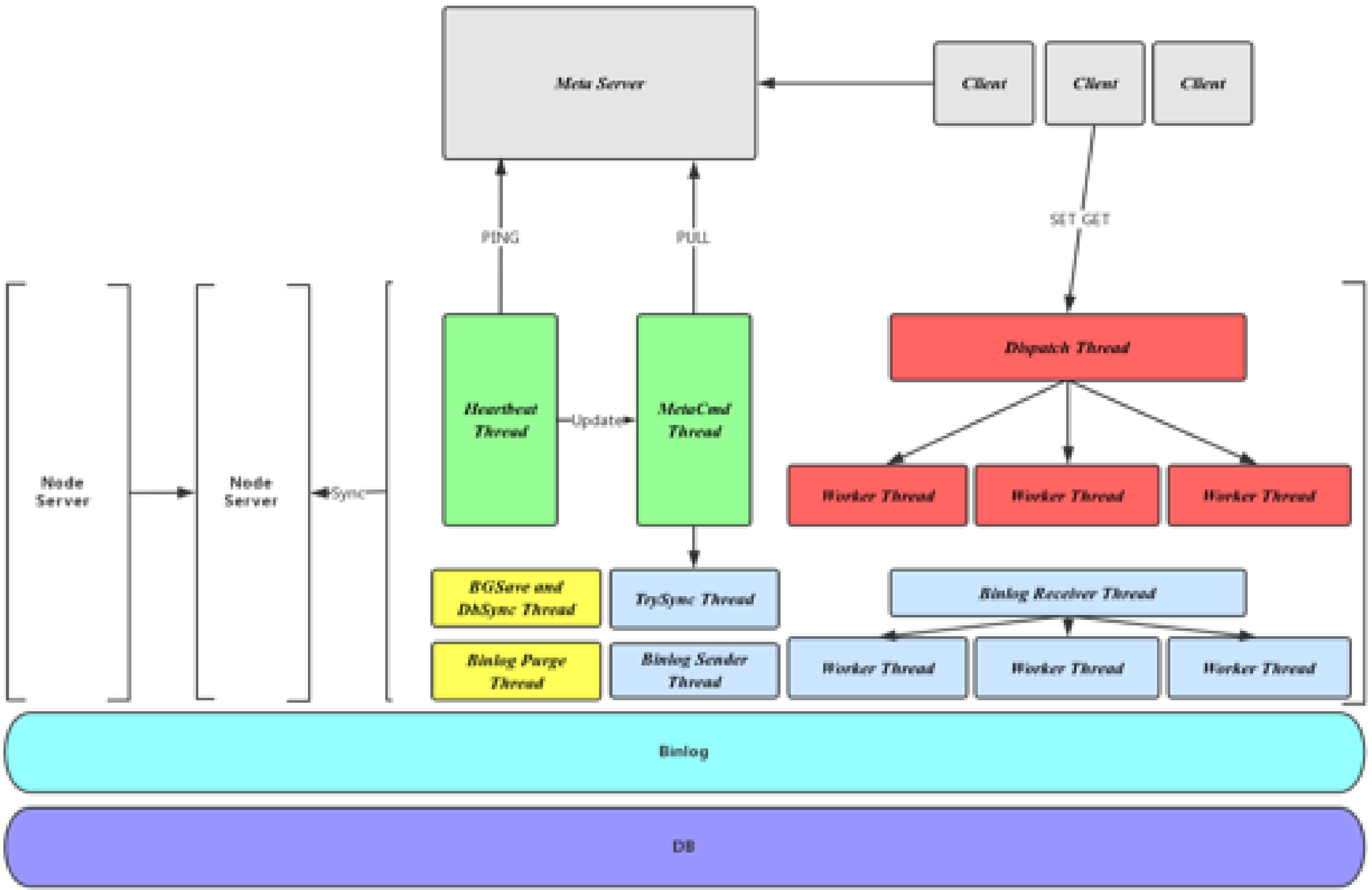


Table
Key Space
Partitions
Duplicate
Duplicate
Master
Slave
Slave
Master
Slave
Slave
Table
Table
Node Server
Node Server
Node Server
Node Server
Node Server
Node Server


Meta Server
Client
Client
Client
PING
PULL
SET GET
Node Server
Node Server
Sync
Heartbeat Thread
Update
MetaCmd Thread
Dispatch Thread
Worker Thread
Worker Thread
Worker Thread
BGSave and DbSync Thread
TrySync Thread
Binlog Receiver Thread
Binlog Purge Thread
Binlog Sender Thread
Worker Thread
Worker Thread
Worker Thread
Binlog
DB

# 开源项目

- https://github.com/Qihoo360
- https://github.com/Qihoo360/pika
- https://github.com/Qihoo360/pink
- https://github.com/baotiao/zeppelin

# Further reading:



- Ghemawat, Gobioff, & Leung. Google File System, SOSP 2003.
- G. DeCandia, et al., Dynamo: Amazon's Highly Available Key-Value Store
- B. Cooper, et al., PNUTS: Yahoo!'s Hosted Data Serving Platform
- L. Lamport, Paxos Made Simple
- Diego Ongaro, In Search of an Understandable Consensus Algorithm
- F. Chang, et al., Bigtable: A Distributed Storage System For Structured Data