架构实战营 - 模块4

第5课：常见存储系统剖析

李运华

前阿里资深技术专家(P9)

教学目标

1. 学习如何快速学习存储系统
2. 掌握 Redis、HBase、Clickhouse、HDFS 的技术本质

掌握技术原理，把握系统本质

目录

如何学习存储系统？

学习步骤

1.理解
技术本质

理解系统的核心技术本质，技术本质决定了应用场景和性能量级。

【案例】
1. Redis 是 K-V 存储系统
2. HBase 是 sorted map

2. 明确
部署架构

学习存储系统支持的部署架构，明确其架构的本质

【案例】
1. Redis 支持3种部署架构
2. HBase 只有1种部署架构

3. 研究
数据模型

研究存储系统提供的数据模型，包含哪些概念，如何应用

【案例】
1. Redis 支持多种数据结构
2. HBase 的 table，column 等

4. 模拟
业务场景

模拟一些常见业务场景，完整的实现一个案例，并测试其性能

【案例】
1. 如何用 Redis 来存储关注关系？
2. 如何用 HBase 存储关注关系？

每次都要自己测试性能的话，很耗费时间和精力，怎么办？

如何学习官方文档 - 按图索骥

1.理解
技术本质

概要介绍：https://hbase.apache.org/
技术本质：https://hbase.apache.org/book.html#_architecture

2. 明确
部署架构

架构介绍：https://hbase.apache.org/book.html#_architecture
官方文档没有架构图，可以直接搜索图片 HBase architecture，有很多文章基于官方文档画的架构图

3. 研究
数据模型

数据模型：https://hbase.apache.org/book.html#datamodel
schema 设计：https://hbase.apache.org/book.html#schema

4. 模拟
业务场景

案例学习：https://hbase.apache.org/book.html#casestudies

Redis

Redis 介绍 - Remote Dictionary Server

Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache, and message broker.

【技术本质 解读】
1. in-memory：意味着性能高，但同时意味着数据持久化不是核心，可能丢数据
2. data structure store：数据结构存储，而不是关系数据，也不是文件存储

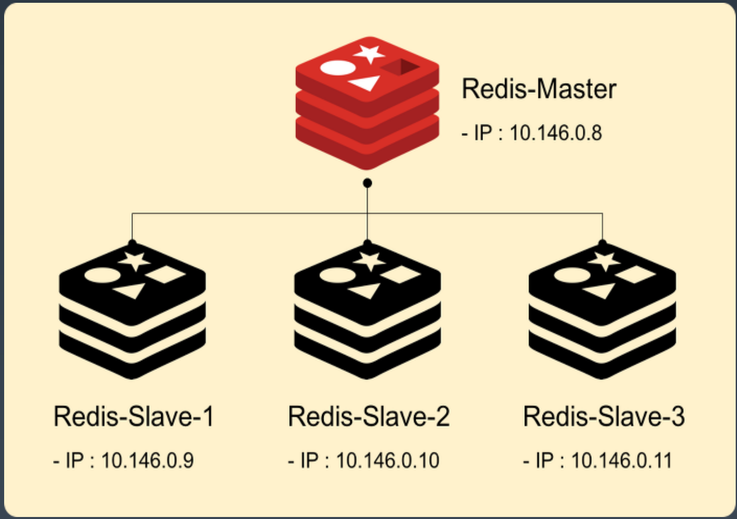
【用途】
database，cache，message broker

【性能量级】
单机 TPS 5~10万

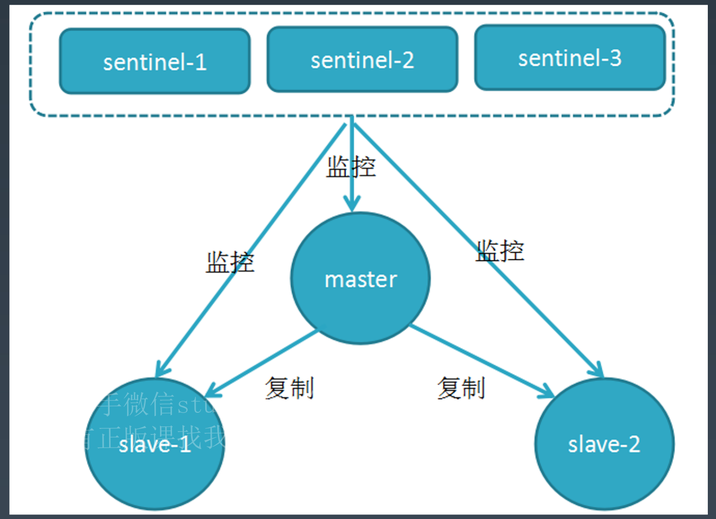
【相关知识】
关系数据：数据之间的关系非常密切，互相依赖和影响，核心特征就是读的时候 join，写的时候用事务保证一致性
非关系数据：数据之间关系疏松，互相独立，数据间的一致性要求很低

用 Redis 存储机票数据是否可以？

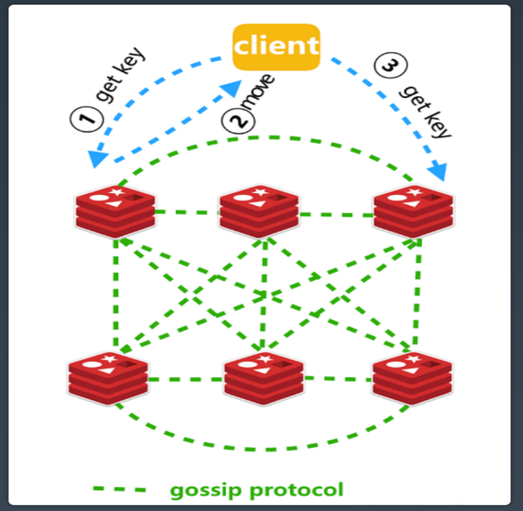
Redis 部署架构



【技术本质】
1. 主从复制，读写分离
2. 无自动切换功能



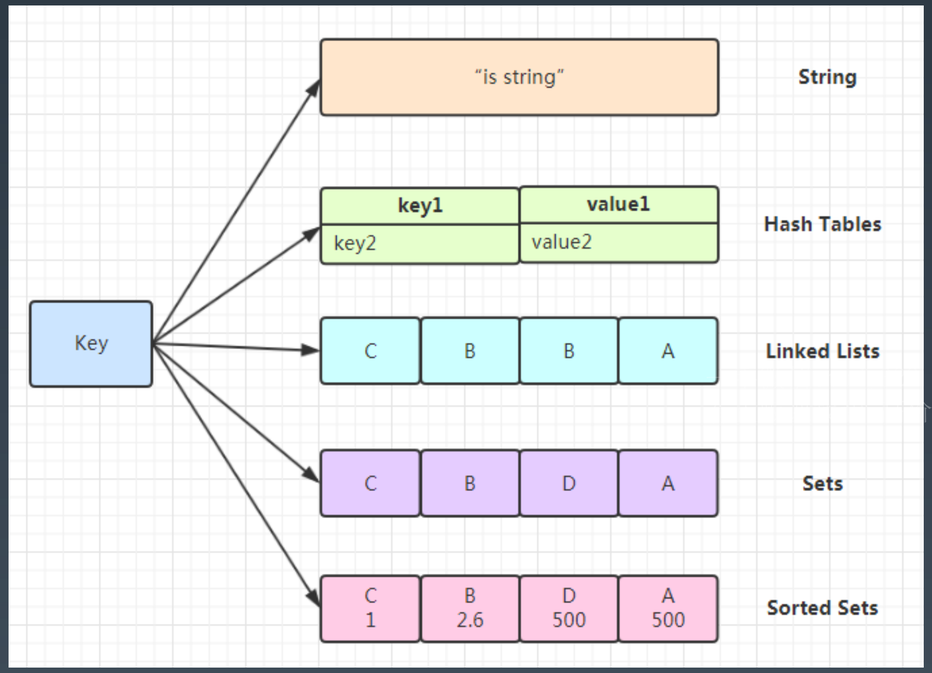
【技术本质】
1. 主从复制，读写分离
2. Master 故障时 sentinel 自动切换



【技术本质】
1. 分片集群

技术本质决定应用场景，技术细节决定具体方案！

Redis 数据模型



【总体结构】

K-V 存储，单个查找快，不支持范围查找

【数据结构】

1. String 类型
2. Hashtable：hash 表
3. Linked list：可重复，插入顺序排序
4. Set：不可重复，无序
5. Sorted set：不可重复，按照 score 排序

模拟业务场景 - 用 Redis 实现关注列表存储

【方案1】
1. Key：用户 ID + follower
2. Value：选择 List，List 是有序的，可以重复

【具体方案】
1. 新增关注：需要扫描 List 判断是否重复，不重复则尾部追加
2. 取消关注：需要扫描 List 找到粉丝 ID 然后删除
3. 拉黑：和取消关注一样

【方案分析】
新增关注和取消关注都需要扫描整个 List，性能较低，某些爆红的账户会有性能问题

【方案2】
1. Key：用户 ID + follower
2. Value：选择 Sorted set，有序但不能重复

【具体方案】
1. 新增关注：使用关注时的 timestamp 作为 score，无需扫描，Redis 会判断是否重复
2. 取消关注：直接删除
3. 拉黑：和取消关注一样

【方案分析】
无论是性能还是实现复杂度，都比List要更优

HBase

HBase 介绍

Apache HBase is an open-source, distributed, versioned, non-relational database modeled after Google's Bigtable: A Distributed Storage System for Structured Data by Chang et al. Just as Bigtable leverages the distributed data storage provided by the Google File System, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS. .

Use Apache HBase™ when you need random, realtime read/write access to your Big Data.

"A Bigtable is a sparse, distributed, persistent multidimensional sorted map"

【技术本质 解读】

1. no-relational：非关系型数据；versioned：多版本的
2. after Bigtable：参考 Bigtable 的原理，multidimensional sorted map
3. on top of Hadoop and HDFS：基于 Hadoop 和 HDFS，底层存储结构是 LSM

【用途】

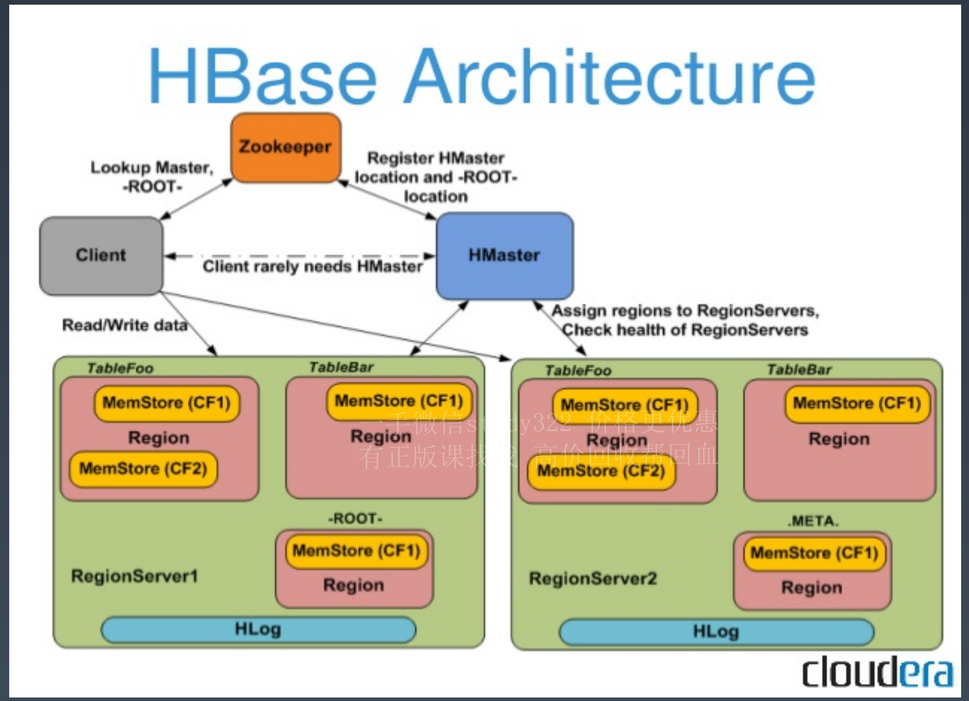
Big Data 存储

【参考性能量级】

四台32核主机每秒插入70000条，读取大约是25000条，扫描100条以内记录，每秒15000条（读取比写入慢？）

到底多大算 Big Data？1亿条数据算么？

HBase 部署架构



【技术本质】

1. 分片集群
2. ZooKeeper 做 HMaster 的切换，RegionServer 由 HMaster 管理

HBase 数据模型

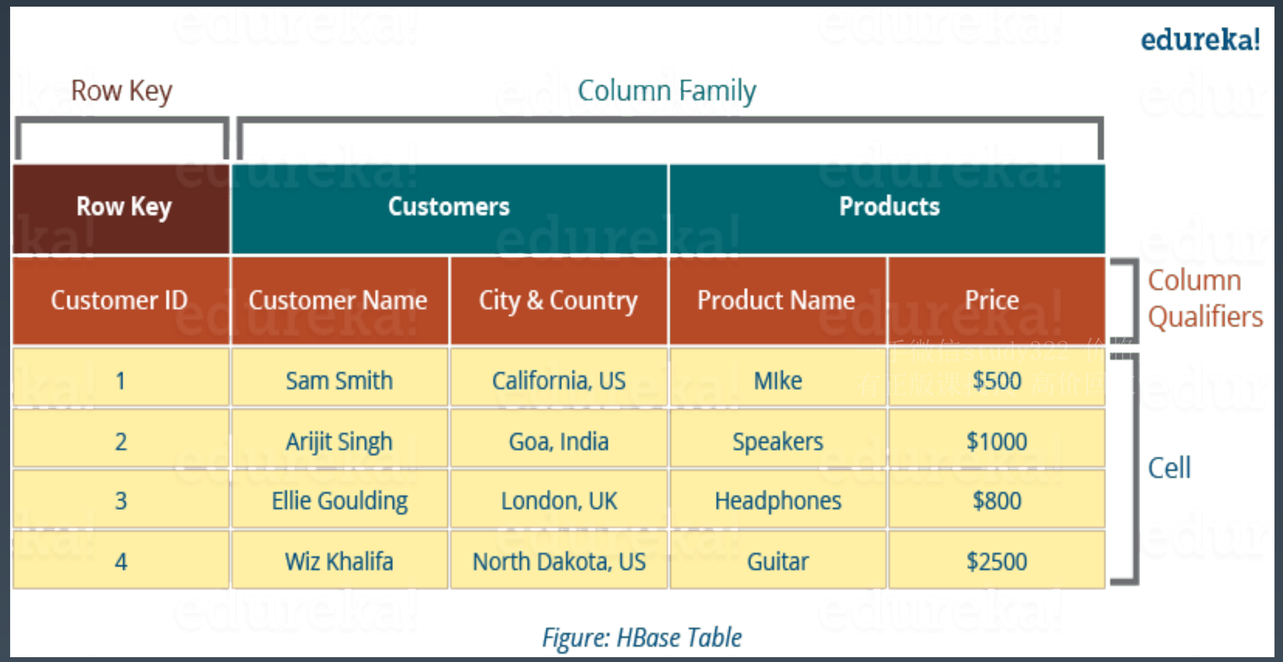


Figure: HBase Table

Table：An HBase table consists of multiple rows

Row：consists of a row key and one or more columns with values associated with them.

Column Family：physically colocate a set of columns and their values, often for performance reasons.

Column：consists of a column family and a column qualifier

Cell：a combination of row, column family, and column qualifier, and contains a value and a timestamp

Timestamp：is written alongside each value, and is the identifier for a given version of a value.

模拟业务场景 - 关注列表存储方案1

	follows				
AK	1:foo	2:bar	3:baz	4:troy	count:4
foo	1:bar	2:AK	count:2		

【方案1】
1. Key：用户 ID
2. Value：关注顺序作为 Column，被关注的用户 ID 作为 value，增加 count 列作为关注总数

【具体方案】
1. 新增关注：先读取 count 列计算出关注顺序，再以此顺序作为 Column
2. 取消关注：需要遍历整个 follows column family，并修改 count列
3. 拉黑：和取消关注一样

【方案分析】
性能太低，需要读取出来整个列表，然后遍历整个列表

模拟业务场景 - 关注列表存储方案2

【方案1】
1. Key：用户 ID + follower ID
2. Value：follower 的全名

【具体方案】
1. 新增关注：直接插入新的一条记录
2. 取消关注：直接删除记录
3. 拉黑：和取消关注一样
4. 查看列表：scan 前缀为“用户 ID”的 key

【方案分析】
1. 读写性能都很好
2. 一个关注关系要一条数据

	f
AK+foo	James Foo:1
AK+bar	Jimmy Bar:1
AK+baz	Ricky Baz:1
AK+troy	Troy:1
foo+bar	Jimmy Bar:1
foo+AK	AK:1

关注数量如何存储？

HDFS

HDFS 介绍

HDFS is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware.

HDFS provides high throughput access to application data and is suitable for applications that have large data sets.

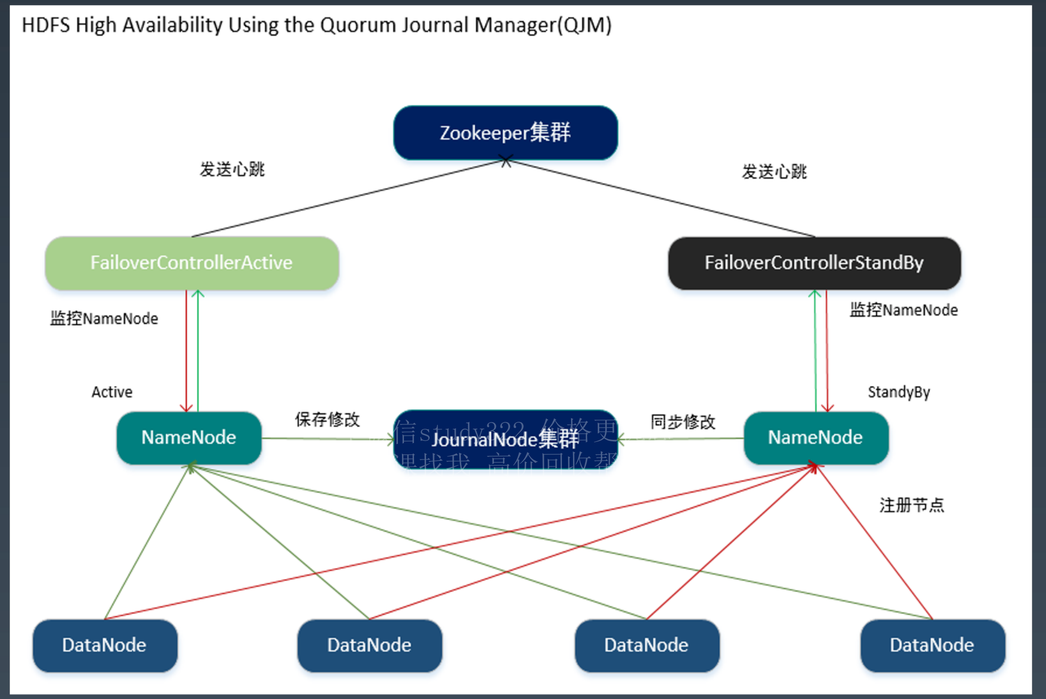
【技术本质 解读】
1. file system：这是文件存储，不是关系数据，也不是数据结构
2. distributed：分布式的文件存储，不是 Linux 上的 ext 文件系统这种
3. low-cost hardware：运行在低成本硬件，而不是 IOE 的高成本硬件

【用途】
large data sets：大数据存储

【参考性能量级】
性能可横向伸缩，瓶颈是带宽

为啥要开发 HDFS?

HDFS 部署架构



【技术本质】

1. 分片集群
2. 大文件存储，不适合存储只有几 K 的小文件

HDFS 数据模型

HDFS supports a traditional hierarchical file organization. A user or an application can create directories and store files inside these directories. The file system namespace hierarchy is similar to most other existing file systems; one can create and remove files, move a file from one directory to another, or rename a file. HDFS does not yet implement user quotas. HDFS does not support hard links or soft links.

简单来说，这就是一个文件系统，你需要自己来规划好目录和文件就可以了。

Clickhouse

Clickhouse 介绍

ClickHouse® is a column-oriented database management system (DBMS) for online analytical processing of queries (OLAP).

【技术本质 解读】
1. column-oriented：列式存储
2. DBMS：数据库管理系统，
3. OLAP：OLAP 场景（MySQL 是 OLTP）

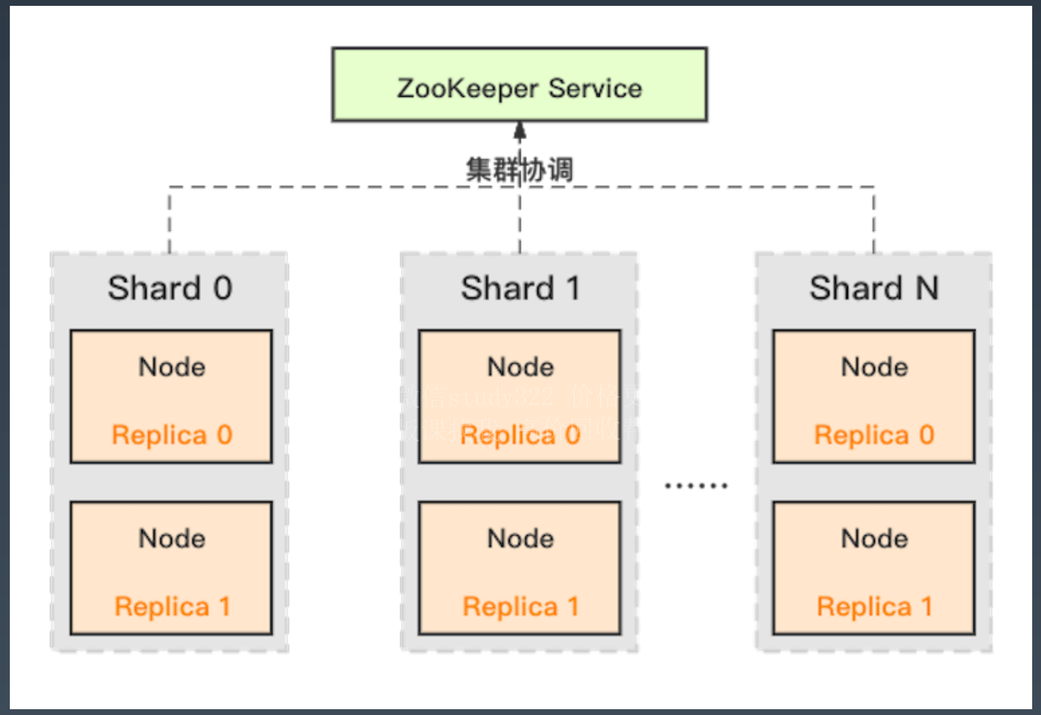
为何SQL不说
读写性能？

【用途】
OLAP

【参考性能量级】
官方有详细的测试对比：https://clickhouse.tech/benchmark/dbms/

【相关知识】
OLTP：联机事务处理，执行大量增删改查，关注响应速度，高并发、数据一致性
OLAP：联机分析处理，执行少量复杂查询，关注吞吐量，很少修改数据
行式存储：表中的一行记录存储在一个数据块中
列式存储：表中的一列数据存储在一个数据块中

Clickhouse 部署架构

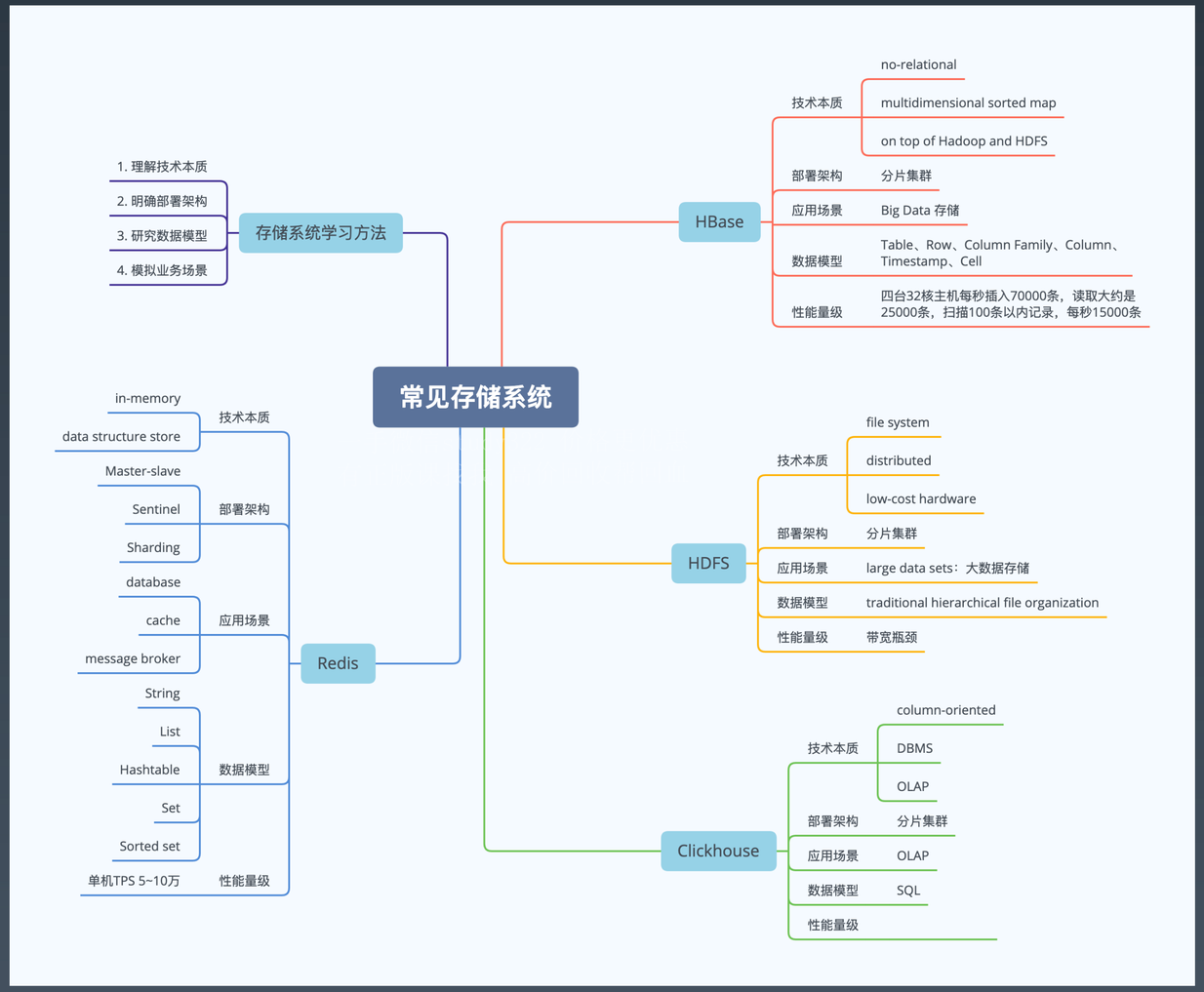


【技术本质】
1. 分片集群
2. ZooKeeper 管理，分片独立复制

Clickhouse 数据模型

基于 SQL 表设计即可

本节思维导图


常见存储系统
存储系统学习方法
1. 理解技术本质
2. 明确部署架构
3. 研究数据模型
4. 模拟业务场景
HBase
技术本质
no-relational
multidimensional sorted map
on top of Hadoop and HDFS
部署架构
分片集群
应用场景
Big Data 存储
数据模型
Table、Row、Column Family、Column、Timestamp、Cell
性能量级
四台32核主机每秒插入70000条，读取大约是25000条，扫描100条以内记录，每秒15000条
HDFS
技术本质
file system
distributed
low-cost hardware
部署架构
分片集群
应用场景
large data sets：大数据存储
数据模型
traditional hierarchical file organization
性能量级
带宽瓶颈
Clickhouse
技术本质
column-oriented
DBMS
OLAP
部署架构
分片集群
应用场景
OLAP
数据模型
SQL
性能量级
Redis
技术本质
in-memory
data structure store
部署架构
Master-slave
Sentinel
Sharding
应用场景
database
cache
message broker
数据模型
String
List
Hashtable
Set
Sorted set
性能量级
单机TPS 5~10万

随堂测验

【判断题】

1. Redis 不适合存储关系型数据
2. HBase 主要适合离线大数据存储
3. HDFS 可以用来存储视频、日志等文件
4. Clickhouse 是 OLAP 系统，可以代替 Hadoop 之类的离线分析平台
5. 存储系统部署架构有多种会更有利于架构设计，可以根据场景灵活应用

【思考题】

Clickhouse 做数据分析和 Hadoop 做数据分析有什么优点？

Q&A

茶歇时间

八卦，趣闻，内幕……

THANKS