


高并发高性能 数据库设计-未来篇



• 讲师：KimmKing •

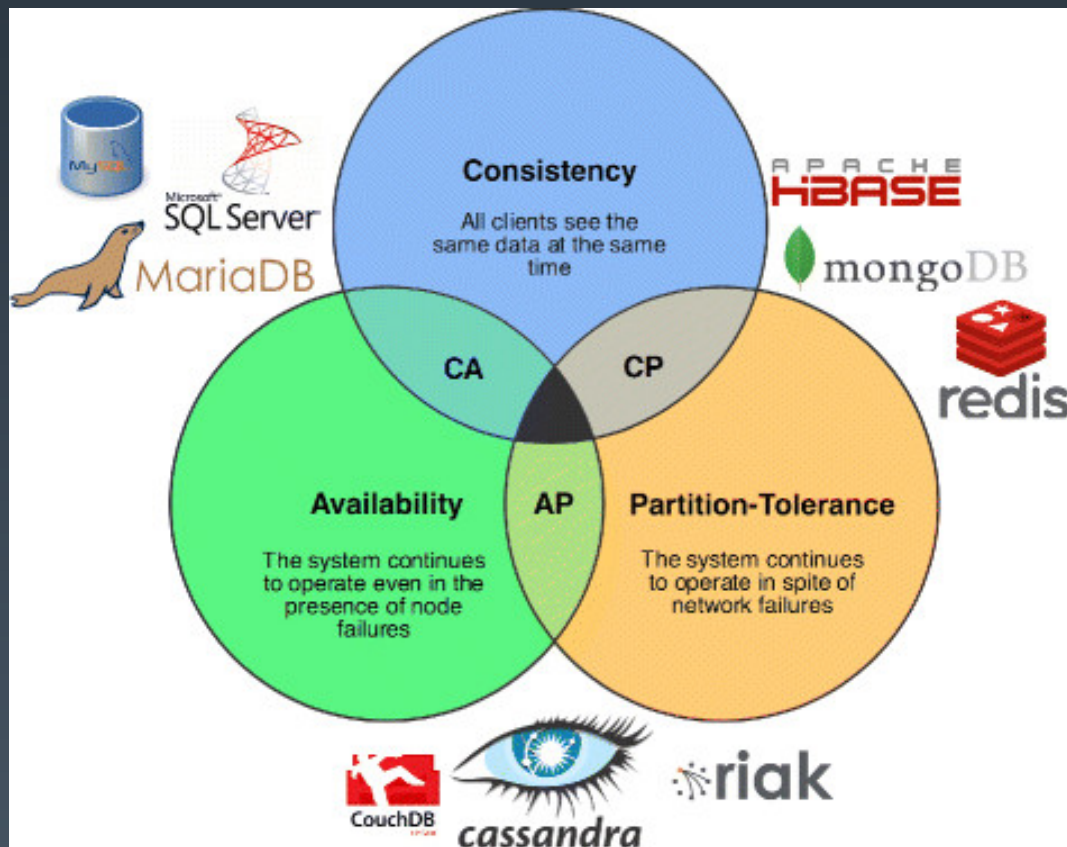
目录 | Contents

1. 分布式数据库
2. 云原生与数据库网格
3. 下一代的分布式数据库技术

一、分布式数据库

摩尔定律失效

分布式崛起

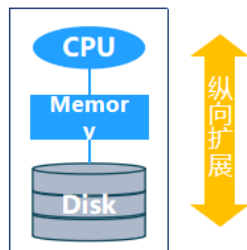


一、分布式数据库

关系数据库 -> 数据中间件 -> 分布式数据库

数据库与大数据的边界在模糊

Share Everything 架构



概念：单机关系型数据库，CPU、内存、磁盘是完全共享的；

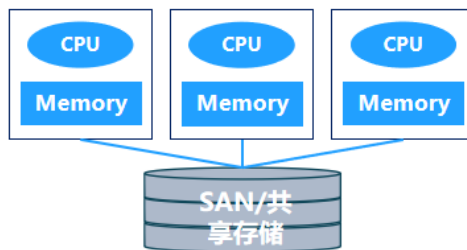
优点：技术成熟、架构简单、部署运维便捷；

缺点：可扩展性差，并行处理能力弱，高可用性不佳，处理数据量小；

场景：适用于OLTP，小规模关系型数据OLAP；

典型代表：Oracle、MySQL；

Share Disk 架构



概念：每台Server都有自己的CPU和内存，但共享磁盘系统；

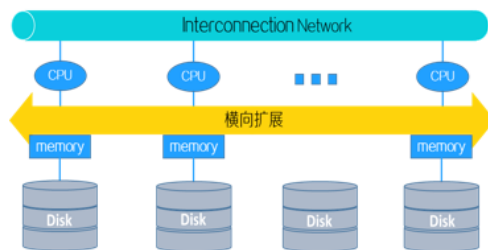
优点：高可用（负载均衡、秒级故障切换、集群自动监控）；

缺点：可扩展性较差，通常不超过10节点，处理数据量较小；

场景：适用于OLTP，小规模关系型数据OLAP；

典型代表：Oracle RAC、DB2 PureScale；

Share Nothing 架构



概念：各个处理单元都有自己私有的CPU/内存/硬盘等，不共享任何资源；

优点：大规模横向扩展，并行处理能力强，高可用，处理大量数据；

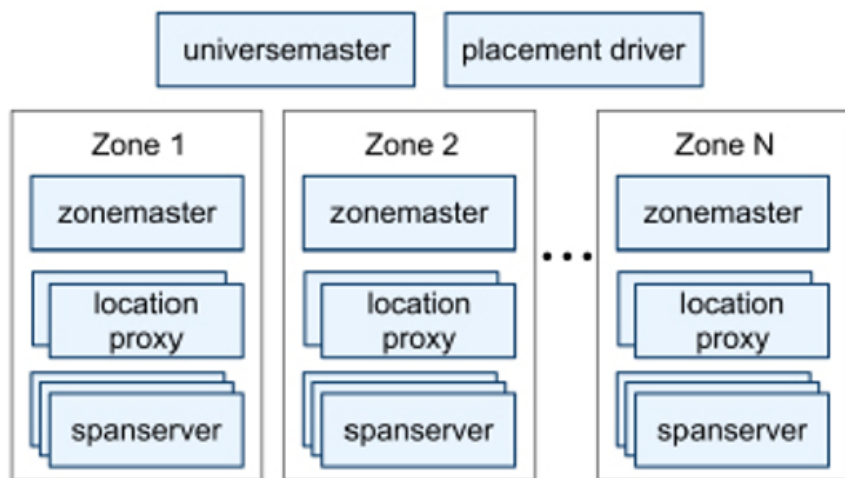
缺点：管理运维相对较复杂，对网络带宽要求高；

场景：适用于高并发的OLTP业务，大数据的OLAP业务；

典型代表：DataEngine MPP，GreenPlum；

一、分布式数据库

什么是 Google Spanner



一个universe是一个数据库

Zone是部署单元

一个Zone在一个DC

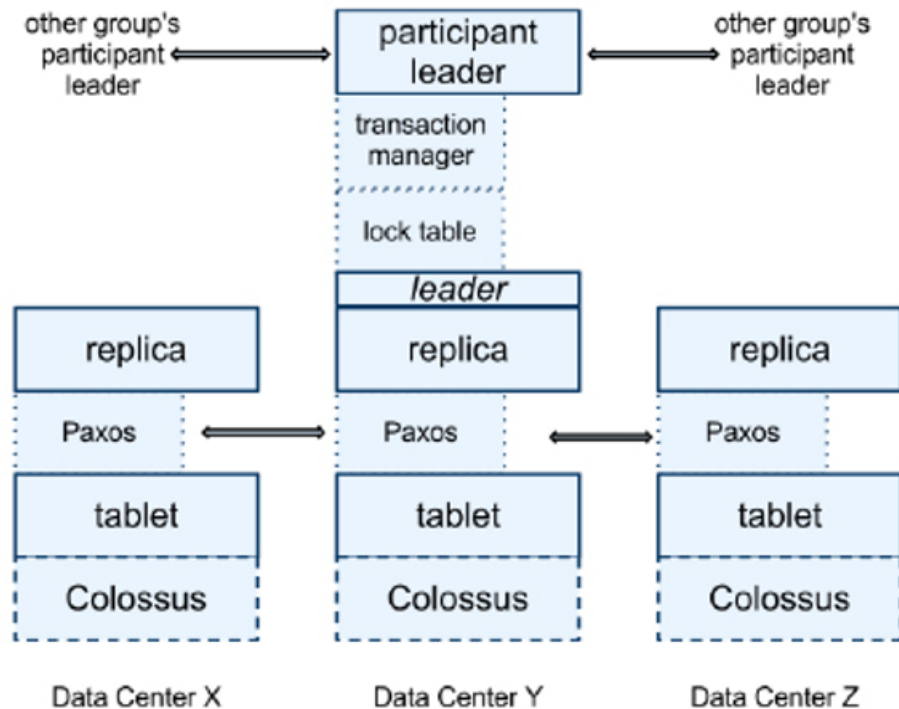
一个Zone包括100-几千span

client连proxy找到span

pd定期处理数据转移和reloadbalance

Figure 1: Spanner server organization.

一、分布式数据库



paxos group

事务管理器在group里
多副本

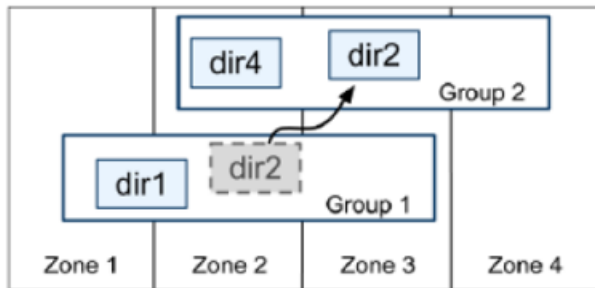
tablet--> k-v

Colossus继承自Google File System。

Figure 2: Spanserver software stack.

一、分布式数据库

什么是 Google Spanner

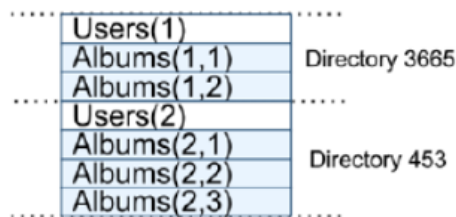


一个paxos组包含多个目录
目录是最小的key集合
目录太大就会分裂
一个后台任务来异步处理
会根据用户访问的位置调整

一、分布式数据库

```
CREATE TABLE Users {
  uid INT64 NOT NULL, email STRING
} PRIMARY KEY (uid), DIRECTORY;

CREATE TABLE Albums {
  uid INT64 NOT NULL, aid INT64 NOT NULL,
  name STRING
} PRIMARY KEY (uid, aid),
  INTERLEAVE IN PARENT Users ON DELETE CASCADE;
```



行数据变了k-v

Method	Returns
<i>TT.now()</i>	<i>TTinterval</i> : [<i>earliest</i> , <i>latest</i>]
<i>TT.after(t)</i>	true if <i>t</i> has definitely passed
<i>TT.before(t)</i>	true if <i>t</i> has definitely not arrived

TrueTime类似于HLC

一、分布式数据库

什么是 Google Spanner

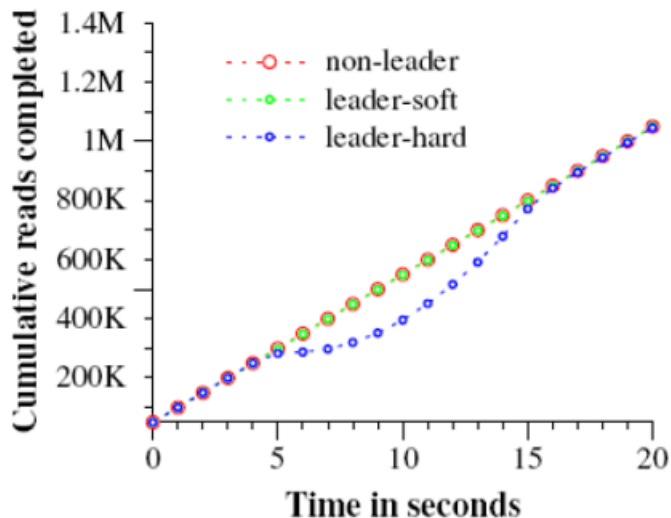


Figure 5: Effect of killing servers on throughput.

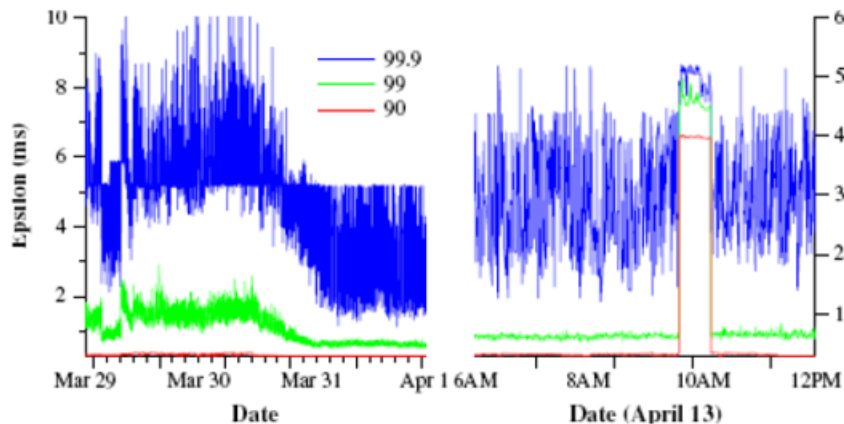
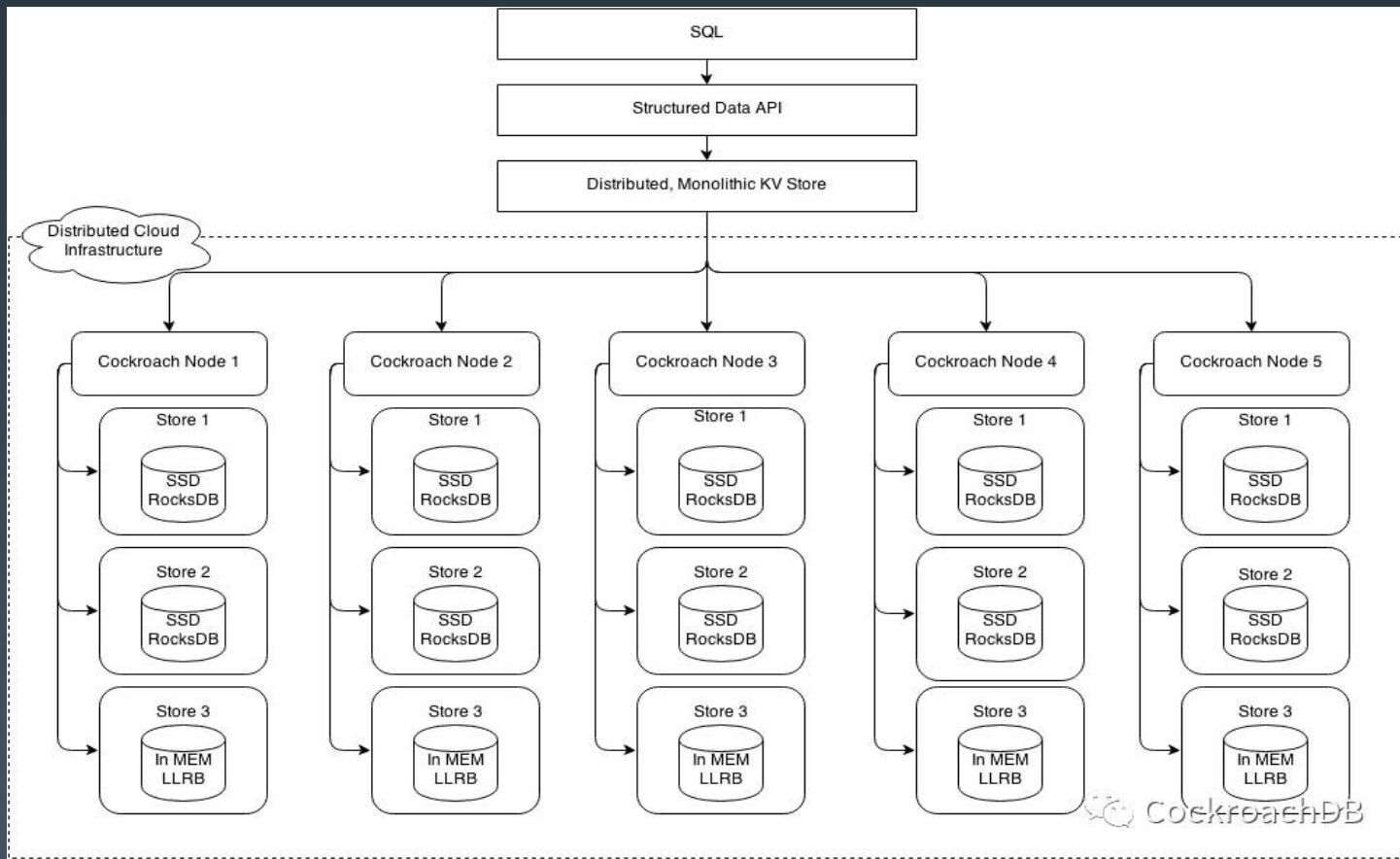


Figure 6: Distribution of TrueTime ϵ values, sampled right after timeslave daemon polls the time masters. 90th, 99th, and 99.9th percentiles are graphed.

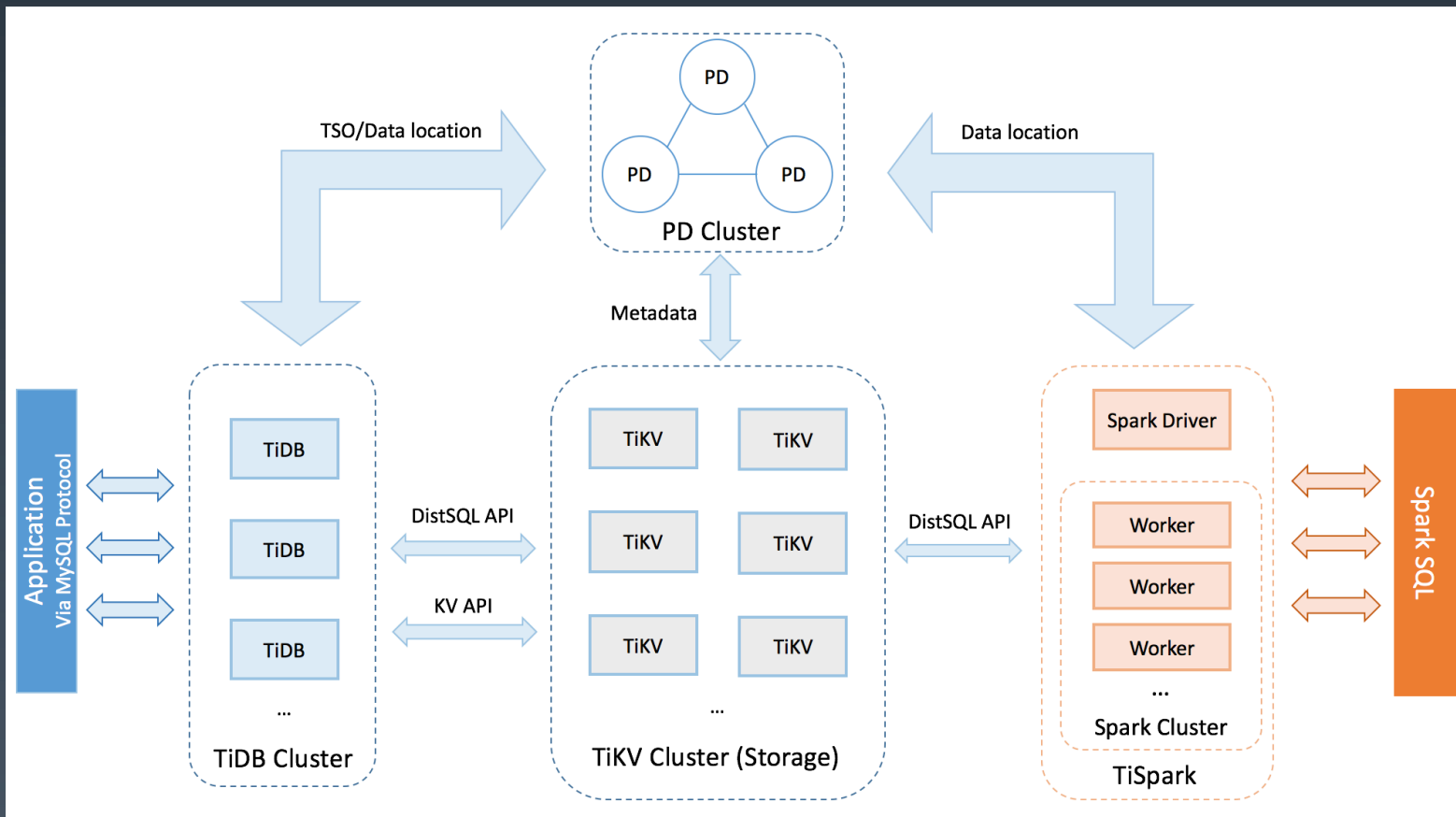
一、分布式数据库

Cockroach



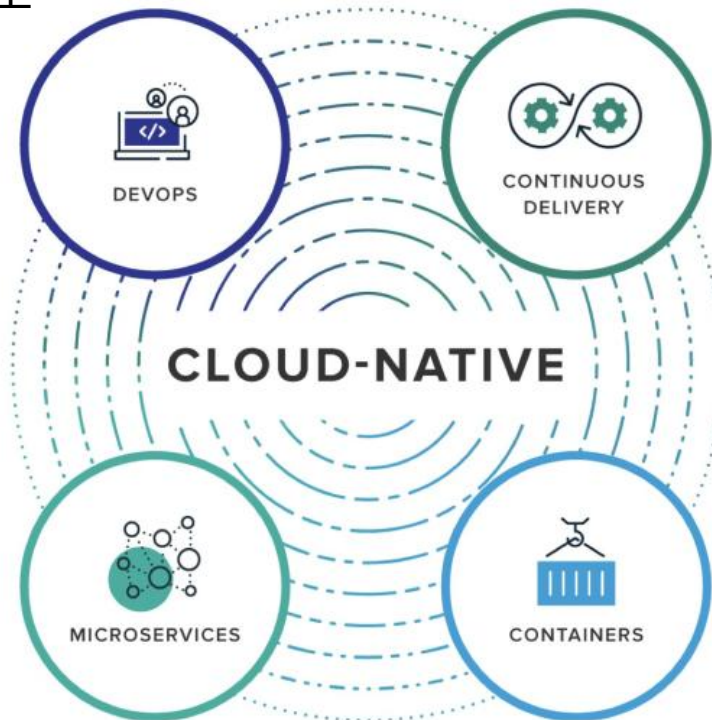
一、分布式数据库

TiDB



二、云原生与数据库网格

云原生



云原生 Cloud-Native

1. 微服务
2. 容器
3. 持续交付
4. DevOps

二、云原生与数据库网格

App Definition and Development

Database

Streaming & Messaging

Application Definition & Image Build

Continuous Integration & Delivery

Orchestration & Management

Scheduling & Orchestration

Coordination & Service Discovery

Remote Procedure Call

Service Proxy

API Gateway

Service Mesh

Runtime

Cloud Native Storage

Container Runtime

Cloud Native Network

Provisioning

Automation & Configuration

Container Registry

Security & Compliance

Key Management

Certified Kubernetes - Distribution

Certified Kubernetes - Hosted

Certified Kubernetes - Installer

PaaS/Container Service

Observability and Analysis

Monitoring

Logging

Tracing

Chaos Engineering

Serverless

Kubernetes Certified Service Provider

Kubernetes Training Partner

Members

CLOUD NATIVE Landscape

Cloud Native Computing Foundation

Red Hat

IBM

Google

Microsoft

Oracle

Alibaba

Ant Group

JD

Meituan

Netease

Qihoo 360

Sina

Tencent

Wang

Xiaomi

Yi

Zhang

Zhihu

Zoo

Others

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

l.cncf.io

Special

Cloud Native Landscape

Cloud Native Computing Foundation

Red Hat

IBM

Google

Microsoft

Oracle

Alibaba

Ant Group

JD

Meituan

Netease

Qihoo 360

Sina

Tencent

Wang

Xiaomi

Yi

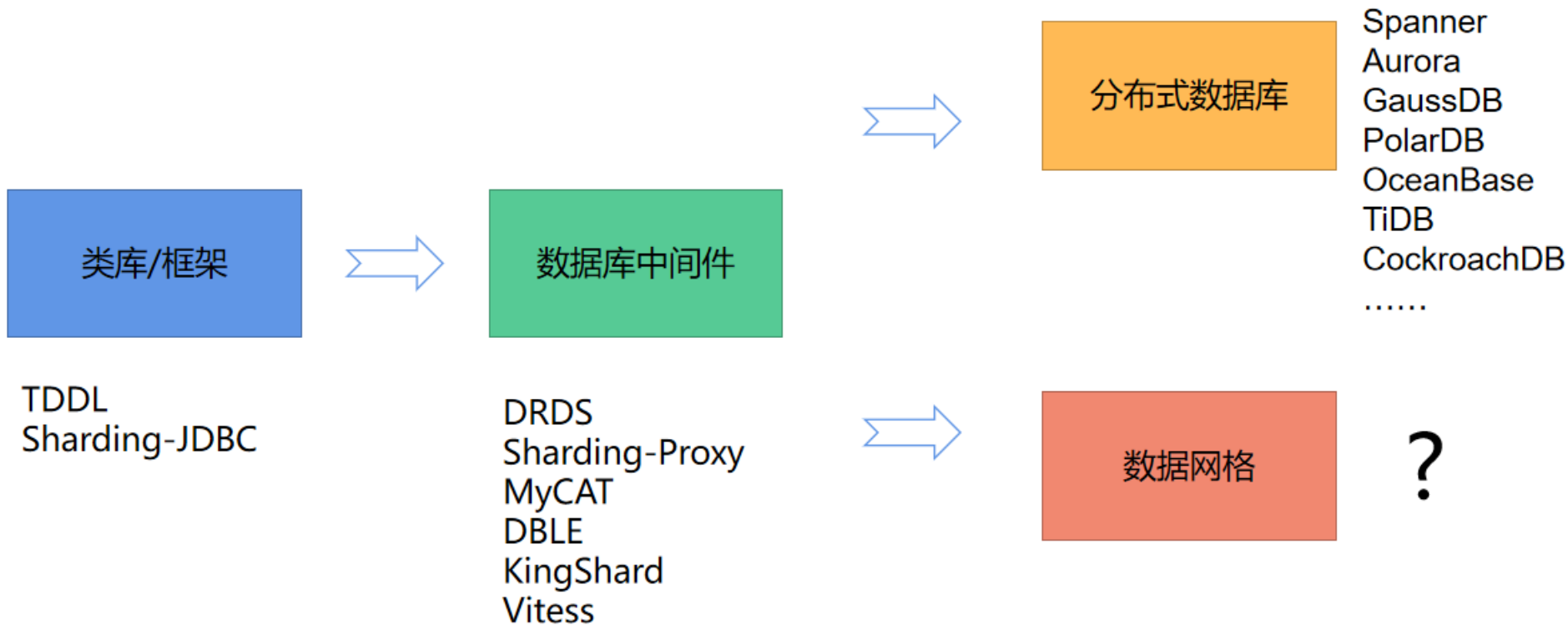
Zhang

Zhihu

Zoo

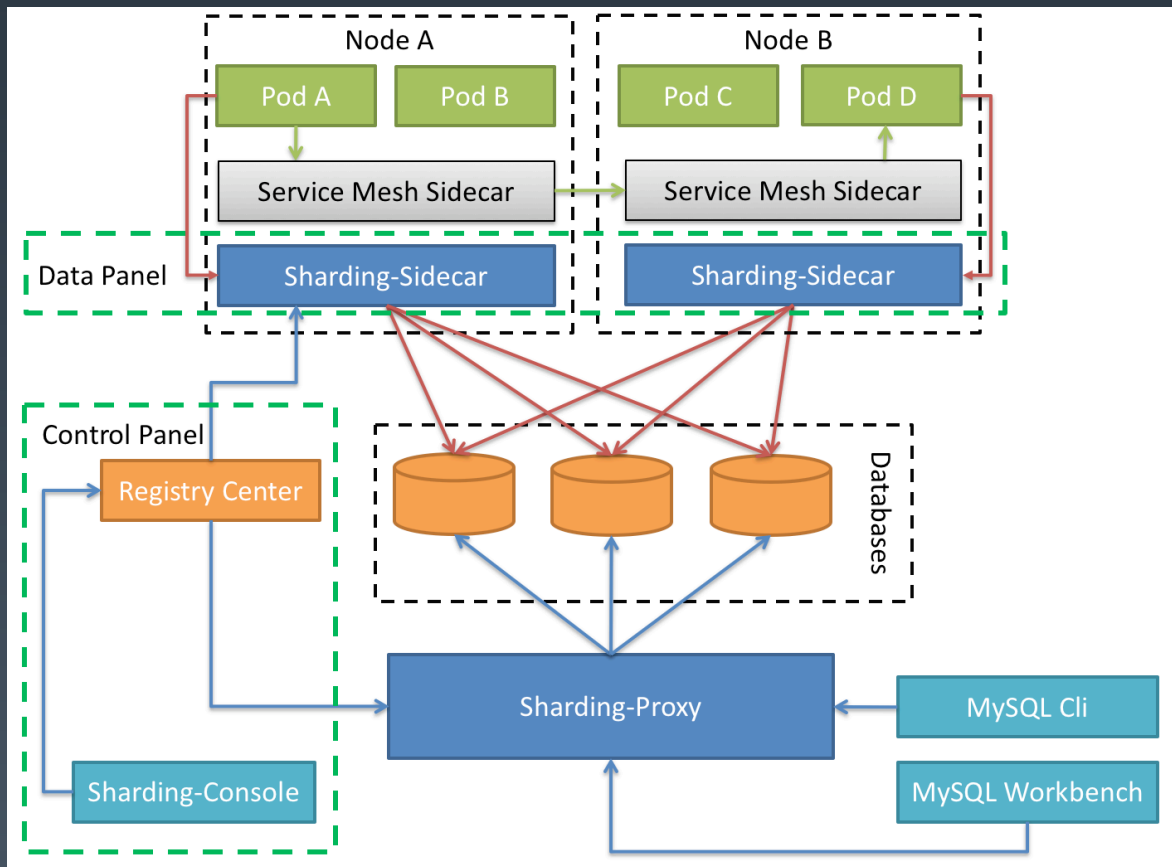
Others

二、云原生与数据库网格



二、云原生与数据库网格

数据库网格



二、云原生与数据库网格

数据库网格

Level 6: Sharding-Engine (6.x+)

Level 5: Sharding-Sidecar (5.x+)

Level 4: Sharding-Scaling (4.x+)

Level 3: Sharding-Proxy中间件 (3.x+)

Level 2: Sharding-JDBC框架 (1.x+)

Level 1: MySQL数据库提供的能力

We're
Here Now

三、下一代数据库技术

公有云+私有云+混合云

云原生+数据网格+分布式数据库+专用型数据库

微服务+单元化 ==> 彻底的中台

完全的 HTAP

THANKS! |  极客大学