Airbnb的基础数据构架

张振

airbnb

NJSD
中国（南京）软件开发者大会
2016

提纲

- 简介
- Airbnb的数据构架
- 深入介绍
  - 集群同步工具：Reair
  - 分布式系统管理框架：Helix

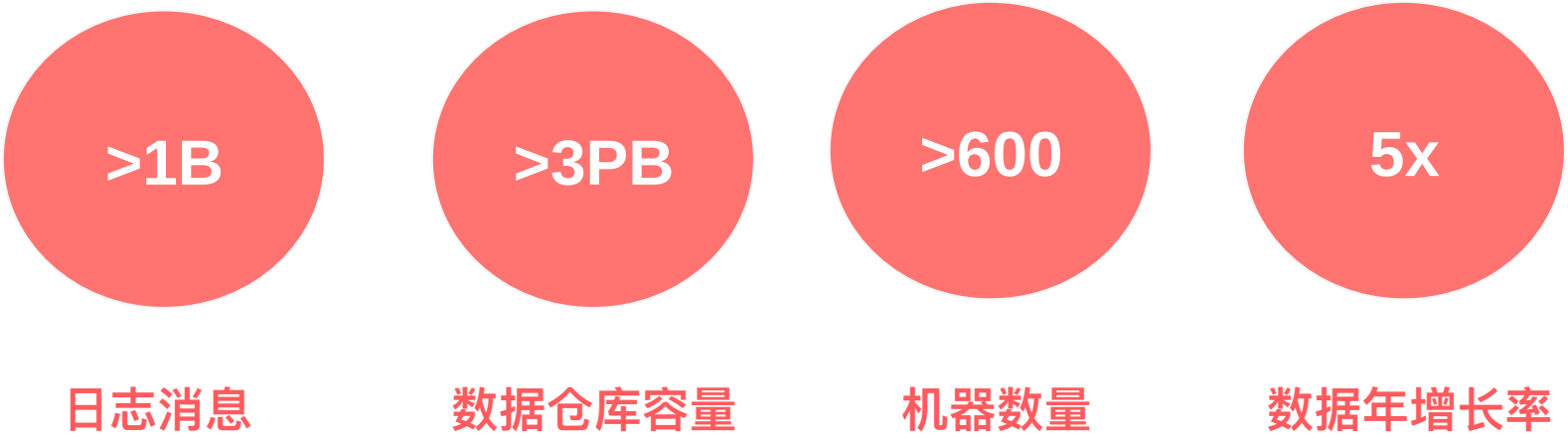
提倡数据的文化

- 数据是决策的重要依据
- 跟踪指标
- 检验试验假设
- 构建机器学习模型
- 深度挖掘商业洞察

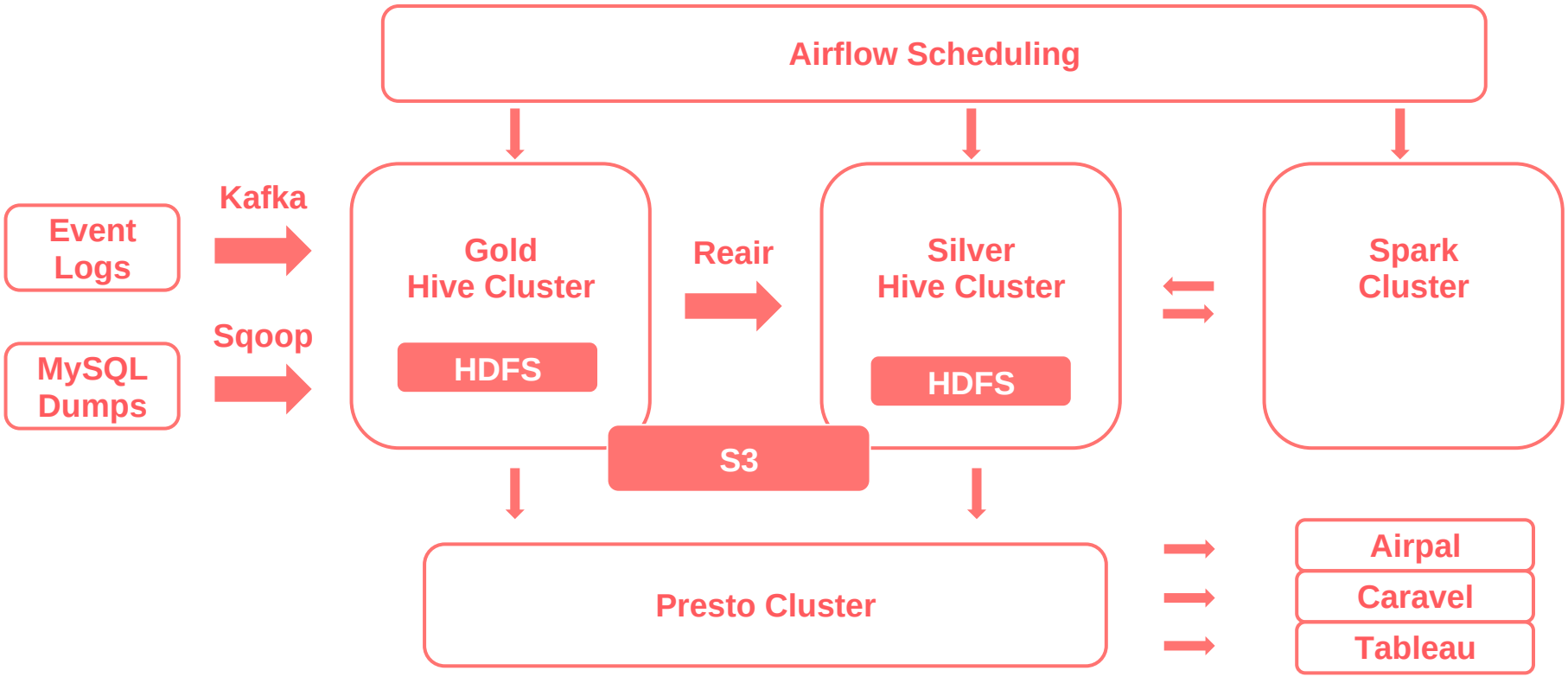
数据基础构架的基本原则

- 开源软件的使用
- 首选标准组件和方法
- 确保可扩展性
- 解决数据用户的实际问题
- 留有余量

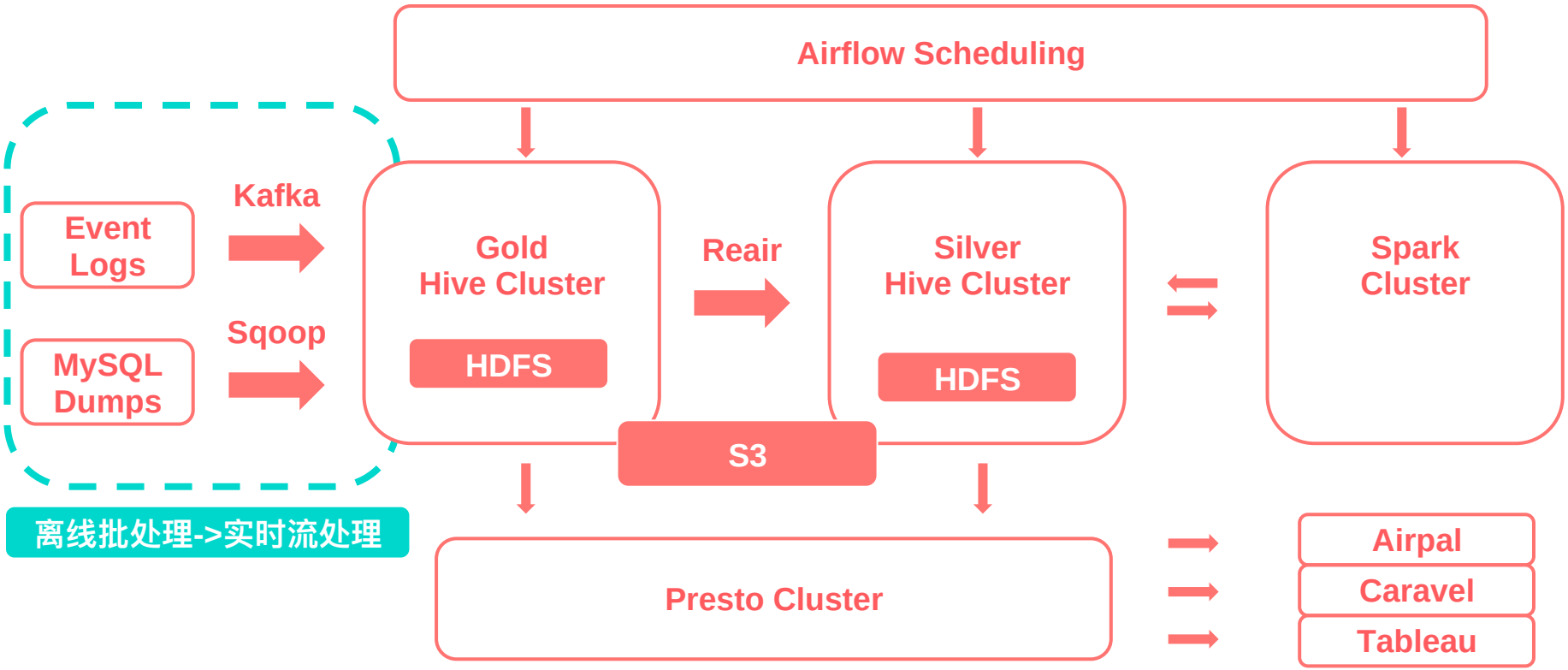
Airbnb数据基础构架的规模



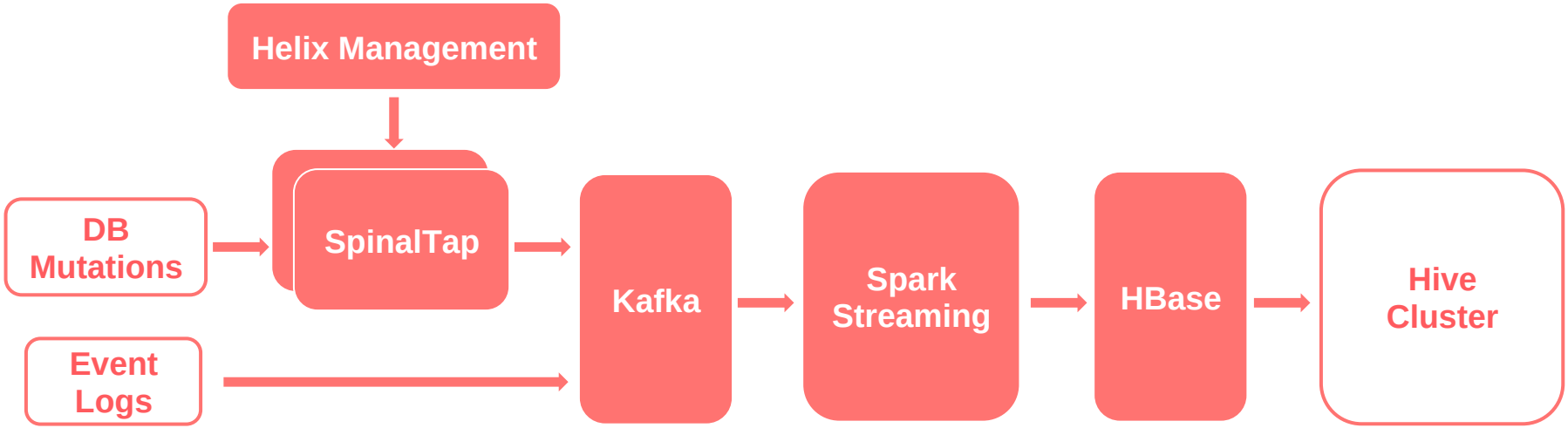
数据构架


Airflow Scheduling
Kafka
Event Logs
Sqoop
MySQL Dumps
Gold Hive Cluster
HDFS
Reair
Silver Hive Cluster
HDFS
Spark Cluster
S3
Presto Cluster
Airpal
Caravel
Tableau

当前的工作

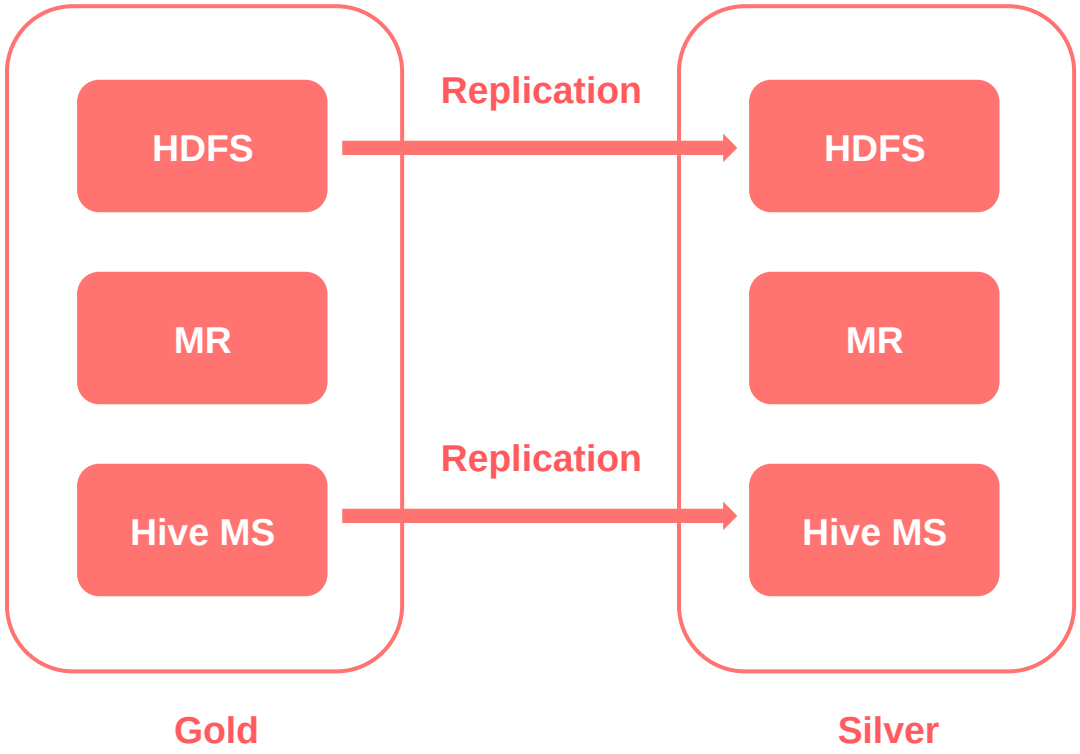

Airflow Scheduling
Event Logs
Kafka
MySQL Dumps
Sqoop
Gold Hive Cluster
HDFS
Reair
Silver Hive Cluster
HDFS
Spark Cluster
S3
离线批处理->实时流处理
Presto Cluster
Airpal
Caravel
Tableau

基于流的实时数据采集


Helix Management
DB Mutations
SpinalTap
Event Logs
Kafka
Spark Streaming
HBase
Hive Cluster

深入介绍

Reair


HDFS
MR
Hive MS
Replication
Replication
HDFS
MR
Hive MS
Gold
Silver

两个独立的集群

优势

- 用户作业的错误隔离
- 方便容量规划
- 保证SLA
- 易于测试新版本
- 灾难恢复

不利

- 用户容易混淆
- 数据同步
- 运营成本

数据仓库同步策略

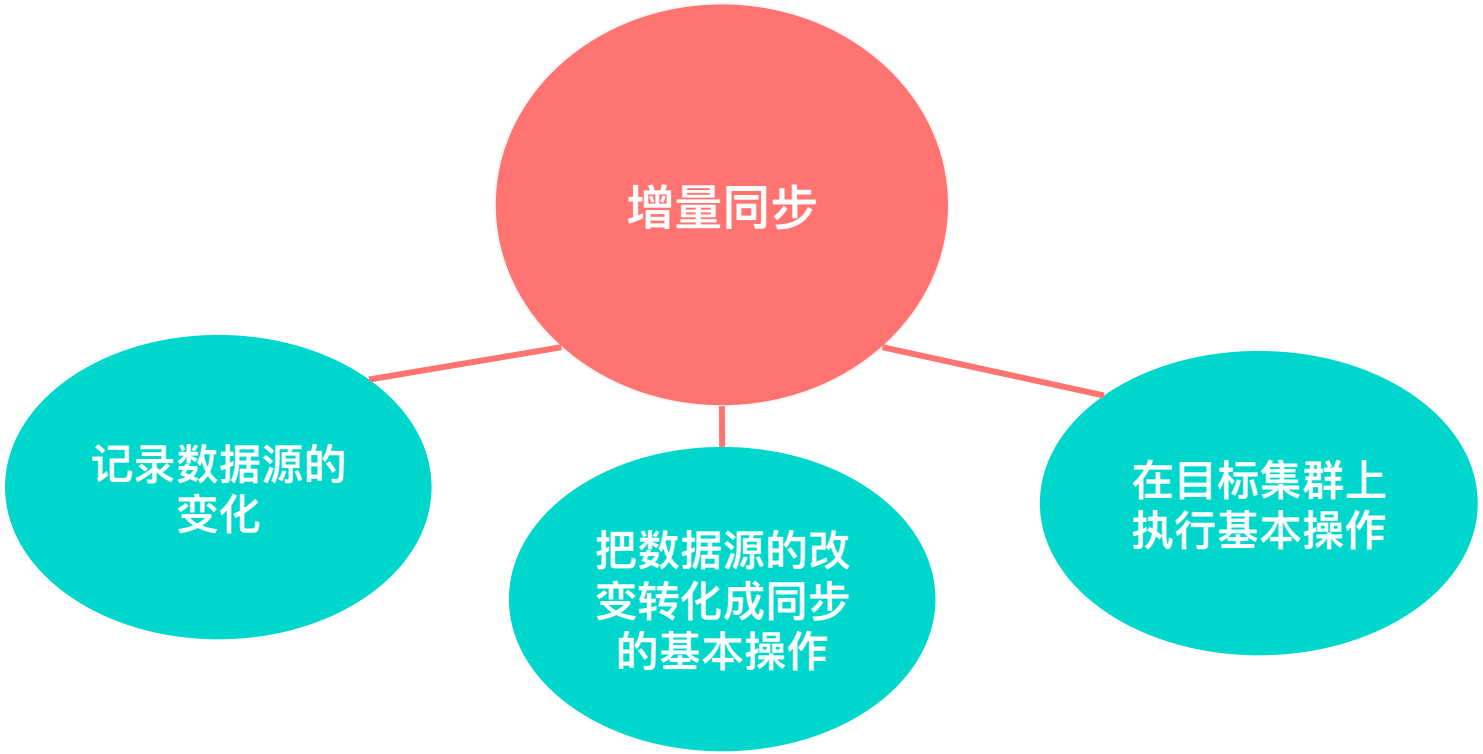
批量同步

- 扫描HDFS, Metastore
- 拷贝相关数据
- 简单，不需要维护状态
- 高延时

增量同步

- 记录数据源的变化
- 拷贝到目标集群，或者在目标集群重新执行操作
- 复杂，需要维护很多状态
- 低延时 (秒级)

Reair 概况



记录数据源的变化

- Hive 提供 hooks API，在特定点回调用户定义的操作
  - 执行前（Pre-execute）
  - 执行后（Post-execute）
  - 错误（Failure）
- 使用 post-execute，在审计日志（audit log）中记录被创建的对象
- 在执行查询的关键调用路径上

将数据源变化转变成基本操作

Operation/Object	Database	Table	Partition
Copy	CopyDatabase	CopyTable	CopyPartition
Drop	DropDatabase	DropTable	DropPartition
Rename	N/A	RenameTable	RenamePartition

将数据源变化转变成基本操作

CREATE TABLE srcpart (key STRING) PARTITIONED BY (ds STRING)

Copy table srcpart

INSERT OVERWRITE TABLE srcpart PARTITION(ds=‘1’) SELECT key FROM src

Copy partition srcpart/ds=1

ALTER TABLE srcpart SET FILEFORMAT TEXTFILE

Copy table srcpart

ALTER TABLE srcpart RENAME to srcpart_old

Rename table srcpart to srcpart_old

同步基本操作流程的一个例子

拷贝一个表

1. 检查数据源是否存在
2. 检查目标数据是否存在，并且是否相同
3. 如果不是，用distcp将源数据拷贝到一个临时地点
4. 验证拷贝的正确性
5. 如果拷贝成功，将拷贝的内容移动到最终地点
6. 加载相关的metadata
7. 可重复执行性（idempotent）并且保证数据的完整性

在目的端运行基本操作

- 串行执行
  - 容易推导出操作的流程
  - 执行较慢
- 并行执行
  - 快速且可扩展
  - 操作顺序至关重要
    - E.g. create table before copying a partition
  - 基本操作的DAG

错误处理

- 能够在任意点重新执行
- 所有的基本操作都是可以重复执行的（idempotent）
- 保存Checkpoint

Helix

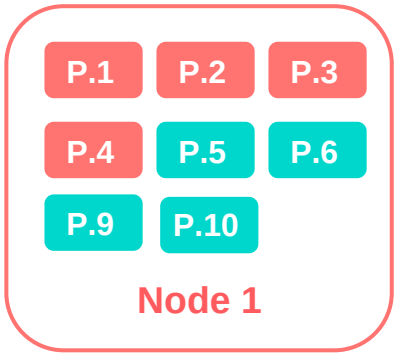
- Helix 是一个分布式资源管理框架：a cluster management framework for partitioned and replicated distributed resources
- Apache top level 项目，由Linkedin开源
- Helix 提供:
  - 自动将逻辑资源和partition映射到物理节点
  - 节点错误检测和恢复
  - 动态增加逻辑资源
  - 自动负载均衡
  - Multitenancy

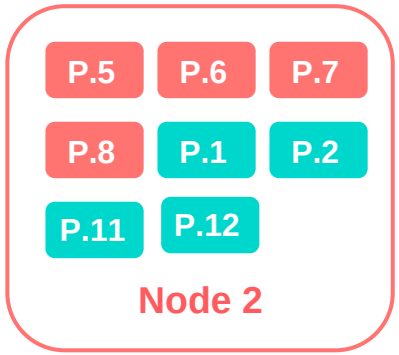
一个例子

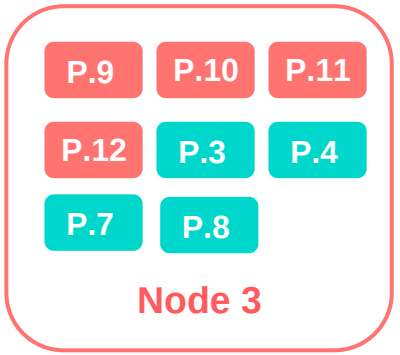
分布式数据库

主备份

从备份







Partition 管理

多个备份 Replicas
一个主备份
Even Distribution

容错

错误检测
状态改变
Even Distribution
No SPOF

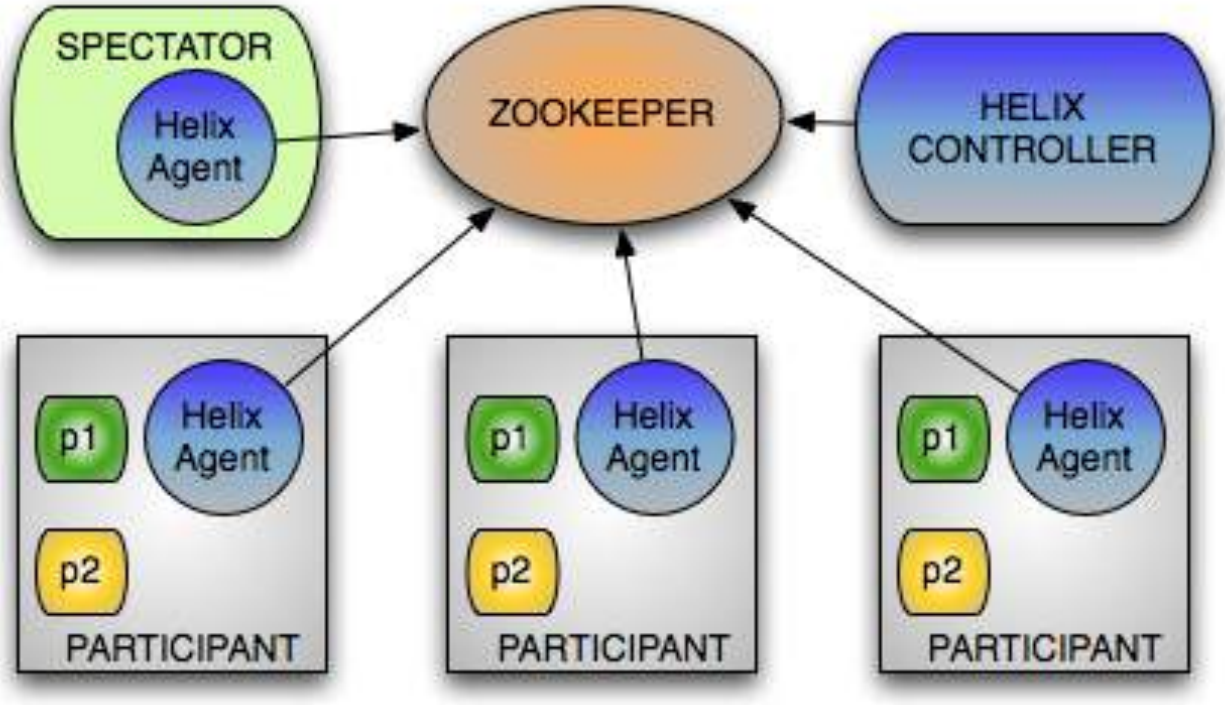
可伸缩性 Elasticity

Min Downtime
Min Data Movement
Throttle Data Movement

Helix的基本概念

- Resource: 逻辑上partitionable实体，例如一个数据库
- Partition: Resource的一部分
- 备份: 一个partition的拷贝，以及它的期望状态
- 状态转换和状态机
- 参与者（Participant）: A node that actually hosts a distributed resource
- 观察者（Spectator）: A node that observes the state of Participants (e.g. a router)
- 控制者（Controller）: 协调所有参与者的状态转换

Helix的构架


SPECTATOR
Helix Agent
ZOOKEEPER
HELIX CONTROLLER
p1
Helix Agent
p2
PARTICIPANT
p1
Helix Agent
p2
PARTICIPANT
p1
Helix Agent
p2
PARTICIPANT

问题？