hosted by Alibaba Group 阿里巴巴集团 APACHE HBASE

The Application of HBase in New Energy Vehicle Monitoring System

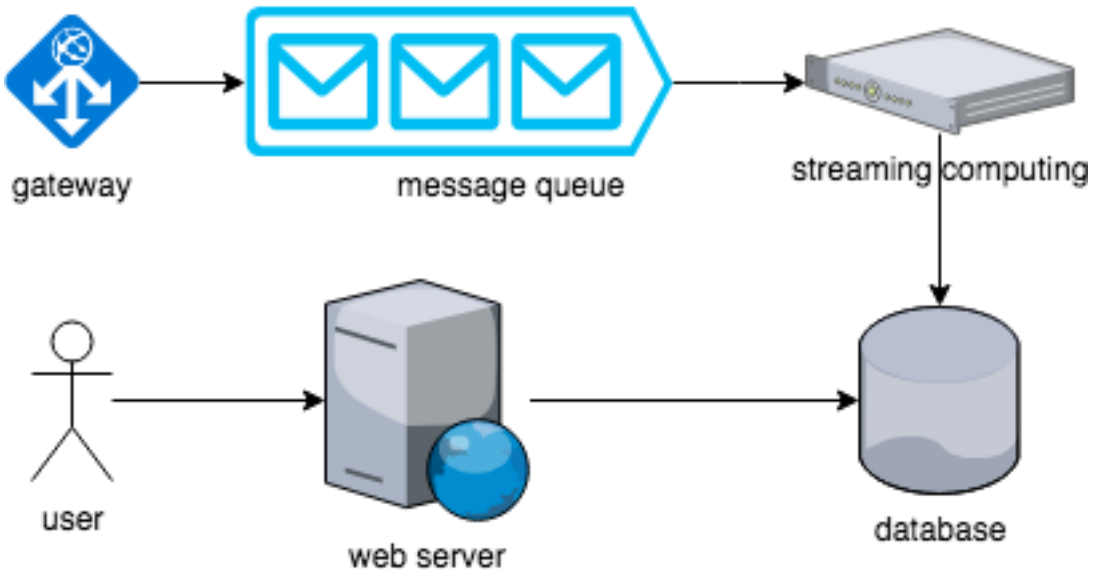
颜禹 Yan Yu

Burnish
博尼施科技 Burnish Technology Co. Ltd

content

Backgroud

- 100k running vehicles online
- send 2 packages per minute every vehicle.
- data space
  - the origin package size is 1KB.
  - parsed package size is about 7KB.
  - one vehicle will produce 20mb data per day.
  - 2TB data were generated per day.
  - 2.9 billion rows need to write to HBase every day.
- concurrency
  - 3.3k persistent tps
  - 100k persistent connections
  - 3.3MB origin data needs to parse per second
  - 23.1MB parsed data needs to storage in HBase per second


gateway
message queue
streaming computing
user
web server
database

Challenges

- Small team
- Limited funds, machines,
- Short deadline
- System integration
- How to handle the huge amount of vehicle data
- Demands are very foggy.

Decisions

- Language
- Message queue
- Database
- Develop flow
  - Micro service
  - Monolithic service
- Deploy and maintain
  - Cloud
  - Native data center

Language

- C/C++
  - High performance
  - Hard to integrate
  - Long development time
- Java
  - High performance
  - Rich third part packages
  - Easy to integrate with big data system, i.e. Hadoop, HBase, spark
- Python
  - Sprint development
  - Rich third part packages
  - Performance issue with multi thread
- Golang
  - Easy to write multi thread program
  - There is no Golang developer in our team

Message

- Redis
  - High performance
  - High memory requirement
  - Hard to scale
- Celery
  - More fit for distribution task
  - Easy to develop with python
  - Redis or rabbitmq as it’s backend
- Kafka
  - Write to disk first to ensure the message security
  - Support consumer group
  - Auto balance
  - Enough performance for our system
  - Easy to scale
- Rabbitmq
  - Classic message queue
  - Performance

Database

- MySql
  - Relational database
  - Fit for storage static information
  - ORM support
- MongoDB
  - Document based
  - ORM support
  - Hard to maintain and scale
- Hbase
  - Column database
  - High write performance
  - Easy to handle TB
  - Easy to scale
- OpenTSDB
  - Time series database
  - Based on HBase

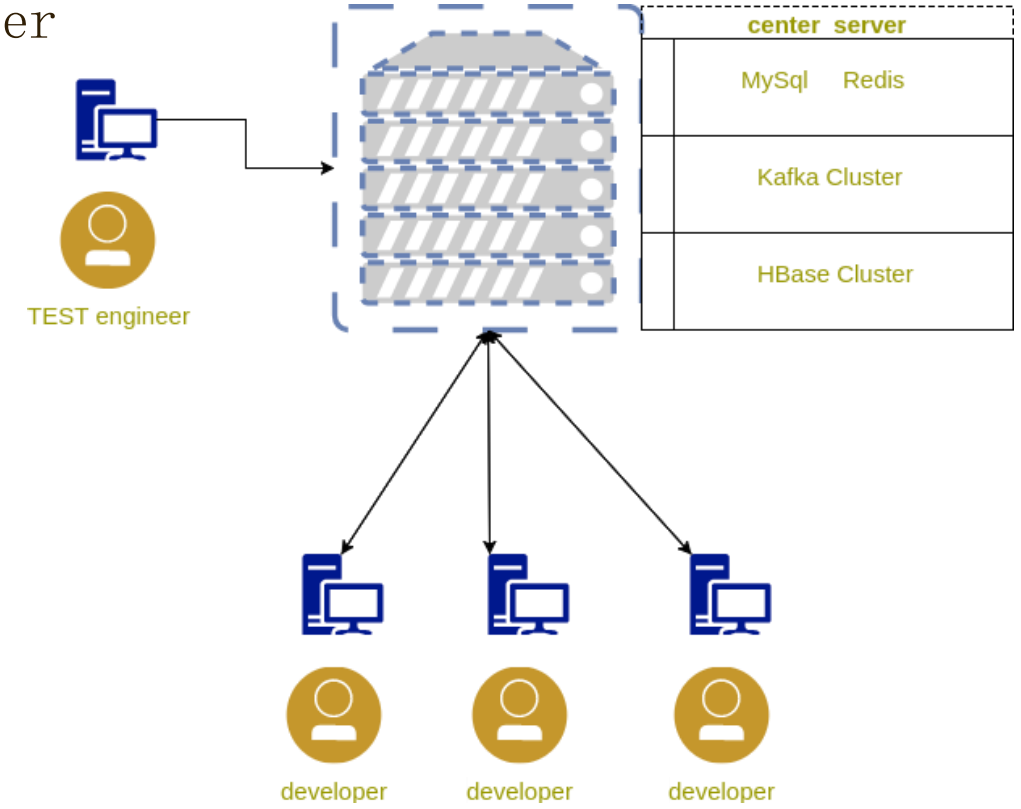
Monolithic service vs micro service

- Monolithic service
  - Easy to develop when system is not very complicate
  - Acceleration for development
  - Build the basic system due the the foggy demands
- Micro service
  - Easy to scale in a complicate system
  - Rapid iteration
  - More developers requirement

Develop flow

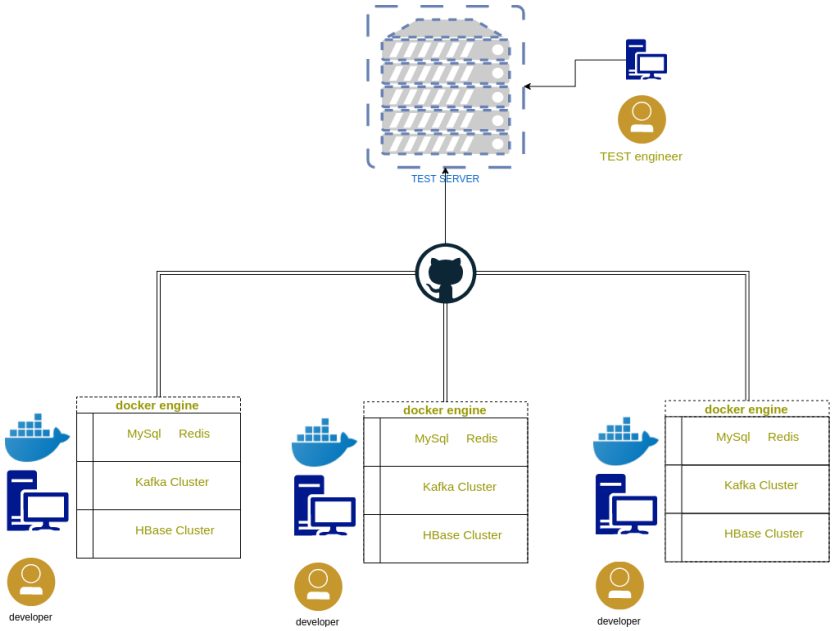
Dependences on central server.

- Dependences on central server.
  - Easy to setup on one server
  - Single point failure risk
  - Confliction over multi developers



Develop flow

- Dependences on individual docker engine.
  - Easy to setup with docker-compose
  - No single point risk
  - High memory develop machine requirement(starting from 32GB)



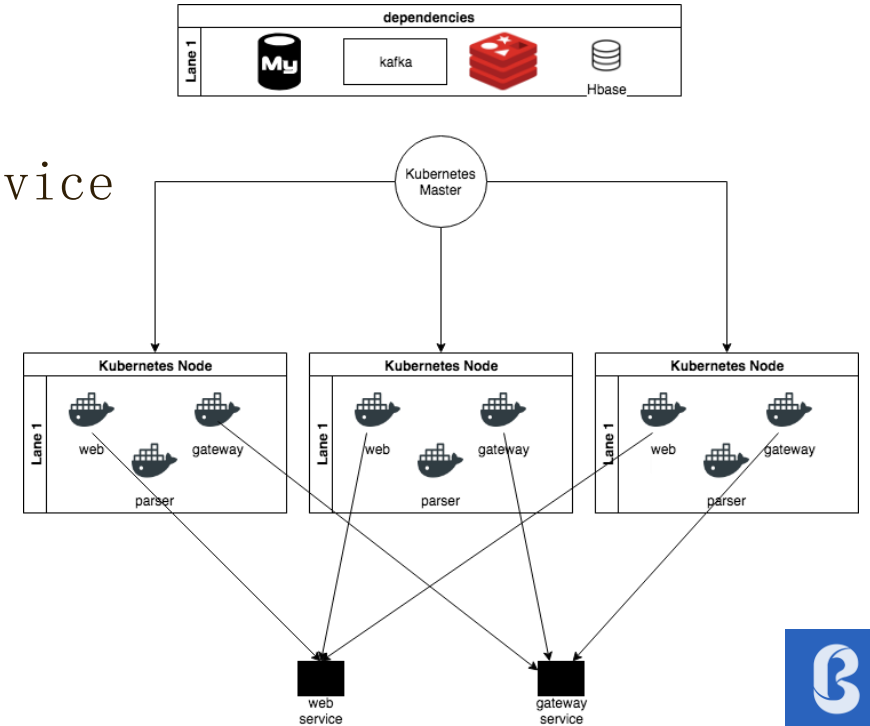
Deploy and maintain

- Cloud
  - Easy setup
  - Low cost with small scale
  - Fast deployment
  - No employees
- Native data center
  - Hard setup
  - Expensive cost with small scale
  - Professional employee to maintain our data center

Deploy and maintain

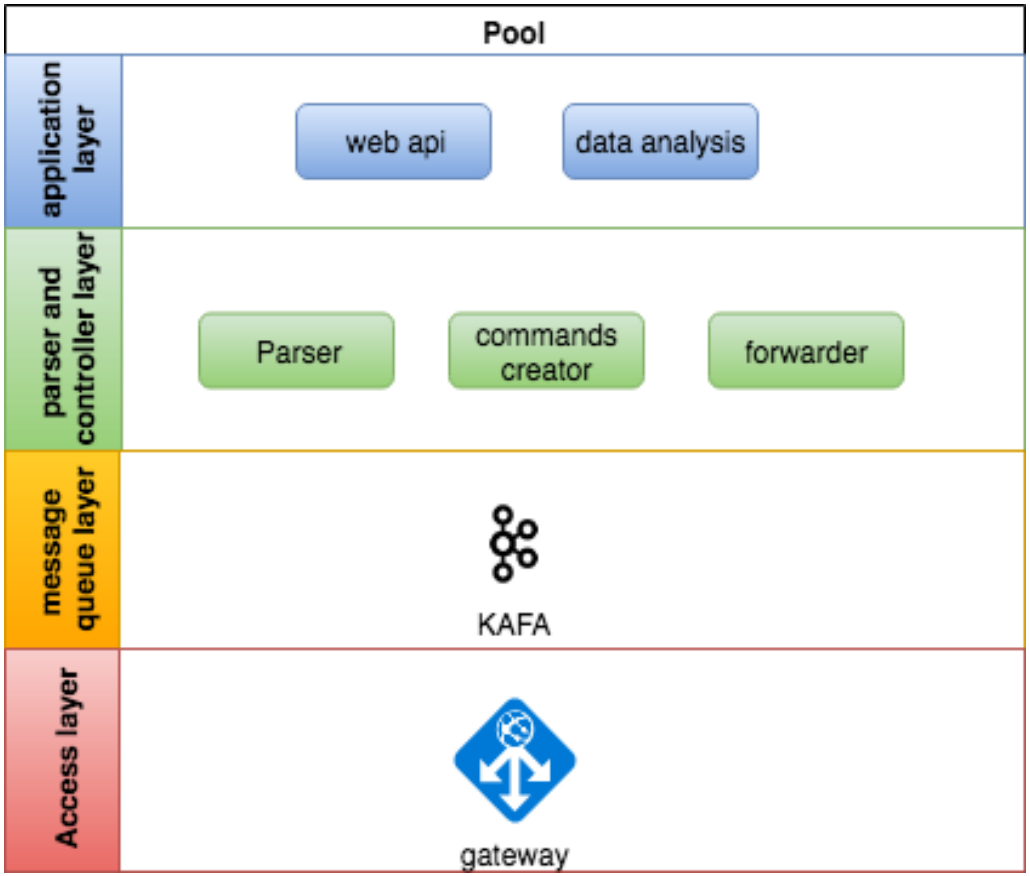
- Deploy system with kubernetes
  - Easy to management
  - Rapid scale
  - Compute and storage split separation

- Deploy basic component with cloud service
  - Fast deploy
  - Careless
  - Easy to get high available service
  - No employees

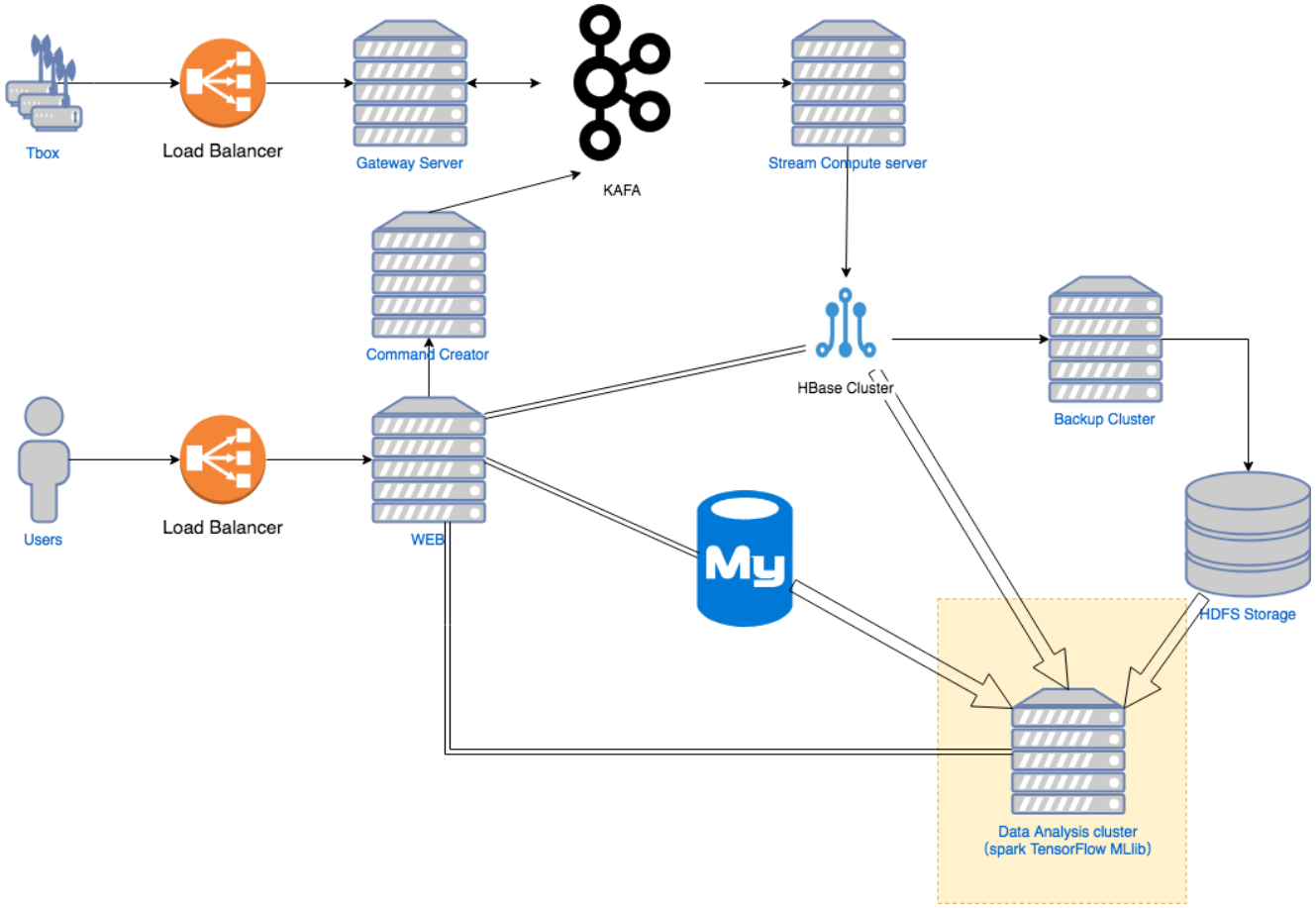


- Individual develop environment with docker and docker-compose.
- Deploy system with kubernetes to reduce the operation cost.
- Develop with pure python code.
- Just build the basic system, another demands delay to second phase development.

The System Architecture



The System Architecture



System maintain

- Application scale
  - Application scale with kubernetes
  - Basic component with cloud service
- CI/CD
  - CI with jenkins
  - CD with jenkins and kubernetes
- Data Backup
  - Mysql backup
  - Hbase backup
  - Message backup

Why HBase?

- High write performance
- Quick response for query
- Easy to scale
- SQL support with phoenix
- Aliyun provide HBase SAAS

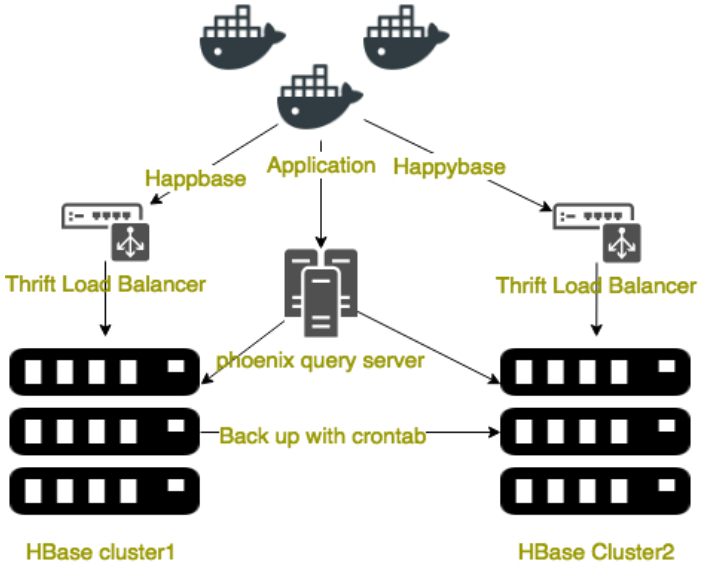
Connect to HBase cluster with python

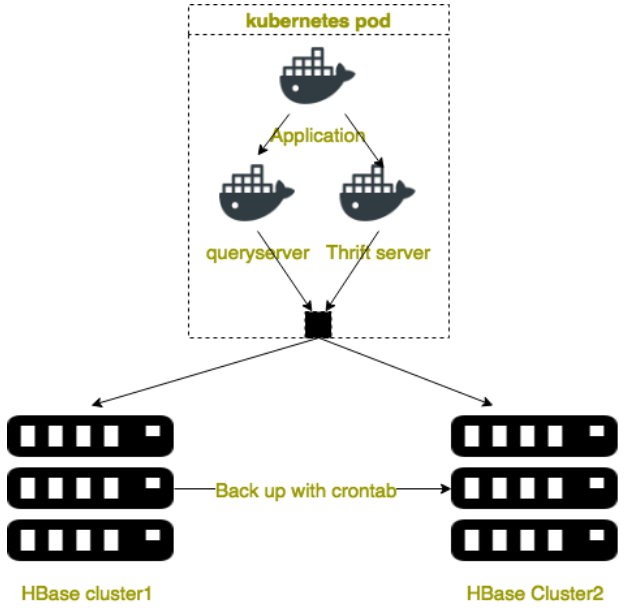
- Provide native java API
- Connect HBase with thrift
- Happybase provide pythonic API
- SQL support with phoenix

challenges with HBase

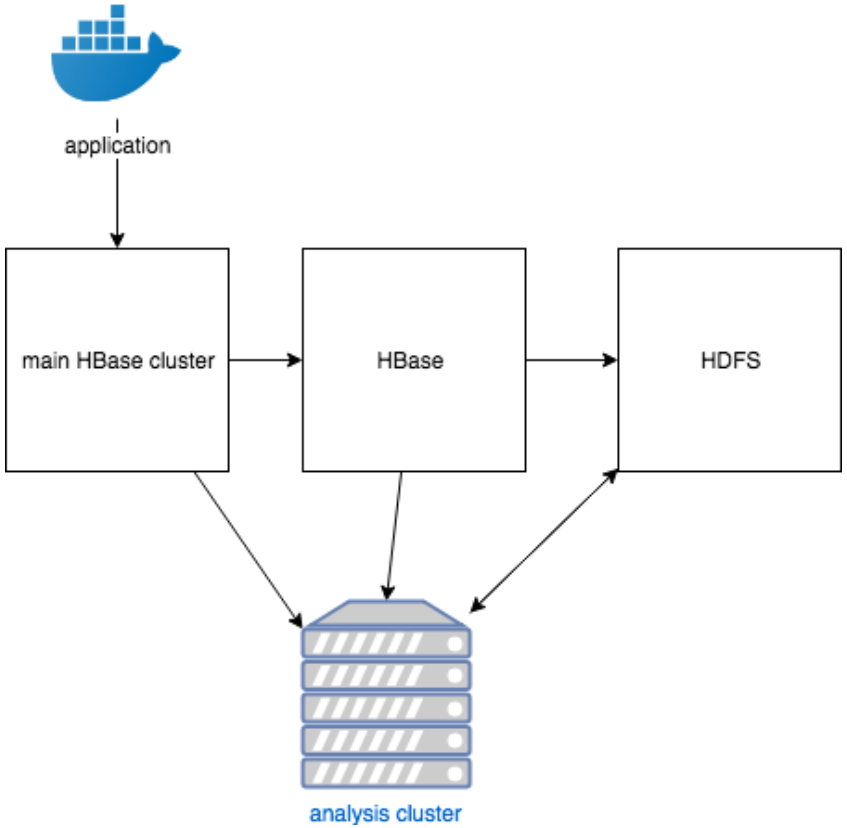
- Row key design
  - Hash prefix + timestamp
- Second index
  - Import phoenix support
  - Insert index manually
- Table design
  - Short column name
  - Carefully design the table with demands (i.e. the mileage of every single vehicle)
- Complex query is very slow.
  - Create index
  - Export some results to HDFS or MySql (kylin?)

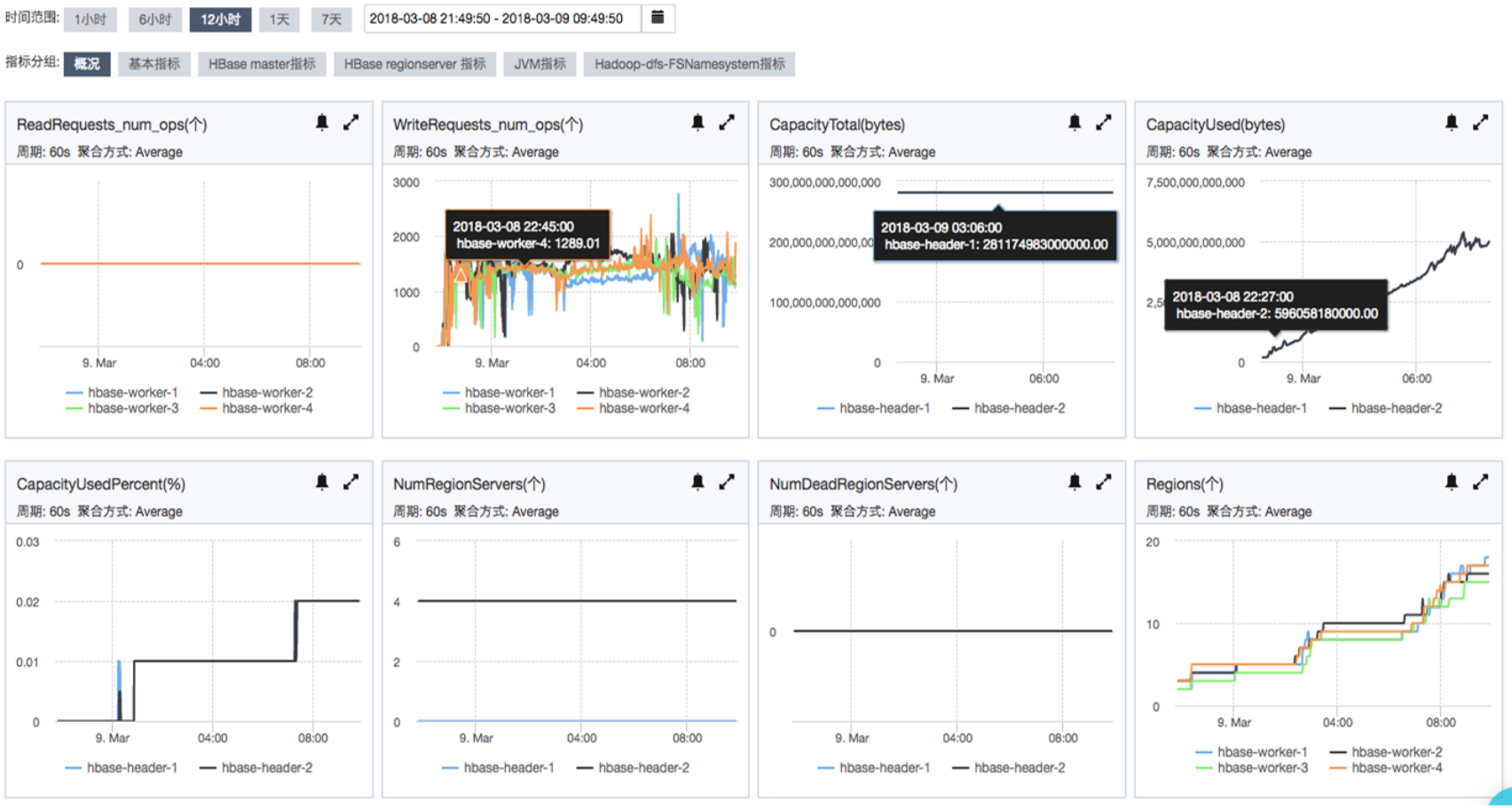
Hbase Cluster





Data Backup Approach




时间范围: 1小时 6小时 12小时 1天 7天 2018-03-08 21:49:50 - 2018-03-09 09:49:50
指标分组: 概况 基本指标 HBase master指标 HBase regionserver 指标 JVM指标 Hadoop-dfs-FSNamesystem指标
ReadRequests_num_ops(个)
周期: 60s 聚合方式: Average
0
9. Mar
04:00
08:00
hbase-worker-1
hbase-worker-2
hbase-worker-3
hbase-worker-4
WriteRequests_num_ops(个)
周期: 60s 聚合方式: Average
3000
2000
1000
0
2018-03-08 22:45:00
hbase-worker-4: 1289.01
CapacityTotal(bytes)
周期: 60s 聚合方式: Average
300,000,000,000,000
200,000,000,000,00
100,000,000,000,000
2018-03-09 03:06:00
hbase-header-1: 281174983000000.00
06:00
hbase-header-1
hbase-header-2
CapacityUsed(bytes)
周期: 60s 聚合方式: Average
7,500,000,000,000
5,000,000,000,000
2018-03-08 22:27:00
hbase-header-2: 596058180000.00
CapacityUsedPercent(%)
周期: 60s 聚合方式: Average
0.03
0.02
0.01
NumRegionServers(个)
周期: 60s 聚合方式: Average
6
4
2
NumDeadRegionServers(个)
周期: 60s 聚合方式: Average
Regions(个)
周期: 60s 聚合方式: Average
20
10

Pain point

- Complex query with HBase API still very slow
- Phoenix needs create index to the query speed
- Phoenix query still very slow if there is no index in HBase
- Complex query needs big size of index in HBase
- The queryserver between python and phoenixdb is very weak

Conculsion

- Introduced the background of monitoring system
- Our decisions of the system
- Why we choose HBase as our main database
- How we deploy and maintain the system
- Introduced the practice of HBase in the system

Prospects

- Rewrite high performance component with golang.
- Split the monolithic system into micro service when the system becomes complex
- Data analysis
  - Fault diagnosis
  - Predict the vehicle status
- Data compression
- Opentsdb
- Combine the elasticsearch and Hbase in our application.

hosted by Alibaba Group 阿里巴巴集团 APACHE HBASE

Thanks