基于Apache Flink的平台化构建及运维优化经验

公司： 腾讯 Tencent

职位： 高级工程师

演讲者：施晓罡 / 郑灿彬

个人介绍

Self-introduction

施晓罡

robbieshi@tencent.com

Apache Flink Committer，腾讯高级工程师，目前在TEG数据平台部参与实时计算平台的研发。在加入腾讯之前，于2016年在北京大学获得博士学位，研究方向专注于大规模数据的管理和分析，包括查询优化、流数据处理和迭代计算等，曾在SIGMOD, TODS和IPDPS等国际顶级会议和期刊上发表多篇论文。

背景介绍

Background

210 Million

每秒接入消息数峰值

Maximum number of messages received per second

17 Trillion

日接入消息数

Number of messages received per day

3 PB

日接入数据量

Amount of data received per day

20 Trillion

日实时计算量

Number of real-time computations per day

8800

单集群规模

Number of nodes in a single cluster

30000

集群规模

Total number of nodes in clusters

6 Million

日离线任务数

Number of offline tasks per day

470 PB

存储容量

Amount of data stored in clusters

为何选择 Flink?

Why Flink?

提供了Exactly-Once的容错语义

Achieve exactly-once semantics when tolerating failures

有效的状态管理支持

Provide efficient support for state management

丰富的实时计算的语义表达

Provide powerful programming interfaces

出色的执行性能

Exhibit good performance

Flink 作业生命周期

Lifecycle of Flink Jobs

开发

Development

测试

Test

部署

Deployment

运维

Operating

通过平台化提高从应用开发到上线运维的效率

Improve the efficiency as a platform

Oceanus

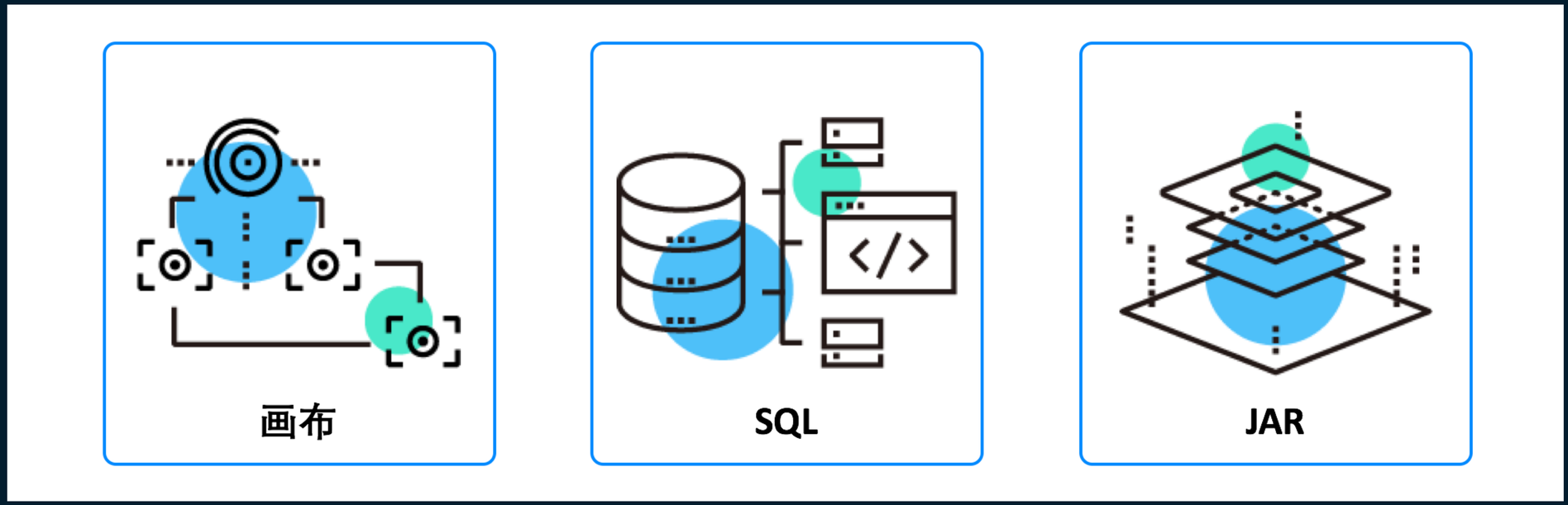
一站式可视化实时流计算平台

面向内部用户

A one-stop platform in house for developing and operating real-time applications

Oceanus 简介

Introduction to Oceanus



提供 Web 端多种应用构建方式

Provide various methods to facilitate the development of real-time applications

Oceanus 简介

Introduction to Oceanus

提供了对外部数据的声明和管理，方便用户数据的读取和写出

Provide data definition for external sources and sinks.

Oceanus 简介

Introduction to Oceanus

支持实时查看应用计算结果

用户可以选择在Web端查看或者接入小马报表*

Allow the retrieval of computing results at real-time.

Users can either retrieve the results at Oceanus with original results or use Xiaoma to obtain the visualization results.

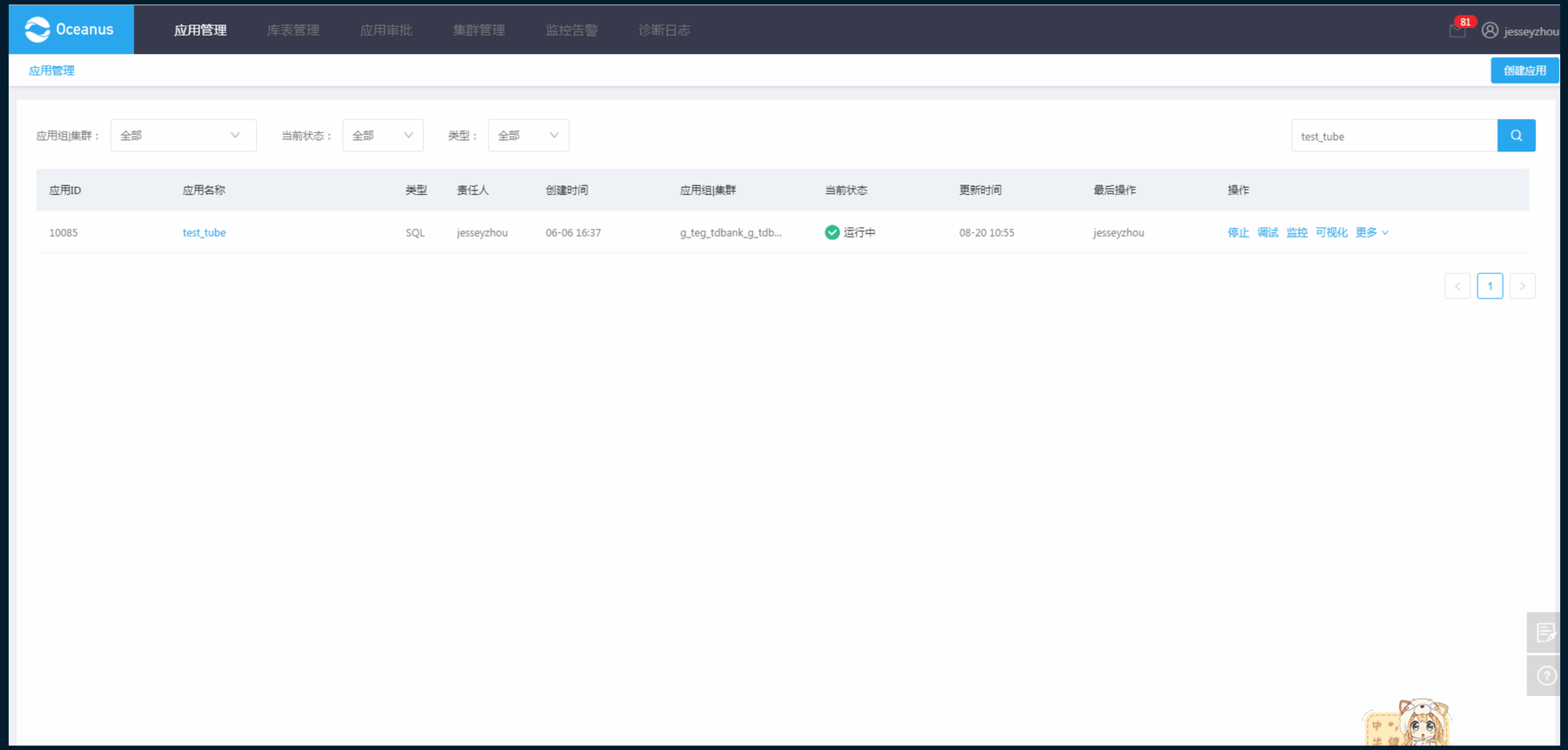
* 小马报表: https://xiaoma.qq.com

Oceanus 简介

Introduction to Oceanus

允许用户快速验证开发应用的正确性

Users can verify their applications by generating test data.



Oceanus 简介

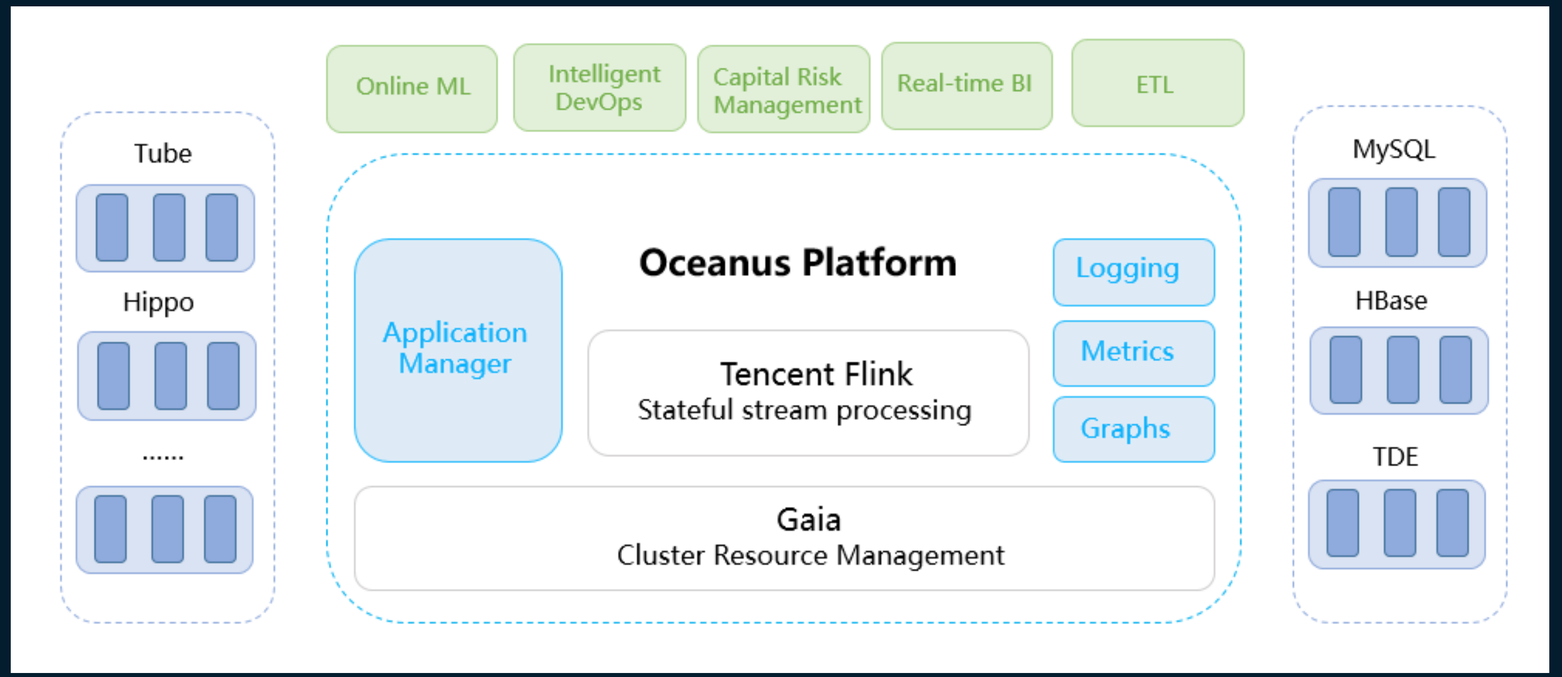
Introduction to Oceanus

使用 Gaia 系统进行集群资源管理和调度

用户可以有效便捷的进行作业部署，并提供了弹性缩容和扩容

Employs Gaia to manage cluster resources.

Users can easily deploy their applications and achieve elastic scalability.



Oceanus 简介

Introduction to Oceanus

提供细粒度的指标曲线，让运维人员可以实现精细化运营

Allow real-time and fine-grained demonstration of metrics

功能改进

Improvements

Oceanus 对 Flink 内核进行了大量的改进来提高其可用性

Oceanus does a lot of work to improve the usability of Flink

- **在 Flink 1.7 中贡献了超过 30 个 Pull Request**

  Contribute >30 pull requests in Flink 1.7

- **提供了比官方版本多 30 个以上的 Table API 和 SQL 函数**

  Provide >30 Table API and SQL functions than the community version

- **提供了对 AsyncIO 算子超时处理**

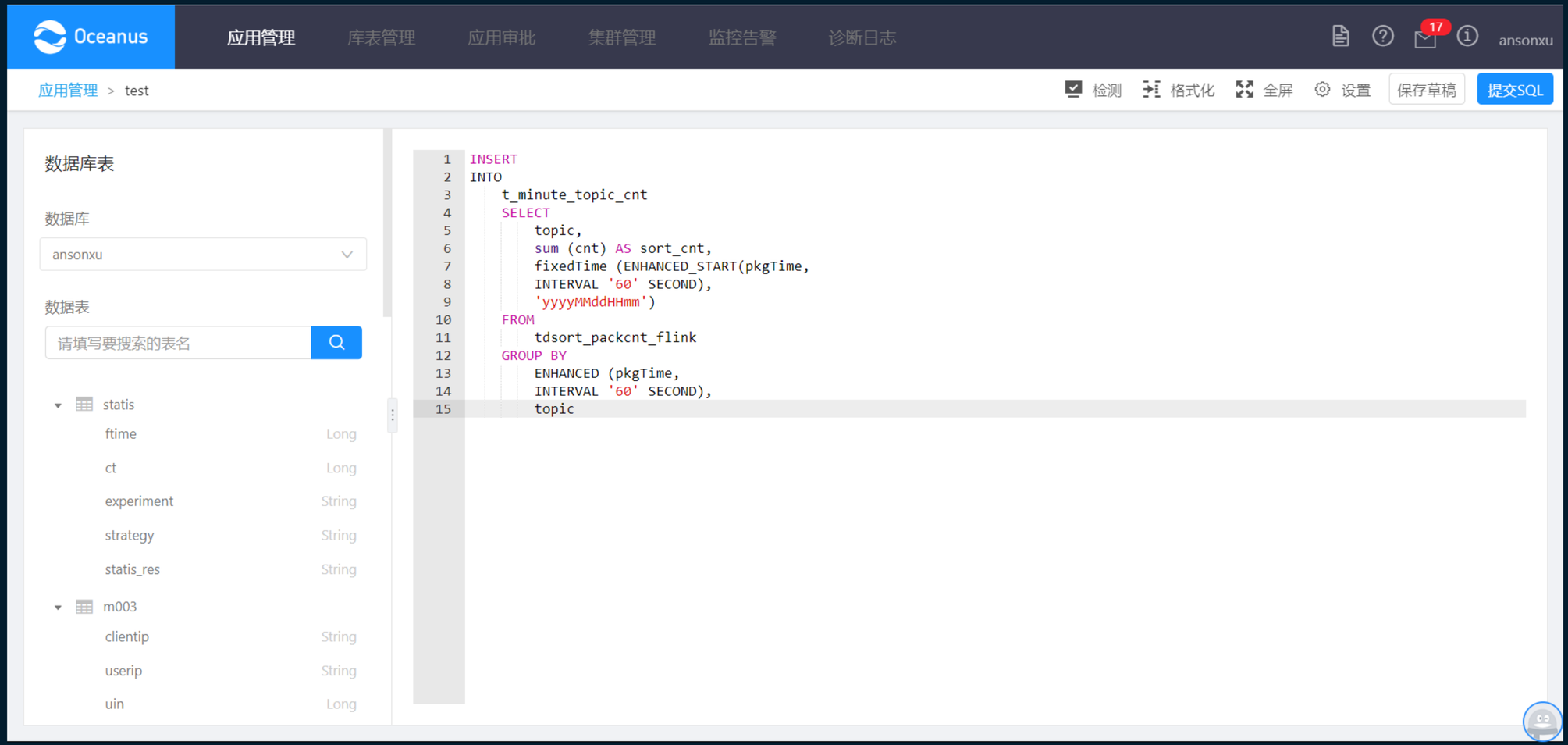
  Allow the handling of timeout events of AsyncIO operators

- **提供了增强窗口来对延迟数据进行更好的处理**

  Allow efficient handling of late elements in windows

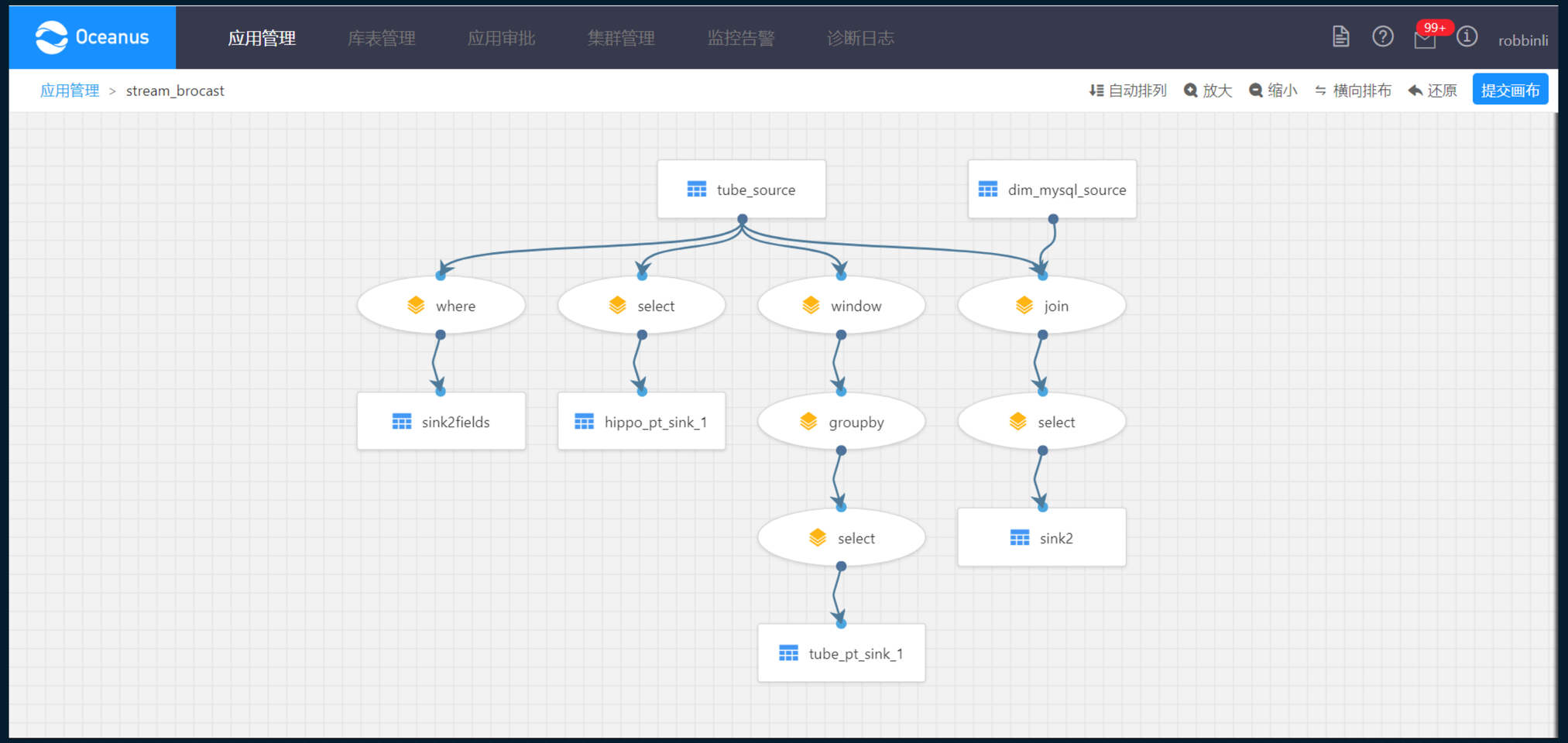
用户案例 – 统计分析

Use case – Data Statistics



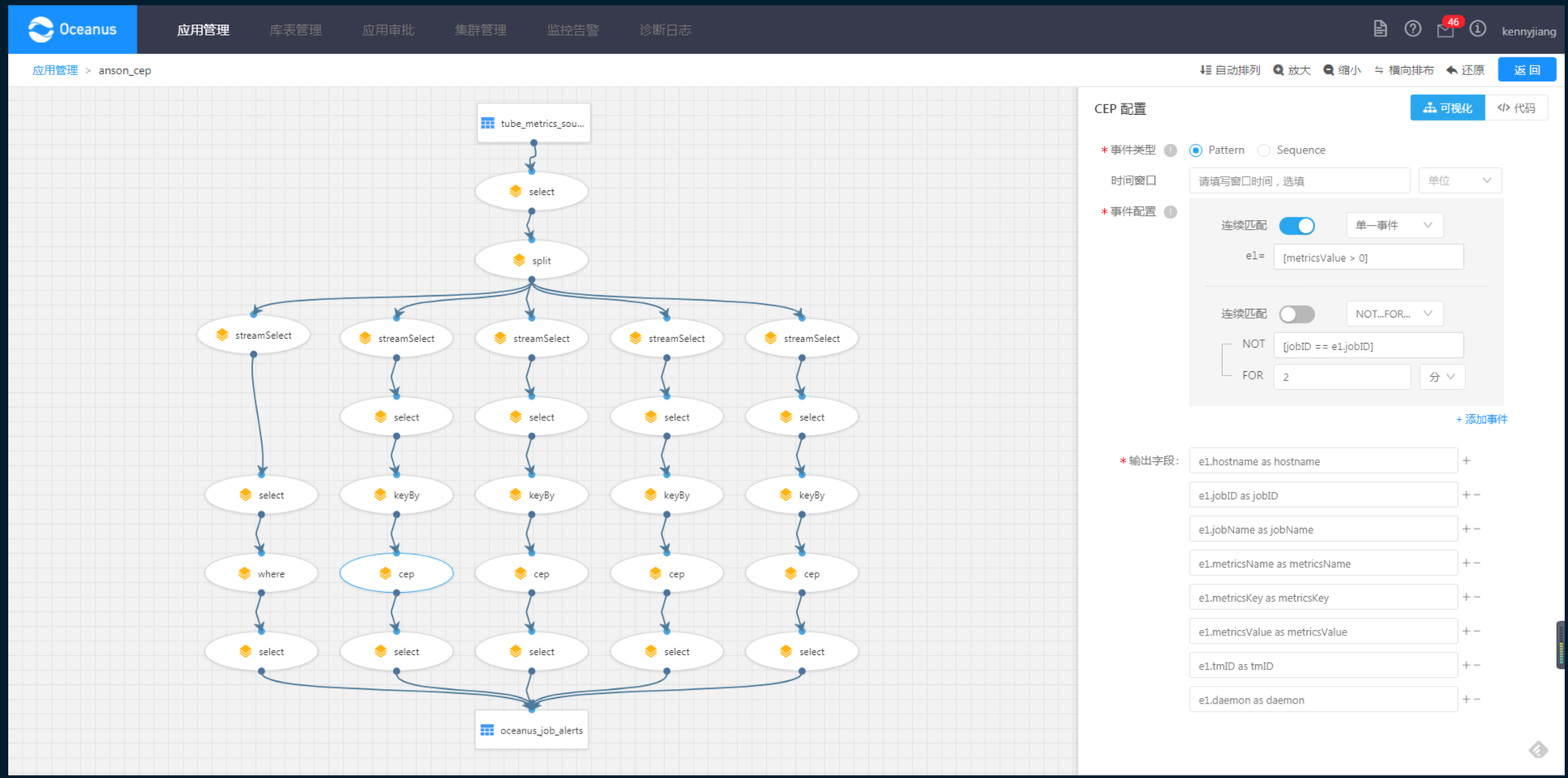
用户案例 – ETL

Use case – ETL



用户案例 – CEP

Use case – CEP



个人介绍

Self-introduction

郑灿彬

felixzheng@tencent.com

腾讯高级工程师，Apache Flink开源社区贡献者，目前在腾讯云AI与大数据基础部门参与流计算服务的研发。4年大数据经验，曾负责Hadoop、Spark等大数据基础组件的运维开发调优工作，目前专注于Serverless流计算服务研发及Flink内核优化。

Stream Compute Service

位于云端的流式数据汇聚、计算服务

面向外部用户

Provide efficient collecting and computing for data streams on Tencent Cloud

SCS 简介

Introduction to SCS

监控系统

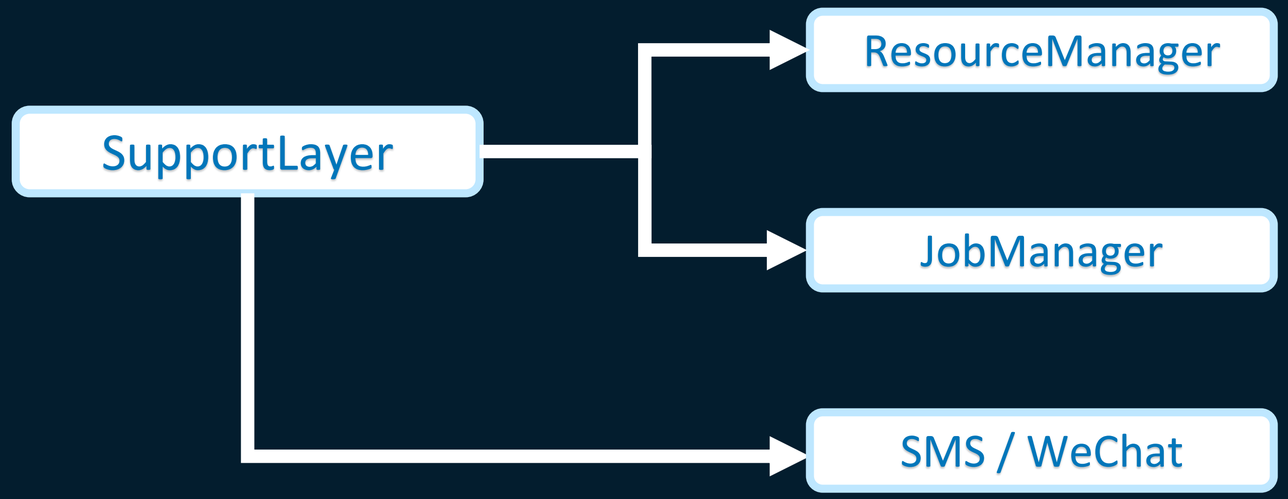
Monitor System

- Flink 作业以 per-job 模式在 YARN 上运行，通过支撑服务周期性检查作业状态

  Flink job runs on YARN in per-job cluster mode, and supportlayer periodically checks app and job status

- 如果作业状态异常，支撑服务发送短信/微信告警

  Once something is wrong with app or job, supportlayer sends SMS, WeChat etc.



State Backend 选择

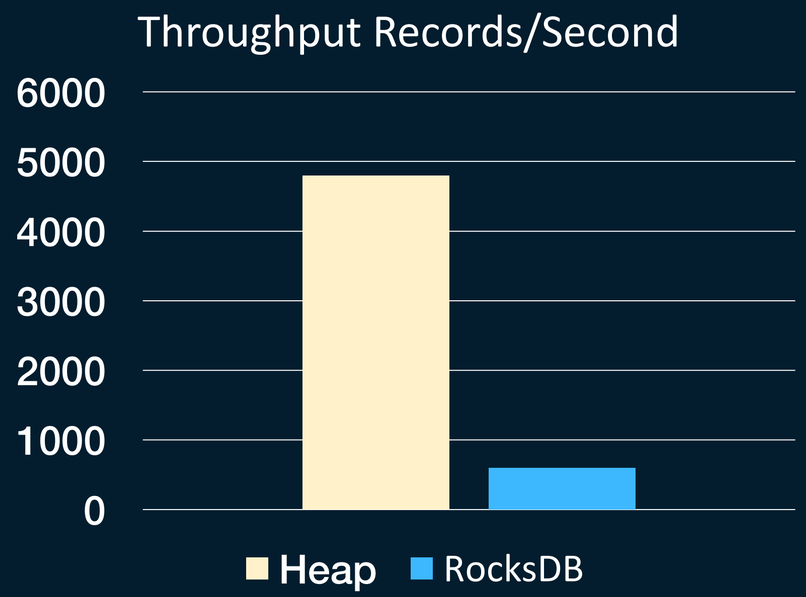
State Backend Puzzle

- HeapStateBackend 能够保证高吞吐低延时，RocksDBStateBackend 则可以保存大量状态

  Heap memory could ensure maximum performance, while RocksDB could store huge number of states

- 同一个作业实例，HeapStateBackend 或者 RocksDBStateBackend 只能二选一

  For a single job instance, you can only choose either HeapStateBackend or RocksDBStateBackend



10% ~ 30%

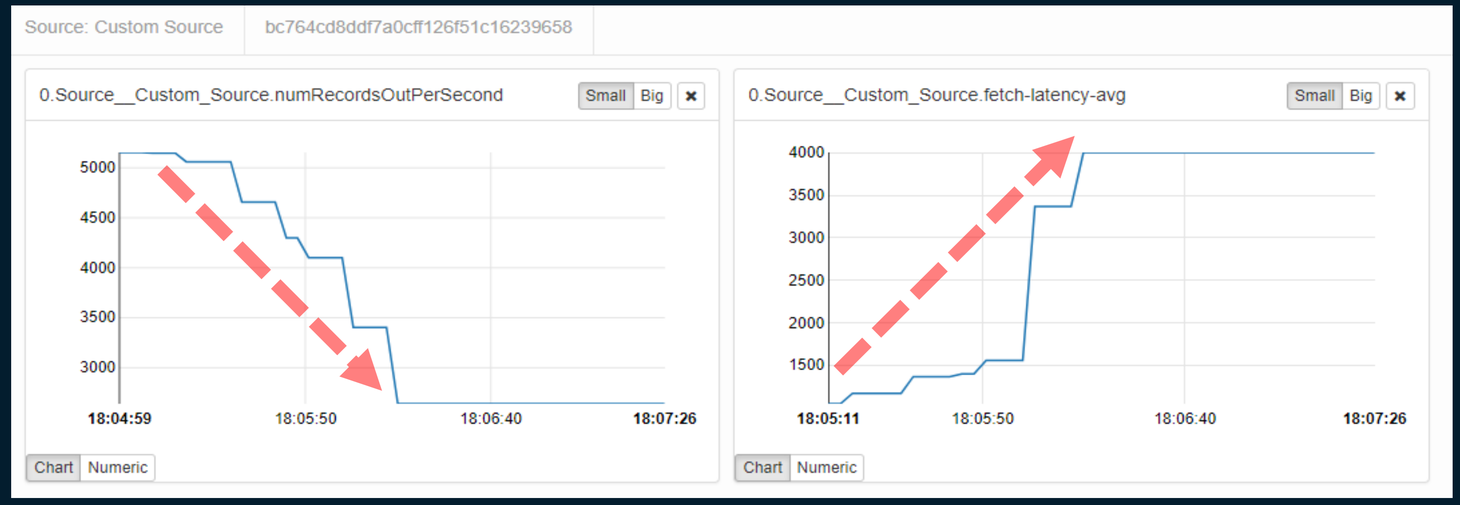
State Backend 选择

State Backend Puzzle

- 一般的处理方式是先使用 Heap，如果出现 OOM，则重新启动实例并切换为 RocksDB

The transitional way is: use heap first, and if OOM happens, restart the job with RocksDB

```
80   -- Continuous JOIN On Two Clickstreams
81   INSERT INTO KafkaSink1
82   SELECT s1.time_, s1.client_ip, s1.uri, s1.protocol_version, s2.status_code, s2.date_
83   FROM KafkaSource1 AS s1, KafkaSource2 AS s2
84   WHERE s1.time_ = s2.time_ AND s1.client_ip = s2.client_ip;
```



挑战

Challenges

- 作业 crash 后才发送告警，我们需要预警

  When alarm arrived, job / app may have already crashed, so we need early warning

- 切换 State Backend 可能会造成状态丢失

  Potential state data loss while switching State Backends

智能监控系统简介

Introduction to Smart Monitor System

- 收集并实时分析作业 Metric 指标，提前发现问题和瓶颈，及时发送告警或者触发其他动作

  Real-time alert & action system for identifying issues and bottlenecks

- 基于 Flink CEP 的规则系统

  Rule system based on Flink CEP library

告警与反馈链路

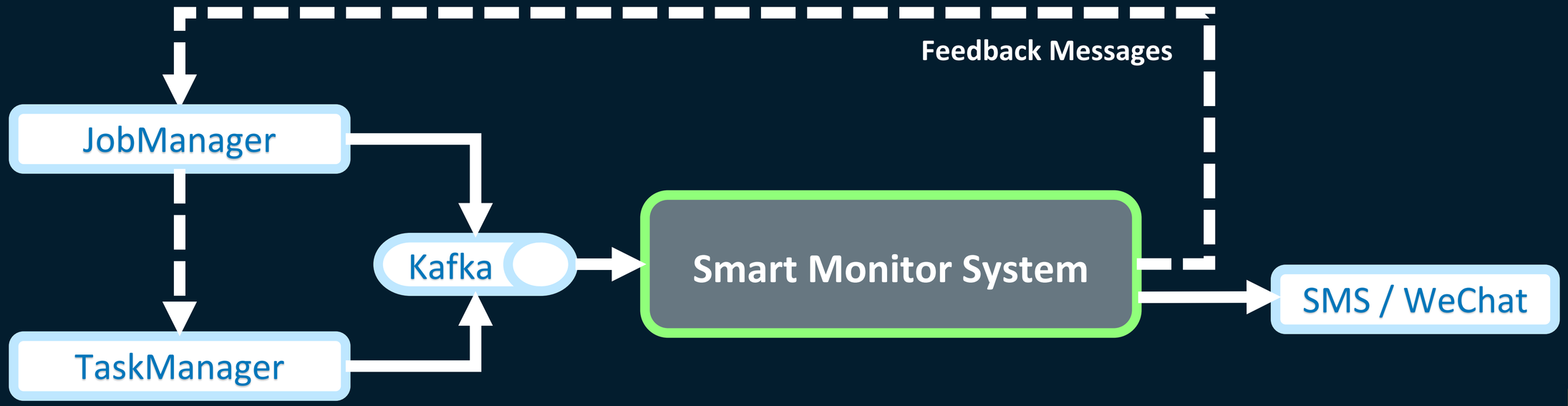
Alert & Action Channels

- 对于普通事件，仅发送告警

  For usual events, send alerts only

- 对于其他事件，则调用 JobManager REST 接口（拓展）触发进一步动作

  For other events, call REST APIs (extended) of JobManager to trigger further actions



告警事件

Alert Events

- 事件级别：INFO，WARN，FATAL

  Event Level: INFO, WARN, FATAL

JobManager Event Name	Level
JOB_MANAGER_HEAP_MEM_LOAD_TOO_HIGH	FATAL
JOB_MANAGER_HEAP_MEM_LOAD_HIGH	WARN
JOB_MANAGER_HEAP_MEM_LOAD_LOW	INFO
JOB_MANAGER_CPU_LOAD_TOO_HIGH	WARN
JOB_MANAGER_CPU_LOAD_HIGH	INFO
JOB_MANAGER_CPU_LOAD_LOW	INFO
JOB_MANAGER_GC_OLD_TOTAL_TIME_LONG	WARN
JOB_MANAGER_GC_OLD_COUNT_QUICKLY_INCREASE	WARN
JOB_MANAGER_FULL_RESTARTS_INCREASE	WARN
JOB_MANAGER_NUM_RUNNING_JOBS_BECOME_ZERO	FATAL
JOB_MANAGER_NUM_TASK_MANAGERS_DECREASE	FATAL
JOB_MANAGER_FAILED_CHECKPOINT_INCREASE	WARN

TaskManager Event Name	Level
TASK_MANAGER_HEAP_MEM_LOAD_TOO_HIGH	FATAL
TASK_MANAGER_HEAP_MEM_LOAD_HIGH	WARN
TASK_MANAGER_HEAP_MEM_LOAD_LOW	INFO
TASK_MANAGER_NON_HEAP_MEM_LOAD_TOO_HIGH	FATAL
TASK_MANAGER_NON_HEAP_MEM_LOAD_HIGH	WARN
TASK_MANAGER_NON_HEAP_MEM_LOAD_LOW	INFO
TASK_MANAGER_CPU_LOAD_TOO_HIGH	WARN
TASK_MANAGER_CPU_LOAD_HIGH	INFO
TASK_MANAGER_CPU_LOAD_LOW	INFO
TASK_MANAGER_OPERATOR_LATENCY_HIGH	WARN
TASK_MANAGER_OPERATOR_DATA_SKEW	WARN
TASK_MANAGER_SOURCE_THROUGHPUT_GRADUALLY_DECREASE	WARN
TASK_MANAGER_SINK_THROUGHPUT_GRADUALLY_DECREASE	INFO
TASK_MANAGER_GC_OLD_TOTAL_TIME_LONG	WARN
TASK_MANAGER_GC_OLD_COUNT_QUICKLY_INCREASE	WARN

监控规则
Monitor Rules

- JobManager 规则
  JobManager Rules
- TaskManager 规则
  TaskManager Rules

Smart Monitor System

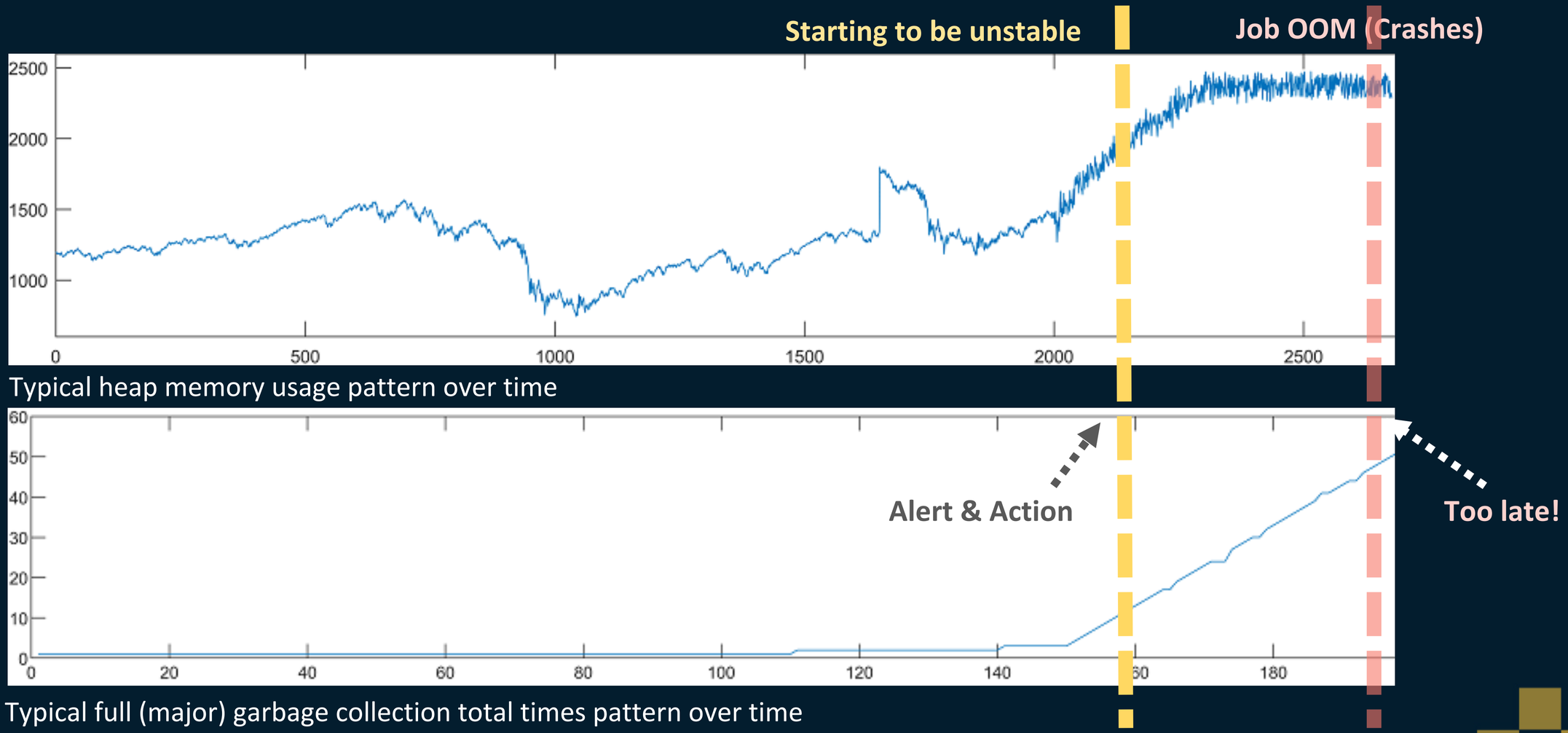
TASK_MANAGER_HEAP_MEM_LOAD_HIGH

Define Pattern

```
.begin(Define Data Gathering Event)
.where(FilterByMetricName("Status.JVM.Memory.Heap.Used"))
.times(10)
                                    …
.followedBy(Define High Used Heap Processing Event)
.where(ProcessEventBasedOnGatheredData(event))
                                    …
.within(Time.minutes(5));
```

典型监控案例

Typical Matched Pattern

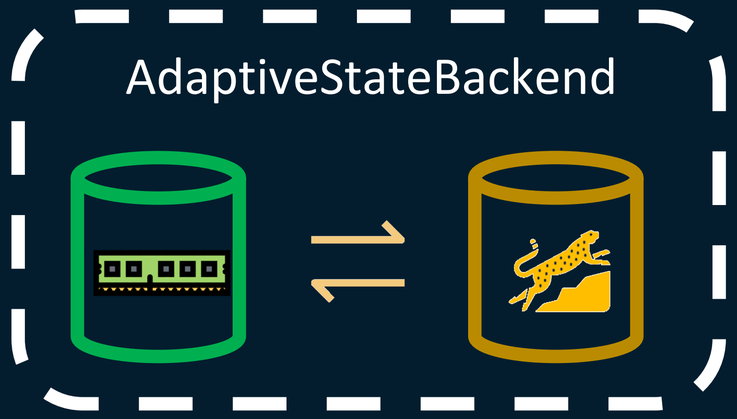


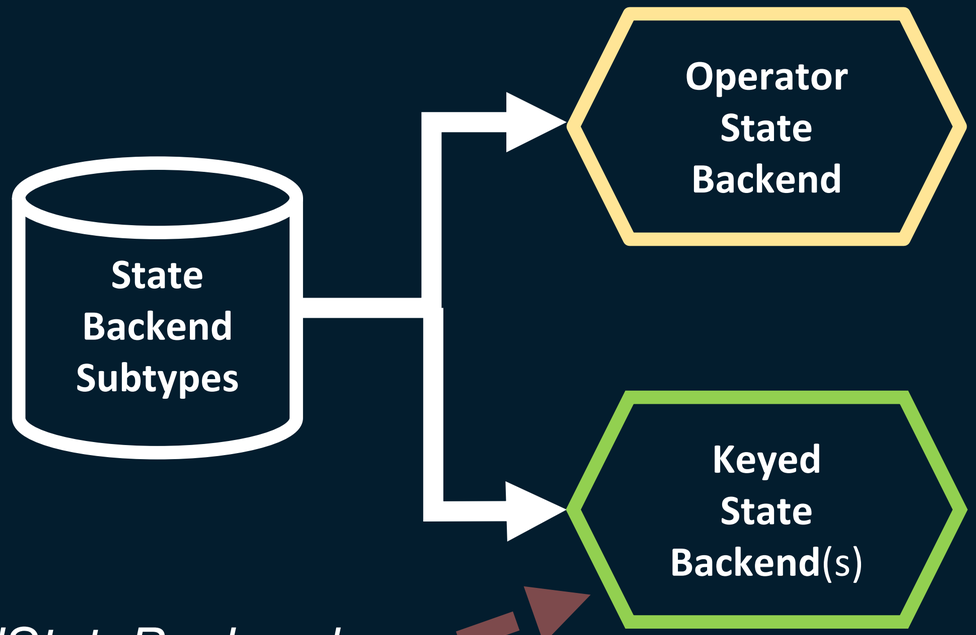
自适应状态管理器

AdaptiveStateManager

- 可以尝试将 HeapStateBackend 和 RocksDBStateBackend 结合起来？如何结合?
  What about combining the two? How?

- 根据用途区分，StateBackend 分为 OperatorStateBackend 和 KeyedStateBackend
  StateBackend could be divided into two categories: OperatorStateBackend and KeyedStateBackend





结合智能监控系统

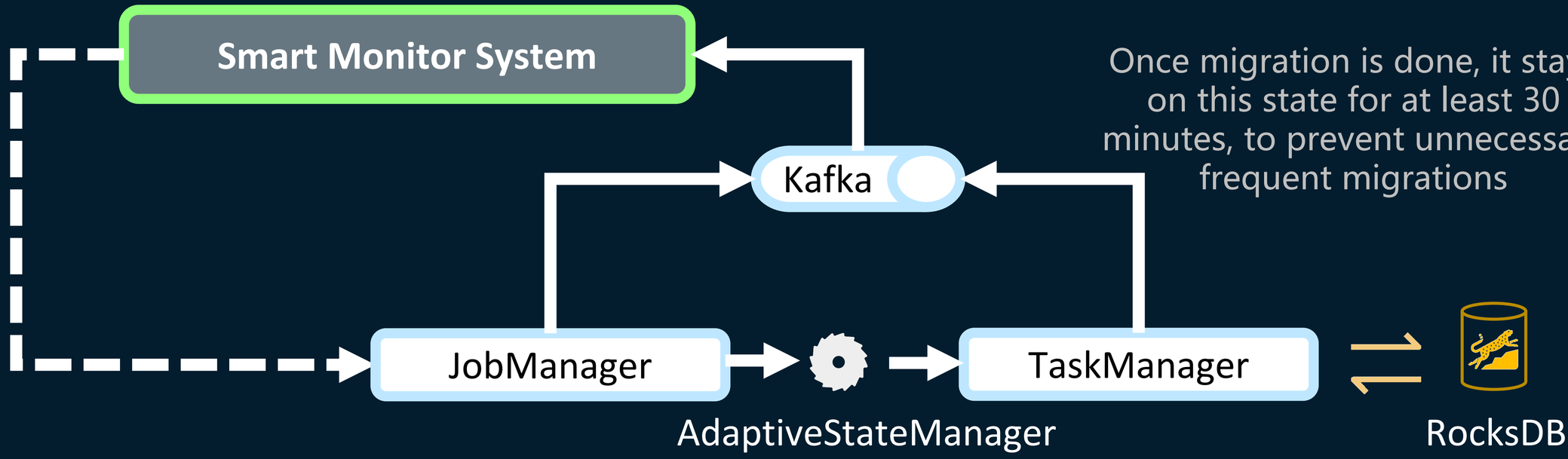
Work with Smart Monitor System

- 实时根据状态负载切换 State Backends

  Live state backend migration scheme to prevent OOM at peak times while keeping a good performance without restarting the job

- 接收来自 JobManager 的 StartMigration 信号并启动 Backends 切换

  Lives in TaskManager and receives StartMigration messages from Smart Monitor System via JobManager

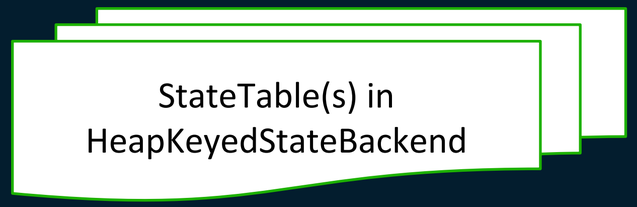


Once migration is done, it stays on this state for at least 30 minutes, to prevent unnecessary frequent migrations

StateBackend切换

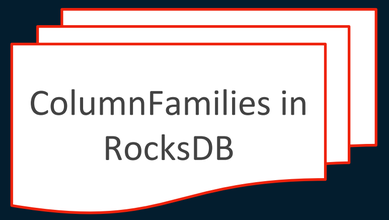
Switch Schema

Collections of StateTable(s) for a specific range of keys:

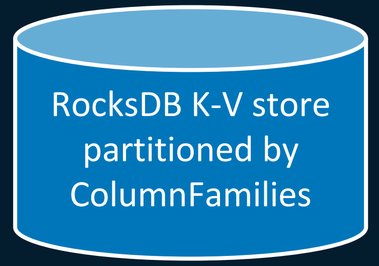


Look at one StateTable:

Collections of ColumnFamily(s):



Namespace + KeyGroupIndex + Key as the access key



优化效果

Overall Performance

- State backend 切换耗时（ 500, 000 记录）， HDD ~10s, SSD <3s, Ramdisk 1~3s

  State backend migration time (500, 000 entries) on HDD ~10s, SDD <3s, Ramdisk 1~3s

- OOM 异常事件明显减少

  OOM events become rare now

- State backend 切换时作业运行较为平稳（波动较少）

  Job runs smoothly while migrating state backend (few jitters)

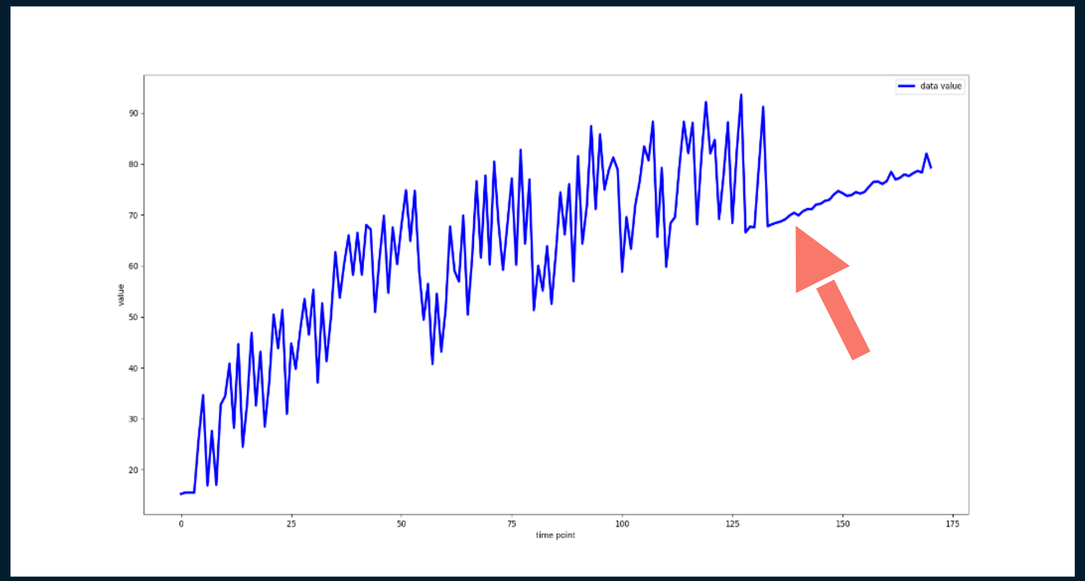
- 不影响作业的故障恢复

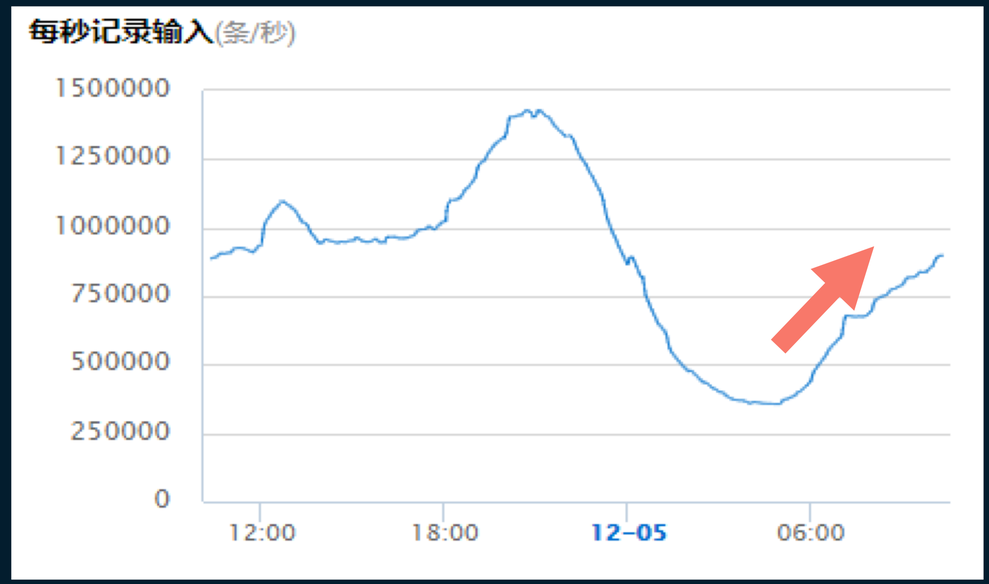
  Resilient and could recover quickly after interruptions

更智能的监控系统

More Intelligent Metrics Monitor System

- 通过机器学习算法预测规则不能发现的潜在异常（统计与无监督 + 分类）
  Predict potential exception that rules can not detect
- 预测业务负载，更精确地评估资源用量（ Arima or LSTM ）
  Predict input traffic for accurate resource usage estimation





未来工作

Future work

- 完善 Metrics 指标

  Add new metrics

- 监控系统更加智能化

  More intelligent monitor system

- 低延时自动扩缩容

  Auto rescale with low latency

FLINK FORWARD

欢迎加入我们

We are hiring


THANKS