Lessons learnt from Migrating to a Stateful Streaming Framework

Wei-Che(Tony) Wei

Software Engineer , Appier

Appier

Outline

- Use case in Appier
- Moving from micro batch to true streaming
- Tips to conquer obstacles during the migration
- Challenge from design in streaming way

Appier

Outline

- **Use case in Appier**
- Moving from micro batch to true streaming
- Tips to conquer obstacles during the migration
- Challenge from design in streaming way

Appier

About Appier

Appier is a technology company which aims to provide artificial intelligence platforms to help enterprises solve their most challenging business problems.

http://www.appier.com/en/index.html

Appier

Use case

- Retargeting
- Fraud detection
- Real time recommendation
- ...

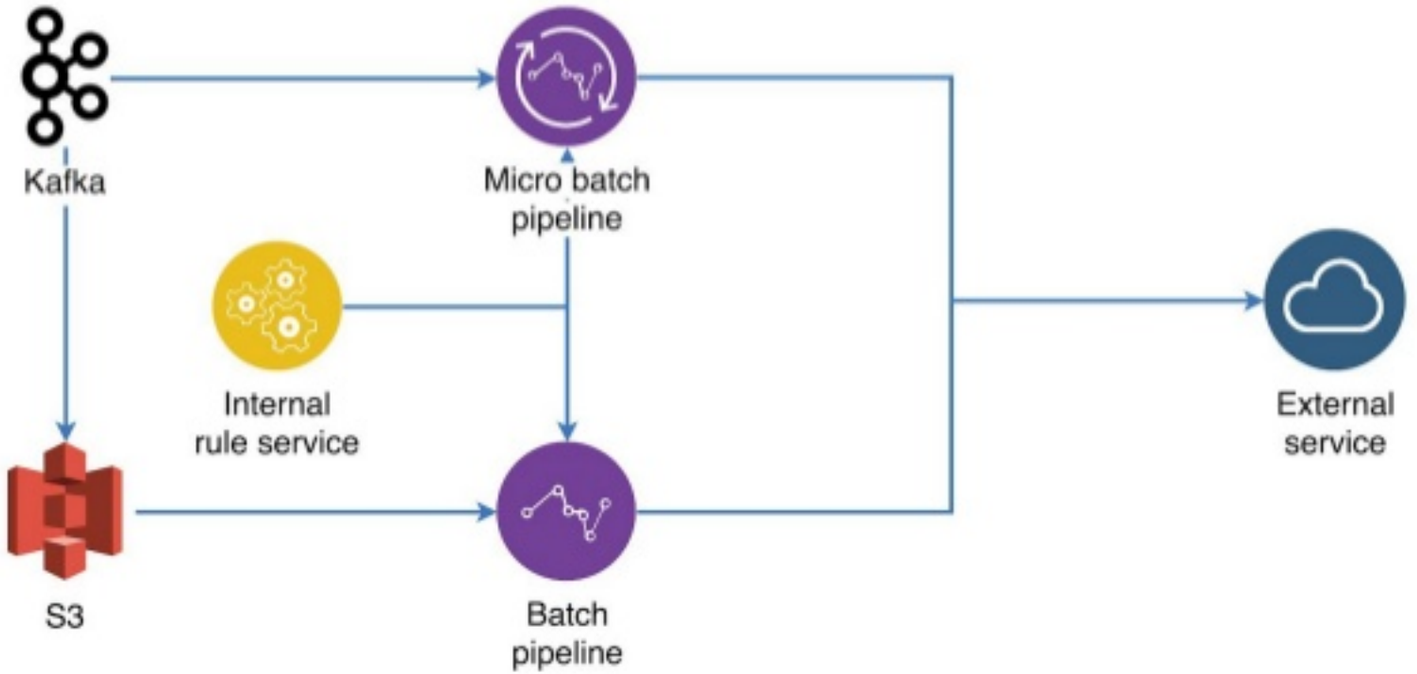
Appier

Outline

- Use case in Appier
- **Moving from micro batch to true streaming**
  - Disadvantage from micro batch
  - New requirements
  - Our solution by using Flink
- Tips to conquer obstacles during the migration
- Challenge from design in streaming way

Appier

Dynamic Rule Service: Previous design



Appier

Disadvantage from micro batch

- High latency
- Struggle with back pressure
- Hard to maintain states

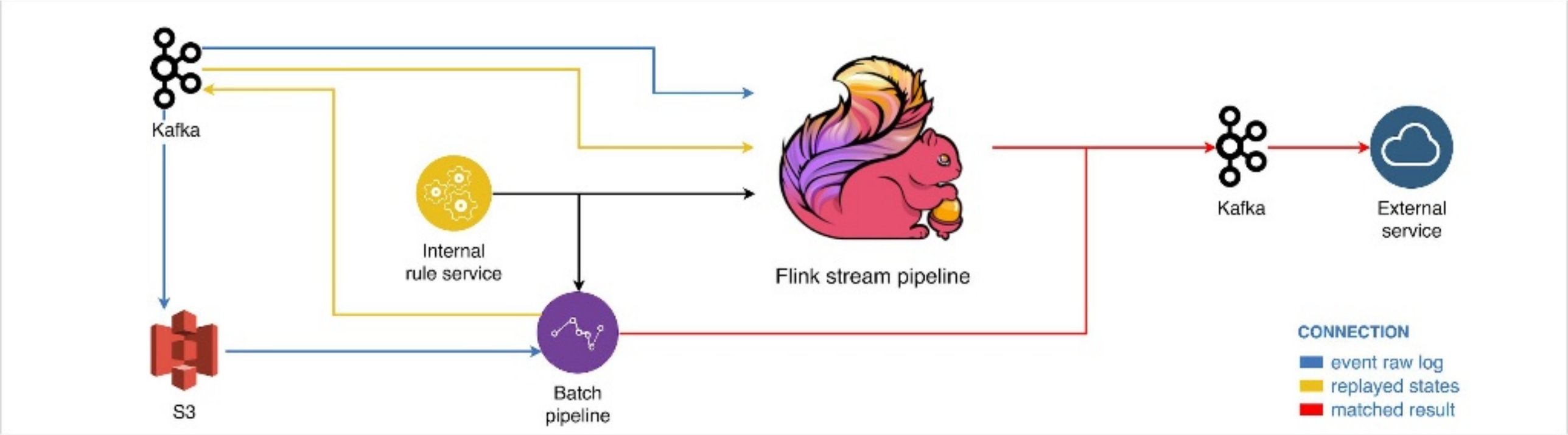
Appier

New requirements

- “Who visited this site twice in the past week”
- “Who viewed this product in the site but didn’t put into cart”
- “Who visited the certain pattern of some pages and bought this product”

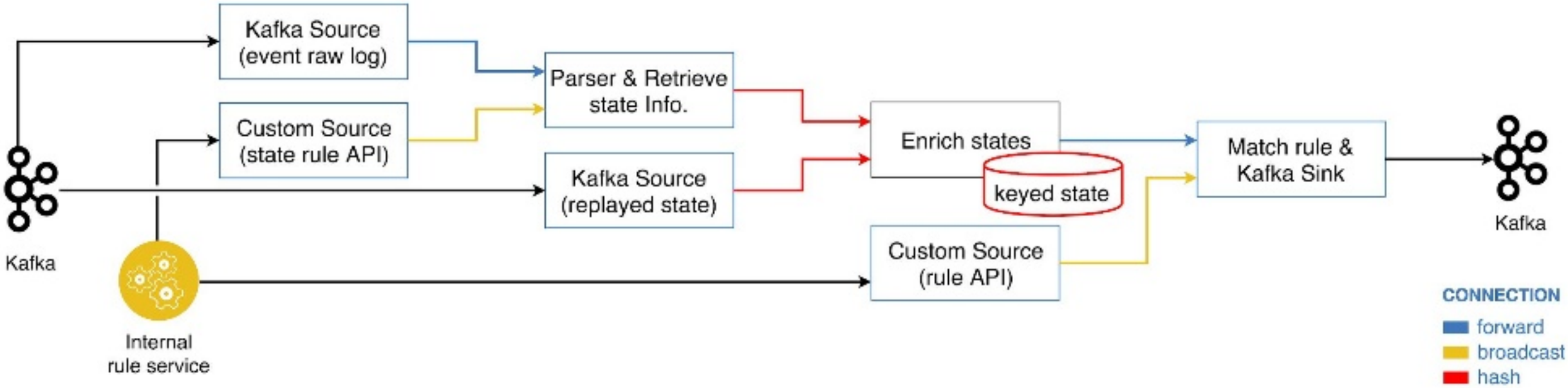
Appier

Dynamic Rule Service with Apache Flink



Appier

Detail of JobGraph (Apache Flink 1.4.0 release)



Appier

Improvements

- Support more needs for stateful rules
- Flexible architecture
- Performance efficiency
- Cost efficiency

Appier

Outline

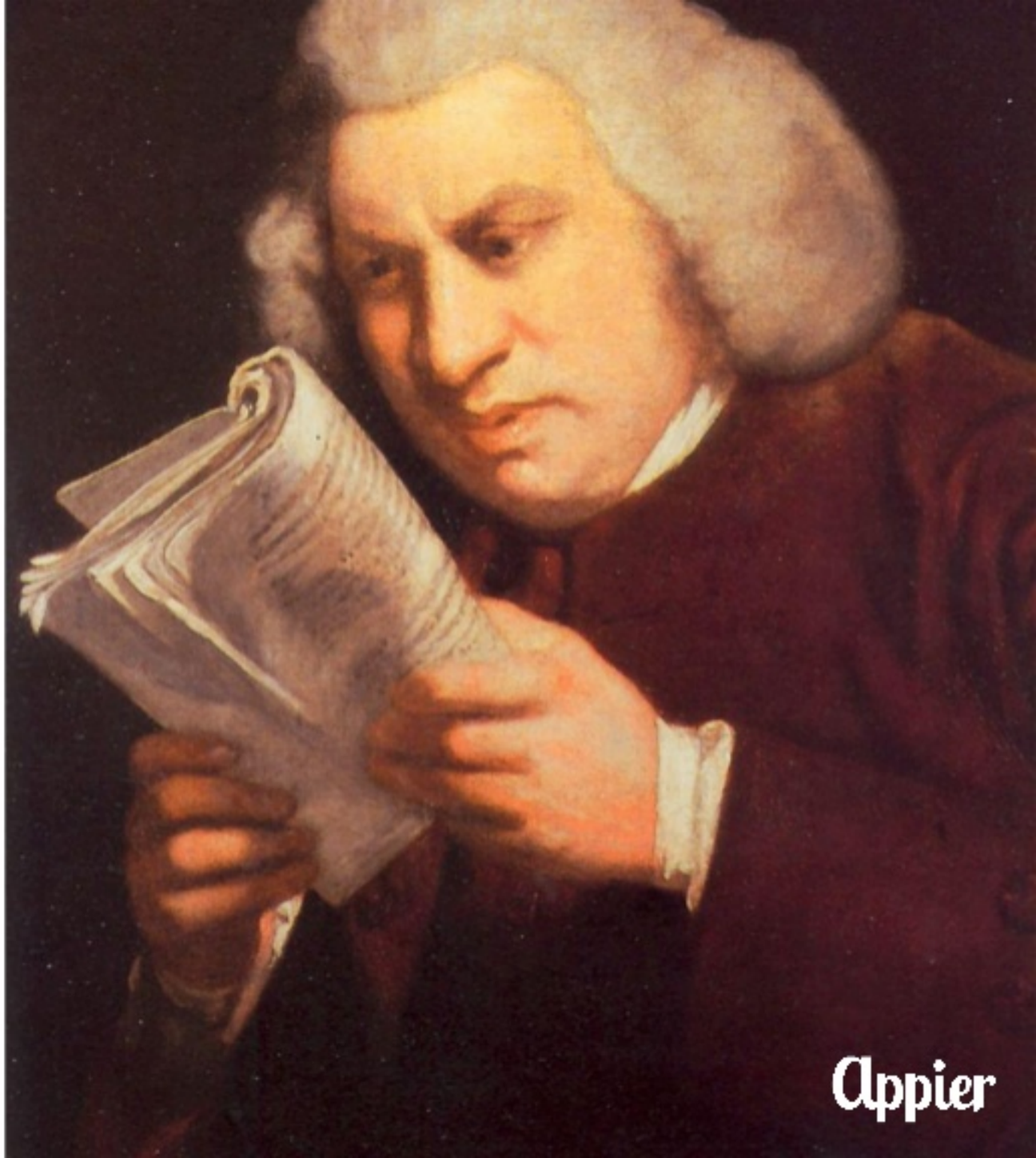
- Use case in Appier
- Moving from micro batch to true streaming
- **Tips to conquer obstacles during the migration**
- Challenge from design in streaming way

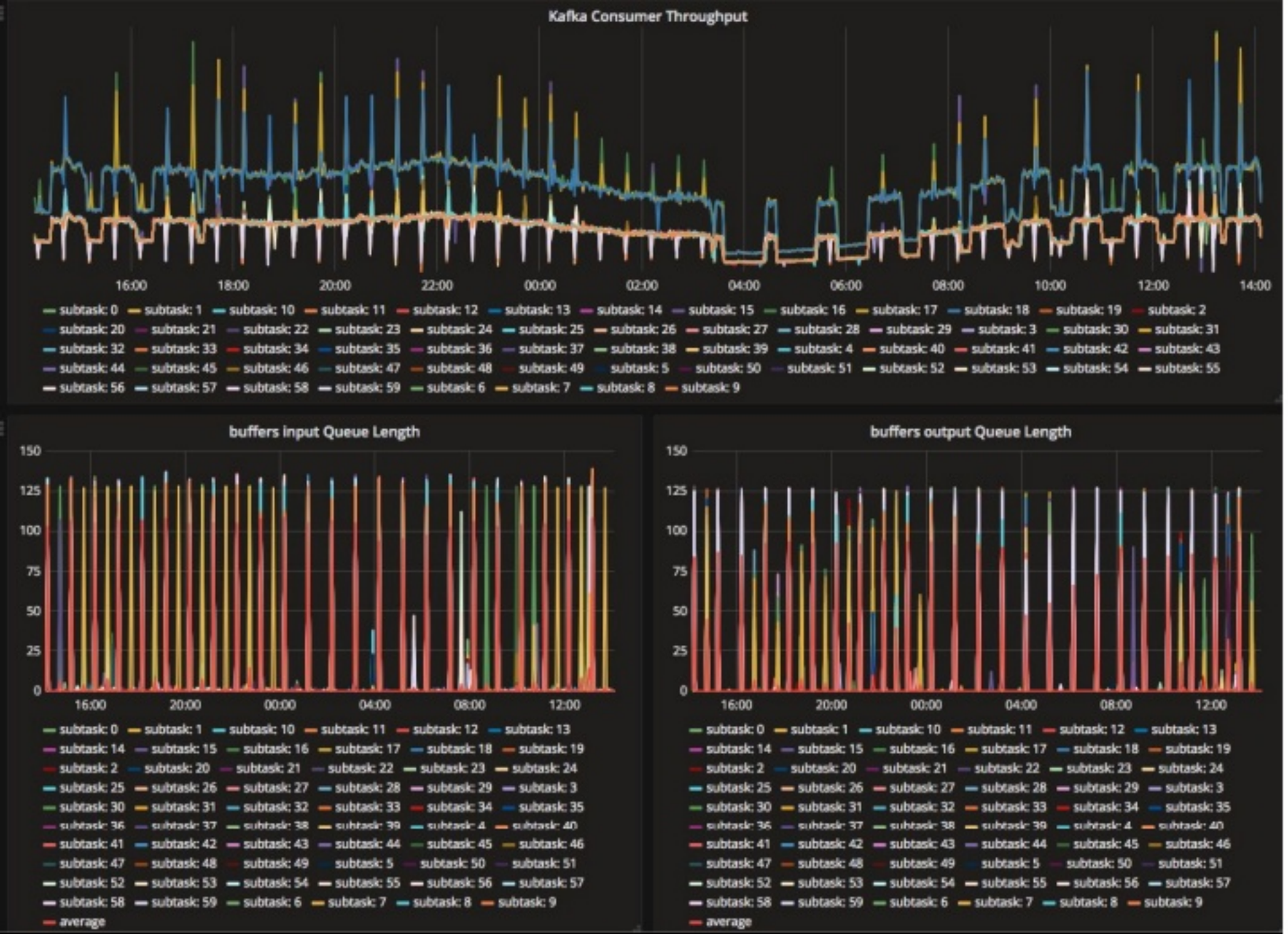
Appier

Tip 1:
Document and Mailing List are your best friends

Case Study:
taskmanager.exit-on-fatal-akka-error: true

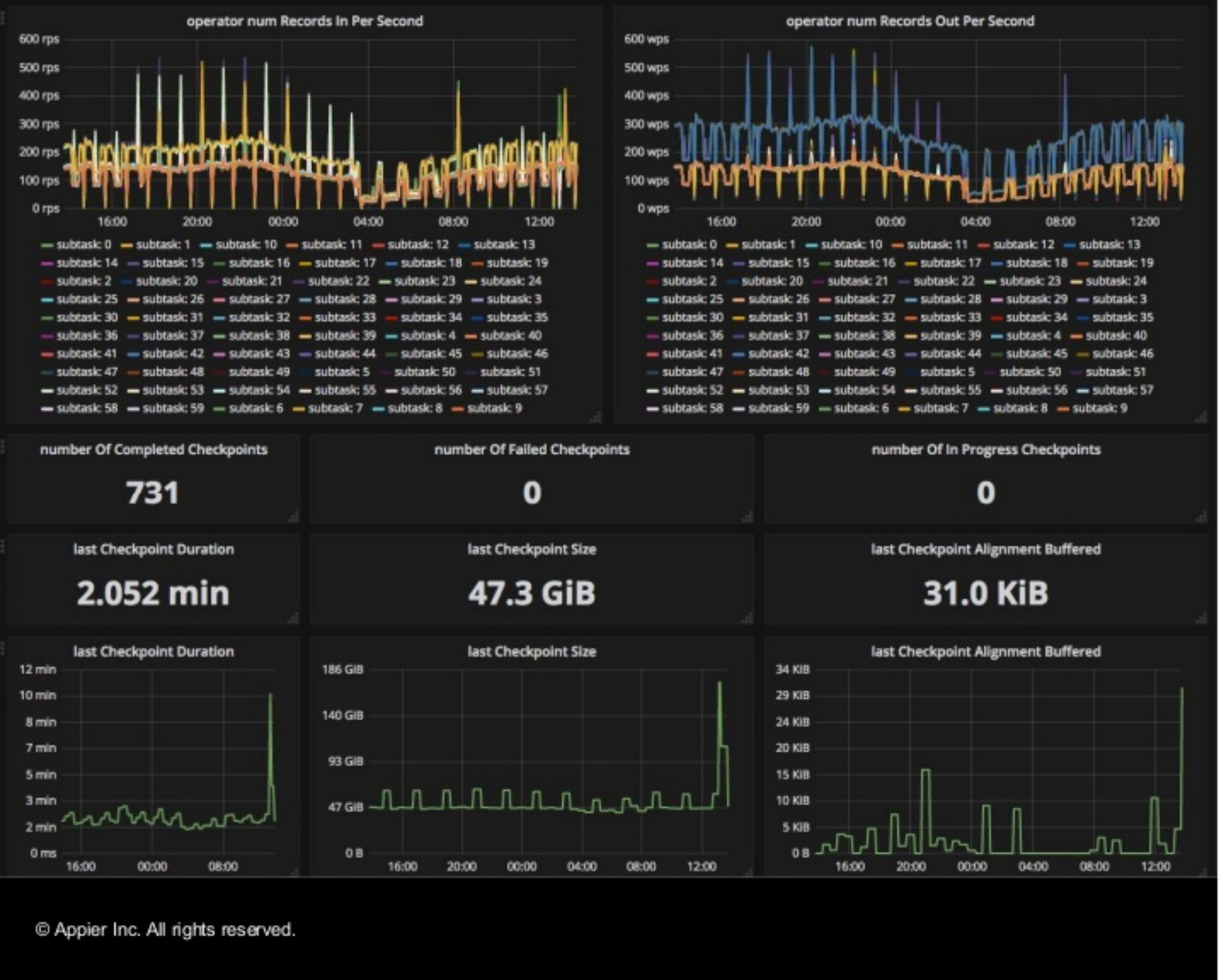
https://ci.apache.org/projects/flink/flink-docs-release-1.4/ops/config.html#jobmanager--taskmanager





Tip 2:
Monitor and alert are important

Appier



Tip 2:
Monitor and alert are important

Appier

Tip 2:
Monitor and alert are important

Appier

Tip 2:
Monitor and alert are important

Appier

Prometheus Alerts Graph Status Help

FlinkCheckpointAlignmentTimeOver30Secs (0 active)

```
alert: FlinkCheckpointAlignmentTimeOver30Secs
expr: max(flink_taskmanager_job_task_checkpointAlignmentTime)
  BY (job_name, job_id, task_attempt_num, task_name) / 1000 / 1000 / 1000 > 30
for: 100m
labels:
  receiver: rt-alert
annotations:
  description: 'Flink Job: {{$labels.job_name}}({{$labels.job_id}}) checkpoint alignment
    time takes more than 30 seconds on task {{$labels.task_name}}'
```

FlinkCheckpointDuration10Mins (0 active)

FlinkCheckpointFailed (0 active)

FlinkProcessTimeOver30Secs (0 active)

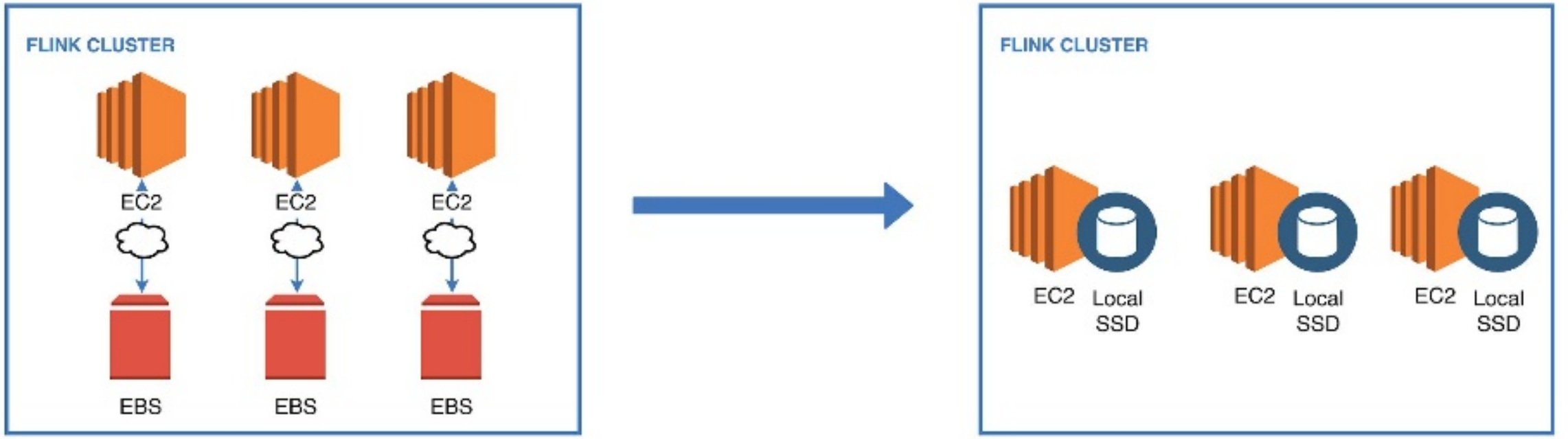
```
alert: FlinkProcessTimeOver30Secs
expr: sum(increase(flink_taskmanager_job_task_operator_process_time_latency{operator_name="match_rule"}[2m]))
  BY (job_name, job_id, task_attempt_num) / sum(increase(flink_taskmanager_job_task_operator_numRecordsIn{operator_name="match_rule"}[2m]))
  BY (job_name, job_id, task_attempt_num) > 30000
for: 10m
labels:
  receiver: rt-flink-alert
annotations:
  description: 'Flink Job: {{$labels.job_name}}({{$labels.job_id}}) process time takes
    more than 30 seconds for 10 minutes.'
```

FlinkTaskManagerHighGC (0 active)

```
alert: FlinkTaskManagerHighGC
expr: max(increase(flink_taskmanager_Status_JVM_GarbageCollector_G1_Old_Generation_Time[10m])
  + increase(flink_taskmanager_Status_JVM_GarbageCollector_G1_Young_Generation_Time[10m]))
  WITHOUT (tm_id) / 10 / 60 / 1000 > 0.03
for: 30m
labels:
  receiver: rt-flink-alert
annotations:
  description: Flink Taskmanager instance {{$labels.instance}} high GC ratio
```

Tip 2: Monitor and alert are important

Appier

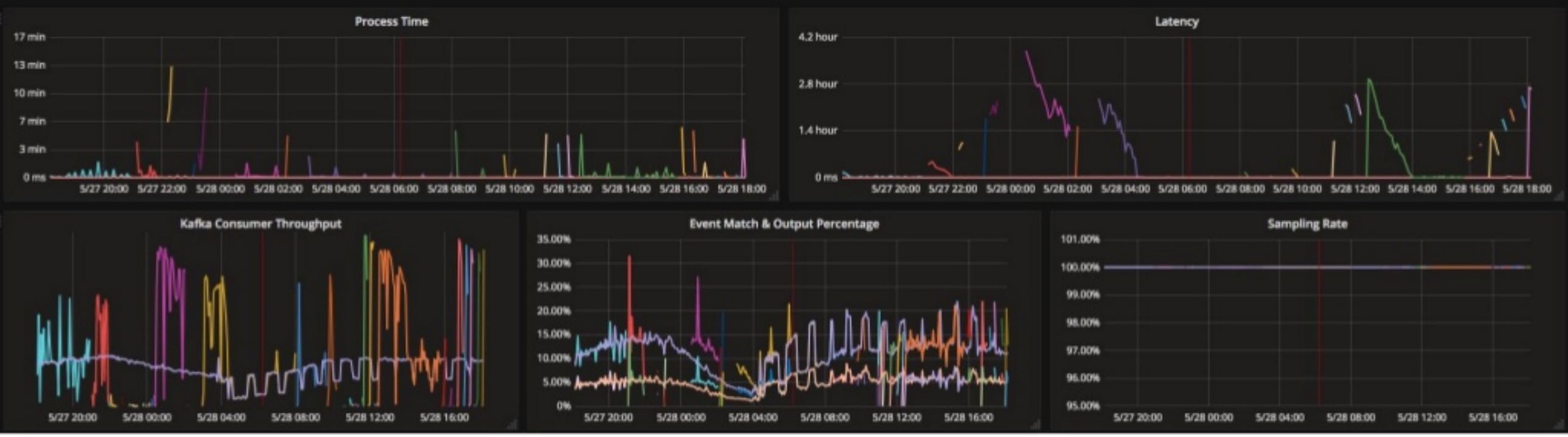


Tip 3:
Be familiar with your environment and your job

Case Study:

checkpoint stuck with rocksdb statebackend and s3 filesystem

Appier



Tip 3:

Be familiar with your environment and your job

Case Study: Verify bottleneck of our streaming job

- ✗ network performance issue
- ✔ memory bound job met the resource limitation

Appier

Outline

- Use case in Appier
- Moving from micro batch to true streaming
- Tips to conquer obstacles during the migration
- **Challenge from design in streaming way**

Appier

Challenge from design in streaming way

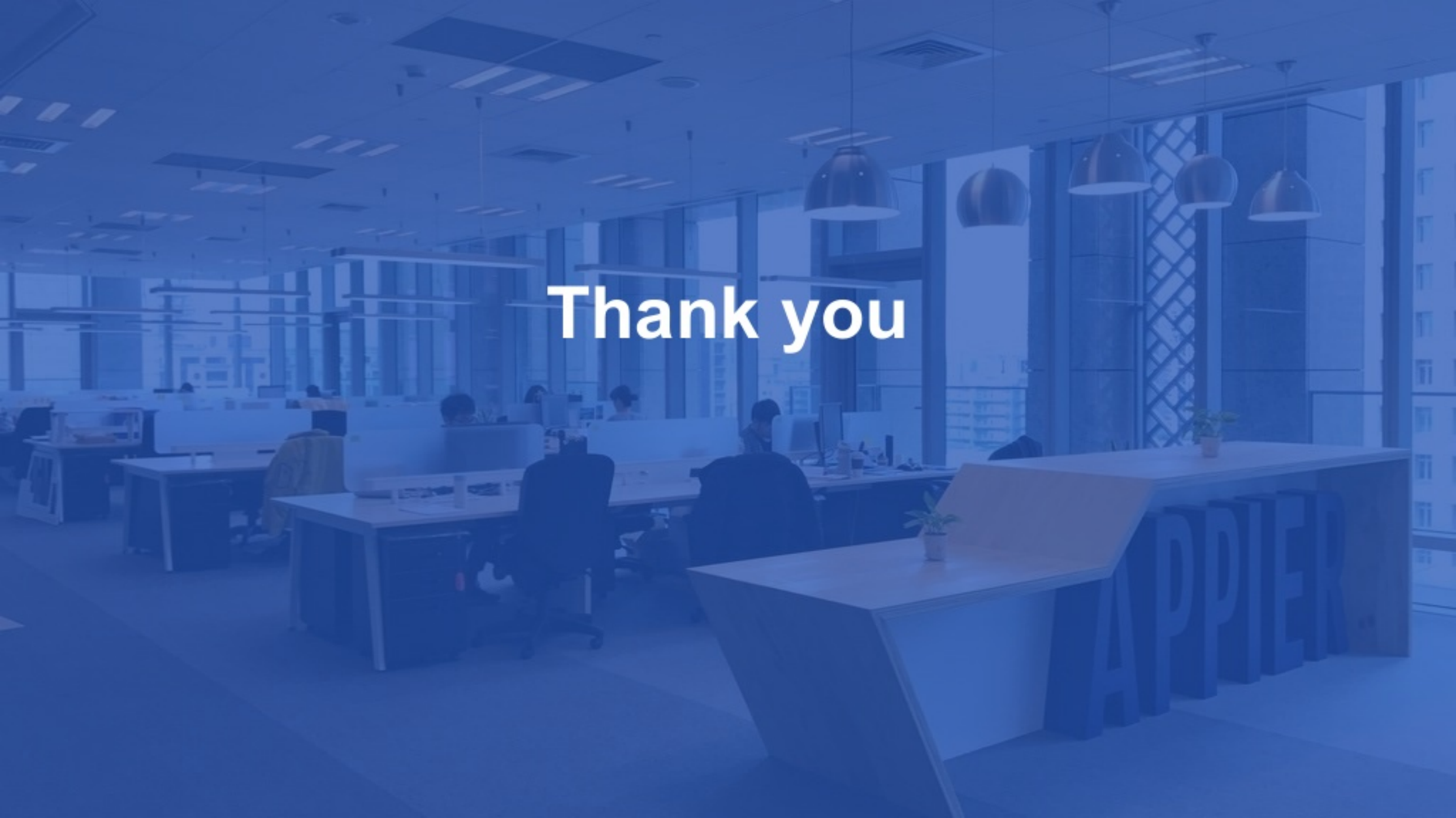
- **Replay expired data or rebuild states are complex**
- **How to expose user-defined metrics well**
  - Current matched count for each rule
    - Too expensive to use querible states
- **End-to-end verification is a tough job**
  - How to verify those replay data is prepared

Appier

TL;DR

- Documents and information from mailing list help a lot.
- Monitor and alert let you respond to problem and diagnose it quickly.
- Stateful streaming is environment sensitive. Be careful of it.
- Community is always behind you.

Appier


Thank you
APPIER