Large Scale Real Time Ad Invalid Traffic Detection with Flink

Feng LI, Juan Gentile - Criteo

Flink在大规模实时无效广告流量检测中的应用

李锋, Juan Gentile – Criteo

FLINK FORWARD

Criteo: Our Mission

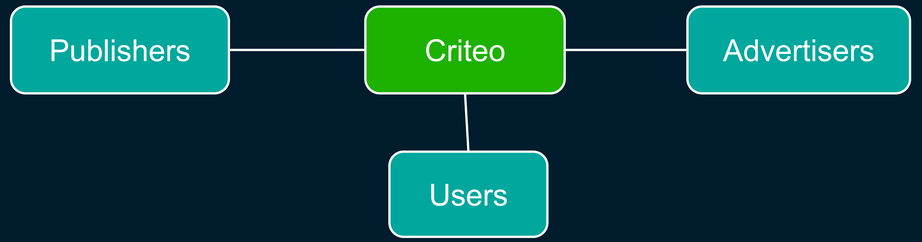
criteo

愿景

criteo

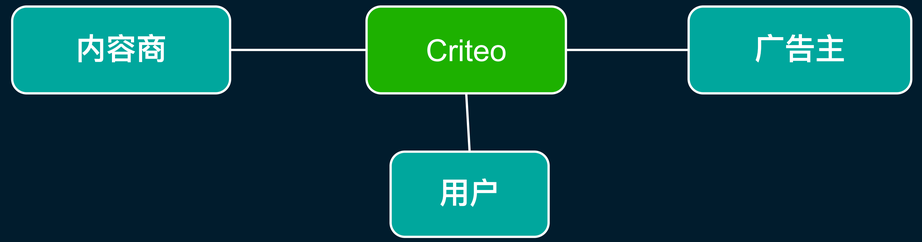
Criteo: Business Model

- Personalized Ads with Real Time Bidding
- ML for Optimization/Recommendation



Criteo：商业模式

- 通过实时竞标系统进行个性化广告推荐
- 机器学习进行竞标，以及广告推荐



FLINK FORWARD

Criteo: In Numbers

- Processing 300+ Billion Requests Daily
- Displaying 120+ Billion Ads Daily
- Peak Time 4M+ QPS for RTB

- 2 CDH5 CLUSTER OF 4000 NODES (92TB, 200 GB RAM, 48 cores)

criteo

Criteo：一些数字

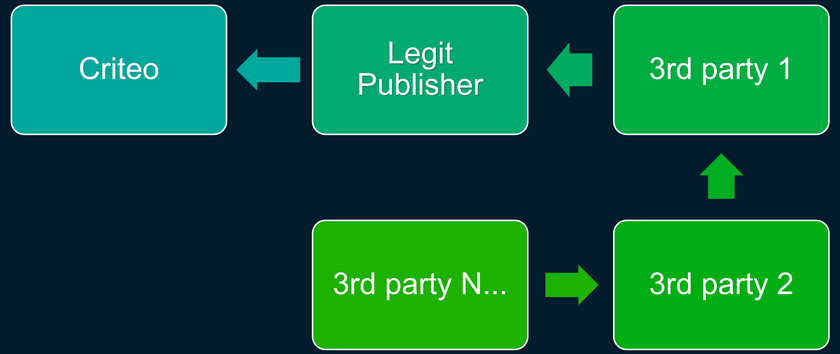
- 每天处理3000亿个竞标请求
- 每天展示1200亿个广告
- 峰值4百万 QPS

- 4000个节点的 Hadoop 集群(92TB, 200 GB RAM, 48 cores)

criteo

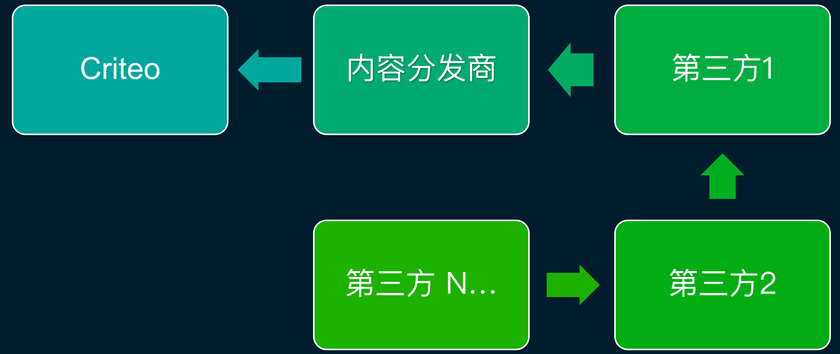
FLINK FORWARD

Classical Ad Fraud Scenario



criteo

广告欺诈的一个场景



criteo

FLINK FORWARD

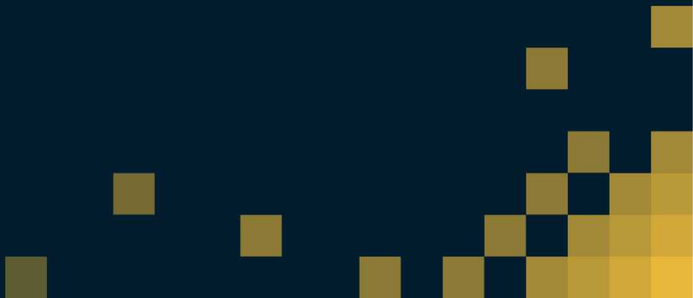
criteo

How to Detect

- Understand what are non-human behaviors
- Define a set of rules to detect those non-human behaviors and mitigate

怎么检测广告欺诈

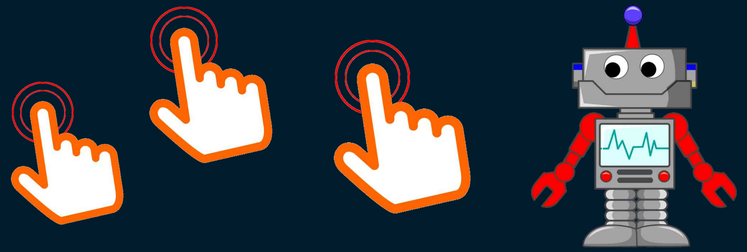
- 理解分辨哪些是机器行为
- 设定一系列的规则检测并且过滤这些行为



criteo

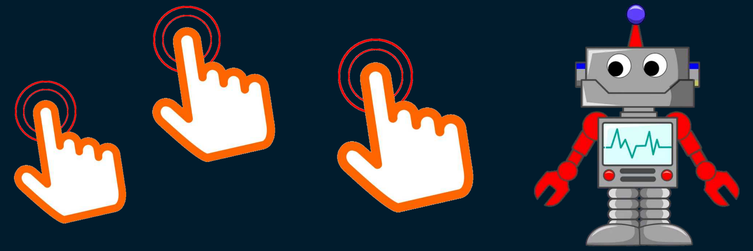
Toy Example: Over Clickers

- We need to detect and mitigate all the users that clicks more than X during last hour (sliding window)



简单实例：过量点击者

我们需要过滤所有在过去一小时点击广告量超过 X 的用户



FLINK FORWARD

criteo

Architecture

架构

criteo

Detection: What Matters

- Delivery Guarantee
- Fault Tolerant
- State Management
- Performance
- Advance features (Event Time Processing, Watermark Management, Windowing)

检测模块需要

- Delivery guarantee
- 容错性
- State 管理
- 性能
- 有用的特性（事件事件处理，watermark 管理，窗口支持）

Spark Streaming Restrictions

- Mini-batch
- Too many parameters to tune, hard to get it right
- Processing time only (at the time)
- Stateless by nature

Spark 的局限性

- Mini–batch
- 太多的参数需要调整，而且很难调整对
- 不支持事件时间处理 （当时）
- 没有 state

FLINK FORWARD

criteo

Kafka Stream Limitations

- Low level api
- No complex streaming features (watermarks, windowing)
- Not for heavy lifting work

Kafka Stream 的局限

- 太多底层 api
- 不支持很多 streaming 特性（比如 watermarks，窗 etc）
- Not for heavy lifting work

criteo

FLINK FORWARD

Streaming Done Right

- Flexible event time support
- Robust state management
- Short development cycle

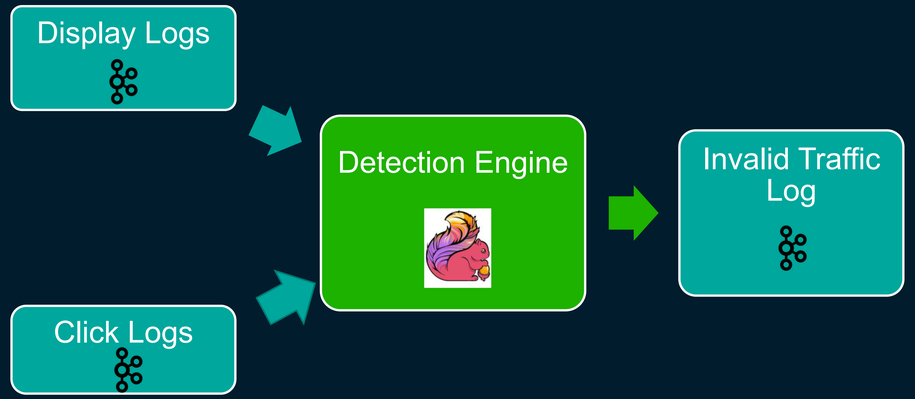
criteo

Streaming 需要

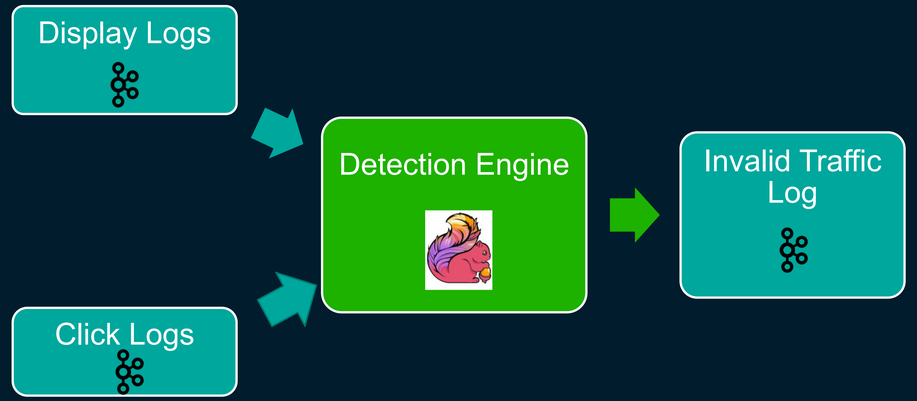
- 灵活的事件时间支持
- 稳定的 state 管理
- 快速的开发时间

criteo

Detection Architecture

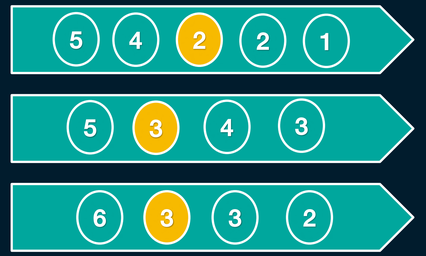


欺诈检测的架构



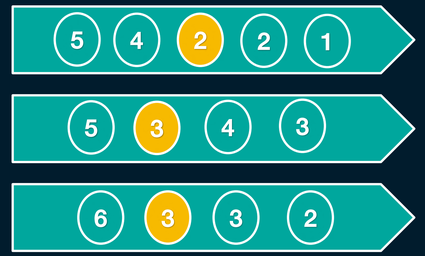
FLINK FORWARD

Time Series Management



- Single Kafka partition is ordered
- Time of partition max(watermark)
- Global time min(partitions)

时间序列管理

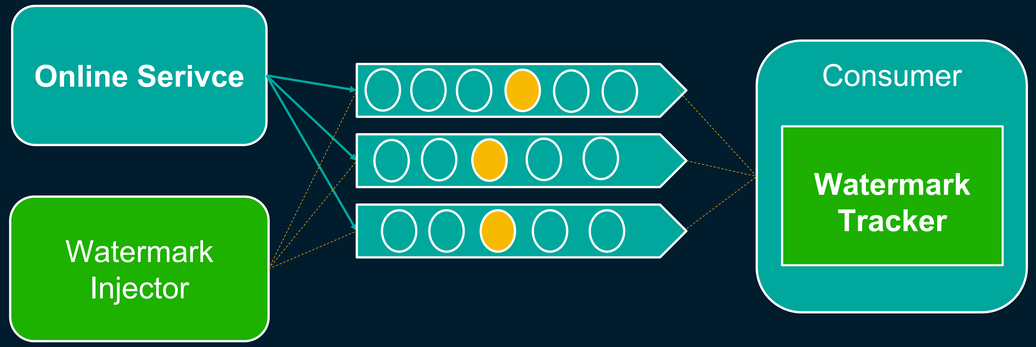


- 单个 kafka 分区是有序的
- 分区时间 max(watermark)
- 全局时间 min(partitions)

criteo

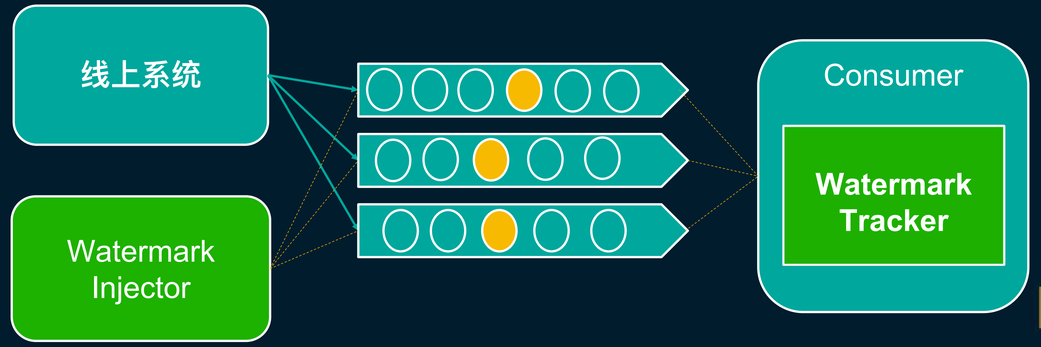
FLINK FORWARD

Time Series Management



criteo

时间序列管理



criteo

A Cross DC Story

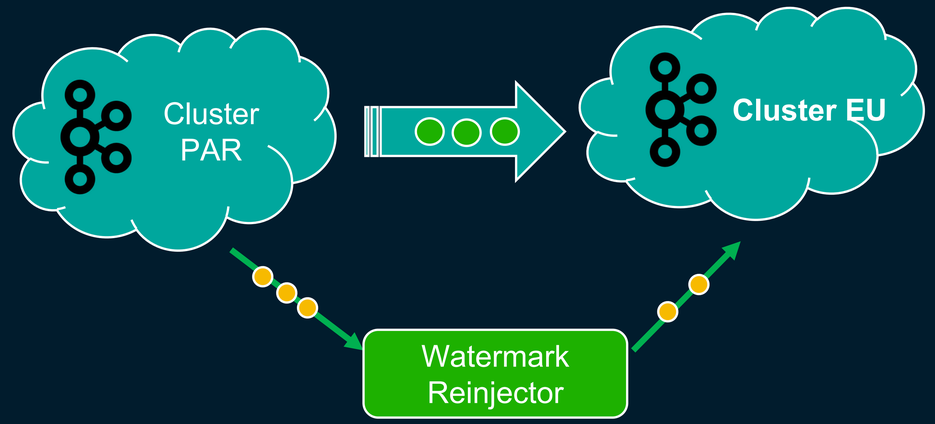
- Our detection needs cross multiple DCs
- We replicate kafka streams of multiple DCs to one
- We need to reinject watermarks (source and dest kafka topic might have different number of partitions)

跨数据中心

- 我们的检测需要跨越多个数据中心
- 我们线上复制多个数据中心 kafka 集群到一个总的 kafka 集群
- 我们需要重新注入 watermark

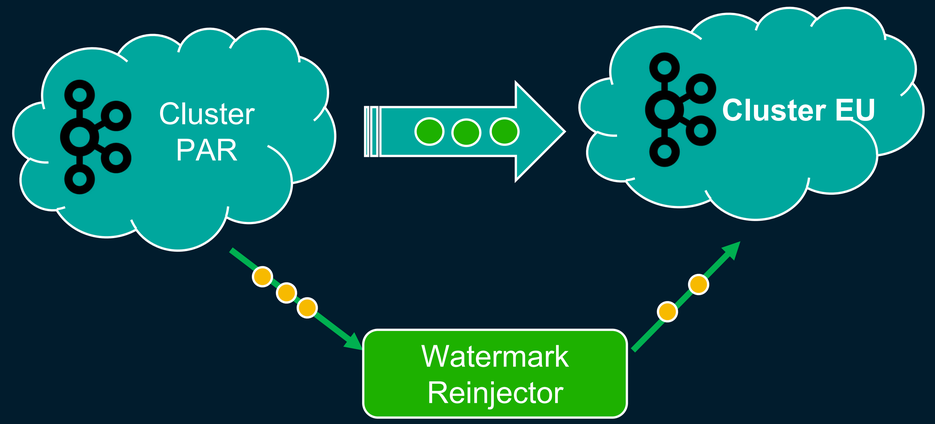
FLINK FORWARD

A Cross DC Story



criteo

跨数据中心



criteo

Checkpointing

We use RocksDB State Backend

- Best for large state, long windows
- Incremental Checkpointing
- serialization (or deserialization) needed to cross the JNI boundary

Checkpointing

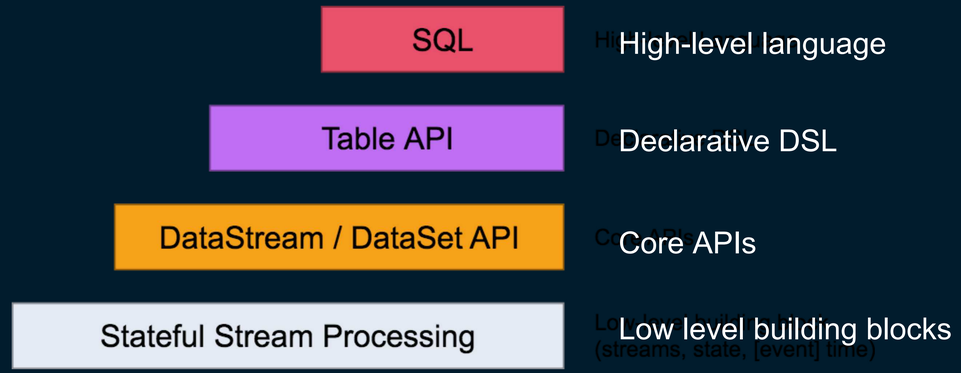
我们使用 RocksDB 作为 state 后端

- 对于大的 state，长 window 工作的很好
- 递增式 checkpointing
- 需要序列化

FLINK FORWARD

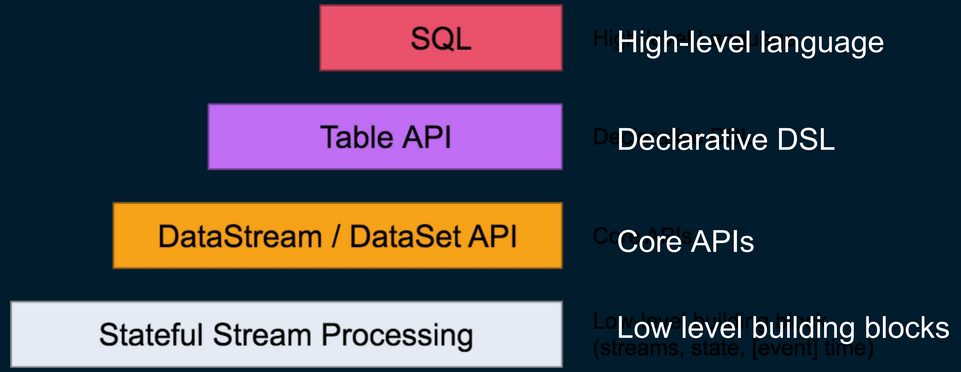
criteo

Flink SQL



criteo

Flink SQL



Toy Example: Over Clickers

```
SELECT uid, ...
FROM click_log
GROUP BY uid, HOP(`timestamp`, interval '5' minute,
interval '1' hour)
```

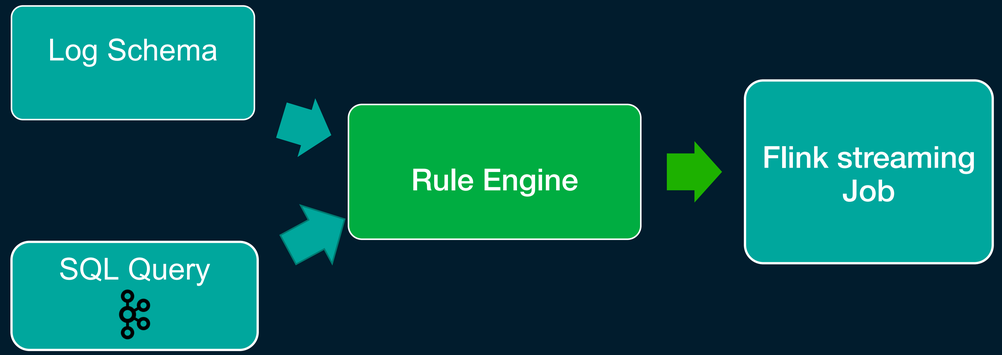
简单例子：过量点击者


```
SELECT uid, ...
FROM click_log
GROUP BY uid, HOP(`timestamp`, interval '5' minute,
interval '1' hour)
```

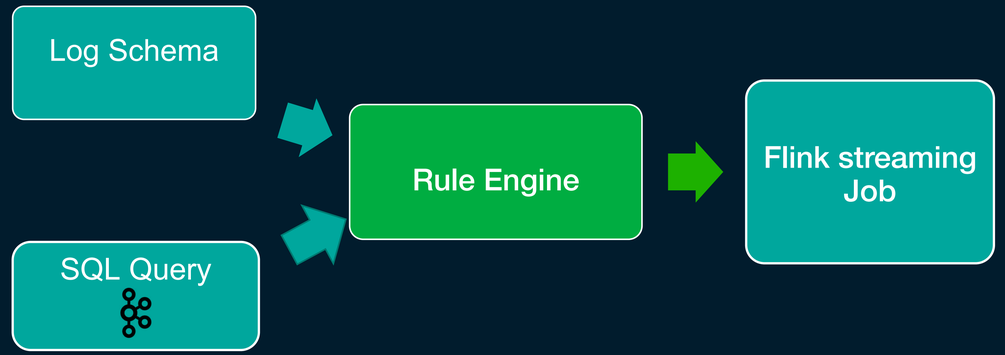
FLINK FORWARD

criteo

Rule Engine



规则引擎



criteo

Flink In Production

- Incremental checkpointing
- Exactly once guarantee
- High throughput
- Deployed on Yarn (plans for migrating to mesos)
- Savepoint before each new release
- Monitoring with grafana/kibana

生产环境中的Flink

- 递增 checkpointing
- 一次仅一次的投递保证
- 高吞吐
- 发布在yarn集群（计划迁移到 mesos ）
- 每一次发布新的版本之前存 savepoint
- 通过 grafana，kibana 实时监测

FLINK FORWARD

criteo

Happy User Most of the Time!

大多数时间都是工作的!

criteo

FLINK FORWARD

But, Things Can Go Wrong…

```
  SELECT uid, ...
- FROM click_log
+ FROM enriched_click_log
  GROUP BY uid, HOP(`timestamp`, interval '5' minute,
interval '1' hour)
```

criteo

有的时候会出一些问题

```
  SELECT uid, ...
- FROM click_log
+ FROM enriched_click_log
  GROUP BY uid, HOP(`timestamp`, interval '5' minute,
interval '1' hour)
```

criteo

What we observe after release

- Overblacklisting
- We are counting twice all the clicks !!!

一次新的发布之后

- 过量的过滤流量
- 每一次点击我们算了两次

What Happened

- With restoring from savepoint, all paritions of old topic are restored
- With flink topic/partition discovery, partitions from the new topic are discovered
- Flink is consuming from both topics

发生了什么

- 从 savepoint 恢复的时候，所有 kafka 就得分区被重新读取
- Flink 的 discovery 模块会自动发现新的分区，新的 topic 的分区被发现
- Flink 从两个 topic 的所有分区读取

FLINK FORWARD

criteo

What We Can Do

- Set new uid for the new Kafka Source
- Not really possible in Flink SQL since uid is not exposed
- Delete savepoint and restart…

怎么解决这个问题

- 给新的 Kafka Source 设置新的 uid （SQL 里边不是很可行，因为没有设置 uid 的借口）
- 删除 savepoint，重新启动 Flink…

criteo

Future Plans

- Migrate to Flink1.6
- HA mode with zookeeper

计划

- 升级到 Flink1.6
- 使用高可用性模式

FLINK FORWARD

criteo

Thanks

谢谢

criteo