

Processing billions of events in real time at Twitter

Twitter Events:

- 400 Billion
- PB level
- **Multi Sources** - Hadoop、Vertica、Manhattan 分布式数据库、Kafka、Twitter Eventbus、GCS、BigQuery 和 PubSub

现状

为了处理这些源和平台中的这些类型的数据，Twitter 数据平台团队已经构建了很多内部工具，

例如用于批处理的 Scalding，用于流的 Heron，用于批处理和实时处理的名为 TimeSeries Aggregator (TSAR) 的集成框架，以及用于数据发现和消费的 Data Access Layer。

然而，随着数据的快速增长，大规模仍然给工程师们用来运行管道的数据基础设施带来了挑战。

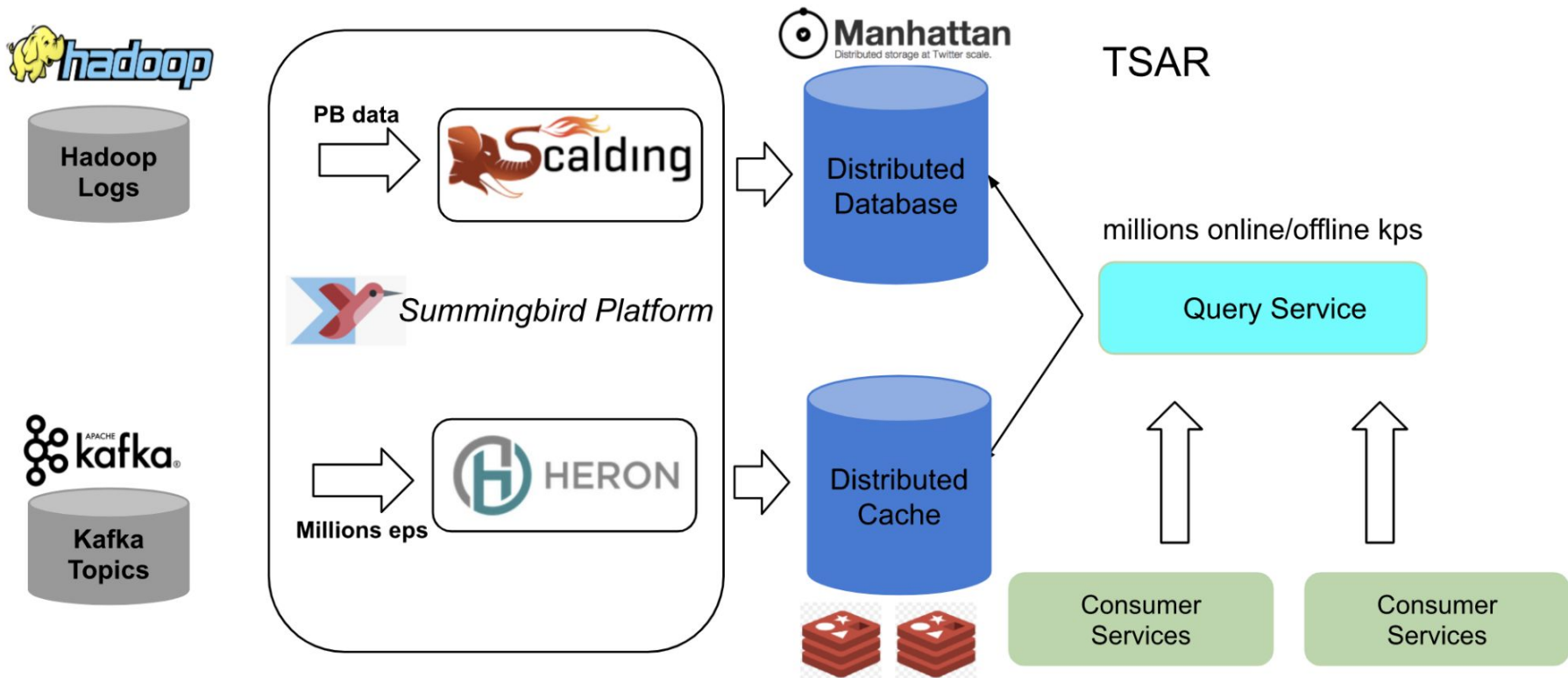
比如，有一个交互和参与的管道，能够以批处理和实时的方式处理大规模数据。由于数据规模的快速增长，对流延迟、数据处理的准确性和数据的实时性提出了更高的要求。

现状

对于交互和参与的管道，Twitter从各种实时流、服务器和客户端日志中采集并处理这些数据，从而提取到具有不同聚合级别(various levels of aggregations)、时间粒度(various time granularities)和其他度量维度(other metrics dimensions)的 Tweet 和用户交互数据。

这些聚合的交互数据尤其重要，并且是真正来自 Twitter 的广告收入服务和数据产品服务检索影响和参与度指标信息。此外，还需要保证对存储系统中的交互数据进行快速查询，并在不同的数据中心之间实现低延迟和高准确性。为了构建这样一个系统，Twitter把整个工作流分解为几个部分，包括预处理、事件聚合和数据服务。

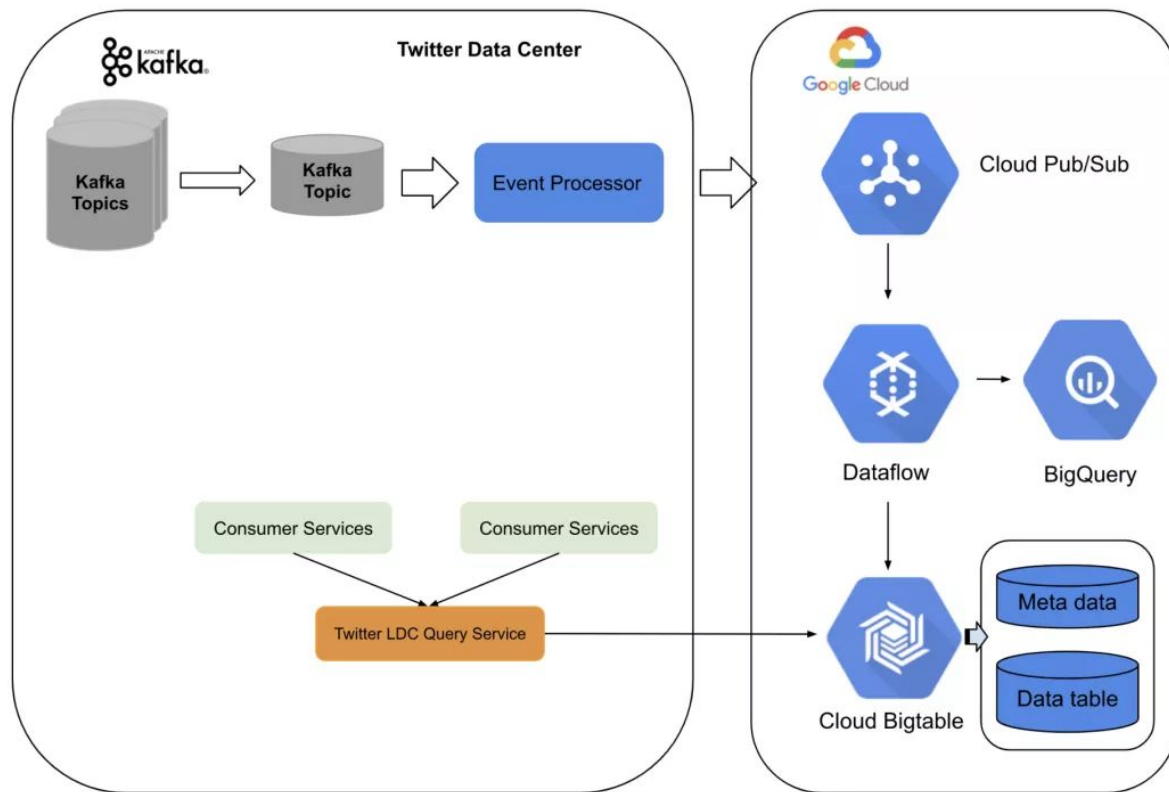
Old architecture



Pain Points

- 由于实时处理的数据规模大、吞吐量高，对于实时管道来说，可能会发生数据丢失、数据不准确的问题。
- 对于 Heron 拓扑结构，当发生更多的事件需要处理，Heron Bolt 无法不能及时处理时，拓扑结构内会产生背压 ([back pressure](#))。另外，由于垃圾收集成本很高，Heron Bolt 将会非常缓慢。当系统长期处于背压状态时，Heron Bolt 会积累喷口滞后 (spout lag)，这表明系统延迟很高。通常当这种情况发生时，需要很长的时间才能使拓扑滞后下降。更多的时候，也有很多 Heron 流管理器的“死亡”（流管理器管理拓扑组件之间的图元路由），而滞后不断上升。
- 当前的操作方案是重启 Heron 容器，将流管理器唤醒，以使 Bolt 能够重新启动处理流。这会在操作过程中造成事件丢失，从而导致 Nighthawk 存储中的聚合计数不准确。
- 对于批处理组件，Twitter 构建了几条 heavy computation pipelines，这些管道用于处理 PB 级数据，每小时运行一次，将数据汇入 Manhattan。
- 集中式 TSAR 查询服务整合了 Manhattan 和 Nighthawk 的数据，为客户服务提供数据服务。由于实时数据的潜在损失，TSAR 服务可能为客户提供较少的聚合指标。

New architecture - Kappa architecture



新旧架构比较

Metrics		New Architecture	Old Architecture	
		Real Time Pipelines on Kafka and Dataflow	Real Time Pipelines on Heron	Batch Pipelines on Hadoop
Processing	Events Processed	~ 4 million events/second	~ 4 million events/second	PB data scale daily
	Real Time Compute Throughput	~600MB/s - ~1GB/s	~30MB/s - ~100MB/s *	N/A
	Batch Compute Cost	N/A	N/A	~300,000,000 PB Millis
Latency		~10s	~10s - 10 min	~1 day
Aggregation Correctness	Events Loss When Restarting	No	Yes	No
	Late Events Handling	Yes	No	No

* Heron compressed data internally for streaming

References

- What is Lambda Architecture? <https://databricks.com/glossary/lambda-architecture>
- What is Kappa Architecture? <https://hazelcast.com/glossary/kappa-architecture/>
- https://blog.twitter.com/engineering/en_us/topics/infrastructure/2021/processing-billions-of-events-in-real-time-at-twitter-