Challenges in Stream Processing

Karthik Ramasamy
Co-founder and Chief Product Officer
Streamlio
Challenges in Stream Processing

- Market Challenges
  - Increasing the market share
  - Identifying business value

- Technology Challenges
  - Unified Streaming Platform
  - Simplifying Compute
  - Self Regulating Streaming Systems
  - Combining Batch and Streaming
Broaden Applicability of Streaming

- Monitoring
- Microservices
- Workflows
- Model Inference
- Analytics
Stream Processing Pattern

Data Ingestion -> Messaging -> Compute -> Storage -> Results Storage

Data Processing

Data Serving
Ingestion, Storage and Compute - Apache Pulsar

- Data visible as soon as written
- Fault tolerant/Replicated
- Support multiple access patterns
- Infinitely Expandable
- High throughput/Low Latency
- Reduced TCO
Separation of Storage and Serving in Apache Pulsar

**SERVING**
- Brokers can be added independently
- Traffic can be shifted quickly across brokers

**STORAGE**
- Bookies can be added independently
- New bookies will ramp up traffic quickly
Simplifying Compute

Often stitched together as large programs

Functional API - Concise, Compact but complicated

SQL - easy but addresses small subset
**Serverless Approach to Data Processing - Pulsar Functions**

- **ETL**

```java
import java.util.function.Function;
public class ExclamationFunction implements Function<String, String> {
    @Override
    public String apply(String input) {
        return input + "!";
    }
}
```

- **Reactive Services**

```java
import org.apache.pulsar.functions.api.Context;
import org.apache.pulsar.functions.api.PulsarFunction;
public class CounterFunction implements PulsarFunction<String, Void> {
    @Override
    public Void process(String input, Context context) throws Exception {
        for (String word : input.split("\.")) {
            context.incrCounter(word, 1);
        }
        return null;
    }
}
```

- **Classification**

- **Real-time Aggregation**

- **Event Routing**

- **Microservices**
Self Regulating Streaming Systems

Self tuning

Several tuning knobs
Time consuming tuning phase

The system should take as input an SLO and automatically configure the knobs.

Self stabilizing

Stream jobs are long running
Load variations are common

The system should react to external shocks and automatically reconfigure itself.

Self healing

System performance affected by hardware or software
delivering degraded quality of service

The system should identify internal faults and attempt to recover from them.
Combining Streaming and Batching

Stream Data Silo

Batch Data Silo
Combining Streaming and Batching
Readings

Anomaly Detection in *Strata San Jose’17*

Real-Time Data Streams Using Heron

Twitter Heron: Towards Extensible *ICDE’17*

Streaming Engines

Dhalion: Self-Regulating *VLDB’17*

Stream Processing in Heron

MillWheel: *VLDB’13*

Fault-Tolerant Stream Processing at Internet Scale

Apache Pulsar - [Website](#)

Messaging/Storage in Apache Pulsar

Twitter Heron: Stream *SIGMOD’15*

Processing at scale

The Dataflow Model: A Practical *VLDB’15*

Approach to Balancing Correctness, Latency and Cost in Massive-Scale, Unbounded Out-of-Order Data Processing