Real-time Air Quality Anomaly Detection using Apache Flink

Ruquaiya Shuaibu

Apache Flink

- Flink is a distributed dataflow engine for stateful stream processing
- Key innovations:
  - True streaming (event-at-a-time, not micro-batches)
  - Distributed snapshots for fault tolerance (Chandy-Lamport algorithm)
  - State management at scale
  - Low-latency processing (millisecond range)
- Used in production for real-time analytics, ETL pipelines, event-driven applications

Why this project

Interest in understanding distributed stream processing architectures

Key technical questions:

- How does Flink achieve fault tolerance with exactly-once semantics?
- How does state management work in a distributed setting?
- What are the trade-offs between latency, throughput, and consistency?

Technical Goals and Exploration

- Primary goal: Understand Flink's architecture through hands-on implementation
- Technical aspects to explore:
  - Dataflow model
  - Windowing mechanisms
  - State backends
  - Checkpointing
  - Parallelism & partitioning
- Concrete implementation: Anomaly detection pipeline as test workload
  - Requires stateful operations (windowing, aggregations)
  - Pushes Flink's state management capabilities
  - Measurable performance metrics

Technical Architecture

- Pipeline Architecture

```
Data Source → Flink Stream → Windowing → Stateful Processing → Sink
```

- Key components I’ll use:
  - DataStream API
  - Window operators (for temporal aggregations)
  - State API (for tracking historical patterns)
  - Checkpointing (for fault tolerance)

Benchmarking and Analysis Plan

- Performance metrics:
  - Throughput (events/second)
  - Latency (end-to-end processing time)
  - State size growth
  - Checkpoint overhead
- Configurations to compare:
  - Different parallelism levels
  - Different state backends (RocksDB vs heap)
  - Different checkpointing intervals
- Analysis: Understanding trade-offs in Flink's design decisions

Progress and Timeline

Progress so far:
Project selected and approved

Initial research on Flink architecture

Identified data sources

Remaining timeline:

Week 1: Set up Flink environment, begin data ingestion pipeline

Week 2:: Implement anomaly detection algorithms, basic visualization

Week 3: Benchmarking, performance analysis

Week 4: Complete testing, write report

Questions?