Cloud Computing

Project Milestone – 4

Name: Kanishk Ghai

Student id: 100815261

GitHub: https://github.com/Kan-G1/Cloud-computing-milestone4

Link to Recording :
https://drive.google.com/drive/folders/1DZ2deQaedbJSOqtJOBXsDiAXn1xQ49DV?usp=sharing

Discussion Question

Compare the advantages and disadvantages of using Dataflow vs microservices in preprocessing the smart reading.

Advantages of Dataflow

1. Unified streaming/batch model: Apache Beam handles both batch and real-time streaming in a single API, making it easier to switch modes without rewriting logic.

2. Built-in stateful processing: Windowing and aggregation (e.g., sum readings per hour) are natively supported without external storage.

3. Simpler ops: One job to deploy, monitor, and manage via the Dataflow UI.

4. Auto-scaling: Automatically adjusts the number of workers based on data volume with no manual configuration.

Disadvantages of Dataflow

1. Tight coupling: All logic is in one pipeline; changing one step requires redeploying the whole job.

2. Higher latency for complex pipelines: Batch windows introduce delays compared to pure event-driven processing.

3. Language limitation: Primarily Python/Java; SDK constraints on certain operations.

4. Cost: Dataflow workers are charged even when idle, whereas microservices on GKE can scale to zero.

Advantages of Microservices

1. Independent deployability: Each service (filter, convert, etc.) can be deployed, updated, and scaled independently without affecting others.

2. Technology flexibility: Each service can use different languages or libraries.

3. Event-driven reactivity: Each service triggers immediately when a relevant message arrives, enabling very low latency.

4. Resilience: Failure in one service doesn't crash the whole pipeline; Pub/Sub buffers messages.

5. Reusability: Individual services can be reused across different pipelines or applications.

Disadvantages of Microservices

1. Operational complexity: Managing many separate containers, deployments, and subscriptions requires more DevOps work.

2. State management overhead: Stateful operations (like deduplication in the voting logger) require external stores (Redis, DB), adding infrastructure.

3. Debugging difficulty: Distributed tracing across multiple services is harder than inspecting a single Dataflow job's DAG.

4. Network overhead: Every inter-service call goes through Pub/Sub, introducing message serialization costs.

5. At-least-once delivery: Pub/Sub may deliver duplicates; services must handle idempotency.