STREAM PROCESSING AT LINKEDIN: APACHE KAFKA & APACHE SAMZA

Processing billions of events every day
Neha Narkhede

- Co-founder and Head of Engineering @ Stealth Startup

- Prior to this...
  - Lead, Streams Infrastructure @ LinkedIn (Kafka & Samza)
  - One of the initial authors of Apache Kafka, committer and PMC member

- Reach out at @nehanarkhede
Agenda

- Real-time Data Integration
- Introduction to Logs & Apache Kafka
- Logs & Stream processing
- Apache Samza
- Stateful stream processing
The Data Needs Pyramid

Maslow's hierarchy of needs

- Physiological
- Safety
- Love/Belonging
- Esteem
- Self actualization

Data needs

- Data collection
- Data processing
- Understanding
- Automation
Real-time Data Integration
- Introduction to Logs & Apache Kafka
- Logs & Stream processing
- Apache Samza
- Stateful stream processing
Increase in diversity of data

- **1980+**
  - Database data (users, products, orders etc)

- **2000+**
  - Events (clicks, impressions, pageviews)
  - Application logs (errors, service calls)
  - Application metrics (CPU usage, requests/sec)

- **2010+**
  - IoT sensors

Siloed data feeds
Explosion in diversity of systems

- Live Systems
  - Voldemort
  - Espresso
  - GraphDB
  - Search
  - Samza

- Batch
  - Hadoop
  - Teradata
Data integration disaster
Centralized service

- Espresso
- Voldemort
- Oracle
- User Tracking
- Logs
- Operational Metrics
- Hadoop
- Log Search
- Monitoring
- Data Warehouse
- Social Graph
- Rec Engine & Life
- Search
- Security
- ... Email

Data Pipeline

Production Services
Agenda

- Real-time Data Integration
- Introduction to Logs & Apache Kafka
  - Logs & Stream processing
  - Apache Samza
  - Stateful stream processing
Kafka at 10,000 ft

- Distributed from ground up
- Persistent
- Multi-subscriber
Key design principles

- Scalability of a file system
  - Hundreds of MB/sec/server throughput
  - Many TBs per server

- Guarantees of a database
  - Messages strictly ordered
  - All data persistent

- Distributed by default
  - Replication model
  - Partitioning model
Kafka adoption
Apache Kafka @ LinkedIn

- 175 TB of in-flight log data per colo
- Low-latency: ~1.5ms
- Replicated to each datacenter
- Tens of thousands of data producers
- Thousands of consumers
- 7 million messages written/sec
- 35 million messages read/sec
- Hadoop integration
Logs

The data structure every systems engineer should know
The Log

- Ordered
- Append only
- Immutable

1st record
next record written

0 1 2 3 4 5 6 7 8 9 10 11 12
The Log: Partitioning

Partition 0

Partition 1

Partition 2
Logs: pub/sub done right

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Write to A</td>
</tr>
<tr>
<td>11</td>
<td>Write to B</td>
</tr>
<tr>
<td>7</td>
<td>Read from A</td>
</tr>
<tr>
<td>11</td>
<td>Read from B</td>
</tr>
</tbody>
</table>

Destination system A (time = 7)
Destination system B (time = 11)
Logs for data integration

User updates profile with new job

KAFKA

Newsfeed  Search  Hadoop  Standardization engine
Agenda

- Real-time Data Integration
- Introduction to Logs & Apache Kafka

**Logs & Stream processing**
- Apache Samza
- Stateful stream processing
Stream processing $= f(\text{log})$
Stream processing = f(log)
Apache Samza at LinkedIn

User updates profile with new job

KAFKA

Newsfeed  Search  Hadoop  Standardization engine
Latency spectrum of data systems

RPC

Asynchronous processing (seconds to minutes)

Synchronous (milliseconds)

Latency

Batch (Hours)
Agenda

- Real-time Data Integration
- Introduction to Logs & Apache Kafka
- Logs & Stream processing
- Apache Samza
- Stateful stream processing
public interface StreamTask {
    void process (IncomingMessageEnvelope envelope, MessageCollector collector, TaskCoordinator coordinator);
}

getKey(), getMsg()  

sendMsg(topic, key, value)  

commit(), shutdown()
Samza Architecture (Logical view)
Samza Architecture (Logical view)
Samza Architecture (Physical view)

Host 1

Samza container 1

Samza container 2

Host 2
Samza Architecture (Physical view)

Host 1

Node manager

Samza container 1

Host 2

Node manager

Samza container 2

Samza YARN AM
Samza Architecture (Physical view)

Node manager
Samza container 1
Kafka
Host 1

Node manager
Samza container 2
Kafka
Host 2

Samza YARN AM

Host 1

Host 2
Samza Architecture: Equivalence to Map Reduce

Node manager

Map Reduce

HDFS

Host 1

Node manager

Map Reduce

YARN AM

HDFS

Host 2
### M/R Operation Primitives

- **Filter**: records matching some condition
- **Map**: record = \( f(\text{record}) \)
- **Join**: Two/more datasets by key
- **Group**: records with same key
- **Aggregate**: \( f(\text{records within the same group}) \)
- **Pipe**: job 1’s output \( \Rightarrow \) job 2’s input
M/R Operation Primitives on streams

- **Filter**: records matching some condition
- **Map**: $\text{record} = f(\text{record})$
- **Join**: Two/more datasets by key
- **Group**: records with same key
- **Aggregate**: $f(\text{records within the same group})$
- **Pipe**: job 1’s output $\Rightarrow$ job 2’s input

Requires state maintenance
Agenda

- Real-time Data Integration
- Introduction to Logs & Apache Kafka
- Logs & Stream processing
- Apache Samza

- **Stateful stream processing**
Example: Newsfeed

User ... posted "...
User 989 posted "Blah Blah"
User 567 posted "Hello World"

Status update log

Fan out messages to followers

External connection DB

567 -> [123, 679, 789, ...]
999 -> [156, 343, ...]

Push notification log

Refresh user 123's newsfeed
Refresh user 679's newsfeed
Refresh user ...'s newsfeed
Local state vs Remote state: Remote

100–500K msg/sec/node

Samza task partition 0

100–500K msg/sec/node

Samza task partition 1

Remote state

Disk

❌ Performance

❌ Isolation

❌ Limited APIs

1–5K queries/sec ??
ex: Cassandra, MongoDB, etc
Local state: Bring data closer to computation
Local state: Bring data closer to computation
Example Revisited: Newsfeed

User 567 posted "Hello World"
User 989 posted "Blah Blah"
User 567 posted "Hello World"

Status update log

New connection log

User ... followed ...
User 123 followed 567
User 890 followed 234

User ... followed ...
User 989 posted "Blah Blah"
User ... posted "...

Fan out messages to followers

Push notification log

567 -> [123, 679, 789, ...]
999 -> [156, 343, ...]

Refresh user 123's newsfeed
Refresh user 679's newsfeed
Refresh user ...'s newsfeed
Fault tolerance?

Node manager

Samza container 1

Kafka

Host 1

Node manager

Samza container 2

YARN AM

Samza

Kafka

Host 2
Fault tolerance in Samza
Slow jobs

- Drop data
- Backpressure
- Queue
  - In memory
- On disk (KAFKA)
Real time data integration is crucial for the success and adoption of stream processing.

Logs form the basis for real time data integration.

Stream processing = f(logs)

Samza is designed from ground-up for scalability and provides fault-tolerant, persistent state.
Thank you!

- The Log

- Apache Kafka
  - http://kafka.apache.org

- Apache Samza
  - http://samza.incubator.apache.org

- Me
  - @nehanarkhede
  - http://www.linkedin.com/in/nehanarkhede