Distributed Research on Emerging Applications & Machines
Department of Computational & Data Sciences
Indian Institute of Science, Bangalore

DREAM:Lab

Introduction To Distributed Stream Processing

Shilpa Chaturvedi
M. Sc. (Research)
Computational and Data Sciences

18-Jan-17

4 V's Of Big Data

- Volume

  Terabytes and petabytes of data

- Velocity

  Millions of data per second

- Variety

  Different forms -images, bits, text

- Veracity

Streams are common

- Web & Social Networks
  - Twitter, Facebook, Internet packets
- Cybersecurity
  - Telecom call logs, financial transactions, Malware
- Internet of Things
  - Smart Transport/Power/Water networks

Models Of Computation

- Batch Compuatation

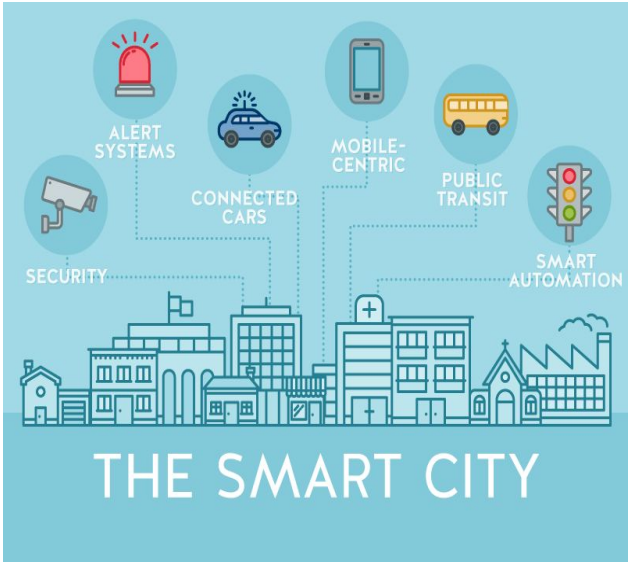
Processing archived data

- Real Time Compuatation

Processing incoming streaming data

Motivation: Internet of Things(IoT)

- *Data Source:* **Billions of sensors** monitoring physical infrastructure, human beings and virtual entities in realtime, distributed across network
- Data streams drive **realtime processing, analytics & decisions making**



http://1.bp.blogspot.com/-N5JfgIr1DaE/Vbpn0iIwYXI/AAAAAAAAjws/D1_DeUmxf8c/s1600/Smart-Cities-Features.png

Streaming Applications

Some **use cases** of streaming applications

- Realtime traffic information processing for navigation applications
- Water quality and power management in the Smart Campus
- Trending topics in social media like Twitter
- Network Traffic Management
- Fraud Detection in Transaction

Key Requirements

- Streams Of Incoming Data
- Near Real Time Processing
- Minimum Latency
- Scalable
- In memory Processing

Degree Of Parallelism

- Task Parallelism

Multiple Copies of Tasks Executing

- Data Parallelism

Splitting the data into chunks

Example - Word Count

Custom Solution

Say you are given a task to monitor environment status from sensor data in some geographical region. What will be the software components you need ?

Given Rate : 6K data per second

Tasks - Parsing -> Fetching Locations -> Grouping-> Average

Custom Solution

- Queues
- Polling Queue Logic
- Actual Code Logic
- Logic handle data movement
- Logic For Handling Parallelism

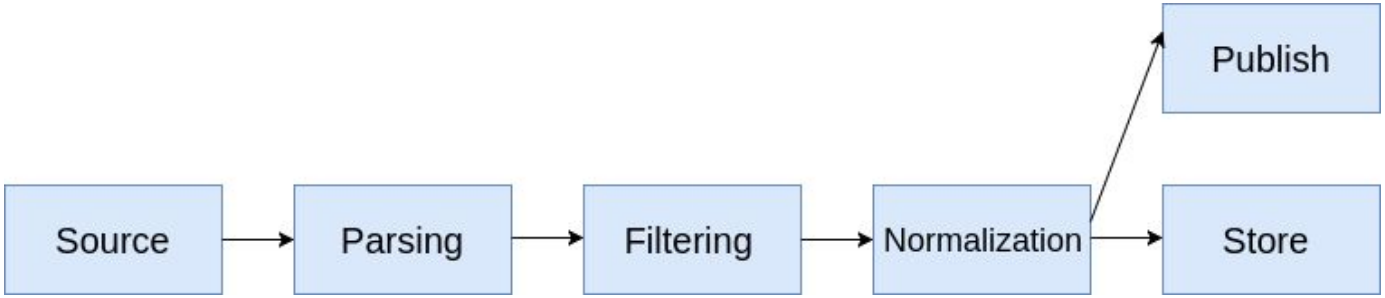
Distributed Stream Processing System [DSPS]

- DSPS are **Big Data** platforms tailored for *scalable processing* of *streaming data*, with *minimum latency*
- **Data Streams**: *Unbounded* sequence of messages
- **Message** contents are *opaque* to the system.
- **Tasks** are arbitrary code logic that operates on one message at a time
- Applications composed as **Directed Acyclic Graph** (DAG)
  - Tasks as vertices, Data Streams as edges

Stream Processing ➲ Continuous Dataflows

- How do you "compose" analytics that run continuously over streaming data?
- Application defined as Directed Acyclic Graph (DAG)
  - Vertices are *Tasks*
  - Edges are *streams*
  - Streams carry *tuples/events (name:value)* or *messages (opaque)*
  - Tasks process one or more messages/tuples and generate zero or more messages/tuples
  - Message routing?

An Example Pipeline

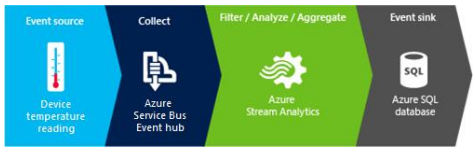


Applications are also called Pipelines or Topologies

Each Task may have more than one instances

Distributed Stream Processing Systems

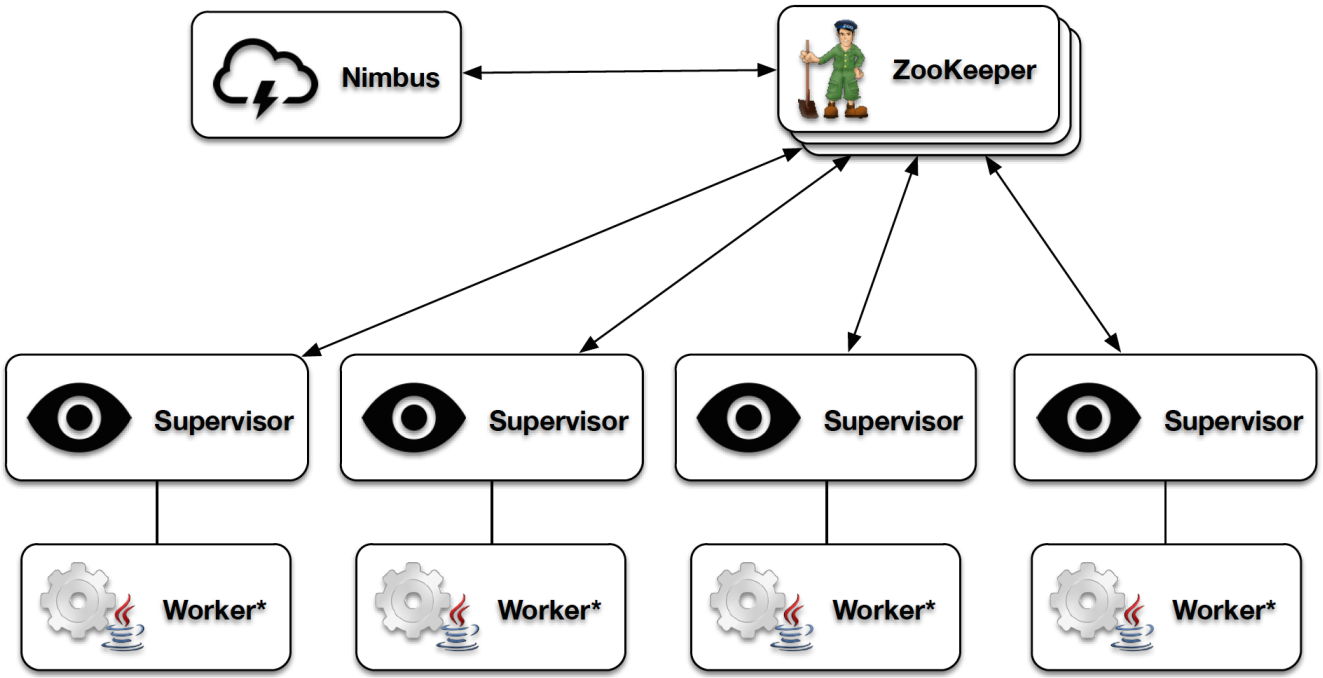
- Aurora – Early Research System
- Borealis – Early Research System
- **Apache Storm**
- Apache S4
- Apache Samza
- Google MillWheel
- Amazon Kinesis
- LinkedIn Databus
- Facebook Puma/Ptail/Scribe/ODS
- Azure Stream Analytics



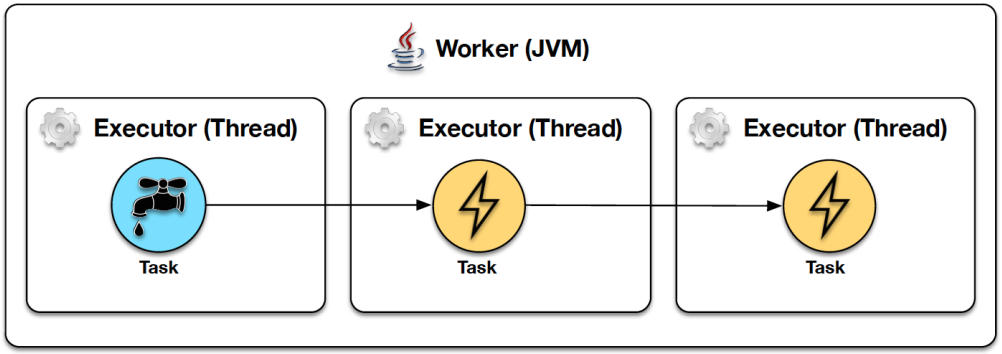
Apache Storm

- Popular DSPS
- Created by Nathan Marz at Twitter
- Applications are called Topologies
- Tasks are called Bolts
- Spouts are special task that act as source of data

Storm Cluster View

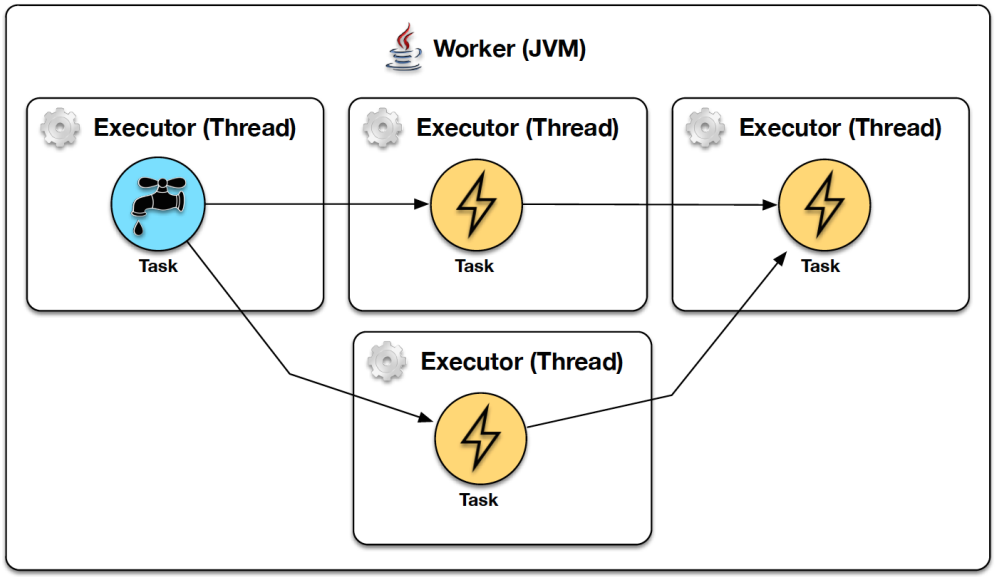


Parallelism



1 Worker,
Parallelism = 1

Parallelism



1 Worker,
Parallelism = 2

Weak Scaling

- Given a message rate and message rate supported by a task on single thread on one machine
- What if message rate increases ?
- Queuing → Latency
- Can we increase number of threads / increase machines

Reading

- Ankit Toshniwal, et al. Storm@twitter. In *ACM SIGMOD,* 2014
- Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters, Zaharia, et al, *USENIX HotCloud*, 2012, https://www.usenix.org/conference/hotcloud12/workshop-program/presentation/zaharia
- Leonardo Neumeyer, et al, S4: Distributed Stream Computing Platform. In *ICDMW* 2010