

BIG DATA PRODUCTS AND PRACTICES

Venkatesh Vinayakarao

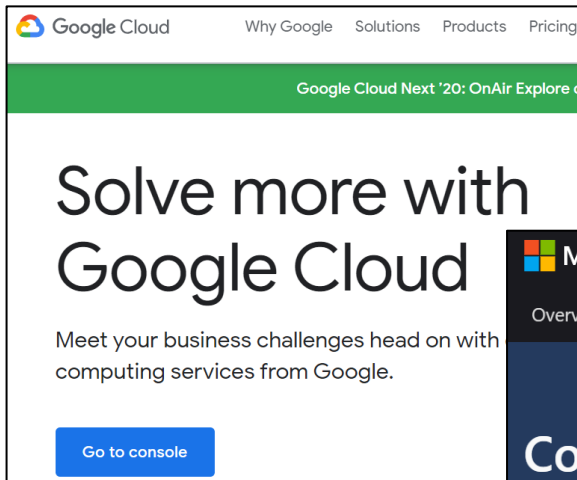
venkateshv@cmi.ac.in

<http://vvtesh.co.in>

Cloud Platforms

Cloud Services:

- SaaS
- PaaS
- IaaS



Google Cloud

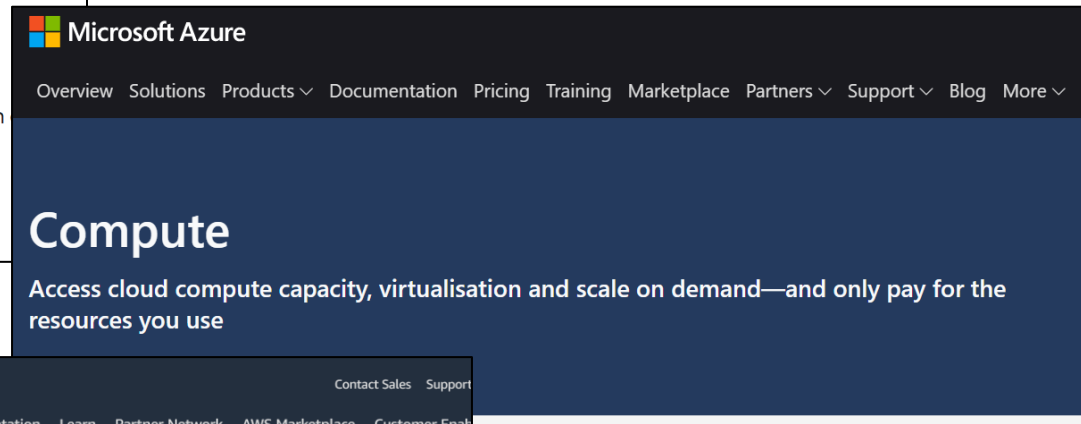
Why Google Solutions Products Pricing

Google Cloud Next '20: OnAir Explore c

Solve more with Google Cloud

Meet your business challenges head on with computing services from Google.

[Go to console](#)

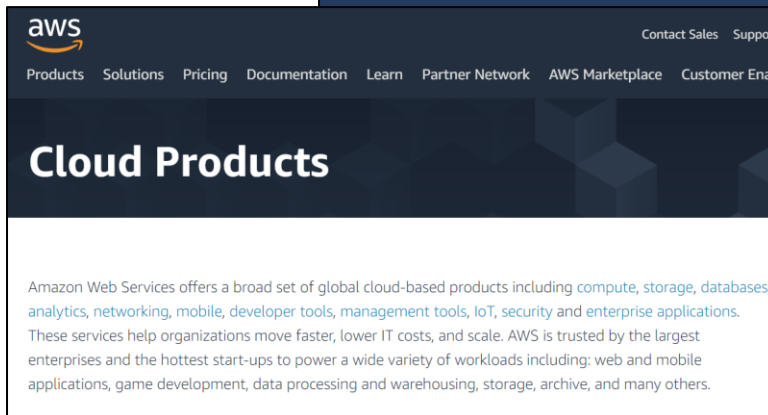


Microsoft Azure

Overview Solutions Products Documentation Pricing Training Marketplace Partners Support Blog More

Compute

Access cloud compute capacity, virtualisation and scale on demand—and only pay for the resources you use



aws

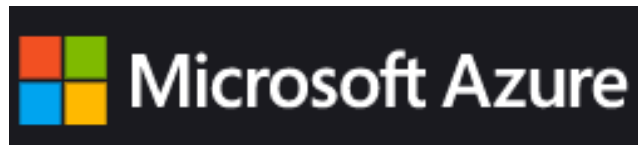
Contact Sales Support

Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enab

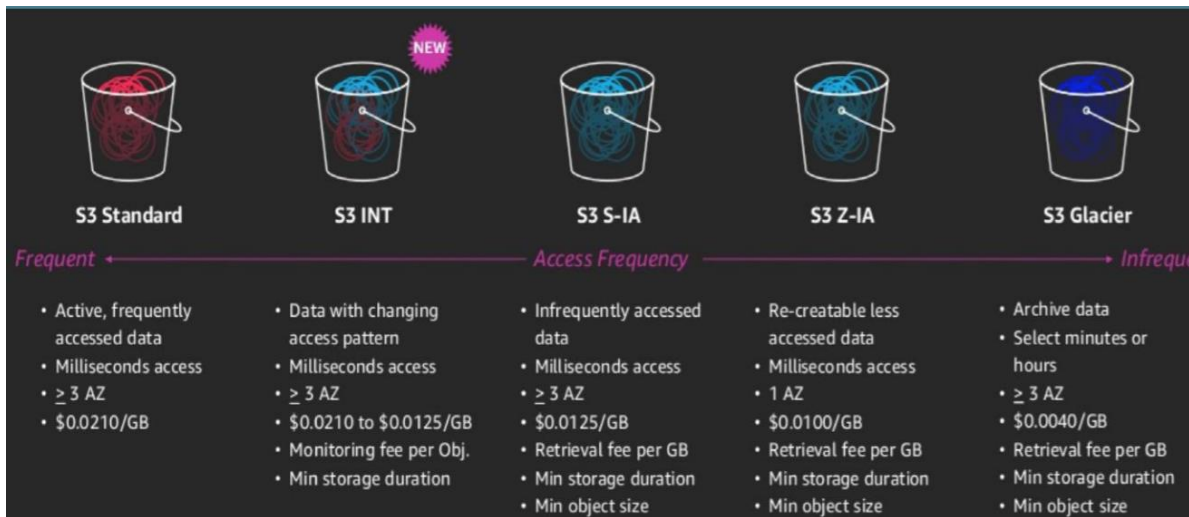
Cloud Products

Amazon Web Services offers a broad set of global cloud-based products including [compute](#), [storage](#), [databases](#), [analytics](#), [networking](#), [mobile](#), [developer tools](#), [management tools](#), [IoT](#), [security](#) and [enterprise applications](#). These services help organizations move faster, lower IT costs, and scale. AWS is trusted by the largest enterprises and the hottest start-ups to power a wide variety of workloads including: web and mobile applications, game development, data processing and warehousing, storage, archive, and many others.

Cloud Platforms



Storage Service (Amazon S3 Example)



Compute Services (Google Cloud Example)



Network Services (Azure Example)

- Azure Traffic Manager is a DNS-based traffic load balancer that distributes traffic optimally to services across global Azure regions, while providing high availability.
- Traffic Manager directs client requests to the most appropriate service endpoint.

Building Great Apps/Services

- Popular Use Cases
 - Visualization
 - Crawling/Search
 - Log Aggregation
 - Synchronization
 - Stream Processing

Tableau

A

B

Left pane- Displays the connected data source and other details about your data.

Canvas- Displays information about how the data source is set up and options for combining the data.

Data grid- Displays the fields in your data source as rows.

Shipping Dashboard

Running Total Shipping Costs

Shipping Cost

Avg Shipping Cost by Country

Tableau - Building a Dashboard

File Data Worksheet Dashboard Story Analysis Map Format Server Window Help

Dashboard - Layout

Device Preview

Size Automatic

Sheets

- Avg Shipping Cost by...
- Shipping Cost
- Product Orders
- Running Total Shippi...

Objects

- Horizontal
- Image
- Vertical
- Web Page
- Text
- Blank

Tiled Floating

Show dashboard title

Priority

- Critical
- High
- Medium
- Low

Market (All)

Running Sum of Shipping Co...

Month of Ship Date

Month of Ship Date

Month of Ship Date

Month of Ship Date

Month of Ship Date

Same Day

First Class

Second Class

Standard Class

600K

400K

200K

0K

2013

2015

2013

2015

2013

2015

2013

2015

2013

2015

0

10

20

30

40

50

60

Avg. Shipping Cost

© OpenStreetMap contributors

Data Source

Avg Shipping Cost by Country

Shipping Cost

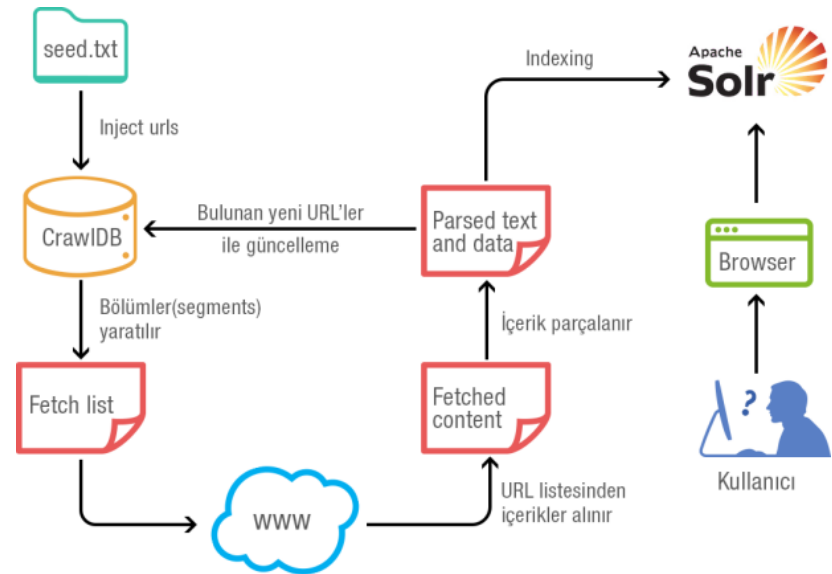
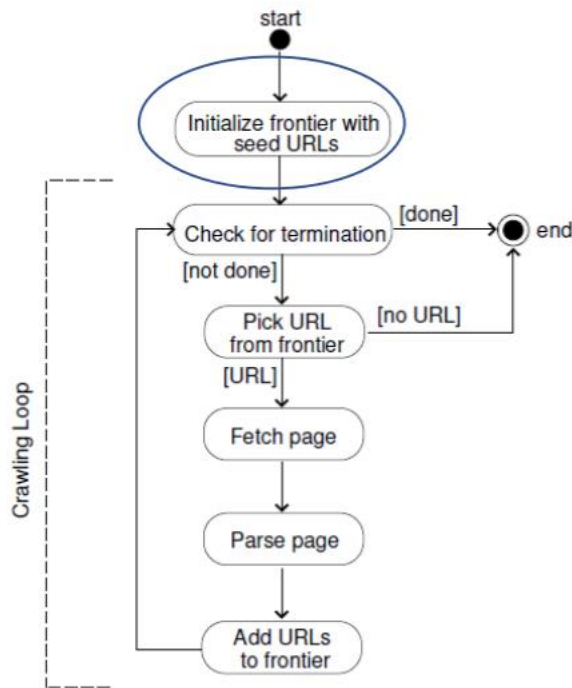
Product Orders

Running Total Shipping Costs

Dashboard 1

165 marks 1 row by 1 column SUM of AVG(Shipping Cost): 4,483.1

Crawling with Nutch



Solr Integration Image Src: <https://suyashaoc.wordpress.com/2016/12/04/nutch-2-3-1-hbase-0-98-8-hadoop-2-5-2-solr-4-1-web-crawling-and-indexing/>

Log Files are an Important Source of Big Data



Log4j

log4j.properties Syntax

Following is the syntax of *log4j.properties* file for an appender X:

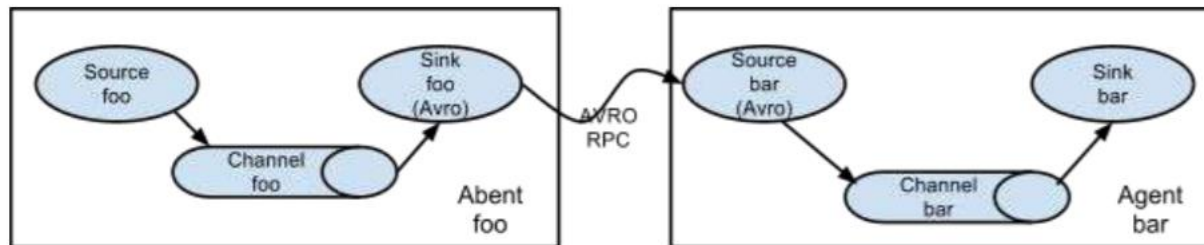
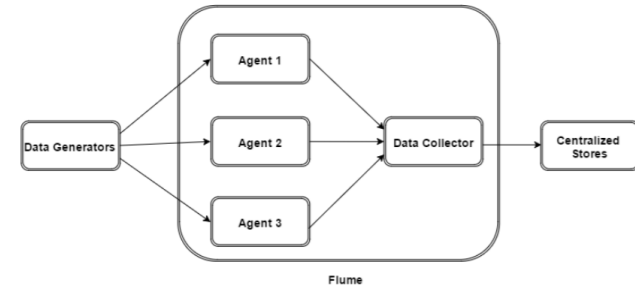
```
# Define the root logger with appender X
log4j.rootLogger = DEBUG, X

# Set the appender named X to be a File appender
log4j.appender.X=org.apache.log4j.FileAppender

# Define the layout for X appender
log4j.appender.X.layout=org.apache.log4j.PatternLayout
log4j.appender.X.layout.conversionPattern=%m%n
```

```
1 package test;
2
3 import org.apache.log4j.Logger;
4
5 public class Log4JTest {
6     static Logger log = Logger.getLogger(Log4JTest.class);
7
8     public static void main(String[] args) {
9
10         log.debug("This is a debug message");
11     }
12 }
13
```

Flume



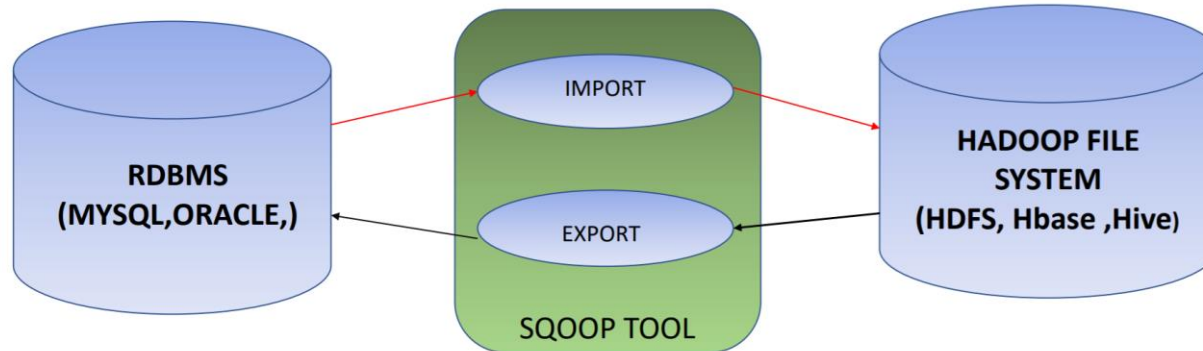
Flume event

```
usingFlume.sources = usingFlumeSource
usingFlume.channels = memory

usingFlume.sources.usingFlumeSource.type = avro
usingFlume.sources.usingFlumeSource.channels = memory
usingFlume.sources.usingFlumeSource.port = 7877
usingFlume.sources.usingFlumeSource.bind = 0.0.0.0
```

Flume Config Files

Sqoop



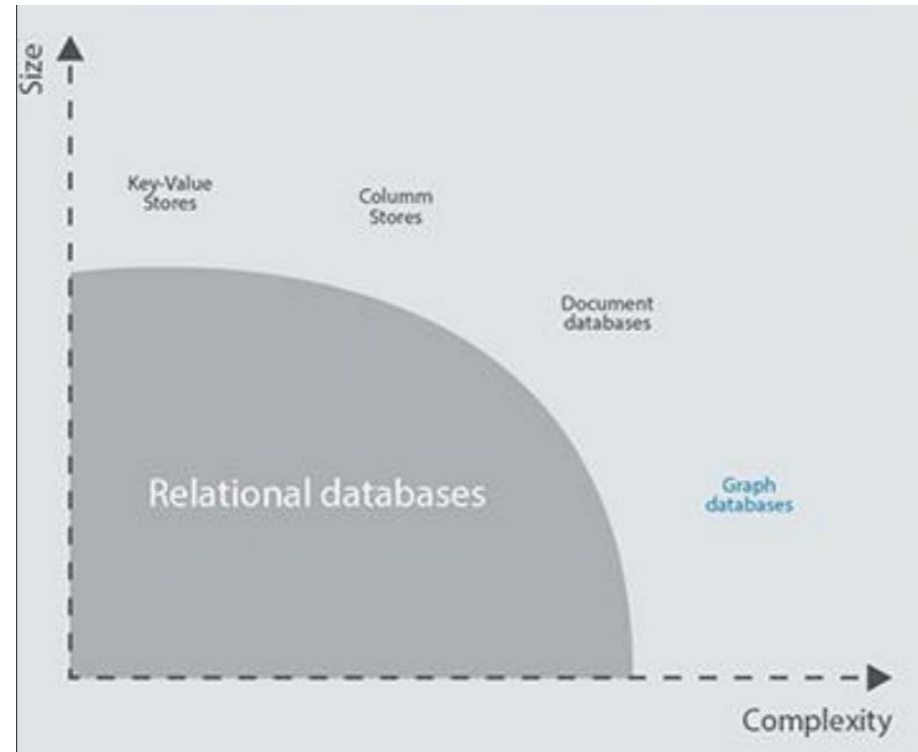
Designed for efficiently transferring bulk data between RDBMS and Hadoop



GraphDB – Neo4j



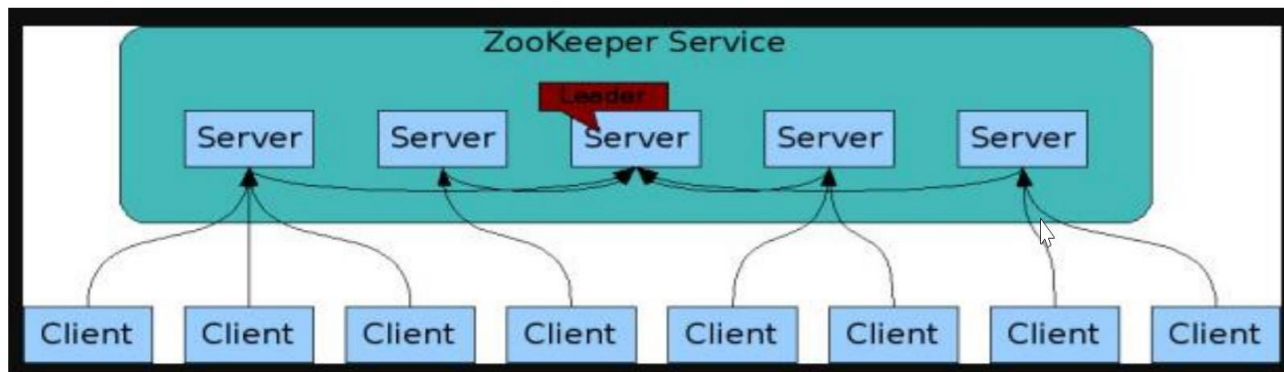
ACID compliant graph database management system



neo4j	Cypher	Most famous graph database, Cypher O(1) access using fixed-size array
DSE Graph DATASTAX AURELIUS	Gremlin	Distributed graph system based on Cassandra
ArangoDB	AQL	Multi-model database (Document + Graph)
OrientDB	OQL	Multi-model database (Document + Graph)

Synchronization with Apache ZooKeeper

A centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services

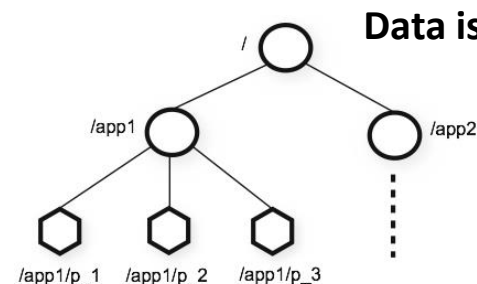


A Zookeeper Ensemble Serving Clients

It is simple to store data using zookeeper

```
$ create /zk_test my_data  
$ set /zk_test junk  
$ get /zk_test  
junk  
$ delete /zk_test
```

Data is stored hierarchically



Stream Processing

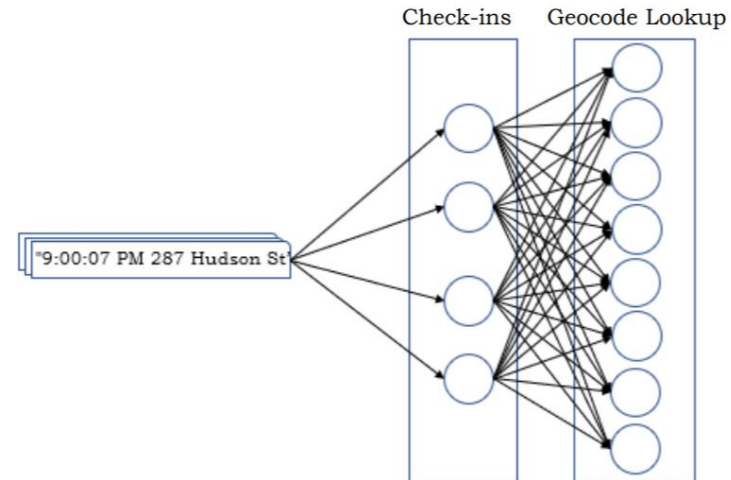
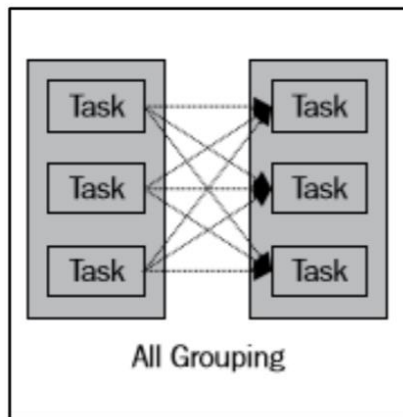
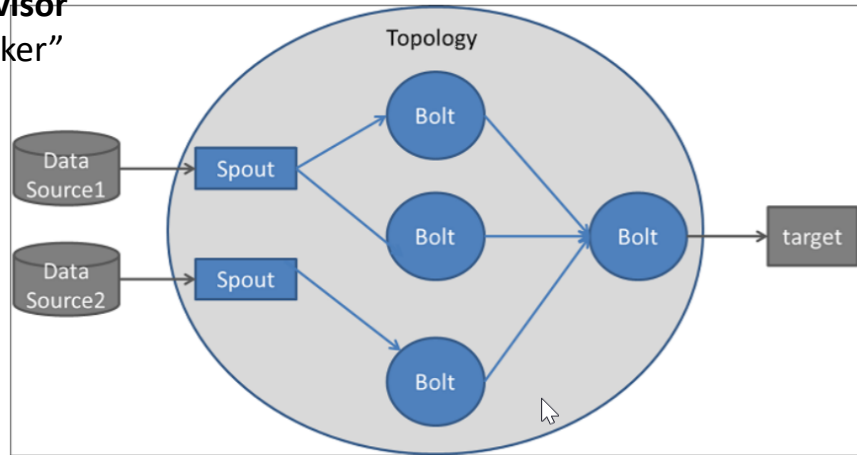
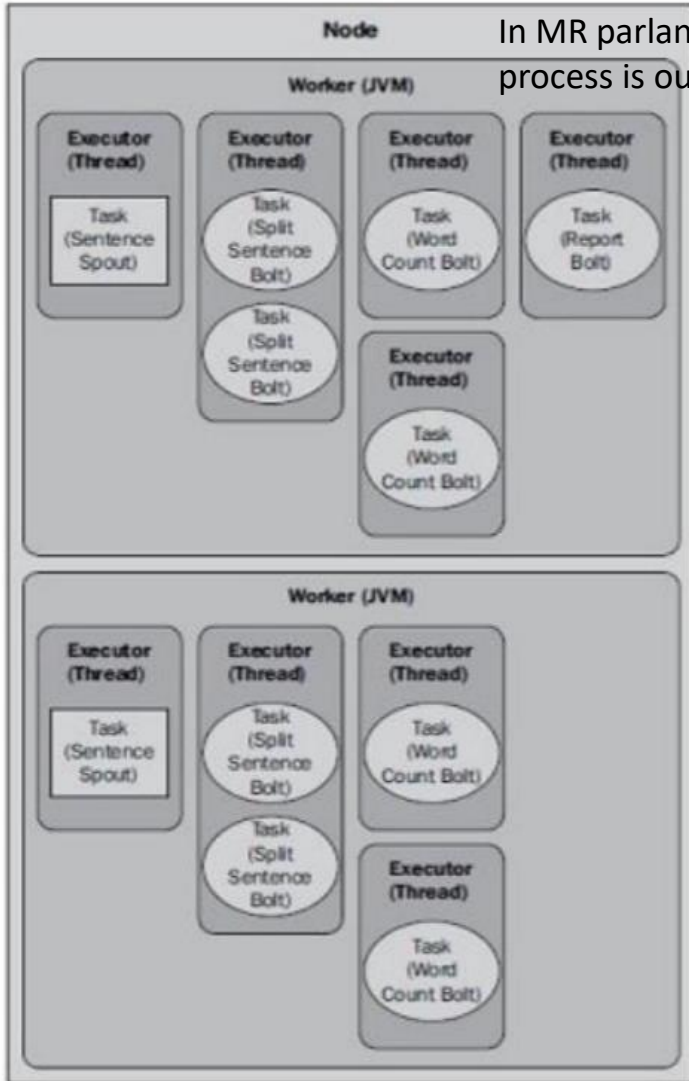
- Process data as they arrive.



Stream Processing with Storm

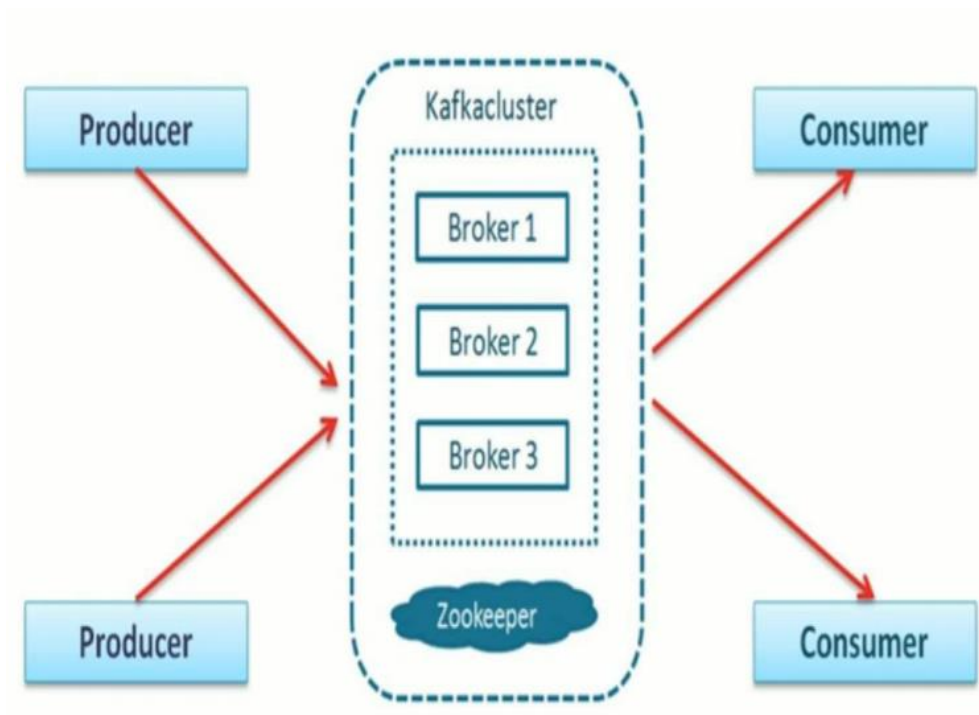
One of these is a master node. “Nimbus” is the “job tracker”!

In MR parlance, “Supervisor” process is our “task tracker”



Apache Kafka

- Uses Publish-Subscribe Mechanism

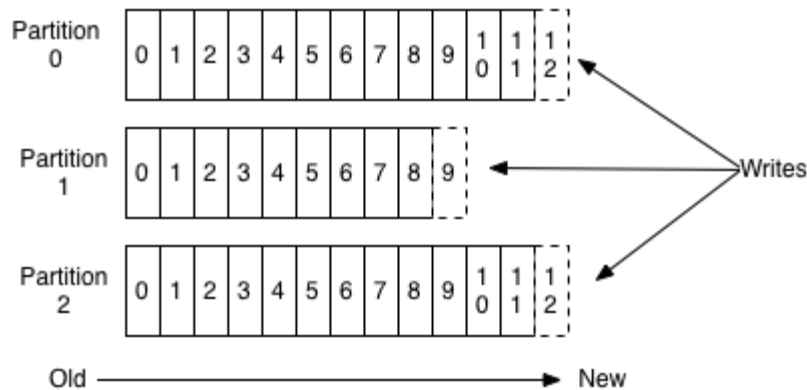


Kafka – Tutorial (Single Node)

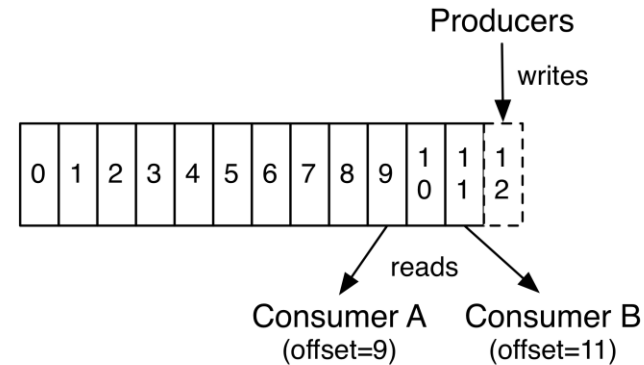
- Create a topic
 - `> bin/kafka-topics.sh ---topic test`
- List all topics
 - `> bin/kafka-topics.sh -list`
 - `> test`
- Send messages
 - `> bin/kafka-console-producer.sh --topic test`
 - This is a message
 - This is another message
- Receive messages (subscribed to a topic)
 - `> bin/kafka-console-consumer.sh --topic test --from-beginning`
 - This is a message
 - This is another message

Kafka – Multi-node

Anatomy of a Topic



- topic is a stream of records.
- for each topic, the Kafka cluster maintains a partitioned log
- records in the partitions are each assigned a sequential id number called the *offset*



Kafka Brokers

- For Kafka, a single broker is just a cluster of size one.
- We can setup multiple brokers
 - The broker.id property is the unique and permanent name of each node in the cluster.
 - > bin/kafka-server-start.sh config/server-1.properties &
 - > bin/kafka-server-start.sh config/server-2.properties &
 - Now we can create topics with replication factor
 - > bin/kafka-topics.sh --create --replication-factor 3 --partitions 1 --topic my-replicated-topic
 - > bin/kafka-topics.sh --describe --bootstrap-server localhost:9092 --topic my-replicated-topic
 - Topic: my-replicated-topic PartitionCount:1 ReplicationFactor:3
 - Partition: 0 Leader: 2 Replicas: 1,2,0

Streams API

```
StreamsBuilder builder = new StreamsBuilder();
KStream<String, String> textLines = builder.stream("TextLinesTopic");
KTable<String, Long> wordCounts = textLines
    .flatMapValues(textLine -> Arrays.asList(textLine.toLowerCase().split("\\W+")))
    .groupByKey((key, word) -> word)
    .count(Materialized.<String, Long, KeyValueStore<Bytes, byte[]>>as("counts-store"));
wordCounts.toStream().to("WordsWithCountsTopic", Produced.with(Serdes.String(), Serdes.Long()));
```

Apache Kinesis

- Amazon Kinesis Data Streams is a **managed service** that scales elastically for real-time processing of streaming big data.

“Netflix uses Amazon Kinesis to monitor the communications between all of its applications so it can detect and fix issues quickly, ensuring high service uptime and availability to its customers.” – Amazon

(<https://aws.amazon.com/kinesis/>).

Create Kinesis stream

Kinesis stream name*

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Shards

A shard is a unit of throughput capacity. Each shard ingests up to 1MB/sec and 1000 records/sec, and emits up to 2MB/sec. To accommodate for higher or lower throughput, the number of shards can be modified after the Kinesis stream is created using the API. [Learn more](#)



▶ Estimate the number of shards you'll need

Number of shards*

You can provision up to 199 more shards before hitting your account limit of 200. [Learn more or request a shard limit increase for this account](#)

Total stream capacity Values are calculated based on the number of shards entered above.

Write MB per second

Records per second

Read MB per second

* Required

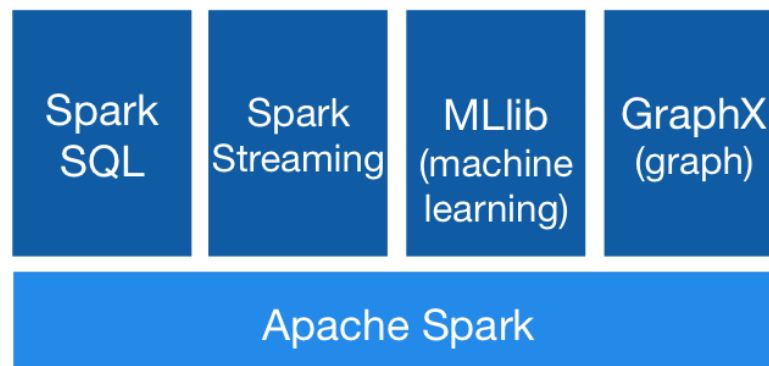
Cancel

Create Kinesis stream

Amazon Kinesis capabilities

- Video Streams
- Data Streams
- Firehose
- Analytics

Apache Spark (A Unified Library)



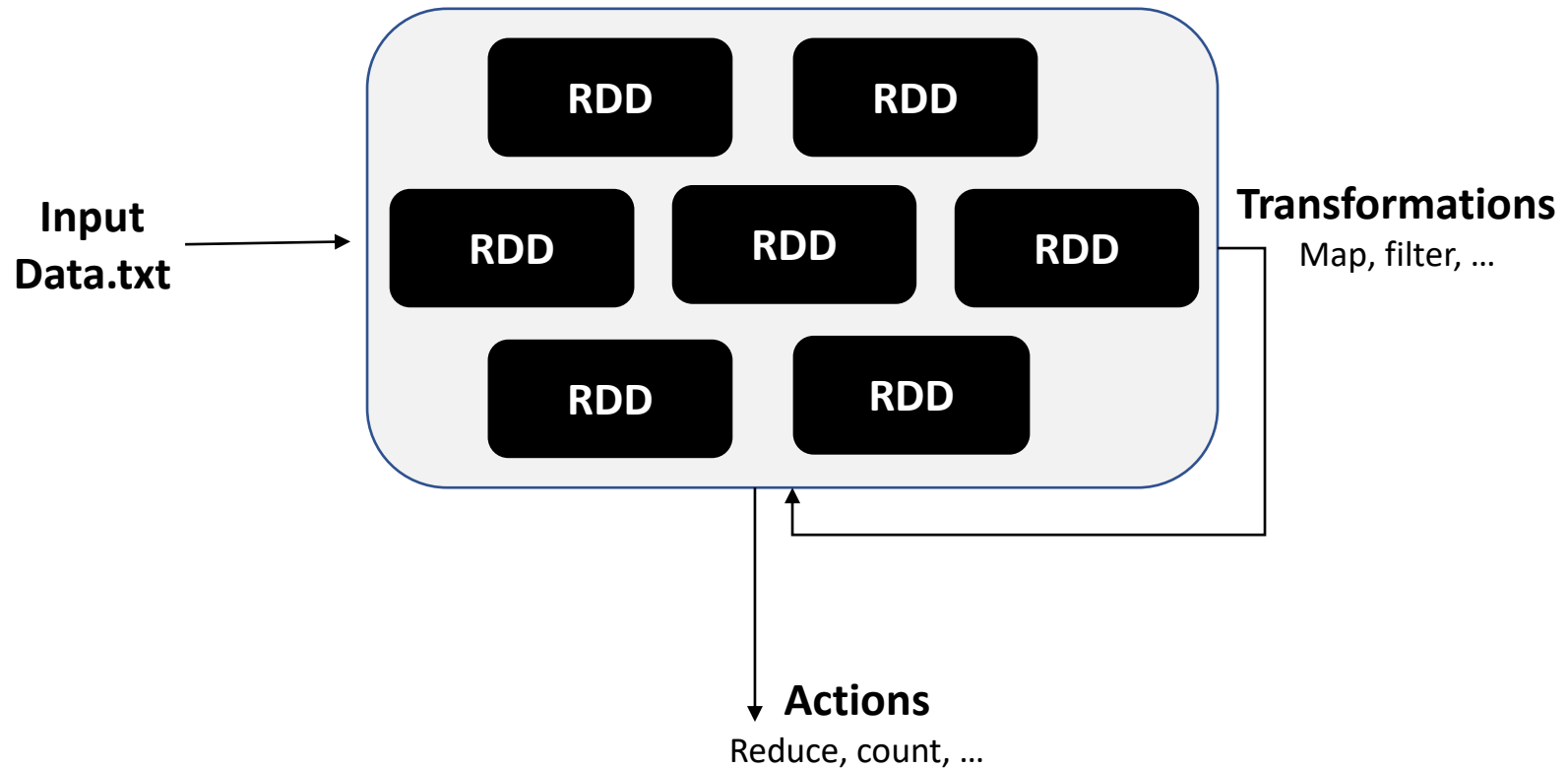
```
df = spark.read.json("logs.json")
df.where("age > 21")
  .select("name.first").show()
```

In spark, use data frames as tables

Spark's Python DataFrame API
Read JSON files with automatic schema inference

<https://spark.apache.org/>

Resilient Distributed Datasets (RDDs)



Spark Examples

```
data = [1, 2, 3, 4, 5]
distData = sc.parallelize(data)
```

distributed dataset can
be used in parallel

```
distFile = sc.textFile("dta.txt")
distFile.map(s => s.length).
  reduce((a, b) => a + b)
```

Map/reduce

```
"""MyScript.py"""
if __name__ == "__main__":
    def myFunc(s):
        words = s.split(" ")
        return len(words)

    sc = SparkContext(...)
    sc.textFile("file.txt").map(myFunc)
```

passing functions
through spark

Thank You