Popular scaling approaches

BITS Pilani
Pilani Campus

Akanksha Bharadwaj
CSIS Department

innovate achieve lead

BITS Pilani
Pilani Campus

SE ZG583, Scalable Services
Lecture No. 2

Partitioning and Sharding

Introduction

- In many large-scale solutions, data is divided into *partitions* that can be managed and accessed separately.

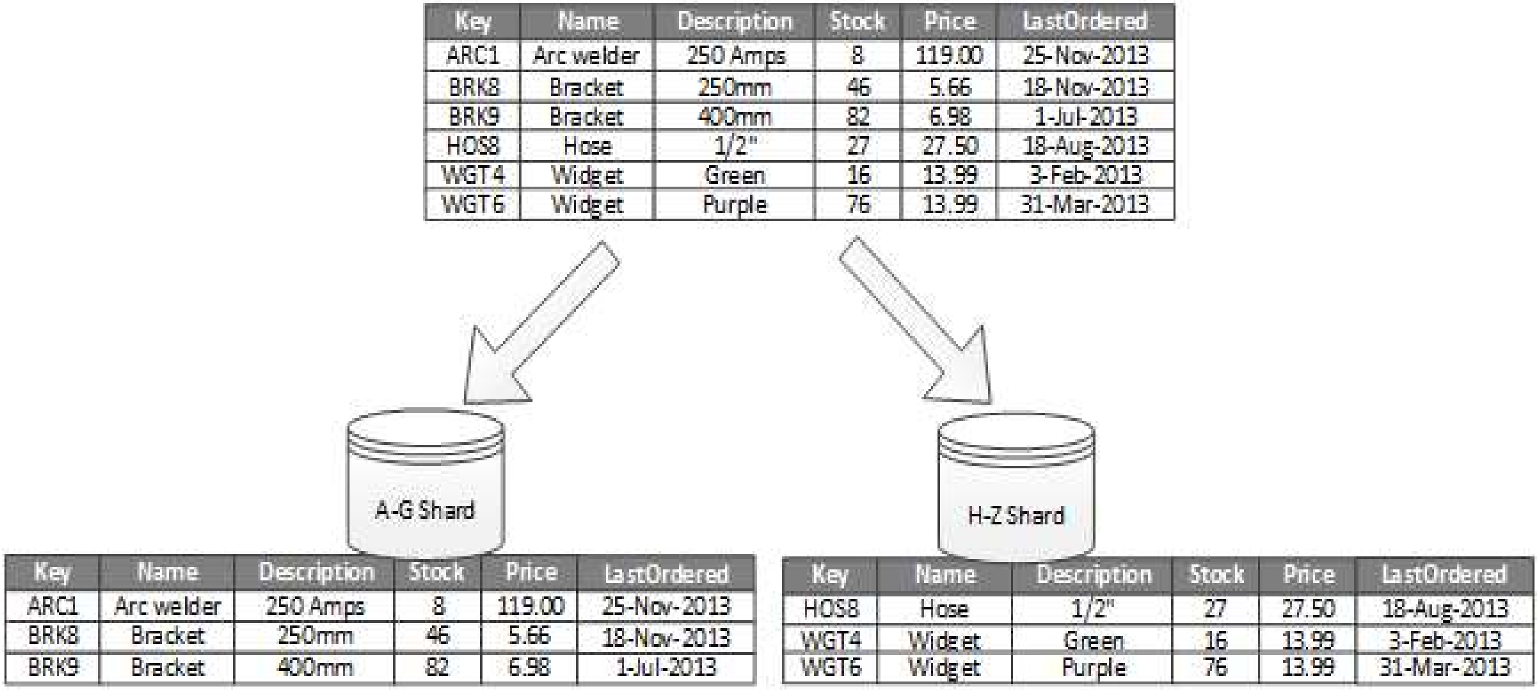
Why partition data?

- Improve scalability
- Improve performance
- Improve security
- Provide operational flexibility
- Improve availability

Types of Partitioning

- Horizontal partitioning
- Vertical partitioning
- Functional partitioning

Horizontal partitioning (Sharding)



Key	Name	Description	Stock	Price	LastOrdered
ARC1	Arc welder	250 Amps	8	119.00	25-Nov-2013
BRK8	Bracket	250mm	46	5.66	18-Nov-2013
BRK9	Bracket	400mm	82	6.98	1-Jul-2013
HOS8	Hose	1/2"	27	27.50	18-Aug-2013
WGT4	Widget	Green	16	13.99	3-Feb-2013
WGT6	Widget	Purple	76	13.99	31-Mar-2013

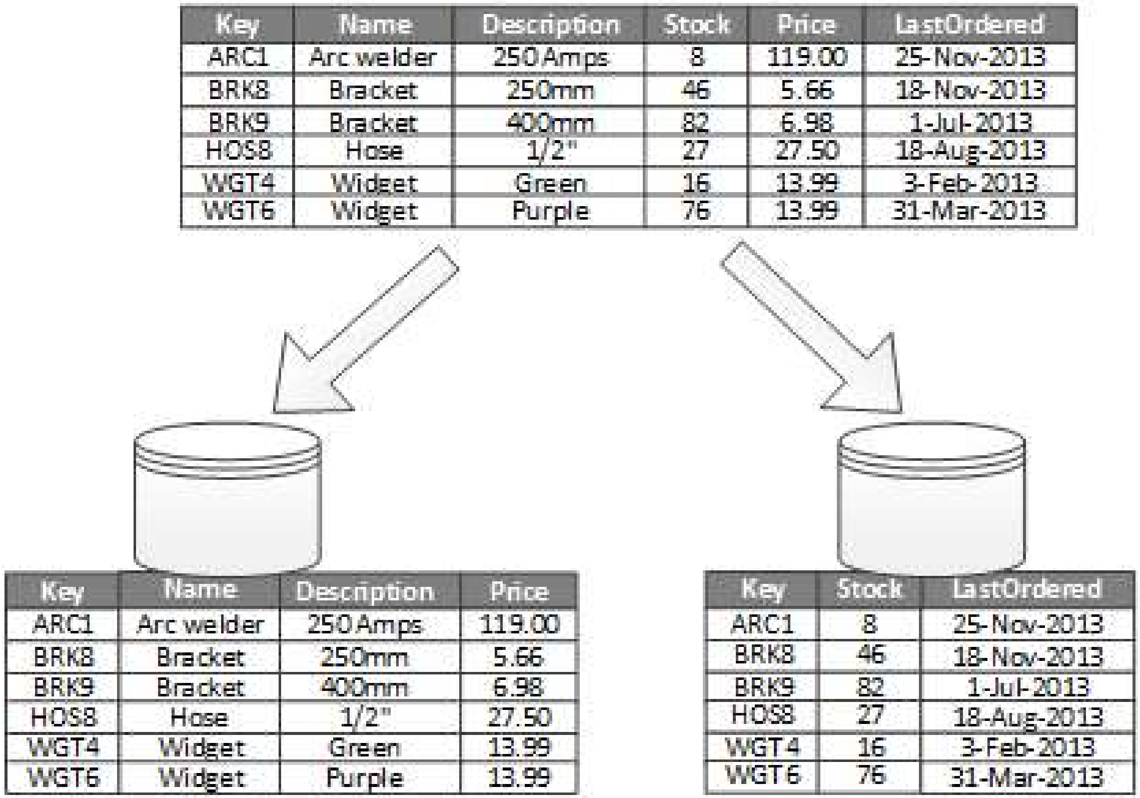
A-G Shard

Key	Name	Description	Stock	Price	LastOrdered
ARC1	Arc welder	250 Amps	8	119.00	25-Nov-2013
BRK8	Bracket	250mm	46	5.66	18-Nov-2013
BRK9	Bracket	400mm	82	6.98	1-Jul-2013

H-Z Shard

Key	Name	Description	Stock	Price	LastOrdered
HOS8	Hose	1/2"	27	27.50	18-Aug-2013
WGT4	Widget	Green	16	13.99	3-Feb-2013
WGT6	Widget	Purple	76	13.99	31-Mar-2013

Vertical partitioning

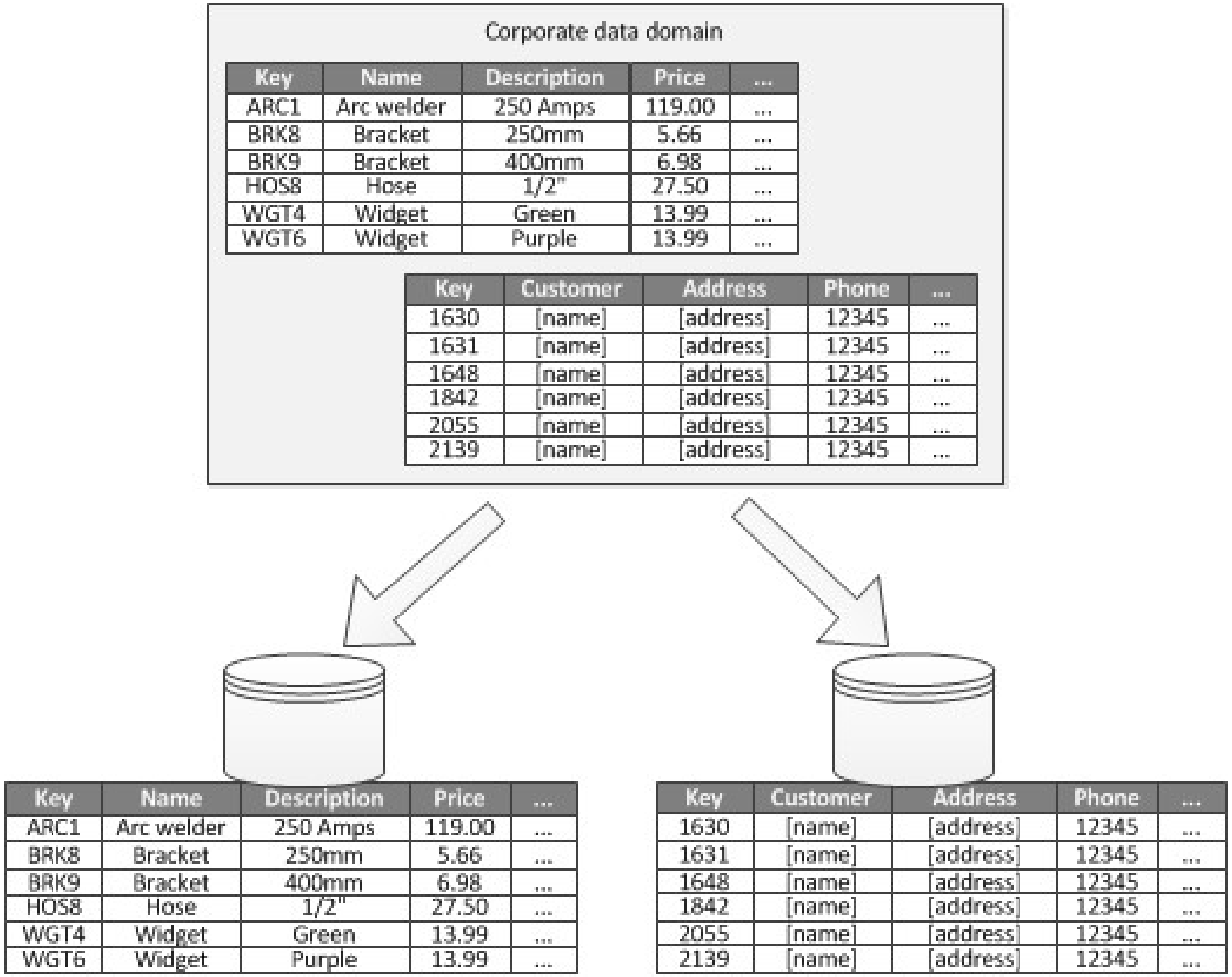


Key	Name	Description	Stock	Price	LastOrdered
ARC1	Arc welder	250 Amps	8	119.00	25-Nov-2013
BRK8	Bracket	250mm	46	5.66	18-Nov-2013
BRK9	Bracket	400mm	82	6.98	1-Jul-2013
HOS8	Hose	1/2"	27	27.50	18-Aug-2013
WGT4	Widget	Green	16	13.99	3-Feb-2013
WGT6	Widget	Purple	76	13.99	31-Mar-2013

Key	Name	Description	Price
ARC1	Arc welder	250 Amps	119.00
BRK8	Bracket	250mm	5.66
BRK9	Bracket	400mm	6.98
HOS8	Hose	1/2"	27.50
WGT4	Widget	Green	13.99
WGT6	Widget	Purple	13.99

Key	Stock	LastOrdered
ARC1	8	25-Nov-2013
BRK8	46	18-Nov-2013
BRK9	82	1-Jul-2013
HOS8	27	18-Aug-2013
WGT4	16	3-Feb-2013
WGT6	76	31-Mar-2013

Functional partitioning



Corporate data domain

Key	Name	Description	Price	...
ARC1	Arc welder	250 Amps	119.00	...
BRK8	Bracket	250mm	5.66	...
BRK9	Bracket	400mm	6.98	...
HOS8	Hose	1/2"	27.50	...
WGT4	Widget	Green	13.99	...
WGT6	Widget	Purple	13.99	...

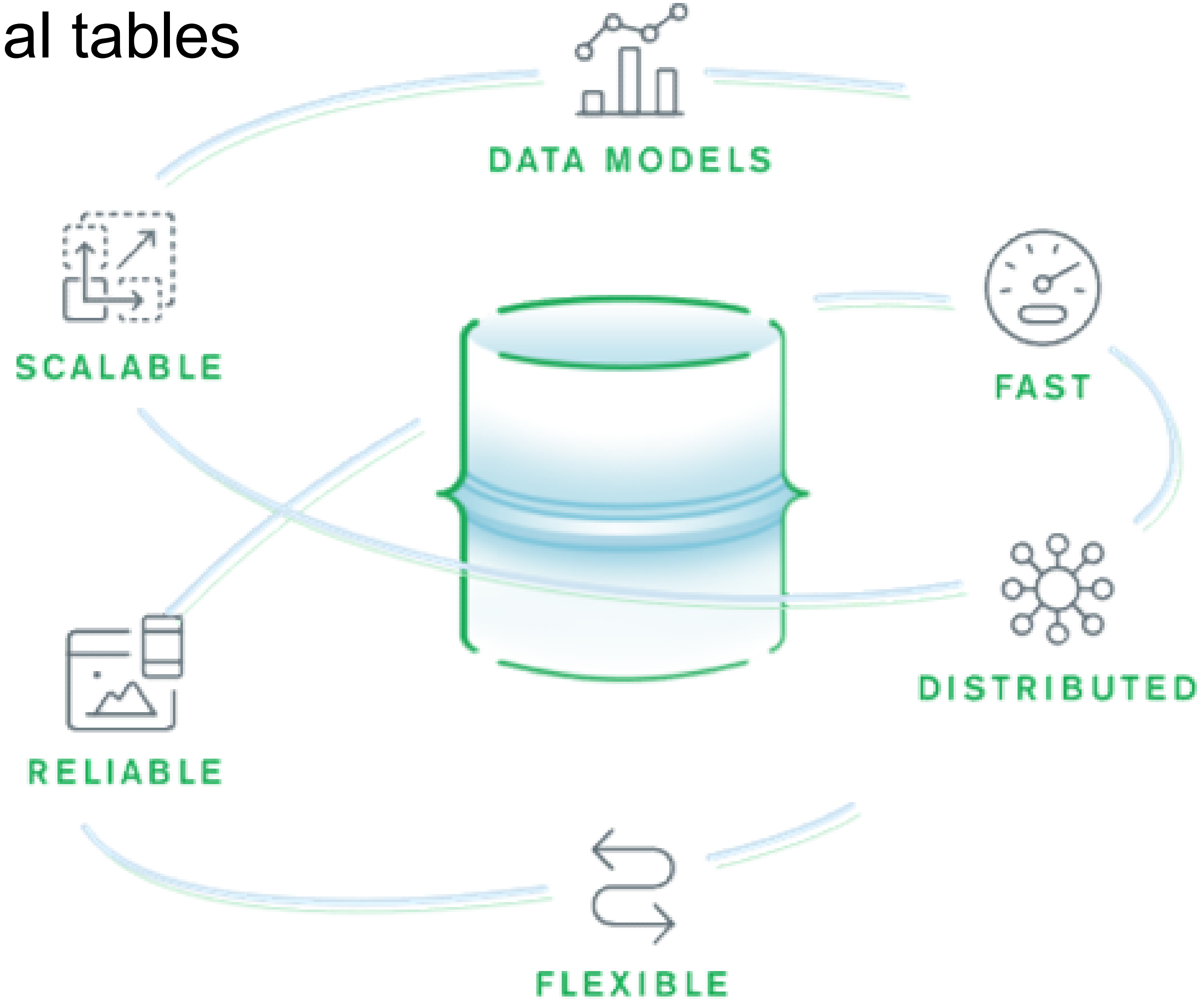
Key	Customer	Address	Phone	...
1630	[name]	[address]	12345	...
1631	[name]	[address]	12345	...
1648	[name]	[address]	12345	...
1842	[name]	[address]	12345	...
2055	[name]	[address]	12345	...
2139	[name]	[address]	12345	...

Key	Name	Description	Price	...
ARC1	Arc welder	250 Amps	119.00	...
BRK8	Bracket	250mm	5.66	...
BRK9	Bracket	400mm	6.98	...
HOS8	Hose	1/2"	27.50	...
WGT4	Widget	Green	13.99	...
WGT6	Widget	Purple	13.99	...

Key	Customer	Address	Phone	...
1630	[name]	[address]	12345	...
1631	[name]	[address]	12345	...
1648	[name]	[address]	12345	...
1842	[name]	[address]	12345	...
2055	[name]	[address]	12345	...
2139	[name]	[address]	12345	...

NoSQL

NoSQL databases are non tabular, and store data differently than relational tables



Document model

- These NoSQL databases replace the familiar rows and columns structure with a document storage model.
- Document-Oriented NoSQL DB stores and retrieves data as a key value pair

Graph model

- It is database that uses graph structures for semantic queries with nodes and edges

- The entity is stored as a node with the relationship as edges.

- Every node and edge has a unique identifier.

Key-value model

- In this NoSQL database model, a key is required to retrieve and update data.

Column-based

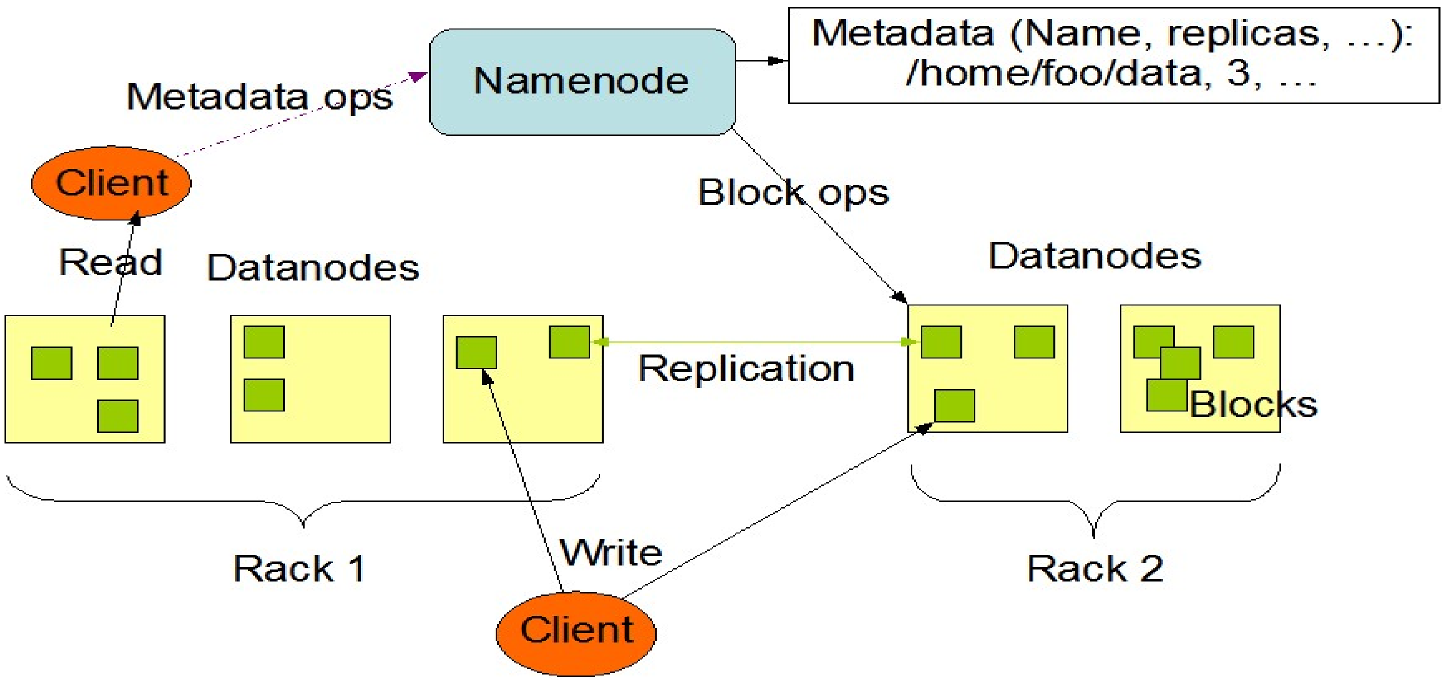
- Column-oriented databases work on columns and are based on BigTable paper by Google.

- Every column is treated separately. Values of single column databases are stored contiguously.

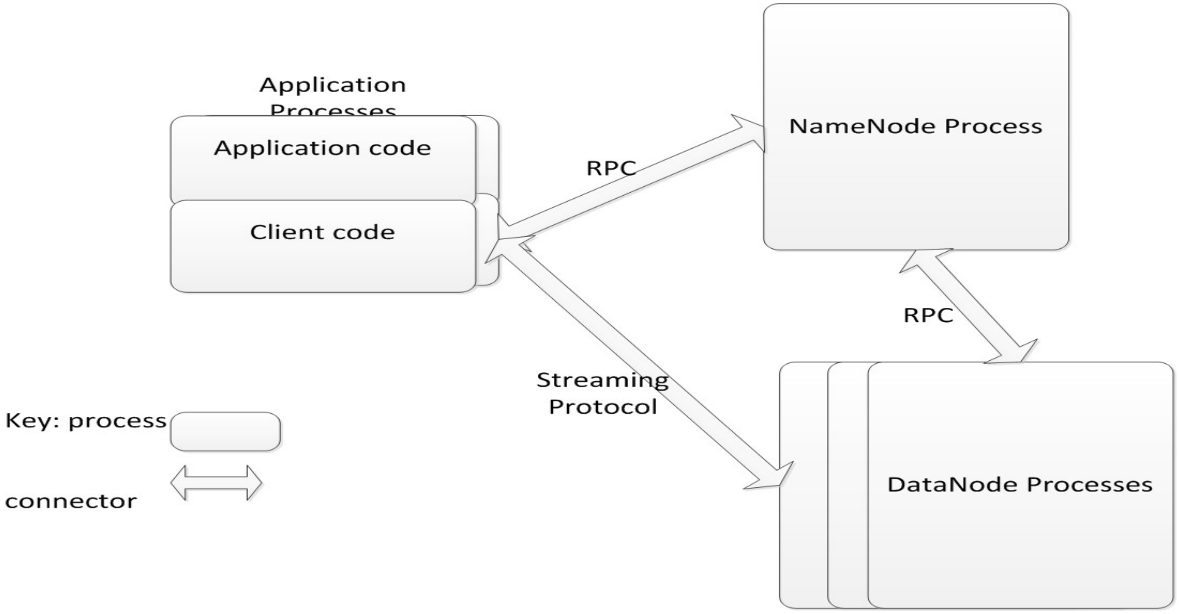
HDFS

- The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware.

HDFS Architecture



HDFS Components



HDFS Write

- Application writes as to any file system
- Client buffers until it gets 64K block
- Client informs NameNode it wishes to write a new block
- NameNode returns list of three DataNodes to hold block
- Client sends block to first DataNode and informs DataNode of other two replicas.
- First DataNode writes block and sends it to second DataNode. Second DataNode writes block and sends it to last DataNode.
- Each DataNode reports to client when it has completed its write
- Client commits write to NameNode when it has heard from all three DataNodes.

HDFS Write – Failure Cases

- Client fails
  - Application detects and retries
  - Write is not complete until committed by Client
- NameNode fails
  - Backup NameNode takes over
  - Log file maintained to avoid losing information
  - DataNodes maintain true list of which blocks they each have
  - Client detects and retries
- DataNode fails
  - Client (or earlier DataNode in pipeline) detects and asks NameNode for different DataNode.
  - Since each block is replicated three times, a failure in a DataNode does not lose any data.

Goals of HDFS

- Fast recovery from hardware failures
- Access to streaming data
- Accommodation of large data sets
- Portability

How MapReduce Works

MapReduce are two functions: Map and Reduce. They are sequenced one after the other.

- The **Map** function takes input from the disk as <key,value> pairs, processes them, and produces another set of intermediate <key,value> pairs as output.

- The **Reduce** function also takes inputs as <key,value> pairs, and produces <key,value> pairs as output.

Combine and Partition

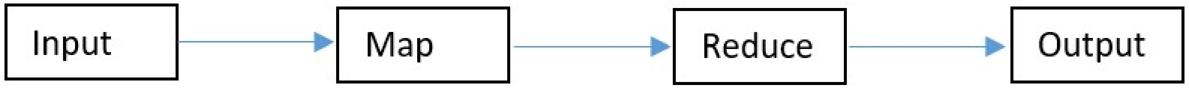
There are two intermediate steps between Map and Reduce.

- **Combine** is an optional process. The combiner is a reducer that runs individually on each mapper server. It reduces the data on each mapper further to a simplified form before passing it downstream.

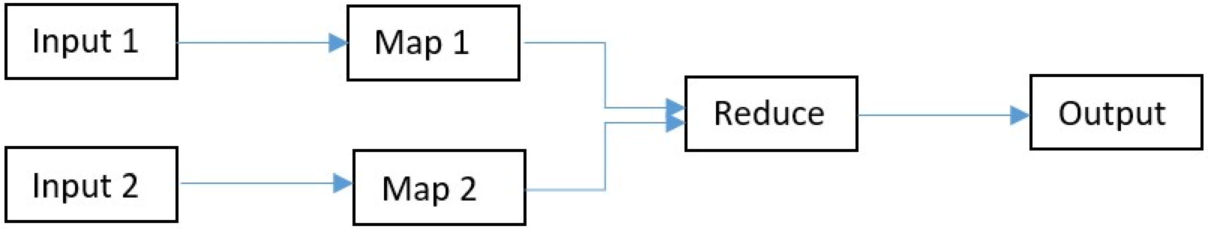
- **Partition** is the process that translates the <key, value> pairs resulting from mappers to another set of <key, value> pairs to feed into the reducer. It decides how the data has to be presented to the reducer and also assigns it to a particular reducer.

MapReduce Pattern

Input-Map-Reduce-Output



Input-Multiple Maps-Reduce-Output



MapReduce Pattern

Input-Map-Combiner-Reduce-Output

Input
Map
Combiner
Reduce
Output

Example: Word count problem

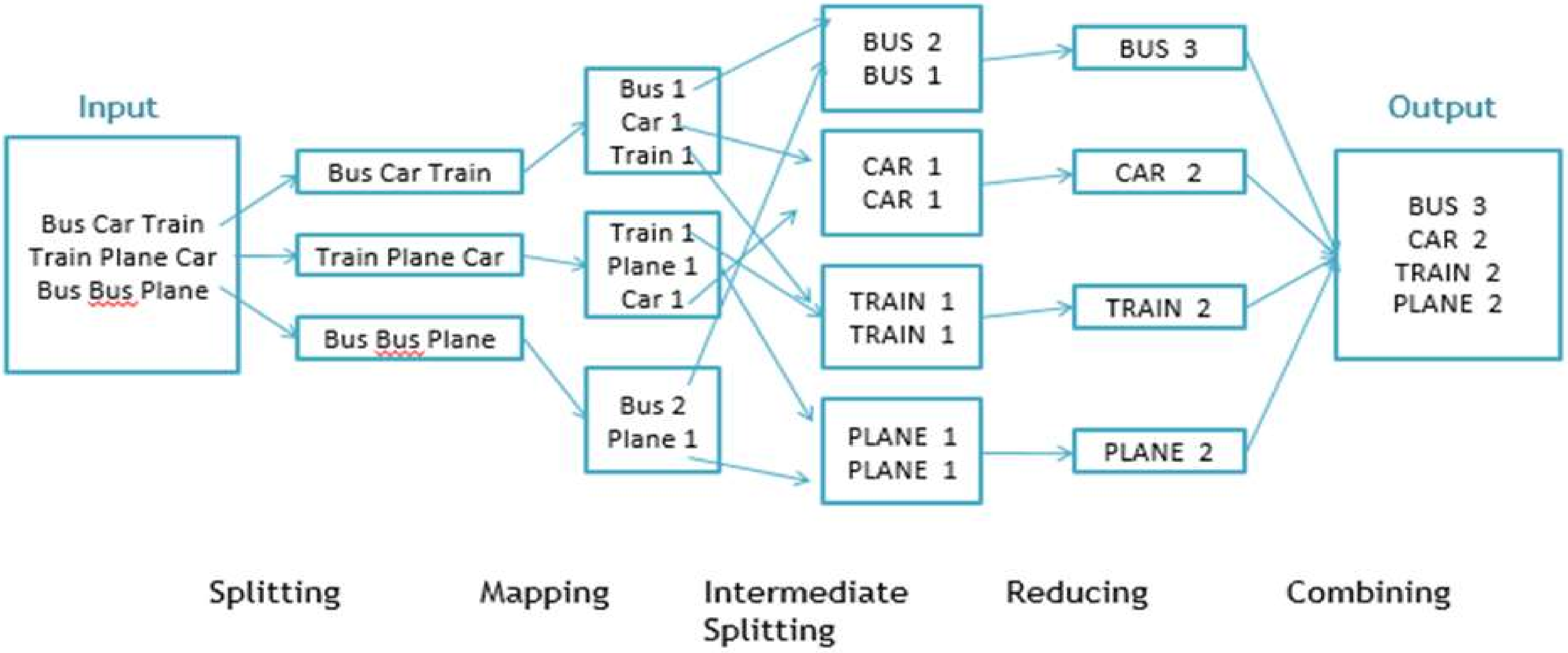


Image: dzone

innovate achieve lead

BITS Pilani
Pilani Campus

Managing high velocity data streams

Content Delivery Network

- While back in 2005 about 1 billion people used the internet on a daily basis, today there are 3.5 billion internet users that share 4 Exabytes (4,000,000,000 Gigabytes) of data every single day.
- Basically a CDN is nothing more than a bunch of globally distributed computers that are directly connected and move data from one end to another.
- A good example of this is YouTube.

YouTube working

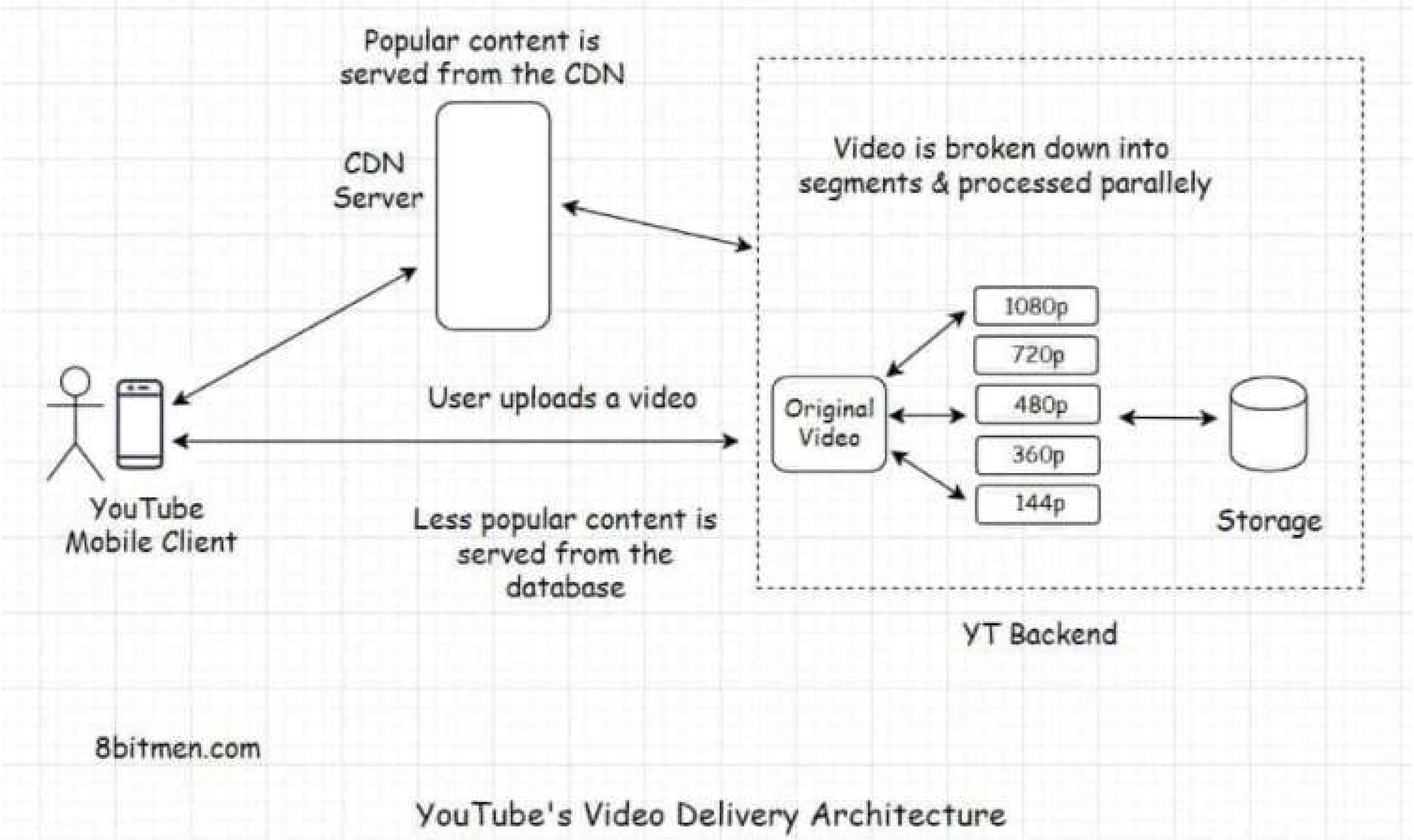
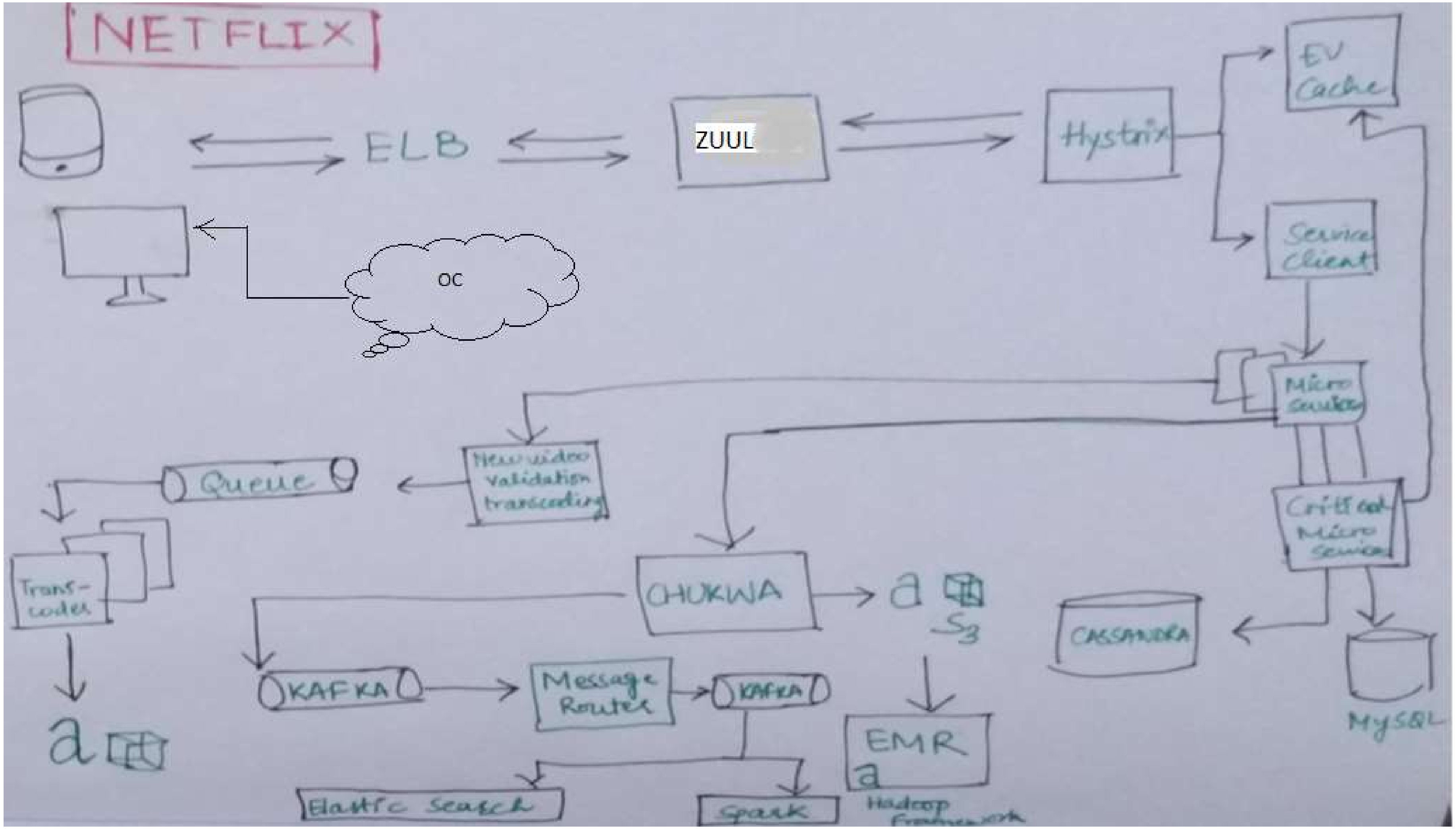

Popular content is served from the CDN
CDN Server
Video is broken down into segments & processed parallely
1080p
720p
Original Video
480p
360p
144p
User uploads a video
YouTube Mobile Client
Less popular content is served from the database
Storage
YT Backend
8bitmen.com
YouTube's Video Delivery Architecture

Image: google

Video streaming: Netflix

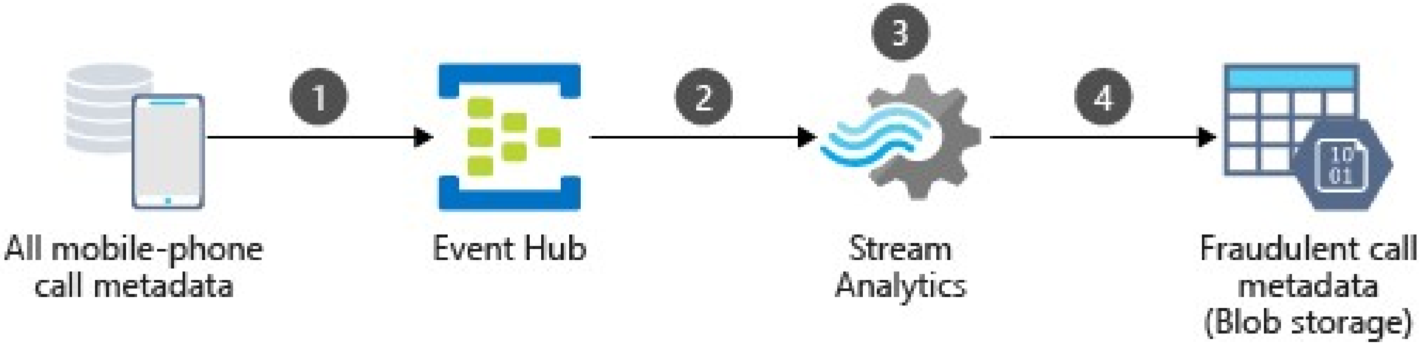
- Netflix launched in 1998. At first they rented DVDs through the US Postal Service. But Netflix saw the future was on-demand streaming video
- In 2007 Netflix introduced their streaming video-on-demand service
- It starts when you hit ‘Play.’
- When Netflix hands off your video to your ISP, they must carry it through their network to your home.

Netflix Architecture



Real Time Fraud Detection

Architecture


1
2
3
4
All mobile-phone call metadata
Event Hub
Stream Analytics
Fraudulent call metadata (Blob storage)
10
01

- Mobile phone call metadata is sent from the source system to an Azure Event Hubs instance.
- A Stream Analytics job is started, which receives data via the event hub source.
- The Stream Analytics job runs a predefined query to transform the input stream and analyze it based on a fraudulent-transaction algorithm.
- The Stream Analytics job writes the transformed stream representing detected fraudulent calls to an output sink in Azure Blob storage.

Web conferencing: Zoom

- Zoom customers with Business subscriptions can enjoy three times as many video participants in their meetings at no additional cost — and without doing a thing.
- such an increase is made possible in the way the Zoom platform is engineered
- From the very beginning, Zoom was engineered to be cloud-native and optimized for video.

Web conferencing: Zoom

There are two important aspects of Zoom's technology stack:

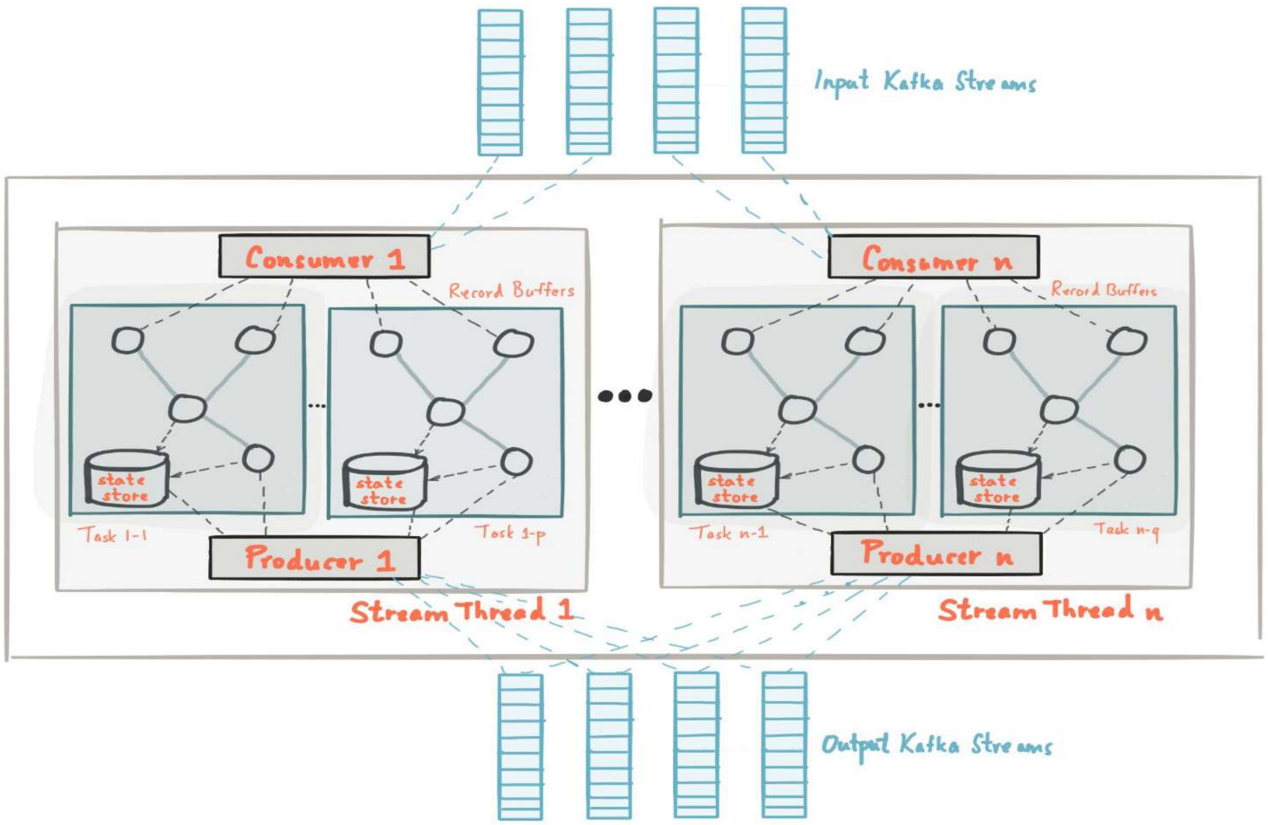
- Cloud network
- Video architecture
    - Distributed architecture
    - Multimedia routing
    - Multi-bitrate encoding
    - Application layer quality of service

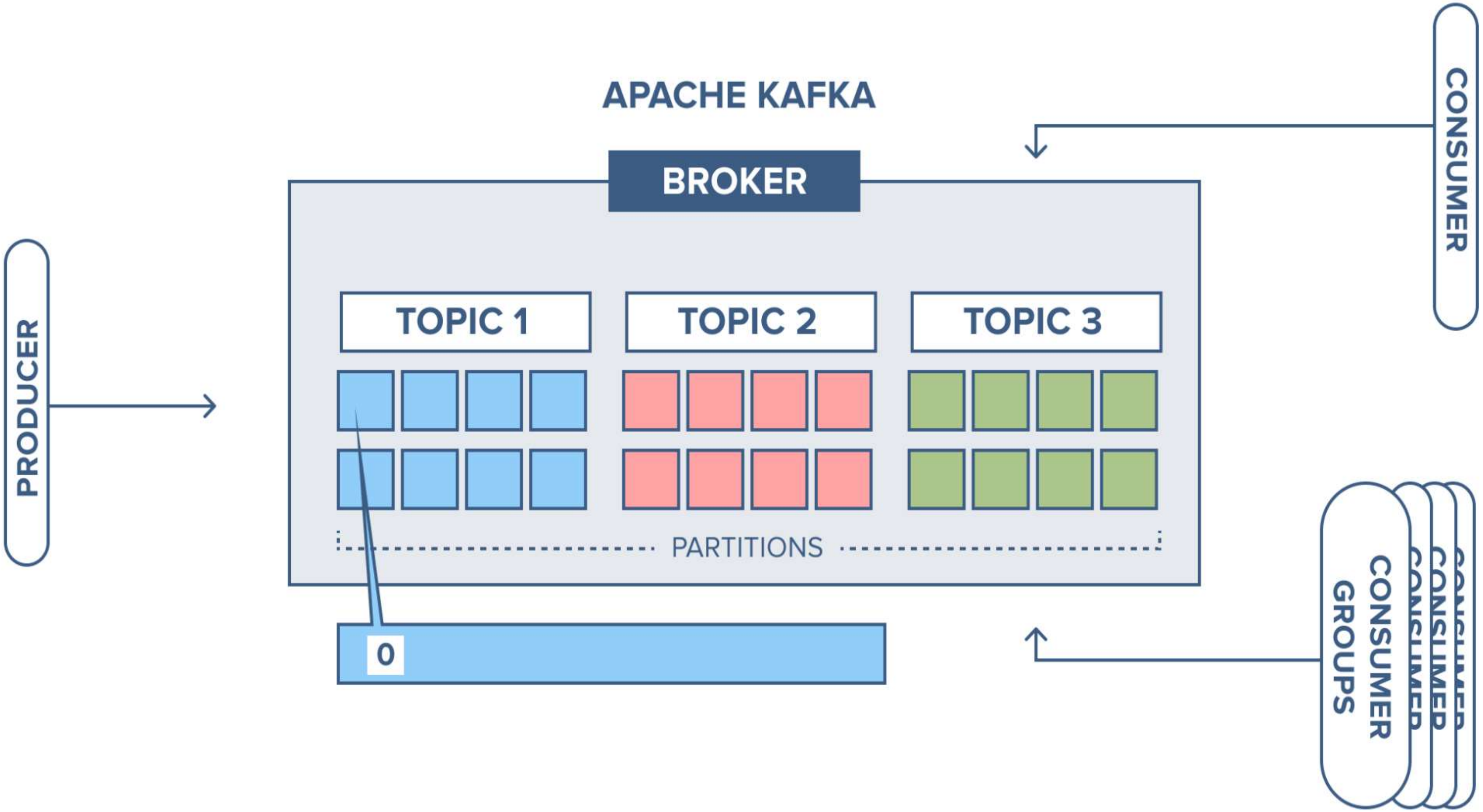
What is Kafka?

- Apache Kafka is a **publish-subscribe based durable messaging system**.
- A messaging system sends messages between processes, applications, and servers.
- Apache Kafka is a software where topics can be defined (think of a topic as a category), applications can add, process and reprocess records.

Kafka

- Kafka Streams simplifies application development by building on the Kafka producer and consumer libraries




APACHE KAFKA
BROKER
TOPIC 1
TOPIC 2
TOPIC 3
PARTITIONS
0
PRODUCER
CONSUMER
CONSUMER GROUPS
CONSUMER
CONSUMER
CONSUMER

Kafka Related Concepts

- Kafka Topics
- Partitioning
- Kafka brokers
- Replication
- Kafka Producers
- Kafka Consumers
- Kafka Connect
- Kafka Streams

Kafka

There are close links between Kafka Streams and Kafka in the context of parallelism:

- Each **stream partition** is a totally ordered sequence of data records and maps to a Kafka **topic partition**.
- A **data record** in the stream maps to a Kafka **message** from that topic.
- The **keys** of data records determine the partitioning of data in both Kafka and Kafka Streams, i.e., how data is routed to specific partitions within topics.

What is Edge Computing?

- Edge computing is a distributed information technology (IT) architecture in which client data is processed at the periphery of the network, as close to the originating source as possible.

- It helps to provide server resources, data analysis, and artificial intelligence to data collection sources and cyber-physical sources like smart sensors and actuators

Edge computing: IoT systems

- Rapidly increasing numbers of IoT devices and resultant data, mean that new techniques are needed to meet customer requirements and ensure effective management need to be explored

Key Benefits of Edge for the IoT

- Low latency
- Longer battery life for IoT devices
- Access to data analytics and AI
- Resilience
- Scalability
- More efficient data management

Example: IoT Image and Audio Processing

- IoT edge introduces new ways of analysing data without having to backhaul the entire image or audio stream
- An edge cloudlet can be used to process the image, video or audio data to determine key information, such as licence plate numbers or the number of people in an area.

Self Study

- https://developer.cisco.com/docs/webex-meetings/#!architecture/overview
- Kafka

References

- https://docs.microsoft.com/en-us/azure/architecture/best-practices/data-partitioning
- https://www.mongodb.com/nosql-explained
- https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
- https://netflixtechblog.com/
- http://highscalability.com/youtube-architecture
- https://docs.microsoft.com/en-us/azure/architecture/example-scenario/data/fraud-detection
- https://blog.zoom.us/
- https://kafka.apache.org/11/documentation/streams/architecture
- https://www.gsma.com/iot/wp-content/uploads/2018/11/IoT-Edge-Opportunities-c.pdf
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7696529/
- Reading material available on Confluent