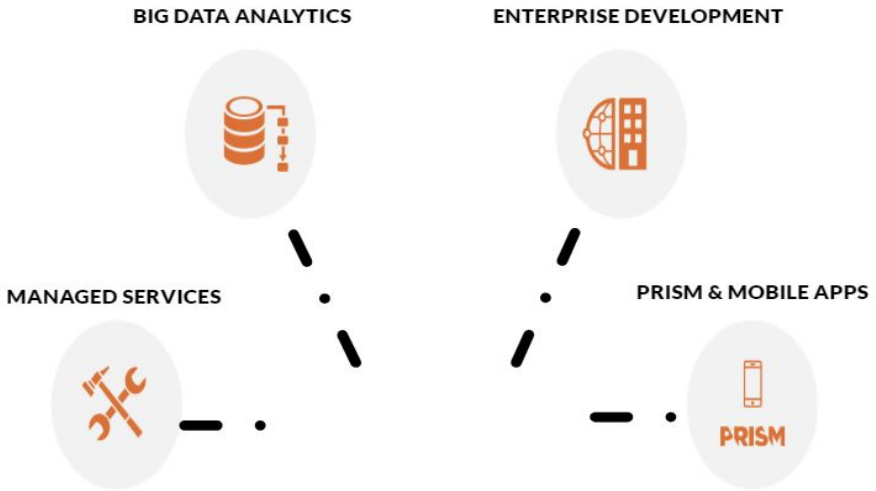
IoT Data Analytics with Lambda Architecture
Dec 2018

Clairvoyant was founded in 2012!

- One of the the fastest growing big data & enterprise development companies
- Founded by highly experienced technologists passionate about designing and building scalable platforms
- Extensive experience in providing strategic and architectural consulting on Big Data platforms and implementations
- Headquartered at Chandler, Arizona

Our offerings:



Our Products
Kogni
Insights

www.clairvoyantsoft.com

Offices
USA - Chandler | Dallas | Boston | Seattle | Washington DC
India - Pune
Canada - Toronto

Today's Presenters

Amit Sutar - Technical Lead (Big Data Practice) at Clairvoyant
https://www.linkedin.com/in/amitbsutar/

Bhushan Rokade - Big data engineer at Clairvoyant
https://www.linkedin.com/in/bhushan-rokade-37a4b070/

Aniruddha More - Senior Software Engineer (Big Data Practice) at Clairvoyant
https://www.linkedin.com/in/aniruddha-more-28ba8240/

Maneesh Kumar - Big data Engineer at Clairvoyant
https://www.linkedin.com/in/maneesh-k-bishnoi/

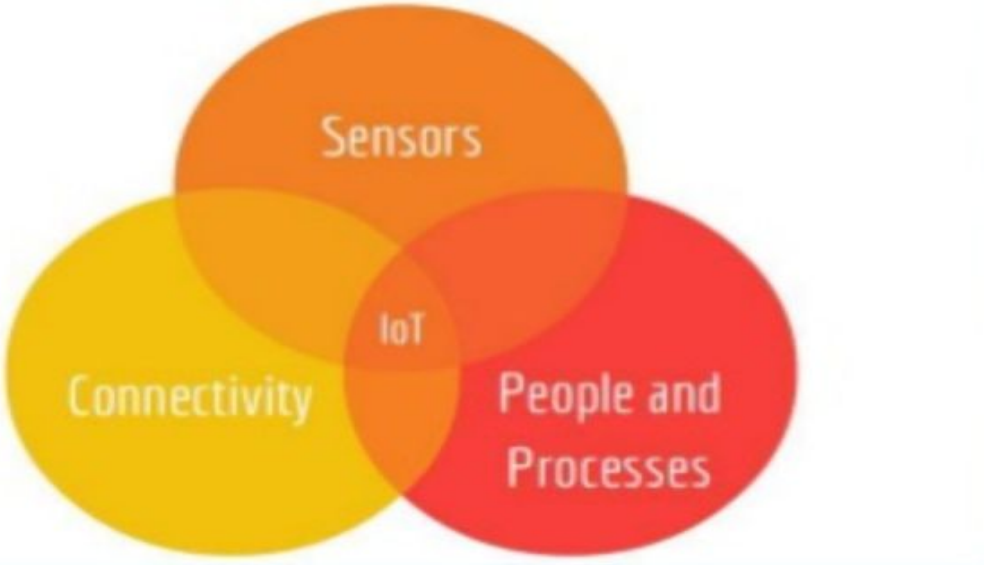
Agenda

- Quick Recap of previous IoT Meetup
- What's new in this edition?
- Need of IoT Data Analytics and Introduction
- Lambda Architecture along with Use Cases
- Demo 1
- Tools and Frameworks
- Demo 2

Quick Recap … What is IoT

The Internet of Things (IoT) concept began as a simple idea of connected devices.

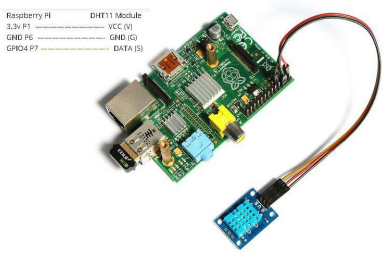
The Internet of Things (IoT) is the network of physical objects—devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity—that enables these objects to collect and exchange data.



Quick Recap … What is IoT

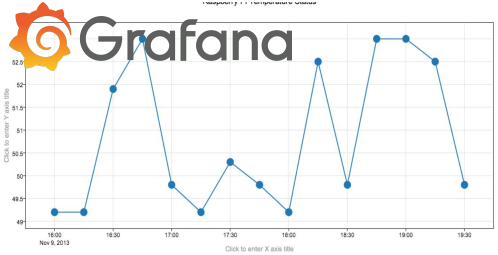
Courtesy: **YouTube** (https://www.youtube.com/watch?v=bCxZsexIUl0)

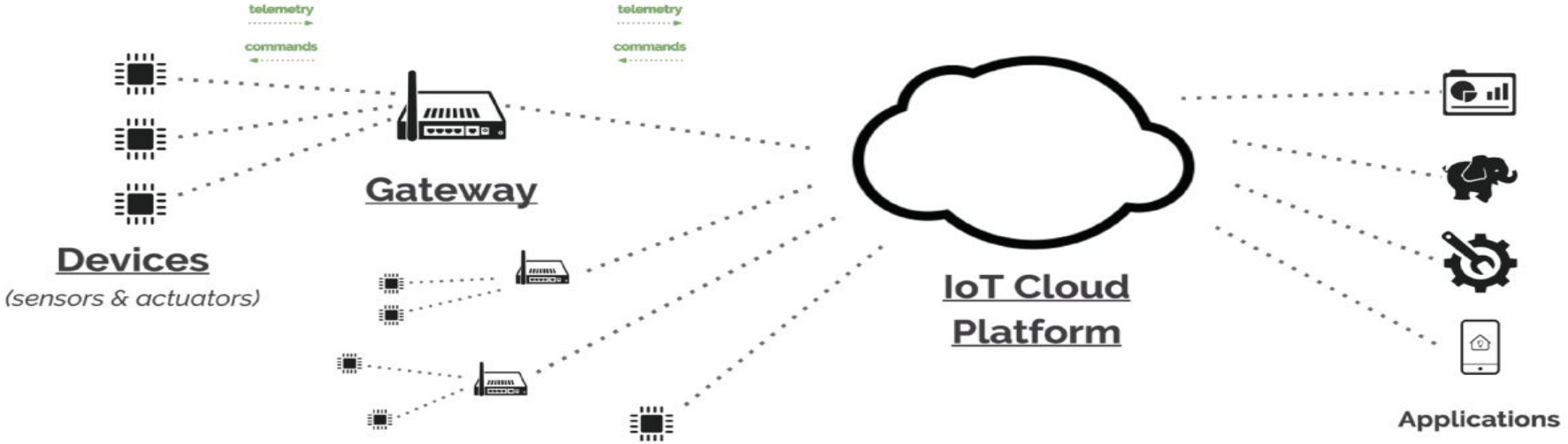
Quick Recap … Architecture for our IoT Use Case



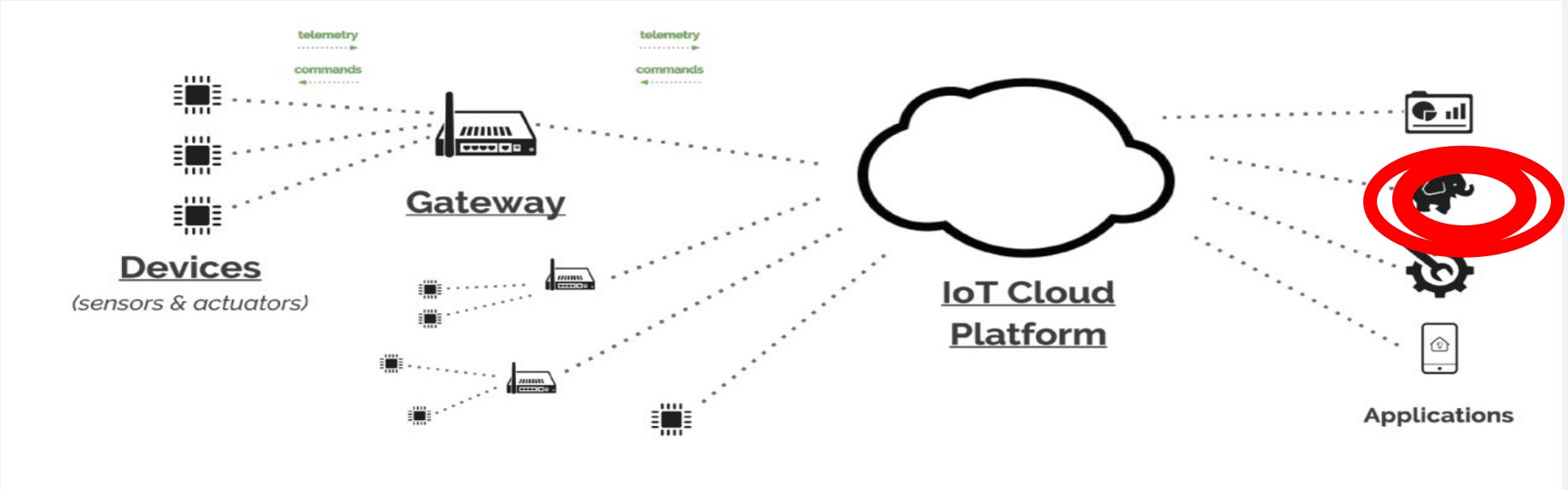
kura

Kapua





What's new in this edition?



Need of IoT Data Analytics

- **Analytics** rather than hardware novelties
- **Insights & Predictions**
- **Variety of use cases**
- **Detect Patterns**
- **New business opportunities**

IoT Analytics Platform Built – Intro

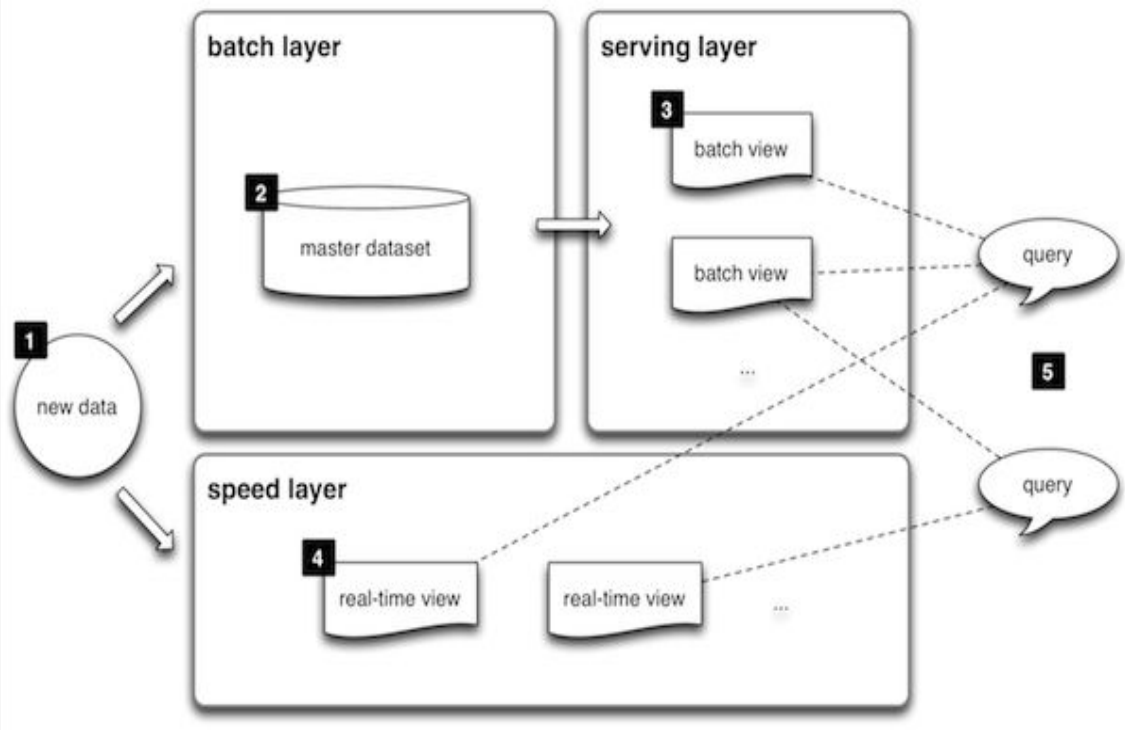
- Choosing right **Software Components**

- Handling **Live Streaming and Historical data**

- **Lambda Architecture**

  In order to accommodate the demand for real-time analytics , we need to design a system that can provide balance between the concept of “single version of truth” and “real-time analytics”. Lambda Architecture is one such method. Lambda Architecture was invented by Nathan Marz to provide fault-tolerant and scalable data processing architecture.

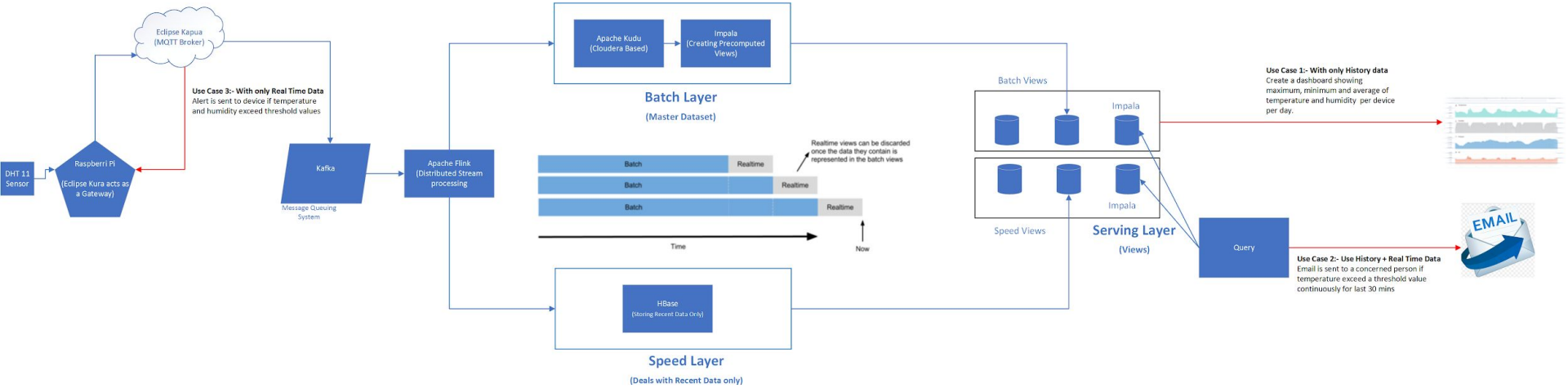
Lambda Architecture



- All data entering the system is dispatched to both the **batch layer** and the **speed layer** for processing.
- The **batch layer** has two functions: (i) managing the master dataset (an immutable, append-only set of raw data), and (ii) to pre-compute the batch views.
- The serving layer indexes the batch views so that they can be queried in low-latency, ad-hoc way.
- The **speed layer** compensates for the high latency of updates to the serving layer and deals with recent data only.
- Any incoming query can be answered by merging results from batch views and real-time views.

Image source: Internet

Design


Eclipse Kapua
(MQTT Broker)
Use Case 3:- With only Real Time Data
Alert is sent to device if temperature and humidity exceed threshold values
DHT 11 Sensor
Raspberri Pi
(Eclipse Kura acts as a Gateway)
Kafka
Message Queuing System
Apache Flink
(Distributed Stream processing
Apache Kudu
(Cloudera Based)
Impala
(Creating Precomputed Views)
Batch Layer
(Master Dataset)
Batch
Realtime
Realtime views can be discarded once the data they contain is represented in the batch views
Time
Now
HBase
(Storing Recent Data Only)
Speed Layer
(Deals with Recent Data only)
Batch Views
Impala
Speed Views
Impala
Serving Layer
(Views)
Query
Use Case 1:- With only History data
Create a dashboard showing maximum, minimum and average of temperature and humidity per device per day.
EMAIL
Use Case 2:- Use History + Real Time Data
Email is sent to a concerned person if temperature exceed a threshold value continuously for last 30 mins

Components - What fits where and how?

Apache Kafka

- Kafka is a high-throughput, low-latency platform for handling real-time data feeds.
- An utility which **reads data from Kapua and publish to a topic on Kafka.**

Apache Flink

- It's a framework and distributed processing engine for **stateful computations** over unbounded and bounded data streams.
- Flink has been designed to run in all common cluster environments, perform computations at **in-memory** speed and at any scale.

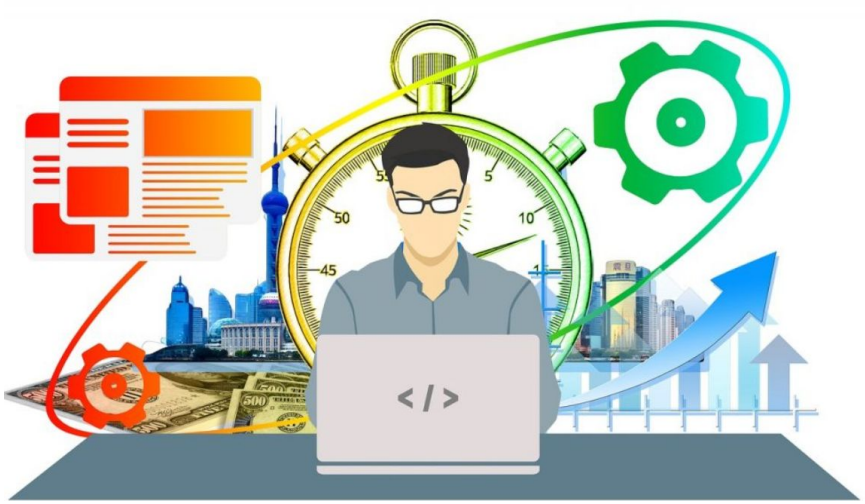
Apache Kudu

- Kudu has tight integration with **Apache Impala**, allowing you to use Impala to insert, query, update, and delete data from tables using Impala's SQL syntax
- This is an alternative to using the Kudu APIs to build a custom Kudu application.

Apache Hbase

- It is mainly used for **ad hoc read** which is retrieved from most recent data entered into the system.

Use Cases

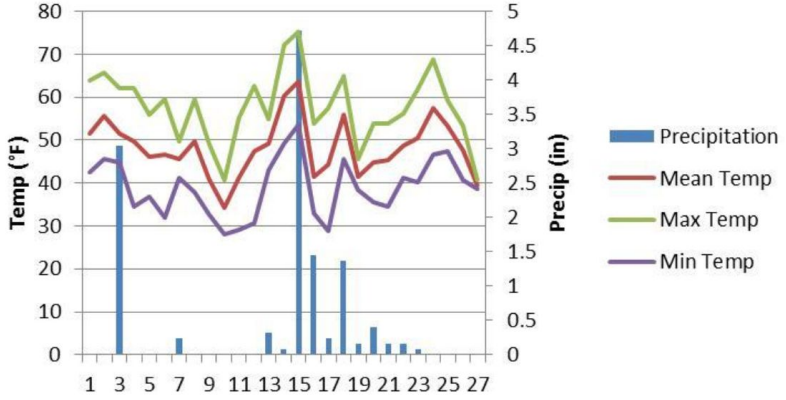


Use Case 1 -

Problem Statement: -

‘Show maximum, minimum and average temperature/humidity per device per day.’

Data:- History Data

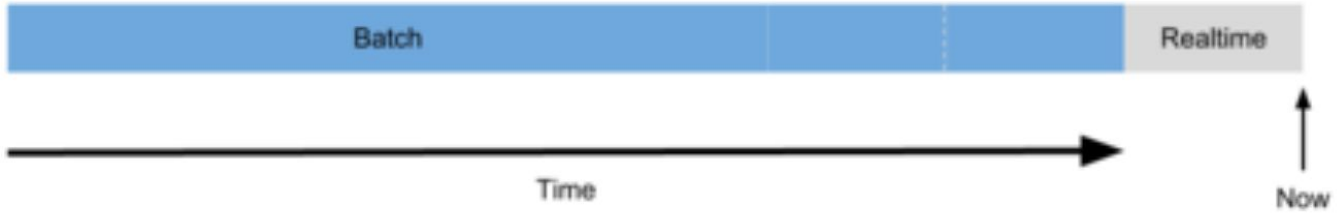


Use Case 2 -

Problem Statement: -

'As per recent research in healthcare sector; an ill person exposed to temperature above 30 degree Celsius for half an hour then there are high chances of that person getting affected with health issues. Hence we need to continuously monitor the temperature for 30 mins and if it is above the set threshold then respective team is notified'

Data:- History + Real Time Data



Use Case 3 -

Problem Statement: -

'If a temperature and humidity coming from sensor rise above threshold value for a particular device then raise an alarm'

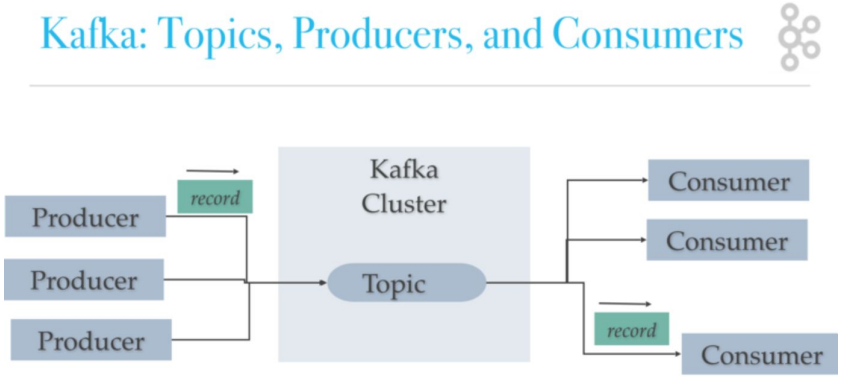
Data:- Real Time Data


LIVE DEMO

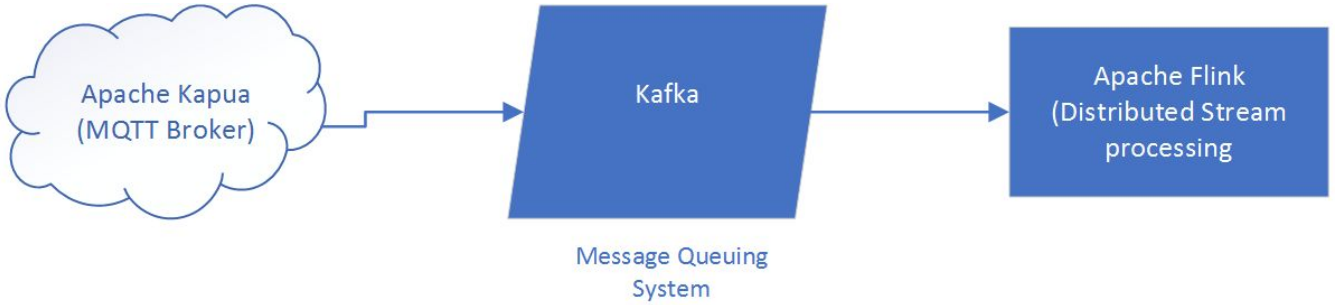
Technology Stack

Apache Kafka

- Kafka is a publish-subscribe, distributed, message queuing system
- It allows users to publish messages to a certain topic; this is then distributed to the subscribers of the topic.
- All messages written to Kafka are persisted and replicated to peer brokers for fault tolerance
- Messages can stay for a configurable period of time (i.e., 7 days, 30 days, etc.)



Role of Kafka in our design :



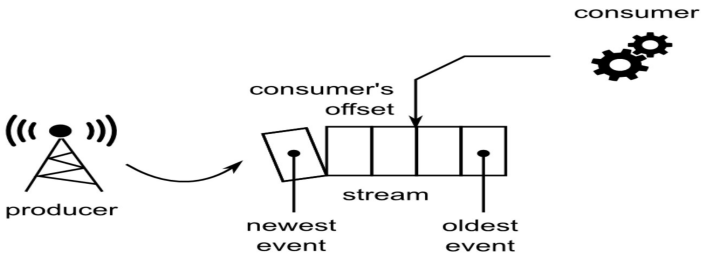
Apache Flink

Apache Flink is an '***Open Source Stream Processing Framework***'

A stream is a sequence of data elements made available over time

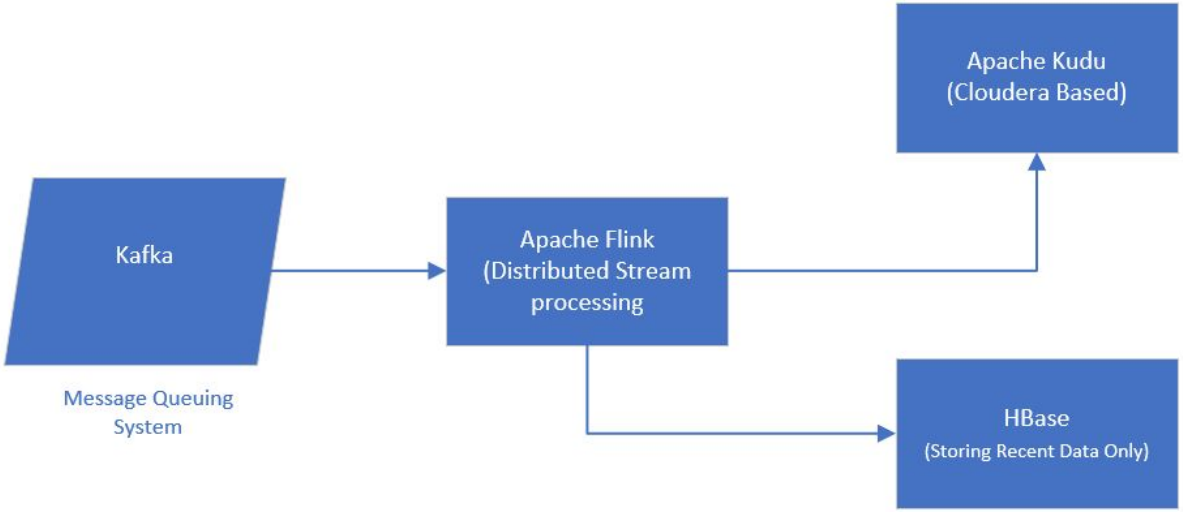
Streams can be :-

- *Unbounded* - have a start but no defined end (streaming application, e.g. twitter feed)
- *Bounded* - have a defined start and end (batch processing e.g. log files)



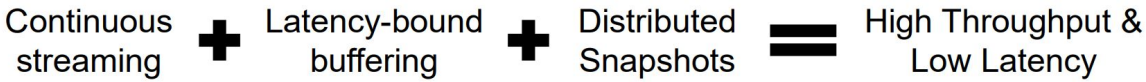
Flink

Role of Flink in our design :



Flink Features:

1. High Throughput and Low Latency :-



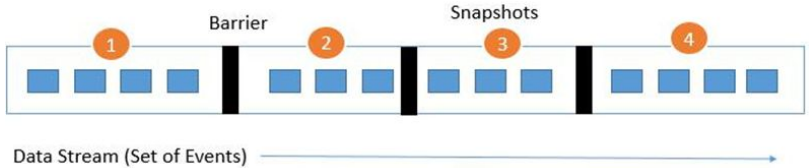
Latency Bound Buffering:-

- Sending individual records over a network causes significant overhead.
- Buffering is the option.
- Disadvantage of buffering is that it adds latency because records are collected in a buffer instead of being immediately shipped.
- Flink ensures that each buffer is shipped after a certain period of time regardless of how much it is filled.

Flink Features Cont..

2. Fault Tolerance :-

- Check pointing is Flink's backbone for providing consistent fault tolerance
- It keeps on taking consistent snapshots
- In case of any failure, Flink stops the executors and resets them and starts executing from the latest available checkpoint.



Flink Check Pointing & Barriers

- Stream barriers are core elements of Flink's snapshots.
- They are ingested into data streams without affecting the flow.
- Each barrier carries a unique ID. They group sets of records into a snapshot.

- **Flink processes each record exactly once.**

Flink Features Cont..

3. Flexible streaming windows

- Flink supports data-driven windows
- Can design a window based on time, counts, or sessions.
- A window can also be customized which allows us to detect specific patterns in event streams.

4. Stream and Batch in One Platform

- Flink provides APIs for both batch and stream data processing
- So once you set up the Flink environment, it can host stream and batch processing applications easily
- Considers batch processing as the special case of streaming.

Flink DataStream API

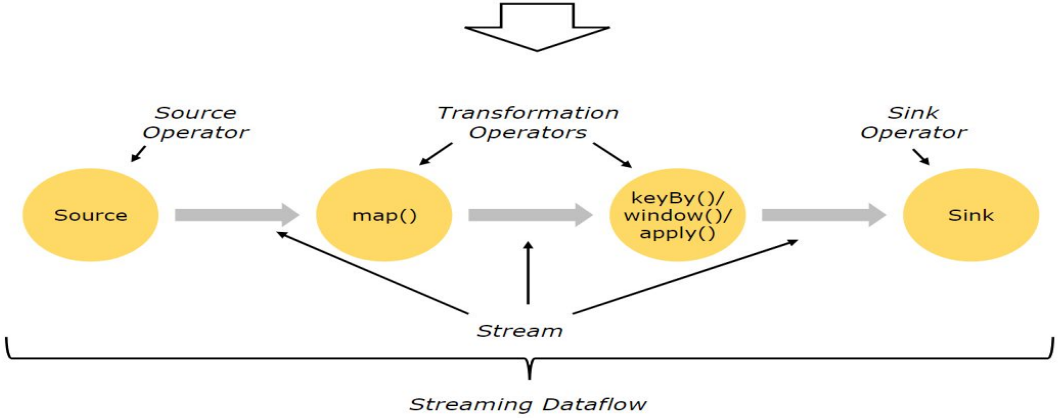
```
DataStream<String> lines = env.addSource(
                                new FlinkKafkaConsumer<>(…));

DataStream<Event> events = lines.map((line) -> parse(line));

DataStream<Statistics> stats = events
        .keyBy("id")
        .timeWindow(Time.seconds(10))
        .apply(new MyWindowAggregationFunction());

stats.addSink(new RollingSink(path));
```

Source

Transformation

Transformation

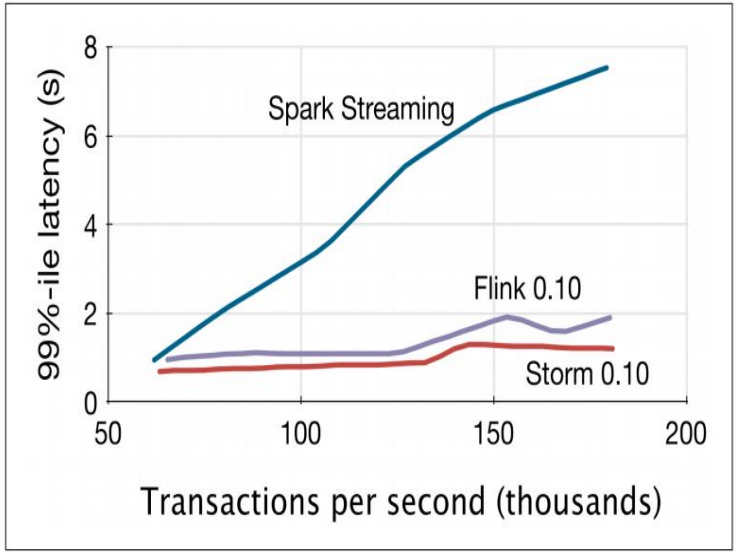
Sink



Flink Vs Spark

Computation Model	Real time/ Event based	Near Real Time/ Micro batches
Abstraction	Batch Over Streaming	Streaming Over Batch
API	Dataset – Batch API DataStream – Streaming API	RDD – Batch API Dstream – Streaming API
Window Criteria	Record Based or User Defined	Time Based
Memory Management	Automatic	Configured
Fault Tolerance Guarantee	Exactly Once	

Flink Performance: The Yahoo! Streaming Benchmark

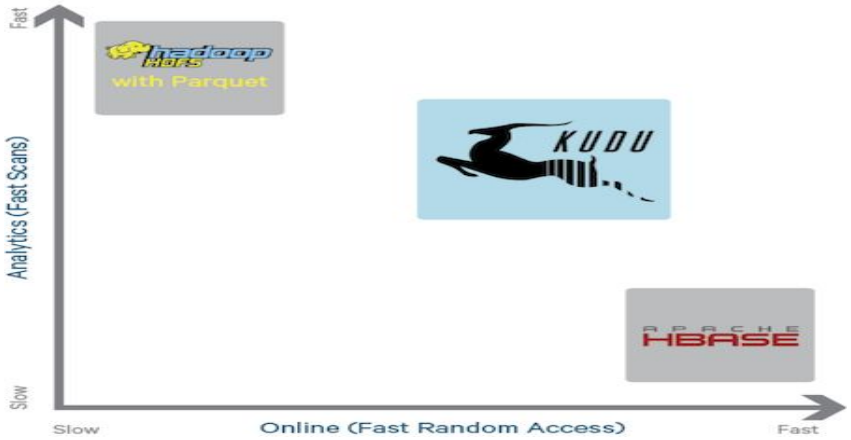


Conclusion:-

- Storm and Flink show low latencies at relatively high throughputs.
- Spark streaming supports high throughputs, but at a relatively higher latency.
- After Flink was rewritten with its own Window Function then it gave much more throughput that Storm.
- Flink runs with exactly once guarantees, Storm with at least once.

Apache Kudu

- Kudu is an open source storage engine for structured data which supports -
  - efficient analytical access patterns
  - low-latency random access
- Kudu: Like HDFS and HBASE in one.
- Storage for Fast Analytics on Fast Data



Why Kudu

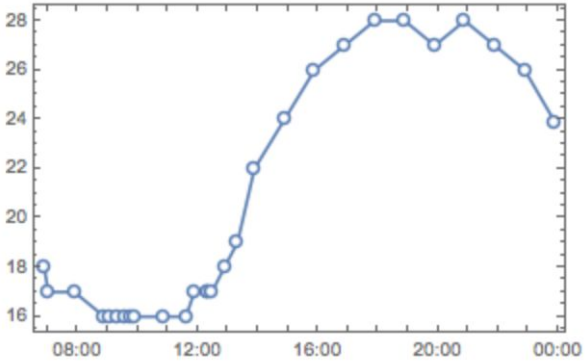
- Fast Columnar Data storage
- Fast Inserts/Updates
- Low-latency random access
- Tabular, SQL like schema
- As per TPC-H (Analytics benchmark), Kudu works well in querying data with low latency than HDFS*

*https://kudu.apache.org/kudu.pdf

Kudu Use Case

Kudu is best for use cases requiring a simultaneous combination of sequential and random reads and writes

- Time Series
  Examples: Stream market data; fraud detection & prevention; network monitoring
  Workload: Insert, updates, scans, lookups



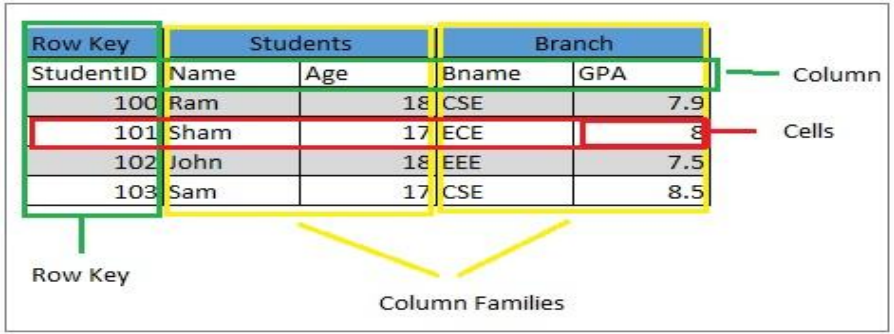
Role of Kudu in our design :

Apache HBase

HBase is a NoSQL database used for real-time data streaming

HBase Unique Features

- A column family oriented database
- HBase is built for low latency operations
- Used extensively for random read and write operations
- HBase stores large amount of data in terms of tables



Sample Hbase Storage Schema

Why to Choose HBase?

e.g. A table for a popular web application may consist of billions of rows. If we want to search particular row from such huge amount of data, HBase is the ideal choice as query fetch time in less.

Where is HBase used?

Telecom Industry

Problem Statement:

- Storing billions of CDR (Call detailed recording) log records generated by telecom domain
- Providing real-time access to CDR logs and billing information of customers considering cost effectiveness

Solution:

HBase is used to store billions of rows of call detailed records.

If 20TB of data is added per month to the existing RDBMS database, performance will deteriorate.

To handle a large amount of data in this use case, HBase is the best solution. HBase performs fast querying and display records.

That apart, HBase can be used **-**

- Whenever there is a need to write heavy applications.
- Performing online log analytics and to generate compliance reports.

Role of HBase in our design :


LIVE DEMO


?
Q
&
&
A
!

What's Next....

- Migration of this implementation to Google Cloud Platform (GCP)
- Add Machine Learning component
- IoT Devices and Communication Security

Thank You!

References

https://www.meetup.com/Pune-Big-Data-Conference-Group/events/256964209/ - Meetup link 15th Dec 2018

https://medium.com/hashmapinc/what-you-need-to-know-about-real-time-architectures-for-iot-time-series-problem-sets-9792f2f8f7bf

https://blog.codecentric.de/en/2016/07/iot-analytics-platform/

https://blog.codecentric.de/en/2016/04/smack-stack-hands

https://www.meetup.com/Pune-Big-Data-Conference-Group/events/254644466/ - Meetup 22nd Sep 2018

https://www.youtube.com/watch?v=LEdMUwsfmYA - **Valarm specifically working on Industrial IoT Solutions**

https://www.iotone.com/guide/iot-and-telematics/g449 - **IoT And Telematics**

https://mycariq.com/ - The company which has platform to support the smart device used in vehicles - Device was referred during the presentation.