Big Data on Cloud Native Platform

Rajesh Balamohan
Sunil Govindan

Speaker Bio

Rajesh Balamohan

Principal Engineer 2 @ **Cloudera**

Apache Hive, ORC Committer & Apache Tez PMC and Committer

@rajeshbalamohan

Sunil Govindan

Engineering Manager @ **Cloudera**

Apache Hadoop, Submarine, YuniKorn PMC member & Committer

@sunilgovind

Agenda

- Why Big Data workloads need to migrate to Cloud
- Aspects of Enterprise Ready Cloud Platform
- Challenges of Big Data on Cloud Platform

Why Big Data workloads need to migrate to cloud ?

About (Big) Data itself...

Key thought process from the customers about today's DATA are,

"Ability to consistently extract accurate business proposition from data"

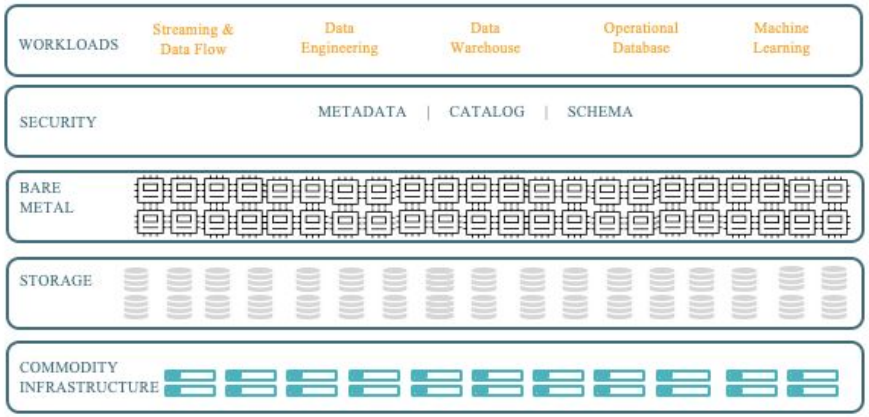
"Data will grow over time - probably, exponentially"

*"Data analytics returns profound business insights only when you have access to **more data**"*

So how do we keep **data** available as needed (*to get value from that data*) ?

Data Architecture Evolution: Gen 1

On Premise



Co-located Storage/Compute
Large, Shared Clusters
(CDH / HDP)

gen-1

Data volumes are growing exponentially and on-prem is not cost effective & scalable!

Cloud Adoption Trend

*"The worldwide infrastructure as a service (IaaS) market grew **37.3%** in 2019 to total $44.5 billion, up from $32.4 billion in 2018, according to Gartner, Inc."*

Cloud Adoption is growing at a rapid pace, why ?

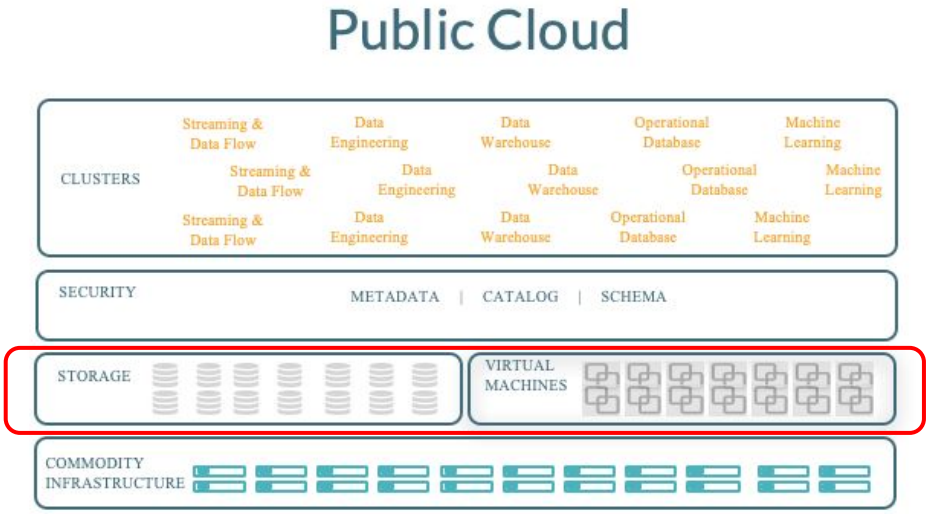
"Cloud computing offers access to *data storage* and *compute* on a more **scalable**, **flexible** and **cost-effective** than can be achieved with an on-premises deployment"

Why Big Data workloads need Cloud?

Some high level advantages:

- **Pay as you go** : No hardware acquisitions, thus Zero CAPEX
- **Self Serve** : Easier Accessibility
- **Cost Effective** & On-Demand
- **Highly Elastic** : Can scale 100s of nodes up/down easily
- **No** more installation/upgrade **hassles**
- **Disaggregated Storage**

Data Architecture Evolution: Gen 2



Disaggregated Storage/Compute
Multiple Clusters
(EMR/ HDInsight)

gen-2

Hadoop in the Public Cloud!

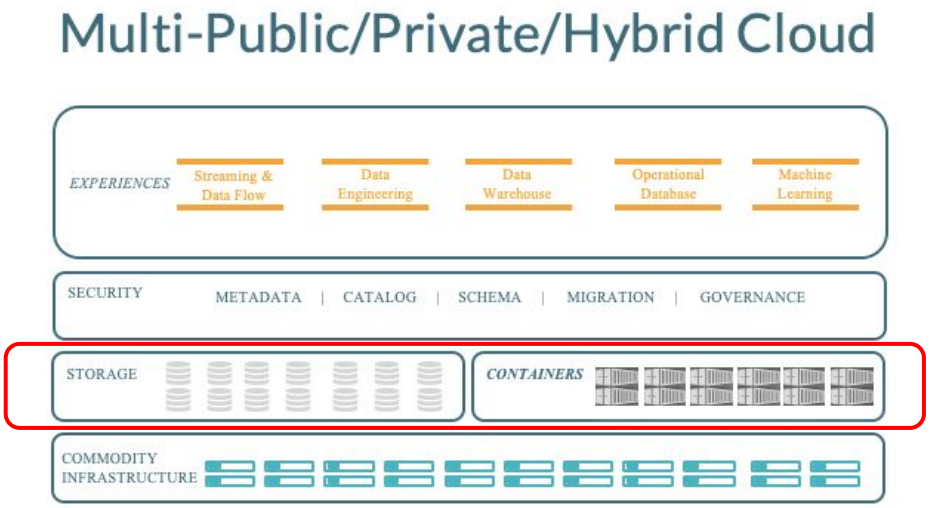
Big Data in Cloud

Hadoop: *"Decade Two, Day Zero"*

Philosophy towards a modern Data Architecture

- Disaggregate storage, compute, security and governance
- Build for extremely large-scale using distributed systems
- **Leverage open source for open standards and community scale**
- Continuously evolve the ecosystem for innovation at every layer, independently

Data Architecture Evolution: Gen 3



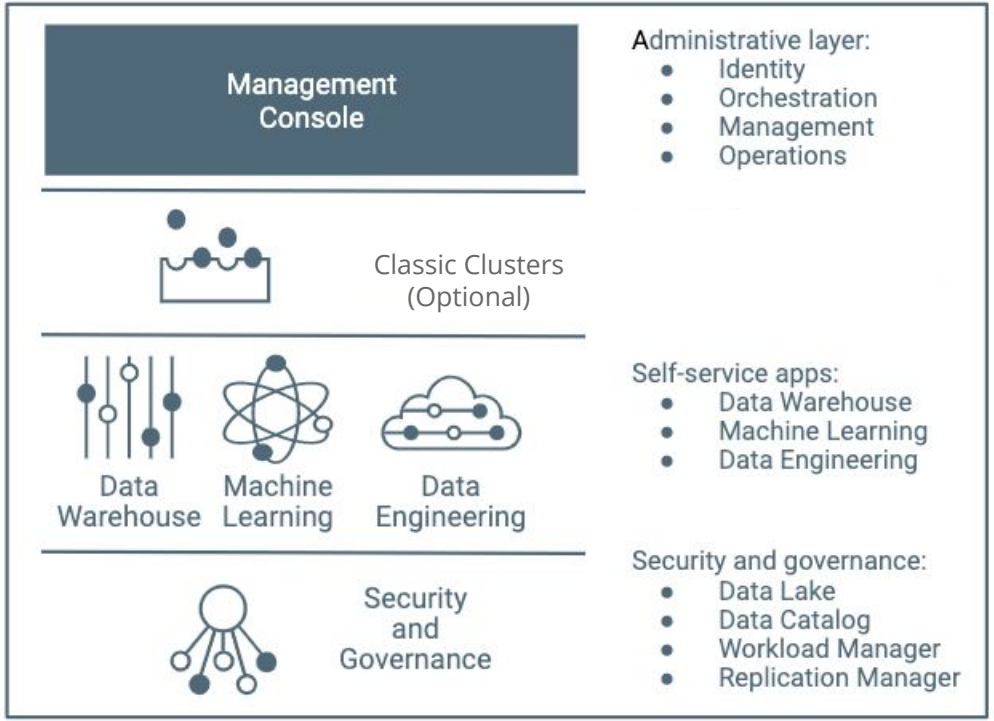
Disaggregated Storage/Compute
Multi-Tenant, Containerized "SaaS Experiences"
(CDP)

gen-3

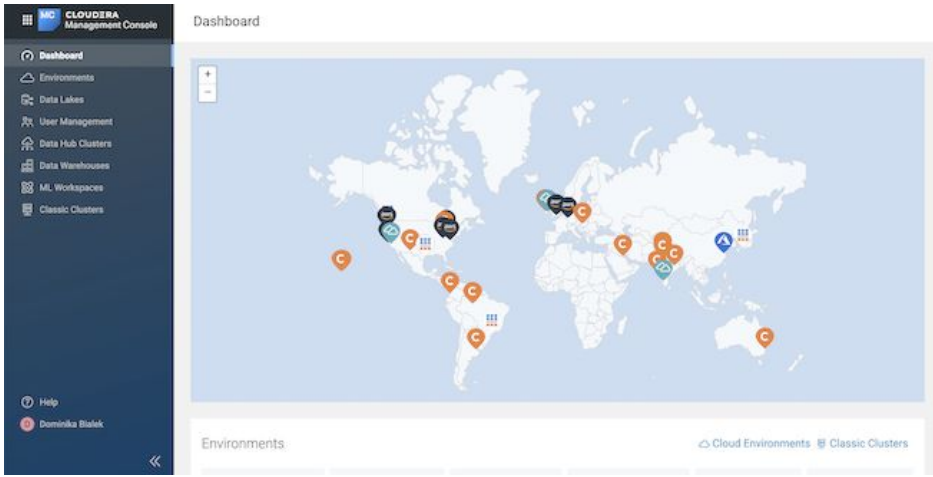
Aspects of Enterprise Ready Cloud Platform

Critical Aspects of Enterprise Cloud Platform

- Manage and monitor **multiple clusters**
- Secure data via single window
- Authentication & Authorization via single window
- Replicate data across multiple clusters on need basis
- Profile and debug queries across multiple clusters via single window
- Multiple experiences depending on the user (Data Engineering, Streaming, Fast Analytics, Data profiling etc)



Manage multiple clusters in central place



Ability to have control over the data end to end

Provide access & control of data to end-users right from ingestion phase to prediction phase.

Big Data Challenges on Cloud Platform

Challenges related to

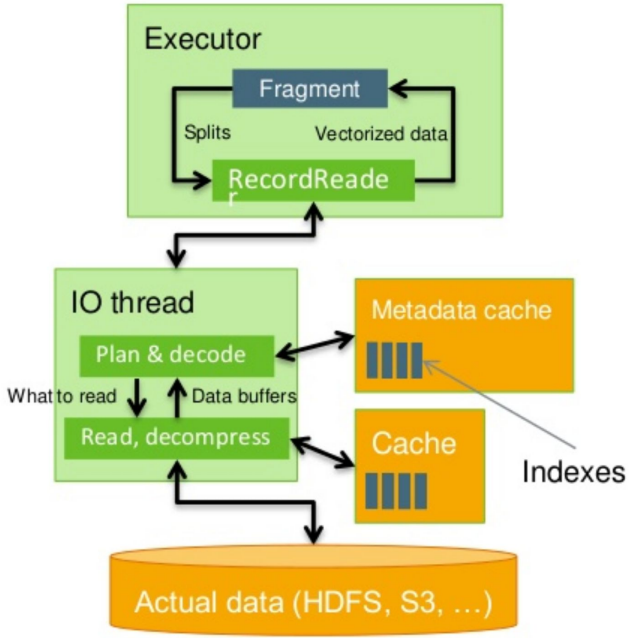
- Storage
- Network
- Compute
- Throttling
- Security
- Hardware Specs

Consistency & Latency Issues with ObjectStores

- Eventual Consistency Issues
  - Certain ObjectStores provide eventual consistency (e.g S3)
    - New files may not be visible for listing (until safely propagated internally).
    - Opening deleted file may be possible due to consistency issues
  - S3Guard
    - Uses “DynamoDB” to persist metadata changes. Provides consistent view of S3 objects for processing.
    - Supports DynamoDB on-demand (i.e no need to explicitly set capacity limits).
- Renames can be expensive
  - Rename = “Copy + Delete” in ObjectStores like S3.
  - Need to build stack which reduces rename operations or favours direct write to destination
- OS Page cache is not leveraged as data is read over network

Intelligent Caching for Query Performance

- Avoid reading same data from ObjectStores
  - Systems like **Hive/LLAP and Impala** cache data locally for improving query performance.



Reduce Network Latency

- Reduce number of SSL connections to ObjectStores
  - Added lazySeek implementation to reduce connection breakages.

AutoScaling

- Determining the right cluster size can be challenging.
- AutoScaling helps in scaling up/down instances depending on workload
  - Concurrency Based AutoScaling
    - Helps in controlling number of parallel queries
  - Query Isolation
    - When queries scan beyond a certain limit, new clusters are automatically spun up.

Affinity Policies for better Network Throughput

- AutoScaling policies allow you spin up instances across different availability zones
  - By default cloud providers tend to spread instances across AZ for availability.
- Impacts network throughput for nodes with 10Gbps speed
- Set affinity policy to have the instances in the same availability zone

Spin up Time

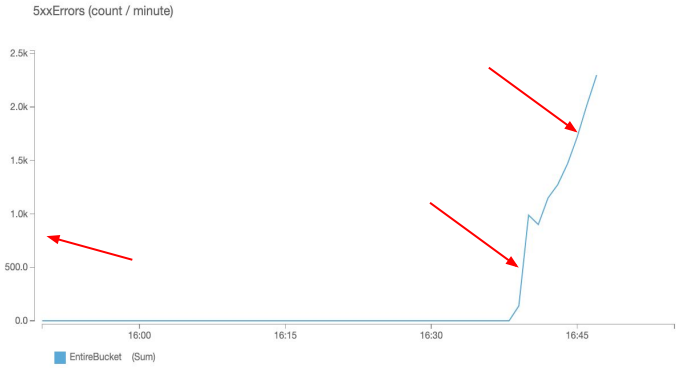
- Cluster/Compute spin up time plays a crucial role in adoption and reducing cost.
- Containerized deployments help a lot in reducing spin up time significantly with K8S
  - 10s of seconds as opposed to minutes

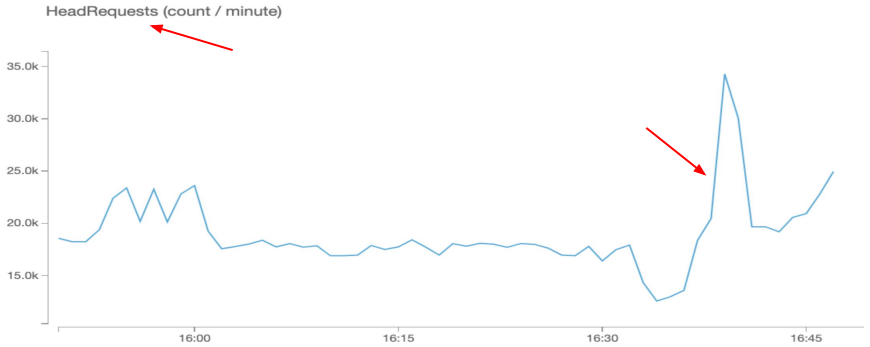
K8S: Pods can have same hostname/port

- Pods can have same hostname/port after restart
- This causes trouble for processes tracking nodes based on hostname/port
- Added flexibility in the stack to take care of this situation
  - E.g TEZ-4179: [Kubernetes] Extend NodeId in tez to support unique worker identity

Throttling

- Cloud services throttle requests
  - Throttling limits vary across cloud vendors
- Critical to monitor throttling metrics
  - Desirable to enable metrics logging in ObjectStore
  - Accuracy limited to per minute in most of the objectstores





Throttling

System trying to resend data over SSL on receiving 503 (throttling) causing CPU spike

Security

- Perimeter Security
- Encrypted data at rest
- Transfer of intermediate data encrypted
- Need to use optimised libs for improving transport security

Hardware Specs across Cloud Vendors

```
AWS Disk:
==========
[hive@query-executor-0-1 lib]$ dd if=/dev/zero of=/apps/llap/work/testing.txt bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB) copied, 3.72015 s, 289 MB/s

Azure Disk:
==========
[hive@query-executor-0-3 work]$ dd if=/dev/zero of=/apps/llap/work/testing.txt bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB) copied, 12.075 s, 88.9 MB/s
```

- Watch out for hardware specs across cloud vendors.
  - E.g SSD in Azure can have different perf characteristics than AWS
- OS settings have to be tweaked accordingly
  - E.g network, disk settings
- Choose optimal instance for the workload
  - E.g Instances with high density disks may not be needed as data is stored in ObjectStore
  - Too little disk space can hurt intermediate data being written out.

Tomorrow ...

- Plenty of challenges to run Big Data workloads on Cloud
  - Great efforts from Open Source community!
- Users need "*No vendor lock in*"
  - An Open Data layer for multi-cloud (SODA, CSI etc with infinite possibilities)
  - Network standards across clouds (CNI)
  - Data Lineage and governance for user (Apache Atlas)
  - Security and access as open standard (Apache Ranger)
- Users are looking for an *Open Data Architecture* for multiple clouds which is enterprise ready!

Thank You

- References
  - Cloudera Data Platform (Multi Cloud): https://docs.cloudera.com/cdp/latest/index.html
  - Hadoop: Decade two, Day zero: https://blog.cloudera.com/hadoop-decade-two-day-zero/

- Cloudera is hiring!
  - https://www.cloudera.com/careers/locations/apac.html

Q/A