Bigdata and Hadoop

Haridas Narayanaswamy
https://haridas.in
06/Mar/2019

Quick reminder

1. HYD and BLR: Online Attendance registration link

https://docs.google.com/spreadsheets/d/1DAuclSqS6xdv4QtixUQIs6OjpxXRLVEEmnCeAkgD9z8/edit#gid=0

2.

docker pull haridasn/hadoop-2.8.5

docker pull haridasn/hadoop-cli

Agenda

- Introduction to the big-data problems
- How we can scale systems.
- Hadoop introduction
- Setup a Hadoop cluster on your laptop.

IO bound vs CPU bound problems

- Downloading a file
- Bitcoin mining (proof of work)
- Watching a movie
- ETL jobs
- ML training

Latency Numbers

- nano second - unit speed

```
Latency Comparison Numbers (~2012)
----------------------------------
L1 cache reference                           0.5 ns
Branch mispredict                            5   ns
L2 cache reference                           7   ns                      14x L1 cache
Mutex lock/unlock                           25   ns
Main memory reference                      100   ns                      20x L2 cache, 200x L1 cache
Compress 1K bytes with Zippy             3,000   ns        3 us
Send 1K bytes over 1 Gbps network       10,000   ns       10 us
Read 4K randomly from SSD*             150,000   ns      150 us          ~1GB/sec SSD
Read 1 MB sequentially from memory     250,000   ns      250 us
Round trip within same datacenter      500,000   ns      500 us
Read 1 MB sequentially from SSD*     1,000,000   ns    1,000 us    1 ms  ~1GB/sec SSD, 4X memory
Disk seek                           10,000,000   ns   10,000 us   10 ms  20x datacenter roundtrip
Read 1 MB sequentially from disk    20,000,000   ns   20,000 us   20 ms  80x memory, 20X SSD
Send packet CA->Netherlands->CA    150,000,000   ns  150,000 us  150 ms
```

Application Architectures

Online

Standalone

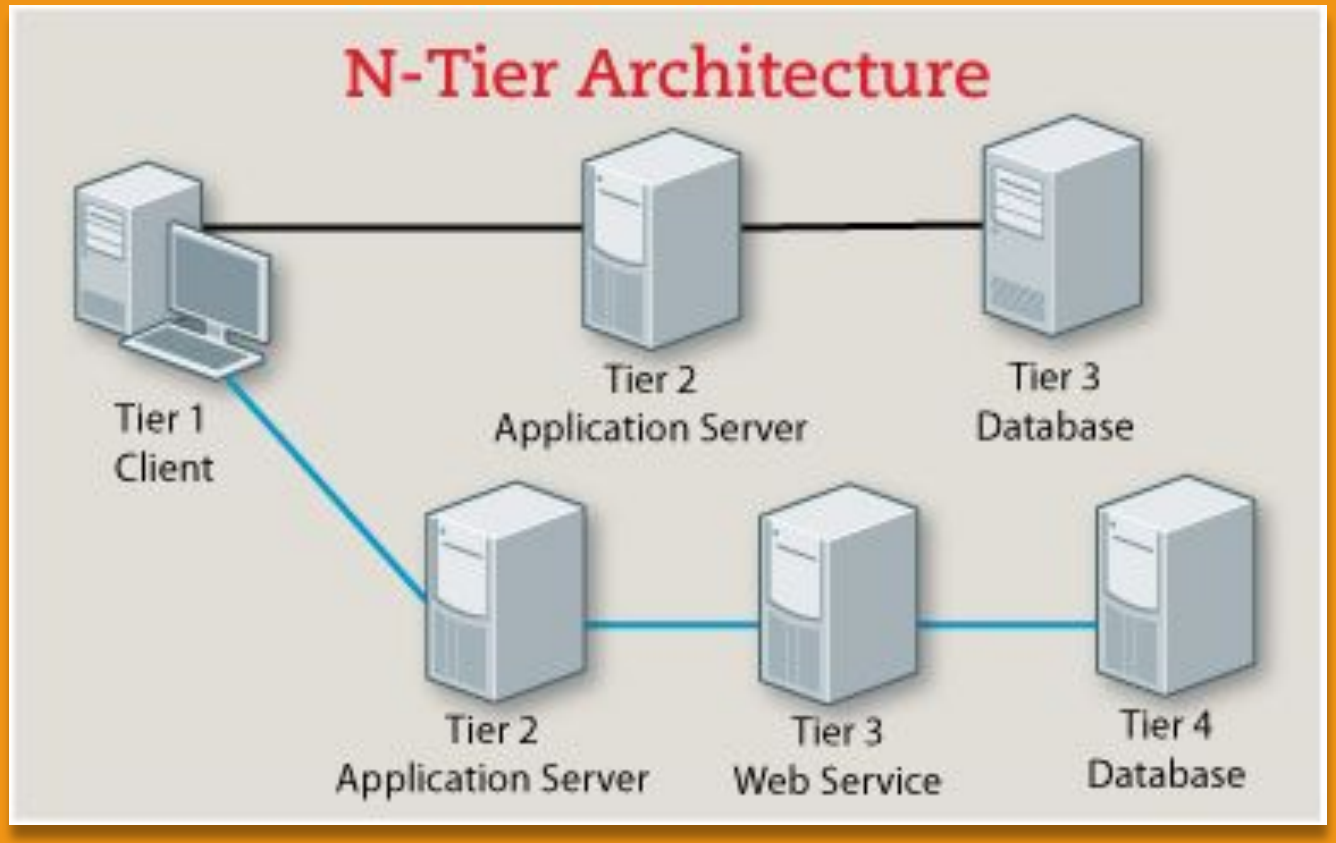
Distributed

Monolithic

- All on single machine
- Every application starts here

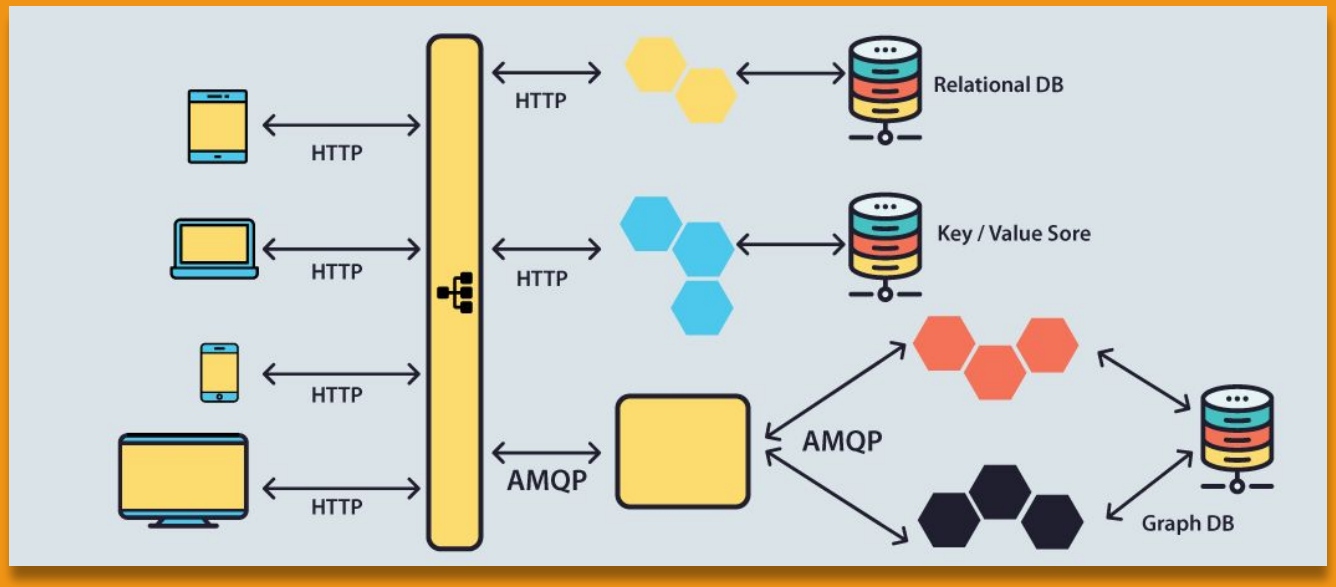
N-tier systems

- Multiple services in your application
- Separation of concerns
- Master-slave database systems
- Load balancer
- Caching layer
- Pretty good for standard application workloads.



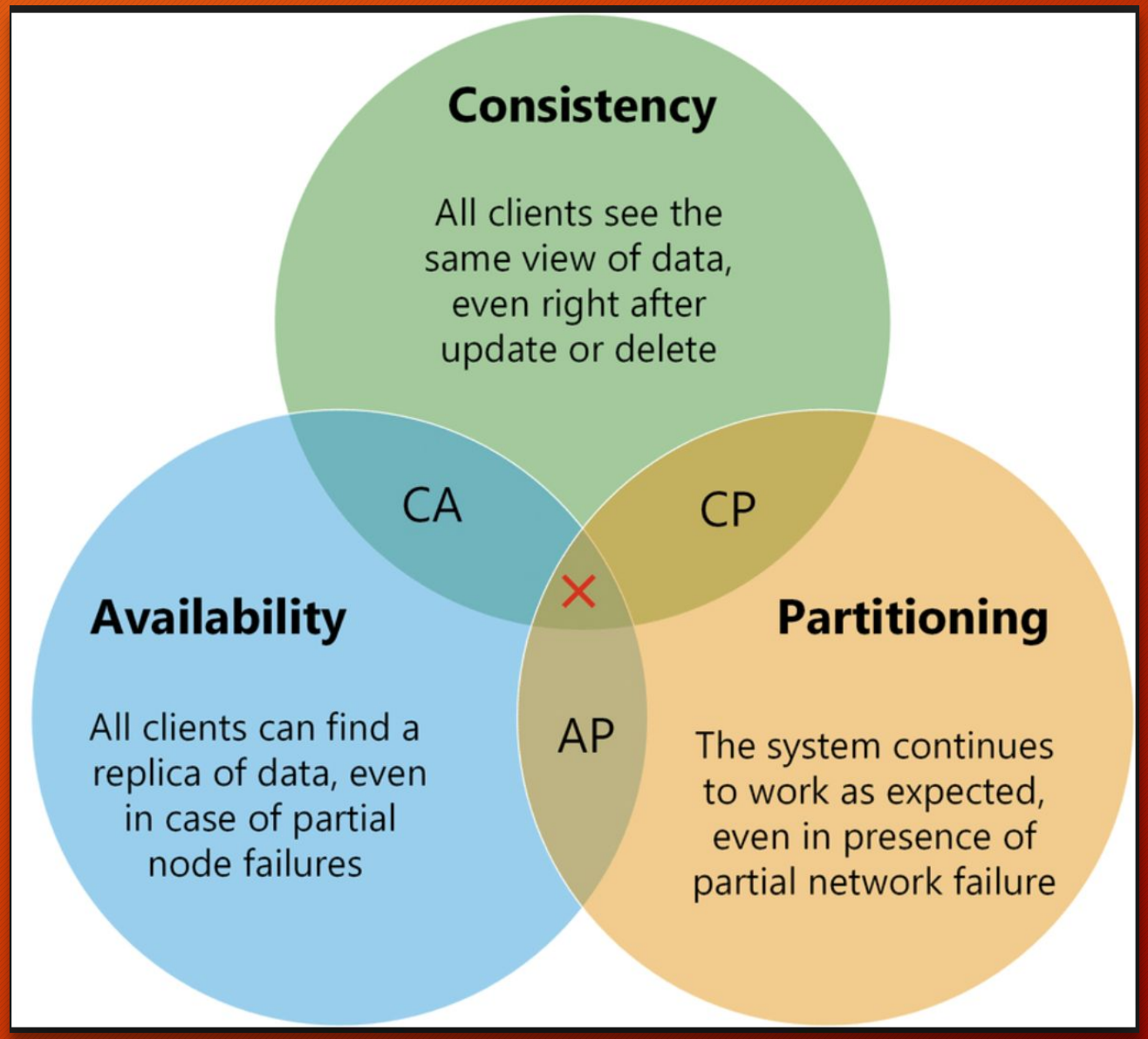
SOA and Microservices

- Service Oriented Architecture
- Each service will do only one task – fine grained separation, hence named as micro-services
- API Gateways
- Messaging Systems
- Scalable Databases accessible to micro services
- Caching services
- Etc.



CAP Theorem

- **Consistency** - Reads from all machines on the cluster would give same data.
- **Availability** - All operation would produce some result, even though they aren't consistent.
- **Partition tolerance** - System perform normally even after some nodes got disconnected from network.
- Eventual Consistent systems
- Split brain problem



Bigdata platforms

Where they fit in ?

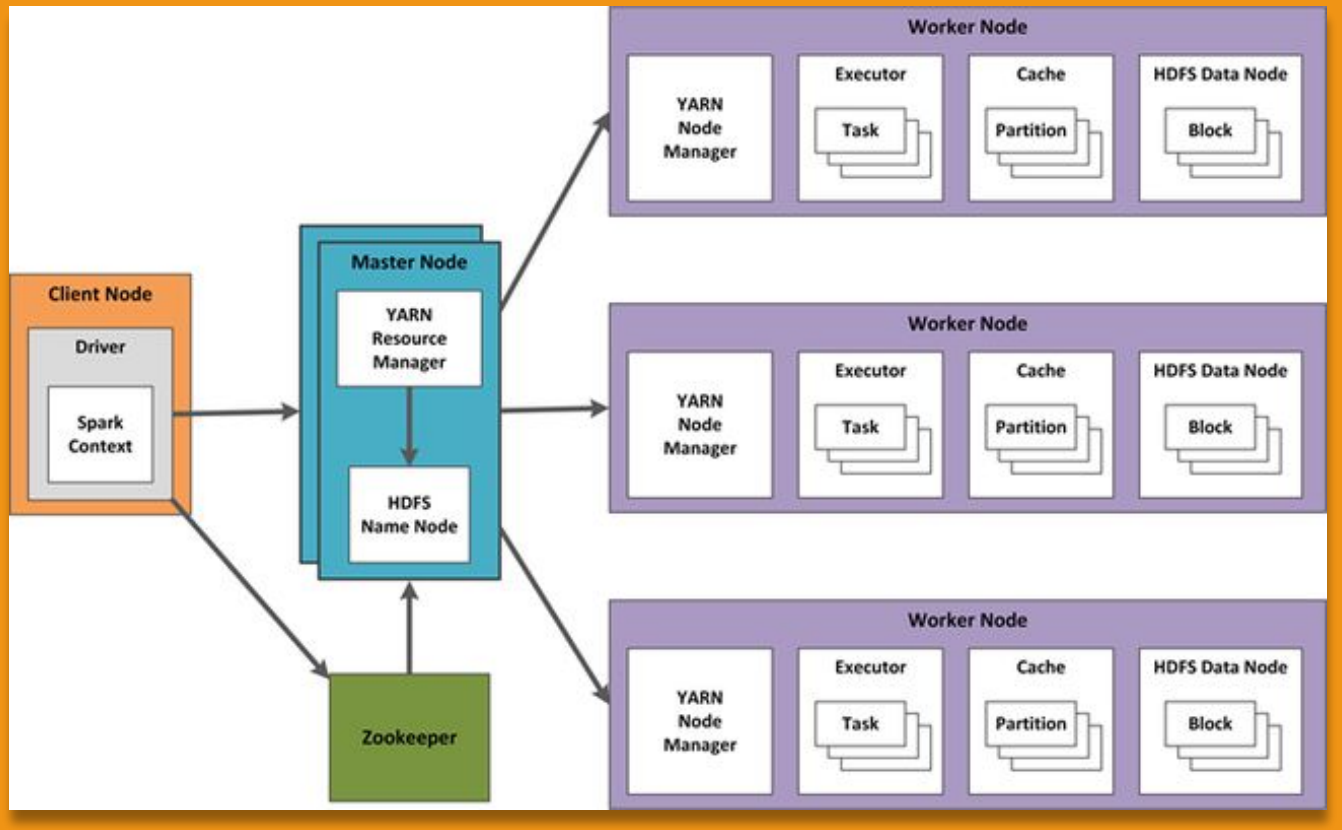
Offline vs Online systems

Count number of unique hits from India

- Consider this scenario for Facebook
- Daily total log file comes is in Petabytes
- One machine can’t save it
- Other scenarios
  - User click tracking
  - Tracking effectiveness of an Ad
- All flavours of ETL jobs

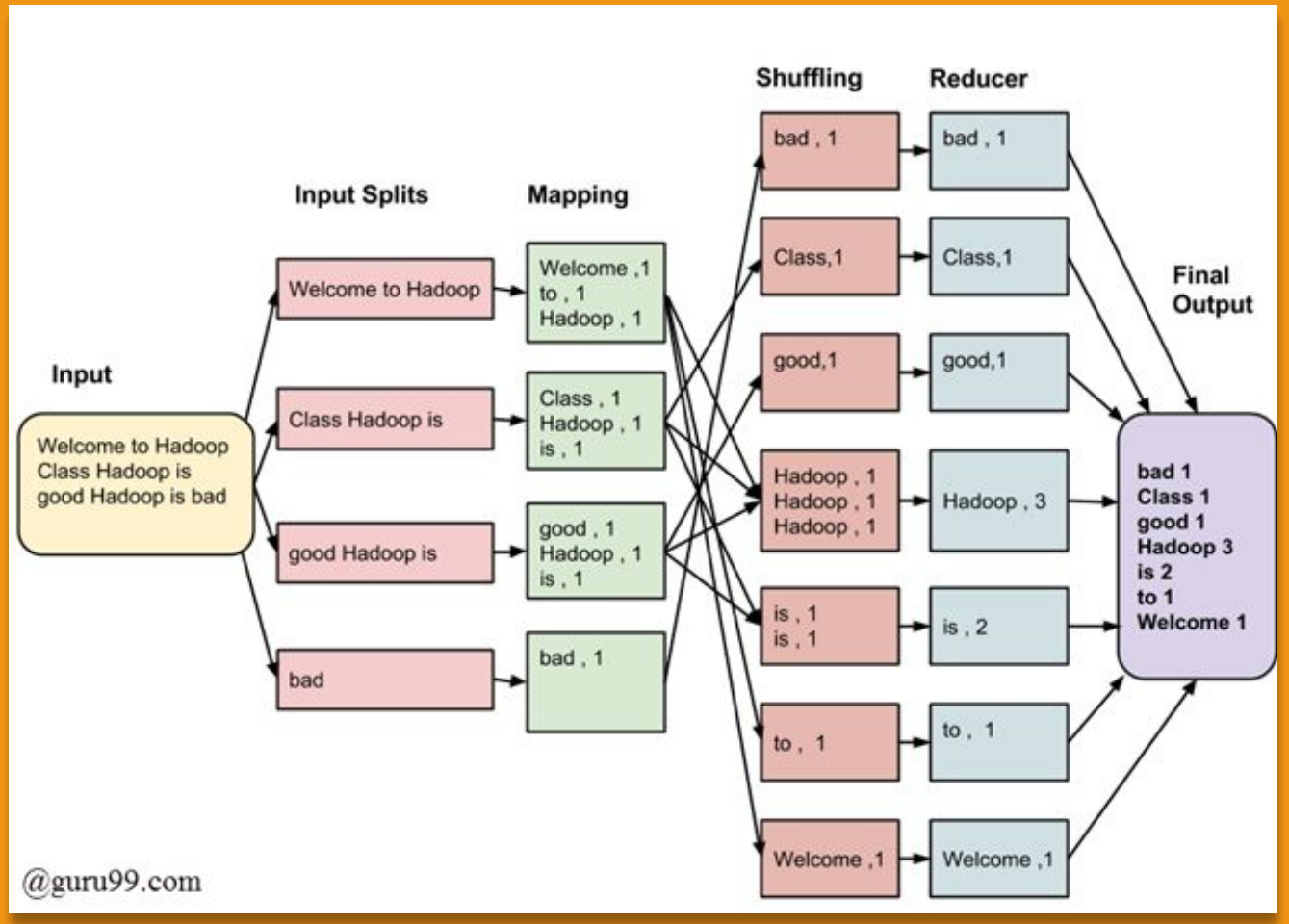
Hadoop

- Every node can store the data - HDFS
- Every node can do processing on data it holds locally.
- Easily scalable
- Redundant and fault tolerant
- Main Components
  - Name Node
  - Data Node
  - Resource Manager
  - Node Manager



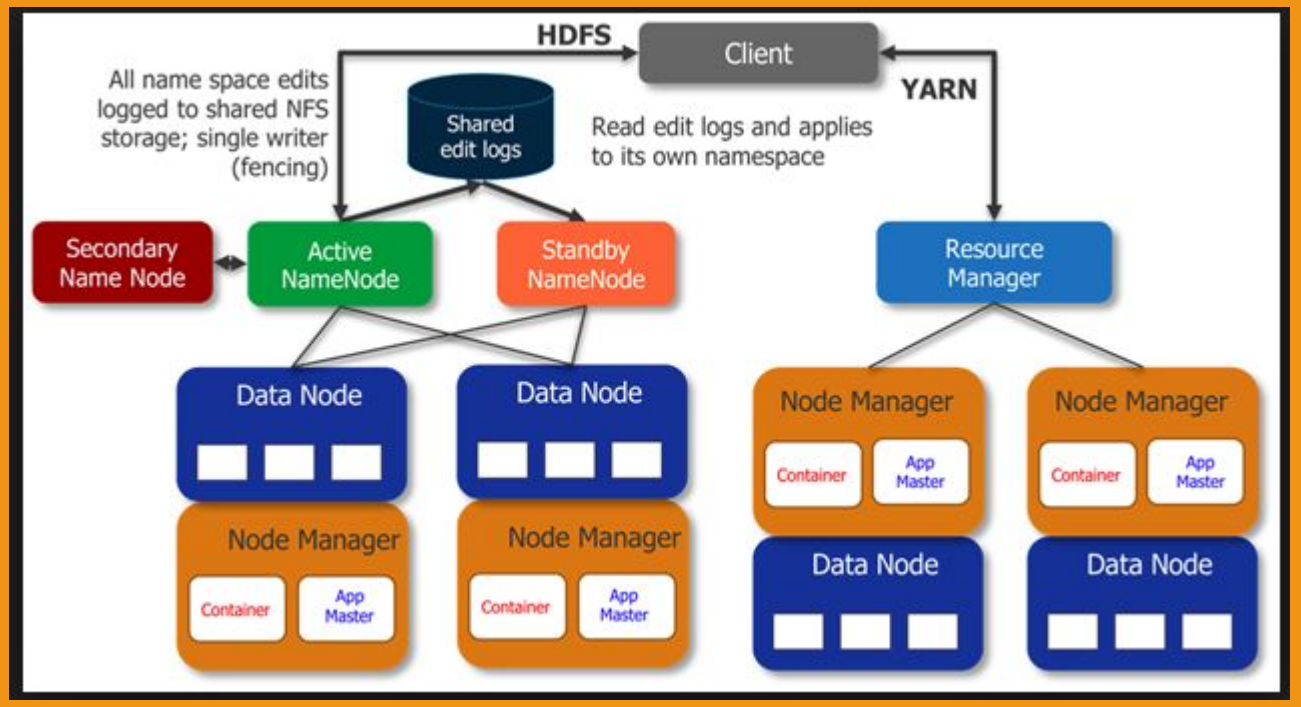
Map-Reduce compute model

- Main stages of map-reduce
- Copy code not data.
- Data locality
- How we can use map-reduce to count unique hits from a region ?



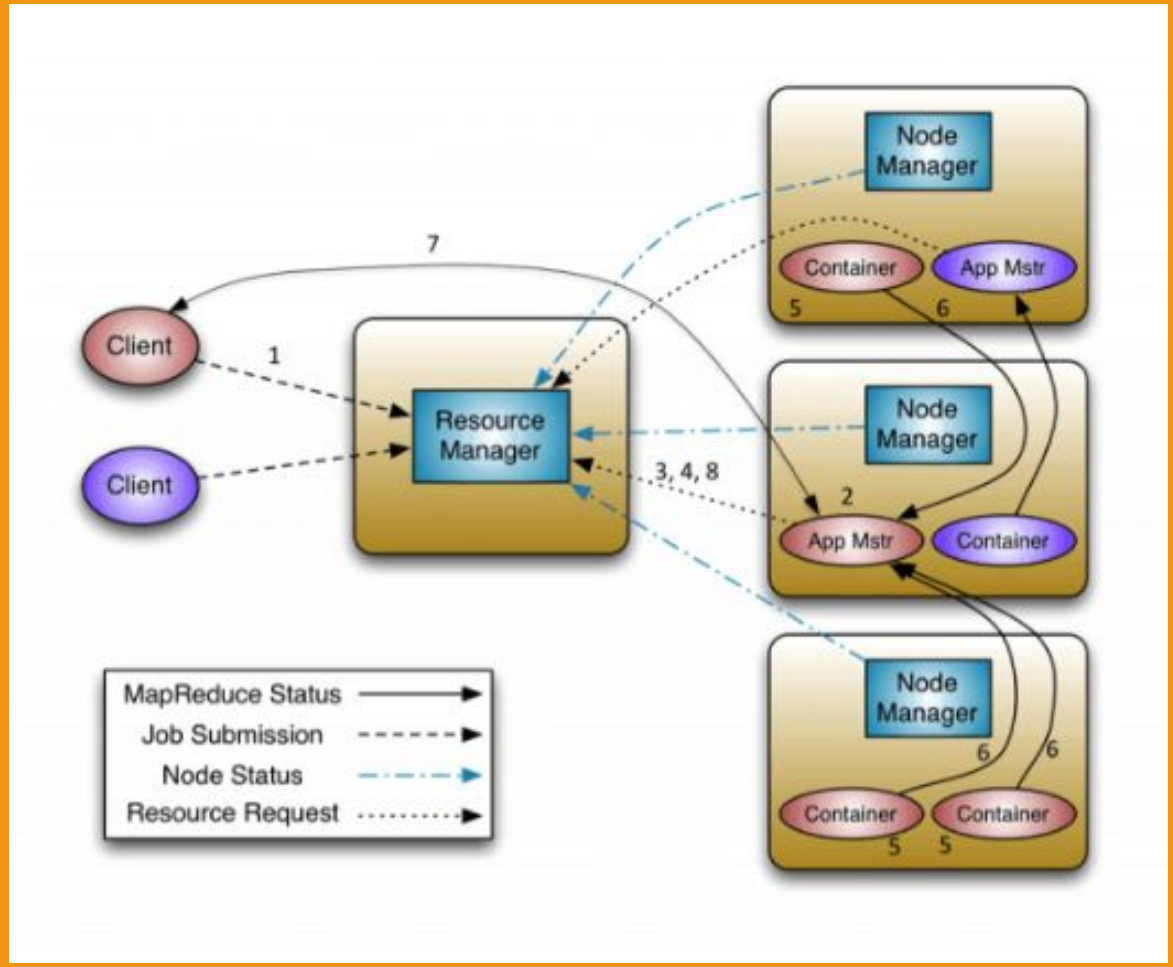
Yarn and HDFS

- Main stages of map-reduce
- Dynamic programming ?
- Bring code to data location
- Data locality
- How we can use mapreduce to count unique hits from a region ?



Yarn Framework

- Client Submits jobs into cluster
- AM handles application orchestration, once it has been started by RM
- Containers are actual resources (CPU/Ram/IO) allocated to AM.
- Job progress tracking



Hybrid environments

- Storage on cloud storages s3/azure storage
- Execution engine on our cloud or Kubernetes.

Workshop

Next: Spark on Hadoop Yarn and Pyspark

ON 13-March-2019

Thank you

Haridas N <hn@haridas.in>