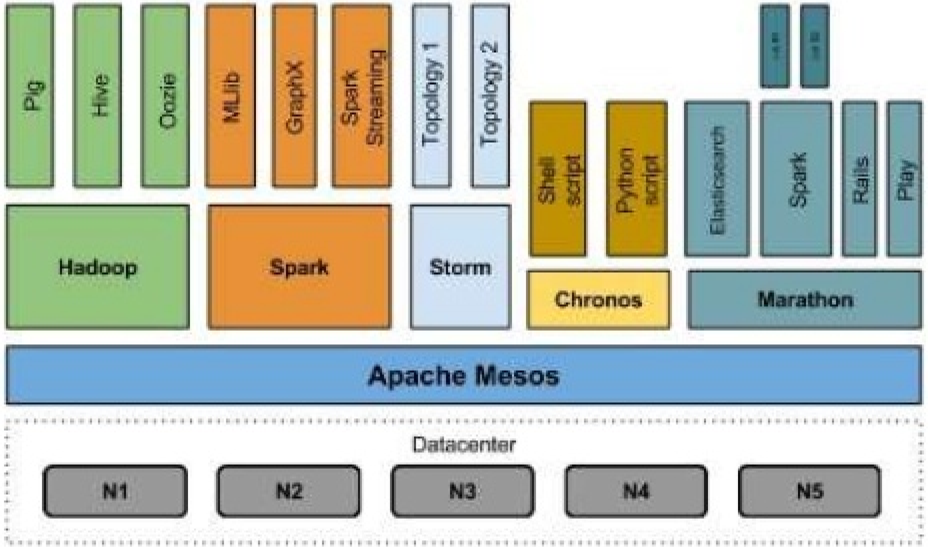
# Introduction to Apache Mesos

Omid Vahdaty, DevOps ninja

# What is Mesos?

What is Mesos?



# What is Mesos?

- is a cluster manager that provides efficient resource isolation and sharing across distributed applications or frameworks.

- Open source

- sits between the application layer and the operating system and makes it easier to deploy and manage applications in large-scale clustered environments more efficiently

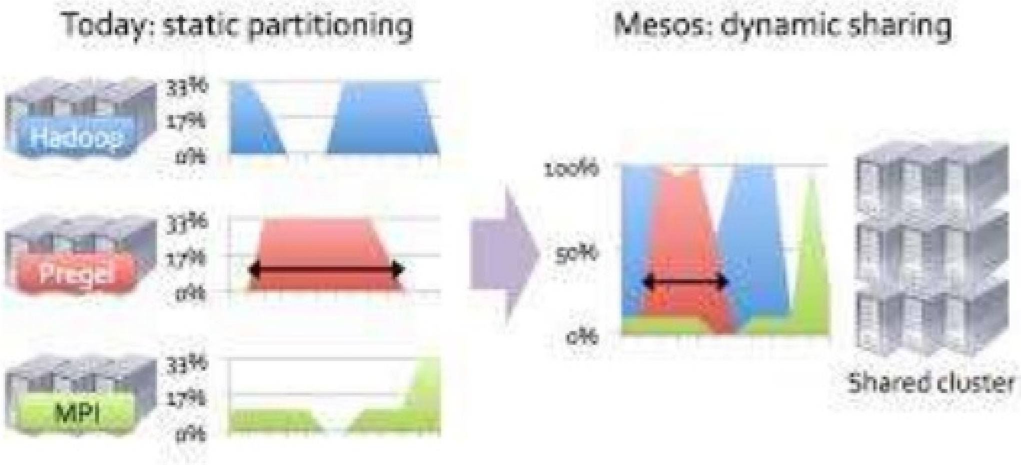
- sits between the application layer and th akes it easier to deploy and manage applications in large-scale clus fficiently

# Benefits

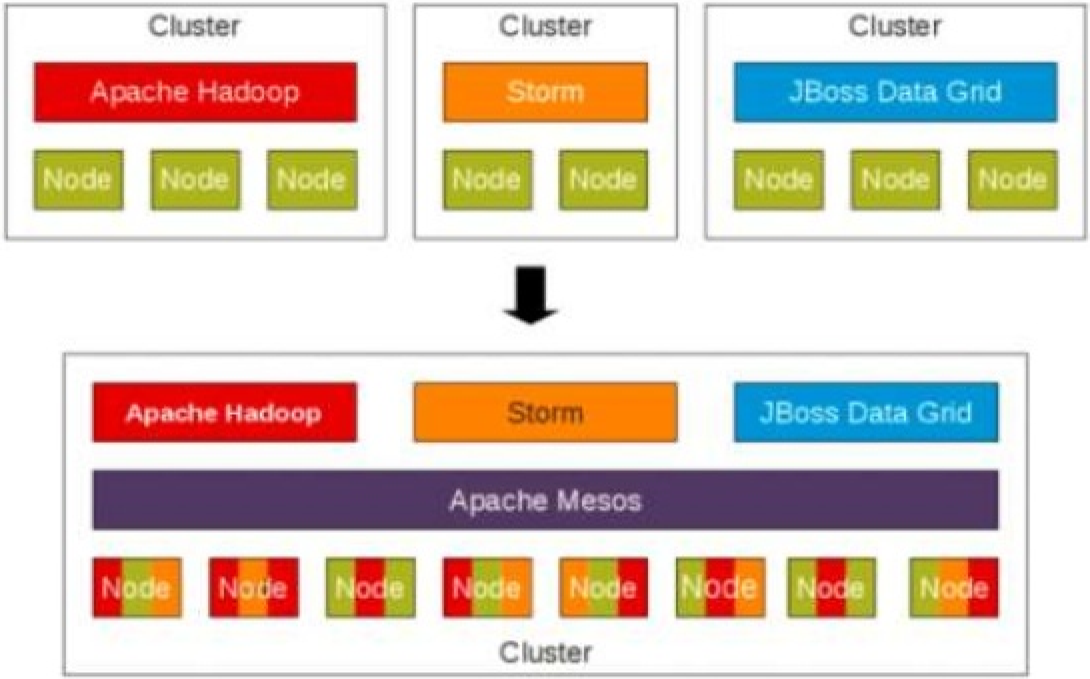
- leverages features of the modern kernel—"cgroups" in Linux
- provides isolation for CPU, memory, I/O, file system, rack locality, etc.
- introduces a distributed two-level scheduling mechanism called resource offers.
- decides how many resources to offer each framework, while frameworks decide which resources to accept and which computations to run on them
- The idea is to deploy multiple distributed systems to a shared pool of nodes in order to increase resource utilization.
- it could easily switch resources away from framework1 (for example, doing big-data analysis) and allocate them to framework2 (for example, a web server)
- is essentially data center kernel—which means it's the software that actually isolates the running workloads from each other.

# Components

- **Chronos scheduler**, a cron replacement for automatically starting and stopping services (and handling failures) that runs on top of Mesos.
- **Marathon** that provides API for starting, stopping and scaling services (and Chronos could be one of those services).

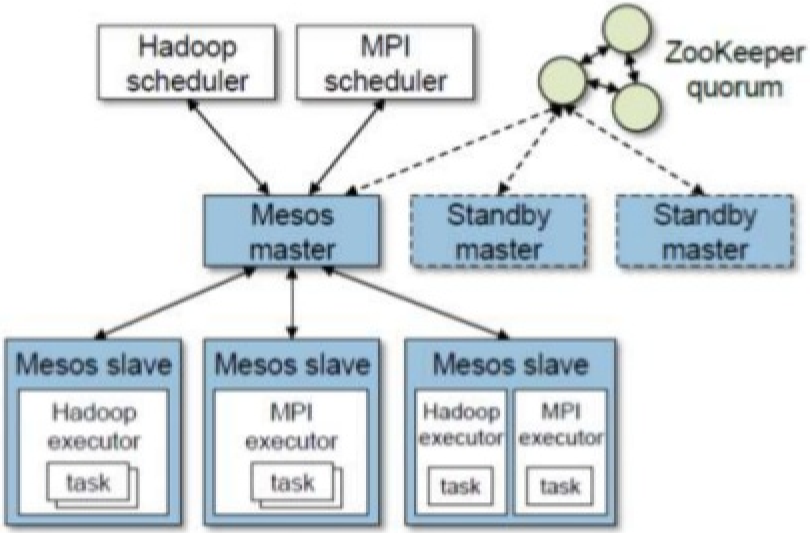


# Example


Cluster
Apache Hadoop
Node
Node
Node
Cluster
Storm
Node
Node
Cluster
JBoss Data Grid
Node
Node
Node
Apache Hadoop
Storm
JBoss Data Grid
Apache Mesos
Node
Node
Node
Node
Node
Node
Node
Node
Cluster

# Architecture

- consists of a master process that manages slave daemons running on each cluster node, and frameworks that run tasks on these slaves.
- The master implements fine-grained sharing across frameworks using resource offers.
- Each resource off
- The master decide n organizational policy, such as fair sharing or priority.



# Components

- a **scheduler** that registers with the master to be offered resources
- an **executor** process that is launched on slave nodes to run the framework's tasks.
- the master determines how many resources to offer to each framework
- the frameworks schedulers select which of the offered resources to use

# Features

Fault-tolerant replicated master using ZooKeeper

Scalability to thousands of nodes

Isolation between tasks with Linux containers

Multi-resource scheduling (memory and CPU aware)

Java, Python and C++ APIs for developing new parallel applications

Web UI for viewing cluster state

# Use Cases

- Hadoop on Mesos
- Spark on Mesos
- Storm on Mesos
- Jenkins on mesos
- Dockers on mesos
- Scaling architecture of your app
- Parallel programing (c++ API exits)