YARN

Apache Hadoop Next Generation Compute Platform

Apache Hadoop & YARN

- **Apache Hadoop**
  - De facto Big Data open source platform
  - Running for about 5 years in production at hundreds of companies like Yahoo, Ebay and Facebook
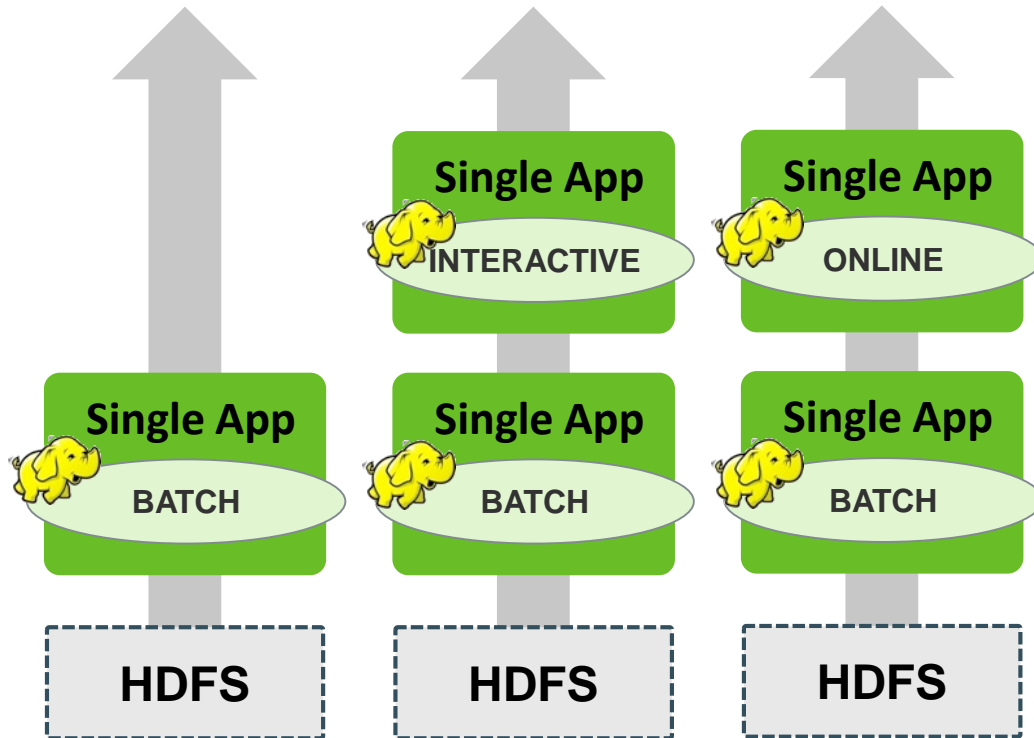
- **Hadoop 2**
  - Significant improvements in HDFS distributed storage layer. High Availability, NFS, Snapshots
  - YARN – next generation compute framework for Hadoop designed from the ground up based on experience gained from Hadoop 1
  - YARN running in production at Yahoo for about a year
  - YARN awarded Best Paper at SOCC 2013

Hortonworks

1st Generation Hadoop: Batch Focus

HADOOP 1.0
Built for Web-Scale Batch Apps

| Single App INTERACTIVE | Single App ONLINE |

| Single App BATCH | Single App BATCH | Single App BATCH |

| HDFS | HDFS | HDFS |

All other usage patterns MUST leverage same infrastructure

Forces Creation of Silos to Manage Mixed Workloads

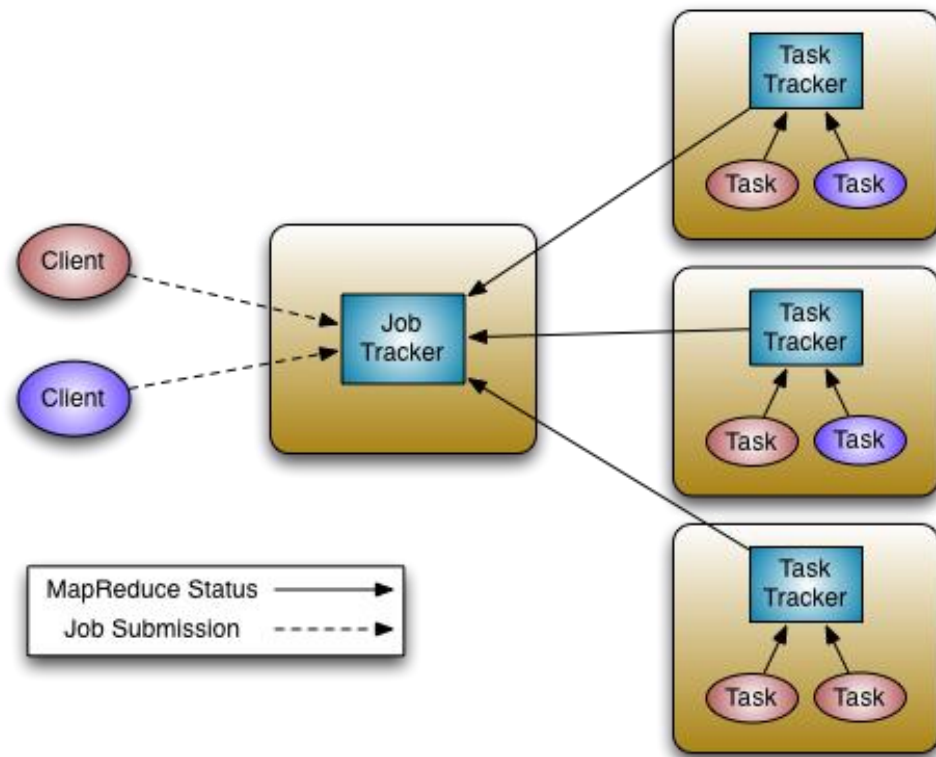Hortonworks

Hadoop 1 Architecture

JobTracker

Manage Cluster Resources & Job Scheduling

TaskTracker

Per-node agent

Manage Tasks

Hadoop 1 Limitations

Lacks Support for Alternate Paradigms and Services

Force everything needs to look like Map Reduce

Iterative applications in MapReduce are 10x slower

Scalability

Max Cluster size ~5,000 nodes

Max concurrent tasks ~40,000

Availability

Failure Kills Queued & Running Jobs

Hard partition of resources into map and reduce slots

Non-optimal Resource Utilization

Our Vision: Hadoop as Next-Gen Platform

Single Use System

Batch Apps

Multi Purpose Platform

Batch, Interactive, Online, Streaming, …

HADOOP 1.0

MapReduce

(cluster resource management
& data processing)

HDFS

(redundant, reliable storage)

HADOOP 2.0

MapReduce

(data processing)

Others

YARN

(cluster resource management)

HDFS2

(redundant, highly-available & reliable storage)

Hadoop 2 - YARN Architecture
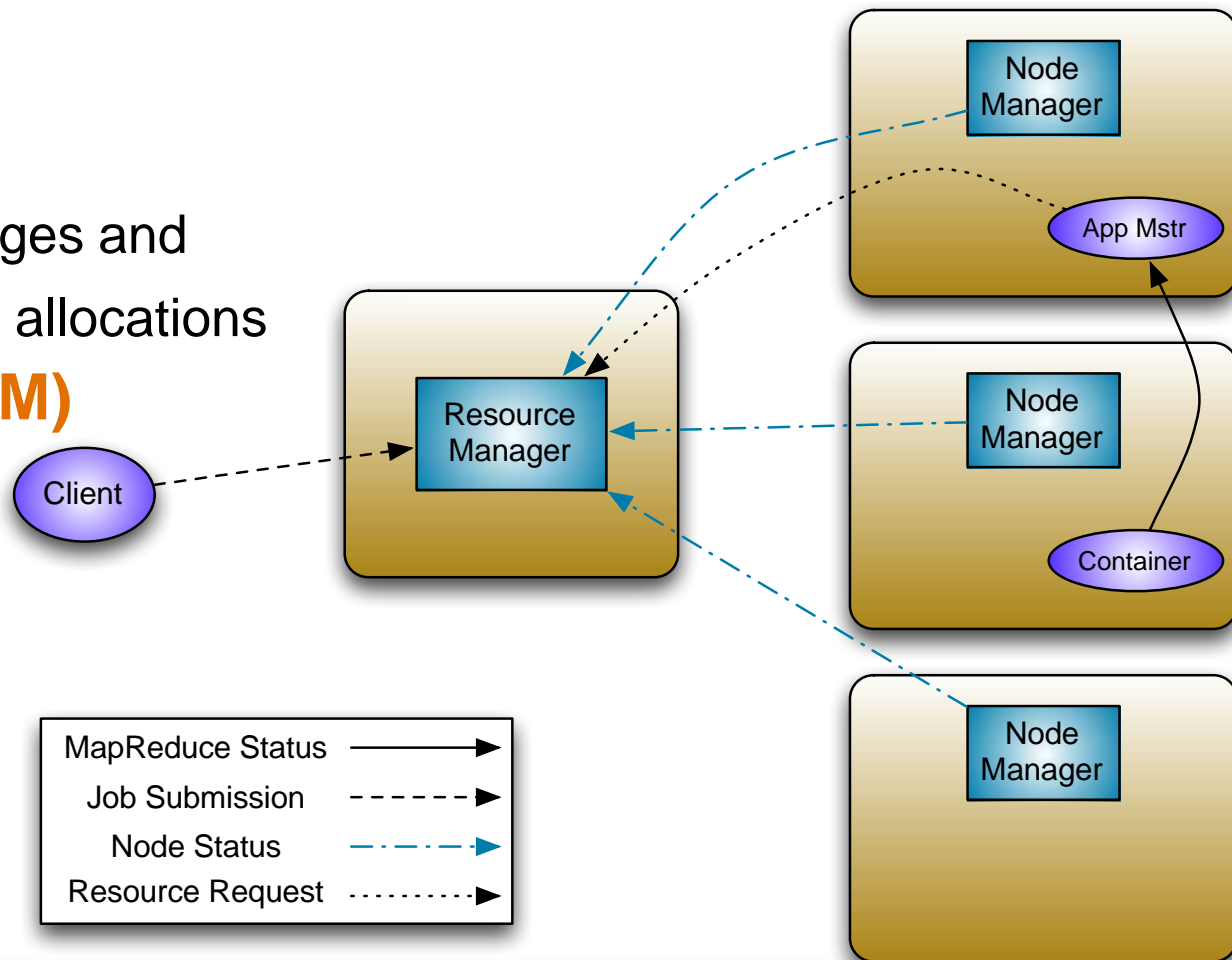
ResourceManager (RM)

Central agent - Manages and allocates cluster resources

NodeManager (NM)

Per-Node agent - Manages and enforces node resource allocations

ApplicationMaster (AM)

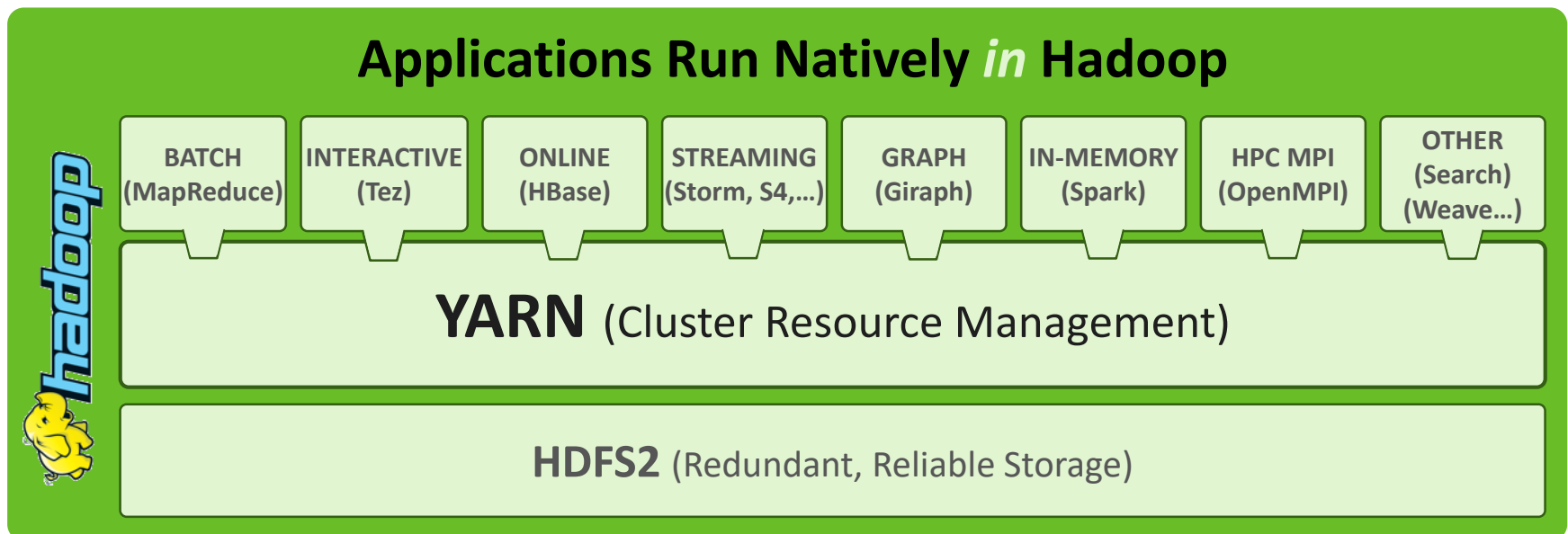Per-Application – Manages application lifecycle and task scheduling

MapReduce Status	──────▶
Job Submission	------▶
Node Status	─·─·─▶
Resource Request	·········▶

YARN: Taking Hadoop Beyond Batch

Store ALL DATA in one place…

Interact with that data in MULTIPLE WAYS

with Predictable Performance and Quality of Service

Applications Run Natively *in* Hadoop

BATCH (MapReduce)	INTERACTIVE (Tez)	ONLINE (HBase)	STREAMING (Storm, S4,…)	GRAPH (Giraph)	IN-MEMORY (Spark)	HPC MPI (OpenMPI)	OTHER (Search) (Weave…)

YARN (Cluster Resource Management)

HDFS2 (Redundant, Reliable Storage)

5 Key Benefits of YARN

1. **New Applications & Services**

2. **Improved cluster utilization**

3. **Scale**

4. **Experimental Agility**

5. **Shared Services**

Key Improvements in YARN

Framework supporting multiple applications

- Separate generic resource brokering from application logic
- Define protocols/libraries and provide a framework for custom application development
- Share same Hadoop Cluster across applications

Cluster Utilization

- Generic resource container model replaces fixed Map/Reduce slots. Container allocations based on locality, memory (CPU coming soon)
- Sharing cluster among multiple application

Key Improvements in YARN

Scalability

- – Removed complex app logic from RM, scale further
- – State machine, message passing based loosely coupled design
- – Compact scheduling protocol

Application Agility and Innovation

- – Use Protocol Buffers for RPC gives wire compatibility
- – Map Reduce becomes an application in user space unlocking safe innovation
- – Multiple versions of an app can co-exist leading to experimentation
- – Easier upgrade of framework and application

Key Improvements in YARN

Shared Services

- Common services needed to build distributed application are included in a pluggable framework

- Distributed file sharing service

- Remote data read service

- Log Aggregation Service

YARN: Efficiency with Shared Services
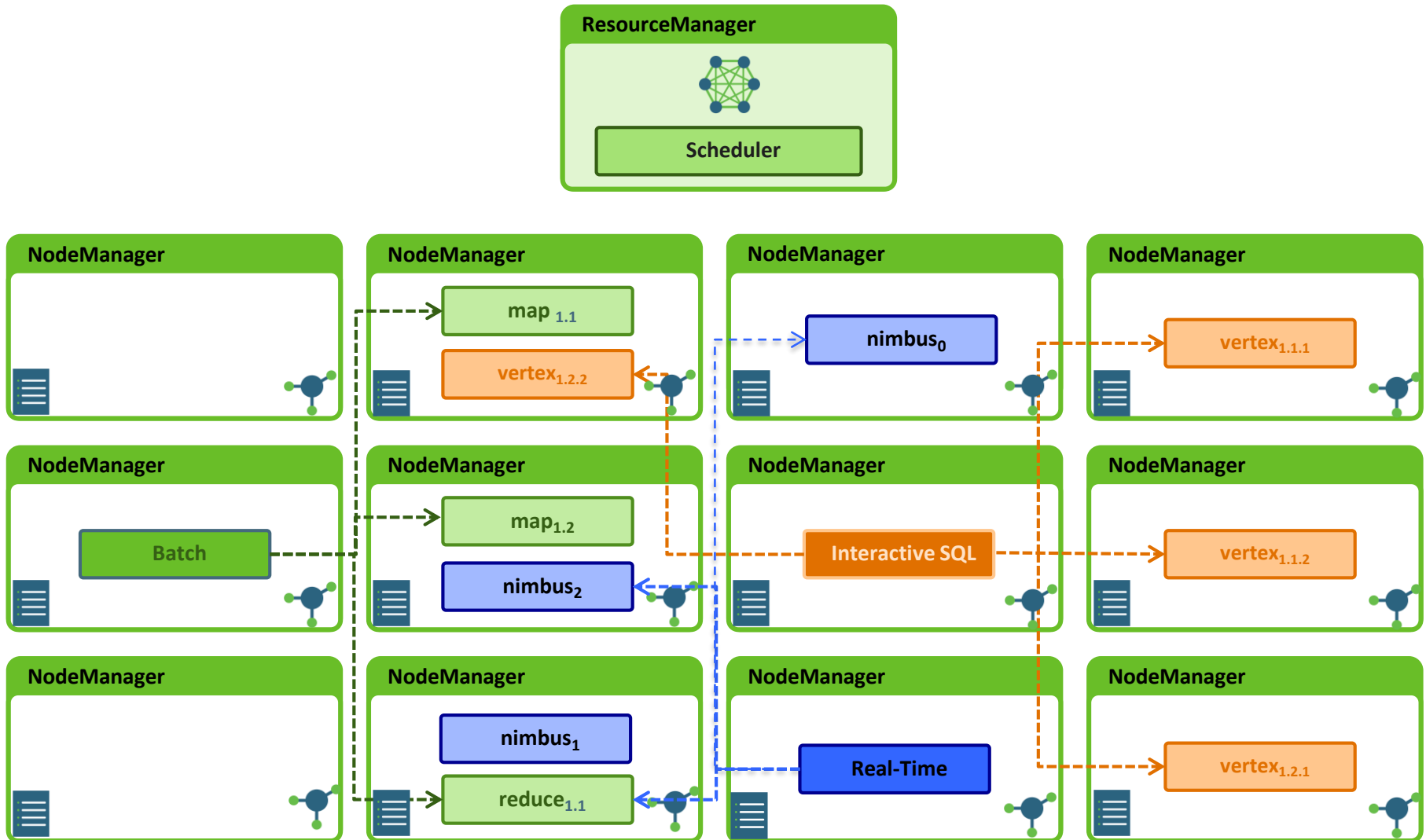

Yahoo! leverages YARN

40,000+ nodes running YARN across over 365PB of data

~400,000 jobs per day for about 10 million hours of compute time

Estimated a 60% – 150% improvement on node usage per day using YARN

Eliminated Colo (~10K nodes) due to increased utilization

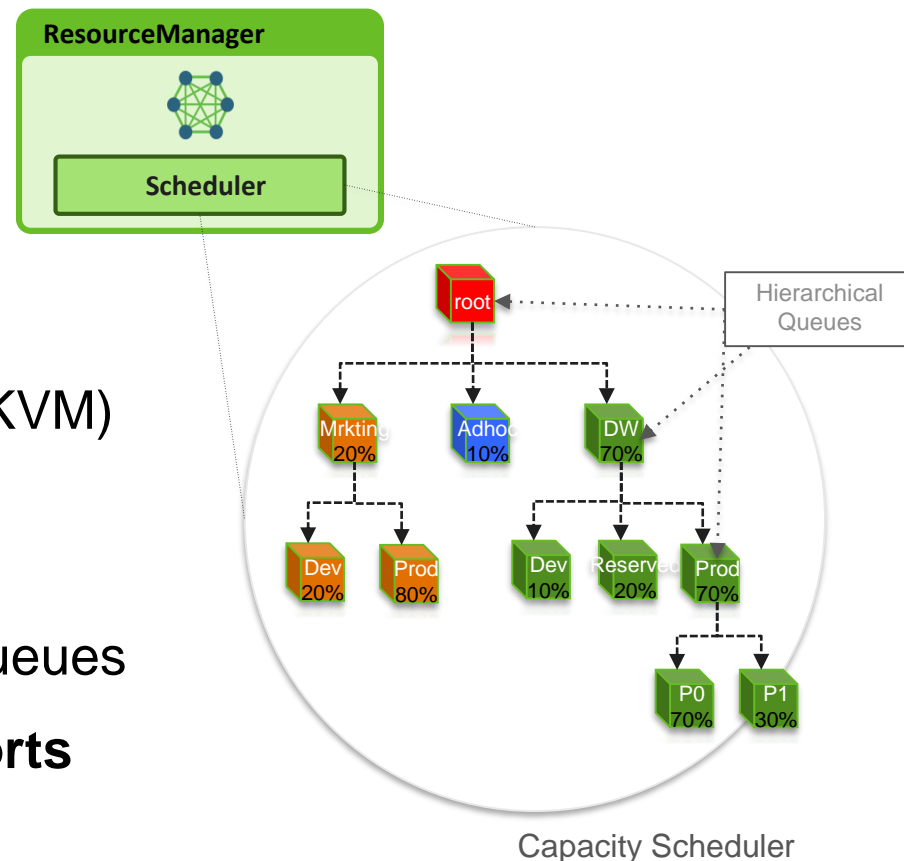For more details check out the YARN SOCC 2013 paper

YARN as Cluster Operating System

ResourceManager

Scheduler

NodeManager

NodeManager

$map_{1.1}$

$vertex_{1.2.2}$

NodeManager

$nimbus_0$

NodeManager

$vertex_{1.1.1}$

NodeManager

Batch

NodeManager

$map_{1.2}$

$nimbus_2$

NodeManager

Interactive SQL

NodeManager

$vertex_{1.1.2}$

NodeManager

NodeManager

$nimbus_1$

$reduce_{1.1}$

NodeManager

Real-Time

NodeManager

$vertex_{1.2.1}$

Multi-Tenancy is Built-in

- **Queues**

- **Economics as *queue-capacity***
  - Hierarchical Queues

- **SLAs**
  - Cooperative Preemption

- **Resource Isolation**
  - Linux: cgroups
  - Roadmap: Virtualization (Xen, KVM)

- **Administration**
  - Queue ACLs
  - Run-time re-configuration for queues

Default Capacity Scheduler supports all features

ResourceManager

Scheduler

root

Hierarchical Queues

Mrkting 20%

Adhoc 10%

DW 70%

Dev 20%

Prod 80%

Dev 10%

Reserved 20%

Prod 70%

P0 70%

P1 30%

Capacity Scheduler

YARN Eco-system

Applications Powered by YARN

Apache Giraph – Graph Processing

Apache Hama - BSP

Apache Hadoop MapReduce – Batch

Apache Tez – Batch/Interactive

Apache S4 – Stream Processing

Apache Samza – Stream Processing

Apache Storm – Stream Processing

Apache Spark – Iterative applications

Elastic Search – Scalable Search

Cloudera Llama – Impala on YARN

DataTorrent – Data Analysis

HOYA – HBase on YARN
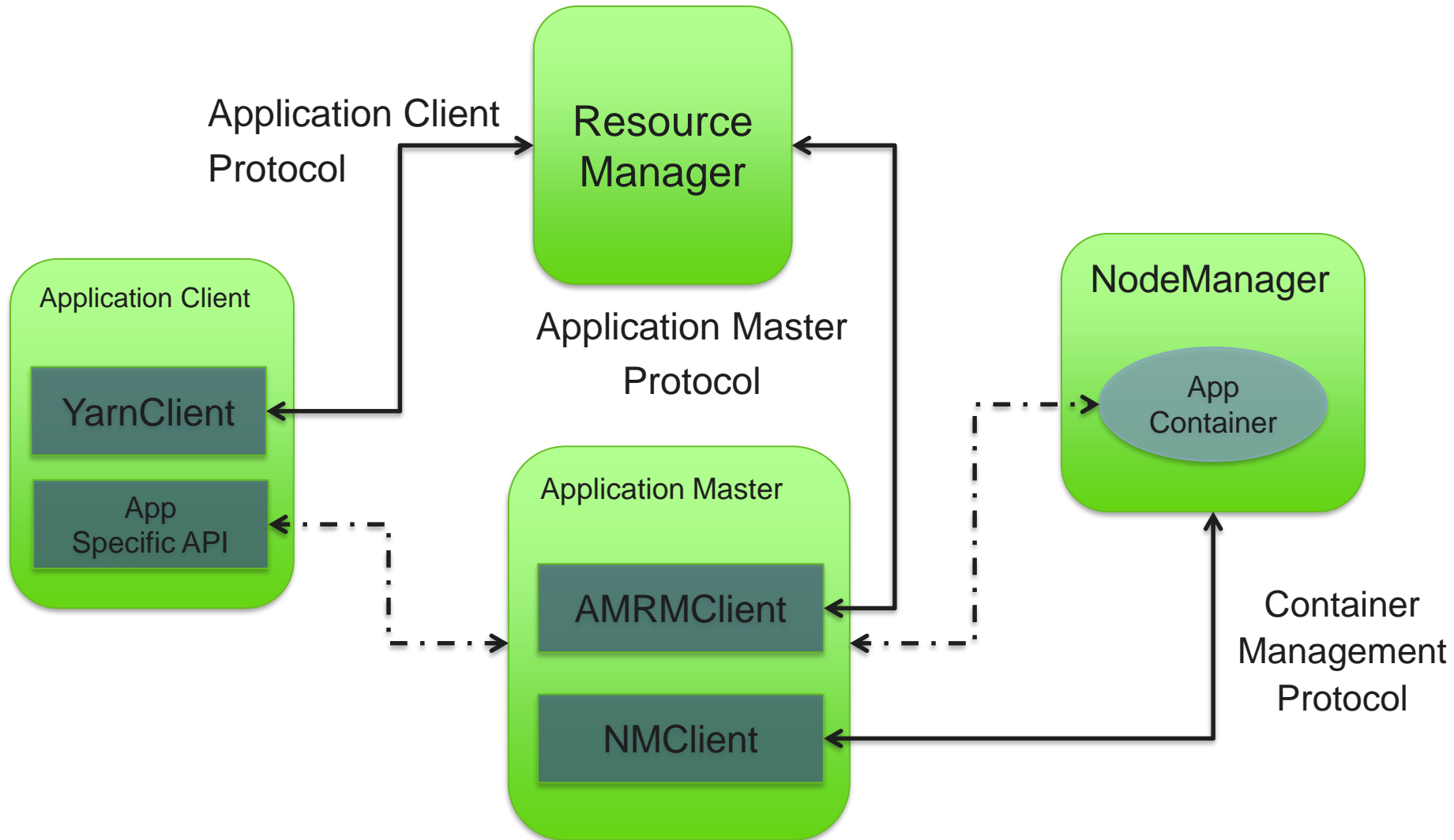
There's an app for that...
YARN App Marketplace!

Frameworks Powered By YARN

Apache Twill

REEF by Microsoft

Spring support for Hadoop 2

YARN Application Lifecycle

BYOA – Bring Your Own App

Application Client Protocol: Client to RM interaction

- Library: YarnClient
- Application Lifecycle control
- Access Cluster Information

Application Master Protocol: AM – RM interaction

- Library: AMRMClient / AMRMClientAsync
- Resource negotiation
- Heartbeat to the RM

Container Management Protocol: AM to NM interaction

- Library: NMClient/NMClientAsync
- Launching allocated containers
- Stop Running containers

Use external frameworks like Twill/REEF/Spring

YARN Future Work

- **ResourceManager High Availability**
  - Automatic failover
  - Work preserving failover
- **Scheduler Enhancements**
  - SLA Driven Scheduling, Low latency allocations
  - Multiple resource types – disk/network/GPUs/affinity
- **Rolling upgrades**
- **Generic History Service**
- **Long running services**
  - Better support to running services like HBase
  - Service Discovery
- **More utilities/libraries for Application Developers**
  - Failover/Checkpointing

Key Take-Aways

- **YARN is a platform to build/run Multiple Distributed Applications in Hadoop**

- **YARN is completely Backwards Compatible for existing MapReduce apps**

- **YARN enables Fine Grained Resource Management via Generic Resource Containers.**

- **YARN has built-in support for multi-tenancy to share cluster resources and increase cost efficiency**

- **YARN provides a cluster operating system like abstraction for a modern data architecture**

Apache YARN

The Data Operating System for Hadoop 2.0

Flexible

Enables other purpose-built data processing models beyond MapReduce (batch), such as interactive and streaming

Efficient

Increase processing **IN** Hadoop on the same hardware while providing predictable performance & quality of service

Shared

Provides a stable, reliable, secure foundation and shared operational services across multiple workloads

Data Processing Engines Run Natively IN Hadoop

BATCH	INTERACTIVE	ONLINE	STREAMING	GRAPH	MICROSOFT	SAS	OTHERS
MapReduce	*Tez*	*HBase*	*Storm, S4, …*	*Giraph*	*REEF*	*LASR, HPA*	

YARN: Cluster Resource Management

HDFS2: Redundant, Reliable Storage

Thank you!

Download Sandbox: Experience Apache Hadoop

Both 2.0 and 1.x Versions Available!

http://hortonworks.com/products/hortonworks-sandbox/

Questions?