What is Hadoop?

- Open-source software for storing enormous data sets across distributed commodity servers and then running distributed analysis applications in parallel over the data.

**Hadoop 1.x**
- MapReduce (resource mgmt, job scheduler, data processing)
- HDFS (distributed storage)

**Hadoop 2.x**
- MapReduce (data processing)
- YARN (resource mgmt, job scheduler)
- HDFS (distributed storage)
- Others (spark, mpi, giraph, etc)
Why Hadoop on Virtualized Infrastructures*?

• Scheduling and resource utilization
  - Elastic scalability on demand

• Datacenter efficiency
  - Share your computing resources

• Ease-of-Deployment
  - Lower barrier to adoption

http://wiki.apache.org/hadoop/Virtual%20Hadoop*
Hadoop on the Cloud

- Cloud Computing traditionally separates storage and computation.

Amazon Web Services

Nova (Compute)  Glance (VM Images)  Swift (Object Storage)
Hadoop is designed around Data Locality
Can Hadoop be deployed in virtual infrastructures?
Yes, But...

- Cloud hardware configurations should support data locality
- Hadoop’s original topology awareness breaks
  - Placement of >1 VM containing block replicas for the same file on the same physical host increases correlated failures
- Need VMWare’s NodeGroup aware topology
  - HADOOP-8468
Data Locality complicates Hadoop Lifecycle

- Shouldn’t upgrade Hadoop clusters by replacing VMs
  - Assuming we’re not using instances with external storage volumes, we need to upgrade software on existing instances

Create VM from pre-built image

Install/upgrade software

Configure and restart services
HOPS and HDFS
HDFS in Hadoop 1.0

Client

read /home/jd/myDna.bam

read block X

DN

DN

DN

DN

SpoF

NN

mgmt & reports
High Availability in HDFS 2.0

FailoverController
- Active
- Standby

FailoverController
- Active
- Standby

NN
- Active
- Standby

JN
- Journal Nodes

ZK
- Quorum

DN
- Data Nodes
HDFS in HOPS

Up to 48 nodes

NDB

72 million 100-byte transactional reads/sec*

Master

NN

NN

NN

DN

DN

DN

DN

*http://mikaelronstrom.blogspot.se/2012/05/mysql-cluster-72-achieves-43bn-reads.html
HOPS HDFS Features

• Increased read and write throughput

• Supports TBs of metadata (vs GBs in vanilla HDFS)

• Easy to customize metadata
  - Block-level indexing

• NDB provides online backups, geographic replication, online add-node

• **Major challenge for users:**
  Installing, configuring, and managing HOPS
HOPS has typically four different stacks*

ResourceMgr
NN
ssh, agent, chef, collectd

NodeMgr
DN
ssh, agent, chef, collectd

NDBD
MGMD
ssh, agent, chef, collectd

MYSQLD

*Configured stacks include apps, dependencies, and firewalls.
Plus a Dashboard Stack

REST APIs

Web Application

JClouds

Glassfish

ssh, chef, collectd-server
How do we deploy our PaaS?

Data Center

- NDBD
- NDBD
- MGMD

- DashB
- NN
- NN

- DN
- DN
- DN
You could bootstrap the Dashboard, old style..
Go to “EC2” in AWS web console and select “Instances”
Use the default “N. Virginia (us-east-1)” region.
Click on “Launch Instance”
On the next page, pick the “Ubuntu Server 12.04 LTS” 64-bit image.
select “Create a new Key Pair.”
click “Create and Download your key pair.”
save this file or you won’t be able to SSH into the instance we’re about to launch.

$ chmod +x cloudera-manager-installer.bin
$ sudo ./cloudera-manager-installer.bin

*http://blog.cloudera.com/blog/2013/03/how-to-create-a-cdh-cluster-on-amazon-ec2-via-cloudera-manager/*
Oh yes, I forgot it’s an Open PaaS

• By Open PaaS, we mean that we should be able to deploy our PaaS on different platforms with no code changes:
  - **Public Clouds**: AWS, Rackspace, etc
  - **Private Clouds**: OpenStack, Docker, etc
  - **Bare-Metal**

• No vendor lock-in.
Direct deployment of PaaS on AWS from a Portal

Public Cloud (AWS)

NDBD

NDBD

MGMD

Website

DashB

NN

NN

DN

DN

DN
Launch via Dashboard when few public IPs available

Private Cloud (OpenStack)

- NDBD
- NDBD
- MGMD
- DashB
- NN
- NN
- DN
- DN
- DN

HOPS Website
How do we install the software on the VMs?
The Dashboard is typically a custom VMI

- **AWS**
  - Provides APIs to automate the creation and loading of custom VM images (VMIs)*

- **OpenStack**
  - Glance is OpenStack’s HTTP-based image server (slow due to serialized access)**

*http://techblog.netflix.com/2013/03/ami-creation-with-aminator.html

**http://blog.gridcentric.com/bid/318277/Boosting-OpenStack-s-Parallel-Performance
Other nodes are configured using Chef
Chef to Install software on Nodes

Chef Recipes are infrastructure as code: idempotent & composable
But Chef and VMIs are not enough....
HOPS Open PaaS

HOPS PaaS API (YAML)

- Chef
- VMIs
- JClouds
  - Create VMs
- BitTorrent
  - Reduce Install Times
- ssh

- Amazon Web Services
- OpenStack
- Bare Metal
Define the cluster in single file
PaaS Orchestration Language

---

name: HopsTest
environment: dev
recipes: [ssh, chefClient, collectd]
authorizePorts: [8080, 8181, 3306, 4343, 3321]

provider: ~
  name: aws-ec2
  instanceType: m1.large
  login: ubuntu
  image: eu-west-1/ami-35667941
  region: eu-west-1
  zones: [eu-west-1a]

---

Define the cloud you will deploy on, and default node types

`global scope (all nodes)`
PaaS Orchestration Language

**nodes:**
- ndbd  
  number: 2
- mgmd  
  number: 1  
  recipes: [mysql]
- namenode  
  number: 2  
  repository: http://myrepo.git
- datanode  
  recipes: [nodemgr]  
  number: 3
- dashboard  
  recipes: [resourcemgr]  
  number: 1

# chefAttributes: {...}  
# authorizePorts: [...]  
# roles: [...]  
# image: ...  
# bittorrent: true

---

*Should be the name of a chef recipe*

*Explicitly add chef recipes to this node*

*Use a custom recipe for the NameNode*

*other parameters per node-group*
Language Design Decisions

- Declarative language
- Chef-friendly default behaviors
- Configure nodes by adding recipes or roles
  - Users can fork our default recipes and customize them.
  - Per node-group parameters supported.
- Language implementation can be changed to use a different configuration mgmt system, if needed.
Node provisioning can fail for a variety of reasons.

Stragglers will appear as clusters grow in size.

Low default number of parallel operations can be issued to AWS, OpenStack, etc.

Benchmark VMs to identify poorly performing ones.
Provisioning Scheduler

• Respawn & provision failed and slow VMs.

• Scheduler executes Chef recipes as a series of phases
  - Chef recipes are decomposed into the following phases:
    init, bittorrent, install, configure

• BitTorrent phase
  • Useful for larger software packages and private clouds.
  • Standard package repos used in the event of failure in the install phase.
Related Work

• Virtualization and Hadoop
  - Project Serengeti (VMWare)
  - Project Savanna (Hortonworks & OpenStack)
  - Elastic MapReduce (Amazon Web Services)

• Administration of Hadoop Clusters
  - Cloudera Manager with Puppet
  - Hortonworks Ambari with Puppet

• Open PaaS
  - Cloudify provide a general OpenPaaS middleware based on a groovy DSL, with additional support for chef.
Tack för mig!
THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:

“MY CODE’S COMPILING.”

VM’S LAUNCHING

HEY! GET BACK TO WORK!

VM LAUNCHING!

OH. CARRY ON.

[XKCD Comic 303]