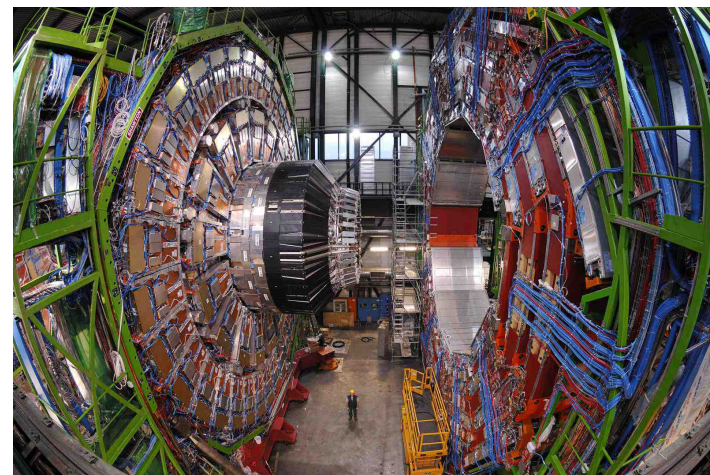




Elastic Data Processing

**Big Data Frameworks Course
Spring 2015**

Lirim Osmani





Workloads are evolving

Server vs Cloud



TRADITIONAL WORKLOADS

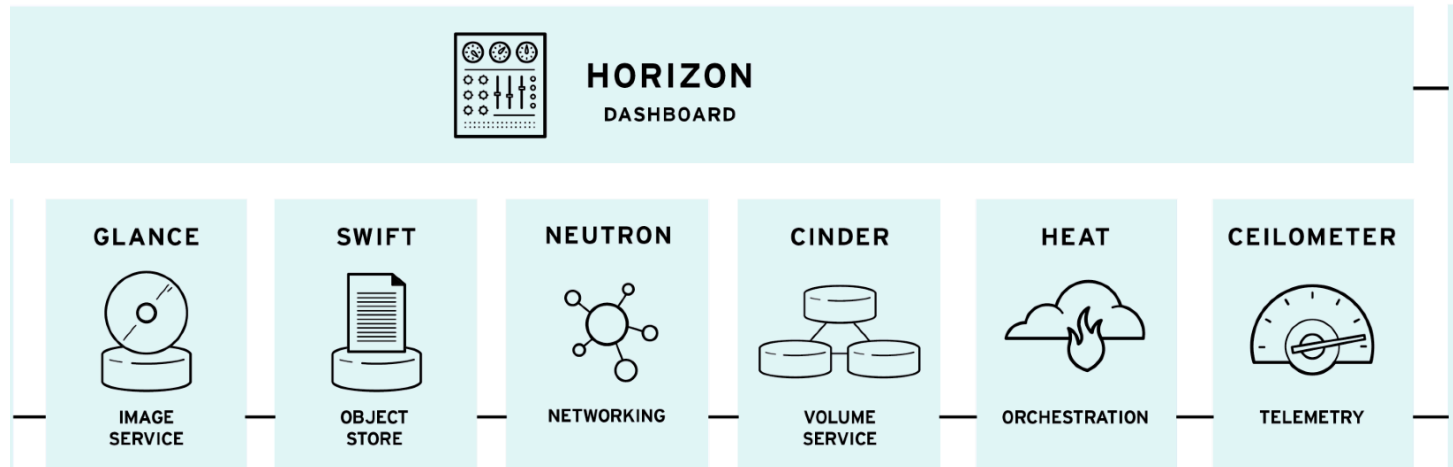
- Typically resides on a single large server machine;
- Cannot tolerate any downtime;
- Needs expensive high availability tools;
- Application scales up rather than out;

CLOUD WORKLOADS

- Workload resides on multiple Virtual Machines;
- Tolerates VM failure – if one fails, another quickly replaces it;
- Fault tolerance often build into workload ;
- Application scales out rather than up;



OpenStack Architecture



- Modular architecture;
- Designed to easy scale out;
- Based on a set of core services;



Why OpenStack?

- Brings Public Cloud like capabilities to the datacenter;
- Provides massive on-demand scalability;
 - 1,000's → 10,000's of VMs;
- It's OPEN!;
 - Provides flexibility to customize and interoperate;
- Community development = higher “feature velocity”;
 - Features and functions you need, faster to spin a proof-of-concept;
- Constantly “under development”Achilles heel;

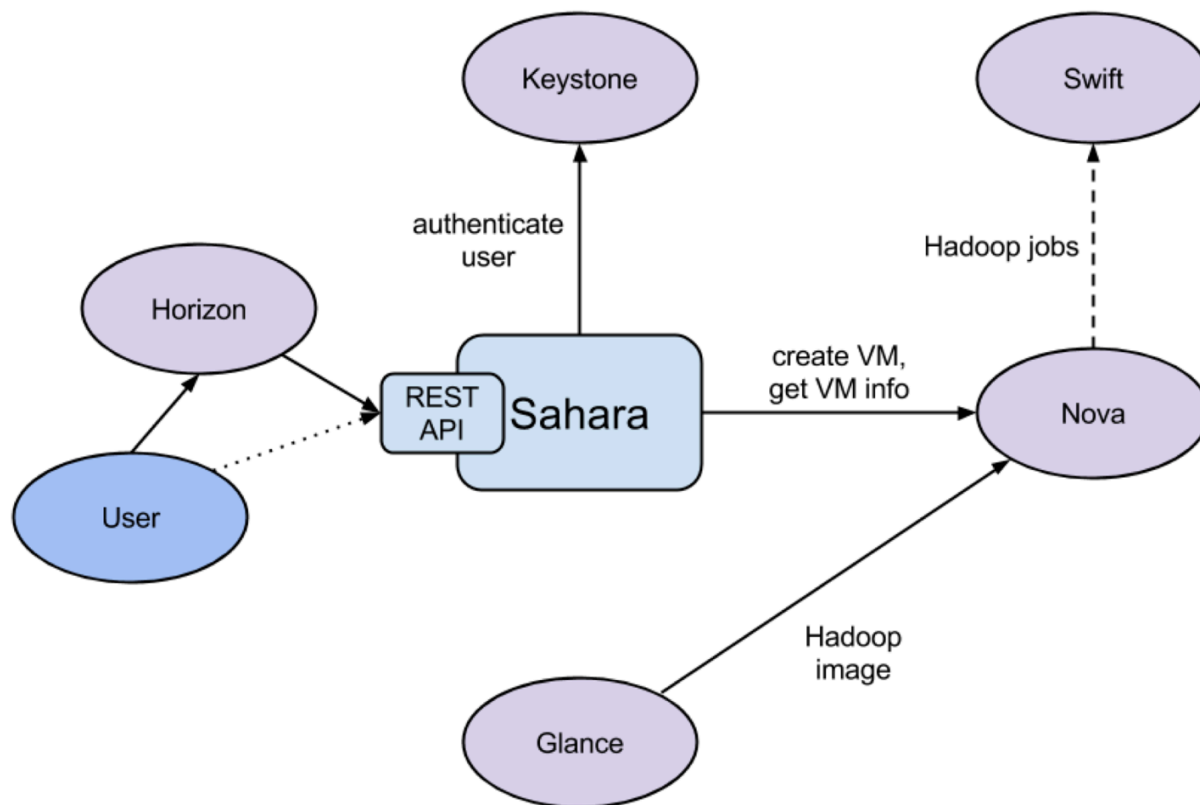


OpenStack Data Processing Sahara

- Provides a scalable data processing stack and associated management interfaces;
- Provision and operate Hadoop/Spark clusters;
- Schedule and operate Hadoop/Spark jobs;
- Self-service provisioning of clusters;
- Utilization of compute capacity for bursty workloads;
- Run big-data workloads in few clicks without expertise in cluster operations;



Sahara Integration with OpenStack





Sahara Features - Jobs

- EDP – API to execute MapReduce jobs without exposing details of underlying infrastructure (similar to AWS EMR)
 - Pluggable data sources: Swift, HDFS, Ceph
 - Supported job types: Jar, Pig, Hive
- User-friendly UI or ad-hoc analytics queries based on Hive or Pig;
- Spark jobs on Spark standalone clusters;
- Access and Storage of job binaries/output in Swift or Sahara's own database;
- Configuration of jobs at submission time;
- Execution of jobs on existing clusters;



Sahara Features – Cluster Operations

- Hadoop/Spark clusters operation and provisioning
 - Templates for Hadoop/Spark cluster configuration
 - REST API for cluster startup and operations
 - Manual cluster scaling (add/remove nodes)
 - Swift(storage) integration
- UI integrated in dashboard (Horizon)
- Plugin mechanism for integration with different Hadoop distributions – Vanilla apache, Hortonworks, Cloudera, Spark



Sahara - General workflow

- Sahara provides two level of abstractions for API and UI for: cluster provisioning and analytics-as-a-service
- For cluster provisioning:
 - Select cluster version (Hadoop,Spark);
 - Select base image with or without pre-installed software;
 - Define cluster configuration, including size and topology and setting the different type of cluster parameters (e.g.heap size);
 - Provision the cluster: Sahara will provision VMs, install and configure your cluster (Hadoop,Spark);
 - Operation of the cluster: add/remove nodes;
 - Terminate the cluster when it's not needed anymore;



Sahara - General workflow

- For analytics as a service the workflow is:
 - Select one of the predefined cluster versions;
 - Configure the job;
 - Choose type of job:pig,hive,jar-file,etc;
 - Provide the job script source or jar location;
 - Select input and output data location;
 - Select location for logs;
 - Set the limit for the cluster size;
 - Execute the job:
 - All cluster provisioning and job execution will happen transparently to the user;
 - Cluster will be removed automatically after the job completion;
 - Get the results of computation (e.g.Swift)



Sahara - General workflow – Templates

- Node Group Template:
 - Describes a group of nodes within a cluster;
 - Contains a list of Hadoop(Spark) processes that will be launched on each instance in a group;
 - Encapsulates hardware parameters (flavor) for the node VM and configuration for Hadoop(Spark) processes running on the node;
- Cluster Template:
 - Brings node group templates together to form a Cluster;
 - This template defines what Node Group will be included and how many instances will be created in each;
- The Cluster:
 - Represent the Cluster itself with pre-installed images software ready for deployment;

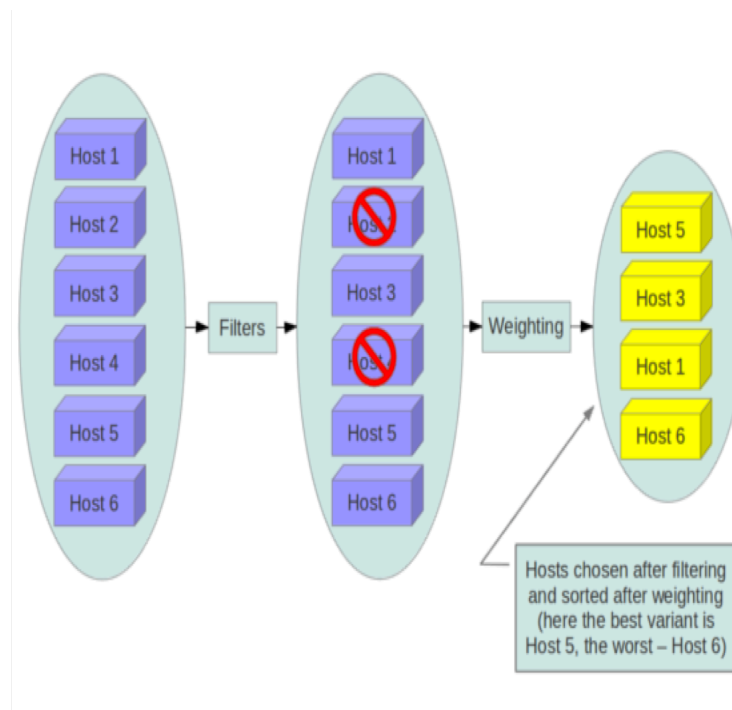


Elastic Data Processing – How?

Workload based scheduler

Flexible Scheduler Driver with filters:

- CoreFilter;
- RamFilter;
- RetryFilter;
- AvailabilityZoneFilter;
- ComputeFilter;
- ComputeCapabilitiesFilter;
- ImagePropertiesFilter;
- ServerGroupAffinityFilter;
- ServerGroupAntiAffinityFilter;





Scheduler examples

- `AvailabilityZoneFilter`:
 - return hosts where `node_availability_zone` name is the same as the one requested;
- `RamFilter`:
 - return hosts where $(\text{free ram} * \text{ram_allocation_ratio})$ is greater than requested ram;
- `ComputeFilter`: return hosts where asked `instance_type` (with extra specs) match capabilities;



Elastic Data Processing – How?

Metrics in Ceilometer

Nova

Name	Type	Unit	Resource	Origin	Note
instance	Gauge	instance	inst ID	both	Duration of instance
instance:<type>	Gauge	instance	inst ID	both	Duration of instance <type> (openstack types)
memory	Gauge	MB	inst ID	notification	Volume of RAM in MB
cpu	Cumulative	ns	inst ID	pollster	CPU time used
cpu_util	Gauge	%	inst ID	pollster	Average CPU utilisation
vcpus	Gauge	vcpu	inst ID	notification	Number of VCPUs
disk.read.requests	Cumulative	request	inst ID	pollster	Number of read requests
disk.write.requests	Cumulative	request	inst ID	pollster	Number of write requests
disk.read.bytes	Cumulative	B	inst ID	pollster	Volume of read in B
disk.write.bytes	Cumulative	B	inst ID	pollster	Volume of write in B
disk.root.size	Gauge	GB	inst ID	notification	Size of root disk in GB
disk.ephemeral.size	Gauge	GB	inst ID	notification	Size of ephemeral disk in GB
network.incoming.bytes	Cumulative	B	iface ID	pollster	number of incoming bytes on the network for a VM interface
network.outgoing.bytes	Cumulative	B	iface ID	pollster	number of outgoing bytes on the network for a VM interface
network.incoming.packets	Cumulative	packets	iface ID	pollster	number of incoming packets for a VM interface
network.outgoing.packets	Cumulative	packets	iface ID	pollster	number of outgoing packets for a VM interface



Elastic Data Processing

Cluster Templates

openstack

Project

- Compute
- Network
- Object Store
- Orchestration
- Database
- Data Processing
 - Clusters
 - Cluster Templates
 - Node Group Templates
 - Job Executions

Data Processing - Cluster Templates

Cluster Templates Upload Template + Create

<input type="checkbox"/>	Name	Plugin	Hadoop Version	Node Groups	Description
<input type="checkbox"/>	large-cluster	vanilla	2.4.1	control: 1 worker: 10	
<input type="checkbox"/>	small-cluster	vanilla	2.4.1	worker: 3 control: 1	

Displaying 2 items



Elastic Data Processing

- Support for external HDFS;

The screenshot shows a 'Create Data Source' dialog box with the following fields and options:

- Create Data Source *** (tab)
- Name: *** (text input): input data
- Data Source Type: *** (dropdown menu): HDFS
- URL: *** (text input): hdfs://namenode0/my/data
- Description:** (text input)

On the right side of the dialog, there is a grey box with instructions:

Create a Data Source with a specified name.
Select the type of your Data Source.
You may need to enter the username and password for your Data Source.
You may also enter an optional description for your Data Source.

At the bottom right, there are two buttons: 'Cancel' and 'Create'.



Elastic Data Processing

- Job Type selection;

The screenshot shows a 'Create Job' dialog box with a close button (X) in the top right corner. It has two tabs: 'Create Job' (active) and 'Libs'. The 'Create Job' tab contains the following fields:

- Name:** A text input field.
- Job Type:** A dropdown menu with 'Streaming MapReduce' selected.
- Choose a main binary:** A dropdown menu with '-- not selected --'.
- Description:** A text input field.

On the right side of the dialog, there is a help panel with the following text:

Create a Job with a specified name.
Select the type of your Job:

- Pig
- Hive
- MapReduce
- Java Action

Choose or create your main binary. Additional libraries can be added from the "Libs" tab.

For MapReduce or Java Action jobs, "mains" are not applicable. You are required to add one or more "libs" for these jobs.

You may also enter an optional description for your Job.

At the bottom right of the dialog are 'Cancel' and 'Create' buttons.




Elastic Data Processing

Job re-launch, with new data and parameters

Sahara - Job Executions

Job Executions

 Delete Job executions

<input type="checkbox"/>	ID	Status	Actions
<input type="checkbox"/>	e3a0f9b6-cea2-4f37-b8f3-fd8491cba291	SUCCEEDED	<div><div>Delete Job execution</div><div>More ▾</div><div>Relaunch On Existing Cluster</div><div>Relaunch On New Cluster</div></div>

Displaying 1 item



EDP – Command line interface

Image Management

```
$ sahara
```

```
...
```

```
Positional arguments:
```

```
<subcommand>
```

```
image-add-tag    Add a tag to an image.  
image-list       Print a list of available images.  
image-register   Register an image from the Image index.  
image-remove-tag Remove a tag from an image.  
image-show       Show details of an image.  
image-unregister Unregister an image.
```



EDP – Command line interface

Node group, cluster and job templates

```
$ sahara
  node-group-template-create    Create a node group...
  node-group-template-delete    Delete a node group...
  node-group-template-list      Print a list of available...
  node-group-template-show      Show details of a node...
  cluster-template-create       Create a cluster template.
  cluster-template-delete       Delete a cluster template.
  cluster-template-list         Print a list of available...
  cluster-template-show         Show details of a cluster...
  job-template-create           Create a job template.
  job-template-delete           Delete a job template.
  job-template-list             Print a list of job...
  job-template-show             Show details of a job...
```



EDP – Command line interface

Data sources and job binaries

```
$ sahara
```

```
...
```

```
<subcommand>
```

data-source-create	Create a data source that provides job input receives job output.
data-source-delete	Delete a data source.
data-source-list	Print a list of available data...
data-source-show	Show details of a data source.
job-binary-create	Record a job binary.
job-binary-delete	Delete a job binary.
job-binary-list	Print a list of job binaries.
job-binary-show	Show details of a job binary.



EDP - Command line interface

Clusters and jobs

```
$ sahara
...
<subcommand>
  cluster-create  Create a cluster.
  cluster-delete  Delete a cluster.
  cluster-list    Print a list of available clusters.
  cluster-show    Show details of a cluster.
  job-create
  job-delete      Delete a job.
  job-list        Print a list of jobs.
  job-show        Show details of a job.
```



Thank you.