Benchmarking Sahara-based Big-Data-as-a-Service Solutions

Zhidong Yu, Weiting Chen (Intel)
Matthew Farrellee (Red Hat)
May 2015
Agenda

- Why Sahara
  - Sahara introduction
  - Deployment considerations
  - Performance testing and results
  - Future envisioning
- Summary and Call to Action
Why Sahara: Cloud features

- You or someone at your company is using AWS, Azure, or Google
- You’re probably doing it for easy access to OS instances, but also the modern application features, e.g. AWS’ EMR or RDS or Storage
- [expecting anyone to choose openstack infra for their workloads means providing app level services, e.g. Sahara & Trove]
- [app writers apps are complex enough without having to manage the supporting infra. examples outside cloud in mobile (feedhenry, parse, kinvey)]
Why Sahara: Data analysis

- [all this true for database provisioning, and that’s a known quantity]
- [all this especially true for data processing, which many developers are only recently (compared to RDBMS) integrating into their applications]
- [data processing workflow, show huge etl effort, typical workflow does not even take into account infra to run it]
- [flow into key features of sahara]
Agenda

- Why Sahara
- Sahara introduction
- Deployment considerations
- Performance testing and results
- Future envisioning
- Summary and Call to Action
Sahara features

- Repeatable cluster provisioning and management operations
- Data processing workflows (EDP)
- Cluster scaling (elasticity), Storage integration (Swift, Cinder, HCFS)
- Network and security group (firewall) integration
- Service anti-affinity (fault domains & efficiency)
Sahara architecture
Sahara plugins

- Users get choice of integrated data processing engines
- Vendors get a way to integrate with OpenStack and access users
- Upstream - Apache Hadoop (Vanilla), Hortonworks, Cloudera, MapR, Apache Spark, Apache Storm
- Downstream - depends on your OpenStack vendor
Agenda

- Why Sahara
- Sahara introduction
- Deployment considerations
- Performance testing and results
- Future envisioning
- Summary and Call to Action
Storage Architecture

- Tenant provisioned (in VM)
  - HDFS in the same VMs of computing tasks vs. in the different VMs
  - Ephemeral disk vs. Cinder volume

- Admin provided
  - Logically disaggregated from computing tasks
  - Physical collocation is a matter of deployment
  - For network remote storage, Neutron DVR is very useful feature

- A disaggregated (and centralized) storage system has significant values
  - No data silos, more business opportunities
  - Could leverage Manila service
  - Allow to create advanced solutions (.e.g. in-memory overlayer)
  - More vendor specific optimization opportunities

Scenario #1: computing and data service collocate in the VMs
Scenario #2: data service locates in the host world
Scenario #3: data service locates in a separate VM world
Scenario #4: data service locates in the remote network
## Compute Engine

<table>
<thead>
<tr>
<th></th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>Best support in OpenStack</td>
<td>• Slow to provision</td>
</tr>
<tr>
<td></td>
<td>Strong security</td>
<td>Relatively high runtime performance overhead</td>
</tr>
<tr>
<td>Container</td>
<td>• Light-weight, fast provisioning</td>
<td>Nova-docker readiness</td>
</tr>
<tr>
<td></td>
<td>• Better runtime performance than VM</td>
<td>Cinder volume support is not ready yet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Weaker security than VM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not the ideal way to use container</td>
</tr>
<tr>
<td>Bare-Metal</td>
<td>• Best performance and QoS</td>
<td>• Ironic readiness</td>
</tr>
<tr>
<td></td>
<td>• Best security isolation</td>
<td>Worst efficiency (e.g. consolidation of workloads with different behaviors)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Worst flexibility (e.g. migration)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Worst elasticity due to slow provisioning</td>
</tr>
</tbody>
</table>

- Container seems to be promising but still need better support
- Determining the appropriate cluster size is always a challenge to tenants
  - e.g. small flavor with more nodes or large flavor with less nodes
Data Processing API

- **Direct Cluster Operations**
  - Sahara is used as a provisioning engine
  - Tenants expect to have direct access to the virtual cluster
    - e.g. directly SSH into the VMs
  - May use whatever APIs comes with the distro
    - e.g. Oozie

- **EDP approach**
  - Sahara’s EDP is designed to be an abstraction layer for tenants to consume the services
    - Ideally should be vendor neutral and plugin agnostic
  - Limited job types are supported at present

- **3rd party abstraction APIs**
  - Not supported yet
  - e.g. Cask CDAP
Deployment Considerations Matrix

- **Data Processing API**
  - Legacy (Sahara native)
  - 3rd party APIs
  - Vanilla, Spark, Storm, CDH, HDP, MapR

- **Distro/Plugin**
  - VM, Container, Bare-metal

- **Compute**
  - Tenant vs. Admin provisioned
  - Disaggregated vs. Collocated

- **Storage**
  - Performance results in the next section
  - HDFS vs. other options
Agenda

- Why Sahara
- Sahara introduction
- Deployment considerations
- Performance testing and results
- Future envisioning
- Summary and Call to Action
Testing Environment

- Host OS: CentOS7
- Guest OS: CentOS7
- Hadoop 2.6.0
- 4-Nodes Cluster
- Baremetal
  - OpenStack using KVM
    - qemu-kvm v1.5
  - OpenStack using Docker
    - Docker v1.6
Ephemeral Disk Performance

- Two factors bring ??% overhead
  - access pattern change accounts for 10%
  - virtualization overhead accounts for ??%

HDFS over RAID5 brings extra 10% performance overhead

Run Time (sec) vs. TO BE UPDATED

Lower is better
Collocated HDFS Performance

- To be added.

TO BE Replaced by real data
Swift Performance

- To be added.

TO BE Replaced by real data
Bare-metal vs. Container vs. VM

- Docker provides similar disk write results with KVM
  - ~15% performance loss for both KVM and Docker
- Docker uses less resources than KVM

TO BE UPDATED
Agenda

- Why Sahara
- Sahara introduction
- Deployment considerations
- Performance testing and results
- Future envisioning
- Summary and Call to Action
Future of Sahara

- Architected for disaggregated computing and storage
  - Supporting more storage backend
  - Integration with Manila
- Better support for container and bare-metal (Nova-docker and Ironic)
  - Murano as an alternative?
- EDP as a PaaS like layer for Sahara core provisioning engine
  - Data connector abstraction
  - Binary/job management
  - Workflow management
  - Policy engine for resource and SLA management
  - Auto-scale, auto-tune
- Sahara needs to be open to all the vendors in the big data ecosystem
  - A complete big data stack may have many options at each layer
  - e.g. acceleration libraries, analytics, developer oriented application framework (e.g. CDAP)
  - Requires a more generic plugin/driver framework to support it
Summary and Call-to-Action

- Great improvement in Sahara Kilo release. Production ready with real customer deployments.
- A complete Big-Data-as-a-Service solution requires more considerations than simply adding a Sahara service to the existing OpenStack deployment.
- Preliminary benchmark results show ....
- Many features could be added to enhance Sahara. Opportunities exist for various types of vendors.

Join in the Sahara community and make it even more vibrant!
BACKUP
## DD Testing Result

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host with Multiple Disks</td>
<td>100 MB/s x 4 = 400 MB/s</td>
</tr>
<tr>
<td>Container with Multiple Disks</td>
<td>90 MB/s x 4 = 360 MB/s</td>
</tr>
<tr>
<td>Host in RAID5</td>
<td>320 MB/s</td>
</tr>
<tr>
<td>VM in RAID5</td>
<td>270 MB/s</td>
</tr>
<tr>
<td>Container in RAID5</td>
<td>270 MB/s</td>
</tr>
</tbody>
</table>

Multiple Disks Configuration: 1 x 1TB SATA HDD for System, 4 x 1TB SATA HDDs for Data, RAID 5 Configuration: 5 x 1TB SATA HDD

`dd` command: `dd if=/dev/zero of=/mnt/test1 bs=1M count=8192 (4096, 8192, 16384, 24576) conv=fdatasync`
Sort IO Profiling

2-phases in Sort running period for disk write
- Shuffle Map-Reduce Data -> Use temp folder to store intermediate data (40% total throughput)
- Write Output -> HDFS Write (60% total throughput)
Storage Suggestion in Computing Node

- Dedicate storage disks to spread disk io for better performance
- A system disk for operating system
- Several data disks for tmp folder and HDFS

For operating system, it can be used to allocate boot, root, and swap partition in this disk. RAID is also available in this disk for better failover.

For intermediate data, it is used to assign a volume for intermediate data in mapreduce configuration.

For HDFS, it is used to process HDFS. It can also be replaced with any kind of external storage like swift.
Storage Strategy in Sahara

Transient Cluster
Use external storage (Swift, External HDFS) to persist data.
Pros
- persist data in external storage
- terminate computing node after finishing the jobs
Cons
- lost performance from external storage

Long Run Cluster
Use ephemeral storage/cinder storage for better performance.
Pros
- better performance using internal storage
Cons
- may still need backup from external storage
DVR enhance network performance

- Additional external network in computing node
- Using iPerf and 1Gb in internal and external network
- DVR provide better performance from Instance to Host

Instance communicates with different hosts. The scenario usually uses in control path from host to instances.

Instances communicate between hosts. The scenario usually uses in Internal HDFS data communication.

Instance communicates with the same host. The scenario usually uses in control path from host to instances. Put datanode in the host with data locality may bring a new concept for data persistent

Two instances in the same host. Internal HDFS data communication.
Disclaimer

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Include as footnote in appropriate slide where performance data is shown:

- § Configurations: [describe config + what test used + who did testing]
- § For more information go to http://www.intel.com/performance.

Intel, the Intel logo, {List the Intel trademarks in your document} are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

© 2015 Intel Corporation.