May 5, 2016, Frascati

Resource Virtualization

WP4.1 Computing Virtualisation
Dr. Alvaro Lopez Garcia
aloga@ifca.unican.es
Spanish National Research Council – CSIC

WP4.2 Storage Virtualisation
Marcus Hardt, KIT
Patrick Fuhrmann, DESY

WP4.3 Network Virtualisation
Zdenek Sustr, CESNET
Patrick Fuhrmann, DESY
WP4.1 Update on Computing Virtualisation

Alvaro Lopez Garcia

aloga@ifca.unican.es
Spanish National Research Council - CSIC
WP 4.1.1 Container support.
- OpenStack and OpenNebula: **done**.
- CNRS working on packaging for OpenStack.

WP 4.1.2 Repository Synchronization.
- New roadmap available, first implementation ready by end of May.

WP 4.1.3 OCCI support.
- Support for containers: **done**
- Support for pre-emptible instances: **in progress**. (will be in first R.)
WP 4.1.4 Improved Scheduling.

- Synergy: (Fair Share for Open Stack) in progress,
  - testbed available at INFN-PD
  - adapting to Liberty and preparing the release.
- Fair Share Scheduler (ONE):
  - In progress, targeting first version with limited functionality for the release.
- Partition Director: first implementation done, working on QA.
- Preemptible instances: first implementation done, testing in progress, testbed available at CSIC, working on QA.
- DEMO: preemptible instances: [https://www.youtube.com/watch?v=c0f0fag3RGE](https://www.youtube.com/watch?v=c0f0fag3RGE)
WP 4.1.4 Improved Scheduling.

DEMO: preemptible instances: https://www.youtube.com/watch?v=c0f0fag3RGE
WP 4.1.5 Docker in batch systems.
- Running docker in userspace: done, working in QA. (e.g. testing)

WP 4.1.6 Tosca Orchestration.
- OpenNebula support of custom types (IM): done.
- OpenStack support of custom types: in progress
- Same demo: min 4:39

WP4.X IAM support.
- OpenNebula support: not ready.
- OpenStack support: ready, documentation in GitBook.
Milestone Status

- **MS19 status – Month 15**

  Support for container execution with trusted container repository and standard interfaces.
  - Container support in OpenStack → **done**.
  - Container support in OpenNebula → **done**.
  - Repository sync → in progress (**good progress**).
  - OCCI support → **done**.

- **MS22 status – Month 15**

  Implementation of spot-instances mechanisms
  - Pilot testbed → **done**.
  - Blueprint → **in progress** (long running task).
  - Implementation → **in progress** (mainly QA fixing).
WP4.2 Storage Virtualisation

Benjamin, Marcus, Bas
From KIT

Vladimir, Enrico, Andrea
From INFN

Marina, Paul, Patrick, Jürgen, Marina
From DESY

Staszek
From PNSC
4.2.1.A RDA
- Preparing for DENVER is ongoing as result for the TOKYO meeting

4.2.1.B SNIA-CDMI Extension
- Continuous definition with SNIA (INDIGO has a better idea of data management than SNIA 😊)
- Defining a Version 1 prototype for storage capabilities
- Simplified (TAPE, DISK, TAPE_DISK), (pushed by Sandro, result from Champion meeting)

4.2.1.C CDMI Implementation
- Good progress
- Already 4 CDMI endpoints deployed running reference implementation against different endpoints.
- First endpoint already able to change capabilities, but only with REST nor now, not yet with CDMI. (DEMO yesterday)
- Creating endpoint monitoring (compatibility checking) website
- PSNC integrating object storage, not yet clear which solution is preferred.

Identity Harmonization
- Demonstrated prototype (Amsterdam) working
- Now with lower priority to get the QoS done fast.
QoS The Big Picture

Platform as a Service

Canonical Storage Property Information System

IaaS

D&M

GUI

REST API

D&M

D&M

GUI

REST API

INDIGO-DataCloud RIA-653549
General CDMI Approach

INDIGO - DataCloud RIA-653549
WP4.3 Update on Network Virtualisation

Zdenek Sustr

CZNET
4.3.1 Intracloud networking

- **OpenStack**: networking features available over OCCI, contributing to OOI.
  - Had to be redesigned several times to clarify requirements.
  - Now there is an open pull request for OOI, too late for INDIGO Prototype but will be included in 1st release

- **OpenNebula**: Initial plans for NOW (wrapper around 3rd party orchestrator)
  - Open Nebula announced virtual router support in release 5.
  - We decided to wait for it (originally was due in March) but it still has not been released.

- **AWS**: Streamlining of existing code: seen as opportunity to train developers. Ongoing.

- **Side product**: pOCCI.
  - Test suite for OCCI compliance (not restricted to network, performs ETSI tests)
  - "Test Descriptions for Cloud Interoperability"
  - Beta should be out this week. Information a few weeks old.
4.3.2 Evaluation of SDN technologies from vendors

Product evaluation by Santer Reply

- Evaluation of available technologies, first outcome to be published as deliverable 4.4 in September.
- There is an extensive and growing shared document.
  - Comments are welcome but work is still in progress
  - [https://docs.google.com/document/d/1TzyhTr6f4stQbOfRn3GcV5jwKwlF25Q_2eN4MCoU3o](https://docs.google.com/document/d/1TzyhTr6f4stQbOfRn3GcV5jwKwlF25Q_2eN4MCoU3o)
4.3.3 Appliances and Services
- Priority lowered in favor of other activities
- Now mainly in cooperation with WP3
  - Appliances as building blocks for INDIGO pilot infrastructure.
Outline of WP4.1 (Computing)

WP4.1.1 - Container support
WP4.1.2 - Repository integration
WP4.1.3 - OCCI extension to support containers
WP4.1.4 - Improved scheduling
WP4.1.5 - Integration of cont. in batch systems
WP4.1.6 - IaaS TOSCA orchestration

Milestone status
**Goal**

- **Objective**: Execute containers as light-weight virtual machines.
- Not introducing API changes $\rightarrow$ normal resources for the CMF.
- **Similar functionality** on both CMFs.
- Higher level features are outside WP: orchestration, application, etc.
General remarks on container support II

OpenStack initial considerations

- LXC support in-tree, officially supported.
- OpenStack nova-docker driver available out-tree.
- OpenStack Magnum: new component for managing containers (Kubernetes).
  - Gaining a lot of momentum.
  - Overlaps with WP5 components.
  - We will follow up the developments, but not focusing on them.
OpenNebula initial considerations

- Driver available only for LXC.
- No Docker driver.
- Several approaches to Docker integration presented in ONE Conf. in Barcelona [1]
  - Remark: no Docker support planned.
- Megam project, similar to OpenStack Magnum.

Container Support in OpenStack: status

  - Evaluation of LXC driver, nova-docker driver.
  - Evaluation of Magnum project by CERN → different API.

- Openstack + Docker testbeds deployed at LIP, INFN-PD and CSIC.

- nova-docker bugs solved:
  - #1560951
No Docker hypervisor driver foreseen by Docker developers.

WP4.1 has developed it, filling one of the gaps in ONE development.

ONEdock driver available

- [https://github.com/indigo-dc/onedock](https://github.com/indigo-dc/onedock), officially announced in OpenNebula blog

**ONEDock features**

- No API changes.
- Contextualization.
- Network management.
- VNC support.
- etc.
<table>
<thead>
<tr>
<th>Container Support Matrix</th>
<th>Images</th>
<th>Net</th>
<th>Volumes</th>
<th>Snapshots</th>
<th>Quotas</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OpenStack</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Libvirt + LXC</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>nova-docker</td>
<td>Y</td>
<td>Y</td>
<td>N [2]</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td><strong>OpenNebula</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Container support issues I

Docker-related issues

- Not possible to attach a volume (i.e. block device) in runtime.
- Do we need this? Not possible in Docker, therefore not possible in both ONE and OpenStack.
- Functionality is implemented, developers are doing bikeshedding: they do not agree on the CLI.
- If we need this we need to push for this functionality.
- OpenStack (nova-docker) and Docker related bugs:
  - https://bugs.launchpad.net/nova-docker/+bug/1321817
  - https://github.com/docker/docker/pull/8348
  - https://github.com/docker/docker/pull/8826
  - https://github.com/docker/docker/issues/8829
Other related issues

- OpenStack and ONE restrict to one driver in each compute node.

- Dedicated compute node, excluding those nodes from running other VMs.

- Non-documented, risky possibility to run two hypervisor drivers in OpenStack in one host.
Outline of WP4.1 (Computing)

WP4.1.1 - Container support

WP4.1.2 - Repository integration

WP4.1.3 - OCCI extension to support containers

WP4.1.4 - Improved scheduling

WP4.1.5 - Integration of cont. in batch systems

WP4.1.6 - IaaS TOSCA orchestration

Milestone status
Goal

- Sync between INDIGO DockerHub and local sites repository

Status

- Stalled task, no progress.
Outline of WP4.1 (Computing)

WP4.1.1 - Container support
WP4.1.2 - Repository integration
WP4.1.3 - OCCI extension to support containers
WP4.1.4 - Improved scheduling
WP4.1.5 - Integration of cont. in batch systems
WP4.1.6 - IaaS TOSCA orchestration

Milestone status
OCCI support

Goal

- Manage containers through the OCCI interface.

Status

- Tested OCCI + OpenStack + nova-docker: INFN-PD, LIP, CSIC.
- ooi available for OpenStack
- Tested OCCI + OpenNebula + ONEDock: UPV.
- rOCCI available for OpenNebula
- Ongoing implementation of OCCI 1.2
- If several networks are defined there is an error in ooi when creating a VM/Container, existing bug:
  - https://bugs.launchpad.net/ooi/+bug/1524935
Outline of WP4.1 (Computing)

WP4.1.1 - Container support
WP4.1.2 - Repository integration
WP4.1.3 - OCCI extension to support containers
WP4.1.4 - Improved scheduling
WP4.1.5 - Integration of cont. in batch systems
WP4.1.6 - IaaS TOSCA orchestration

Milestone status
Fair-share scheduling

Goal

- Provide a priority queuing mechanism for handling the user resource requests that cannot be immediately fulfilled.

Status

- Synergy deployed and tested at INFN-PD and CC-IN2P3.
- Integration in OpenStack StackForge ongoing.
- Adapting code to latest OpenStack version (Liberty).
- ONE integration ongoing, testbed being deployed at INFN-TO.
Partition director

Goal

- Balancing (batch – cloud) control system.

Status

- Implementation in progress.
- Testbed setup at INFN-CNAF.
Spot-instances support

Goal

- Implement a mechanism for the preemption or termination of second-class VMs.

Status

- Blueprint submitted to OpenStack developers [3], still under discussion.
- Implementation ongoing, no changes in the functionality.
- Pluggable implementation (HostManager + Scheduler).
- Testbed deployed at CSIC infrastructure.

Outline of WP4.1 (Computing)

WP4.1.1 - Container support
WP4.1.2 - Repository integration
WP4.1.3 - OCCI extension to support containers
WP4.1.4 - Improved scheduling
WP4.1.5 - Integration of cont. in batch systems
WP4.1.6 - IaaS TOSCA orchestration

Milestone status
Batch system integration

Goal

- Explore the possibility to execute containers in batch systems.
- Explore the access to GPUs and Infiniband interconnects.

Status

- New implementation started.
- Docker with CUDA 7.5 and pyOpenCL.
- Testing together job submission + GPGPUS together.
- Planning to use the CAFFEE CUDA neural network.
- Writing the procedure for running those applications.
Outline of WP4.1 (Computing)

WP4.1.1 - Container support
WP4.1.2 - Repository integration
WP4.1.3 - OCCI extension to support containers
WP4.1.4 - Improved scheduling
WP4.1.5 - Integration of cont. in batch systems
WP4.1.6 - IaaS TOSCA orchestration

Milestone status
IaaS TOSCA Orchestration

Goal

- Provide IaaS orchestration using TOSCA in both CMFs.

TOSCA support global status

- TOSCA support will leverage the OpenStack tosca-parser
- OpenStack support based on Heat.
- OpenNebula support based on the IM.
Detailed status I

TOSCA parser status

- Parser is being extended to support INDIGO requirements.
- Changes are being accepted upstream:
  - INDIGO (UPV) 2nd global contributor to tosca-parser.

TOSCA in OpenStack status

- Working on support through TOSCA parser + Heat translator.
- Changes are being accepted upstream.
  - INDIGO (CERN) 6th global contributor to heat-translator.
- Heat being deployed at CSIC.
Detailed status II

TOSCA in OpenNebula status

- Work ongoing, orchestration support is done through the Infrastructure Manager (IM).

- TOSCA parser + TOSCA to RADL translator based support.

- The IM supports all the INDIGO TOSCA types
Outline of WP4.1 (Computing)

WP4.1.1 - Container support
WP4.1.2 - Repository integration
WP4.1.3 - OCCI extension to support containers
WP4.1.4 - Improved scheduling
WP4.1.5 - Integration of cont. in batch systems
WP4.1.6 - IaaS TOSCA orchestration

Milestone status

INDIGO-DataCloud RIA-653549
Next steps and ETAs

- M9: OpenStack + nova-docker + OCCI pilot.
- M9: ONE + onedock + OCCI pilot.
- M12: Champions meet developers (today).
- M13: Spot instances pilot prototype.
- M13: Automatic sync with INDIGO DockerHub at the site-level.
- M14: INDIGO All-Hands meeting.
Milestones and timelines

**Summarized (DoW) Roadmap**

- **MS19** (Month 15): Support for container execution with trusted container repository and standard interfaces.
- **MS22** (Month 15): Implementation of spot-instances mechanisms
- **MS20** (Month 23): Execution of containers through batch systems.
- **MS21** (Month 30): Proof of concept for accessing hardware accelerators (GPUs) and low latency networks (IB) in containers.
- **MS23** (Month 30): Implementation of Advanced Scheduling Policies
- **MS24** (Month 30): Delivery of the orchestration engine with standard interfaces whenever possible (TOSCA)

**Where are we?**

- Month 12th.
- Detailed roadmap and current progress tracked in OpenProject:
  - [https://project.indigo-datacloud.eu/projects/wp4/timelines/16](https://project.indigo-datacloud.eu/projects/wp4/timelines/16)
Support for container execution with trusted container repository and standard interfaces.

- Due in month 15.
- Container support in OpenStack → done.
- Container support in OpenNebula → done.
- Repository sync → missing.
- OCCI support → done.
MS22 status

Implementation of spot-instances mechanisms

- Due in month 15.
- Pilot testbed → done.
- Blueprint → in progress (long running task).
- Implementation → in progress.
Thanks from Alvaro

Indigo Web: http://indigo-datacloud.eu

Questions to:

aloga@ifca.unican.es
Structure:

4.2.1.A RDA: Paul
4.2.1.B SNIA-CDMI Extension: Paul
4.2.1.C CDMI Implementation: Benjamin

4.2.2.A Dynamic Views: Maciej, Paul

4.2.2.B Identity Harmonisation: Benjamin, Marcus
Important (visible) Gremium for “Storage in Science”

Goals:
- Define Quality of Services in Storage (QoS)
- Provide first implementations in INDIGO

Paris RDA meeting: Setup of Interest group
- Included interested partners from outside INDIGO

Tokyo RDA meeting
- Working Group accepted (i.e. delivery of output at specified timesteps is required)
- Additional input from Japanese libraries

QoS specification published in Milestone M25
- https://owncloud.indigo-datacloud.eu/index.php/s/wfs0Zy0lb76YUD4

Deputy (non German) for RDA Working Group currently being sought
SNIA-CDMI Extension

- Goal Feed back CDMI QoS extensions to SNIA
  - ... so INIDIGO QoS definition is standardized

- Status:
  - Mechanism for patching the reference implementation, garded by SNIA, are established. It seems they trust us.
  - First INDIGO patches have been accepted by SNIA.
  - Generally good working conditions appreciated.
  - INDIGO representatives are joining the weekly SNIA meeting on standards.
  - JSON objects are defined, however CDMI extention details still pending.
CDMI Implementation

Goals

- Implement QoS specification
- Provide WP5 with standardised way to query QoS information
- Allow high-level systems (OneDATA, FTS) to negotiate where they place data based on QoS but as well on price.

Status

- Two Implementations branches (both based on the SNIA reference implementation)
- dCache
  - Developed QoS spec
  - Implementing as native dCache plugin (home-advantage)
- TSM, HPSS and StoRM (StoRM more close to dCache)
  - Development infrastructure in place (git, VMs, ...)
  - QoS specificaiton consumption in progress
  - QoS information retrieval from storage systems in progress
- Possible support for Cloud Storage (S3, Google, ... )
- Possible support for Object Storage (e.g. CEPH, SWIFT,...)
Dynamic Views

Goal

- Translate between object storage and filesystem
- Same object can be accessed with its object ID but is visible in a POSIX file system tree as well.

Status

- Task successfully finished
  - Milestone reached
  - Demo available (Frascati meeting if requested)
- Waiting for users (in contact with WP5 and WP6)
Identity Harmonisation

Goal

- Allow logins with different credentials and technologies
- Map to the same local user
- Accumulate the groups

Status

- Works with SAML and (some) SAML-Tokens
- Retrieve mapping information via SCIM works
- Looking forward to testing with IAM
- Open ID connect mapped to X501 (in dCache for Frascati)
Demo

1. Login with User A (look at UID and GIDs)
2. Login with User B (compare UID and GIDs with User A)
3. Harmonise Identities
   - Current: Commandline call by the admin
   - Future: Account linking at IAM (INDIOG Product) web frontend
4. Login with User A (compare UID and GIDs)
5. Login with User B (see that UID and GIDs are identical to User A)
6. WARNING: DO NOT USE THIS FOR SHARING FILES NEVER!
For something completely different (cross WP activity, already covered by Andrea in the AAI talk)
TTS is deployed one at a site (or service)

- There it can **translate** an OIDC token into an ssh public/private keypair
- It can manage several services
- E.g.: Managing Accounts on a LINUX server.

**Goal:** Allows **any service** to authenticate against OIDC (IAM)

- REST interface (authenticated by OIDC)
- Web Frontend
- Support services via (extensible) shell scripts:
  - e.g. ssh, gridftp, S3, mysql, ...
- Support operations:
  - Create, Remove, Security Incident (freeze, kill)
Demo (Tomorrow)

1. Go to TTS webpage
2. Login via IAM (we used google for development)
3. Create account (observe this from root shell)
4. Remove account (observe this from root shell)
Thanks from Marcus

Indigo Web: http://indigo-datacloud.eu

Questions to:

hardt@kit.edu
4.3.1 Intracloud networking

- Contributing networking code to OOI (OCCI for Open Stack) important for EGI
  - For OCCI 1.1 a simple solution will be provided within 1 month from now.
  - The issue is a bit more difficult as currently we would need two OCCI endpoints, which doesn’t work for all cases.
  - Evaluation is need after details on OCCI 1.2 are available.
- Mostly work on rOCCI (see next slides)
- Working on rOCCI for AWS
- Waiting for OpenNebula V5 release for rOCCI adjustments.
WP 4.3 Tasks

- 4.3.2 Evaluation of SDN technologies from vendors

- Product evaluation by Santer Reply
  - Evaluation of available technologies, first outcome to be published as deliverable 4.4 in September.
  - There is an extensive and growing shared document.
    - Comments are welcome but work is still in progress
    - https://docs.google.com/document/d/1TzyhTr6f4stQbOfRn3GcV5jwKwclF25Q_2eN4MCoU3o

INDIGO-DataCloud RIA-653549
WP 4.3 Tasks

4.3.3 Appliances and Services

- Priority lowered in favor of other activities
- Now mainly in cooperation with WP3
  - appliances as building blocks for INDIGO pilot infrastructure.
WP 4.3 Tasks

4.3.1 Intracloud networking
- Contributing networking code to OOI (OCCI for Open Stack) important for EGI
  - For OCCI 1.1 a simple solution will be provided within 1 month from now.
  - The issue is a bit more difficult as currently we would need two OCCI endpoints, which doesn’t work for all cases.
  - Evaluation is need after details on OCCI 1.2 are available.
- Mostly work on rOCCI (see next slides)
- Working on rOCCI for AWS
- Waiting for OpenNebula V5 release for rOCCI adjustments.

4.3.2 Software Defined Networks (SDN’s)
- Product evaluation by Santer Reply
- Preparation of the deliverable 4.4 (due in September)

4.3.3 Appliances and Services
- Priority lowered in favor of other activities
**rOCCI libraries and apps**

**rOCCI** is a suite of libraries and client/server applications to enable the use of OCCI with cloud management frameworks (CMF) not supporting OCCI natively.

- The libraries (rOCCI-core and rOCCI-api) are in Ruby and implement the OCCI class structure, rendering, parsing and transport.
- rOCCI-cli is currently the only full-featured OCCI client in existence.
- The OCCI-server is a stand-alone state-less service communicating with clients via OCCI, and "translates" their requests to the underlying cloud management framework (CMF), and vice versa.
rOCCI client / server big picture
INDIGO needs this product to enable OCCI in OpenNebula and potentially for public cloud providers.

INDIGO contributes to this product mainly by implementing network management functions for OpenNebula sites.

INDIGO T4.3 contributes network-handling code to the OpenNebula and the AWS backend.
pOCCI (quite unimaginatively named)

- pOCCI is an implementation of ETSI:
  - ETSI: “European Telecommunications Standards Institute”
  - "Test Descriptions for Cloud Interoperability"
  - https://archive.org/details/etsi_ts_103_142_v01.01.01

- Originally intended as an internal testing product for INDIGO Product Teams.
- Found useful also as a stand-alone product
  - Used in Cloud Plugfests http://www.cloudplugfest.org/
  - Implements "real-world" subset of ETSI tests
Some Background

- The ETSI document also specifies tests for handling "abstract" OCCI concepts such as OCCI Category. This is a class that all OCCI Infrastructure classes (Compute, Storage, Network) inherit from, but no real-world cloud framework can instantiate it; only its descendants.

- It sends pre-formatted OCCI queries to the server side and checks replies for correctness and OCCI compliance. For instance: it sends a query for the server to create a virtual machine, and checks if the server responds correctly with an identifier of the machine. It does not check if the server really created the resource, only that it understood and responded appropriately.
Status of pOCCI

- Currently only OCCI http/text rendering is supported (OCCI 1.1). JSON Rendering will be implemented later
  - Introduced with OCCI 1.2, which is due for release any time.

- Feedback has yet to be provided to ETSI.

- Sideremark: The same ETSI document also defines a similar set of tests for CDMI compliance. CDMI is the protocol frame, we reported about in WP4.2.
Thanks from Zdenek

Indigo Web: http://indigo-datacloud.eu

Questions to:

sustr4@cesnet.cz