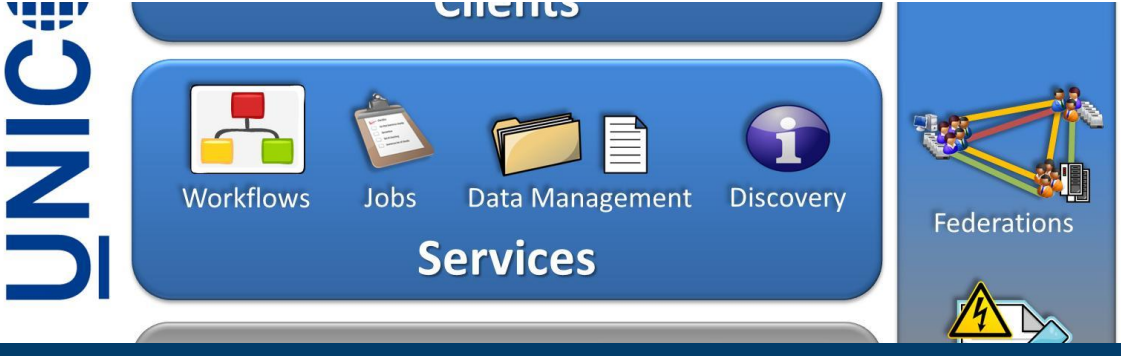
Uniform Resource Access
Compute and Cloud Resources at JSC

2022-05-17 | Björn Hagemeier | Juelich Supercomputing Centre



Part I: UNICORE

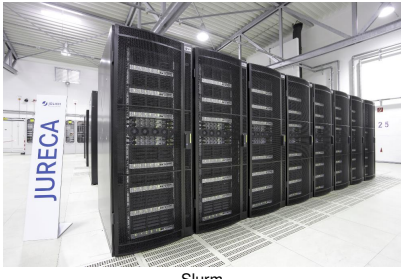
Motivation

Differences of systems

Uniform **I**nterface to **Co**mputing **Re**sources

Various RMS on systems

- JUQUEEN: IBM LoadLeveler
- JURECA: Slurm
- Different job description languages for specifying # of nodes, memory requirements, wall time, ...
- Different parameters on the command line
- Unify and simplify supercomputer access



Slurm

Load Leveler

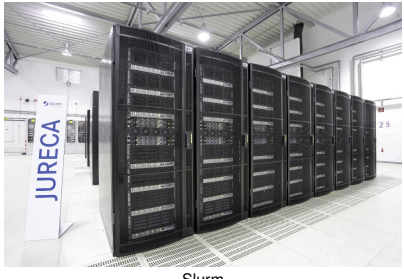
Motivation

Differences of systems

Uniform **I**nterface to **Co**mputing **Re**sources

Various RMS on systems

- JUQUEEN: IBM LoadLeveler
- JURECA: Slurm
- Different job description languages for specifying # of nodes, memory requirements, wall time, ...
- Different parameters on the command line
- Unify and simplify supercomputer access



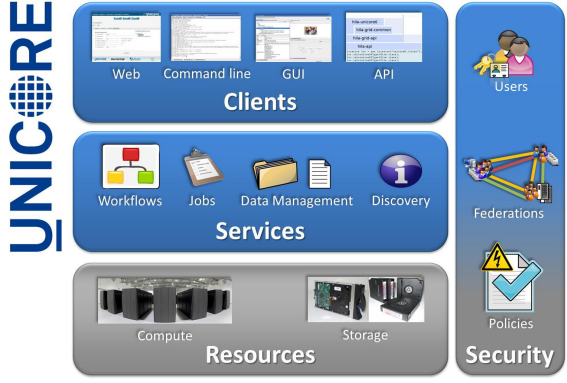
Slurm

Slurm

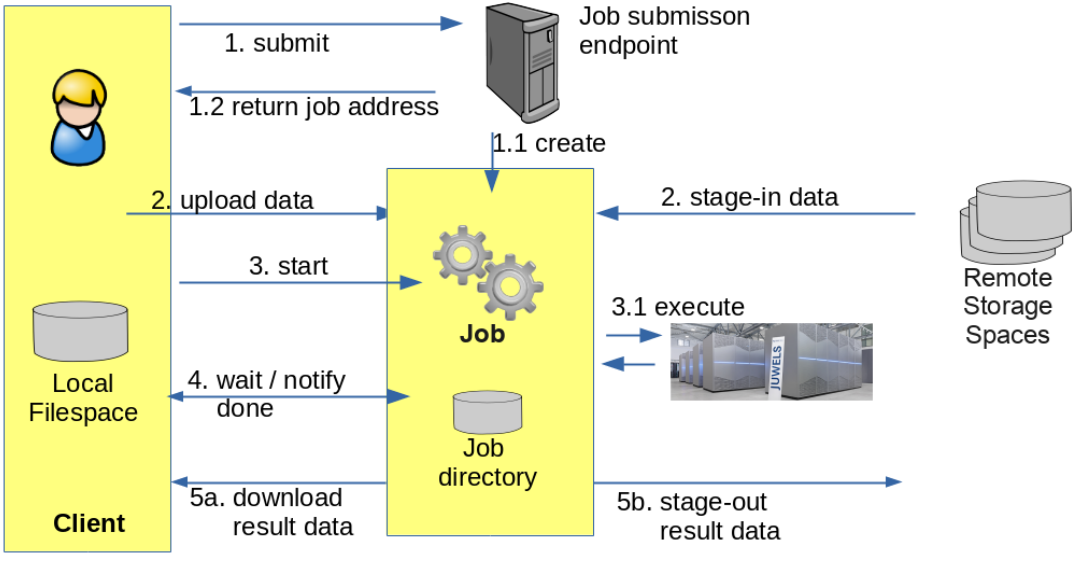
Why UNICORE

Advantages

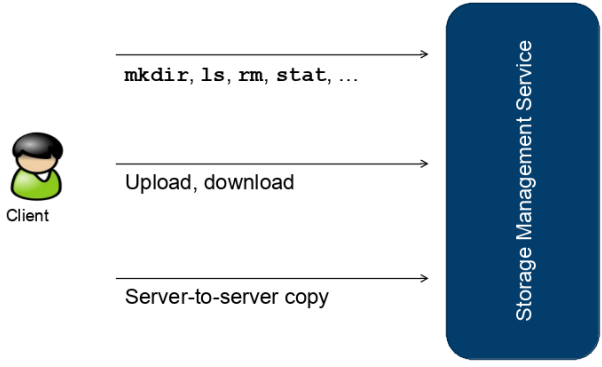
- Hide system specific commands
- Create, submit and monitor jobs
  - Seamless, secure, and intuitive access to distributed compute and data resources
- Multiple clients
- Integrated **data management**
- **Federated identities**
- Open Source:
  `https://github.com/UNICORE-EU`



Job execution model



Data management and file transfer



- File Systems
- Apache HDFS
- S3
- iRODS

Efficient file transfer

UFTP

- Data **streaming** library and **file transfer** tool
- Fully integrated into UNICORE
- Standalone (non-UNICORE) client available
- Client to server and server to server data transfers
- Data staging among UFTP-enabled sites
- Efficient **synchronization** of individual local and remote files using the **rsync** algorithm
- Optional **compression** and **encryption** of data streams

Efficient file transfer

UFTP

Source: SC Asia web site

- Data **streaming** library and **file transfer** tool
- Fully integrated into UNICORE
- Standalone (non-UNICORE) client available
- Client to server and server to server data transfers
- Data staging among UFTP-enabled sites
- Efficient **synchronization** of individual local and remote files using the **rsync** algorithm
- Optional **compression** and **encryption** of data streams
- Awarded “best systemic approach” in SC Asia Data Mover Challenge 2020

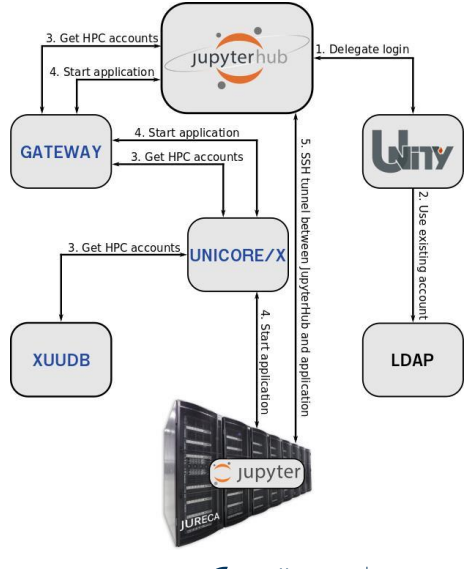
Clients and APIs

- Commandline tools
  - UNICORE Commandline Client (UCC): `https://sourceforge.net/projects/unicore/files/Clients/Commandline%20Client/`
  - UFTP client for high-performance data access: `https://sourceforge.net/projects/unicore/files/Clients/UFTP-Client/`
- RESTful APIs
  - curl, Python Requests
  - `https://sourceforge.net/p/unicore/wiki/REST_API/`
  - PyUNCIORE client library: `https://github.com/HumanBrainProject/pyunicore`

Jupyter Hub @JSC

HPC in your web browser

- UNICORE is an integral part of the Jupyter offering at JSC
- Start Jupyter Labs on JUWELS, JURECA-DC, JUSUF, DEEP, HDFML, or a cloud based VM
- `https://jupyter-jsc.fz-juelich.de/`



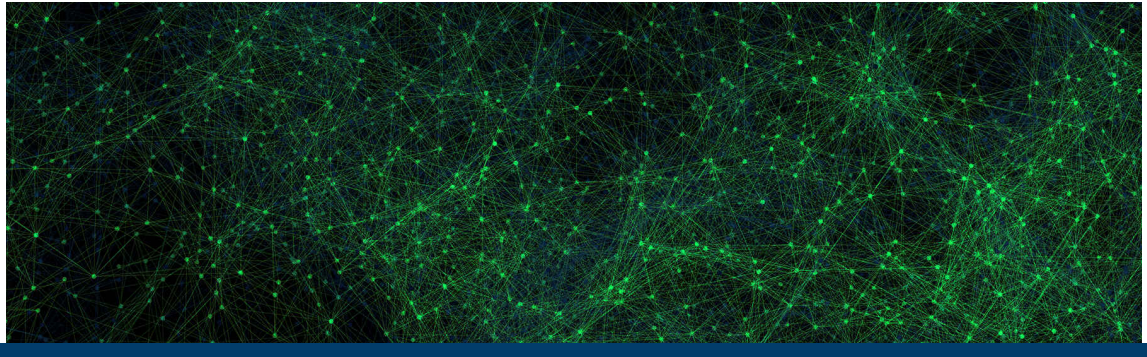
Additional information and support

UNICORE

- Project web site: `https://www.unicore.eu/` for downloads and documentation
- Product support: `unicore-support@lists.sourceforge.net`

UNICORE at FZJ

- User support email: `ds-support@fz-juelich.de`
- Registry: `https://fzj-unic.fz-juelich.de:9112/FZJ/rest/registries/default_registry`
- Documentation: `https://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/UNICOREProduction/unicore_production_node.html`



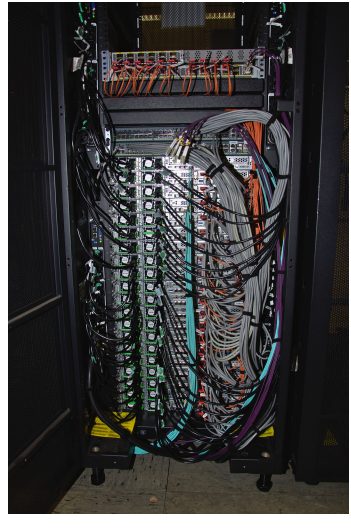
Part II: HDF Cloud

Overview

- OpenStack Infrastructure-as-a-Service (IaaS) environment
  - Compute, storage, network, orchestration, load balancing
  - Run VMs to provide services **linked to LARGEDATA**
  - Orchestration using OpenStack Heat
  - Load Balancer as a Service (LBaaS) using OpenStack Octavia
- Phase 1: 16 compute nodes, 768 VCPUs, 6.1TB RAM
- *Phase 2: 10 compute nodes, 480 VCPUs, 7.7TB RAM*
- **Total: 26 compute nodes, 1248 VCPUs, 13.8 TB RAM**
- 10GbE storage uplink per node up to 80Gb total
- 40GbE internal links
- Further information and reference:
  `https://go.fzj.de/hdf-cloud`

Overview

- OpenStack Infrastructure-as-a-Service (IaaS) environment
  - Compute, storage, network, orchestration, load balancing
  - Run VMs to provide services **linked to LARGEDATA**
  - Orchestration using OpenStack Heat
  - Load Balancer as a Service (LBaaS) using OpenStack Octavia
- Phase 1: 16 compute nodes, 768 VCPUs, 6.1TB RAM
- *Phase 2: 10 compute nodes, 480 VCPUs, 7.7TB RAM*
- **Total: 26 compute nodes, 1248 VCPUs, 13.8 TB RAM**
- 10GbE storage uplink per node up to 80Gb total
- 40GbE internal links
- Further information and reference:
  `https://go.fzj.de/hdf-cloud`



OpenStack

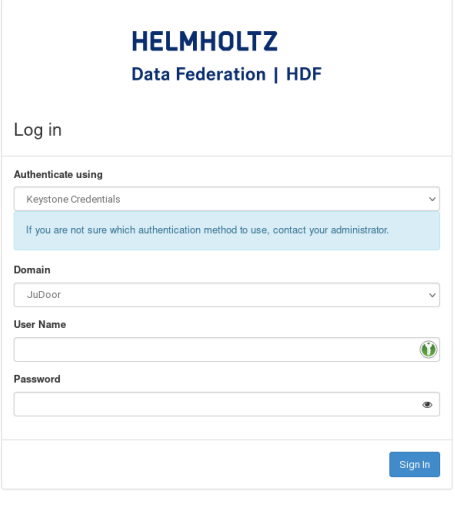
Software and services

- OpenStack Victoria release
  - released 2020-10-14, maintained until 2022-04-27, extended maintenance thereafter
  - upgrade hindered by a switch of the underlying Linux distribution
- Services
  - Keystone – authentication and service registry
  - Horizon dashboard – convenient Web UI appropriate for many simple tasks
  - Nova compute – virtual machine (VM) service
  - Neutron networking – software defined networks
  - Cinder volume – virtual block devices
  - Glance images – template images for VMs
  - Heat orchestration – infrastructure management
  - Octavia load balancing – load balancing as a service
  - Neutron VPNaaS – cross-site (or project) VPNs
  - Sahara – data processing through virtual clusters

Authentication

There are two ways to authenticate

- JSC account
  - username and password
  - usable from both commandline interface and Web UI
  - JuDoor profile → Make changes → enable HDFCloud
- Helmholtz login
  - directly usable only from Web UI
  - commandline access through application credentials
- however: you need a project and allocated resources before using HDF Cloud



Nova

Virtual machine service

Nova manages the lifecycle of virtual machines (VMs) that have

- a number of CPUs
- an amount of main memory
- storage: **system**, ephemeral, swap
- data storage: volumes
- network ports
- a template image containing an operating system

Nova

Virtual machine service

Nova manages the lifecycle of virtual machines (VMs) that have

- a number of CPUs
- an amount of main memory
- storage: **system**, ephemeral, swap
- data storage: volumes
- network ports
- a template image containing an operating system

} Nova

Nova

Virtual machine service

Nova manages the lifecycle of virtual machines (VMs) that have

- a number of CPUs
- an amount of main memory
- storage: **system**, ephemeral, swap
- data storage: volumes
- network ports
- a template image containing an operating system

} Nova

← Cinder

Nova

Virtual machine service

Nova manages the lifecycle of virtual machines (VMs) that have

- a number of CPUs
- an amount of main memory
- storage: **system**, ephemeral, swap
- data storage: volumes
- network ports
- a template image containing an operating system

} Nova

← Cinder

← Neutron

Nova

Virtual machine service

Nova manages the lifecycle of virtual machines (VMs) that have

- a number of CPUs
- an amount of main memory
- storage: **system**, ephemeral, swap
- data storage: volumes ← Cinder
- network ports ← Neutron
- a template image containing an operating system ← Glance

} Nova

Nova

Flavors

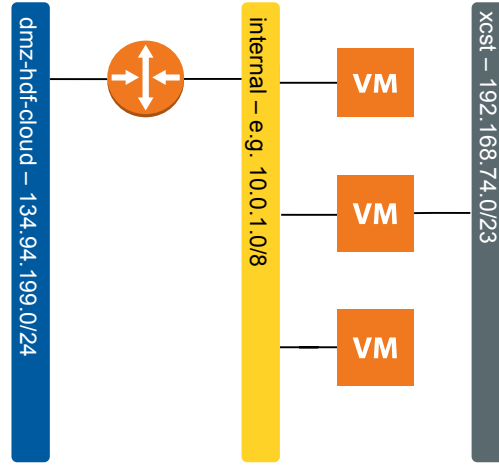
- two classes in general: **ordinary** (10GB root disk) and **large-disk** (30GB root disk)
- first characters determine amount of memory per VCPU: *t*iny, *s*mall, *m*edium, *l*arge, *xl*arge
- going from 1 VCPU per .5GB to 1 VCPU per 8GB
- Example flavors
  - t1
  - m8.large-disk
  - xl16
- custom flavors are possible

RAM \ VCPUs	1	2	4	8	16
.5 GB	t1	-	-	-	-
1 GB	s1	t2	-	-	-
2 GB	m1	s2	t4	-	-
4 GB	l1	m2	s4	t8	-
8 GB	xl1	l2	l4	s8	t16
16 GB	-	xl2	m4	m8	s16
32 GB	-	-	xl4	l8	m16
64 GB	-	-	-	xl8	l16
128 GB	-	-	-	-	xl16

Networking

Specific networks at JSC

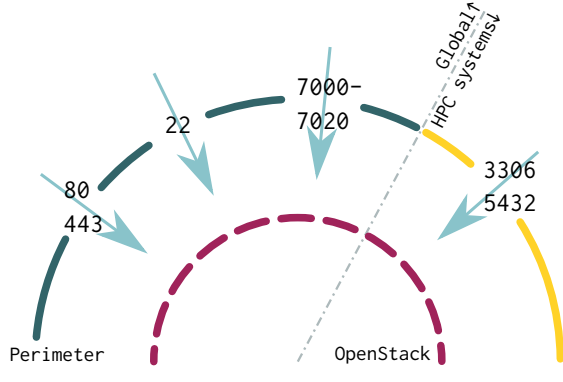
- **floating IPs** realized in router as DNAT/SNAT
- VMs without floating IPs not accessible from the outside and SNATed in outbound connections
- all new projects will be equipped with a **router** and **internal network**, such that you can immediately start working. JSC's **DNS servers** will be configured in the internal network



Network setup

Security groups and perimater firewall

- OpenStack firewall freely configurable
- Restrictions apply for inbound connections in perimeter firewall
  - Globally available services and ports: HTTP (80), HTTPS (443), SSH (22), 7000–7020
  - Available from HPC systems: MySql (3306), PostgreSQL (5432)
- Outbound connections: anything but MTA (25) aka. SMTP



Commandline interface

Prerequisites

- Python virtual environment
- Download credential files from the web interface (cf. authentication)

Run the following in your shell:

```
$ python3 -m venv openstack
$ source openstack/bin/activate
$ pip install python-openstackclient
```

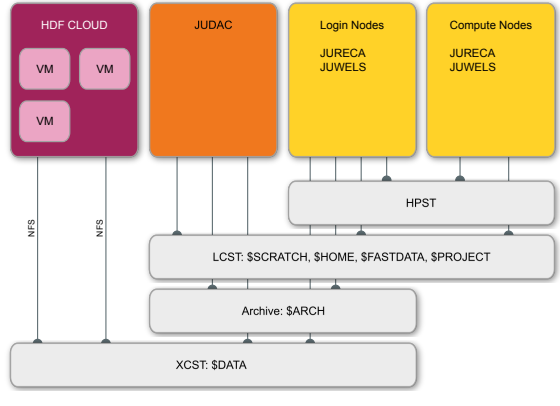
Authentication:

- Option 1: Download and `source openrc.sh`
- Option 2: Download clouds.yaml, put it in one of
  - current working directory as `clouds.yaml` or
  - ∼ `/.config/openstack/clouds.yaml`

JSC Storage Landscape

Availability of file systems

- XCST
  - $DATA on JUDAC and login nodes
  - dedicated NFS export to VMs
- Archive
  - $ARCH on JUDAC and login nodes
- LCST
  - $SCRATCH, $HOME, $FASTDATA, $PROJECT on JUDAC, login and compute nodes
- HPST
  - Login and compute nodes



Data access

VMs and the DATA file system

- Helmholtz Data Federation (HDF) Cloud / OpenStack cluster
  - Hosts virtual machines (VMs) for communities
  - Potentially administered by externals, bound by acceptable use policy
- Enable access to data beyond perimeter of SC facility
  - Web interfaces, databases, post processing, . . .
  - Users of service likely unknown to SC directory information service
- Access Method
  - POSIX file systems ($DATA) accessible in VMs via NFS mount from CES servers
  - Server side UID squashing
    - ensures consistency
    - requires services to manage data accordingly
    - read-write or read-only

Summary

HDF Cloud

OpenStack

- Project web site: `https://www.openstack.org/`
- Documentation: `https://docs.openstack.org/`

HDF Cloud at JSC:

- User support: `ds-support@fz-juelich.de`
- Web dashboard: `https://hdf-cloud.fz-juelich.de/`
- Documentation: `https://go.fzj.de/hdf-cloud`

JUSUF Cloud:

- User support: `sc@fz-juelich.de`
- Web dashboard: `https://jusuf-cloud.fz-juelich.de/`
- Documentation: `https://apps.fz-juelich.de/jsc/hps/jusuf/cloud/`