6th Fenix Research Infrastructure Webinar: Introduction to the JUSUF system at JSC

Tuesday 1 September 2020, 15:00 CEST

Speakers: Benedikt von St. Vieth and Pavel Mezentsev

(Juelich Supercomputing Centre)

Fenix has received funding from the European Union's Horizon 2020 research and innovation programme through the ICEI project under the grant agreement No. 800858.

The webinar is being recorded

Overview of webinar

- JUSUF Overview
- Cluster partition
- Cloud partition
- Q & A

JUSUF – **Jü**lich **SU**pport for **F**enix

- **Prime contractor: Atos**
- **Hybrid HPC/Cloud system with interactive workloads in mind**
  - **Compute partition for regular HPC jobs, ParaStation Cluster Tools**
  - **Cloud partition for IaaS workloads, OpenStack**
- **Provisioned and operated as part of the ICEI project**
- **Co-financed by the EC: Share of resources will be provided Europe-wide**
- **Infrastructure component in the federated pan-European e-infrastructure Fenix build up by BSC, CEA, CSCS, CINECA and JUELICH**

JUSUF – Hardware Overview

- **Based on AMD EPYC Rome CPUs and InfiniBand:**
  - **205 nodes with 2◊ AMD EPYC 7742, 256 GB, 1 TB NVMe SSD (out of these 61 GPU nodes with 1 Nvidia Tesla V100)**
  - **Mellanox HDR InfiniBand full-fat tree interconnect (HDR100 at node level)**
  - **GPFS connection via 40 Gb/s Ethernet per node**
- **4 frontend/login nodes with 100 GE uplink connection**
- **Storage:**
  - **8.2 Tbit/s total bandwidth between compute nodes and GPFS**
  - **2.4 Tbit/s total bandwidth between JUSUF and JUST-IME (connected via HDR InfiniBand)**
  - **JUST-IME can be used as a transparent cache for GPFS with a potential to significantly speed up the IO (there is also built-in MPI support). Note: JUST-IME resources need to be separately requested/granted**

JUSUF – CPU

- **AMD EPYC 7742, 225 Watts TDP**
- **64 cores, 2.25 GHz, up to 3.4 GHz Boost**
- **Up to two-way SMP**
- **Eight channels of DDR4-3200 memory per socket**
- **Max Bandwidth 190.7 GiB/s per socket**
- **4.6 Tflop/s peak performance per node (2◊ CPUs)**
- **PCIe 4.0**

JUSUF Cluster

JUSUF – Getting Resources

- Access can obtained via JuDoor: https://judoor.fz-juelich.de
- More information about the registration, getting access to the system and system documentation is available at https://fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/NewUsageModel/JuDoor.html
- More information about FENIX resources including JUSUF: https://fenix-ri.eu/access

FENIX RI

JUSUF – Access

- SSH access:
  - JUSUF can be accessed via jusuf.fz-juelich.de
  - If a specific login node is needed one can use jusuf**N**.fz-juelich.de where **N** is the number (1-**3**) of the login node
  - Only key-based authentication is supported (no user/password access)
  - The private key needs to be protected with a passphrase
  - A set of hosts/subnets that will be used to access the system has to be provided within a from-clause
  - The manual modification of authorized keys file is forbidden
- JupyterHub:
  - **https://jupyter-jsc.fz-juelich.de**
- UNICORE:
  - Used e.g., for HBP Collaboratory access

System Usage

- **Software Modules (Easybuild)**
  - **The installed software of the clusters is organized through a hierarchy of modules. Loading a module adapts your environment variables to give you access to a specific set of software and its dependencies.**
  - **Preparing the module environment includes different steps:**
    1. **Load a compiler (GCC and Intel are available) and potentially MPI (IntelMPI and ParaStationMPI are available)**
       Example: $ module load Intel ParaStationMPI
    2. **Then load other application modules, which were built with currently loaded modules (compiler, MPI or other libraries)**
       Example: $ module load GROMACS/↓version↑
  - **Software is bundled in Stages that are updated once or twice per year (major software upgrade including base compiler versions, etc.)**
- **Compilation**
  - **Modules modify the user environment (including $PATH) and make compilers directly available**
    - **Use of absolute paths is discouraged. Use $EBROOT↓Package↑ variable if necessary**
  - **Hint: Use compiler wrappers for compilation of MPI applications:**
    - `mpicc, mpicxx, mpif77, mpif90`
  - **Example: Compile an MPI program in C++:**
    - `mpicxx -O2 -o mpi_prog program.cpp`

JUSUF – SLURM

- **Slurm is the Batch System (Workload Manager) used on all production supercomputers at JSC**
  - **JSC uses Slurm together with the ParaStation resource management system developed by ParTec and JSC**
  - **Identical environment to JUWELS and JURECA**
- **Job** scheduling according to priorities. **The jobs with the highest priorities will be scheduled next.**

- Backfilling scheduling algorithm. **The scheduler checks the queue and may schedule jobs with lower priorities that can fit in the gap created by freeing resources for the next highest priority jobs.**

- No node-sharing. **The smallest allocation for jobs is one compute node. Running jobs do not disturb each other.**

- **Accounted CPU-Quotas/job = Number-of-nodes x cores/node x Walltime (corehours)**

SLURM Partitions

Partition	Resource	Value
-p batch (default)	**max. wallclock time (normal / nocont)**	**24 h / 6 h**
	min. / max. number of nodes	**1 / 64**
-p gpus	**max. wallclock time (normal / nocont)**	**24 h / 6 h**
	min. / max. number of nodes	**1 / 46**
-p develgpus	**max. wallclock time (normal / nocont)**	**24 h / 6 h**
	min. / max. number of nodes	**1 / 2**

- **Default node count = 1**
- **Default wallclock time = 1 h**
- **Partition layout is subject to change, potentially additional partitions for interactive use cases will be added**

FENIX RI

SLURM Job Submission

- Search for and enable a project with allocated computing time via
  - `jutil user projects; jutil env activate -p cjsc -A jsc (e.g.)`
- Submit a job requesting 2 GPU nodes for 1 hour, with 128 tasks per node (implied value of ntasks: 256):
  - `sbatch -N2 --ntasks-per-node=128 –p gpu --time=1:00:00 jobscript`
- Submit a job-script in the large partition requesting 32 nodes for 2 hours:
  - `sbatch –N32 -p batch -t 2:00:00 jobscript`
- Here is a simple example of a job script where we allocate 4 compute nodes for 1 hour. Inside the job script, with the srun command we request to execute on 4 nodes with 2 process per node the system command hostname, requesting a walltime of 10 minutes. In order to start a parallel job, users have to use the srun command that will spawn processes on the allocated compute nodes of the job.

```
#!/bin/bash

#SBATCH -J TestJob
#SBATCH -N 4
#SBATCH -o TestJob-%j.out
#SBATCH -e TestJob-%j.err
#SBATCH --time=10
#SBATCH -A <budgetID>

srun --ntasks-per-node=2 hostname
```

NEST on JUSUF

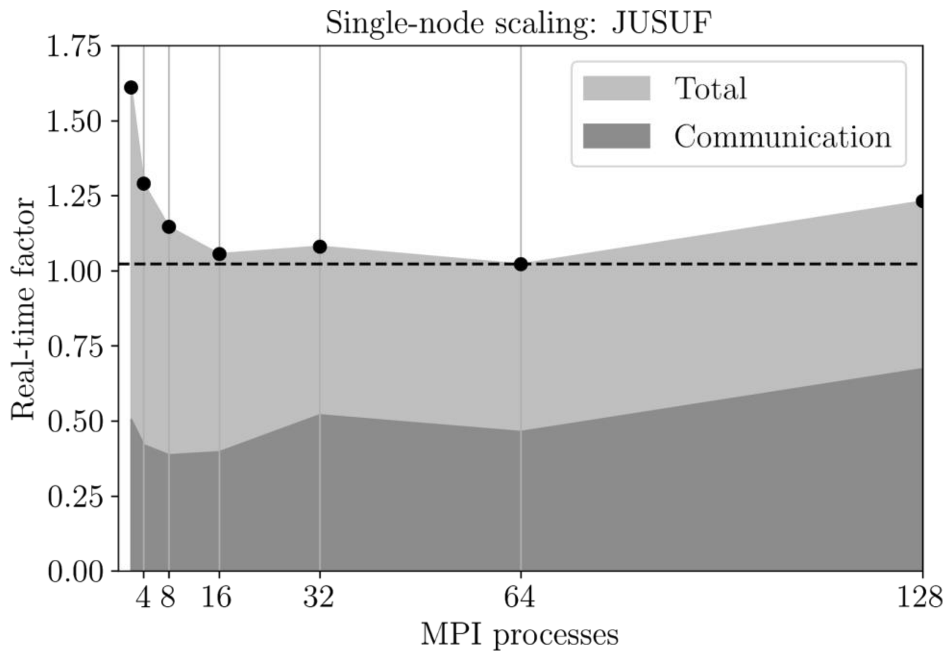
Best practices

- **Always pin OpenMP threads!**
  - **Use** `OMP_PROC_BIND=true`
  - **Remember to set** `OMP_NUM_THREADS`
- **Always bind MPI processes!**
  - **Intel MPI recommended**
    - **Offers greater tuning**
    - **Many** `I_MPI_SHM_*` **env variables**
  - **Use** `I_MPI_PIN_DOMAIN=socket`
  - **No need to use Slurmís** `--cpu-bind`
- **Recommended modules:**
  - `Intel/2019.5.281-GCC-8.3.0`
  - `IntelMPI/2019.6.154`
  - `jemalloc/5.1.0`
  - `GSL/2.5`

Single-node performance

- **Measure simulation performance:**
  - **Real-time factor:**

$$q_{\mathrm{real}} = T_{\mathrm{wall}}/T_{\mathrm{model}}$$



JUSUF Cloud

Cloud Computing - 5th Fenix Webinar

- **Introduction to Cloud Computing**
- **Introduction to the OpenStack Dashboard, with Demo**
- **https://fenix-ri.eu/media/webinars**

FENIX RI

Cloud Computing / OpenStack

- **Cloud computing offers on-demand, self-managed resources, as a service**

Software-as-a-Service (SaaS)
Software delivered via the internet, usually accessible via browser or downloadable client, examples include Google Play Store, Dropbox & Spotify

Platform-as-a-Service (PaaS)
Platform for deploying and building software, examples include operating systems, web servers & databases

Infrastructure-as-a-Service (IaaS)
Computing infrastructure including virtual machines, storage and network, examples include AWS, Azure and Google Cloud Platform

JUSUF Cloud – Access

- **Resources are available via Fenix/PRACE calls**
  1. **Users participate in the call, request a amount of resources**
  2. **ICEI coordination office allocates resources and acts as a information broker to JSC**
  3. **A project is created on JUSUF Cloud and users are pre-assigned to it**
- **Users of the JUSUF Cluster do** not **get access automatically**
- **Dashboard available at: https://jusuf-cloud.fz-juelich.de/**
  - **Access via** Fenix AAI **and/or a** JuDoor **account**
- **System documentation: https://apps.fz-juelich.de/jsc/hps/jusuf/cloud/index.html**
  - **A short howto for typical OpenStack first steps is included**
- **Ticket System: sc@fz-juelich.de, topic JUSUF Cloud**

FENIX RI

JUSUF Cloud – Workloads

- **Platform services (e.g., HBP platforms)**
- **Web services (including workflow management tools)**
- **Databases/repositories**
- **Compute & analytics (to some extend)**

Technologies and Design Considerations

- **RedHat OpenStack Platform 16 (OpenStack Train)**
  - **OpenStack Triple-O (OpenStack on OpenStack) for deployment**
  - **Fully containerized installation with RHEL8 and podman**
- **No (IaaS typical) resource overcommitment for CPU/memory**
  - ➢ **Available resources are limited to real hardware**
- **NFS used as storage backend for VMs, Images, and Block Storage**
  - ➢ **No CoW ⇨ longer instantiation due to copy of rootfs**
- **Node layout**
  - ➢ **Designed to enable HPC and cloud workloads on same hardware: Implies reduced redundancy on node level (e.g., regarding network resources)**
  - ➢ **High availability of critical services should be adressed by additional resources and/or implementation on a service level**

FENIX RI

Network Accessibility

- **Publicly adressable Floating IPs in range 134.94.88.***
- **Every assigned Floating IP reported to FZJ IP management**
- **Due to FZJ security constraints, only specific ports are available from the Internet**
  - **22 (SSH)**
  - **80 (http)**
  - **443 (https)**
  - **7000-7020 (tcp/udp)**

➢ **Make sure you use the proper ports in OpenStack Neutron and your services!**

JUSUF Cloud – Flavors

Flavour	VCPUs	RAM	Disk	Hardware
gpp.s	1	3GB	20	
gpp.m	2	8GB	20	
gpp.l	4	16GB	20	
gpp.xl	16	64GB	20	
gpp-ssd.l	4	16GB	20	NVMe
gpp-ssd.xl	16	64GB	20	NVMe
gpu.m	2	8GB	20	vGPU
gpu.l	4	16GB	20	vGPU
gpu.xl	16	64GB	20	vGPU

JUSUF Cloud – NVMe/vGPU

- **Number of NVMe/vGPU VMs limited due to underlying hardware**
- **vGPUs with very limited live-migration support**
  - **We can not guarantee a non-disruptive operation in case of platform maintenance**
  - **Please define your workloads in an easy-to-reproduce way!**
- **NVMe made available via PCI passthrough**
  - **Data cleanup has to happen on your own**
  - **live-migration in case of platform maintenance not supported**

➢ **Do not use NVMe/vGPU devices where you can avoid them!**

➢ **Use them for short-living computing/data-processing, but not for services like databases!**

JUSUF Cloud – vGPU Example

```
$ openstack server create --flavor gpu.m --security-group your_group --key-name
your_key --network your_net --image CentOS-8-GenericCloud-8.2.2004-
20200611.2.x86_64 gpu-webinar
$ Ö (allocate/associate floating IP)
$ ssh centos@↓Floating_IP↑
$ sudo yum install -y gcc make kernel-devel elfutils-libelf-devel libglvnd libglvnd-
devel pciutils gcc-c++ epel-release dkms
$ curl -o /tmp/NVIDIA-Driver.latest.run https://hpsrepo.fz-
juelich.de/jusuf/nvidia/NVIDIA-Driver.latest
$ chmod 755 /tmp/NVIDIA-Driver.latest.run
$ sudo mkdir /etc/nvidia
$ curl -o /etc/nvidia/gridd.conf https://hpsrepo.fz-juelich.de/jusuf/nvidia/gridd.conf
$ sudo /tmp/NVIDIA-Linux-x86_64-440.56-grid.run
$ sudo shutdown now -r
```

JUSUF Cloud – vGPU Example

```
$ ssh centos@↓Floating_IP↑
[centos@gpu-webinar ~]$ sudo lspci | grep -i nvidia
00:05.0 3D controller: NVIDIA Corporation GV100GL [Tesla V100 PCIe 16GB] (rev a1)
[centos@gpu-webinar ~]$ nvidia-smi
Mon Aug 31 13:21:33 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.56       Driver Version: 440.56       CUDA Version: 10.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GRID V100-4C        On   | 00000000:00:05.0 Off |                  N/A |
| N/A   N/A    P0    N/A /  N/A |    304MiB /  4096MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

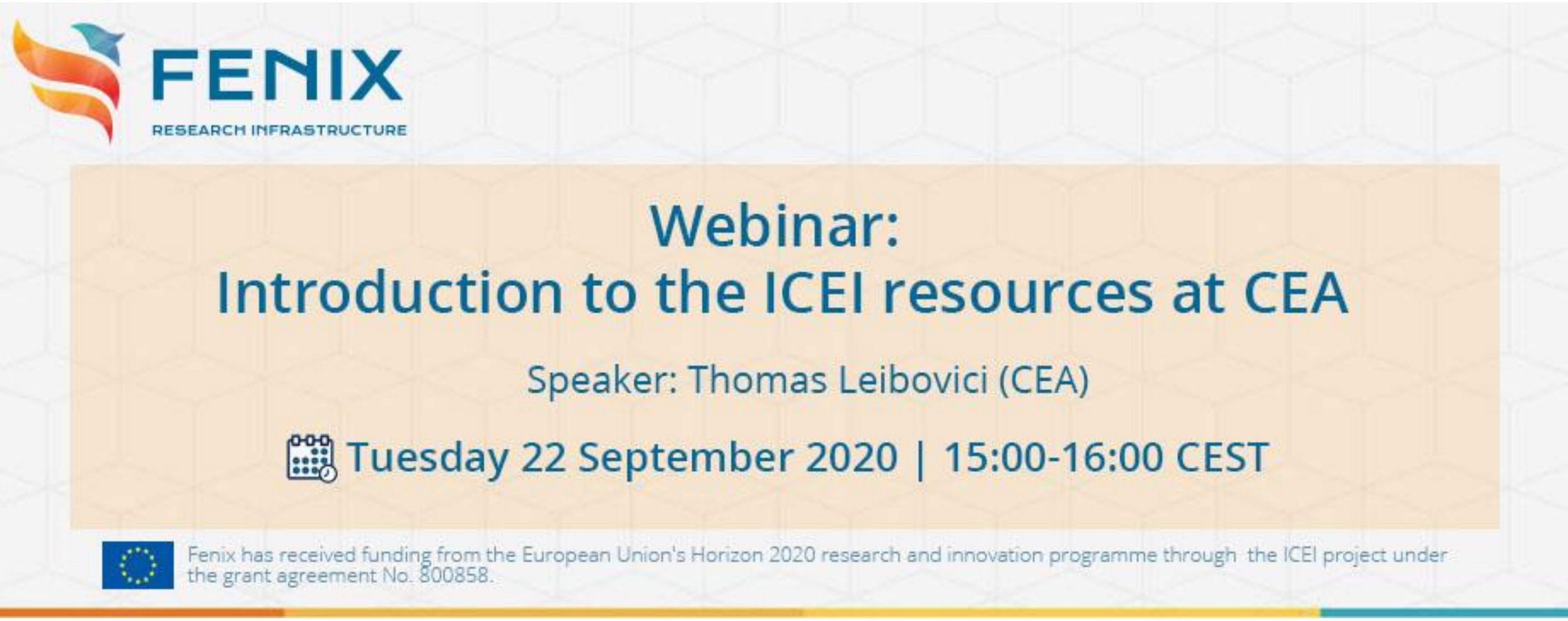
FENIX RI

Access to Fenix Services

- Neuroscientists obtain access through HBP via the HBP/EBRAINS Call
- Upcoming PRACE-ICEI Call for proposals to be out soon

Human Brain Project / EBRAINS:
Call for Proposals for European Neuroscientists at Large for Using ICEI e-Infrastructure Resources
Human Brain Project
EBRAINS
FENIX
RESEARCH INFRASTRUCTURE

- For researchers in need of:
  - **Scalable and interactive computing resources**
  - **Virtual machine services**
  - **Active and archival data repositories**
- All details on access to Fenix resources: https://fenix-ri.eu/access



FENIX
RESEARCH INFRASTRUCTURE

Webinar:
Introduction to the ICEI resources at CEA

Speaker: Thomas Leibovici (CEA)

Tuesday 22 September 2020 | 15:00-16:00 CEST

Fenix has received funding from the European Union's Horizon 2020 research and innovation programme through the ICEI project under the grant agreement No. 800858.

Register at:

https://fenix-ri.eu/events

We would appreciate your feedback!

Please respond to our survey

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

DE LA RECHERCHE À L'INDUSTRIE
cea

CINECA

JÜLICH
Forschungszentrum

EPFL
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ETH zürich

CSCS

Stay tuned!

Sign up for the **Fenix User Forum**:
https://fenix-ri.eu/infrastructure/fenix-user-forum

✉ icei-coord@fz-juelich.de
fenix-ri.eu
@Fenix_RI_eu

Fenix has received funding from the European Union's Horizon 2020 research and innovation programme through the ICEI project under the grant agreement No. 800858.