EOSC-hub

Fusion experience

Combined used of HPC, Cloud and HTC systems

Andrew Lahiff

Culham Centre for Fusion Energy, UK Atomic Energy Authority

eosc-hub.eu

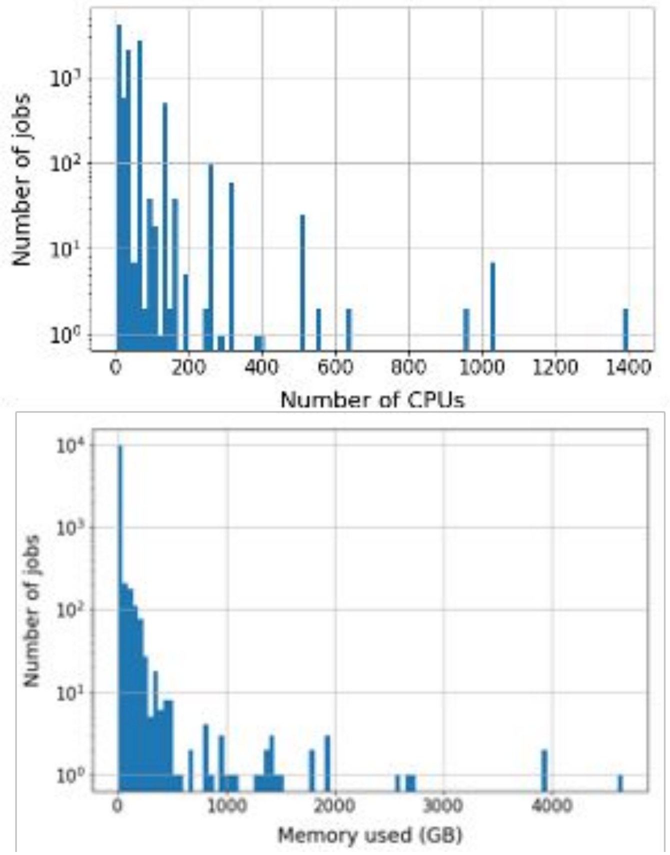
@EOSC_eu

Dissemination level: Public

EOSC-hub receives funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 777536.

Computing in fusion

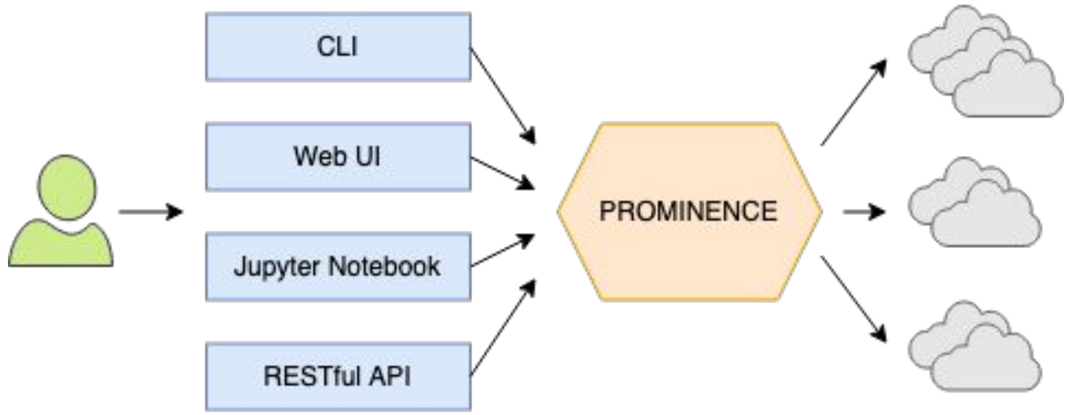
- Wide variety of applications
  - Plasma modelling, materials research, engineering, data processing, uncertainty quantification, rendering, machine learning, …
- Wide variety of languages & compilers
  - FORTRAN, C, C++, Python, IDL, Matlab, …
  - GNU, Intel, PGI
- Extensive use of environment modules & pre-installed software
  - Makefiles for specific HPC clusters
- Most computing in fusion is run on HPC facilities
  - At CCFE over 90% of jobs run locally are HPC
  - However expected that HTC will increase over the coming years



Recent jobs run on a HPC system at CCFE

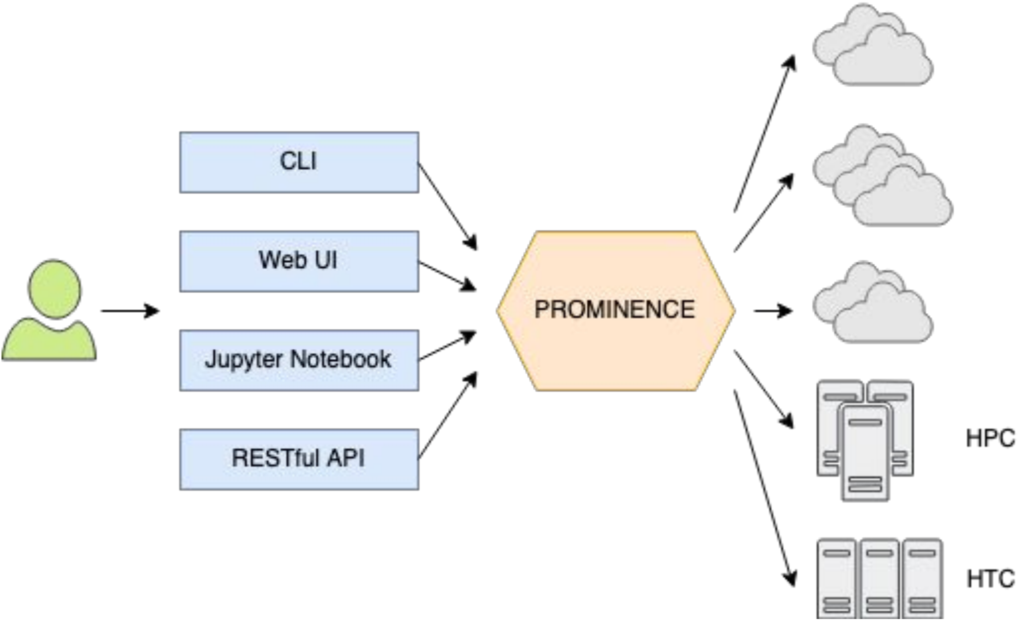
A brief introduction to PROMINENCE

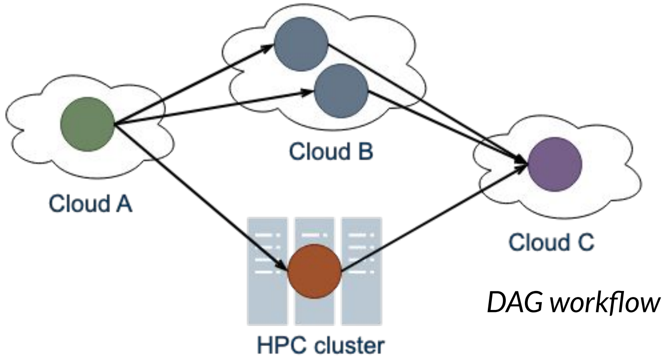
- PROMINENCE allows users to run batch jobs opportunistically & transparently across any number of clouds
  - Used in production for several use cases at CCFE



PROMINENCE & HPC

- Many users have access to multiple HPC systems
  - Typically manually “schedule” & submit jobs across them
- Some fusion workflows involve both HPC & HTC steps
- It would be useful if PROMINENCE could support HPC batch systems in addition to clouds
  - HPC batch systems more common than clouds with low-latency networking

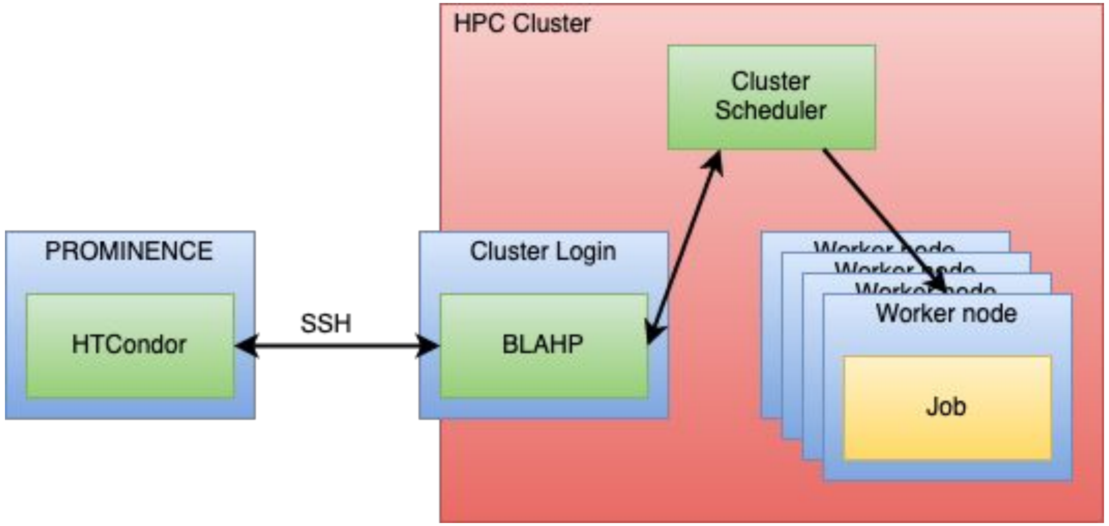




DAG workflow with steps running across multiple clouds and an HPC cluster

Integration with HPC: version 1

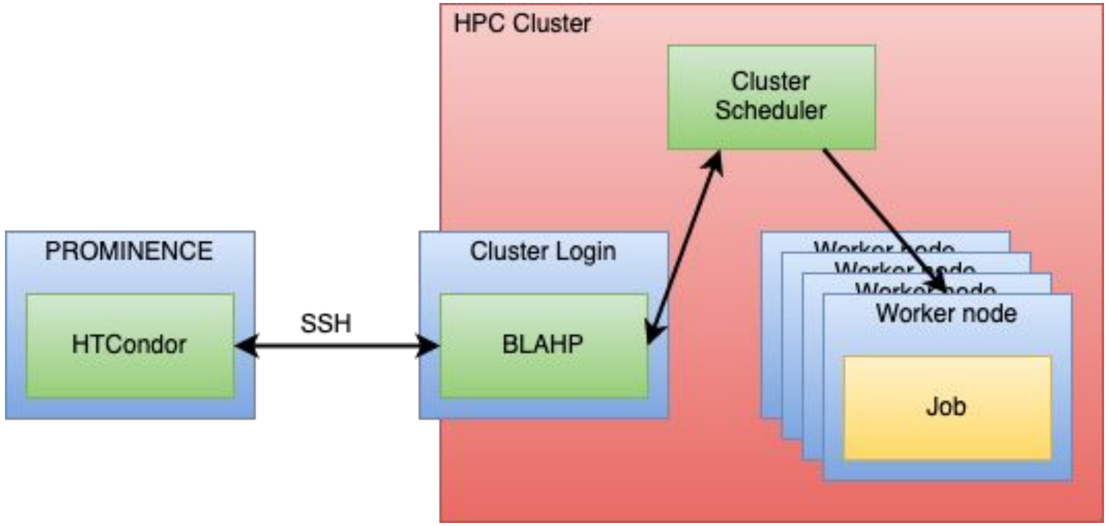
- PROMINENCE internally uses HTCondor as the job queue & to remotely execute jobs anywhere
- Use HTCondor job router to convert vanilla universe jobs into Grid universe
  - Leverage BOSCO functionality for submission to a remote batch system over ssh



http://htcondor.org

Integration with HPC: version 1

- PROMINENCE internally uses HTCondor as the job queue & to remotely execute jobs anywhere
- Use HTCondor job router to convert vanilla universe jobs into Grid universe
    - Leverage BOSCO functionality for submission to a remote batch system over ssh



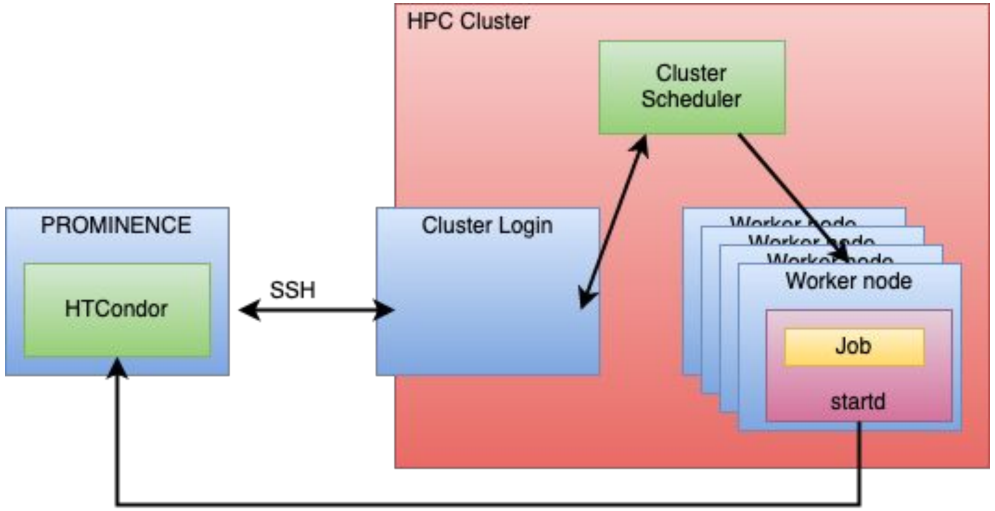
This method works reliably

Limitations

- Loss of functionality because jobs are not being run by a HTCondor startd
- E.g. lose the ability for users to view job stdout/err in real time

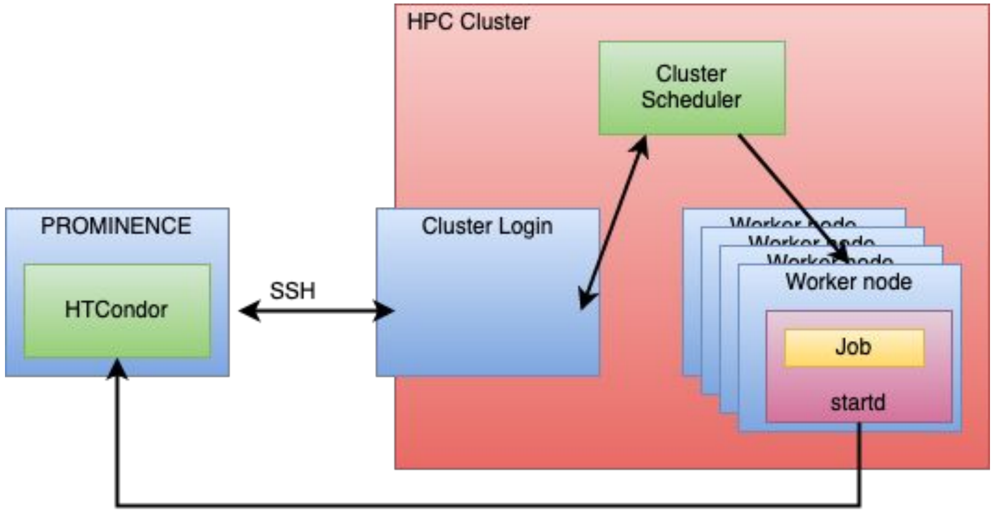
Integration with HPC: version 2

- Submit HTCondor startds (worker nodes) to the HPC system
  - Much more consistent with how we run jobs on clouds
  - Supports streaming of stdout/err in real time
- Could use HTCondor to submit the startds to the HPC system
  - But we used RADICAL-SAGA (simpler)
    - Python module which can submit jobs over ssh



Integration with HPC: version 2

- Submit HTCondor startds (worker nodes) to the HPC system
  - Much more consistent with how we run jobs on clouds
  - Supports streaming of stdout/err in real time
- Could use HTCondor to submit the startds to the HPC system
  - But we used RADICAL-SAGA (simpler)
    - Python module which can submit jobs over ssh



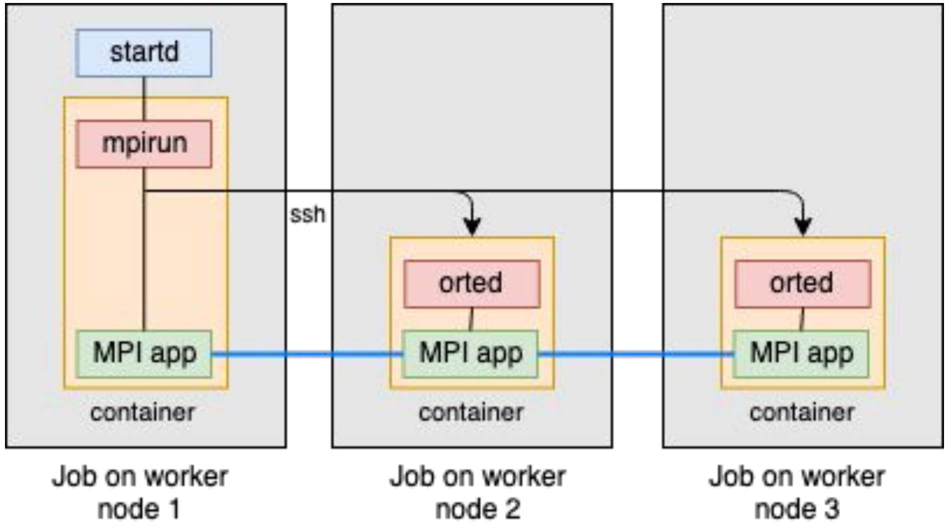
This method also works reliably

Potential limitations

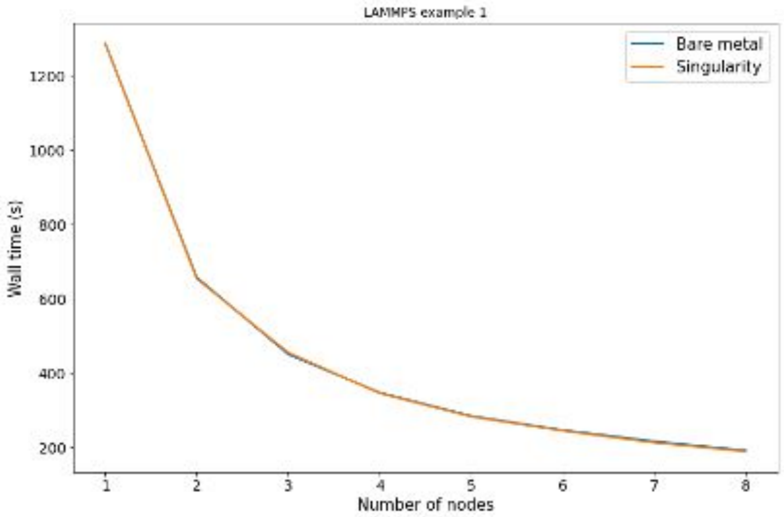
- Assumes that there is outgoing network access from the worker nodes, including port 9618

Multi-node MPI jobs

Running containerised multi-node MPI jobs using low-latency interconnects



Performance of multi-node MPI not affected by containerisation
Example: LAMMPS (used by Materials group at CCFE)



HPC systems generally now have Singularity installed

- If not, udocker can be used

Only use MPI inside the container

- MPI on the host is not used at all
- Avoids issues with MPI version conflicts

Works with OpenMPI, Intel MPI, MPICH

EOSC-hub

Challenges

- MPI, low-latency interconnects & containers
  - Installing/configuring MPI etc in the container so that InfiniBand works is not easy
  - Creating a single image which works on multiple HPC clusters is even more difficult
  - Maybe going back to using MPI on the host is the simplest solution?
    - If necessary jobs can be matched to HPC clusters supporting the appropriate MPI version/flavour
- Access to data
  - OneData & WebDAV clients don’t work on the worker nodes on many HPC systems
  - Access to data via staging-in/out from object storage does work (curl!)
- The hard bit: non-technical issues
  - We’re using a single user account on the HPC system for multiple users in PROMINENCE
  - Some (most/all?) HPC clusters have strict terms of use & policies regarding storage of ssh keys
  - Can any platform supporting multiple users to be allowed to submit jobs?
  - Effect of multi-factor authentication

Thank you for your attention!

Questions?

Contact

EOSC-hub

eosc-hub.eu @EOSC_eu