RUNNING JOBS ON HIGH PERFORMANCE COMPUTERS WITH DIRAC:

Overview of the existing solutions

D&RW 2023

October 18^{th} 2023

Alexandre F. Boyer
alexandre.boyer@cern.ch

European Organization for Nuclear Research
Meyrin, Switzerland

Introduction

DIRAC Workload Management System

- Mostly used for High-Throughput Computing (HTC) purposes.
- Able to federate a large variety of heterogeneous computing resources.
- Mainly Grid Sites. What about Clouds (see Daniela's talk) & High Performance Computers (HPC)?

HPCs can provide massive computing power

- National science programs are consolidating computing resources: a small number of very powerful computing infrastructures may become the norm in the future.
- Running HTC tasks on HPCs can be challenging and requires a significant amount of work.

What can we do, as DIRAC developers, to help you in this journey?

RUNNING HIGH THROUGHPUT COMPUTING TASKS...

Anatomy of a HTC workload

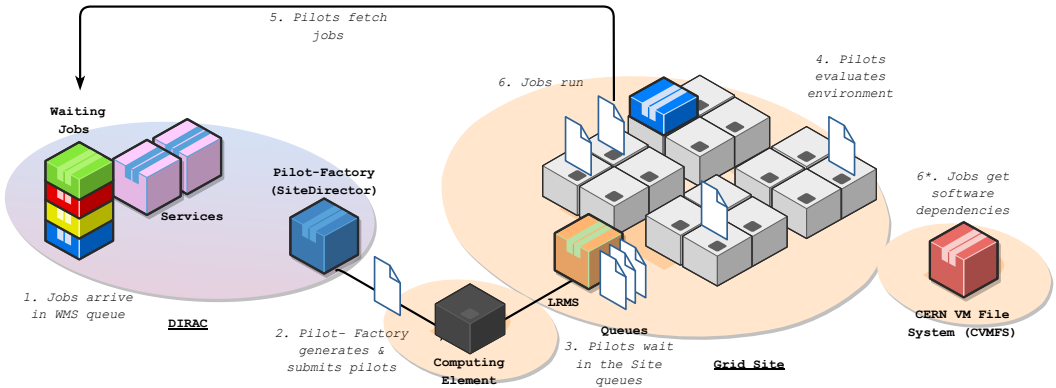
Nature

- Consists of many loosely coupled and independent tasks requiring a large amount of computing power during a long period.

Properties of a typical HTC task

- Minimal input data dependency.
- CPU-intensive task.
- Long execution time.

Executing HTC workoads across distributed computing resources



... ON HIGH PERFORMANCE COMPUTERS

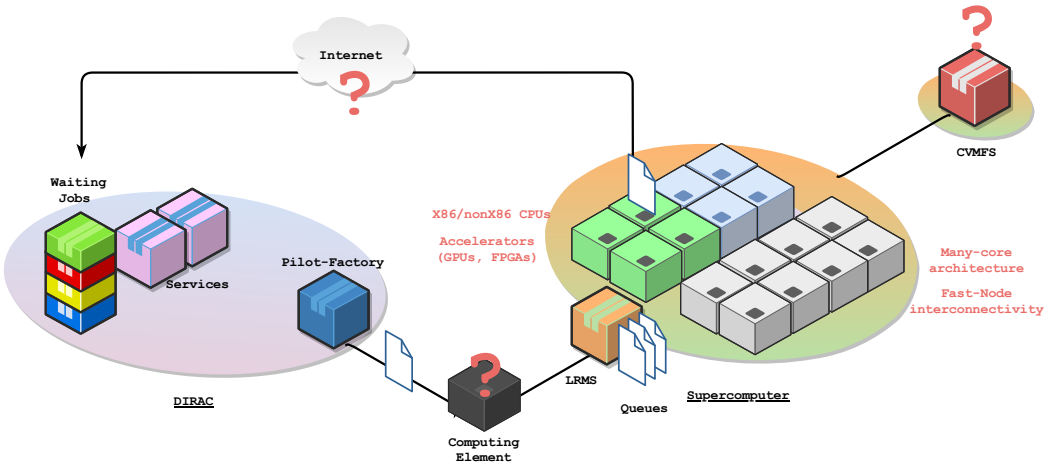
High Performance Computers: definitions and numbers

Supercomputers: the most powerful HPCs of the world

A mainframe computer that is among the largest, fastest, or most powerful of those available at a given time.

- Twice a year, top500.org releases the list of the most powerful HPCs of the world.
- #1 Frontier is composed of 8,699,904 cores.
- In comparison, WLCG provides about 1 million cores (many additional parameters have to be taken into account for a fair comparison though).

DIRAC Workload Management System & HPCs?



Challenges

Software architecture and Distributed Computing

- Software has to be flexible. HPCs may include non-x86 CPUs and accelerators.
- The DIRAC Workload Management System needs to provide the software requirements and maximize the use of the requested resources. We will focus on these aspects in the following section.

⇒ HPCs are very heterogeneous: it is impossible to produce a generic and unique solution that would work for all of them.

⇒ Goal: build small software blocks that can be added together to create a customized solution.

TECHNICAL SOLUTIONS

Solutions based on features

Features

- 1 feature directly affects the chosen paradigm:

+ Do the worker nodes have an external connectivity? Yes (or only via the head node), no.

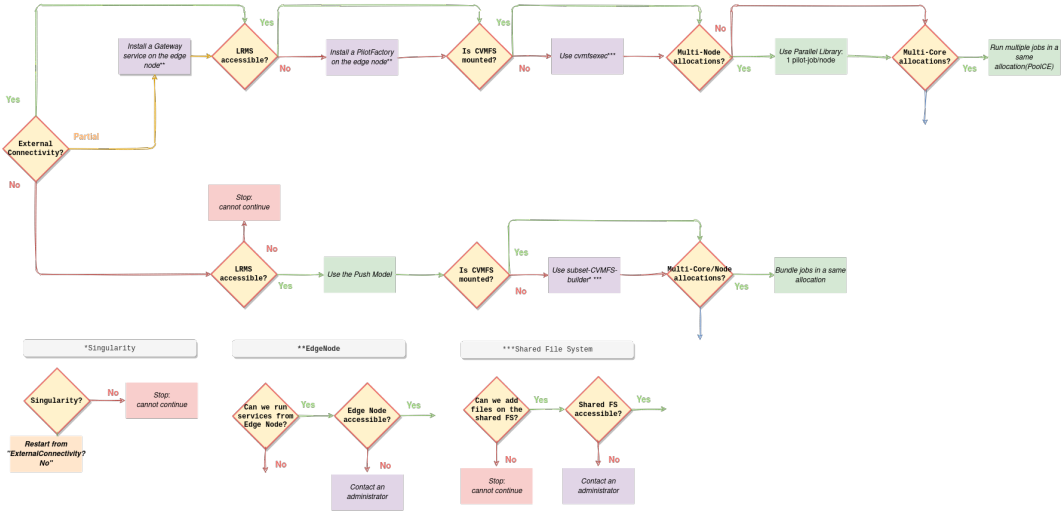
- Other features generate some technical adjustments around the chosen paradigm:

+ Is CVMFS mounted on the worker nodes? yes, no.

+ Is the Batch System accessible from outside? yes, no.

+ What type(s) of allocations can we request? Single core, multi-core, multi-node.

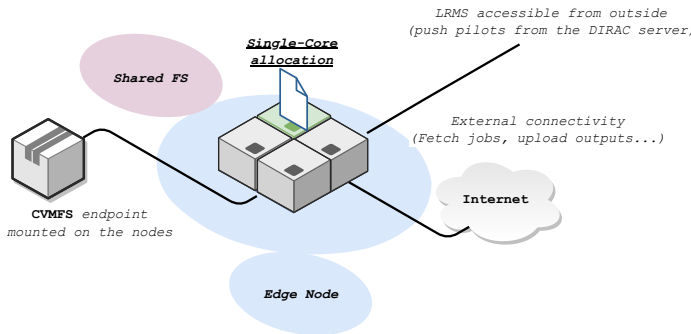
Choosing the right approach



Software solutions: Complete access to the HPC & single-core allocations

Similar to a WLCG grid site

- Uncommon for a HPC.
- Generally the result of a close collaboration with the system administrator of the HPC.



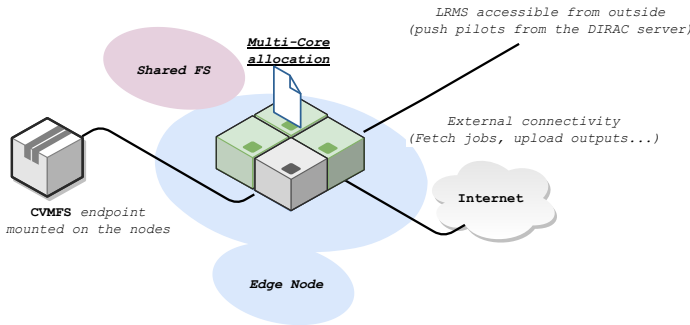
Software solutions: Complete access to the HPC & multi-core allocations

HPCs tend to favor multi-core allocations...

Node partitioning

v7r0 DIRAC

- One pilot-job for many cores on 1 node.
- The pilot-job repeats the following operations until all the available cores are occupied: fetches a job from the DIRAC services and executes it on the node.



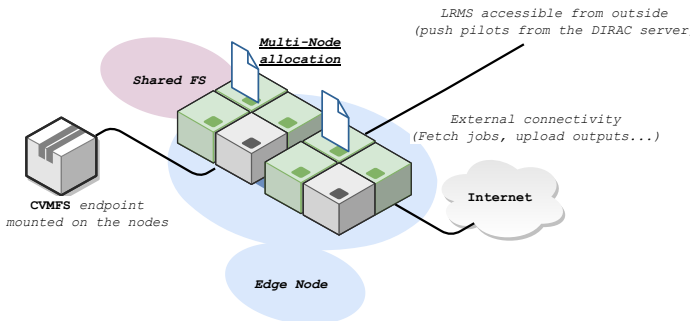
Software solutions: Complete access to the HPC & multi-node allocations

... And even multi-node allocations.

Sub-Pilots

v7r3 DIRAC

- Use of `srun` to install 1 pilot-job per node in parallel.
- The pilot-jobs share the same identifier, status and logs.
- Possibilities to request elastic allocations (e.g. between 1 and 5 nodes).

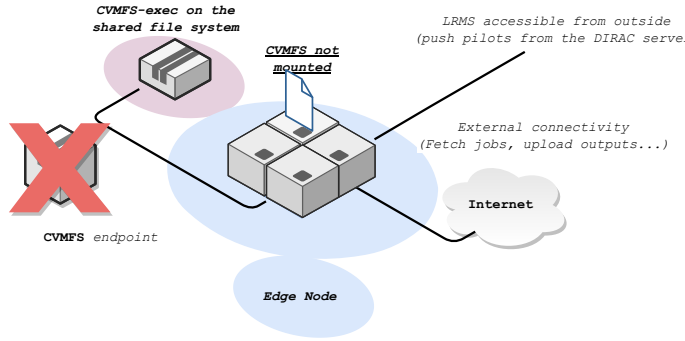


Software solutions: External connectivity but CVMFS not available

By default, HPCs do not provide access to CVMFS.

CVMFS-exec

- Client installed on the shared file system of the HPC.
- Mounts CVMFS as an unprivileged user.
- Requires actions from a DIRAC operator.

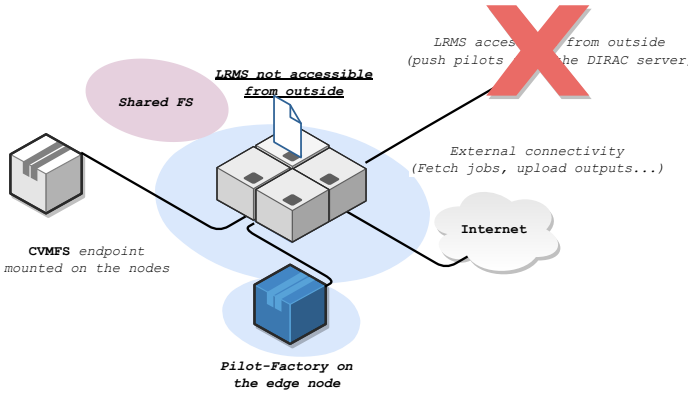


Software solutions: External connectivity but no remote access to the Batch System

Some HPCs can only be accessed via a VPN (No CE, no direct SSH access).

Pilot factory installed on a head node DIRAC

- Pilot-Jobs are directly submitted from the HPC.
- Requires actions from both a system administrator of the HPC (getting the certificate, authorizing cron jobs), and a DIRAC operator (installing the Pilot factory).



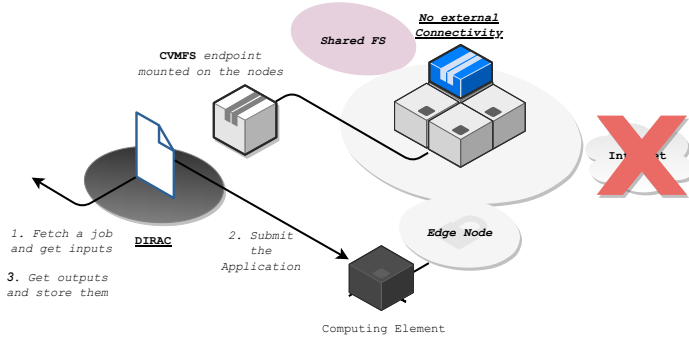
Software solutions: No external connectivity...

Some HPCs do not allow jobs to access external services.

PushJobAgent

v7r3 DIRAC

- Works as a Pilot-Job that would be executed outside of the HPC.
- Fetches jobs, manages their input and output data, and solely submits the application to the HPC.
- Requires a direct access to the Batch System.

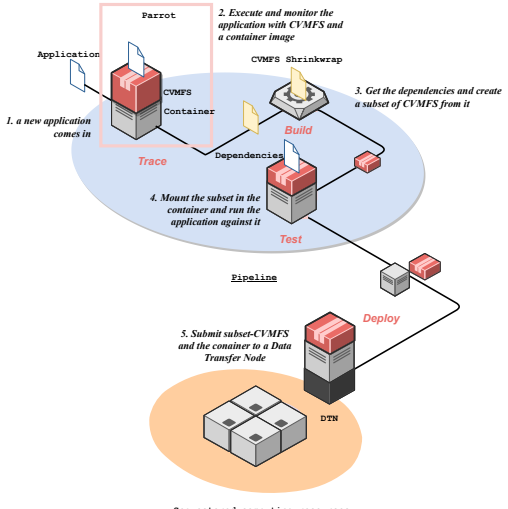


Software solutions: No external connectivity, so no CVMFS

In this context, we cannot leverage CVMFS-exec.

subcvmfs-builder

- Generic solution to create and deploy subsets of CVMFS.
- Takes the form of a Python package and a continuous integration pipeline.
- Example: extracting Gauss dependencies (a few GB) in 2h30: **https://gitlab.cern.ch/lhcb-dirac/subcvmfs-builder-pipeline**



USE CASES (LHCB)

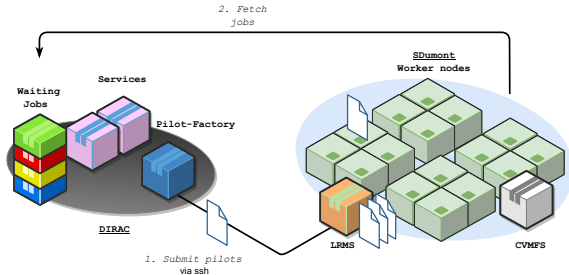
SDumont, LNCC: Development

Features

- Opportunistic resources.
- 36,472 CPU cores, distributed across 1,134 compute node.
- 24 cores and 64Gb of RAM per node.

Environment

+ External connectivity, CVMFS mounted on the nodes.
- Protected by a VPN, but LHCb-specific SSH access available.
- Multi-core and multi-node are preferred.



SDumont, LNCC: Status

Set up the following solutions

- Sub-pilots and node partitioning.
- Test: Pilot factory installed on one of the head node.

Results

- A single-core job on every logical cores available per allocation.
- Elastic allocation: we request a time interval and a variable number of nodes.

Problems & Considered approaches

- Inaccurate CPU work estimates: a lot of our jobs run out of time.

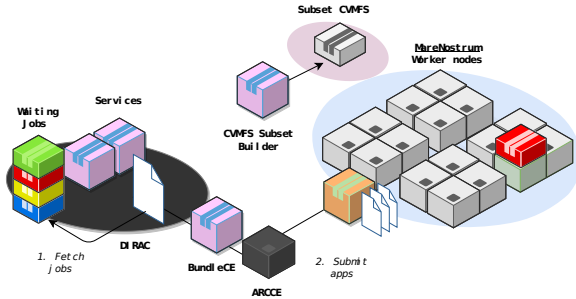
Mare Nostrum, BSC: Development

Features

- 4-month allocations of CPU hours.
- 153,216 cores distributed across 3456 nodes.
- 48 cores and 96Gb of RAM per node.

Environment

+ Access through a CE
- Single-core allocations possible but not preferred.
- No external connectivity, no access to CVMFS.



Mare Nostrum, BSC: Status

Set up the following solutions

- `PushJobAgent` to push jobs.
- `SubCVMFS-Builder` to generate and deploy up-to-date subsets of CVMFS.

Results

- One job per single-core allocation.
- 300 jobs in parallel.
- The subset of CVMFS is regularly updated: no major issue so far.

Problems & Considered approaches

- `PushJobAgent` is simple but consumes a lot of memory: cannot scale.
- Reducing the memory consumption implies important changes within the LHCbDIRAC extension.

CONCLUSION

Conclusion

Main contribution

- Methods and software blocks to integrate HTC tasks on HPCs (constrained environments).
- May benefit to any VOs using DIRAC.

What's next?

- We should mainly focus on reducing the memory footpring of the `PushJobAgent` in the next few months.
- In the meantime, you can already access the documentation to set up the solutions on your side: **https://dirac.readthedocs.io/en/latest/AdministratorGuide/Resources/supercomputers.html**

Thank you for your attention

Questions? Comments?