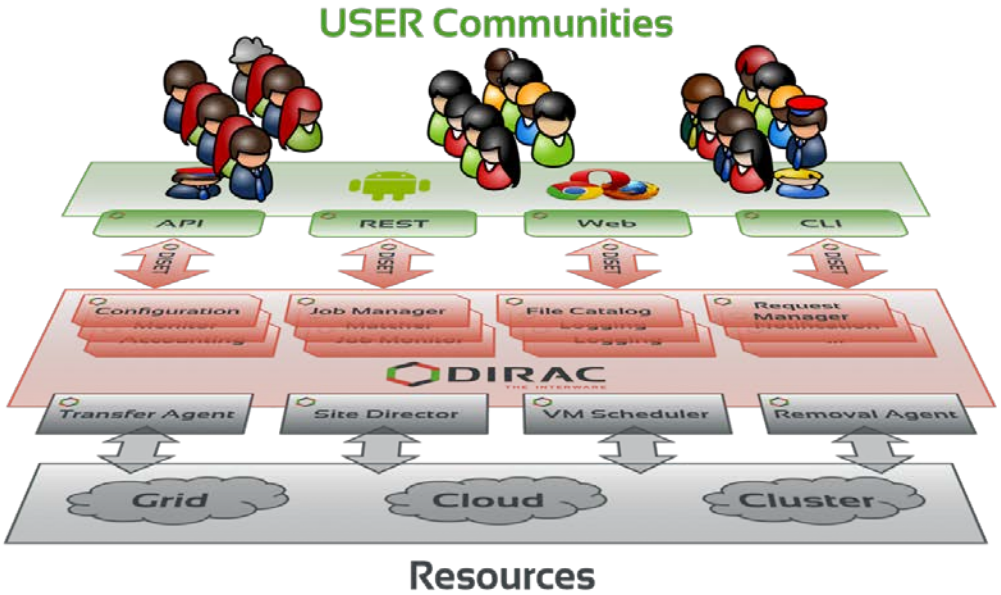
DIRAC SSH CE

Topics

- Introduction
- DIRAC WMS Quick Overview
- DIRAC SSH Computing Element
- Accounting
- Conclusions

DIRAC: the interware

- A software framework for distributed computing
- A **complete** solution to one (or more) user community
- Builds a layer between users and resources

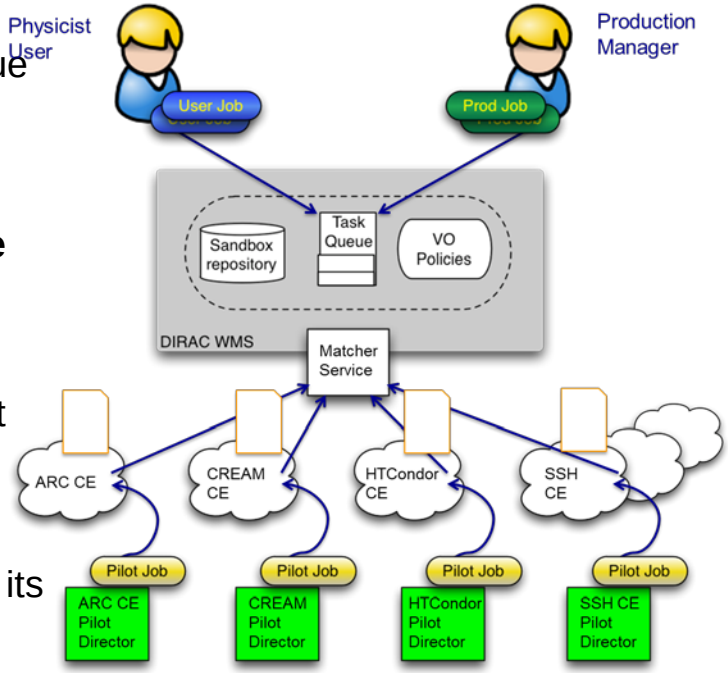


An open source project

- Started as an LHCb project, became experiment-agnostic in 2009
  - First users (after LHCb) end of 2009
- Developed by communities, for communities
  - Open source (GPL3+), GitHub hosted, python 2.7
  - No dedicated funding for the development of the “Vanilla” project
  - Publicly documented, active assistance forum, yearly users workshops, open developers meetings
  - 4 FTE as core developers, a dozen contributing developers
- The DIRAC consortium as representing body
  - CNRS, CERN, University of Barcelona
  - IHEP, KEK, PNNL, University of Montpellier

Job scheduling

- **Users submit jobs**, which are stacked in the DIRAC Task Queue
- DIRAC Pilot Directors submit **Pilot jobs** to computing resources
- After the start, Pilots check the execution environment and requests a job from the **Matcher service** providing the **resource description**
  - OS, capacity, disk space, software, etc
- The Matcher service selects the appropriate user job for the pilot
  - Matching based on (i) the resources description and (ii) job requirements
- The user job description is delivered to the pilot, which prepares its execution environment and executes the user application
- At the end, the pilot uploads the results and output data

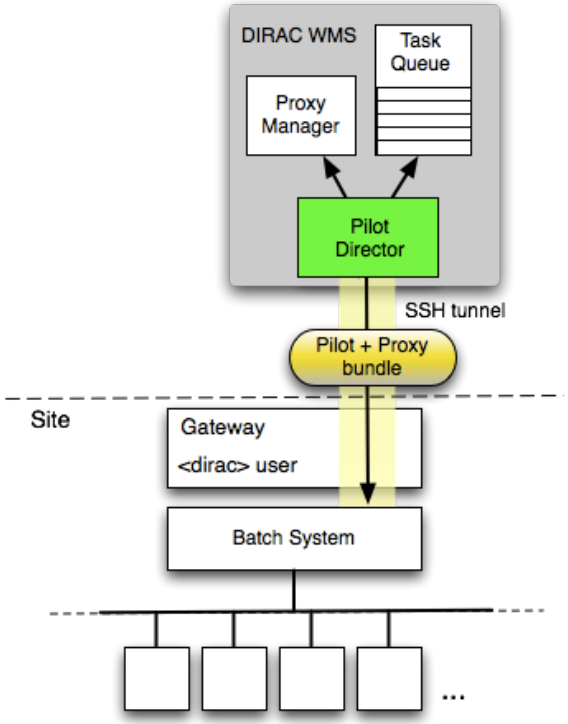


Pilot based WMS

- Efficient job matching to the site properties
  - User jobs submitted to the system are not passed immediately to a selected site but wait in the central repository – Task Queue
- Important decrease of jobs failure rate
  - Users' payload starts in an already verified environment
- No need for resource providers to distinguish individual users
  - Simplifies site management but needs special trust between the site and the community
- Users do not need to bother about the backend
  - Adding/removing sites and/or protocols is handled by Dirac admins
  - Users do not need to know about Cream CE :-)

Standalone computing clusters

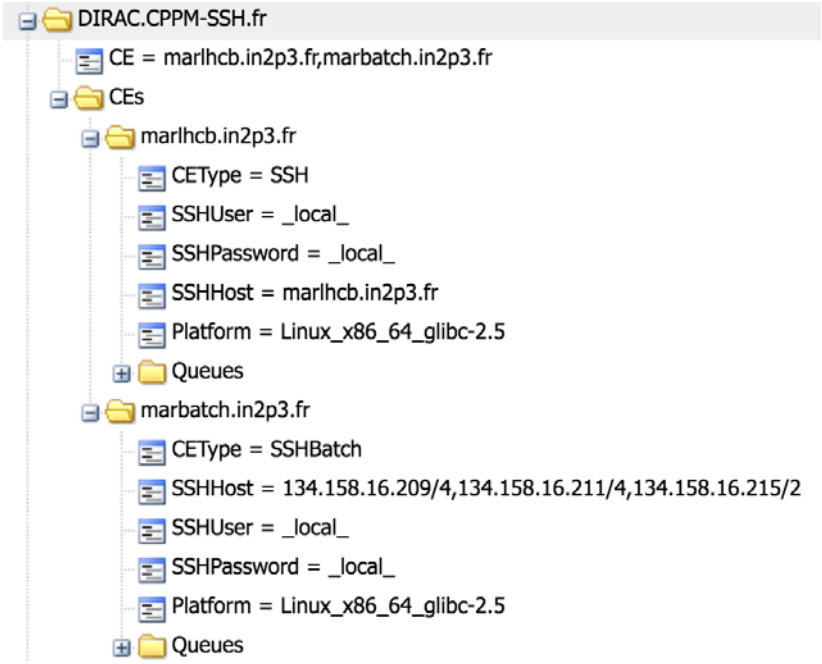
- Off-site Pilot Director
  - Site must only define a dedicated local user account
  - The payload submission through an SSH tunnel
- The site can be
  - Single computer or several computers without any batch system
  - Computing cluster with a batch system
- Pilots are sent as an executable self-extracting archive with the pilot proxy bundled in
- The user payload is executed with the owner credentials
  - No security compromises with respect to external services



SSH CE examples

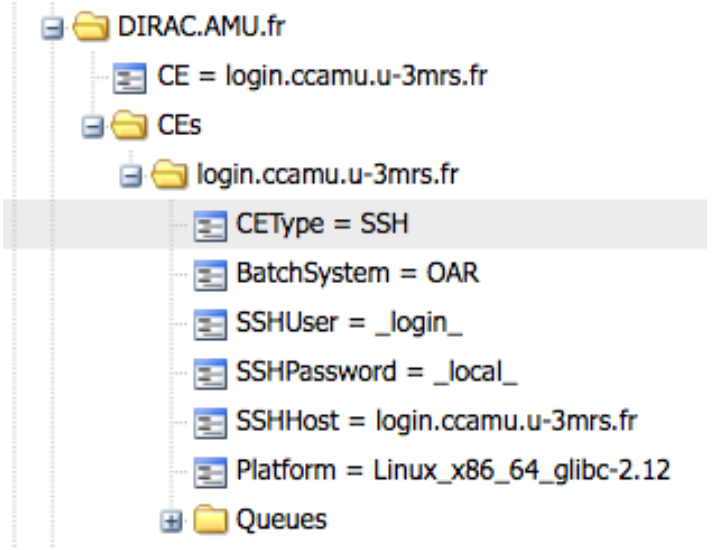
- SSH CE simplest case
  - One host with one job slot

- SSHBatch CE
  - Several hosts form a CE
    - Same SSH login details
    - Number of job slots per host can be specified



SSH CE with a batch system

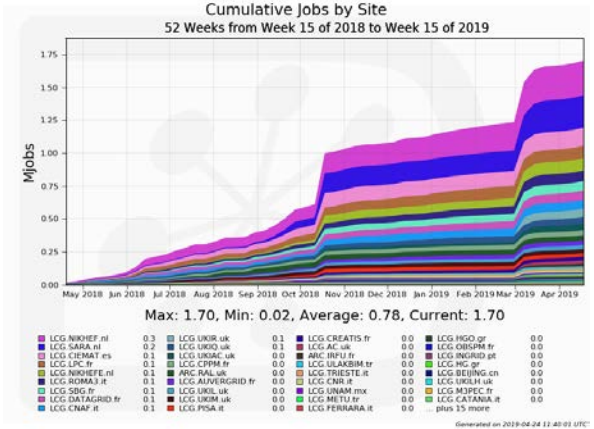
- SSH login to the cluster interactive host
  - Copy several tools, e.g. BatchSystem plugin at the first time
- Submit pilots to the local cluster using a relevant BatchSystem plugin
  - Condor, GE, LSF, Torque (HTC)
  - SLURM, OAR (HPC)
- Site admins only need to allow ssh connexion
- Transparent for DIRAC end users

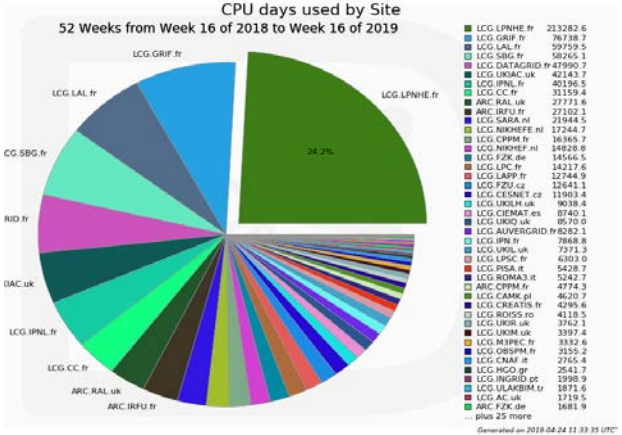


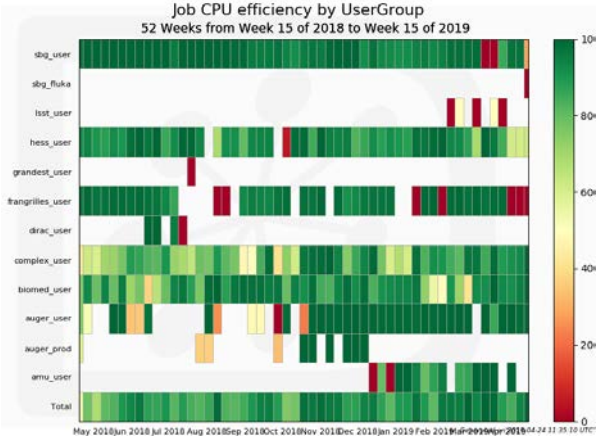
SSH CE standalone

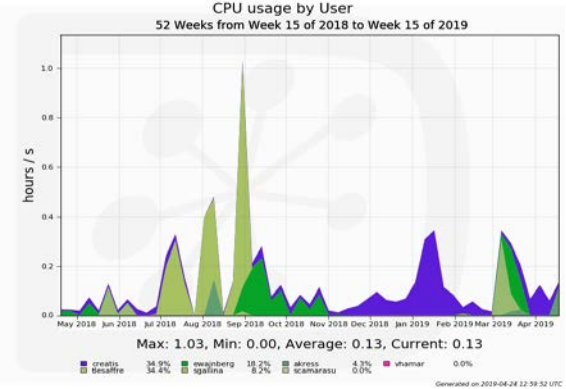
- Can it be used as a standalone tool and not as part of some DIRAC system ?
  - In principle,yes
  - But it makes little sense to consider CEs apart from an orchestration service

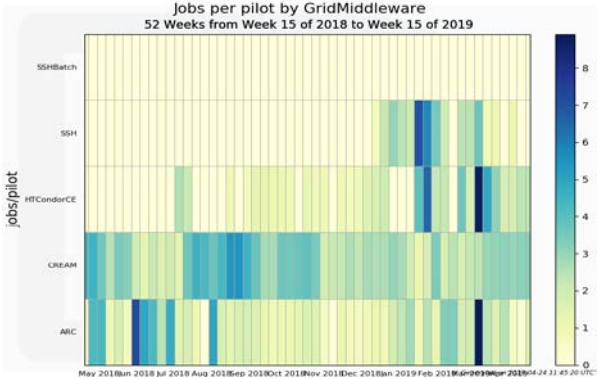
Accounting

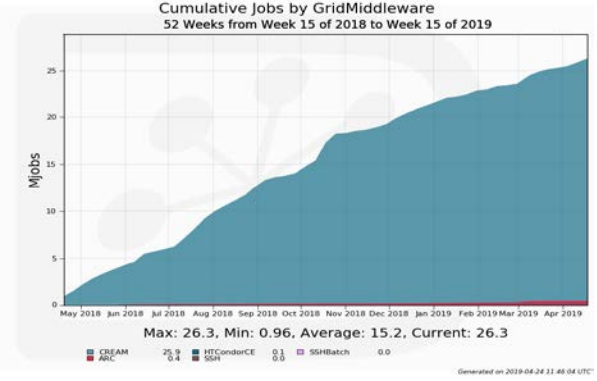

Cumulative Jobs by Site
52 Weeks from Week 15 of 2018 to Week 15 of 2019
Mjobs
May 2018 Jun 2018 Jul 2018 Aug 2018 Sep 2018 Oct 2018 Nov 2018 Dec 2018 Jan 2019 Feb 2019 Mar 2019 Apr 2019
Max: 1.70, Min: 0.02, Average: 0.78, Current: 1.70


CPU days used by Site
52 Weeks from Week 16 of 2018 to Week 16 of 2019
LCG.GRIF.fr
LCG.LAL.fr
LCG.LPNHE.fr
24.2%


Job CPU efficiency by UserGroup
52 Weeks from Week 15 of 2018 to Week 15 of 2019
sbg_user
sbg_fluka
lsst_user
hess_user
grandest_user
frangrilles_user
dirac_user
complex_user
biomed_user
auger_user
auger_prod
amu_user
Total


CPU usage by User
52 Weeks from Week 15 of 2018 to Week 15 of 2019
hours / s
Max: 1.03, Min: 0.00, Average: 0.13, Current: 0.13


Jobs per pilot by GridMiddleware
52 Weeks from Week 15 of 2018 to Week 15 of 2019
Jobs/pilot
SSHBatch
SSH
HTCondorCE
CREAM
ARC


Cumulative Jobs by GridMiddleware
52 Weeks from Week 15 of 2018 to Week 15 of 2019
Mjobs
Max: 26.3, Min: 0.96, Average: 15.2, Current: 26.3

Conclusions

- Framework for building distributed computing systems
  - In particular, Pilot-based Workload Management System
- Aggregates multiple types of computing resources
  - e.g., ARC, CREAM CE, HTCondor, SSH CE
  - If considered of interest, DIRAC SSH CE could be exposed for direct end-user usage
- Transparent access to these resources for end users

Thank you for your attention!

Questions?

CE service functions (I)

- Status of the local batch system : OK
- User identification and authorization according to local policies
  - Unix user per DIRAC community - community can be banned as a whole in case of user misbehavior;
  - Possibility to use glexec mechanism to check user payloads against locally defined policies
- Job submission : OK

CE service functions (II)

- Job execution
  - Delivery of user credentials to the worker nodes
    - OK, part of the executable bundle
  - Delivery of input sandboxes
    - OK, scp to the gatekeeper before local submission
- Job monitoring : OK
- Job results retrieval (output sandboxes) : OK, by scp
- Resources consumption accounting
  - DIRAC accounting or
  - DIRAC reporting to locally used accounting service