Centre de Calcul
de l'Institut National de Physique Nucléaire
et de Physique des Particules

Distributed Computing with DIRAC Interware

V. HAMAR

Marseille, 20/06/2022

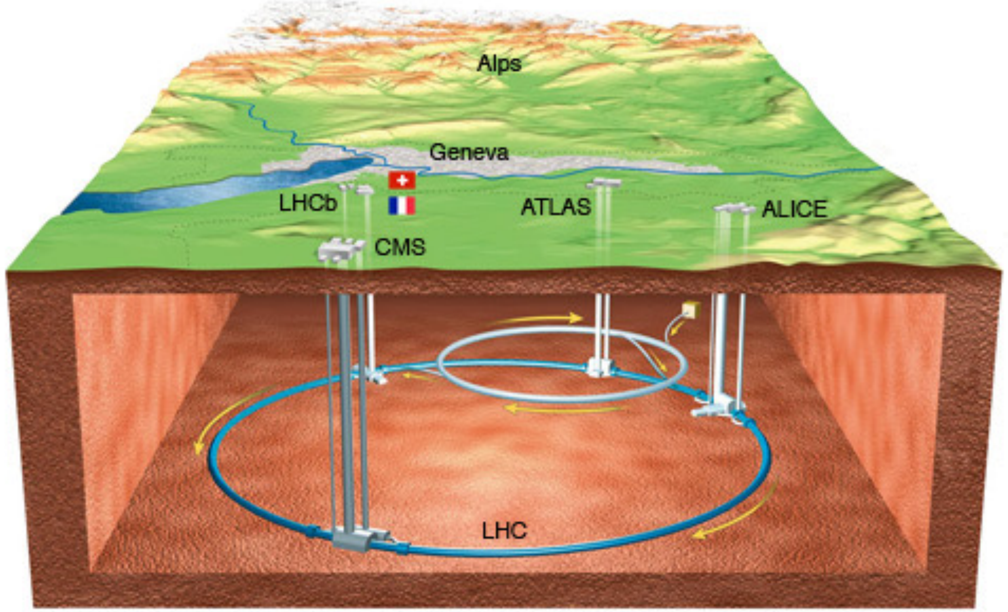
Overview

- Brief history of computing grids for HEP
- DIRAC as the LHCb's solution for distribute computing
- DIRAC – what makes it unique
    - Pilot based WMS architecture
    - Complete solution
    - Open architecture, open-source project
    - Support for multiple communities
- Conclusions

Brief history of computing grids for HEP

- Large Hadron Collider (LHC) Project
  - Scientist started to think about LHC in the early 1980s
  - CERN Council voted to approve the construction of the LHC in December 1994
  - LHC technical design report was published in October 1995

- Experiments approved between 1996 and 1998
  - 31 January 1997 CMS and ATLAS experiments
  - 14 February 1997 ALICE experiment
  - 17 September 1998 LHCb experiment



Brief history of computing grids for HEP

LHC Computing Project

- The centralized model used until then at CERN could not apply to the LHC
  - Very large datasets will be collected, the processing and analysis of the data was the biggest challenge.
- A review in ‘90s concluded that computing resources (CPU and storage) required were far beyond what could be provided by only one site.
- Solution – LHC dedicated computing grid

Brief history of computing grids for HEP

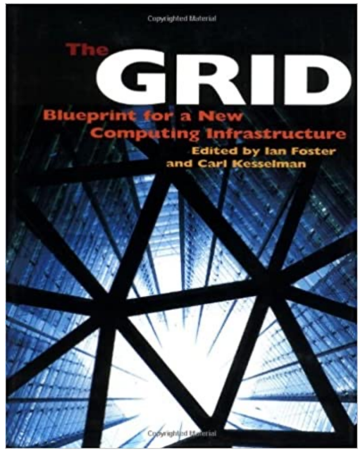
Ian Foster and Carl Kesselman. The Grid: Blueprint for a New Computing Infrastructure. 1998

“The Grid is an emerging infrastructure that will fundamentally change the way we think about - and use – computing. The word grid is used by analogy with the electric power grid, which provides pervasive access to electricity and, like the computer and a small number of other advances, has had a dramatic impact in human capabilities and society.”

“… coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations …”

Grid in a nut-shell – distributed computing system with :

- common middleware, common protocols to access computing and storage resources
- common conventions on resource usage policies



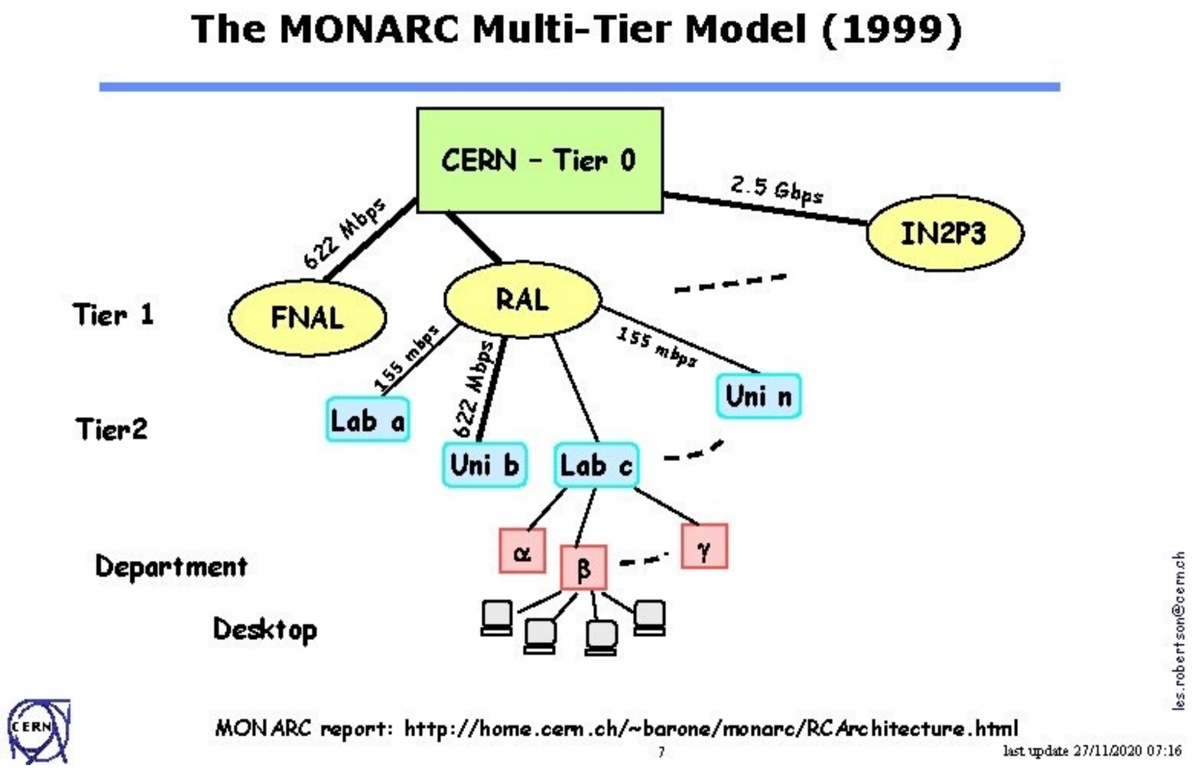
https://www.amazon.com/Grid-Blueprint-Computing-Infrastructure-Elsevier/dp/1558604758

Brief history of computing grids for HEP

Models of Networked Analysis at Regional Centres (MONARC) for LHC Experiments (1988-1999)

Goals of the Project

- A set of feasible Models for the Computing of LHC Experiments
- Guidelines for the Experiments in building their Computing Models


The MONARC Multi-Tier Model (1999)

MONARC report: http://home.cern.ch/~barone/monarc/RCArchitecture.html

7

last update 27/11/2020 07:16

TOPOLOGY

https://www0.mi.infn.it/~perini/monarc_pep/sld003.htm
https://slidetodoc.com/lhc-computing-grid-project-grid-pp-collaboration-meeting

Brief history of computing grids for HEP

LHCC Recommendations (2000)

- A multi-Tier hierarchical model similar to that developed by the MONARC project should be the key element of the LHC computing model.
- Grid Technology will be used to attempt to contribute solutions to this model that provide a combination of efficient resource utilisation and rapid turnaround time.
- Estimates of the required bandwidth of the wide area network between Tier0 and the Tier1 centres arrive at 1.5 to 3 Gbps for a single experiment.
- Joint efforts and common projects between the experiments and CERN/IT are recommended to minimise costs and risks.
- Data Challenges of increasing size and complexity must be performed as planned by all the experiments until LHC start-up.

https://lhcb-comp.web.cern.ch/Reviews/LHCComputing2000/Report_final.pdf

Brief history of computing grids for HEP

The idea: Use a large amount of resources distributed geographically as one big resource.

… sites are heterogeneous (cpu models, batch systems, etc.) and support local users.

Middleware hides this heterogeneity, providing uniform protocols to access resources.

… the challenge : submit a job and find the best place to run this job, taking into account the data associated, the shortest time to start…

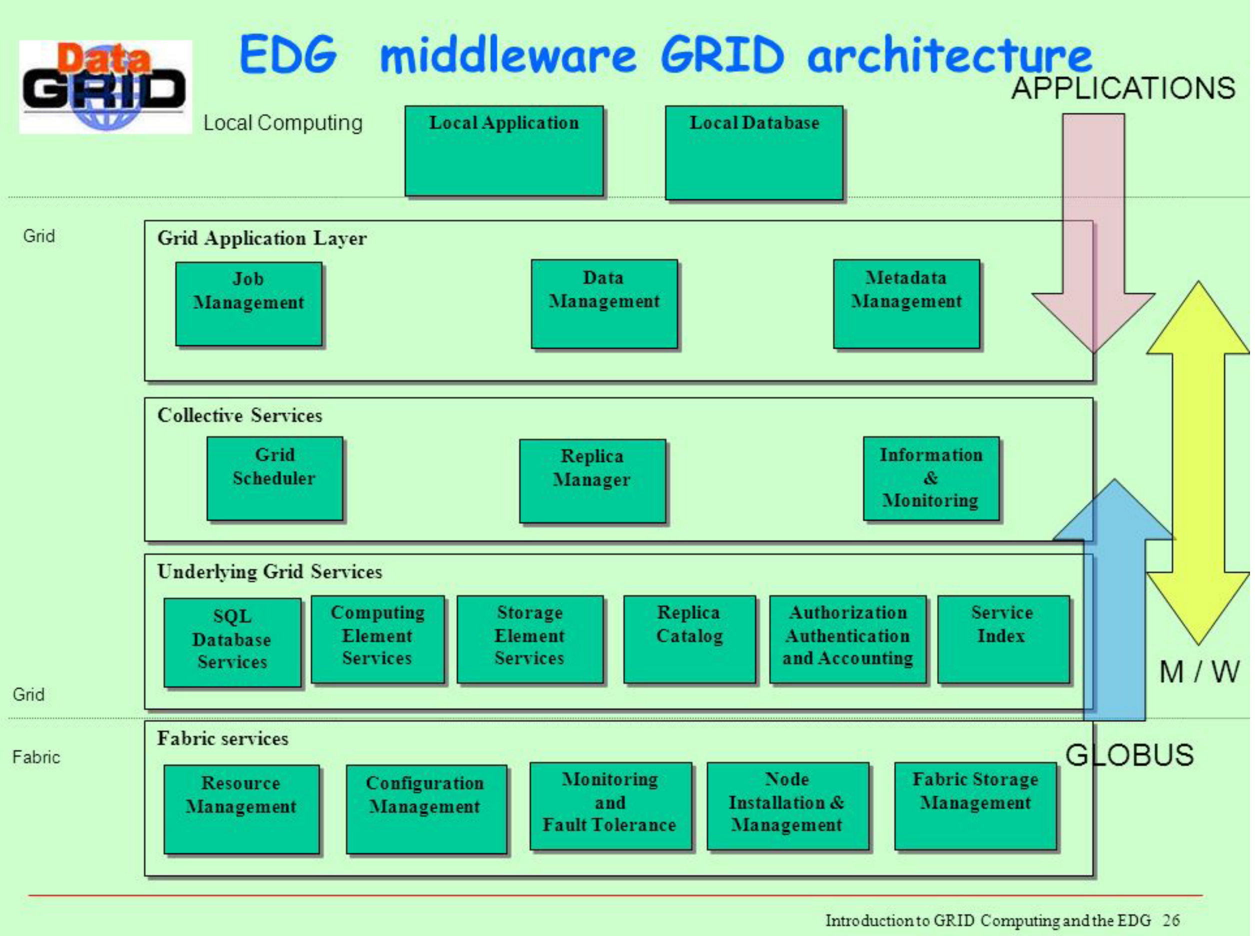
Now looks trivial … but at that time it was a revolution!

Brief history of computing grids for HEP

DataGrid (2001-2003)

- Exploit and build the next generation computing infrastructure providing intensive computation and analysis of shared large-scale databases.
- Middleware development.
- 16 services running in the testbed, some adapted from Globus 2 toolkit.

DataGrid didn't satisfy production level requirements 🙁



EDG middleware GRID architecture

APPLICATIONS

Local Computing: Local Application, Local Database

Grid

Grid Application Layer: Job Management, Data Management, Metadata Management

Collective Services: Grid Scheduler, Replica Manager, Information & Monitoring

Underlying Grid Services: SQL Database Services, Computing Element Services, Storage Element Services, Replica Catalog, Authorization Authentication and Accounting, Service Index

M / W

Grid

Fabric

Fabric services: Resource Management, Configuration Management, Monitoring and Fault Tolerance, Node Installation & Management, Fabric Storage Management

GLOBUS

Introduction to GRID Computing and the EDG 26

https://slideplayer.com/slide/8630667/

https://www.slideserve.com/dotty/grid-computing-at-cern-powerpoint-ppt-presentation

Brief history of computing grids – First steps

LHC Computing Grid (LCG) Project (2002-2007)

- Approved by CERN council in 2001 - In production in 2003
- Goal: to prototype and deploy the computing environment for the LHC experiments.
- Collaboration between:
  - LHC Experiments
  - Regional Computing Centers
  - Physics institutes
- Support for applications:
  - Common tools, software, frameworks, environments, data persistency.
  - Exploiting resources available to LHC experiments
  - gLite middleware (EU Prototype).

https://indico.cern.ch/event/415310/contributions/997358/attachments/849192/1183574/LHCC_comprehensive_review_031124_MKa.pdf
https://wlcg.web.cern.ch/LCG/

Brief history of computing grids for HEP

Data challenge (2004)

- Test and validate the computing models
- Produce simulated data
- Test experiments production frameworks and software
- The four experiments participated

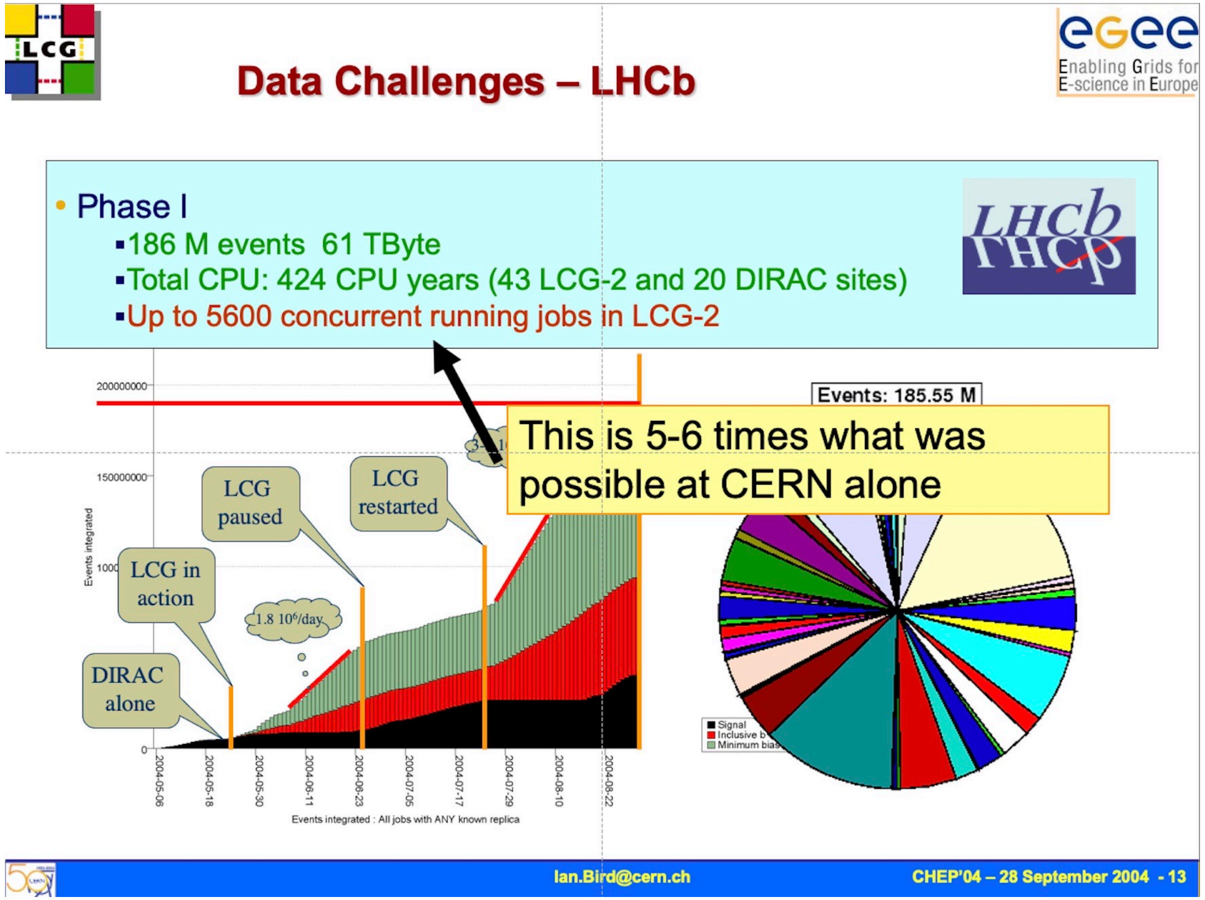
DIRAC results during the Data Challenge in 2004 shows that the Grid can face the LCG project challenge.

First success!

Brief history of computing grids for HEP

Why it was a success?

- DIRAC job user efficiency > 90%
  - while ~60% success rate of LCG jobs.
- The first production system to put in the grid job an script to pull jobs from a queue
  - Sending agent as regular jobs
  - Now known as pilot jobs
- Record of maximum running jobs
  - And orders of magnitude less than what we can do now !
- The scalability of the system allowed to saturate all available resource.



Data Challenges – LHCb

- Phase I
  - 186 M events 61 TByte
  - Total CPU: 424 CPU years (43 LCG-2 and 20 DIRAC sites)
  - Up to 5600 concurrent running jobs in LCG-2

This is 5-6 times what was possible at CERN alone

Ian.Bird@cern.ch CHEP'04 – 28 September 2004 - 13

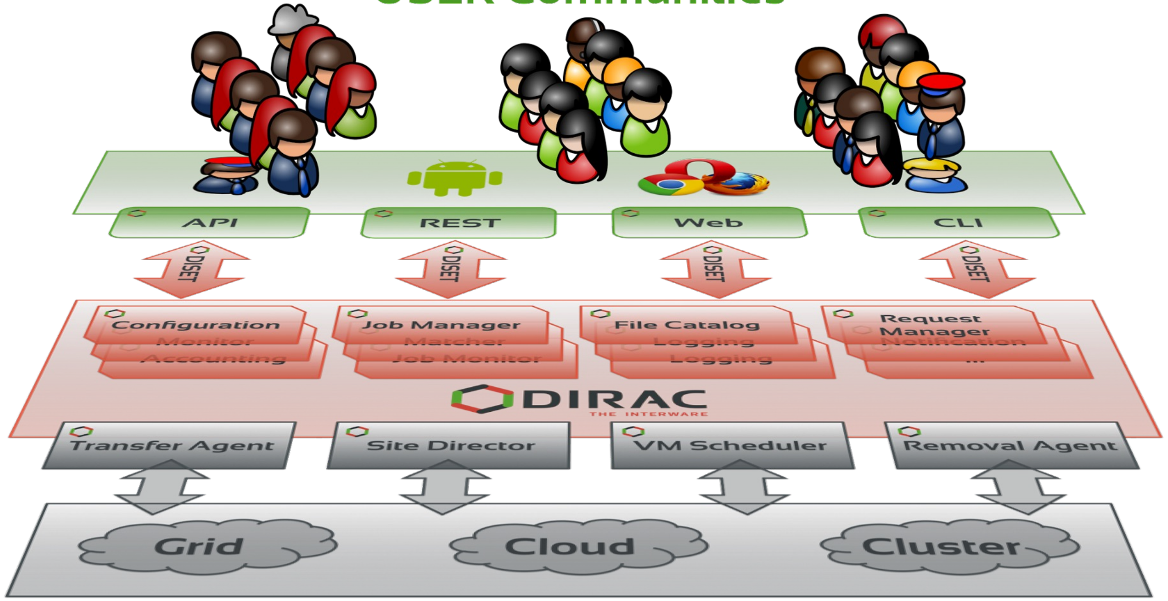
DIRAC

THE INTERWARE

What’s the DIRAC Interware?

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

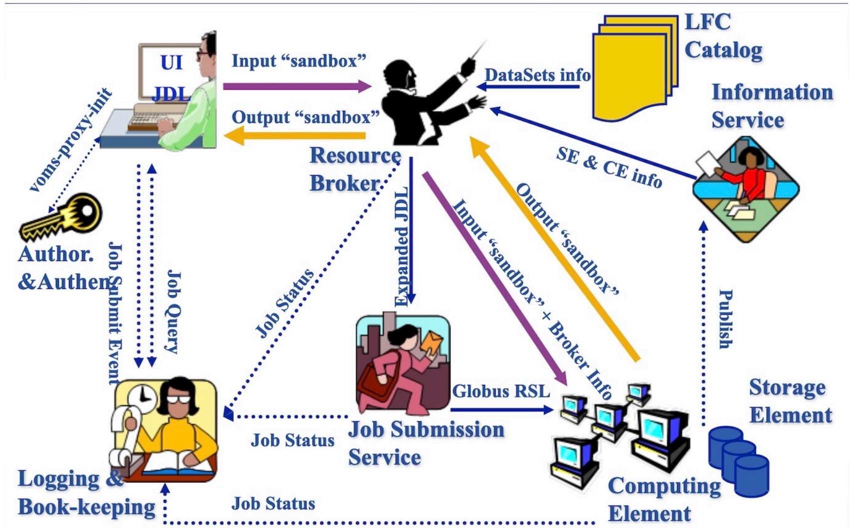
USER Communities



Resources

Centralized WMS architecture

Job Workflow in gLite with centralized WMS architecture



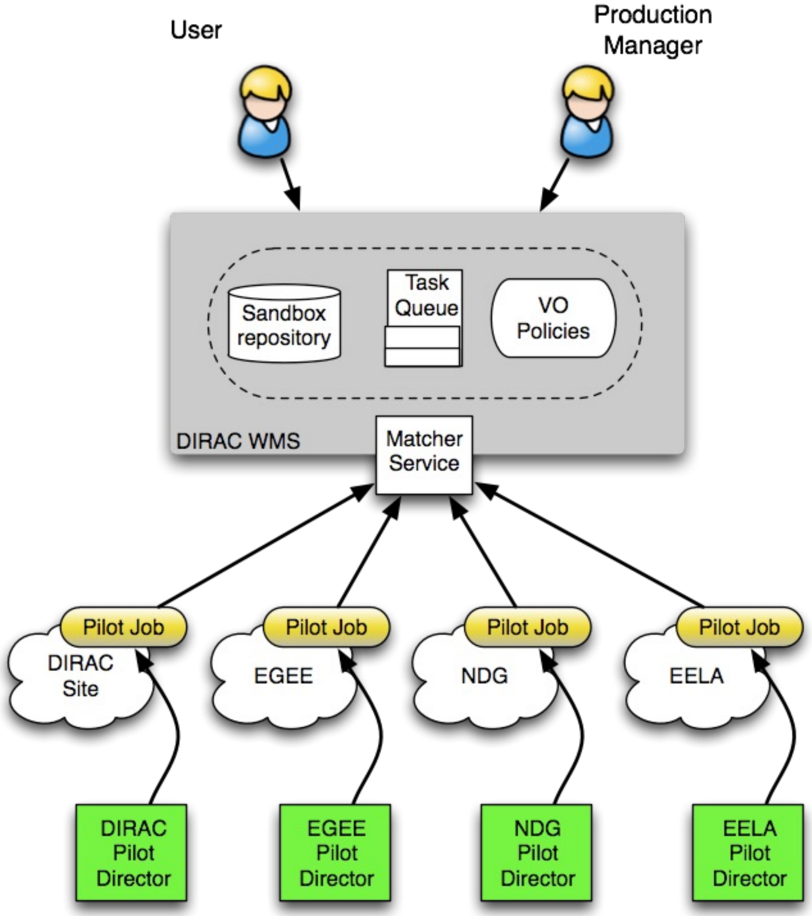
- Operational information is collected for the central Resource Broker (RB)
  - Site capacity and status
  - Data placement
- For each submitted job the RB makes a decision of dispatching it to the most appropriate site
  - Meeting job requirements
  - Least loaded
- An example of *PUSH* scheduling paradigm

WMS architecture with pilot jobs

- Pilot jobs are submitted to computing resources by specialized Pilot Directors using specific access protocols

- Pilots pull user jobs from the central Task Queue and steer their execution on the worker nodes including final data uploading

- An example of *PULL* scheduling paradigm



DIRAC – what makes it unique?

Scalability – the main problem of early grid systems

- Why Resource Brokers were not scalable?
  - Delays in the site status propagation – taking scheduling decisions based on obsoleted data
    - E.g., “black hole” sites falsely reporting that they are “free”
  - Compares each job description with each site to try to find the best match
    - number of jobs X number of sites matching operations require too much computations
  - As a result – necessity to run multiple central RB’s in parallel!

- How DIRAC resolved the scalability problem?
  - Sites are actively seeking jobs
    - If a pilot requests jobs, it means a free slot is ready for use.
  - Drastically less matching operations
    - Matching jobs only for requesting sites
    - Grouping similar jobs and matching by group
  - One Matcher service can handle all the payloads

DIRAC – what makes it unique?

Advantages of the PULL scheduling compared to PUSH:

- User job efficiency : in case of problems in the execution environment on a worker node the pilot job will stop without taking user jobs
- The load balancing is also achieved naturally since the more powerful resource will simply request jobs more frequently
- Expanding resources : It is easier to incorporate new production sites since little or no information about them is needed at the central production service.

DIRAC – what makes it unique?

Advantages of the pull scheduling:

- Centralized policy application.
  - Using tags for sites description and user jobs allows DIRAC to apply centralized policies.
  - Example: Biomed and Covid-19 jobs running in OSG resources.

- Is possible to apply jobs priorities.
  - Central Task Queue gives a general view of all the user payloads which allows to assign relative probabilities for different jobs, i.e. priorities. For example
    - By the users to their jobs
    - Between the different user groups

- Allows to manage heterogeneous resources. Pilot jobs are universal federators!

DIRAC – what makes it unique?

- Following DIRAC success, Atlas and CMS adopted job pilots based systems
  - Alice is using pilots in their gateways.

- The pilot jobs can be submitted directly to the compute elements or RBs.
  - WMS (last version of the RB PUSH scheduling) was decommissioned some years ago.
  - Directly submission to CEs is the only method used today!

DIRAC – what makes it unique?

DIRAC provides a complete solution to user communities

- DIRAC combines various distributed computing and storage resources in a coherent system seen by the user as a single large computer.
  - Data and Workload Management System released in a single software stack,
  - Developed in the same style and language, maintained and deployed with the same procedures and tools
  - Small production teams can run DIRAC services

- Data Management System (DMS)
  - Data is partitioned in files
  - File replicas are distributed over a number of Storage Elements world wide
  - Files are registered in a File Catalog in a single logical name space
    - With metadata and ACL info
  - For most of applications the file access is as simple as file directory

Converting DIRAC software into an open-source project

CCIN2P3

- Started as an LHCb project, experiment-agnostic in 2009
- Developed by communities for communities (HEP, astronomy and life science)
  - Open source (GPL3+), GitHub hosted.
  - Publicly documented.
  - Users workshops.
  - Developers meetings.
  - Hackathons.

LHCb
PROJECT 8
BESIII
JUNO
GridPP
UK Computing for Particle Physics
we-nmr
clic
ilc
cta
cherenkov telescope array
CEPC
NA62
OpenMOLE
the model exploration software
Belle II
3D EISCAT
T2K
SNO+
Fermi
Gamma-ray Space Telescope
JENNIFER2
EU grant n.822070
PIERRE AUGER OBSERVATORY
EGI-ACE
FUTURE CIRCULAR COLLIDER
SKA
SQUARE KILOMETRE ARRAY
NICA

DIRAC
THE INTERWARE

Converting DIRAC software into an open-source project

Behind the scenes:

- A lot of effort ! More than one year of work to re-engineering into a generic framework capable to serve the distributed computing needs of different Virtual Organizations.
    - All the DIRAC systems comprises a generic part which can be extended in a VO specific part
        - There are clearly recipes how write and release extensions which can be discovered and loaded at run time.
    - Some services can run without any extension, DIRAC core functionalities are rich enough.
    - Each DIRAC instance can decide which extensions to install
        - LHCb, ILC, Belle extensions were the first developed
    - DIRAC web portal also can be extended
        - VO specific customizations and applications

Converting DIRAC software into an open-source project

Being experiment agnostic advantages:

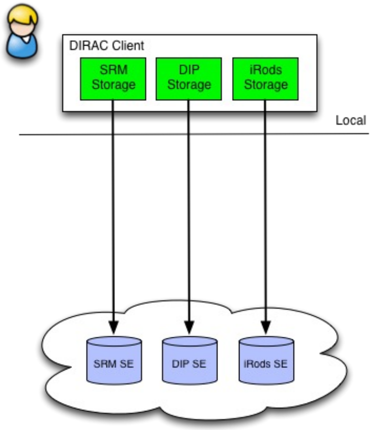
- Allows community developers to contribute to the project.
- Allows the communities to profit from developments by other communities.
  - Example: DIRAC File Catalog (DFC) was developed initially for ILC and BES experiments, actually is a plugin used by several experiments including LHCb.
    - Logical File Catalog (LFC) was plain text, while DFC has a hierarchical structure.
    - The file catalogs must be a centralized service to maintain the data integrity.

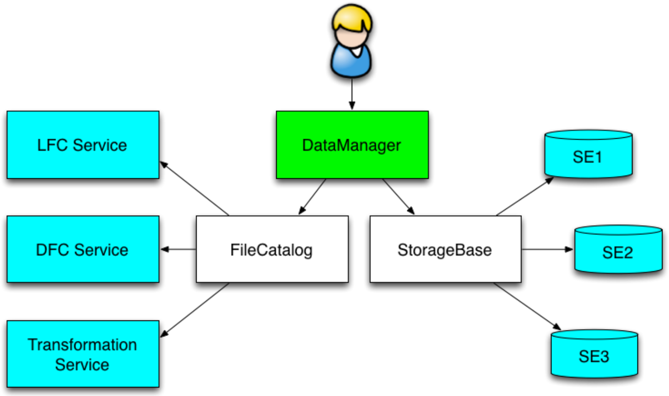
DFC was faster than LFC! LFC is now decommissioned.

https://inspirehep.net/files/08956bbcf68278ce9c76b519341e9187

Converting DIRAC software into an open-source project

- Conceived for extensions
  - Plugins everywhere
    - The project also features a modular architecture allowing users to provide custom plugins for specific functionalities.
      - DIRAC supports flexible ACL rules with plug-ins for various policies that can be adopted by a particular community.
      - Plugins allows DIRAC to communicate with different kinds of computing resources
      - Example of Data Management plug-ins





Converting DIRAC software into an open-source project

DIRAC Consortium

- Created in 2017.
- Goal: support for development, maintenance and promotion of the DIRAC Interware.
- Current members:
    - CNRS, CERN, IHEP, KEK, Imperial College, Université de Montpellier
- The Consortium holds the copyright for the DIRAC software
    - GPL v3
- Organizes workshops, tutorials and other events to promote DIRAC.
    - Last DIRAC User Workshop: https://indico.cern.ch/event/1107386/
    - You can find presentation on the current status and usage of the DIRAC software and services.
    - Site web: http://diracgrid.org/

Multi-VO services support

- Small communities can not afford installation and management of a fully functional DIRAC service
    - No expertise
    - Too complicated
- France-Grilles was the first grid infrastructure project to offer DIRAC services to its users in 2012
- Several multi-VO DIRAC services are now available
    - GridPP, EGI, ILC, DutchGrid, IHEP, etc.

- Behind the scenes
    - A lot of work also
        - Enhanced security
        - Managing VO specific configurations: users, resources, services

      Come up with your resources and we will plugin it into DIRAC services !

EGI Workload Manager

- One of the services in the EOCS Marketplace Catalogue

  https://marketplace.eosc-portal.eu/services/egi-workload-manager

EUROPEAN OPEN SCIENCE CLOUD

- Development team in CPPM
- Managing user jobs running on the EGI computing resources
- Based on the DIRAC Interware distributed computing framework

EGI-ACE Call for Use Cases

https://www.egi.eu/projects/egi-ace/call-for-use-cases/

- Who should apply
  - International researchers, research projects, communities and infrastructures, as well as national research groups needing services and support for:
    - Large-scale data processing, scientific analysis, visualization
    - Hosting data analysis platforms and applications in the cloud
    - Federate and make accessible community-specific compute services in EOSC
- Timeline
  - The call is kept open during 2021 and 2022.
  - Cut-off dates with 2-monthly frequency, followed by the evaluation (within 1 month) of the submitted applications.

DIRAC in France

DIRAC@IN2P3 Master Project

The Project started in 2017

- Scientific Coordinator: A.Tsaregorodtsev, CPPM
- Technical Coordinator: J.Bregeon, LPSC

Participants

- IN2P3: CPPM, LUPM, LPSC, IPHC, CC/IN2P3
- CNRS/INSERM/INSA/U.Lyon/U.St.Etienne: CREATIS

Goal: research, development and promotion of the DIRAC software and services

Developments

- High level workflow management
- Heterogeneous resources management (cloud, HPCs)
- OAuth/OIDC based security infrastructure for DIRAC

Services

- Operator of the EGI Workload Manager service

Conclusions

DIRAC is an example of a product that evolved from a single experiment development to an open-source project exploited by multiple scientifique communities

DIRAC introduced an innovative workload management architecture with pilot jobs which is now adopted by all the large HEP experiments and also beyond the HEP domain

DIRAC offers a complete solution for all the computing and data management tasks for research communities

DIRAC is conceived for extensions to meet specific needs of various scientific applications

DIRAC services are available in multiple large grid infrastructure projects. Just come and use it !

Invitation

If you want to know more about DIRAC Interware join us at the next next DIRAC Tutorial at CPPM library tomorrow at 9am.

Thanks for your attention !

Backup

DIRAC – what makes it unique - Innovative pilot based architecture

- Push scheduling:
  - The scheduler is using the information about availability and status of the computing element to find the best match to a particular job requirement.
  - After the job will be sent to the computing element for execution.
  - The scheduler is an active component and the computing element is passive.

- Pull scheduling:
  - The computing resource is seeking task to be executed.
  - The jobs are accumulated by a production service, validated and put in a waiting queue.
  - Once the computing resource is available, it send request for a job to be done to the production service.
  - The production service chooses a job according to the resources capabilities and the serves it in response to the request.