Bringing distributed computing power to scientific applications

A.Tsaregorodtsev,
CPPM-IN2P3-CNRS, Marseille,
EPD seminar, IHEP, Beijing,
18 July 2023

DIRAC
THE INTERWARE

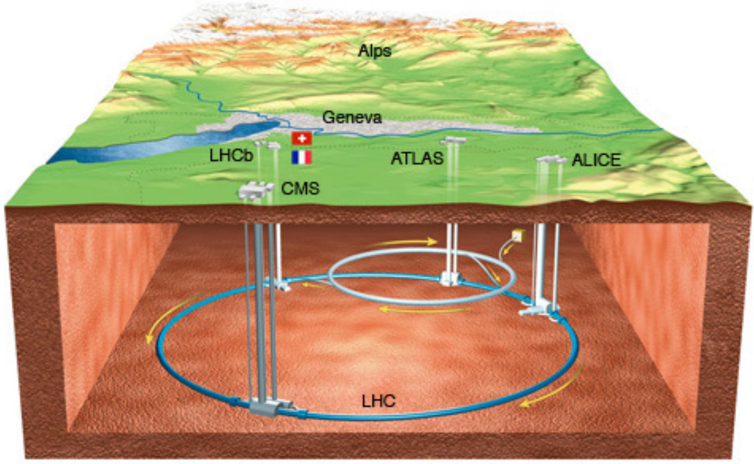
Outline

- Brief history of computing grids for HEP
- DIRAC as the LHCb's solution for distributed 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
  - 14 February 1997 ALICE
  - 17 September 1998 LHCb

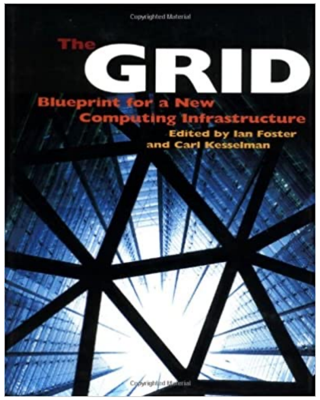


Brief history of computing grids for HEP

- **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



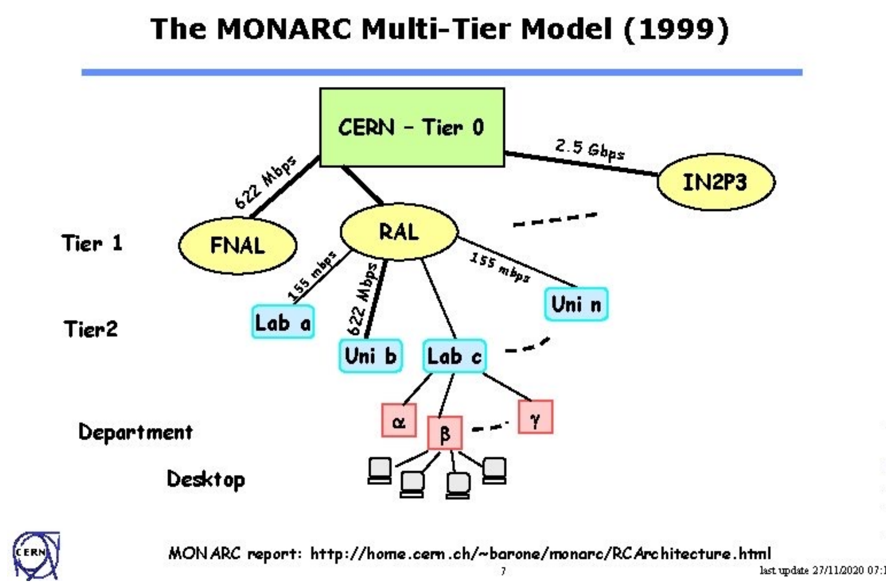
https://www.amazon.com/Grid-Blueprint-Computing-Infrastructure-Elsevier/dp/1558604758

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

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



TOPOLOGY

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

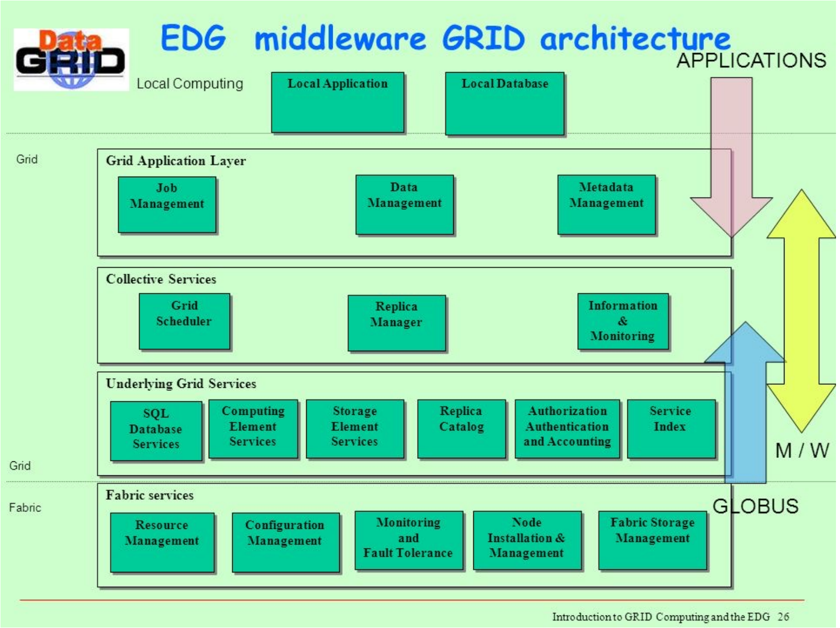
- **DataGrid (2001-2003)**

- 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 the Globus 2 toolkit.

  - DataGrid didn't satisfy production level requirements 🙁



https://slideplayer.com/slide/8630667/
https://www.slideserve.com/dotty/grid-computing-at-cern-powerpoint-ppt-presentation

DIRAC
THE INTERWARE

Brief history of computing grids for HEP

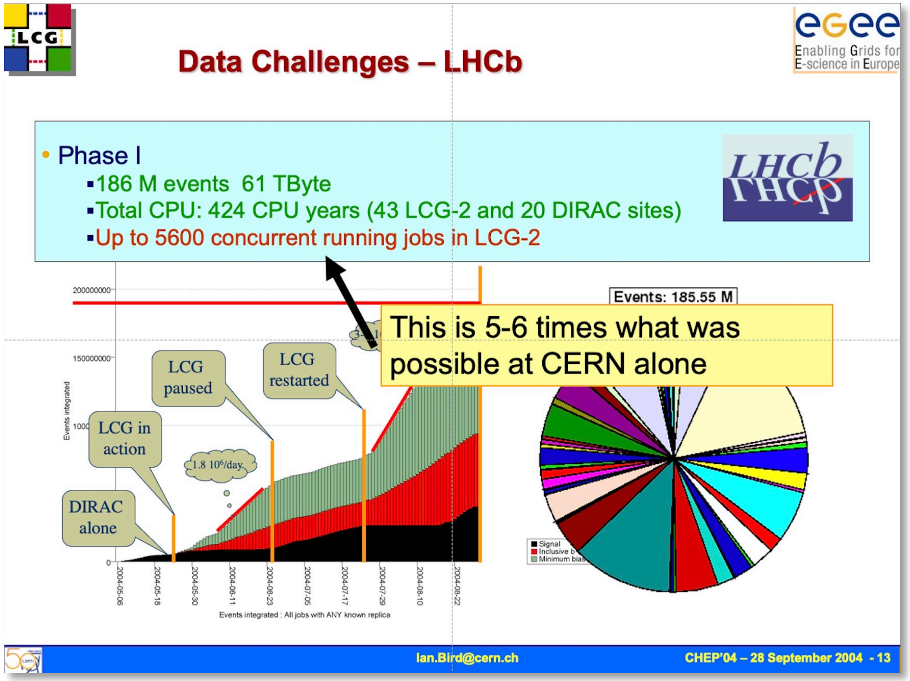
- **Data challenge (2004)**
  - In the framework of the LCG (WLCG) project
  - Test and validate the computing models
  - Produce simulated data
  - Test experiments production frameworks and software
  - The four experiments participated

- **LHCb** participated with the new system called **DIRAC**
- **DIRAC** results during the Data Challenge in 2004 showed that the Grid can face the LCG project challenge.

- **First success!**

Brief history of computing grids for HEP

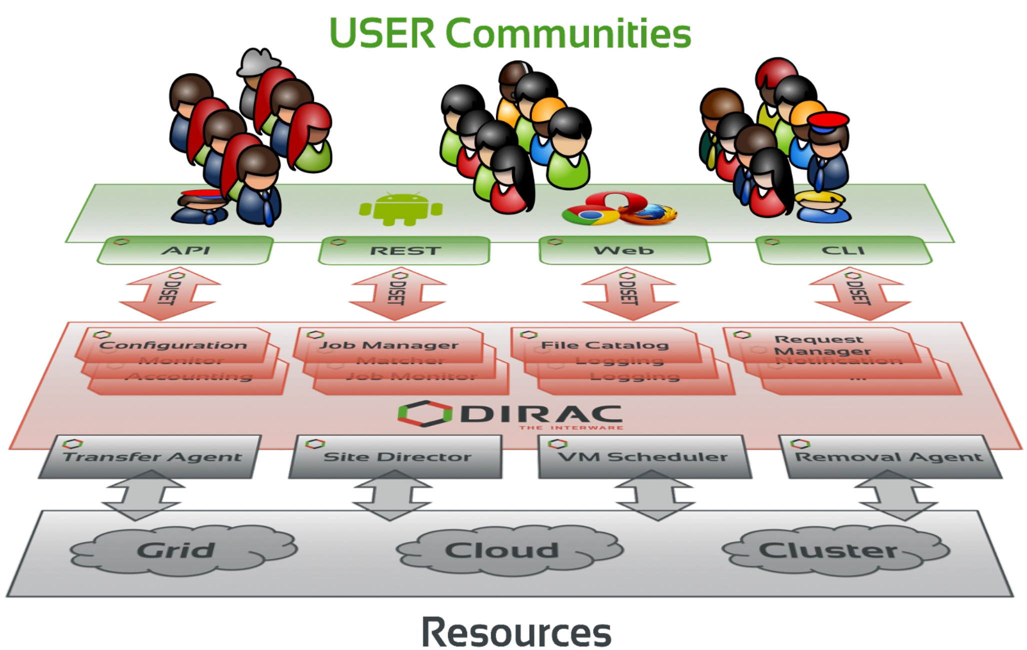
- **Why it was a success?**
- DIRAC job user efficiency > 90%
  - while ~60% success rate of LCG jobs.
- The first production system to employ in the grid job an script to pull jobs from a queue
  - Sending agent as regular jobs
  - Now known as pilot jobs



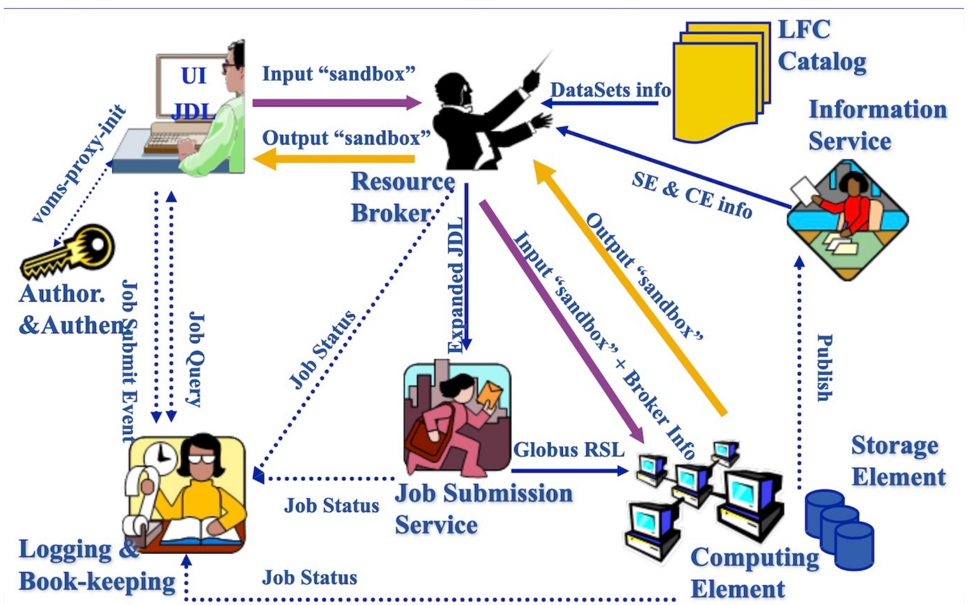
- Record of maximum running jobs
  - And orders of magnitude less than what we can do now !
- The scalability of the system allowed to saturate all available resource of DC'2004

What's the DIRAC Interware?

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



Job Workflow in gLite with centralized WMS architecture

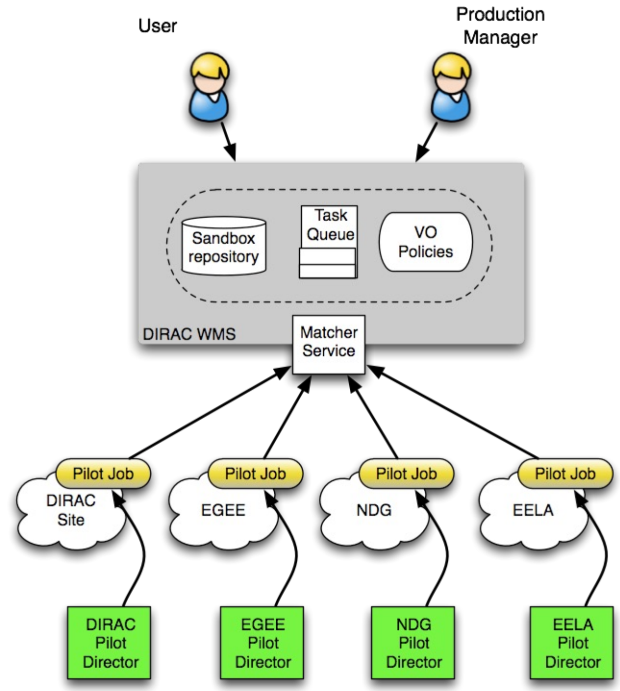


- Operational information is collected for the central Resource Broker (RB)
  - Site capacity and status
  - Data placement
- For each submitted job the RB makes a decision of dispatching it to the most appropriate site
  - Meeting job requirements
  - Least loaded
- An example of *PUSH* scheduling paradigm

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 THE INTERWARE

Grid scalability problem

- **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 THE INTERWARE

Pull vs *Push*

- 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 THE INTERWARE

Pull vs *Push*

- **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!

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

- Glite WMS (last version of the RB ***PUSH*** scheduling) was decommissioned some years ago.

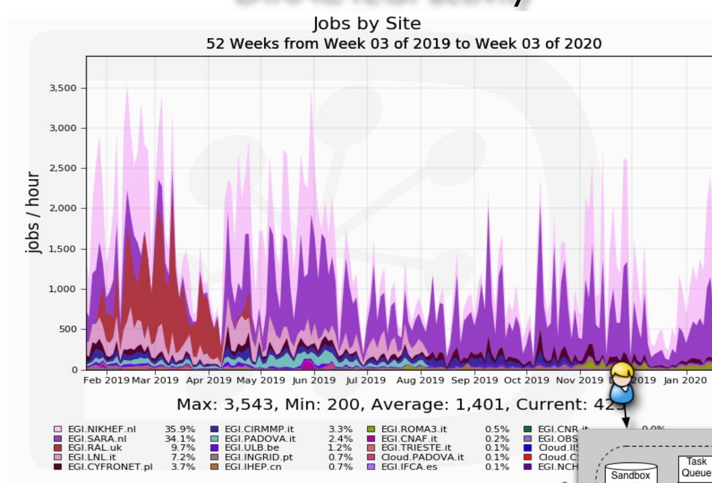
DIRAC – complete solution

- DIRAC combines various distributed computing and storage resources in a single coherent system
  - 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

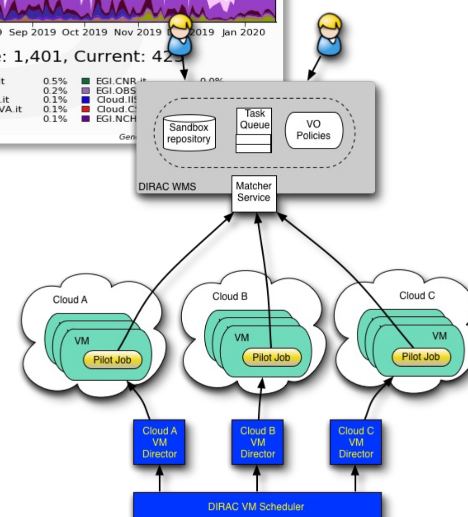
Computing Grids and Clouds

- DIRAC was initially developed with the focus on accessing conventional Grid computing resources
  - WLCG grid resources for the LHCb Collaboration

DIRAC4EGI activity

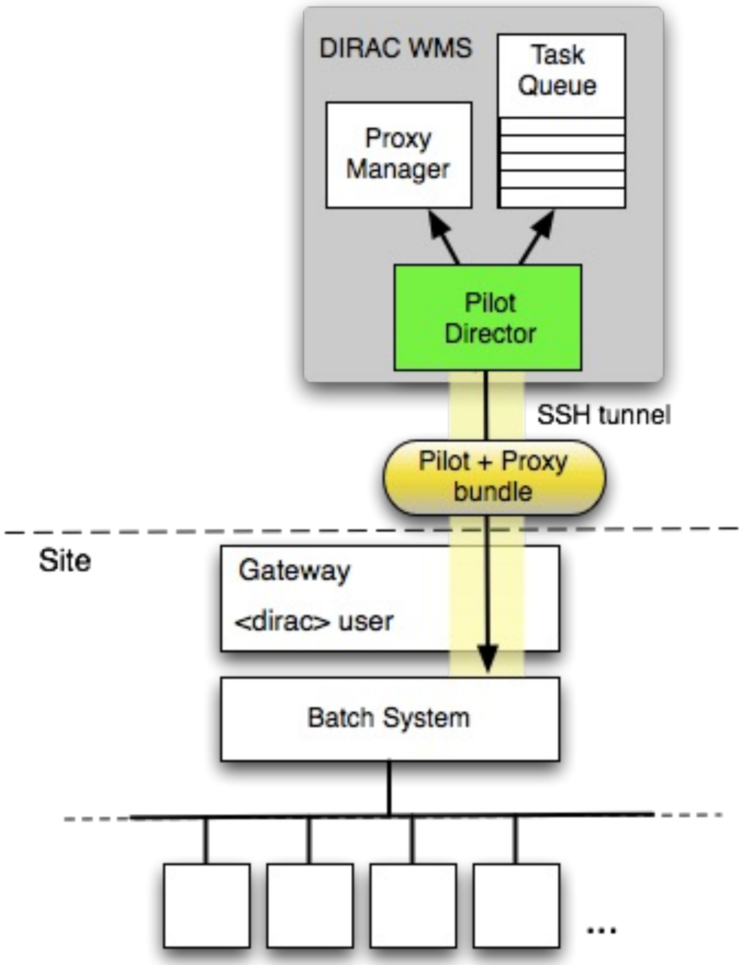


- Grid infrastructures
  - E.g. EGI, WLCG, OSG
  - HTCondorCE, ARC
- Cloud infrastructures
  - EGI Federated Cloud, France-Grilles cloud
- Others
  - Vacuum, Volunteer grids



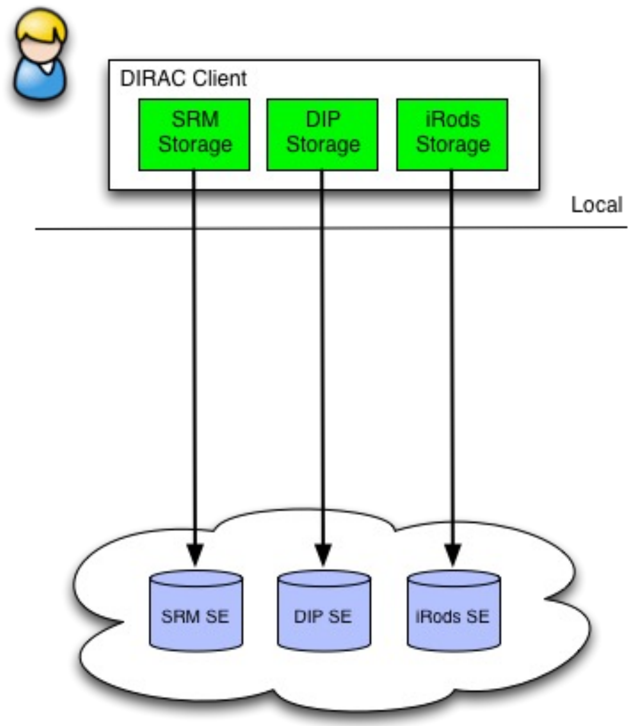
Standalone computing clusters

- Users can connect their own computing resources
  - Not making part of any grid infrastructure
- The user site can be:
  - a single computer or several computers without any batch system
  - a computing cluster with a batch system
    - LSF, BQS, SGE, PBS/Torque, Condor
      - Commodity computer farms
    - SLURM
      - HPC centers



Storage plugins

- Storage element abstraction with a client implementation for each access protocol
  - DIPS – DIRAC data transfer protocol
  - FTP, HTTP, WebDAV
  - SRM, XROOTD, RFIO, DCAP, etc
    - HEP centers specific protocols
    - Using gfal2 library developed at CERN
  - S3, Swift, CDMI: cloud specific data access protocols
- Like with CE's, each SE is seen by the clients as a logical entity
  - With some specific operational properties
    - Archive, limited access, etc
  - SE's can be configured with multiple protocols
- Including new data access technologies requires creating new specific plug-in



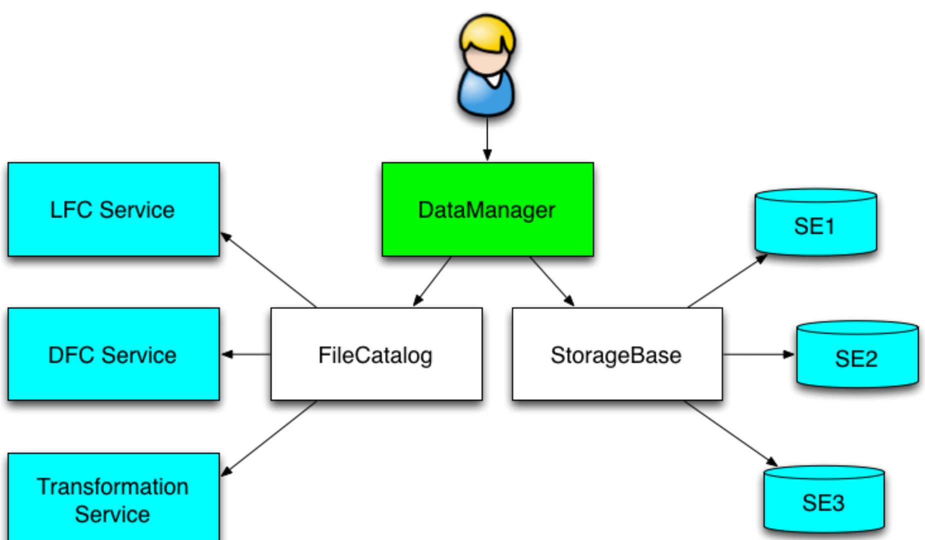
File Catalog Service

- File Catalog is a service to keep track of all the physical file replicas in all the SE's
  - Stores also file properties:
    - Size, creation/modification time stamps, ownership, checksums
    - User ACLs
- DIRAC relies on a *central* File Catalog
  - Defines a single logical name space for all the managed data
  - Organizes files hierarchically like in common file systems

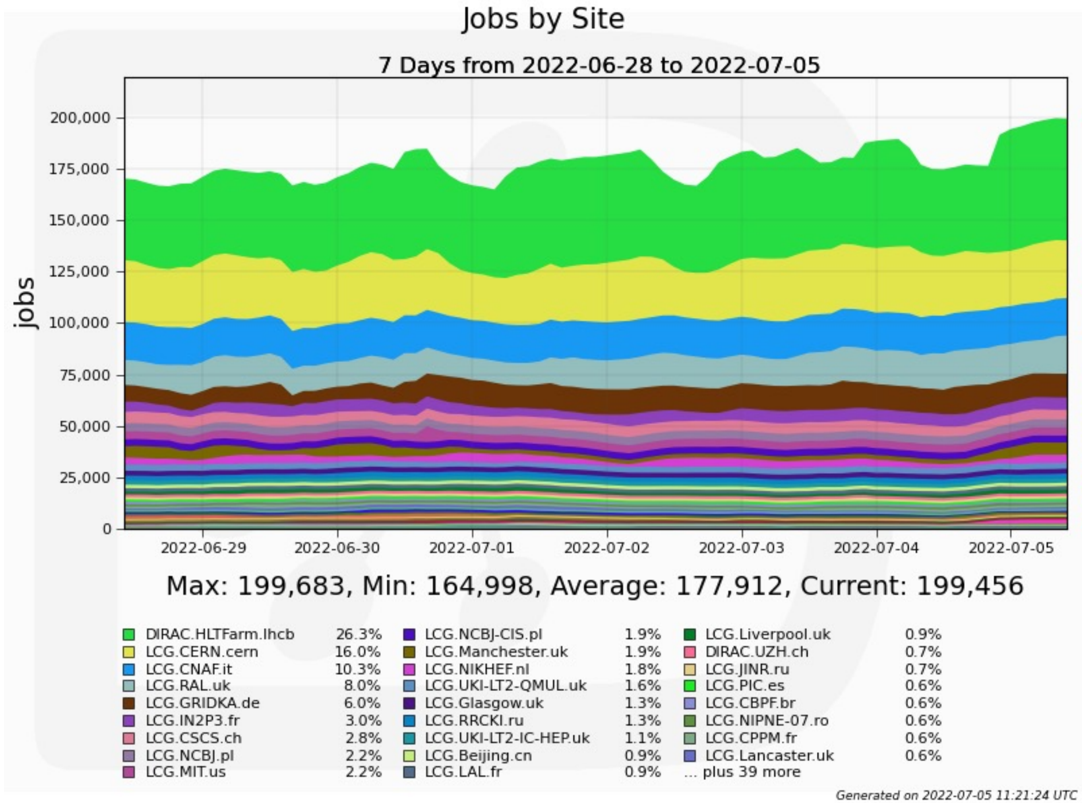
Combined data API

- Together with the data access components DFC allows to present data to users as a single global file system

- DataManager API is a single client interface for logical data operations



LHCb Collaboration



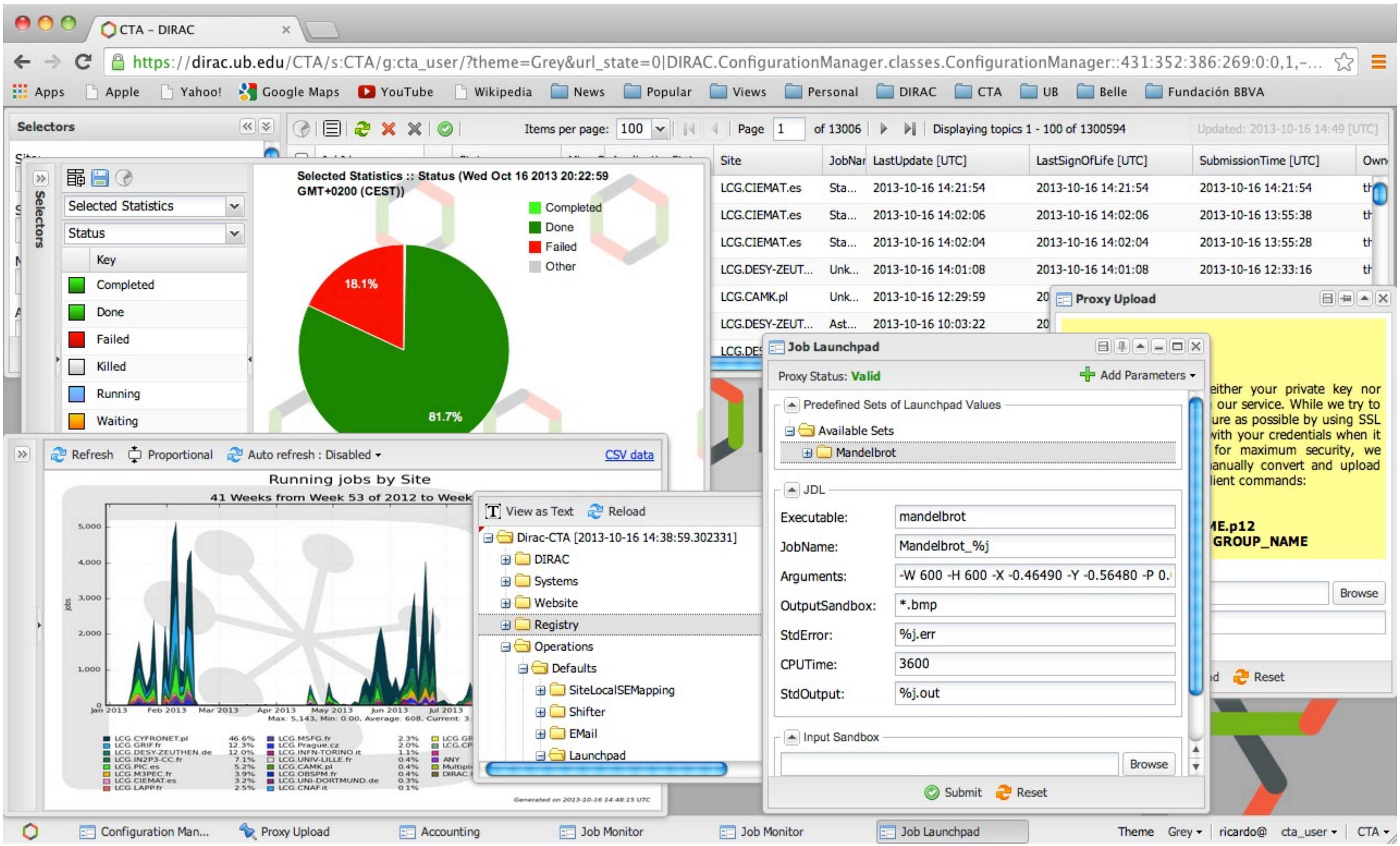
- Up to 200K concurrent jobs in ~80 distinct sites
  - Limited by available resources, not by the system capacity
- Further optimizations to increase the capacity are possible
  - Hardware, database optimizations, service load balancing, etc

User interfaces

- Command line tools
  - Batch system like commands for job submission:
    - **dsub, dstat, doutput**
  - Shell like commands for data management
    - **dls, dcd, dpwd, dchmod**
    - **dput, dget, drepl**

- From the user's perspective DIRAC is presenting multiple heterogeneous distributed computing and storage resources as single large computer

Web Portal applications



Converting into an open-source project

- 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.

Converting into an open-source project

- Behind the scenes:
  - Complete re-engineering into a generic framework capable to serve the distributed computing needs of different Virtual Organizations.

  - Separate generic and VO specific parts
    - 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

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.

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
- The Consortium holds the copyright for the DIRAC software
  - GPL v3
- Organizes workshops, tutorials and other events to promote DIRAC.
  - Next on is in October at KEK, Japan

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, JINR, etc

- DIRAC at IHEP started as a BES III service
  - Afterwards evolved as a multi-VO service supporting also Juno, CEPC, etc

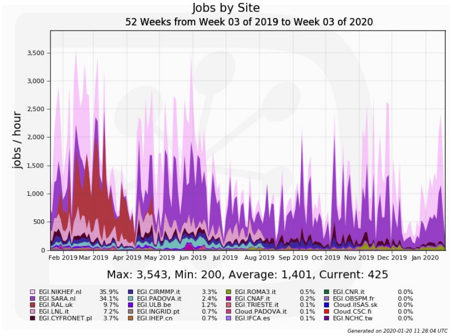
Multi-VO services support

- Behind the scenes
  - Software adaptation
    - Enhanced security
    - Managing VO specific configurations: users, resources, services
  - Possibility to develop, deploy and operate community specific services
    - E.g. specific catalogs, pilot factories
  - Possibility to connect specific computing and storage resources
    - 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

  - Software development by the DIRAC Project
  - Services are hosted in the IN2P3 Computing Center, Lyon, France
  - ~20 user communities, ~700 registered users
    - biomed, astrophysics, complex systems
  - ~12M job processed per year



DIRAC THE INTERWARE

Conclusion

- 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.