DIRAC & Use of Agents

Nick Brook

University of BRISTOL

- Introduction
- Design/architecture issues
- Use in LHCb

DIRAC development team

DIRAC- Distributed Infrastructure with Remote Agent Control

- DIRAC development team

  TSAREGORODTSEV Andrei, GARONNE Vincent, STOKES-REES Ian, GRACIÁNI Ricardo, PATERSON Stuart, CASAJUS Adria, CLOSIER Joel, SABORIDO SILVA Juan, KUZNETSOV Gennady, CHARPENTIER Philippe, BROOK Nick, SMITH Andrew, SANCHEZ Manuel, SANTINELLI Roberto, CASTELLANI Gianluca, BARGIOTTI, Marianne

DIRAC Overview

- LHCb system for the Monte Carlo simulation data production and analysis
- Integrates computing resources available at LHCb production sites as well as on the LCG grid
- Composed of a set of light-weight services and a network of distributed agents to deliver workload to computing resources
- Runs autonomously once installed and configured on production sites
- Implemented in Python, using XML-RPC service access protocol

DIRAC Design Goals

- Light implementation
  - Easy to deploy on various platforms
  - Non-intrusive
    - No root privileges
  - Easy to configure, maintain and operate
- Using standard components and third party developments as much as possible
- High level of adaptability
  - There will be always resources outside LCG domain
    - Sites that can not "afford" LCG, desktops, …
  - Plan to use them all in a consistent way
- Modular design at each level
  - Adding easily new functionality

DIRAC WMS

- DIRAC Workload Management System is itself composed of a set of central services, pilot agents and job wrappers
- Realizes the PULL scheduling paradigm
  - Pilot agents deployed on a site/LCG Worker Nodes pull the jobs from the central Task Queue
- The central Task Queue allows to apply easily the VO policies by prioritization of the user jobs
  - Using the accounting information and user identities, groups and roles
- The job scheduling happens at the last moment
  - Job goes to a resource for immediate execution

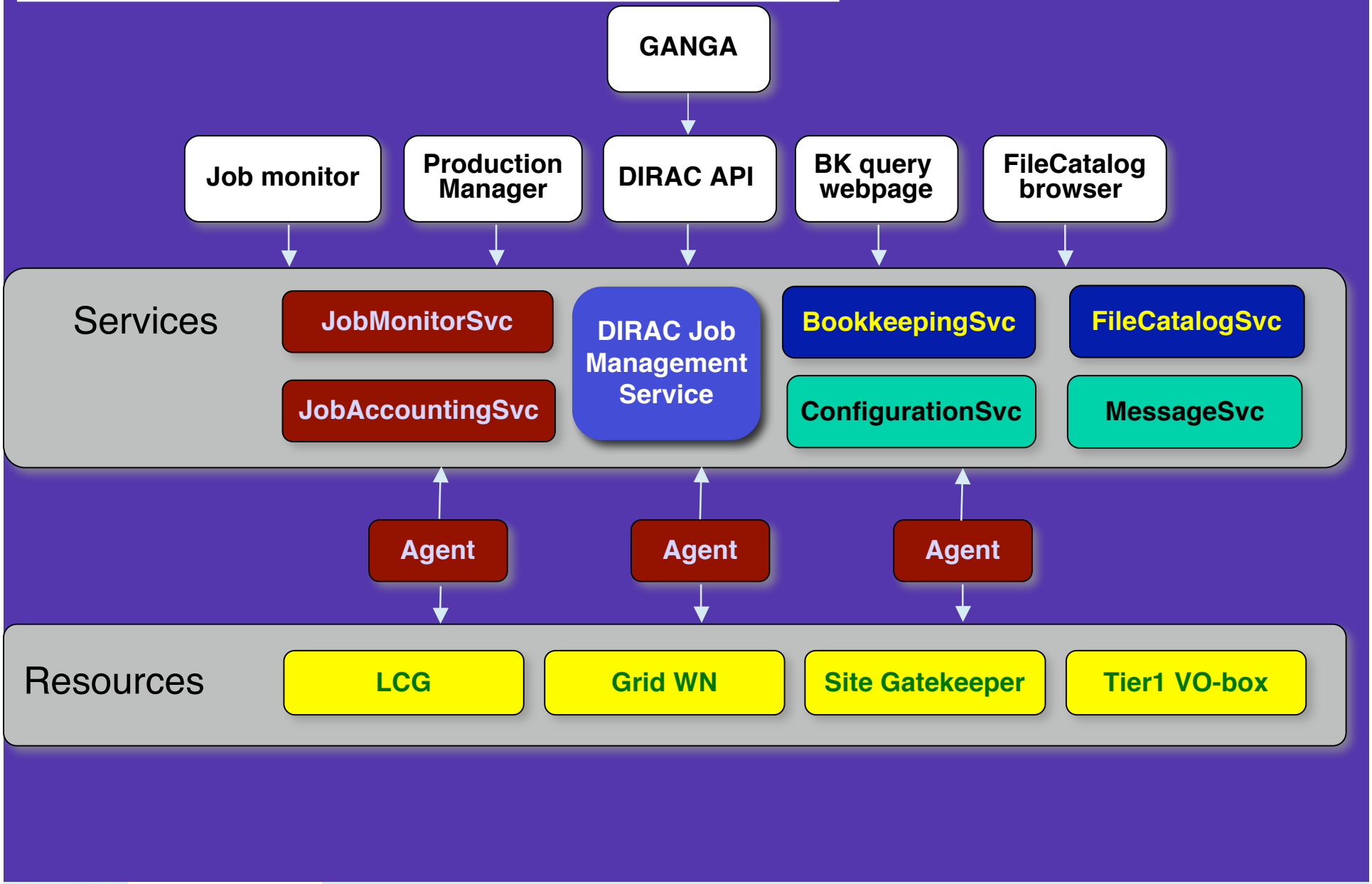
overlay network paradigm

Dynamically deployed agents

How to involve the resources where the DIRAC agents are not yet installed or can not be installed

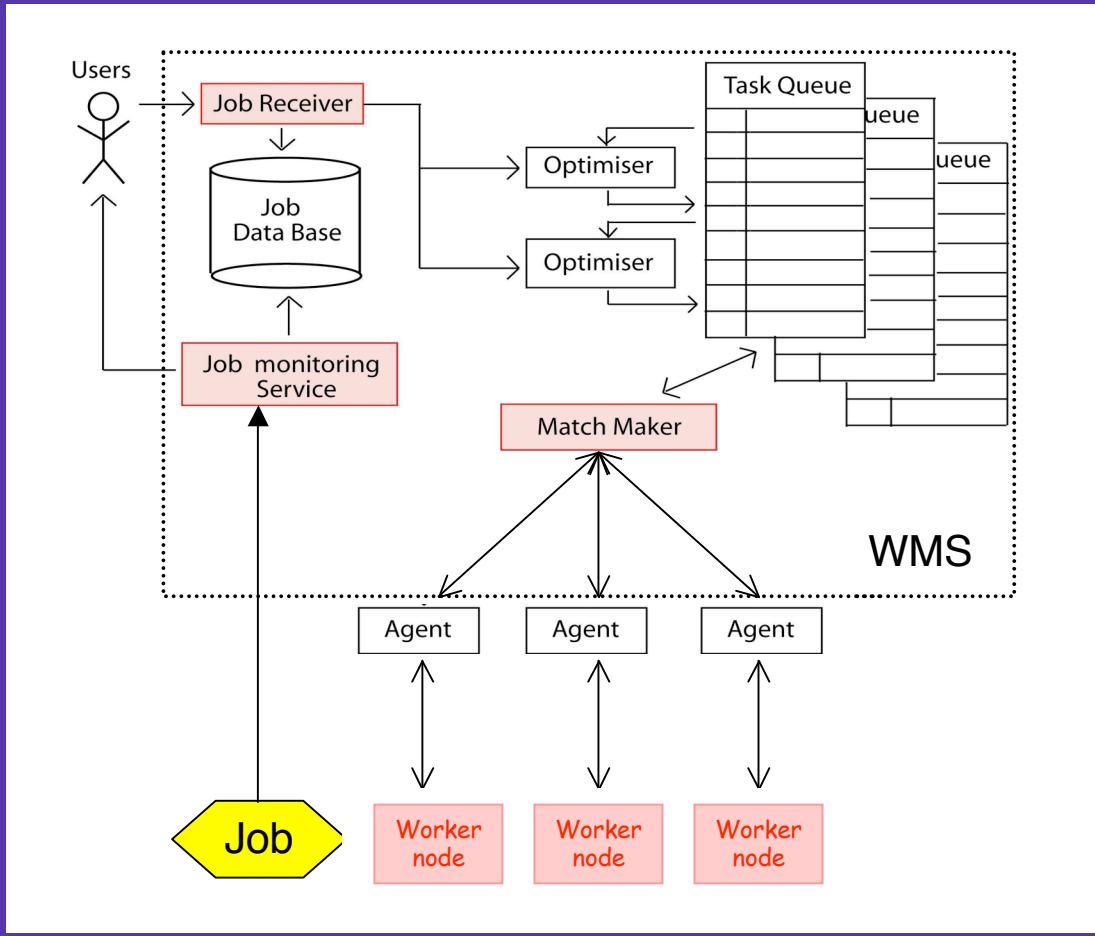
- Workload management with "resource reservation"
  - Sending agent as a regular job
  - Turning a WN into a virtual LHCb production site
- Strategy applied on LCG
  - Effectively using LCG services to deploy DIRAC infrastructure on the LCG resources
- Built-in extra protection
  - Check environment on WN
  - Any problems only agent is lost not the job

DIRAC Services & Resources



DIRAC Workload Management

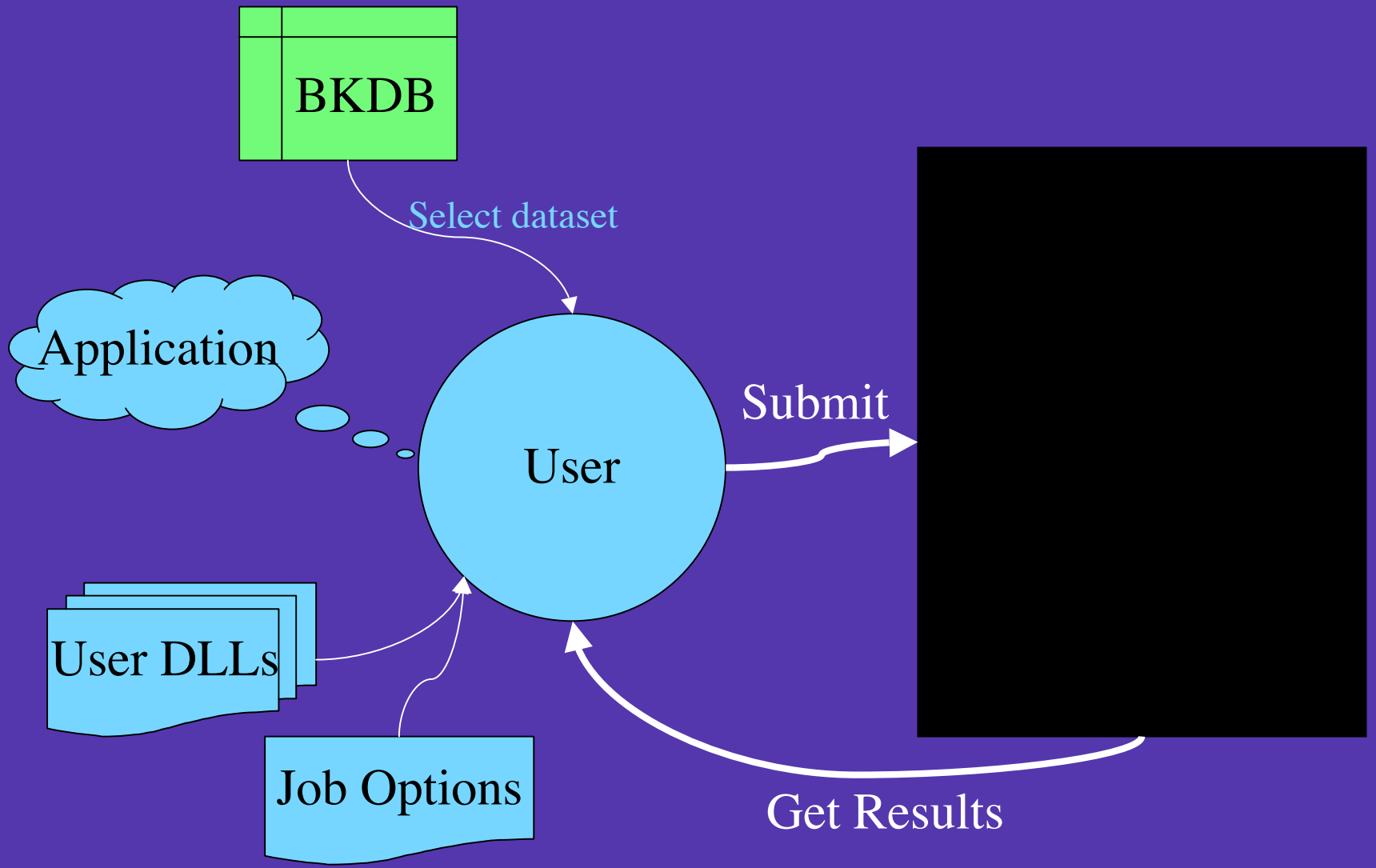
- Realizes *PULL* scheduling paradigm
- Agents are requesting jobs whenever the corresponding resource is free
- Using Condor ClassAd and Matchmaker for finding jobs suitable to the resource profile
- Agents are steering job execution on site
- Jobs are reporting their state and environment to central Job Monitoring service



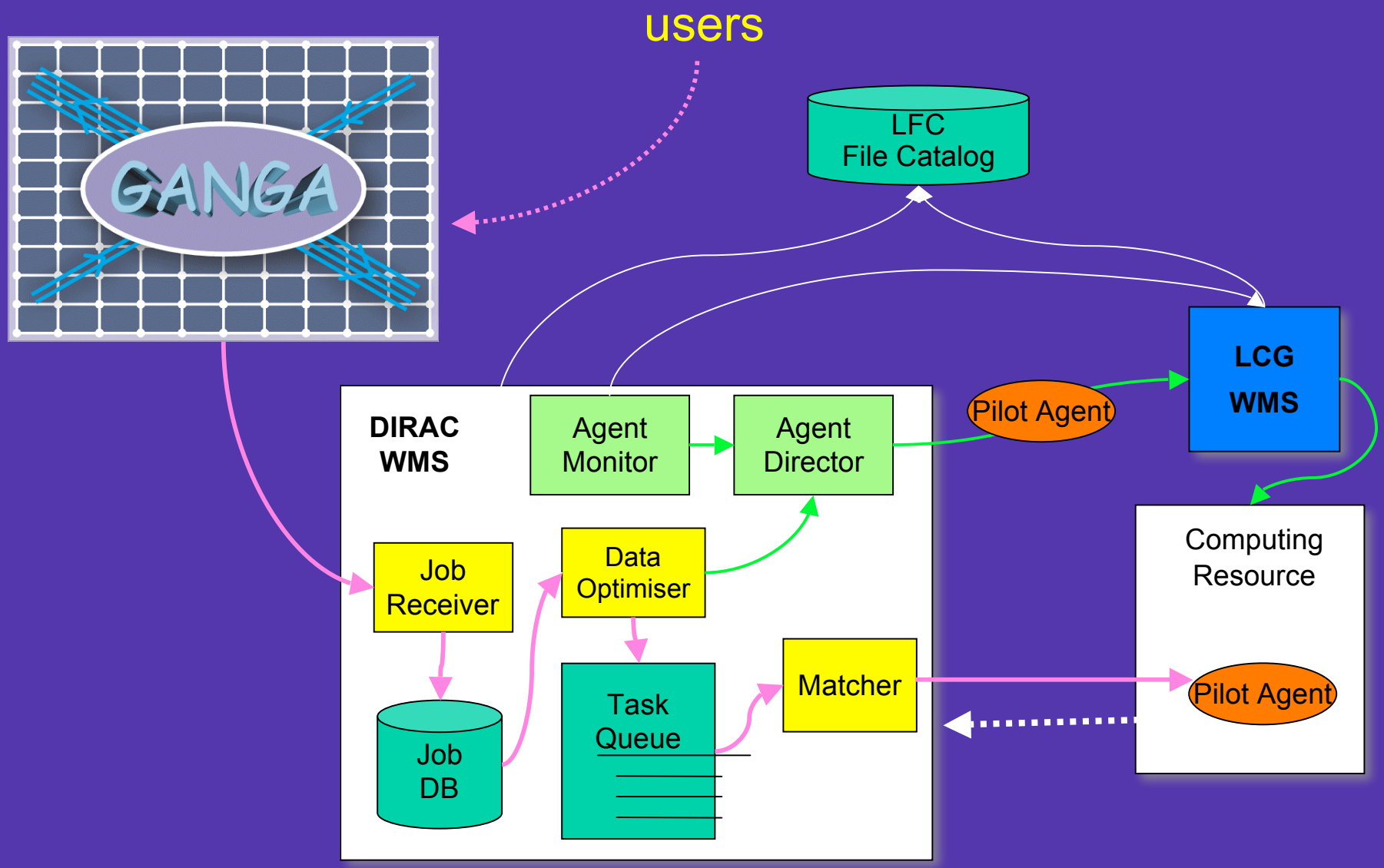
Technology Choice
XML-RPC

- Standard, simple, available out of the box in the standard python library
  - Both server and client
  - Using expat XML parser
- Server
  - Very simple socket based
  - Multithreaded
  - Supports up to 40 Hz requests rates
- Did not need anything more complex
  - SOAP, WSDL, …
  - Avoid performance overhead, compatability problems, …
- DIRAC Secure Transport
  - Extension of HTTPS supporting x509 certificates and grid proxies plus enhancement of native XML-RPC capabilities.

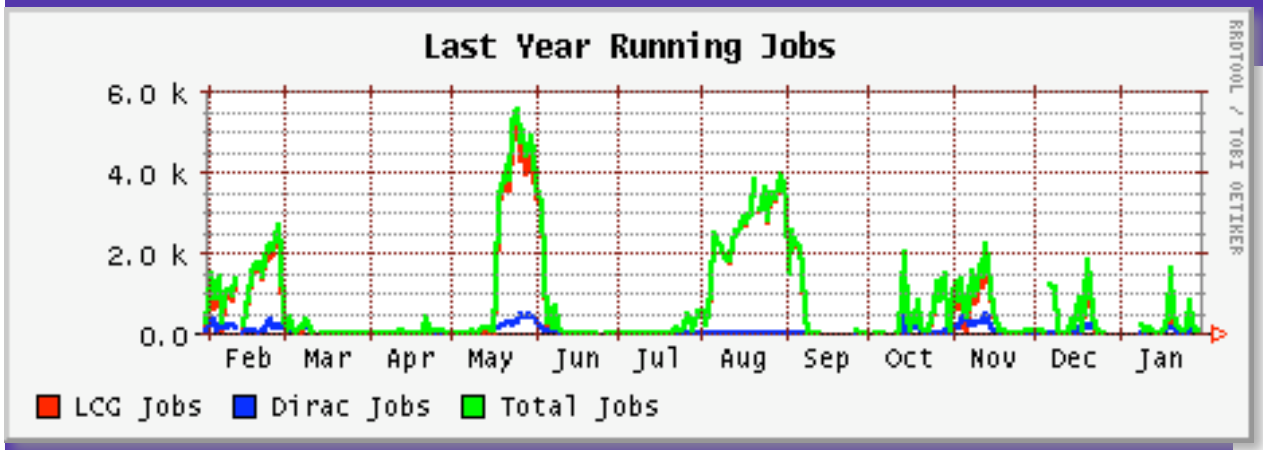
Distributed analysis:user viewpoint



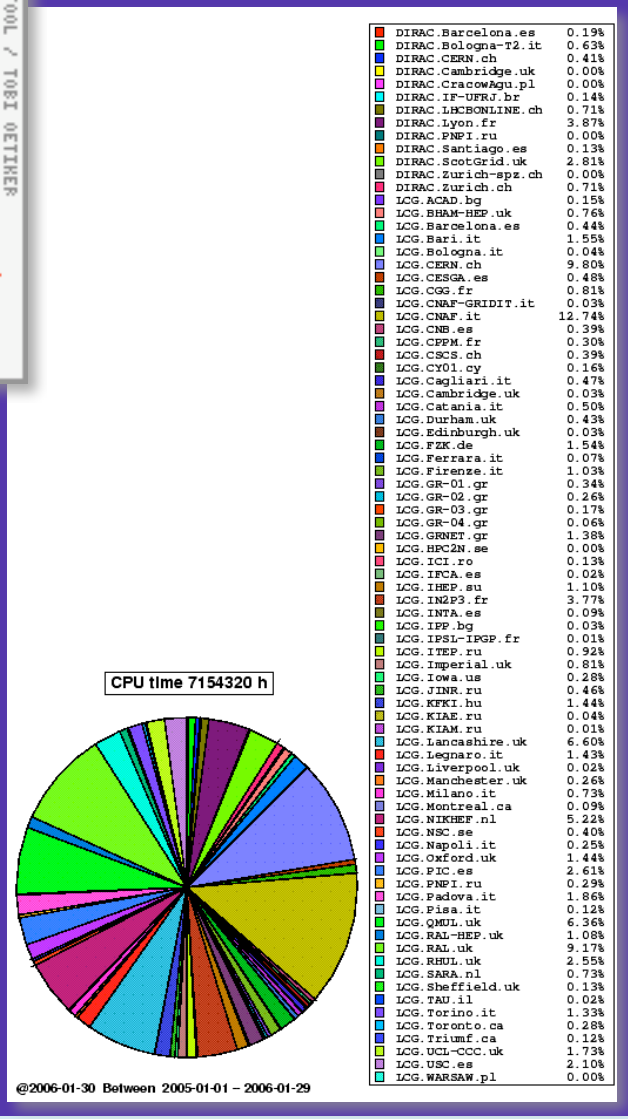
Open the box …


users
GANGA
LFC
File Catalog
LCG
WMS
DIRAC
WMS
Agent
Monitor
Agent
Director
Pilot Agent
Job
Receiver
Data
Optimiser
Computing
Resource
Job
DB
Task
Queue
Matcher
Pilot Agent

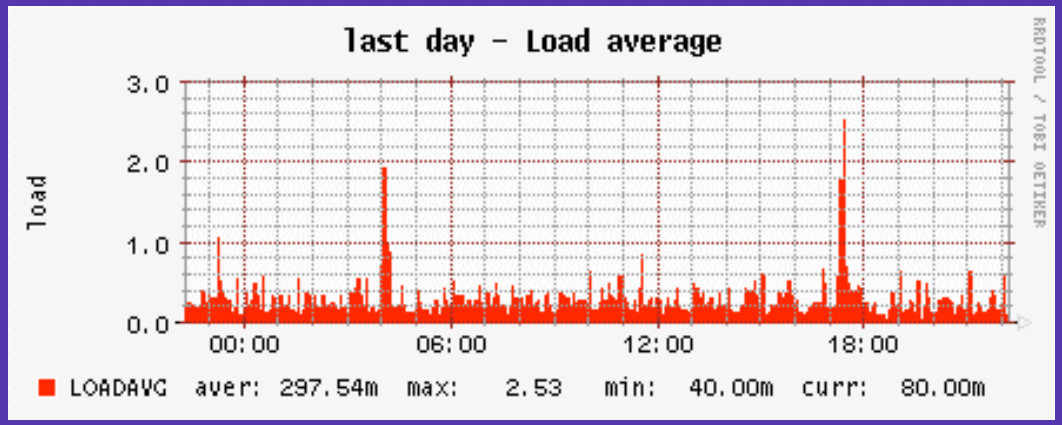
DIRAC production performance

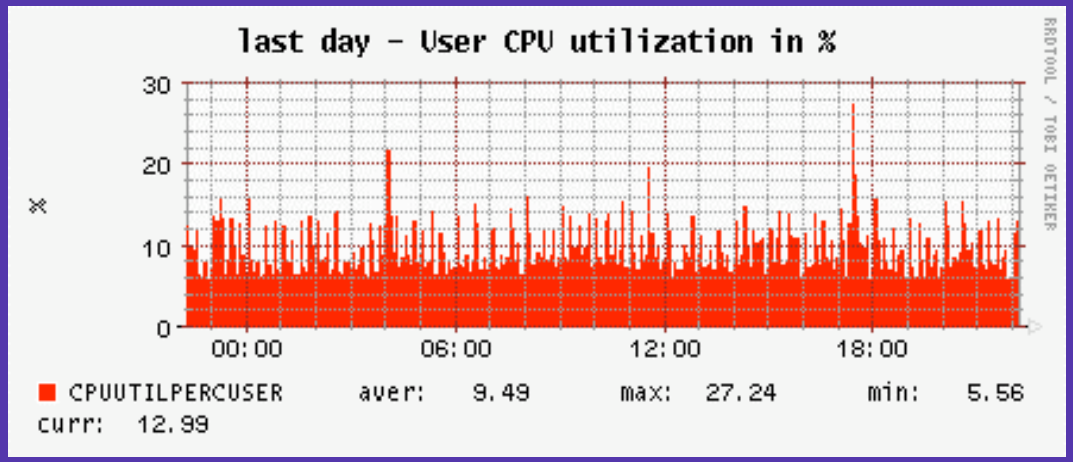


- Up to over 5000 simultaneous production jobs
  - The throughput is only limited by the capacity available on LCG
- ~80 distinct sites accessed through LCG or through DIRAC directly



DIRAC performance

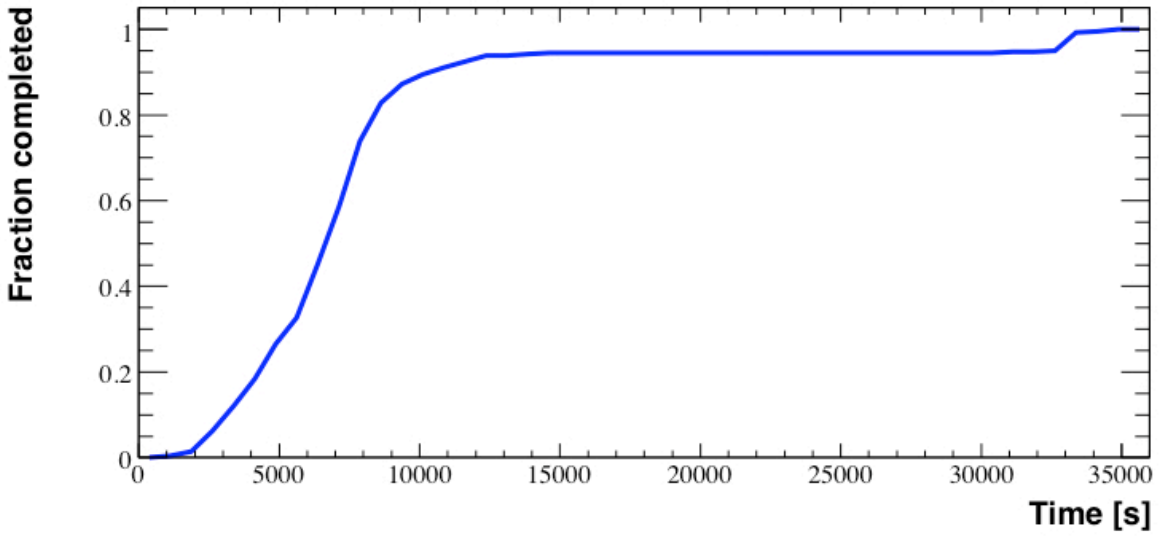




Far from a critical load on the DIRAC servers

Performance: Throughput

Look at the time it takes before the results are back



Analysis of 5M simulated events

(split by GANGA into 500 jobs)

90% of results are back within less than 3 hours

95% in 4 hours

100% in 10 hours

Last 5% caused by Tier 1 site with data access problems

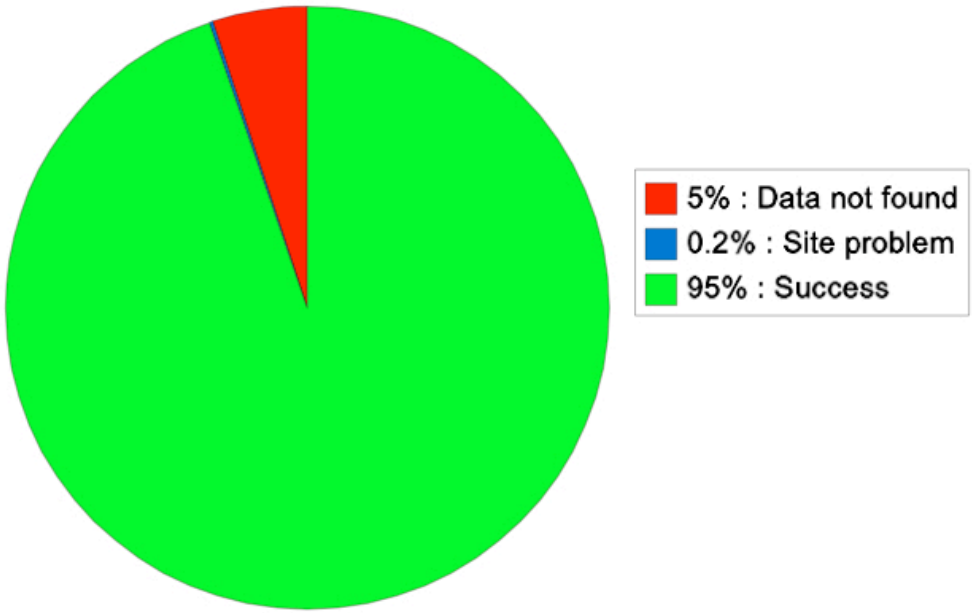
Performance: Reliability

Evaluate the fraction of jobs that succeed.

With success is meant that submission, running and retrieval of correct output all work as expected.

95% of jobs succeed in the first round.

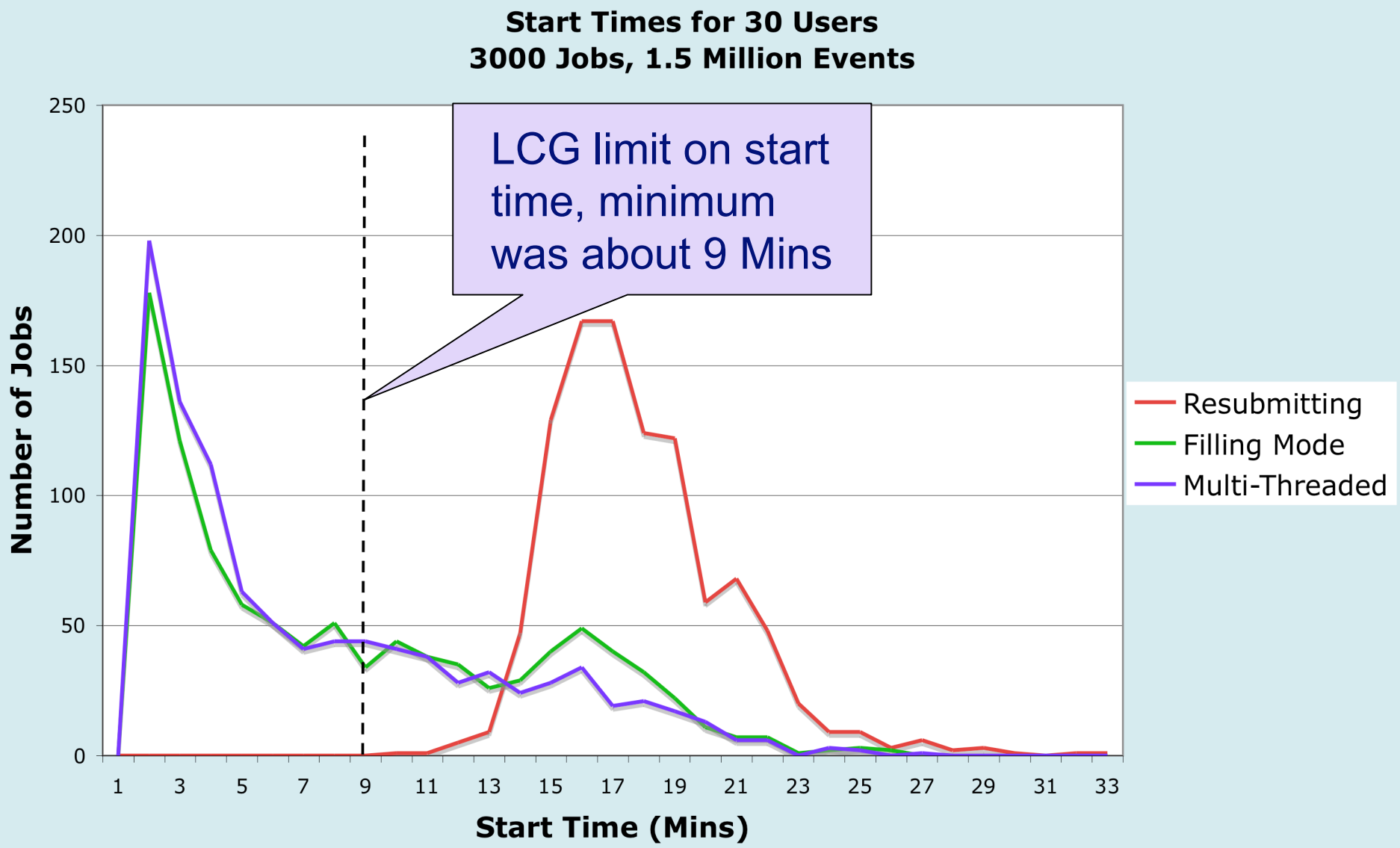
Remainder dominated by occasional inconsistencies in file catalogue.



DIRAC WMS Pilot Agent Strategies

- Several ways to use the DIRAC infrastructure
  - aim to minimise the start time of user analysis jobs
- The following explores some of the possible DIRAC modes of submission
  - 'Resubmission'
    - Pilot Agent submission to LCG with monitoring
    - Multiple Pilot Agents may be sent in case of LCG failures
  - 'Filling Mode'
    - Pilot Agents may request several jobs from the same user, one after the other
  - 'Multi-Threaded'
    - Same as 'Filling' Mode above except two jobs can be run in parallel on the Worker Node

Start Times for 10 Experiments, 30 Users



Summary

- The DIRAC Overlay network paradigm
  - efficient in integrating heterogeneous resources in a single reliable system for simulation data production
- The system is now extended to deal with the Distributed Analysis tasks
  - Workload management on the level of the user is effective
  - Real users (~30) are starting to use the system
- The LHCb Data Challenge 2006 in June
  - Test LHCb computing model before data taking
  - … but also the ultimate test of the DIRAC system