DIRAC-Belle & Amazon EC²

R. G., A. Casajus & T. Fifield

- **Motivation**
- **Strategy**
- **New DIRAC Developments**
- **Results**
- **Summary**

Motivation

- Belle (and Belle II in the future) has large peaks of CPU need for Monte Carlo simulation.
- It has been shown that Amazon CE^2 can be used to provide extra CPU resources (*).
- Need to make a large scale exercise to determine efficiency and cost.
- Use a well established tool easily integrate with exiting resources and to reduce development time.

(*)”Belle Monte-Carlo Production on the Amazon EC2 Cloud” CHEP 2009.

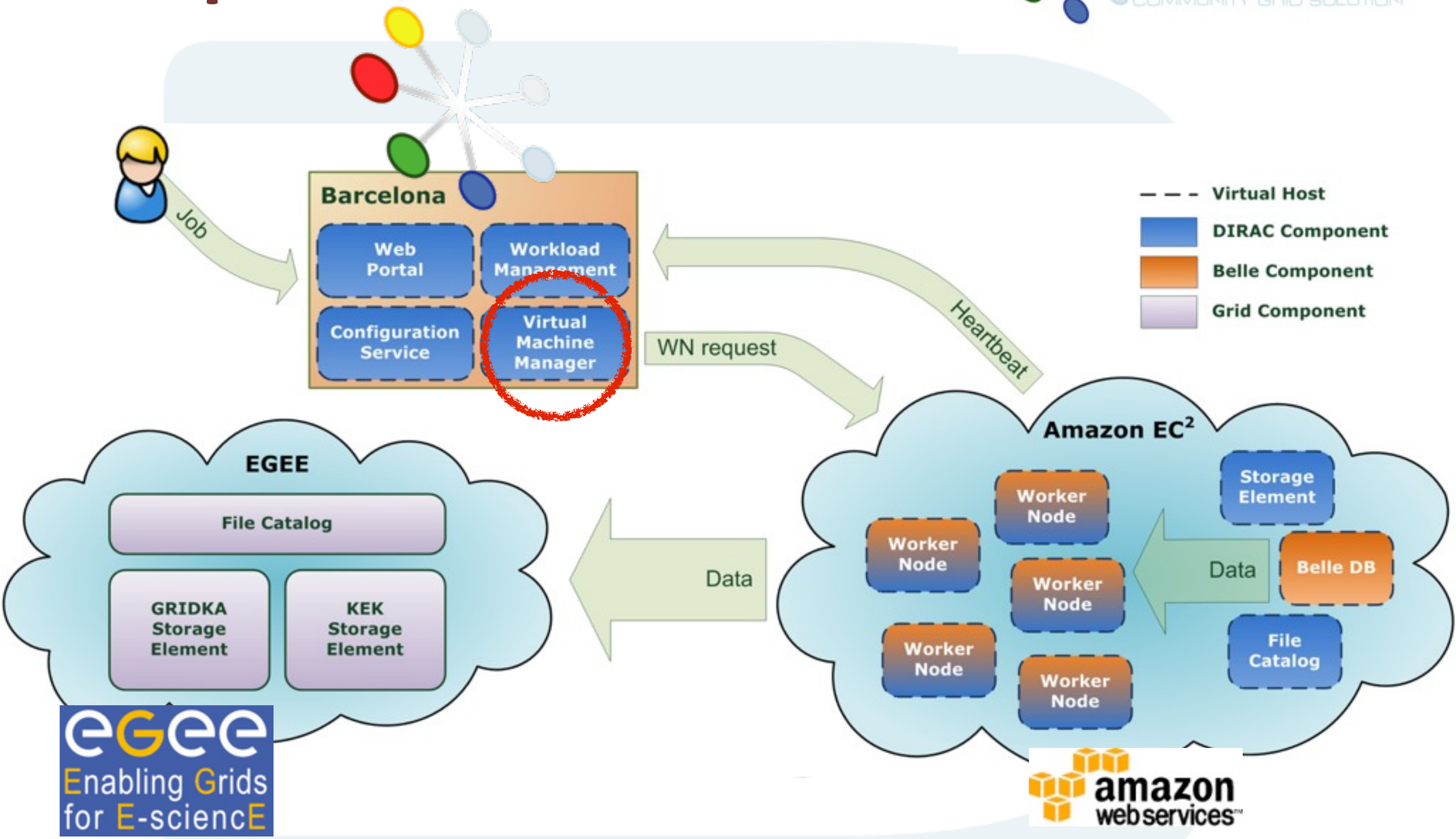
Strategy

- Use DIRAC framework to control execution and integrate cloud resources with grid and local clusters:
  - Grid: DIRAC Pilot Jobs
  - Cloud: DIRAC Virtual Machines
- Divide exercise into different phases:
  - Phase I:
    - Test cloud
  - Phase II:
    - Incorporate local resources
  - Phase III:
    - Incorporate grid resources
- Make extensive use of virtualization for future portability and scalability.

DIRAC Design choices

- **VO Centric**:
  - Gives to the community, the VO, a central role in the relation of its users with their computing resources.
- **Modularity**:
  - To achieve optimal scalability and flexibility, a highly modular design was decided.
- **Pull Scheduling**:
  - Implements pull scheduling with late binding of payload to resource to extract optimal performance out of the ever changing underlying resources.

Proposed DIRAC solution



Central Components (@Barcelona)

- Configuration Server:
  - Describes resources and the way to access them: Sites, CE's, SE's, FC's,...
  - Users and Policies.
  - Location of DIRAC components.
  - Configurable parameters.
- Web Portal:
  - Monitoring
  - Accounting
  - Interactivity
- Workload Management:
  - Job repository and agents to access the resources
- **Virtual Machine:**
  - Scheduler for submission
  - Manager for control and monitoring

Remote Component (@ Amazon)

- Storage Virtual Machine:
  - DIRAC Storage Element/File Catalog
    - cache input data
  - PostgreSQL
    - Belle condition data
- **Execution Virtual Machine:**
  - Job Agent
  - **Virtual Machine Monitor**

Other Components

- User Interface
  - job preparation
  - job submission
- Grid Resources
  - Interfaced by DIRAC
  - Storage Elements
    - KEK, GRIDKA, CYFRONET,...
  - LFC
    - KEK

Pilots -> Virtual Machines (I)

DIRAC Execution Virtual Machine:

- Replaces the pilot.
- Can be installed and configured as necessary.
  - Full root access at boot time.
  - Limit user access at execution time.
- Executes:
  - DIRAC Configuration (private) Slave:
    - reduce bootstrap time for any DIRAC component.
  - DIRAC Job Agent:
    - matches and executes payload.
  - DIRAC **VirtualMachineMonitorAgent:**
    - Report status and usage of VM.
    - Asynchronously uploads output data.
    - Halts the VM is requested or if idle.

Pilots -> Virtual Machines (II)

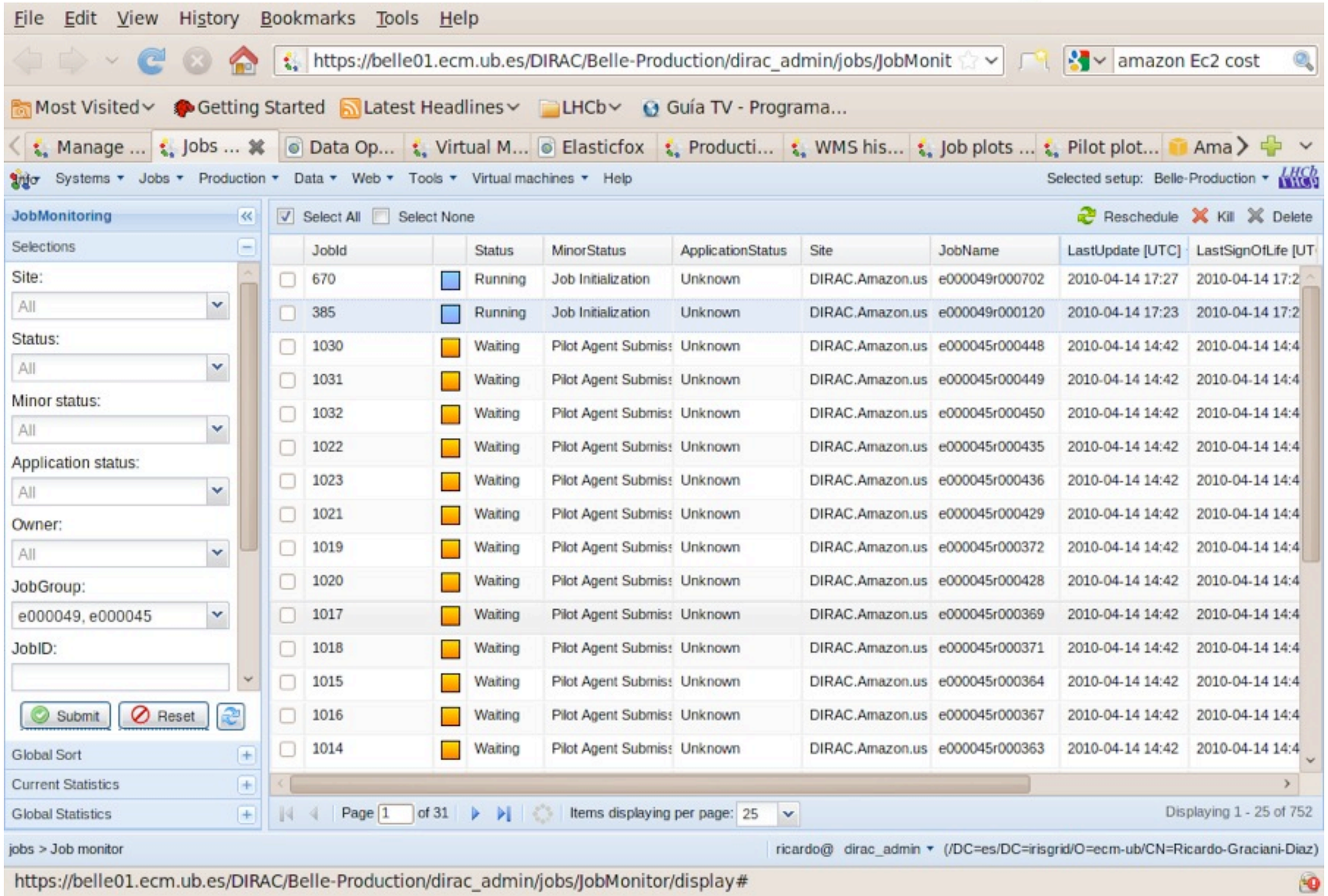
New DIRAC WMS extensions:

- **Virtual Machine Manager** (+DB):
  - Persists information relative to Virtual Machines
  - Receives heart beats from VM's (Monitor Agent)
  - Monitors status
- **Virtual Machine Scheduler**:
  - Checks:
    - TaskQueues (pending Jobs)
    - Running/Submitted VM's
    - Configuration Parameters
  - Decide wether new VM's have to be requested:
    - requires appropriated credentials for provider (Amazon)
  - Registers in the DB the submitted VM's.
  - Can be executed remotely (for absolute privacy of credentials).

And there we go !

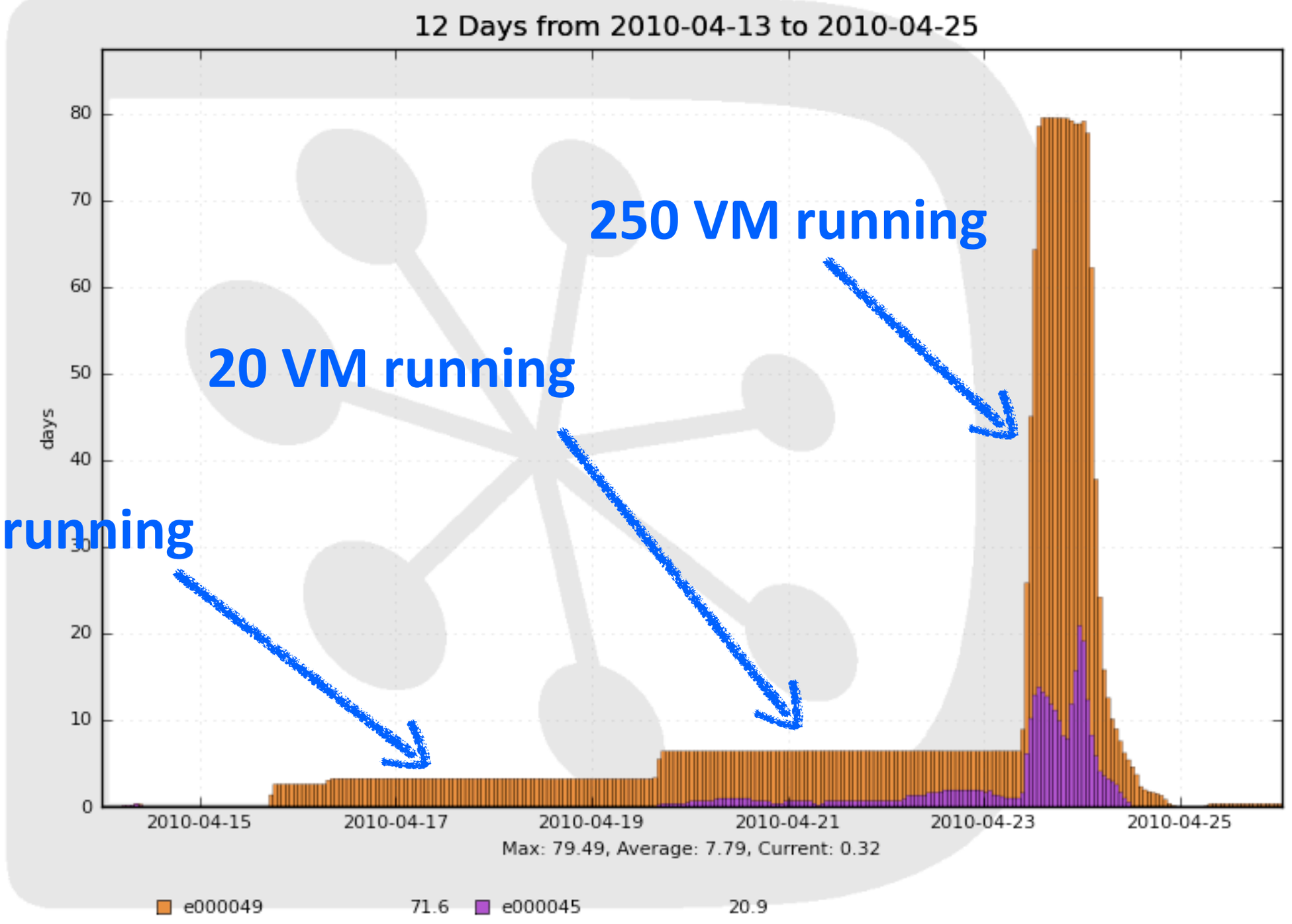
- Tasks taken from Belle official MC requests.
- Preparations:
  - "Execution VM" image with preinstalled Belle SW and **DIRAC**.
  - "SE VM" image with **DIRAC** FC/SE and postgresql DB.
- Input at KEK:
  - collection of bash scripts
  - event description and backgrounds (1 GB/ 1 M evt)
- Output sent back to grid SE's (GRIDKA & KEK).
  - 22 GB / 1 M evt

Starting up (April 14th)



The execution: Phase I

CPU days consumed by simulation Experiment / hour

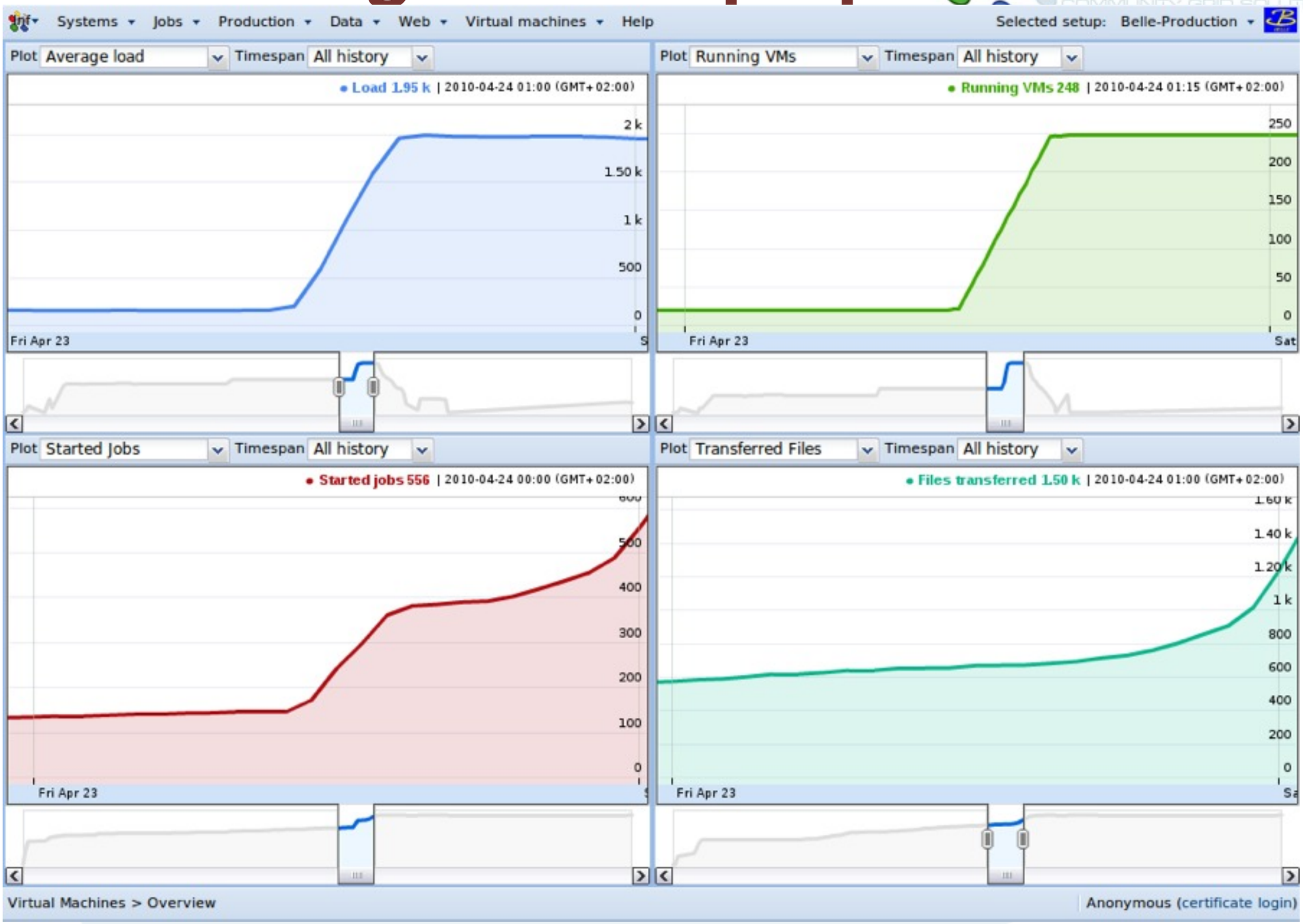


10 VM running

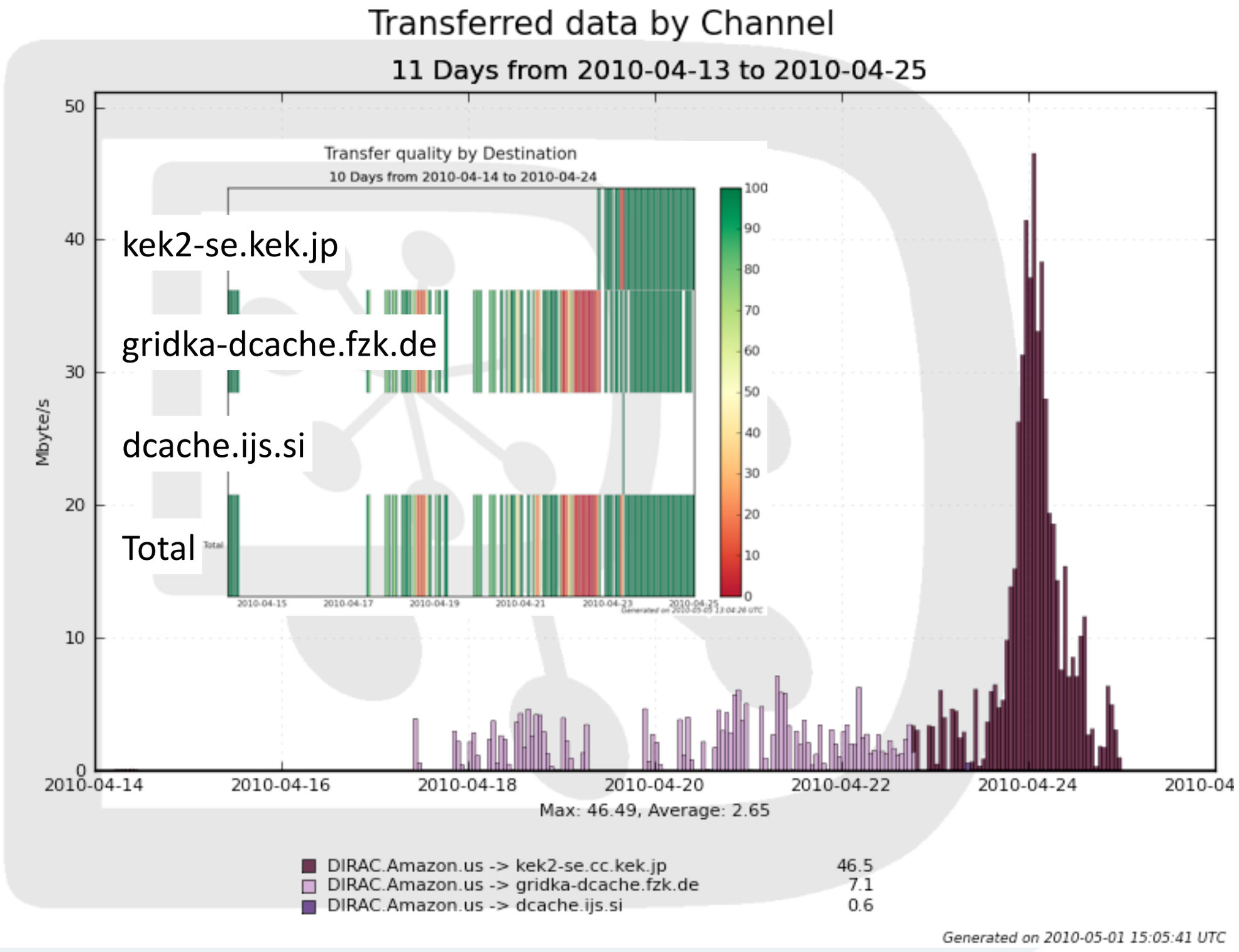
20 VM running

250 VM running

Monitoring the ramp up

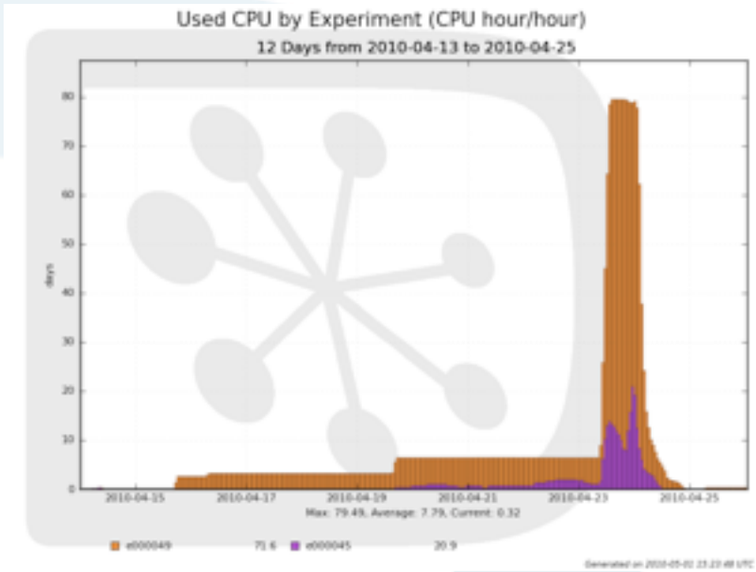


And the data back to grid

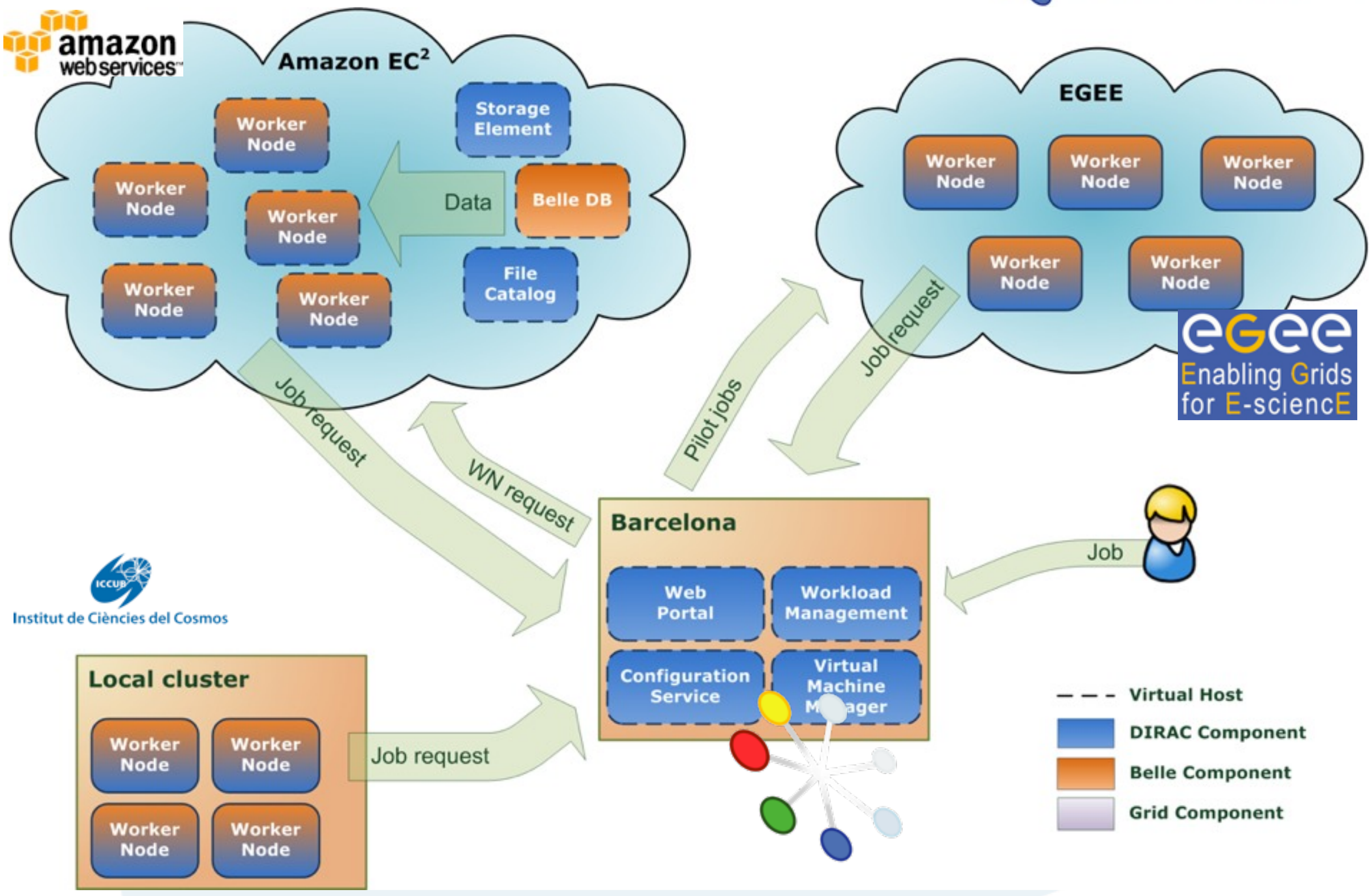


Results (I)

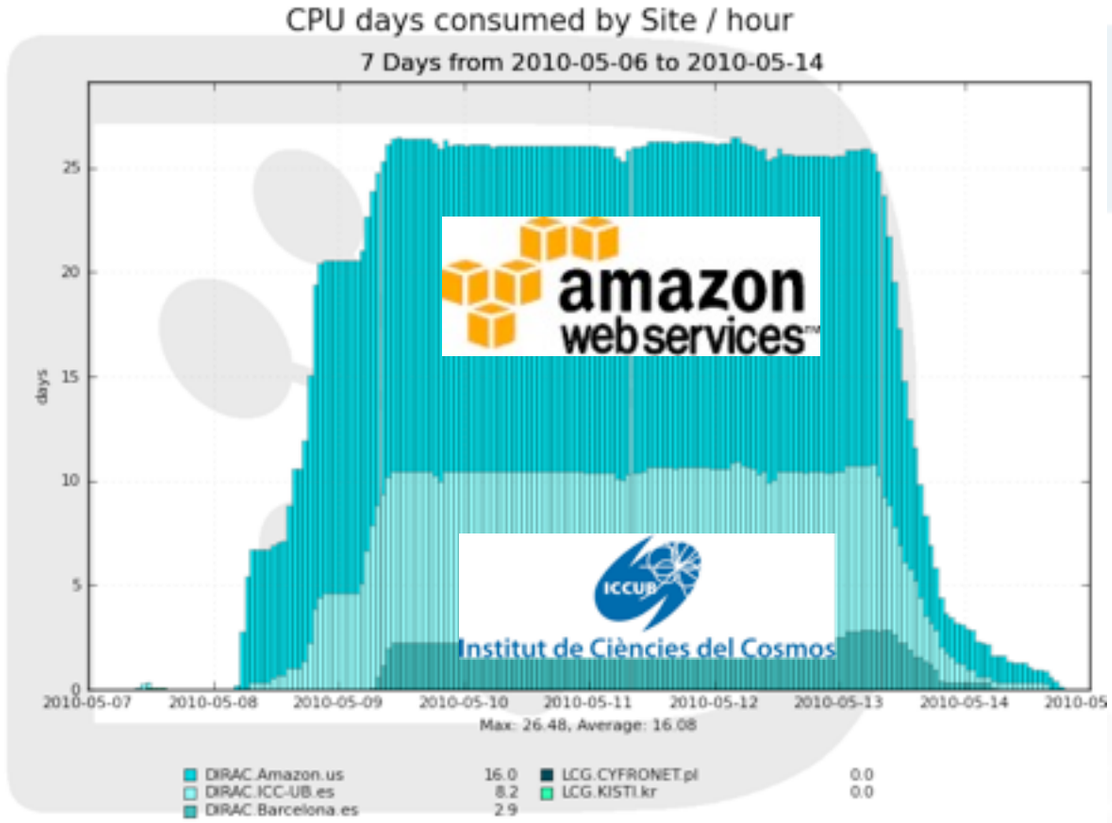
- Phase I (cloud test):
  - ☑ production ready:
    - 5% of Belle production in 10 days
    - 120 M evt (~2.7 TB)
    - 2250 CPU days used
  - ☑ proven stability and scalability:
    - 2000 CPUs peak achieved in < 4 hours
    - > 90 % efficiency in CPU usage
  - first cost estimation:
    - 0.46 USD/10k evt
  - input data pre-uploaded to Amazon SE VM.
  - few bug fixes

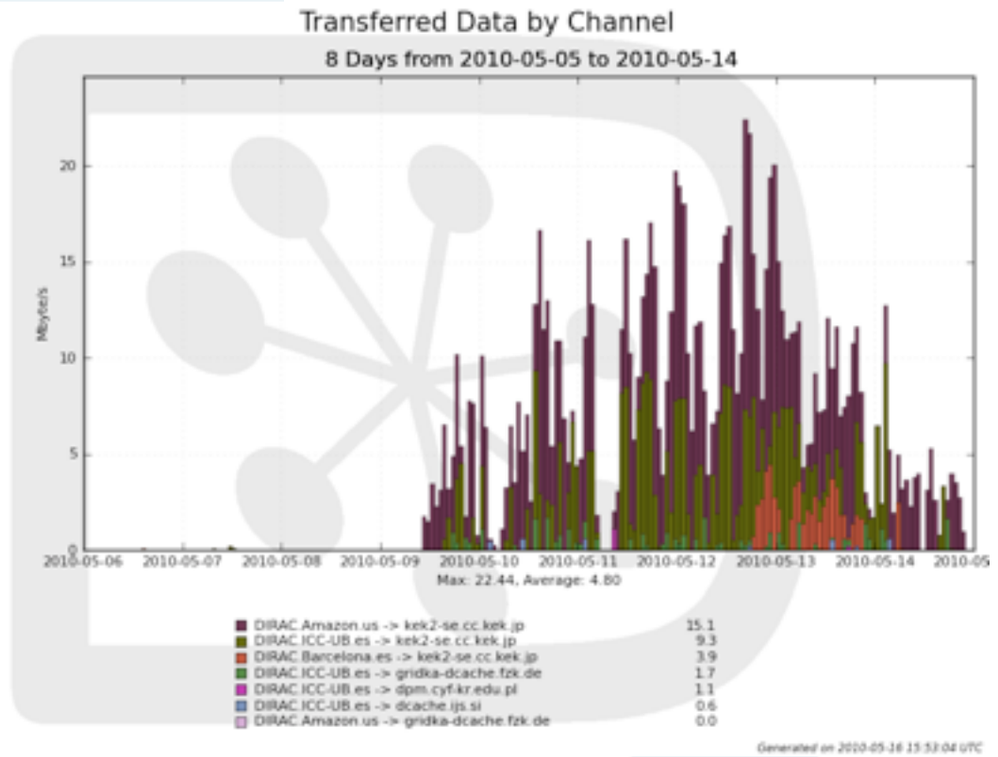


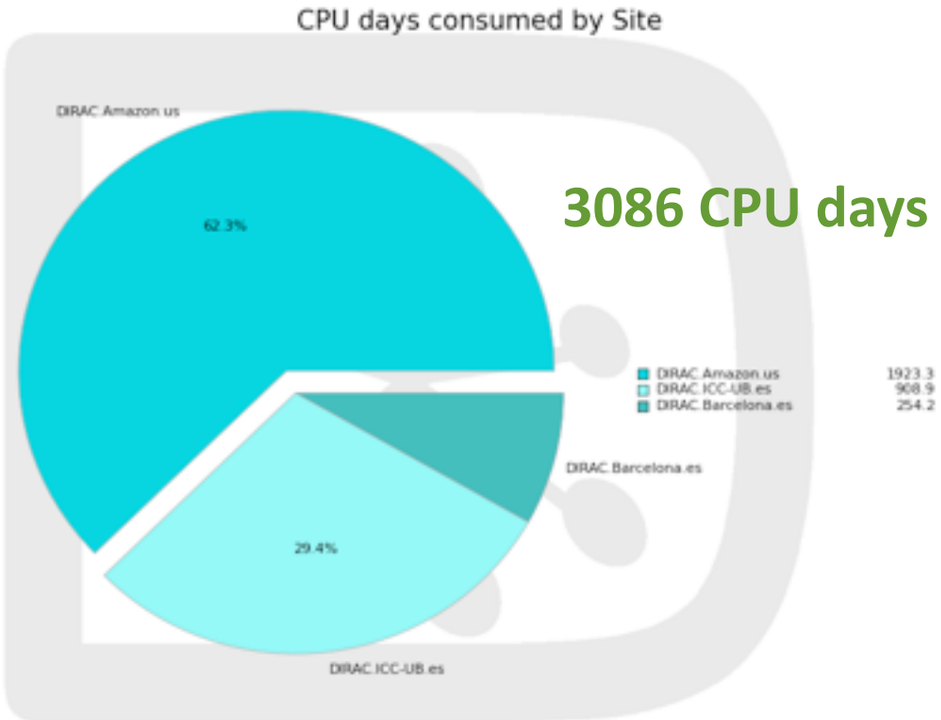
Next steps


amazon web services
Amazon EC²
Worker Node
Worker Node
Worker Node
Worker Node
Worker Node
Storage Element
Belle DB
File Catalog
Data
EGEE
Worker Node
Worker Node
Worker Node
Worker Node
Worker Node
Job request
Pilot jobs
WN request
Job request
Barcelona
Web Portal
Workload Management
Configuration Service
Virtual Machine Manager
Job
Institut de Ciències del Cosmos
Local cluster
Worker Node
Worker Node
Worker Node
Worker Node
Job request
Virtual Host
DIRAC Component
Belle Component
Grid Component

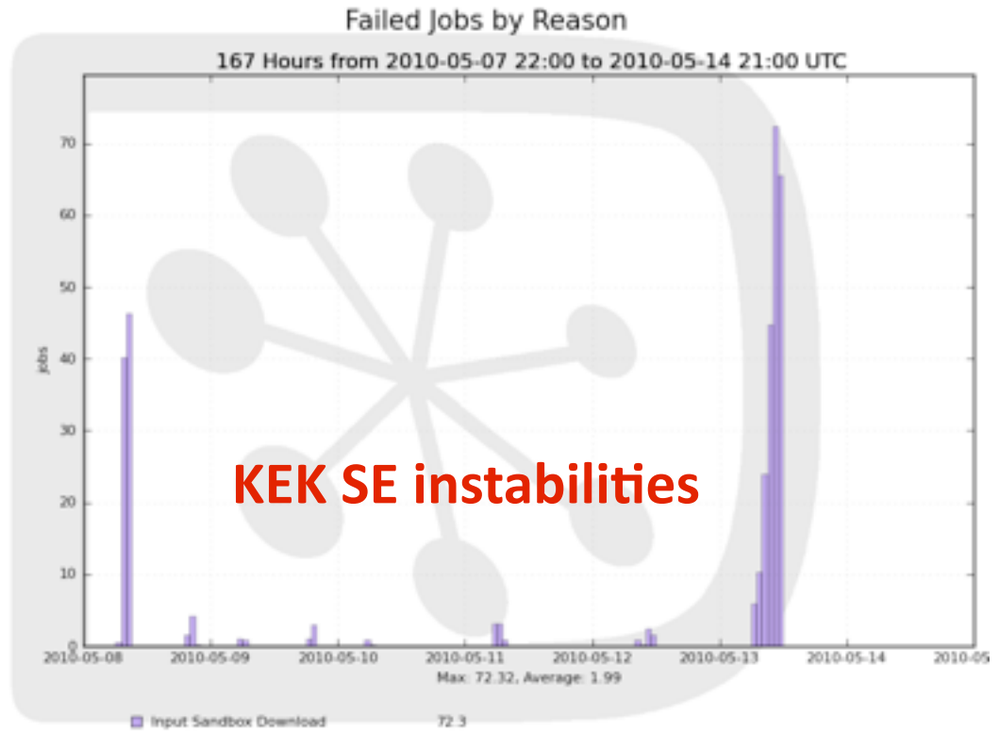
Cloud + Local: Phase II







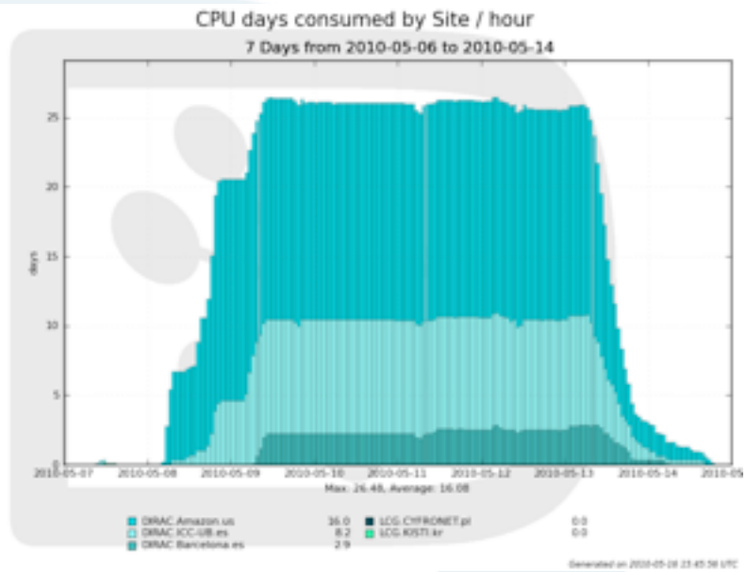
3086 CPU days



KEK SE instabilities

Results (II)

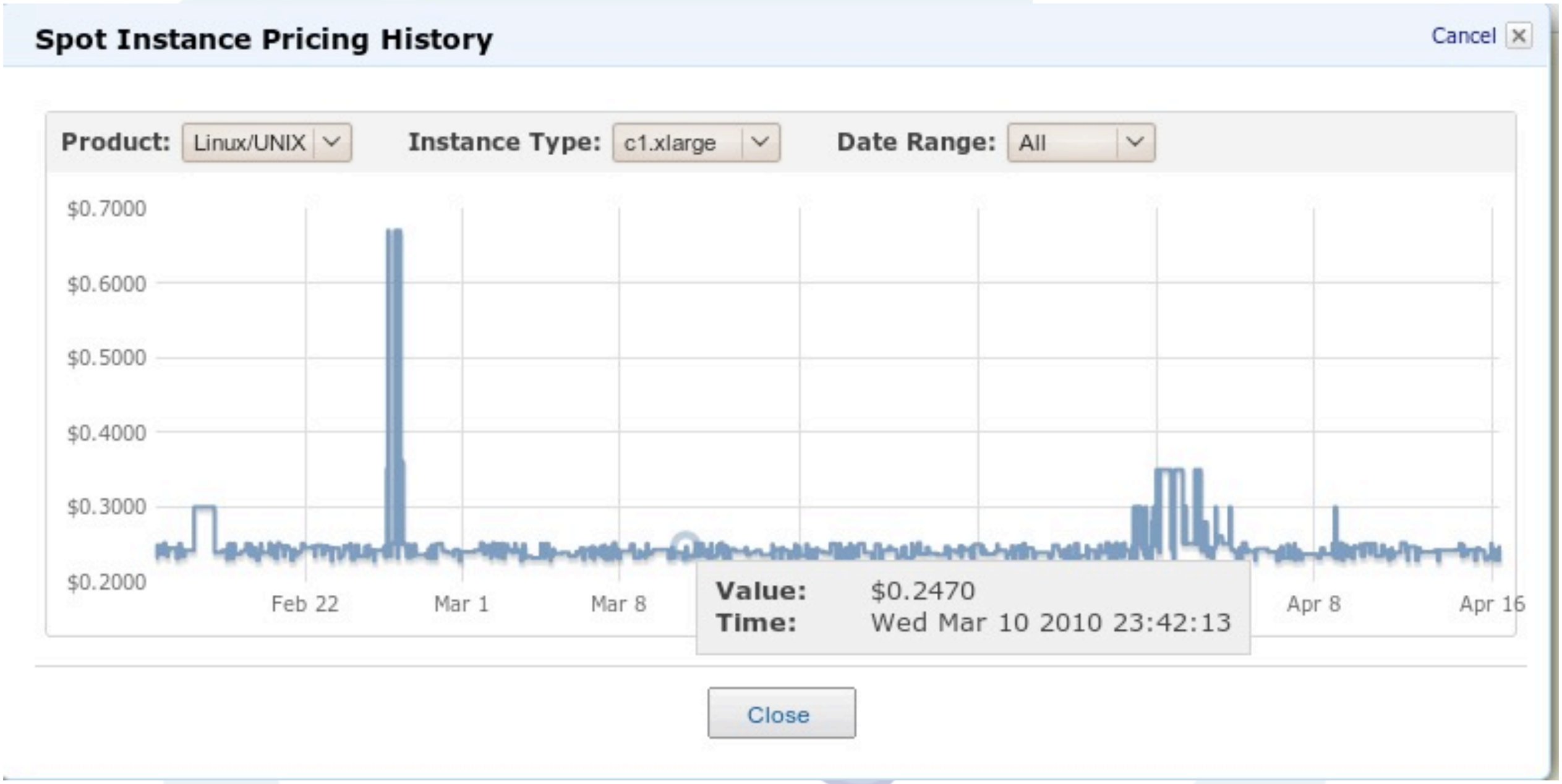
- Phase II (local + cloud integration):
  - ☑ production ready:
    - 7% of Belle production in 6 days
    - 170 M evt (~3.6 TB)
    - 3100 CPU days
  - ☑ proven interoperability:
    - 60 % cloud resources / 40 % local resources
    - transparent integration of of heterogeneous resources
    - > 95 % efficiency in CPU usage
  - ☑ improved cloud cost:
    - 0.20 USD / 10k evt
  - ☑ input data downloaded from KEK SE
  - – issues with stability of grid Storage



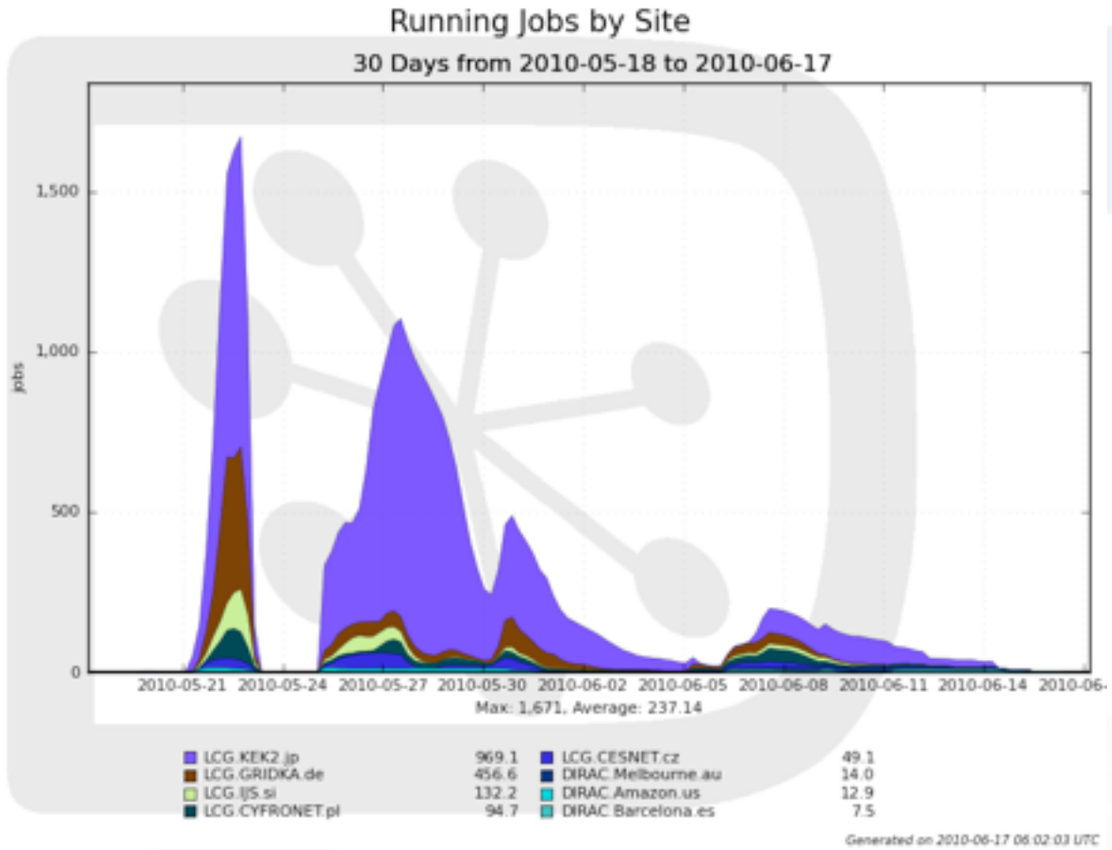
On Amazon Cost

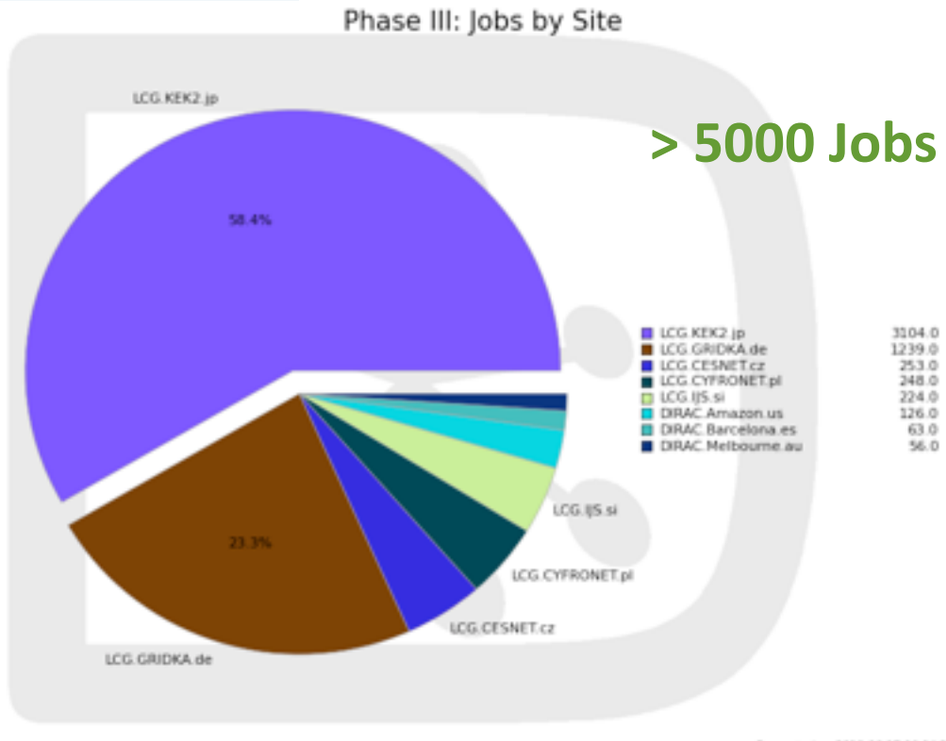
- In all cases we used the High-CPU Extra Large Virtual Machine, that offers the best CPU/cost rate:
  - 7 GB of RAM
  - 8 Cores (20 EC2 Units or 67 HEP Spec 2006)
  - 1.7 TB disk
- Purchase options:
  - On-demand (most expensive): **0.68 USD/hour**
    - fix price, request & pay
  - Reserved (cheapest): **0.24 USD/hour**
    - fix price (1-3 years), pay & use
  - Spot:
    - price fluctuates, request (with top price) & pay

Spot Instance Price History

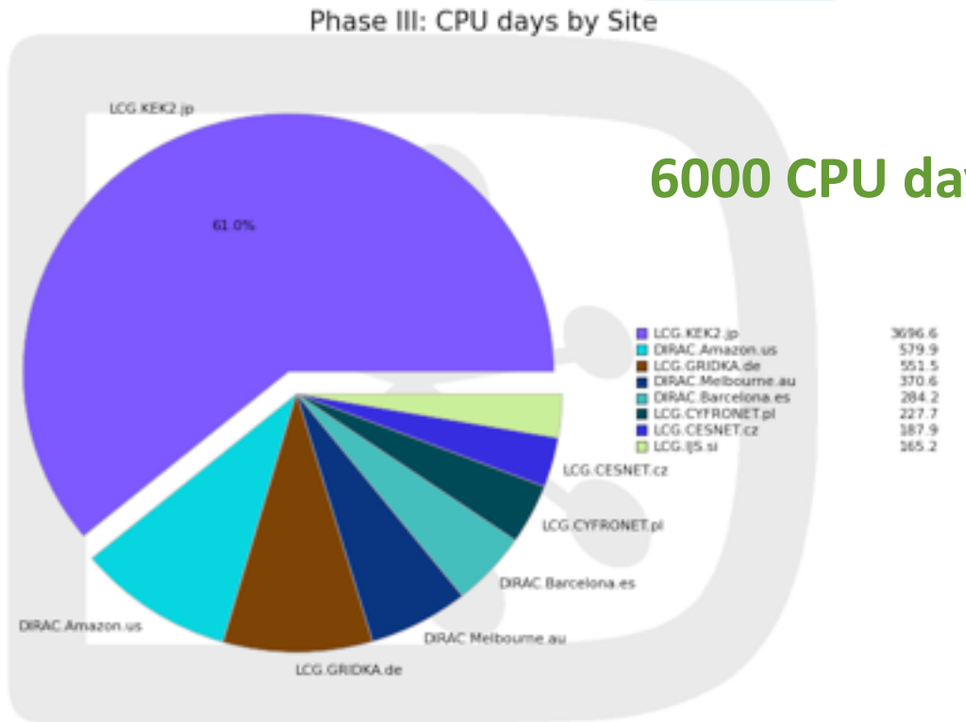


Cloud + Local + Grid: Phase III





> 5000 Jobs



6000 CPU days

Different issues with SE at KEK:

- gridftp hosts restarting.
- limited number of gridftp servers.
- control channel timeout.
- SE getting full.

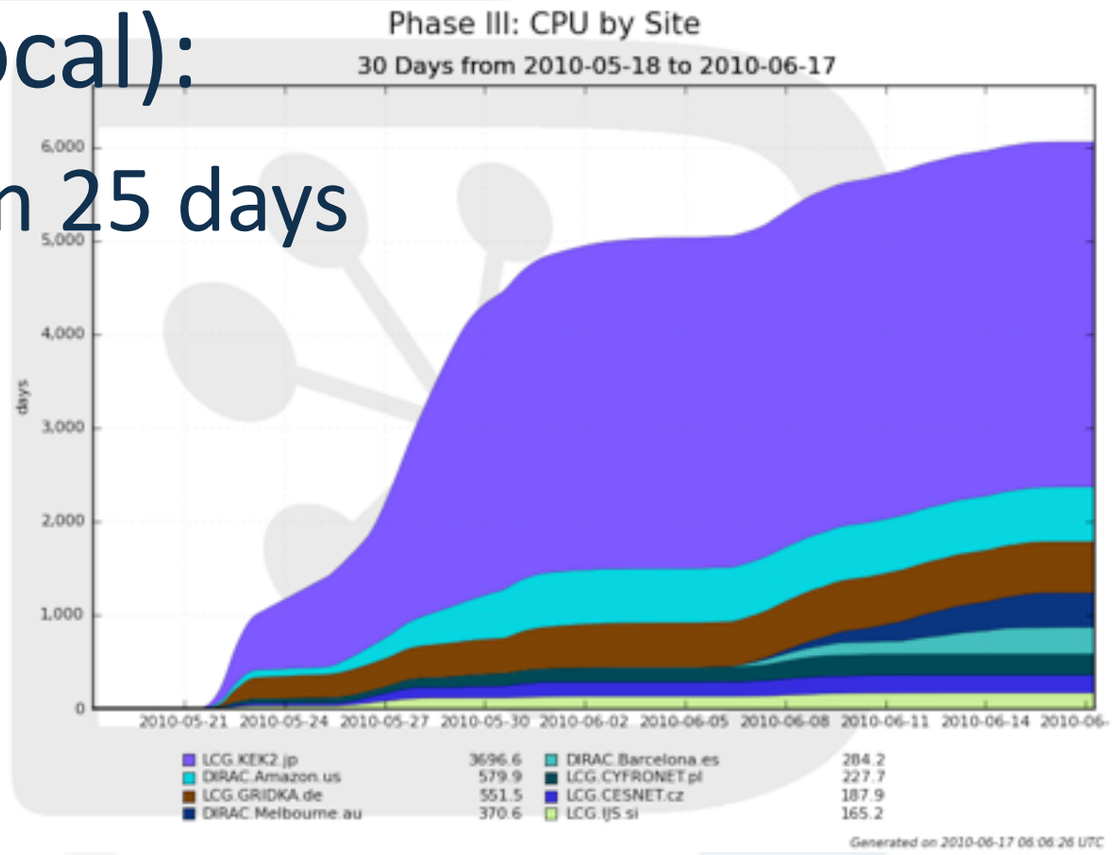
Random SW errors when executing on the grid

- Not seen in local or cloud (multi-process)

Issues with queue lengths

Results (III)

- Phase III (grid + cloud + local):
  - ☑ 15% of Belle production in 25 days
  - –Still running:
    - 6000 CPU days
  - ☑ Full interoperability:
    - cloud + grid + local
    - stability issues on grid SE.
    - main effort to make sure grid environment is appropriate
  - –Other issues:
    - ☑ Jobs can be very long, 1-30 days, only the shorter ones are appropriate for the grid



Summary

- Perfect behavior of DIRAC - Amazon integration:
  - >95% CPU usage efficiency.
  - No errors due to Amazon.
  - 0.20 USD / 10k evt (2009 estimate: 0.68 USD / 10k evt)
- Perfect integration of Local resources:
  - >95% CPU usage efficiency.
  - No errors due to Local clusters.
- Issues with grid resources:
  - Need to add extra redundancy and checks
    - Integrate execution in DIRAC framework
    - Add failover SE's
    - Need to test and validate resources (CE's and SE's)
- Main issue access to "large" input data files