Workload management

FJPPL Computing Workshop

Nadia LAJILI, Suzanne POULAT

& CCIN2P3 Batch Team

Overview

- History
- Workload management
- Pros and cons of GE
- Incident management
- Batch and cloud

History

- BQS : Home made batch system
- OGE : 2011-2013
- Univa GE : since 2013
- SL6 – UGE 8.1.6
- ~23000 virtual cores : one instance for all our needs

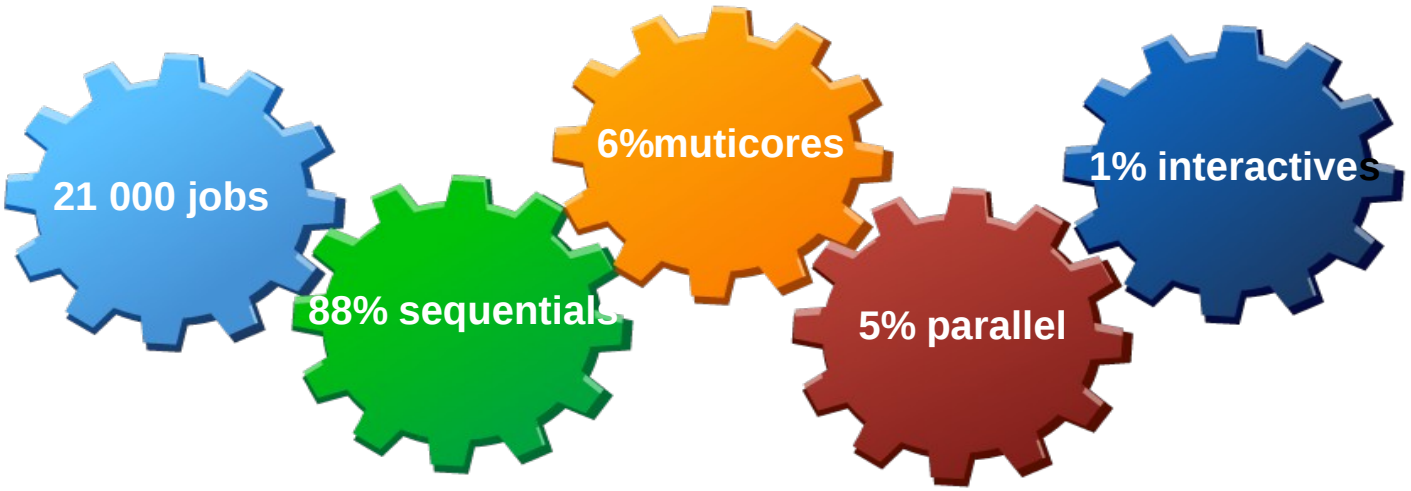
ORACLE support was not satisfactory

"HEP world" faced new requirements :
- **VM,**
- **multicores,**
- **interactive**
- **increase of the needs**

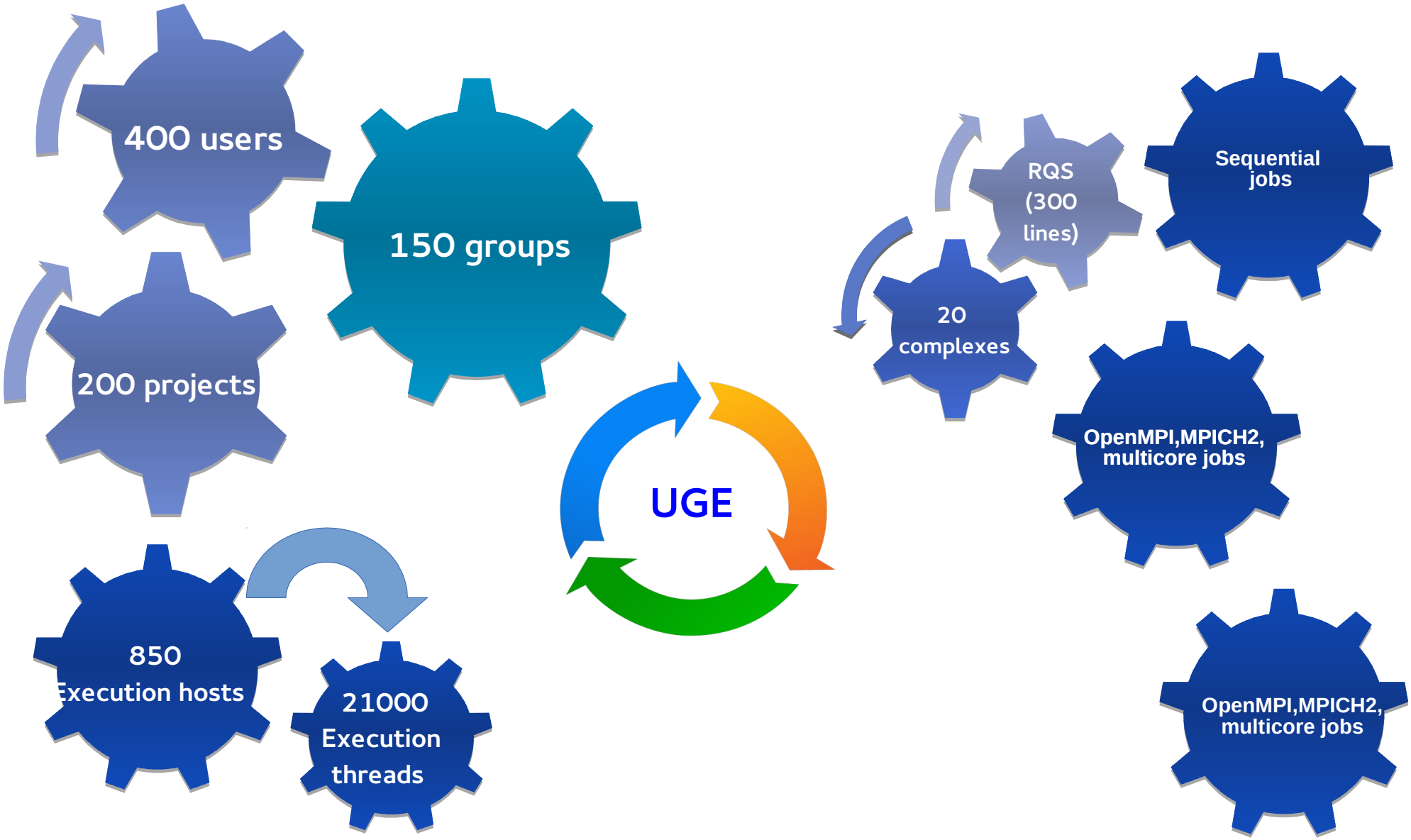
1 FTE for Administration & Operation

A few numbers

- Jobs
  - ~12 000 pending jobs, some are array jobs : 40 000 pending tasks
  - ~21 000 running jobs, some are parallel : 22 000 used slots
  - > 110 000 ended jobs / day
  - > 600 000 qstat / day



Overview


400 users
150 groups
200 projects
850
Execution hosts
21000
Execution
threads
UGE
RQS
(300
lines)
20
complexes
Sequential
jobs
OpenMPI,MPICH2,
multicore jobs
OpenMPI,MPICH2,
multicore jobs

Workload management

- **To regulate the load on the system :**

  Global configuration

  - Fair share (two levels) on group & on projects (200) using <share tree policy>
  - Job flow regulation : via complexes (20) and intensive usage of Resource Quota Sets (340 lines)
  - Scheduler limitations : SCHEDULER_TIMEOUT, MAX_SCHEDULING_TIME, MAX_DISPATCHED_JOBS
  - Manual adjustement for cluster CPU optimization using <override policy> to increase the priority of pending jobs

Batch resources configuration

- Load sensors for disk space and memory usage, integration in "load formula"

  The way to check if the worker is able to run jobs

- Fix the global number of slots and the number of slots by queue according to hardware limit of the worker node

Workload management

- **To prevent the system from heavy load :**
  - **Monitoring platform based on NAGIOS**

– Nagios probes for master monitoring (load)
– Nagios probes checking job efficiency

  - **Home made monitoring tools**

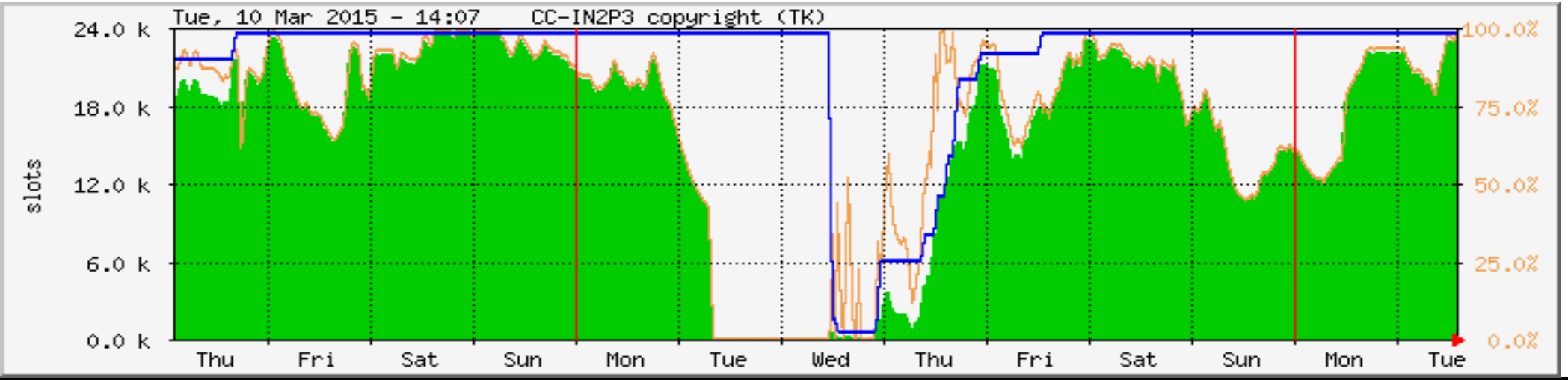
– Dedicated process parsing GE master error emails, block and notify users having problematic jobs
– Cron jobs producing hourly statistics on jobs efficiencies

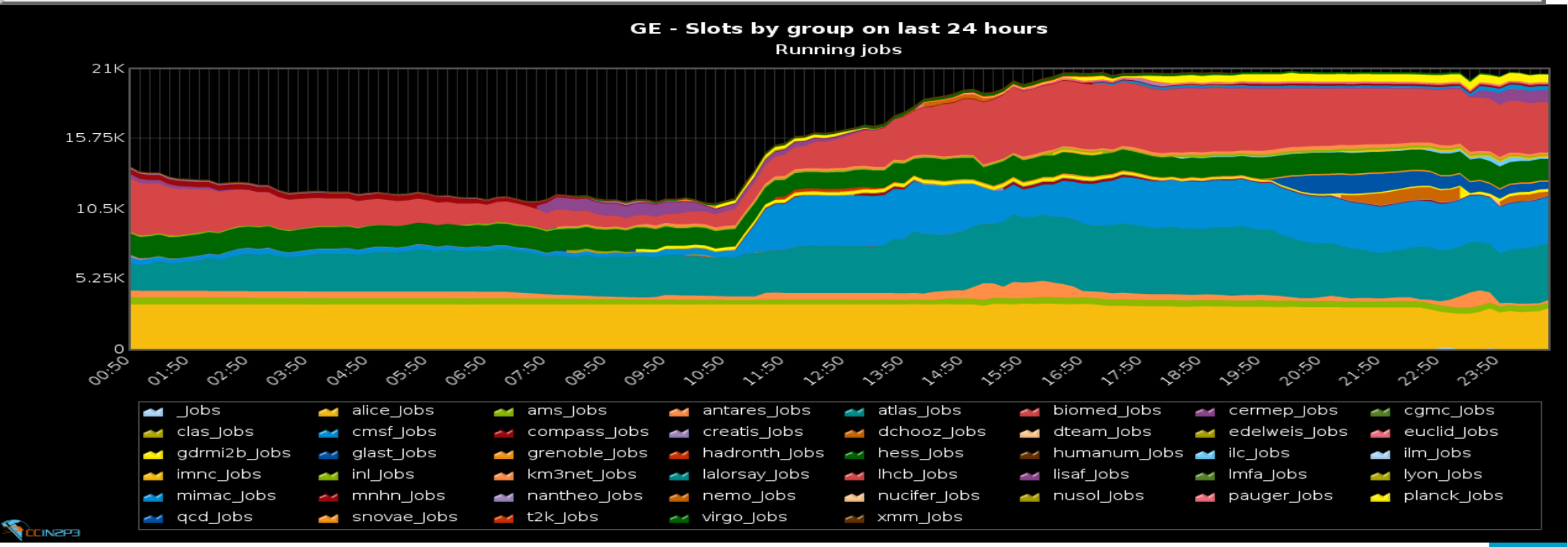
  - **Web portals**

– Main parameters of the system are available (Kibana)
– Many web pages on cluster usage per group (MRTG)
– Real-Time metrics on Running/Ended jobs, workers efficiency (SYMOD)

Workload management

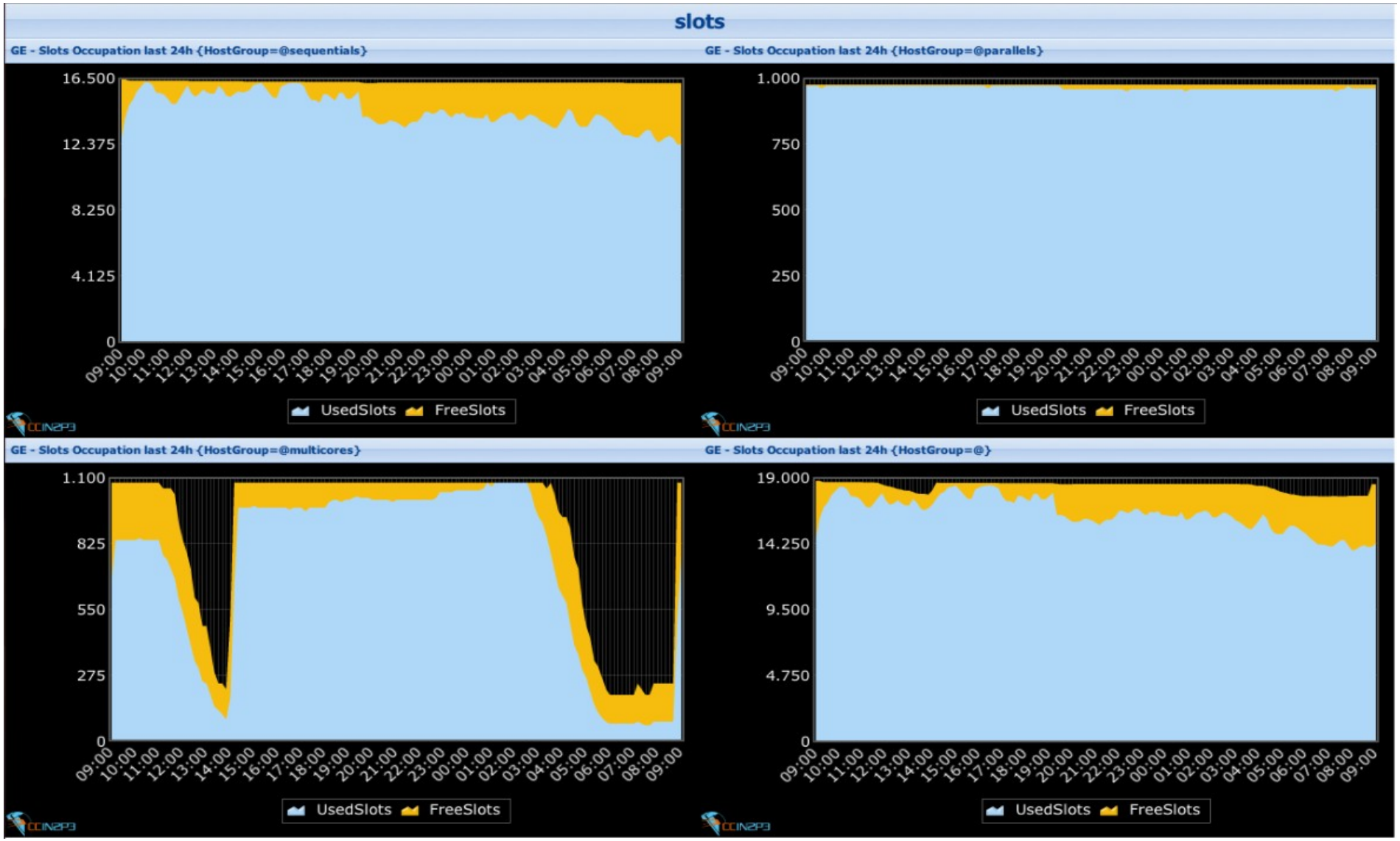
- **Current production by MRTG/SYMOD**





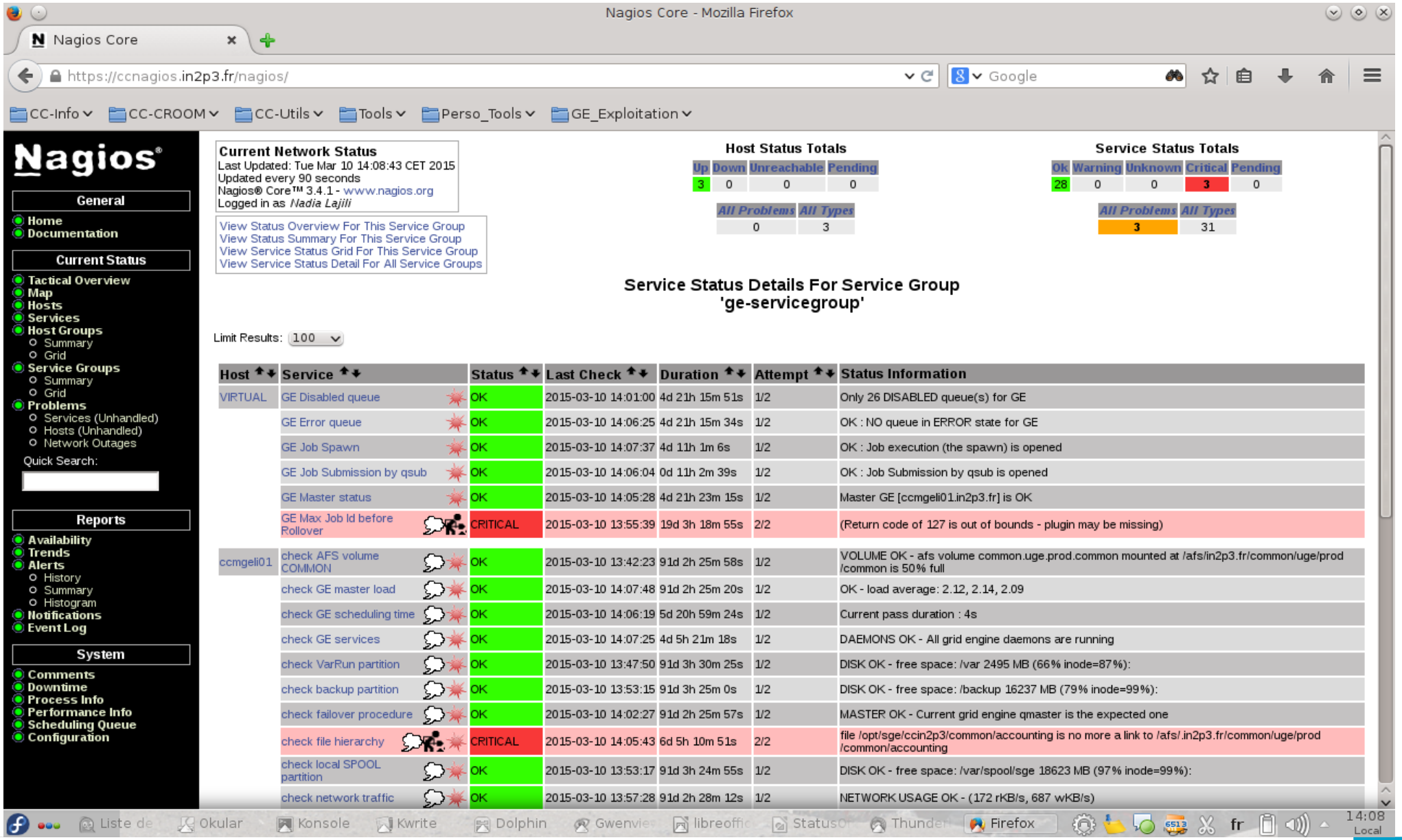
Workload management

- **Slots occupation by type (SYMOD)**



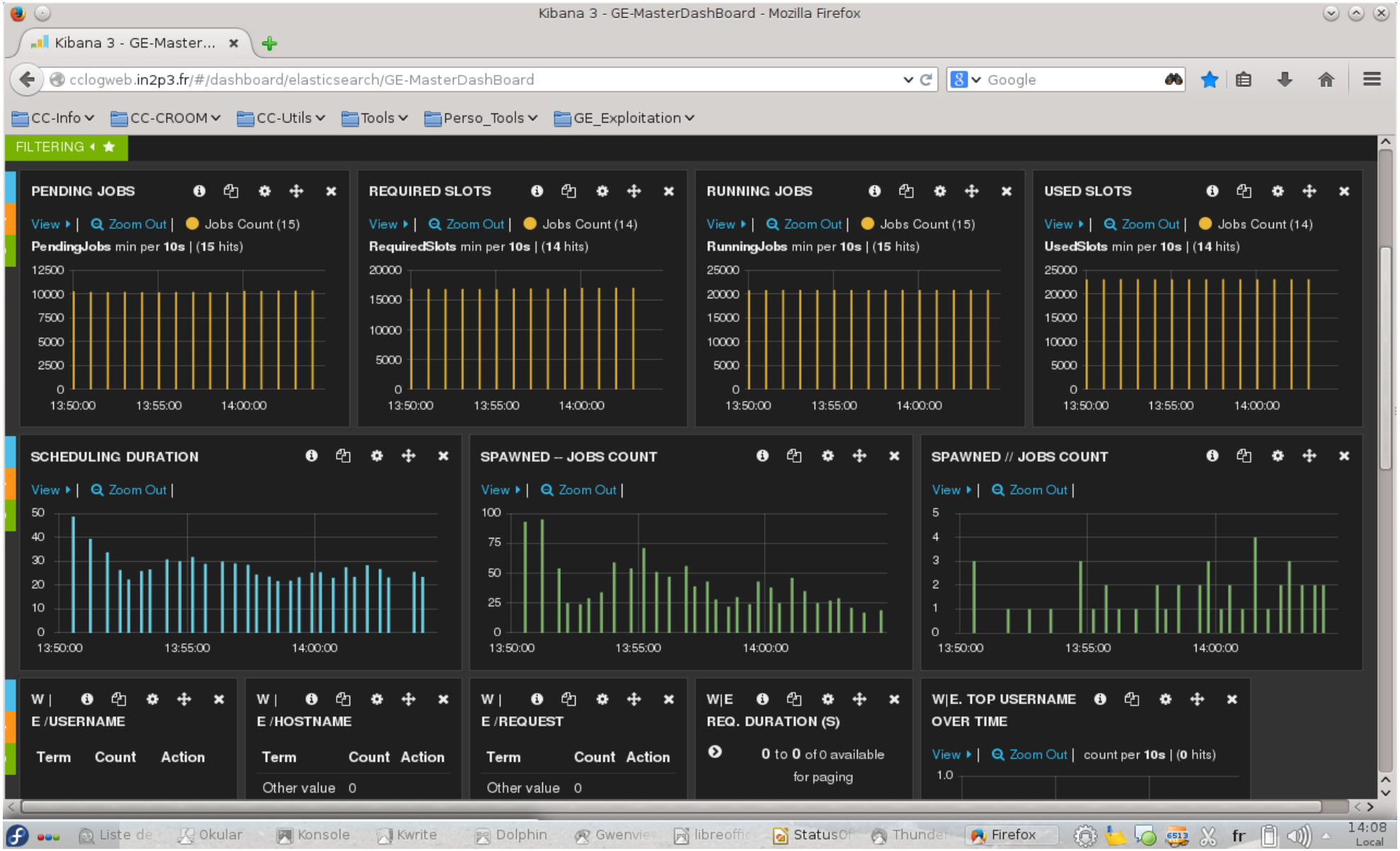
Workload management

- **BATCH Monitoring by NAGIOS**



Workload management

- **Main system parameters and activity by Kibanna**



Pros and Cons of GE

- **Pros of UGE**
  - **Many functionalities :**
    - To distribute fairly resources
    - To regulate load on storage services (RQS, complexes)
    - To optimize cluster CPU usage
  - Patch produced for our environment
  - **Failover procedures** (using a shadow master or a cluster of shadows)
  - **Stable service** (spooling in Postgres DB)
  - **Accounting** (Ended jobs) in Postgres DB (ARCO)

Pros and Cons of GE

- **Pros of UGE**
  - **Good support :** reactivity and user needs
  - **Visibility of the roadmap,** scalable software
  - Regular bug corrections, new releases, improvements (thread RO)
  - Active community (user forums, webinars)

Pros and Cons of GE

- **Cons of UGE**
  - A few incidents related to overload and bugs
  - Can be disturbed by a large amount of short jobs **(better in 8.2.1?)**
  - Very few tools to debug when the system gets in trouble
  - Not possible to mix sequential and muticore jobs in the same @hostgroup
  - In some cases (known bug) : lost of running jobs
  - **RFE not yet implemented :**
    - Merge of qacct, qstat commands
    - Reject job submission when encountering impossible resource requirement specification
    - A way to grant a minimum of running jobs per user
    - Change task priority of an array job

Incident management

- **Detection**
  - **Nagios probe monitoring**
    - GE main components (qmaster, shadow, scheduler...)
    - GE global configuration
    - Queue status (disabled,errors queue)
    - The production (jobs short, failling ..)
  - **Monitoring of the Master node (SMURF)**
    - CPU, memory usage
    - Disk I/O, filsystem, network, processes, load
  - **Kibana portal**
    - Checking GE main parameters (scheduling time, dispatched jobs…)
    - GE global load activity (Jobs R&Q, qacct,qstat requests)
  - **SYMOD**
    - Produce information about pathological jobs

Incident management

- **Solutions**
  - Delete the failing jobs and lock the corresponding user account
  - Remove problematic workers from production
  - Restart completely the service
  - Send request to support in case of misbehaviour

Cloud and batch

- **Cloud Integration in GE with UniCloud software**
  - UniCloud tested in March 2014
  - Possibilities tested :
    - Virtual machines as workers
    - Worker instantiation on the fly
  - The integration to puppet is difficult (need a dedicated server)
  - More suitable to deploy private CLOUD
  - Poorly documented

- **Cloud integration in the batch with htcondor**
  - Currently tested with ATLAS simulation jobs
    - It works : jobs are spawned on VM nodes by the scheduler
    - 20% overhead raised on the system

Questions