Peer-to-peer cooperative scheduling architecture for National Grid Infrastructure

L. Matyska, M. Ruda, S. Toth

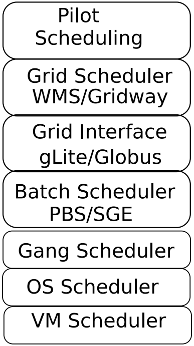
CESNET
Czech Republic

10th March 2010

Job scheduling in Grid

- Many approaches and types of schedulers in standard grid

- Multi-layered approach
- Grid middleware usually deals with the three top layers
  - Pilot scheduling usually more user-centric
- Usually requires remote services available
  - Often leads to local by-pass and direct cluster submits



$\mathcal{META}$ Centrum

$\mathcal{META}$ *Centrum* (`http://meta.cesnet.cz`)

- Anyone remembers term metacomputing?
- Czech national grid infrastructure
  - Under umbrella of CESNET
- Computational resources
  - Mostly clusters
  - Installed across country, centrally managed
- The same team involved in EGEE
  - Computing site, user and VO support, gLite development
- Virtualization and job scheduling as one research focus

Current $\mathcal{META}$ *Centrum* scheduling

Basic features

- Relies on batch schedulers more than usually
- Global batch system instead of multi-level scheduling
- Standard grid interface (gLite/Globus) also available
- Integrated with scheduling of virtual machines

Based on a central PBSPro installation

- Central knowledge of system's state
  - Easy implementation of global scheduling policies
    - Fairshare
  - Avoid problems with multi-level schedulers
    - Job stalled when waiting for cluster in maintenance
    - Local jobs not visible to global scheduler
  - Support for large, multi-site jobs

Deficiencies of the used approach

- Scalability
  - Adding new sites increases burden on central scheduler
- Stability of central-server based solution
  - Just limited support for wide area replication
  - Inability to submit new jobs if central service not up/available
- Local un-usability of a disconnected cluster
  - Leads to frustrated users, by-passing the $\mathcal{META}$ *Centrum* scheduling

Not able to cope with the planned major extension of the national grid infrastructure

New scheduling architecture

Motivation

- Keep positive aspects of a centralized solution
  - Especially the ability to take global decisions
  - While not introducing multi-level scheduling
- Remove (some of) negative aspects of a centralized solution
  - Scalability
  - Use of disconnected resources

General features

- Self-contained scheduler at each site (or even a large cluster)
  - Always able to accept jobs for the whole infrastructures
  - Always able to submit jobs to the local cluster
- Cooperating with similar schedulers at other sites
  - Exchanging information about the whole infrastructure (global state)
  - Ability to make a “global” decision
  - Moving jobs directly between schedulers

New scheduling architecture

Motivation

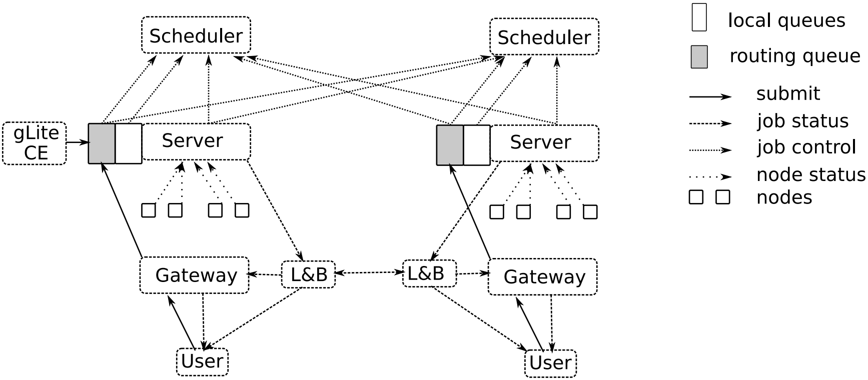
- Keep positive aspects of a centralized solution
  - Especially the ability to take global decisions
  - While not introducing multi-level scheduling
- Remove (some of) negative aspects of a centralized solution
  - Scalability
  - Use of disconnected resources

General features

- Self-contained scheduler at each site (or even a large cluster)
  - Always able to accept jobs for the whole infrastructures
  - Always able to submit jobs to the local cluster
- Cooperating with similar schedulers at other sites
  - Exchanging information about the whole infrastructure (global state)
  - Ability to make a “global” decision
  - Moving jobs directly between schedulers

CESNET

Proposed architecture in more detail



Architecture implementation

Basic features:

- Torque in the hearth of each local scheduler
- Extended with
  - A gateway interface to accept jobs and store them into a routing queue
  - A “global” scheduling strategy
- L&B from gLite as the persistent information storage for job monitoring

Lead on each site to:

- “Standard” Torque instalation
- Extended scheduler managing jobs from more servers
- Jobs submitted through gateway to routing queue
- Scheduler
  - Moves job to a different server where job has to be started
  - Moves job to a local queue where job is started
- Jobs monitored from any gateway, job information stored in L&B

Main development tasks

Cooperative scheduling

- Torque enhancements to support peer-to-peer scheduling
- Maintenance of globally available information used for scheduling
  - Fair-share is using actual accounting information
- Support for multi-site jobs
- Scheduler extensions

PBSPro originally used for better stability across Czech Republic

- Switch to Torque
  - Need to port Kerberos support
  - Need to port scheduling enhancements
- Support for management of virtual machines
  - Magrathea system (extending node states)
  - Direct support for virtualized fabrics must be ported to Torque too

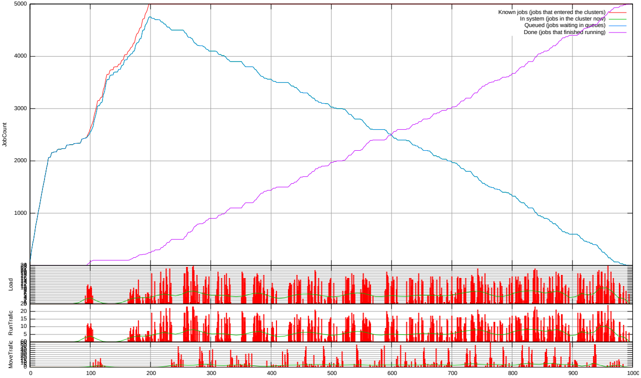
Current status

- Peer-to-peer extensions—prototype done, reasonable overhead
- Fair-share—simple solution done, more development later
- Multi-site jobs—several possibilities in discussion
- Torque scheduler extensions—on-going work
- Kerberos support ported
- Magrathea support—on-going work
- Gateway and L&B usage—next phase

Peer-to-peer overhead—Experimental setup

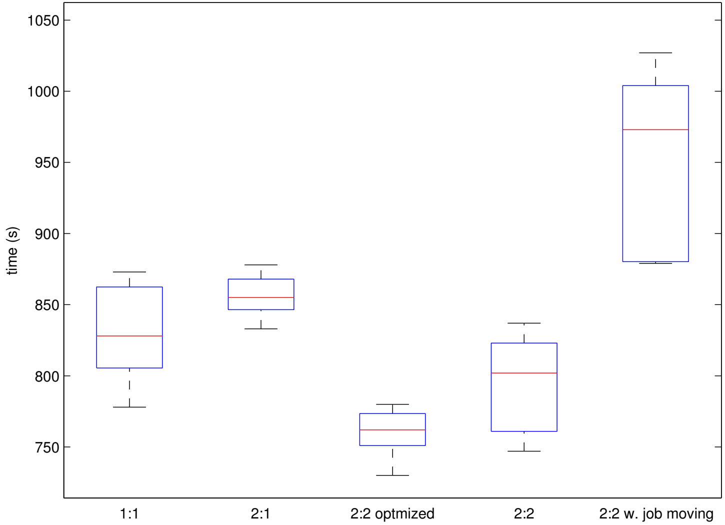
Series of measurements

- Realistic simulation of a production environment using light-VM extension of Linux kernel
- 5000 jobs submitted to 200 nodes on up to 5 sites
- All the jobs run

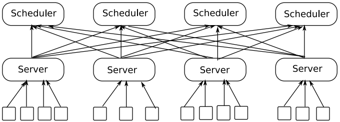


Peer-to-peer overhead—Experimental setup

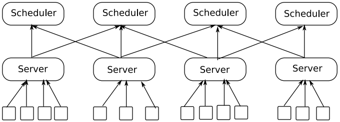
Interaction between schedulers and sites



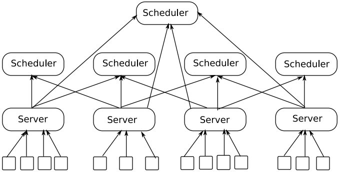
Communication scheme



Original proposal, full information everywhere



“Neighbor” approach, information routing



On demand super-scheduler for multi-site jobs

Conclusion

- Cooperative scheduling architecture supports
  - High scalability (esp. with a proper communication scheme)
  - Independence on remote services and local submit
  - Ability to make decisions based on global state
  - Free job movement between sites based on local scheduler decision
  - Direct inclusion of virtualized resources
  - Easy integration of different gateways (e.g. gLite CE interface)
- Its *META Centrum* implementation underway
  - Based on a Torque system
  - Extended to multi-site scheduling
  - *META Centrum* native gateways
  - Use of gLite L&B for job monitoring
- Initial experiments encouraging (acceptable overhead for peer to peer communication)
- Expected to be in full production already this year

Thank you
Questions?