Domain-oriented services and resources of Polish Infrastructure for Supporting Computational Science in the European Research Space – PLGrid Plus

QosCosGrid Middleware
Tools and Services for Advanced Job Management, Advance Reservation and Co-allocation of Computing Resources

Tomasz Piontek, Krzysztof Kurowski
Poznan Supercomputing and Networking Center

EGI Community Forum
Helsinki 19-23.05.2014
Outline

- QCG general information
- QCG Architecture & Components
- Tools for end-users
- Success stories
- Integration with EGI
- Documentation & Repositories
- Contact
- Questions
The QosCosGrid (QCG) middleware is an integrated system offering advanced job and resource management capabilities to deliver to end-users supercomputer-like performance and structure. By connecting many distributed computing resources together, QCG offers highly efficient mapping, execution and monitoring capabilities for variety of applications, such as parameter sweep, workflows, MPI or hybrid MPI-OpenMP.
QCG functionality

• Automatic steering of various types of complex computing experiments:
  • Simple tasks
  • DAG Workflows with recursive conditions and dependencies on any task status
  • Multi-Dimensional parameter sweep tasks
  • Single cluster Parallel tasks (MPI/OpenMP)
  • Cross-Cluster Parallel tasks
QCG Functionality

- Advance reservation capabilities
- Quality of Service
- Co-allocation of resources and cross-cluster
- Support for interactive tasks
- Possibility to connect to running task with interactive session
- Task status and progress notifications
QCG-Computing

• Deployed on access nodes of the batch systems (SGE, Slurm, torque/maui, LoadLeveler, PBS Pro, Condor, Apple Xgrid)

• Provides remote access to task submission and advance reservation capabilities of LRMS via DRMAA interface

• Compatible with the OGF HPC Basic Profile specification (JSDL and BES)

• Offers basic file transfer mechanisms
QCG-Notification

- Supports the topic-based publish/subscribe pattern for asynchronous message exchange
- Serves as the main message bus between the services, applications and the end-user
- Is capable of sending notifications using variety of transport mechanism, including SOAP, SMTP and, what is a unique feature, the XMPP protocol
QCG-Broker

- Offers scheduling and brokering of jobs capabilities
- Controls the whole experiments execution (including workflows and parameter sweep tasks)
- Provides requested QoS and co-allocates resources
- Stages in/out files and directories
QCG Tools & Clients

- QCG-SimpleClient (command line)
- QCG-Icon (GUI)
- QCG-Science-Gateways (web)
- QCG-QoS-Access (web)
- QCG-Monitoring (web)

- QCG-Data (clone of iDrop, under development)
- QCG-Icon2 (under development)
• Set of commands patterned on queuing system tools
• JSDL, QCG-Simple, QCG-XML description dialects
• Support for interactive tasks
• Automatic staging in/out files
• Notifications about statuses and progress of application (mail, xmpp, QCG-Monitor)
QCG-SimpleClient

- **Submission and controlling of tasks:**
  - `qcg-cancel` - cancel task(s)
  - `qcg-clean` - clean the working directories of given tasks
  - `qcg-connect` – establish **interactive** session to the task
  - `qcg-info` - display detailed information about the given tasks
  - `qcg-list` - list tasks in the system
  - `qcg-peek` - display ending of (stdout, stderr) streams
  - `qcg-proxy` - create user proxy certificate
  - `qcg-refetch` - retry/repeat the transfer of output files/directories
  - `qcg-refresh_proxy` - refresh user proxy certificate for given tasks
  - `qcg-sub` - submit batch or **interactive** tasks to be processed by QCG

- **Resources reservation and control:**
  - `qcg-rcancel` - cancel reservation(s)
  - `qcg-reserve` - reserve resources
  - `qcg-rinfo` - display information about the given reservation(s)
  - `qcg-rlist` - list reservation in the system
[plgpiontek@qcg ~]$ qcg-offer

**HYDRA:**

**Summary:**

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>nodes/cores</th>
<th>share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Resources:</td>
<td>282/5340</td>
<td>100%/100%</td>
</tr>
<tr>
<td>Up Resources:</td>
<td>250/4668</td>
<td>88%/87%</td>
</tr>
<tr>
<td>Used Resources:</td>
<td>114/1710</td>
<td>40%/32%</td>
</tr>
<tr>
<td>Free Resources:</td>
<td>129/2628</td>
<td>45%/49%</td>
</tr>
<tr>
<td><strong>(FreeNodes=87x12,17x16,18x48,7x64)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PartFree Resources:</td>
<td>142/2810</td>
<td>50%/52%</td>
</tr>
<tr>
<td>Reserved Resources:</td>
<td>2/24</td>
<td>0%/00%</td>
</tr>
</tbody>
</table>

**GALERA:**

**Summary:**

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>nodes/cores</th>
<th>share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Resources:</td>
<td>195/2700</td>
<td>100%/100%</td>
</tr>
<tr>
<td>Up Resources:</td>
<td>191/2652</td>
<td>97%/98%</td>
</tr>
<tr>
<td>Used Resources:</td>
<td>139/1426</td>
<td>71%/52%</td>
</tr>
<tr>
<td>Free Resources:</td>
<td>38/456</td>
<td>19%/16%</td>
</tr>
<tr>
<td><strong>(FreeNodes=38x12)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PartFree Resources:</td>
<td>73/698</td>
<td>37%/25%</td>
</tr>
<tr>
<td>Reserved Resources:</td>
<td>9/468</td>
<td>4%/17%</td>
</tr>
</tbody>
</table>

(AvgFreeCoresPerNode=19)

(Utilization=0%)
<table>
<thead>
<tr>
<th>Polecenia</th>
<th>Dyrekczywy #QCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>qcg-cancel</td>
<td>application</td>
</tr>
<tr>
<td>qcg-clean</td>
<td>argument</td>
</tr>
<tr>
<td>qcg-connect</td>
<td>environment</td>
</tr>
<tr>
<td>qcg-info</td>
<td>error/output</td>
</tr>
<tr>
<td>qcg-list</td>
<td>grant</td>
</tr>
<tr>
<td>qcg-peek</td>
<td>host</td>
</tr>
<tr>
<td>qcg-proxy</td>
<td>memory</td>
</tr>
<tr>
<td>qcg-refetch</td>
<td>nodes / procs</td>
</tr>
<tr>
<td>qcg-resub</td>
<td>note</td>
</tr>
<tr>
<td>qcg-sub</td>
<td>notify / watch-output</td>
</tr>
<tr>
<td>qcg-rcancel</td>
<td>preprocess / postprocess</td>
</tr>
<tr>
<td>qcg-reserve</td>
<td>queue</td>
</tr>
<tr>
<td>qcg-rinfo</td>
<td>reservation</td>
</tr>
<tr>
<td>qcg-rlist</td>
<td>stage-in-dir/file</td>
</tr>
<tr>
<td></td>
<td>stage-out-dir/file</td>
</tr>
<tr>
<td></td>
<td>walltime</td>
</tr>
</tbody>
</table>

```bash
#!/bin/bash

#QCG host=nova
#QCG queue=plgrid
#QCG note=Naphthalene

#QCG output=${JOB_ID}.output
#QCG error=${JOB_ID}.error

#QCG stage-in-file=Naphthalene.gjf
#QCG stage-in-file=gaussian.ntf
#QCG stage-out-dir=->result

#QCG nodes=1:1
#QCG walltime=PT10M

#QCG notify=xmpp:tomasz.piontek@plgrid.pl
#QCG watch-output=20,gaussian.ntf

#QCG application=g09
#QCG argument=Naphthalene.gjf
```

… więcej
XMPP Status Notification

(21:27:14) qcg-notification@plgrid.pl:
### QCG-Broker Event Message ###
Job ID: J1378927597488__6054
Task ID: task
- Status: RUNNING

---
# Namespace: http://schemas.qoscosgrid.org/broker/2012/01/notification-topics
# Topic: QCGBroker/Job/Task

(21:27:19) qcg-notification@plgrid.pl:
### QCG-Broker Event Message ###
Job ID: J1378927597488__6054
Task ID: task
- Timestamp: 2013-09-11T21:27:19.175+02:00
- Status: FINISHED

---
# Namespace: http://schemas.qoscosgrid.org/broker/2012/01/notification-topics
# Topic: QCGBroker/Job/Task
QCG-Icon

- Lightweight intuitive application for Windows, MAC OSX and Linux platforms,
- Provides transparent, unified access to applications installed on Grid resources and available via QosCosGrid services.
- Automatically transfers input/output files between Grid and user-side.
- Gives to the user impression of local work.
QCG Monitoring

Job

<table>
<thead>
<tr>
<th>cluster</th>
<th>cores</th>
<th>master node</th>
<th>node arch</th>
</tr>
</thead>
<tbody>
<tr>
<td>meos.mah.poznan.pl</td>
<td>1</td>
<td>m05</td>
<td>x86_64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>node cores</th>
<th>node memory</th>
<th>node os</th>
<th>node properties</th>
<th>nodes</th>
<th>nodes list</th>
<th>note</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>512000</td>
<td>Linux</td>
<td>sandybridge_2600mhz_kepler_k30</td>
<td>1</td>
<td>m05</td>
<td>gaussian</td>
</tr>
</tbody>
</table>

queue: plgrid, started: 2014-04-03T17:17:63, walltime: 00:00:30

Status
QCG Success Stories

- Infrastructural and research EU and national projects
  – (PLGridPLUS, pMedicine, AirPROM, MAPPER, ACGT, BEinGRID, PL-GRID, QosCosGrid, BREIN)

- Production deployment on PL-Grid resources (the most popular middleware in Poland)

- EGI & PSNC/QCG MoU (2012)

- BCC-UNG&PSNC/QCG MoU (2013)

- Part of UMD 3.2.0

- Cooperation with PRACE
PLGrid Statistics (2013)

- **Core-Hours**
  - January: 600,000
  - February: 700,000
  - March: 800,000
  - April: 1,600,000
  - May: 2,000,000
  - June: 1,800,000
  - July: 1,400,000
  - August: 1,600,000
Integration with EGI

- QCG unit support in GGUS
- SAM Nagios probes & Dashboard alerts
- APEL / GRID-SAFE accounting
- Advertising GLUE2 schema information in BDII (in progress)
- Integration with Virtual Organization Management Support (VOMS)
- Part of the UMD 3.2.0 (September 2013)
- QCG-Icon in AppDB
QCG Deployment

- Installation from packages
  - QCG software repository
    - For SL5/SL6 and Debian (QCG-Comp, QCG-Notif)
  - UMD repository
    - SL5 (SL6 in next update with SHA2 support)
- Installation from sources
- Windows/Linux/MacOS QCG-Icon installer
**QosCosGrid**

The QosCosGrid (QCG) middleware is an integrated system offering advanced job and resource management capabilities to deliver to end-users supercomputer-like performance and structure. By connecting many distributed computing resources together, QCG offers highly efficient mapping, execution and monitoring capabilities for a variety of applications, such as parameter sweep, workflows, MPI or hybrid MPI-OpenMP. Thanks to QosCosGrid, large-scale applications, multi-scale or complex computing models written in Fortran, C, C++ or Java can be automatically distributed over a network of computing resources with guaranteed QoS. The middleware provides also a set of unique features, such as advance reservation and co-allocation of distributed computing resources.

**QCG Middleware**

QosCosGrid provides:

- the most efficient remote access to computational resources in a single cluster or many clusters in Poland and Europe,
- automatic steering of various types of complex computing experiments ranging from multi-parameter sweep studies to cross-cluster executions of parallel applications,
- fully transparent integration with parallel programming and execution environments like OpenMPI and ProActive.

**QosCosGrid 3.0 [16 July]**

A new major release of the QosCosGrid has been just published and includes many improvements in the access to applications, data, and the overhead needed to run applications in a grid environment. Users can take advantage of the enhanced support of QCG Client commands, the ability to read transfer rate or advanced QCG Client commands, will allow users to concentrate even more on their scientific work.

**QCG-icon 1.4.3 [21 June]**

Next version of QCG is now available for download. Since the very first release, many improvements
Contact

• General questions: contact@qoscosgrid.org

• Technical questions: support@lists.qoscosgrid.org

• Tomasz Piontek
  • piontek@man.poznan.pl
• Questions?
• Comments!
• Advices ;-)