application level monitoring for LHC experiments with the experiment dashboard

Benjamin Gaidioz

Hepix fall 2007
the experiment dashboard project

- a monitoring project, showing the activity of VOs on the grid,
  - integration of several informations. For example:
    - grid info: jobs, computing/storage resources, topology,
    - VO info: type of job, application exit code, datasets.

- a project from EGEE/ARDA (CERN),

- a framework for collecting and showing information.

“home page”
outline

the project
  the dashboard framework
  operations

the applications
  job monitoring
  data management monitoring

conclusion
  conclusion & future plans
outline

the project
  the dashboard framework
  operations

the applications
  job monitoring
  data management monitoring

conclusion
  conclusion & future plans
most of the applications are implemented using a common Python framework.

clear distinction between information storage, information queries and rendering:

- support for API, CLI.
- performance: thread safety + session pooling + apache/mod_python,

- even a developer’s guide is available, with examples.
the dashboard framework

▶ most of the applications are implemented using a common Python framework.
  ▶ clear distinction between information storage, information queries and rendering:

  ![Diagram of data flow]

  - services
    - collect info
    - periodic summaries
  - data
    - select
    - group by
  - client
    - apache
  - periodic summaries

▶ support for API, CLI.
▶ performance: thread safety + session pooling + apache/mod_python,
▶ even a developer's guide is available, with examples.
features

- based on HTTP,
- multiple output formats (text/xml and text/csv coming for free),
  
  
  
curl -H 'Accept: text/xml' http://...

- framework for API and CLI: based on HTTP, man pages, build of the command-line tool, common options,
  
  from dashboard.api.production.ProductionQuery import ProductionQuery
  query = ProductionQuery('dashb-atlas-prodsys-test.cern.ch', 80)
  sites = query.errors(error='WRAPLCG_STAGEOUT_LCGCR',
                       grouping='site', grid='LCG')

- framework for permanent services (info collectors, computation of summaries, etc.): monitoring, babysitting, common configuration, etc.
  
  - status exported to our main web server,
  - simple alerts in case of warning status: e-mail or SMS.
releasing and build system

- a project module has a stable/unstable/nightly release,
  - RPMs are currently distributed using apt,
  - we also distribute some external RPMs (javascript toolkits, python packages not available in SLC4),
- the build system is based on the python distutils,
  - automatic building of SLC4 RPMs running at night,
  - also builds and installs the latest documentation (docbook),
  - (doesn’t run the unit tests yet.)
  - possibility to trigger the build at anytime if needed.
operations

- applications using the common framework:
  - same tools maintenance, same log files, similar services, etc.
  - maintenance guides, documentation.
- hosts are SLC4 and are quattor managed,
- about maintenance, some recent developments:
  - dashboard services and Apache now running as “dashbop” UNIX user,
  - definition of an operator quattor role,
  - operators can run a limited set of commands: restart httpd, restart dashboard services,
  - permits to implement some simple and safe maintenance by non-experts.
- but still some legacy applications (no framework, SLC3, etc.).
most of our applications don’t generate much load on the same host:

- but it’s not practical to run several per host,
- applications A and B may require at some point different releases of a common dashboard RPM,
- maintenance of A impacts B (restart httpd, same configuration files, same log files.),
- it’s a limit of our framework (not easy to have two configurations).

dashboard applications are probably good candidates for hardware virtualization,

- it could probably be practical for testing purposes (check the installation on a new host or maintenance procedure, run full functionality tests.)
outline

the project
  the dashboard framework
  operations

the applications
  job monitoring
  data management monitoring

conclusion
  conclusion & future plans
grid job monitoring

- show what the jobs are doing on the Grid from the VO point of view:

  - grid info taken from RGMA, GridPP XML files, LCG BDII,
  - VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
  - installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

► show what the jobs are doing on the Grid from the VO point of view:

- grid info taken from RGMA, GridPP XML files, LCG BDII,
- VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
- installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the **VO point of view**:

![Job Summary Diagram](image)

- grid info taken from RGMA, GridPP XML files, LCG BDII,
- VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
- installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the **VO point of view**:

- grid info taken from RGMA, GridPP XML files, LCG BDII,
- VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
- installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the **VO point of view**:

- grid info taken from RGMA, GridPP XML files, LCG BDII,
- VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
- installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the VO point of view:

- grid info taken from RGMA, GridPP XML files, LCG BDII,
- VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
- installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the **VO point of view**: 

  ![Diagram showing job status](image)

  - grid info taken from RGMA, GridPP XML files, LCG BDII,
  - VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
  - installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the **VO point of view**:

- grid info taken from RGMA, GridPP XML files, LCG BDII,
- VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
- installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the VO point of view:

  - grid info taken from RGMA, GridPP XML files, LCG BDII,
  - VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
  - installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the VO point of view:

![Chart showing job summary](chart.png)

- grid info taken from RGMA, GridPP XML files, LCG BDII,
- VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
- installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the VO point of view:

<table>
<thead>
<tr>
<th>site</th>
<th>current status</th>
<th>grid exit status</th>
<th>application exit status</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sub</td>
<td>Unk</td>
<td>Pend</td>
<td>Rm</td>
</tr>
<tr>
<td>ESCS-ESG2 (Manno, Switzerland)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DESY/Hamburg, Germany)</td>
<td>145</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FZK-LEGO (Aarhus, Germany)</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LRZ (Munich, Germany)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pragueCGI (Prague, Czech Republic)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>total</td>
<td>416</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

- grid info taken from RGMA, GridPP XML files, LCG BDII,
- VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
- installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the VO point of view:
  - grid info taken from RGMA, GridPP XML files, LCG BDII,
  - VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
  - installed for all four main experiments, plus the “vlemed” VO.
grid job monitoring

- show what the jobs are doing on the Grid from the **VO point of view**:
  - grid info taken from RGMA, GridPP XML files, LCG BDII,
  - VO info: sent by jobs (using Monalisa, or read bookkeeping DB,
  - installed for all four main experiments, plus the "vlemed" VO.
grid job monitoring: user tasks

- view of the same data from a user running analysis:

- show “my tasks”,
- progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

▶ view of the same data from a user running analysis:

▶ show “my tasks”,
▶ progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

- view of the same data from a user running analysis:

  - show “my tasks”,
  - progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

▶ view of the same data from a user running analysis:

▶ show “my tasks”,
▶ progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

- view of the same data from a user running analysis:

- show “my tasks”,
- progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

▶ view of the same data from a user running analysis:

▶ show “my tasks",
▶ progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

- view of the same data from a user running analysis:

<table>
<thead>
<tr>
<th>EventRange</th>
<th>Appl Exit Code</th>
<th>Grid End Status</th>
<th>Site</th>
<th>Subm</th>
</tr>
</thead>
<tbody>
<tr>
<td>90001255g7XPHQ04TSG3aBesaQ</td>
<td>6</td>
<td>DONE</td>
<td>CERN/PROD</td>
<td></td>
</tr>
<tr>
<td>9010/wKvKXvhsIjUenSvIvKvI69w</td>
<td>6</td>
<td>6</td>
<td>Required application version is not found at the site?</td>
<td></td>
</tr>
</tbody>
</table>

- show “my tasks”,
- progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

- view of the same data from a user running analysis:

<table>
<thead>
<tr>
<th>TaskName</th>
<th>Num of Jobs</th>
<th>Pending</th>
<th>Running</th>
<th>Successful</th>
<th>Failed</th>
<th>Terminated with app1 unknowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_name_1</td>
<td>50</td>
<td>10</td>
<td>30</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>user_name_2</td>
<td>35</td>
<td>15</td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>user_name_3</td>
<td>15</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>user_name_4</td>
<td>25</td>
<td>10</td>
<td>15</td>
<td>5</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

- show “my tasks”,
- progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

- view of the same data from a user running analysis:

- show “my tasks”,
- progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

▶ view of the same data from a user running analysis:

▶ show “my tasks”,
▶ progress of each, investigate the reasons of the failures.
grid job monitoring: user tasks

▶ view of the same data from a user running analysis:

▶ show “my tasks”,
▶ progress of each, investigate the reasons of the failures.
grid job monitoring: summaries

- permits to show precompute summaries (faster queries),

- for application exit codes, VO activities, etc.

- same information as before.
grid job monitoring: summaries

- permits to show precompute summaries (faster queries),

- for application exit codes, VO activities, etc.
- same information as before.
grid site reliability

- uses RGMA and IC-XML info for computing efficiency of sites,
- aware of *middleware resubmissions* to sites A, B and C,
- computation of daily rates, possibility to dig in the information.
grid site reliability

- uses RGMA and IC-XML info for computing efficiency of sites,
- aware of *middleware resubmissions* to sites A, B and C,
- computation of daily rates, possibility to dig in the information.
grid site reliability

- uses RGMA and IC-XML info for computing efficiency of sites,
- aware of *middleware resubmissions* to sites A, B and C,
- computation of daily rates, possibility to dig in the information.
grid job monitoring: ATLAS production

- monitoring offloaded to the dashboard,
- target: “shifters”,
- under “review” by master-shifters.
grid job monitoring: ATLAS production

- monitoring offloaded to the dashboard,
- target: “shifters”,

- under “review” by master-shifters.
grid job monitoring: ATLAS production

- monitoring offloaded to the dashboard,
- target: “shifters”,
- under “review” by master-shifters.
grid job monitoring: ATLAS production

- monitoring offloaded to the dashboard,
- target: “shifters”,

- under “review” by master-shifters.
grid job monitoring: ATLAS production

- monitoring offloaded to the dashboard,
- target: “shifters”,

- under “review” by master-shifters.
grid job monitoring: ATLAS production

- monitoring offloaded to the dashboard,
- target: “shifters”,

- under “review” by master-shifters.
grid job monitoring: ATLAS production

- monitoring offloaded to the dashboard,
- target: “shifters”,

- under “review” by master-shifters.
grid job monitoring: ATLAS production

- monitoring offloaded to the dashboard,
- target: “shifters”,
- under “review” by master-shifters.
grid job monitoring: CMS production

- close collaboration between CMS and the dashboard,
- the CMS production agents publish their status in the dashboard (HTTP POST) in real time,
- dashboard services are computing periodic statistics,
- the view is not implemented with the dashboard:
  - info is retrieved using the dashboard interface,
  - CMS will make their own interface on top.
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

- topology and names are specific to ATLAS, ATLAS datasets,
- info is published directly by the DDM servers, using a specific API (a dashboard API based on HTTP).
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

<table>
<thead>
<tr>
<th>Cloud</th>
<th>Efficiency</th>
<th>Throughput</th>
<th>Files Done</th>
<th>Datasets Done</th>
<th>DQ</th>
<th>Grid</th>
<th>Transfer</th>
<th>Local</th>
<th>Remote</th>
<th>Central</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASGC</td>
<td>21%</td>
<td>3 MB/s</td>
<td>296</td>
<td>0</td>
<td></td>
<td></td>
<td>1195</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNL</td>
<td>19%</td>
<td>12 MB/s</td>
<td>662</td>
<td>13</td>
<td></td>
<td></td>
<td>2788</td>
<td>52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CERN</td>
<td>14%</td>
<td>3 MB/s</td>
<td>289</td>
<td>0</td>
<td></td>
<td></td>
<td>2489</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNAF</td>
<td>65%</td>
<td>2 MB/s</td>
<td>167</td>
<td>4</td>
<td></td>
<td></td>
<td>17</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIC</td>
<td>10%</td>
<td>1 MB/s</td>
<td>101</td>
<td>1</td>
<td></td>
<td></td>
<td>306</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LYON</td>
<td>77%</td>
<td>10 MB/s</td>
<td>1139</td>
<td>2</td>
<td></td>
<td></td>
<td>336</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NCDF</td>
<td>48%</td>
<td>10 MB/s</td>
<td>2186</td>
<td>2</td>
<td></td>
<td></td>
<td>2333</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC</td>
<td>50%</td>
<td>5 MB/s</td>
<td>1021</td>
<td>0</td>
<td></td>
<td></td>
<td>2417</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAL</td>
<td>38%</td>
<td>4 MB/s</td>
<td>345</td>
<td>1</td>
<td></td>
<td></td>
<td>821</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SARA</td>
<td>66%</td>
<td>5 MB/s</td>
<td>530</td>
<td>1</td>
<td></td>
<td></td>
<td>274</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIUMF</td>
<td>64%</td>
<td>1 MB/s</td>
<td>224</td>
<td>9</td>
<td></td>
<td></td>
<td>126</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- topology and names are specific to ATLAS, ATLAS datasets,
- info is published directly by the DDM servers, using a specific API (a dashboard API based on HTTP).
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Source</th>
<th>Target</th>
<th>Status</th>
<th>Size</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>X.txt</td>
<td>10MB</td>
<td>X server</td>
<td>Y server</td>
<td>100%</td>
<td>0KB</td>
<td>0s</td>
</tr>
</tbody>
</table>

- topology and names are specific to ATLAS, ATLAS datasets,
- info is published directly by the DDM servers, using a specific API (a dashboard API based on HTTP).
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

<table>
<thead>
<tr>
<th>Site</th>
<th>Status</th>
<th>Size</th>
<th>Free</th>
<th>OK</th>
<th>File Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>FZ5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FZ2DISK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FZ2TAPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- topology and names are specific to ATLAS, ATLAS datasets,
- info is published *directly* by the DDM servers, using a specific API (a dashboard API based on HTTP).
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

<table>
<thead>
<tr>
<th>Site</th>
<th>Status</th>
<th>TB HBs</th>
<th>KBs</th>
<th>GBytes</th>
<th>OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>F25</td>
<td>10%</td>
<td>1 TB14</td>
<td>0</td>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>F25DISK</td>
<td>0%</td>
<td>0 TB14</td>
<td>0</td>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>F25TAP</td>
<td>0%</td>
<td>0 TB14</td>
<td>0</td>
<td>0</td>
<td>OK</td>
</tr>
</tbody>
</table>

Click on the site name to go to the site page.

- topology and names are specific to ATLAS, ATLAS datasets,
- info is published *directly* by the DDM servers, using a specific API (a dashboard API based on HTTP).
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

- topology and names are specific to ATLAS, ATLAS datasets,
- info is published directly by the DDM servers, using a specific API (a dashboard API based on HTTP).
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

- topology and names are specific to ATLAS, ATLAS datasets,
- info is published directly by the DDM servers, using a specific API (a dashboard API based on HTTP).
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

- topology and names are specific to ATLAS, ATLAS datasets,
- info is published *directly* by the DDM servers, using a specific API (a dashboard API based on HTTP).
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

- topology and names are specific to ATLAS, ATLAS datasets,
- info is published directly by the DDM servers, using a specific API (a dashboard API based on HTTP).
grid data transfers monitoring

- activity conducted with ATLAS DDM (distributed data management),
- show the status and performance of the DDM system:

    ▶ topology and names are specific to ATLAS, ATLAS datasets,
    ▶ info is published *directly* by the DDM servers, using a specific API (a dashboard API based on HTTP).
outlines

the project
  the dashboard framework
  operations

the applications
  job monitoring
  data management monitoring

conclusion
  conclusion & future plans
conclusion & future plans

- framework and operations:
  - installation, maintenance procedures,
  - systematic testing (unit tests).

- applications:
  - grid jobs: more info sources and support for pilot jobs,
  - integrate more our applications with themselves (!),
  - develop the integration with external tools (python API),
  - alert systems.