The HappyFace Project – A Meta Monitoring Tool
Monitoring Sites, Services, Jobs

Armin Burgmeier, for the HappyFace Developers
Outline

- Introduction
  - Current Monitoring Situation
  - How Meta Monitoring improves the Situation
- The HappyFace Project
  - Overview and Features
  - Architecture
  - Installation
  - Recent and ongoing development
- Live Demonstration
- Summary and Outlook
What Needs to be Monitored?

A Grid site consists of many critical components each of which needs to be monitored to make sure the center performs well.

- Grid Infrastructure (Software failures)
- Batch System (Inefficient jobs, failing jobs)
- Storage System (Staging failures, Orphaned files, Replicas, Pools utilization)
- Data Transfer (Slow transfers, failed transfers)

All Information is available but distributed
So what's the Problem?

Overall Monitoring is inconvenient at best:

- Manage many browser tabs or windows
- Each website has its own settings (time ranges, Grid site, ...) that needs to be set up
- Long loading times, can be more than 30 seconds
- Difficult to identify correlations between failures at different services or sites
- Hard to get a quick overview of a site's status, especially for non-experts

Solution: Important information available at a single place
The solution: The HappyFace Project Ver. 2

HappyFace features:

- Collect important Monitoring information from multiple services
  - No generation of new data, only process available data
  - Runs in a short time interval, $O(15 \text{ min})$
- Present the current status of the services in a compact way
- Easy rating system to quickly see whether a service is good or critical
- Detailed information can then be obtained via the source website
- Modular Layout: Individual tests or plots can easily be enabled or disabled; new ones can be developed without interfering with the rest of the system
- History navigation: It is possible to go back in time and check each module's status and other output, including plots
- Generate time-dependent plots of recorded quantities
The HappyFace Project Architecture

HappyFace Core:
- Read global configuration and execute all enabled modules

HappyFace Modules:
- Read module configuration
- Download monitoring data (XML, Plots, ...)
- Run the test and compute module rating
- Write results into database and plots into filesystem
- Write PHP code for web output
Overview of the HappyFace website

Date and Time of current view
History navigation
Go to most recent version

Category Selection and Status
Module Title, Status and Execution Time
Module Output

Category Navigation and Module Status

Date: 25. Aug 2010
Time: 16:18
History Plots

The history of each module's status and, depending on the module, other numerical variables can be plotted.

Module: jobs_statistics

- Start: 2010-08-23 17:23
- Stop: now
- Interval: 48 hours

Variable: total +2 more

Legend: ○ bottom ○ inside ○ right

Variable(s): total, running, ratio10

Here: Number of Jobs at GridKa

- All jobs
- Running jobs
- Jobs with walltime ratio below 10%
HappyFace Installation

Setting up an own HappyFace instance is easy:

- Make sure a **webserver**, **Python**, **PHP** and **SQLite** are installed
- Check out the **HappyFace code**:
  
  ```
  svn co https://ekptrac.physik.uni-karlsruhe.de/public/HappyFace/trunk myHFinstance
  ```
- Add it to the **crontab** so that it runs every 15 minutes:
  
  ```
  */15 * * * * cd /path/to/myHFinstance/HappyFace && ./run.py >/dev/null 2>&1
  ```
- Configure **categories** in **local/cfg/run.local**
- Configure **modules** in **local/cfg/myModule.local**
- Example modules and configuration files available
- Full **documentation** available at
  
  [https://ekptrac.physik.uni-karlsruhe.de/trac/HappyFace/wiki/Version_2](https://ekptrac.physik.uni-karlsruhe.de/trac/HappyFace/wiki/Version_2)
Example modules: dCache Data Transfers module

Watch file transfers from dCache pools

- Warns if transfers take much time
- Warns if transfers are slow
- Indicator for inefficient jobs
- Correlation with problematic pools or host systems is revealed
Example modules: dCache Data Management module

Compare files known by dCache with the CMS DBS database

- For each file known by dCache check which dataset it belongs to
- Serves as a consistency cross check
- Datasets fully on disk are interesting to end-user analyses

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>Bare total</th>
<th>Bare on disk</th>
<th>Total on disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>/MinimumBias/Commission10-v4/RAW</td>
<td>17870 files 49088.3 GB</td>
<td>17647 files 49031.6 GB</td>
<td>21938 files 61256.6 GB</td>
</tr>
<tr>
<td>/MinimumBias/Commission10-PromptReco-v9/RECO</td>
<td>18934 files 41341.1 GB</td>
<td>18596 files 41290.5 GB</td>
<td>21888 files 54982.1 GB</td>
</tr>
<tr>
<td>/Calo/Commission10DB-v2/RAW</td>
<td>10771 files 58263.4 GB</td>
<td>6446 files 31674.9 GB</td>
<td>6474 files 31738.2 GB</td>
</tr>
<tr>
<td>Unassigned</td>
<td>18763 files 80080.8 GB</td>
<td>18518 files 80640.9 GB</td>
<td>18935 files 80461.7 GB</td>
</tr>
</tbody>
</table>

- Size of dataset in #files and GB
- Part of dataset present on disk
- Total space occupied on disk, including replicas
- Orphaned files (available in dCache but not in any dataset)
Example modules: Jobs Statistics module

Monitors Jobs running at a Grid site

- Standalone producer generates common XML file by querying the batch system; producers are already available for PBS, Condor.
- HappyFace module reads the XML, then visualizes the information and stores it into its database
- Decoupling of data generation and data visualization
- Warn if there is a significant number of inefficient jobs in a group
Example modules: RSS module

Allows to display RSS feeds as a non-rated HappyFace module

- Inform shifters about general news or known problems
- Useful for external monitoring sources and service portals which use RSS
  - For example, show open tickets of relevant services

---

**Current Shifter Information**
Tue, 27 Jul 2010, 08:00 - Show module information

---

**#300: Gridka open ggus tickets**
Posted on Mon, 26 Jul 2010 20:39:43 +0200

We currently have two open issues:

1. More and more failing CMS SAMTests, seems to be an issue with BDII reporting wrong information at random frequency:  [https://gus.fzk.de/ws/ticket_info.php?ticket=60420](https://gus.fzk.de/ws/ticket_info.php?ticket=60420)


[2] seems to be solved. [1] seems to have improved a lot, watching until tomorrow, as SAM test have long latency.
Example modules: Summary module

HappyFace can export aggregated information as XML

- The Summary module makes use of it by showing a quick overview of many HappyFace instances: Meta² Monitoring

- Ideally this allows a small shift crew to supervise a large number of sites
More modules

There are many more modules available in HappyFace:

- **dCache pools** information:
  - Rating on various variables, for example free space/total space
  - Verify load balancing
- dCache dataset restore monitor (staging)
- SAM Tests:
  - Test that software is available and well-behaving on worker nodes
  - OPS and Experiment specific tests
- Site Readiness statistics
- PhEDEx transfer errors and statistics
- Firefox Plugin
In development: T2 User Space Monitoring

Information about used disk space per user

- Security: Authentication is realized via CA
- User mode: Only user information available
- Admin mode: Detailed information about each user
- Rating based on number of users over quota

### T2 User Space Monitoring

**Thu, 11 Mar 2010, 11:59**

<table>
<thead>
<tr>
<th>used disk space</th>
<th>22.9 TB</th>
</tr>
</thead>
<tbody>
<tr>
<td>sites: T2_DE_RWTH, T2_DE_DESY</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>users exceeding quota</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>quota of 2.0 TB per user, individual limits for power users</td>
<td></td>
</tr>
<tr>
<td>unmatched directories</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>T2_DE_RWTH</th>
<th>T2_DE_DESY</th>
<th>Total Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power User1</td>
<td>10318 GB</td>
<td>318 GB</td>
<td>10636 GB</td>
</tr>
<tr>
<td>Power User2</td>
<td>104 GB</td>
<td>9818 GB</td>
<td>9922 GB</td>
</tr>
<tr>
<td>DCMS User</td>
<td>—</td>
<td>681 GB</td>
<td>681 GB</td>
</tr>
<tr>
<td>CMS User</td>
<td>307 GB</td>
<td>—</td>
<td>307 GB</td>
</tr>
</tbody>
</table>
CMS to use HappyFace for production job monitoring of CMS jobs at all T1s

- Each T1 sets up an XML producer for their respective batch system
- HappyFace aggregates the information and visualizes it all at one place
- One category for each site
- T2 jobs can also be shown in another category
## Summary

<table>
<thead>
<tr>
<th>Monitoring</th>
<th>Meta Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many browser windows/tabs</td>
<td>All information available on a single website</td>
</tr>
<tr>
<td>Each monitoring site requires individual</td>
<td>Individual site settings are configured once in HF config</td>
</tr>
<tr>
<td>settings</td>
<td></td>
</tr>
<tr>
<td>Long loading times</td>
<td>Single website, no complex database backend</td>
</tr>
<tr>
<td>Difficulty to find correlations</td>
<td>Easy due to having all information available immediately, plus history</td>
</tr>
<tr>
<td>Hard to get quick overview</td>
<td>Just see whether there are red arrows or not; each module has instructions</td>
</tr>
</tbody>
</table>

Modular design allows for easy **Development** and **Deployment** of new modules

- Contributions by Aachen, Göttingen, Hamburg, Karlsruhe
- HappyFace deployed at **many sites**
- **Active development** is ongoing (USM, Mail notification)
Thank you!

Thanks to
all HappyFace developers:

- Aachen: M. Edelhoff, P. Sauerland, O. Tsigenov
- Göttingen: C. Ay, S. Birkholz, J. Meyer, A. Quadt
- Hamburg: F. Nowak, P. Schleper, H. Stadie
- Karlsruhe: V. Büge, A. Burgmeier, V. Mauch, G. Quast, N. Ratnikova, A. Scheurer, M. Zvada

and the Helmholtz Alliance “Physics at the Terascale”

https://ekptrak.physik.uni-karlsruhe.de/trac/HappyFace/wiki/Version_2

Questions?