HappyFace as a generic monitoring tool for HEP experiments

Gen Kawamura, Erekle Magradze, Haykuhi Musheghyan, Arnulf Quadt, Gerhard Rzehorz
II. Physikalisches Institut, Georg-August-Universität Göttingen, Friedrich-Hund-Platz 1 37077
Göttingen Germany
E-mail: haykuhi.musheghyan@uni-goettingen.de

Abstract. The importance of monitoring on HEP grid computing systems is growing due to a significant increase in their complexity. Computer scientists and administrators have been studying and building effective ways to gather information on and clarify a status of each local grid infrastructure. The HappyFace project aims at making the above-mentioned workflow possible. It aggregates, processes and stores the information and the status of different HEP monitoring resources into the common database of HappyFace. The system displays the information and the status through a single interface. However, this model of HappyFace relied on the monitoring resources which are always under development in the HEP experiments. Consequently, HappyFace needed to have direct access methods to the grid application and grid service layers in the different HEP grid systems. To cope with this issue, we use a reliable HEP software repository, the CernVM File System. We propose a new implementation and an architecture of HappyFace, the so-called grid-enabled HappyFace. It allows its basic framework to connect directly to the grid user applications and the grid collective services, without involving the monitoring resources in the HEP grid systems. This approach gives HappyFace several advantages: Portability, to provide an independent and generic monitoring system among the HEP grid systems. Functionality, to allow users to perform various diagnostic tools in the individual HEP grid systems and grid sites. Flexibility, to make HappyFace beneficial and open for the various distributed grid computing environments. Different grid-enabled modules, to connect to the Ganga job monitoring system and to check the performance of grid transfers among the grid sites, have been implemented. The new HappyFace system has been successfully integrated and now it displays the information and the status of both the monitoring resources and the direct access to the grid user applications and the grid collective services.

1. Introduction
The LHC experiments rely on distributed computing resources. Worldwide LHC Computing Grid (WLCG) is a global solution and based on grid technologies/middleware. It has a tiered structure (Tier0 (CERN), 11 Tiers, 140 Tiers). The LHC computing system must manage approximately 15PB of new data for each year. Scientists or users from different facilities and countries should have access to the data in order to run their own analysis. There are certain requirements of computing power and storage space for the WLCG computing model. Each computing center provides integrated storage space and computing power to different users. Corresponding middleware components handle the submission of data and further distribution of them in different middleware sites. The role of monitoring systems is crucial and essential.
It is impossible to have only one monitoring tool for huge collaborations using such large-scale infrastructure. For this reason many monitoring tools have been designed and implemented. Each monitoring system uses its own architecture of presenting data values and is designed for monitoring the definite part of the experiment. Although the majority of monitoring systems can investigate the sites and their troubles in the HEP experiments within WLCG, the site administrators have to check different web sources and even operate grid commands manually in order to get full information about the sites.

2. The concept of monitoring

The preferable components that monitoring tool should have are the followings:

- Single access point: the final output should have a single website, which shows all requested information from existing sources.
- Fast accessibility: provides fast access to the requested information from existing sources.
- Modular structure: the modular structure provides easy/visible output.
- Up-to-date monitoring information: the tool should provide fresh or up-to-date information.
- History functionality: should be able to show old data.
- Comfortable usage: the tool should be easy in terms of usage.

The users of such system are site administrators who need to check the status of the site and also grid users who need to submit jobs to the corresponding site. Grid users can get the overview of the system to recognize possible problems of the particular site in case of failure of their jobs.

3. The HappyFace Project

HappyFace [1] is a meta-monitoring tool designed for the aggregation, processing, storing of different monitoring resources of WLCG [2] [3]. HappyFace is structured in separate modules, which are embedded in a core framework. Each module is responsible for the aggregation and processing of data from specific monitoring sources. The HappyFace web page publishes the module output and provides a single point of access to the monitoring information. The HappyFace core framework, the individual modules, and the HappyFace web page are individually customisable to site-specific requirements. The workflow of HappyFace is shown in Figure 1.

![Figure 1. HappyFace workflow.](image)

Two Python scripts need to be executed. The first script acquire.py reads all HappyFace configuration files and executes the individual modules, which aggregate, process, and store monitoring data. Subsequently, the script render.py accesses the stored data and calls a specific function of each module in order to generate an output for the HappyFace web page.
also takes care of rendering the web page and makes it accessible via a web server. The Python script acquire.py is periodically executed every 15 minutes, e.g. via a cron job on Linux/Unix operating systems. The HappyFace version 3 is written in Python and uses SQLite as a database. The web output is in HTML and as template uses HTML "Mako" [4].

4. Grid-enabled HappyFace
The initial model of HappyFace relied on monitoring resources which are always under development (Figure 2). This made the system unstable.

Accordingly, HappyFace needed to have direct access methods to the grid applications and grid service layers. To cope with this issue, we use a reliable HEP software repository, the CernVM File System (CVMFS) [5], [6], supplying grid environments and available HEP tools. We propose a new implementation and an architecture of HappyFace, the so-called grid-enabled HappyFace. It allows its basic framework to connect directly to the grid user applications and the grid collective services, without involving the monitoring resources in the HEP grid systems (Figure 3).

This approach gives HappyFace several advantages:

- Portability: to provide an independent and generic monitoring system among the HEP grid systems.
- Functionality: to allow users to perform various diagnostic tools in the individual HEP grid systems and grid sites.
- Flexibility: to make HappyFace beneficial and open for the various distributed grid computing environments.

These factors need to be taken into consideration when implementing new extensions in HappyFace to accept new monitoring resources described above. We designed Grid-enabled HappyFace as the extension of the HappyFace system. In order to have access to the grid system there are certain requirements.
• Proper X.509 certificate
• Integration with Grid environments and available HEP tools
• New source code implementation as an extension

Consequently, the structure of classes is the following:

• Grid subprocess class is inherited from Python subprocess class and has added two more
  class variables - GridEnv and CvmfsEnv defined in EnvReader as described below. It also
  exports the X.509 user certificate and the user key and enables the "CVMFS" environment,
  when executing the grid process.
• GridCertificate class checks if the certificate is valid or not. In case it is not valid then it
  generates a new proxy certificate.
• EnvReader class is used for reading the grid environments from HappyFace configuration
  file.
• Different access method classes were implemented, such as the Transfers class which inherits
  from GridSubprocess class and by that has access to the grid system by itself.

The schematic view is shown in Figure 4.

![Figure 4. Grid-enabled HappyFace workflow model.](image)

The new Grid-enabled HappyFace extension was successfully integrated and it provides direct
access to the grid infrastructure.

5. Transfer module
One of the Grid-enabled modules is the so-called Transfer module which checks the performance
of grid transfers among the grid sites where it has been implemented. It shows the status
of transfers between space tokens among the WLCG sites (see Figure 5). This allows system
administrators to know whether the transfers between space tokens succeeded or failed. It
randomly generates files and copies from one space token to any others. The transfer checks
initialized in two different ways, by using UberFTP or SRM (Storage Resource Manager)
commands.

UberFTP is a GridFTP-enabled client that supports both interactive use and FTP commands
on the UberFTP command line to transfer files between two grid sites. UberFTP is intended for
use with computers that have a GridFTP server installed. It supports GSI authentication,
parallel data channels and striping. SRM is a grid middleware protocol and provides dynamic space allocation, file management on shared storage resources on the Grid. Different implementations e.g. dCache, StoRM for underlying storage systems are based on the same SRM specification. The green color indicates successful transfers (see Figure 6). The red color with a tooltip icon indicates failures. By moving the cursor on the icon the tooltip window will appear as shown below.

6. Future work
The new Grid-enabled HappyFace and several new modules have been deployed and tested to operate the distributed grid computing environments. With the new monitoring strategy - which was described in section 4 - it will be possible to adopt HappyFace further for the ATLAS
operational model, which can include, e.g. site stress test builder with ProdSys-2 [7], site specific grid job executor [8] and local dataset monitor in ATLAS DDM with Rucio [9], that helps with the LHC Run 2 demands.

7. Conclusion
This paper summarizes the recent activity of the HappyFace project and its goal. Based on experience and developments during LHC Run 1 and LS1, we have made many changes in the HappyFace system. Many different sources and their different presentation interfere with an efficient administration. The HappyFace monitoring system improves the situation by providing a summary of all important site specific information available on the single website with the possibility to check the centers’ status at a previous point in time. At present, the HappyFace system aggregates, processes and stores information of both the monitoring resources and the direct access to the grid user applications and the grid collective services in the WLCG computing systems. Modularity allows to provide new modules easily and share them among different sites. Currently, several German grid sites use the HappyFace system for the site monitoring

- University of Göttingen
- Karlsruhe Institute of Technology
- RWTH Aachen University

8. Acknowledgments
Many thanks to our work group members and to Eric Buschmann, Lino Gerlach, Fabian Kukuck, Max Robinson, Christian Wehrberger. The HappyFace project is a joined collaboration between Karlsruhe Institute of Technology (KIT) and Göttingen University. Thanks to Stefan Letzelter, Marcus Schmitt, Fred Stober, Günter Quast, Gregor Vollmer at KIT.

9. References
[1] Mauch, Viktor, et al., 2011, "The HappyFace Project" J. Phys.: Conf. Ser. 331 082011
[2] Jung, Christopher and Petzold, Andreas and Zvada, Marian, 2012, "Experiment Representation at the WLCG Tier-1 Center GridKa" EGI Community Forum / EMI Second Technical Conference (EGICF12-EMITC2)
[3] HappyFace https://ekptrac.physik.uni-karlsruhe.de/trac/HappyFace
[4] Mako Templates for Python http://www.makotemplates.org/
[5] Blomer, J. et al., 2012, "Status and future perspectives of CernVM-FS." J. Phys.: Conf. Ser. 396 052013
[6] Blomer, Jakob, Aguado-Snchez, Carlos, Buncic, Predrag, Harutyunyan, Artem, 2011, "Distributing LHC application software and conditions databases using the CernVM file system" J. Phys.: Conf. Ser. 331 042003
[7] Campana, S and ATLAS Collaboration and others, 2014, "Evolution of the ATLAS distributed computing system during the LHC long shutdown" J. Phys.: Conf. Ser. 513 032016
[8] Maeno, Tadashi, et al. 2011, "Overview of ATLAS PanDA workload management." J. Phys.: Conf. Ser. 331 072024
[9] Megino, Fernando, Barreiro, Harald, et al., 2012, "DDM Site Services: A solution for global replication of HEP data" EGI Community Forum / EMI Second Technical Conference