Integrating configuration workflows with project management system

Dimitri Nilsen, Pavel Weber
Karlsruhe Institute of Technology Steinbuch Centre for Computing
Hermann-von-Helmholtz-Platz 1, D-76344 Eggenstein-Leopoldshafen
E-mail: dimitri.nilsen@kit.edu pavel.weber@kit.edu

Abstract. The complexity of the heterogeneous computing resources, services and recurring infrastructure changes at the GridKa WLCG Tier-1 computing center require a structured approach to configuration management and optimization of interplay between functional components of the whole system. A set of tools deployed at GridKa, including Puppet, Redmine, Foreman, SVN and Icinga, provides the administrative environment giving the possibility to define and develop configuration workflows, reduce the administrative effort and improve sustainable operation of the whole computing center. In this presentation we discuss the developed configuration scenarios implemented at GridKa, which we use for host installation, service deployment, change management procedures, service retirement etc. The integration of Puppet with a project management tool like Redmine provides us with the opportunity to track problem issues, organize tasks and automate these workflows. The interaction between Puppet and Redmine results in automatic updates of the issues related to the executed workflow performed by different system components. The extensive configuration workflows require collaboration and interaction between different departments like network, security, production etc. at GridKa. Redmine plugins developed at GridKa and integrated in its administrative environment provide an effective way of collaboration within the GridKa team. We present the structural overview of the software components, their connections, communication protocols and show a few working examples of the workflows and their automation.

1. Introduction
GridKa serves as Tier-1 computing centre for the four main LHC experiments and also provides its computing power and storage for other non-LHC HEP and astroparticle physics experiments. The effective administration of diverse computing resources and services at GridKa depends on many components of the administrative environment. The major components are configuration management system (CMS) based on Puppet [1] and the Redmine project management tool [2]. Currently the whole middleware system installation, configuration and recurring updates at GridKa is managed by Puppet. Redmine is the primary project management tool used by the GridKa team for collaboration, issue tracking and communication tasks. We investigate further automation and optimization of the administration of GridKa resources by integrating configuration and project management systems. In this paper we describe the main building blocks needed for this integration and present first results and status of such integration.
2. Middleware Configuration Management at GridKa

In the following subsections, the basic middleware infrastructure and middleware configuration management are discussed.

2.1. Middleware at GridKa centre

The heterogeneous middleware services deployed at GridKa originate from several middleware providers. gLite [3] is the major middleware provider at GridKa and in World-wide LHC Computing Grid (WLCG). The installation comprises Computing Resource Execution And Management (CREAM), Workload Management System (WMS), services such as information system and user interfaces (UI). Typically, the middleware servers at GridKa contain one gLite service per host.

Globus [4] and UNICORE [5] are other two middleware flavours deployed at GridKa. The services of these flavours are used to access the Karlsruhe Institute of Technology (KIT) clusters by different scientific communities in Germany. The set of middleware products mentioned above, with the exception of Globus, is provided by European Middleware Initiative (EMI) [6]. More detailed description of the middleware infrastructure at GridKa is given in [7].

2.2. Middleware Configuration Management

The effective middleware CMS plays a crucial role in reliable administration of more than 100 middleware servers at GridKa. The main objectives for the middleware configuration system at GridKa is to provide efficient and reliable change management of the middleware servers including fast deployment procedure, fast and easy update and roll back mechanisms, efficient monitoring and problem tracking systems.

The Puppet CMS chosen at GridKa and applied to middleware servers fulfils the requirements mentioned above. Its main functional blocks are:

- subversion [8] repository;
- puppet master server;
- external data source;
- middleware stack;
- foreman dashboard [9].

As shown in Figure 1, all configuration data are stored in Subversion (SVN) repository. There are two middleware configuration versions: a development release and a production release. The development release is used explicitly for development and testing purposes. The changes implemented in the development release after full tests are propagated to the production release. The production release is the configuration version which holds the actual configuration of the middleware hosts. The Puppet master server obtains the configuration from Subversion (SVN) and applies it during the configuration runs to middleware hosts. The real time status information, results of configuration runs for each middleware host are shown in the Foreman dashboard.

Although the implemented configuration framework based on the Puppet tool together with the SVN release management showed its reliability and robustness to changes in the middleware system, it was decided to work on further improvements of such system, on the development of configuration workflows and on the integration of the configuration framework with the project management system.
To ensure high availability of GridKa services, the effective change management at hosts need to be done in a proper way to the best suitable time, depending on cluster usage and man power. Thus, such tasks and workflows need to be well described and standardized and defined in advance. One forgotten step could endanger the whole production of the computing center. As most of these tasks could be converted to machine code in form of scripts or lists of methods it is possible to organize them in configuration workflows and create templates for the future. The collection of single “tasks” combined in complex “workflows” are defined as templates in the project management tools. In the Figures 4 and 2 the examples of single task workflow and multiple tasks workflow respectively are presented. Multiple tasks workflow is constructed a set of sequentially executed single tasks. Figure 3 shows the particular workflow for automated host certificates update implemented and running at GridKa.

4. Tasks
Each unique action in configuration management, which can be applied to hosts or host groups is defined as a “task”. The tasks are implemented as methods and monitoring calls. Written in the Puppet language in form of parameterized classes, tasks can be included as puppet modules to the configuration environment. Multiple developers of the team can access the tasks stored in a central SVN repository. Each task has an unique identifier or “TaskId” which is a string value, so that puppet master, Redmine or other software components are able to checkout the code from repository by “TaskId”. The standardized interface with the main Puppet classes (as shown in Figure 5) provides an abstract description of functions that tasks must implement to be recognized by the Redmine plugin API and the
5. Redmine Integration
Redmine is a web-based project management tool, which is used by GridKa team to track the problem issues for multiple projects. Due to the flexible modular Redmine framework it is possible to extend the main Redmine functionality by adding plugins. The TasksList plugin is being developed at GridKa and provides the functionality to construct the configuration workflows described in Section 3 based on the tasks definitions from tasks repository as shown in Figure 6. The plugin acts as the interface to the tasks repository and allows to create new and customize the existing configuration workflows.

6. Configuration Workflow Manager
The Configuration Workflow Manager (CWM) is a main core engine, which connects Puppet with Redmine TasksList API and manages configuration workflows. CWM is built on top of puppet master in form of ruby scripts. On one hand, CWM connects to Redmine TasksList API and performs search for open tasks by host, host groups or other parameters. On the other hand, it collects task modules from central repository by “TaskId” and pulls them to puppet master. On
the puppet master tasks are located in a separate sandbox directory to provide better overview and accessibility. The execution of each task is repeated until monitoring functions report a successful result or timeout. After tasks have finished CWM removes them from a sandbox and host manifests and mark tasks as “done” in Redmine project management tool. Figure 7 shows all functional components involved in the configuration management workflow execution and their collaboration.

7. Conclusions
The configuration management system based on Puppet tool performs full control of the middleware resources at GridKa. In this paper, the integration of this configuration management system and developed configuration workflows with a project management system are discussed. Puppet tool provides very flexible and efficient language to perform the change and configuration management in the system, but the diversity of the middleware resources and services results in emergence of complex configuration environment. Thus, even small configuration change on the middleware hosts requires to perform several changes in the configuration environment provided by Puppet and assurance that these changes do not introduce any malfunction in the system. The integration of configuration management with project management system provides a new abstraction layer for construction of configuration workflows and helps to efficiently perform the change management. The proposed approach significantly simplifies the administration of the heterogeneous resources. Currently, the set of the configuration workflows is developed and implemented at GridKa, the TasksList plugin and Configuration Workflow Manager are being developed and tested.

References
[1] Puppet webpage, URL http://puppetlabs.com/
[2] Redmine webpage, URL http://www.redmine.org/
[3] gLite webpage, URL http://glite.web.cern.ch/glite/
[4] Globus webpage, URL http://www.globus.org/
[5] UNICORE webpage, URL http://www.unicore.eu/
[6] EMI webpage, URL http://www.eu-emi.eu/home
[7] Nilsen D and Weber P 2012 J. Phys.: Conf. Ser. 396 032079
[8] Subversion webpage, URL http://subversion.apache.org/
[9] Foreman webpage, URL http://theforeman.org/