

A perl script for automated API bundle installations (SAGA)

SAGA-Group

06-13-2011

Abstract

Developing, testing and debugging software packages before making them available on the production environment should be well tested on various working platforms. Packages to be released in future, or an update made to the previous version, are rigorously made bug-free and versatile to the changing environments and dependency packages. These noted steps involved in software deployment cycle of any software package can often be complex and is made short through automated build systems, and continuous integrated frameworks [1]. The best case however is to have a software deployment tool that automates every single step of deployment - developing, bug tracking and distribution. Present paper deals with a simple perl script (mephisto.pl) that automates build process of SAGA (Simple API for Grid Applications) along with its dependencies. It is light-weight, flexible, and portable that can be generalized to any software development process. The paper presented goes through the existing problem faced and mephisto as its possible solution. The paper includes its future possibilities of including mephisto into a continuous integration framework and serve as a software deployment tool for SAGA. The paper tries to compare some of the existing software deployment tools such as NMI-GRIDS testing, NMI build and test lab, and automated building tools such as Buildbot to explain their standing with respect to the script being developed [2].

1. Introduction

Developing software packages require immediate bug fixes and tools for faster deployment. Like many other software packages, SAGA was required to be installed on different platforms to carry out version compatibility tests of its dependency packages, bug fixes and running applications. In the beginning manual installations were employed to get a working copy of SAGA on different machines. This was time consuming since installations required its dependencies to be taken care of, which differed with every working platform. Also any updates made to SAGA or its dependencies should be tested on different platforms but installations had to be implemented prior to any testing. Tracking and reporting bugs from testing-results was also important before deploying bug free SAGA version on production environment or for applications to be carried out. This cyclic process required repeated installations and manually installing them every

time were to slow the development cycle. Also extra care was to be taken to make sure the development community were kept informed of the different bugs suspected, their logs, changes already made and tested. This entire process required a central administrator to co-ordinate all the work that lacked a distributed framework essential for a smooth and exponential development process. Hence the problem was to develop a tool that automated installations, tracked bugs and maintained a log record for the developers which would in turn facilitate parallel independent research work into developing SAGA. This is practiced in many software development process for faster development and to stay stable with its dependency packages on all platforms alike [3]. Thus as an initial step to build a complete deployment tool for SAGA, 'mephisto.pl', a perl script was developed to fill in for automated installations or building SAGA automatically. A simple command line option would install SAGA and all its dependencies irrespective of the underlying computer resource used. In future it would be integrated with Buildbot, an automated software testing tool. The paper will proceed to explain more about mephisto and its working in Section 3 following some of the existing deployment tools and possible discussion in Section 2.

2. Existing methods

There are wide variety of software deployment tools available in the market both commercial and open source. Deployment tools available for grid computing software packages are constrained to a few. CMS and ADEM are among the few deployment tools employed for grid software packages at CERN [4, 5]. A few other tools are aimed just at automating development and building process together termed as building systems. Building systems can be categorized into on-demand, scheduled, and triggered automation [6]. NMI scripts, and buildbot are to name a few.

2.1 NMI scripts and NMI build and test labs

NMI-GRID scripts, an NSF funded project was aimed at developing a software testing structure that automated building and testing of software packages. The primary objective of the project was to automate the entire process without the control being centered on a single local administrator. Any commits done to the subversion repositories by project developers would automatically trigger temporary building process on different working platforms while logging each result either failure or success.

Multiple requests for testing are queued before executing them at a time. All the results are stored which are visible for the entire developers community alike to be reviewed later [7, 3]. Several other NMI building and testing labs available for scientific community are maintained by UW madison, CERN, INFN, and TeraGrid. They provide both infrastructural and software support necessary for development and building process [8].

2.2 Buildbot

Buildbot is one such other way of triggered automation employed by SAGA developers group. The architecture consists of a build master that triggers a series of builds and testing upon each commits done to subversion or cvs or any such repositories or directories. The testing process are to be carried out on its slave machines working on different platforms. All the notifications after each built are sent to all developers community either by mailing or maintaining a web space. All the governing options can be configured prior to any of the developing process. This is an open source project targeted to a wide range of software packages [9].

3. Working of mephisto

Example of the command line statement:

```
perl mephisto install
--target-dir=/home/user/saga
--tmp-dir=/home/user/tmp-saga
```

3.1 Motivation and reason

Buildbot was already used as an automated tool for solving SAGA building problems. But it still required to install all the dependencies of SAGA on different machines where (buildbot slaves) SAGA testing was performed. This operation could not be scaled very quickly to newer machines or platforms to be added further. Thus it required a tool that downloaded all its dependencies temporarily on the disk space and then continued building process. This was necessary as different platforms provided different dependency challenges. It was required to test different versions of the packages at users discretion for complete testing which buildbot failed to provide. Thus mephisto was developed such that it could download all its dependency packages with the specified versions using a single command line option provided by the user and proceed to build SAGA.

3.2 Commands for execution

Given are all the various options available at users disclosure. Mephisto pulls up the packages information from a

pre-defined repository. By default if no option is specified, mephisto pulls up the packages listed on repository at

```
http://static.saga.cct.lsu.edu/
mephisto/repository/latest/INDEX.
```

Expanded version of the Command to run mephisto

```
perl mephisto install/test
--target-dir=/path/to/folder
--tmp-dir=/path/to/folder
--with-saga-version=
--with-boost-version=
--with-globus-version=
--repository= --with-adaptors=
--with-python-version=
--with-postgresql-version=
```

install

This can be used to install a working copy of SAGA and all its dependencies specified.

test

This option not only installs SAGA but also runs the command 'make check' to check the installation. It creates a log file displaying the test results.

target-dir

This option is used to specify the path you need to install SAGA and all packages libraries. By default, it creates a folder at tmp/meph_inst.

tmp-dir

This option is used to specify the path for a temporary folder to download all the packages and create log files. If not mentioned, it creates a folder at /tmp/meph_tmp

repository

Can be used to choose different repositories to install SAGA from. Two repositories available as of now are 'latest' (default) and svn_trunk. Recommended svn_trunk for the most recent SAGA version.

with-saga-version

Choose saga versions available.

with-boost-version

Choose boost versions available $\geq 1.40.0$

with-globus-version

Choose globus version available ≥ 4.0 . This option would automatically invoke installations of X509, globus adaptors too.

with-adaptors

Mention any adaptors to install that would invoke globus installations along with saga-adaptors too.

with-python-version

Choose from the list of python versions available. Must be $\geq 2.6.2$

with-postgresql-version

Choose from postgresql versions available. Should be $\geq 8.4.1$

3.3 Repositories for Mephisto

Mephisto uses two different repositories for SAGA packages which have been pre-defined by the SAGA group, namely:

(i) <http://static.saga.cct.lsu.edu/mephisto/repository/latest/>

(ii) http://static.saga.cct.lsu.edu/mephisto/repository/svn_trunk

When the user doesn't define any repository, the default repository taken by mephisto will be <http://static.saga.cct.lsu.edu/mephisto/repository/latest/>.

3.4 Future of mephisto

This simple script could thus provide a working copy of SAGA on all the different working platforms alike with command line options specified. It could also be used to choose the type of version available. This would be included with buildbot scripts to form a continuous integrated testing tool such that the script should be deployed on various working platforms and later send the notifications of the testing results. This would be completely independent of the underlying working environment [10]. The future would be to develop a way to deploy SAGA on required machines upon obtaining a completely reviewed bug free SAGA version when triggered manually for deployment.

3.5 Availability

A copy of mephisto.pl can be obtained from <http://faust.cct.lsu.edu/trac/mephisto>. Other details and documentation can be found on the same website. An installed version of perl is the only necessary component on your system to run mephisto.pl

4. Conclusion

The paper presented a solution to automate building process for SAGA with its various dependency packages as the first step in the process of developing an entire distributed deployment structure for SAGA. The script can

very easily be modified to apply for automated building process of any software packages. It is flexible, and portable. The perl script can also be easily integrated to any automated testing software tool to become a continuous testing platform for faster software development.

References

- [1] "Software deployment." [Online]. Available: http://en.wikipedia.org/wiki/Software_deployment
- [2] "Build automated tools." [Online]. Available: http://en.wikipedia.org/wiki/List_of_build_automation_software
- [3] A. Pavlo, P. Couvares, R. Gietzel, A. Karp, I. D. Alderman, and M. Livny, "The NMI Build & Test Laboratory: Continuous Integration Framework for Distributed Computing Software," in *The 20th USENIX Large Installation System Administration Conference (LISA)*, December 2006.
- [4] B. Kim, M. Thomas, P. Avery, and F. Wuerthwein, "CMS software deployment on OSG," *Journal of Physics: Conference Series*, vol. 119, no. 5, p. 052023, 2008. [Online]. Available: <http://stacks.iop.org/1742-6596/119/i=5/a=052023>
- [5] Z. Hou, M. Wilde, X. Zhou, I. Foster, and J. Tie, "ADEM: An Automation Tool for Application Software Deployment and Management on OSG," pp. 130–137, December 2009.
- [6] "Build automation." [Online]. Available: http://en.wikipedia.org/wiki/Build_automation#cite_note-5
- [7] NMI-GRIDS, "NMI-Grids: NSF funded project for deploying NMI packages." [Online]. Available: <http://www.grids-center.org/index.php>
- [8] "NMI build and test lab." [Online]. Available: <https://nmi.cs.wisc.edu/>
- [9] B. Warner, "Buildbot: Automate compile/test cycle." [Online]. Available: <http://buildbot.net/buildbot/docs/latest/index.html#Top>
- [10] "Continuous integration." [Online]. Available: http://en.wikipedia.org/wiki/Continuous_integration