StratusLab

Enhancing Grid Infrastructures with Virtualization and Cloud Technologies

Management Infrastructure and Metrics Definition

Milestone MS1 (V1.0)
1 September 2010

Abstract

This document describes the project management bodies, the software development process, and the tools to support them. It also contains a description of the metrics that will be collected over the lifetime of the project to gauge progress.

The information contained in this document represents the views of the copyright holders as of the date such views are published.

Contributors

Name	Partner	Sections
Charles Loomis	CNRS/LAL	All

Document History

Version	Date	Comment
1.0	1 Sept. 2010	First version for internal review.

Contents

List of Tables

1 Management Infrastructure

1.1 Formal Management Bodies

The project's Technical Annex and Collaboration Agreement define two formal management bodies for the project: Technical and Scientific Coordination Group (TSCG or "The Group") and the Project Management Board (PMB).

The TSCG is composed of the Activity Leaders of each activity and the Technical Coordinator. The Group has decided to alternate technical meetings and administrative meetings to ensure that each meeting remains focused, short, and effective. This group ensures that the project follows the defined work plan and progresses towards accomplishing the project's goals.

The PMB is the final executive body within the project consisting of a representative from each partner and is chaired by the Project Coordinator. This group defines the overall project policies (e.g. intellectual property issues) and ensures that partners meet their obligations to the project. This group meets at least one per quarter.

The TSCG and PMB agendas are available via Indico. Minutes of the TSCG are circulated on the project mailing list and attached to the agendas. The minutes of the PMB are available from the project coordinator.

1.2 Software Development Process

The project uses agile software development processes. Agile processes were viewed as the best match to a short project that must evolve quickly and make frequent releases. The overall software development process is managed by WP4.

The software development process consists of a series of sprints each lasting 3 weeks. Each sprint (after the first release in PM4) should result in a new, incremental release of the StratusLab distribution. Each sprint starts with a Sprint Planning meeting where the scope of the upcoming sprint is defined. Each ends with a Sprint Demo meeting where everyone who has contributed to the sprint concretely demonstrates that their developments satisfy some requirement of the sprint definition.

During the Sprint Planning meeting, the Technical Coordinator and Project Coordinator act as "project owners" to guide the overall evolution of the StratusLab distribution based on feedback from the TSCG and PMB.

In addition to the Sprint Planning and Demo meetings, there is a Daily Standup Meeting. This meeting takes place by phone, starts at 10:30 each weekday, and lasts a maximum of 15 minutes. Each contributor to the sprint describes concisely: 1) the progress made in the previous day, 2) any impediments that are blocking progress, and 3) the plans for the coming day. Anyone may join these meetings to track the daily advances and current problems of the project.

To date, this software development process has been working well for the project. The project will continue using this process, with refinements, for the full duration of the project.

2 Supporting Tools

The successful execution of a software development project requires a large set of tools to manage both the project and the software development process.

2.1 Offline Tools

Many of the results of the project will be in the form of documents and presentations. Nearly all of those will require contributions from several different participants, and most will be reused in the future. To facilitate the initial production of content and the sharing of it later, the project has standardized on using LaTeX [9] for documents and PowerPoint [10] for presentations. LaTeX has the additional advantage that the source is plain text and can be managed with the code management tools adopted by the project.

To generate the various project artifacts, there is a strong preference for using maven [2]. Maven provides mechanisms (directly and indirectly) for compiling code in various languages and producing results in different formats. In addition, the management of dependencies through a standard repository will facilitate the testing and release of the StratusLab artifacts. Maven's flexibility will east the integration of distinct components into a coherent StratusLab distribution.

2.2 Online Services

Many of the tools run as online services and require credentials for authentication and authorization. To avoid having separate username and password pairs for each service, the project decided to use a central LDAP server to manage credentials. This has reduced the overheads in maintaining the services and simplified utilization of those services by the project participants.

Table 2.1 contains the services used to support the management and software development processes of the project. The table shows what implementations are used and the URLs where those services can be found.

The table also shows where the services have been deployed. Those marked as 'Amazon' have been deployed using the Amazon Web Services, a commercial cloud service. As the StratusLab distribution matures, the project expects to move those services to cloud resources hosted by the project participants. Similarly, those services residing on physical resources will also eventually move to virtualized cloud resources. This provides both a demonstration and a real-world test of the

StratusLab distribution.

Table 2.1: *Deployed Services*

Service	Implementation	LDAP	Location	URL
LDAP	ApacheDS [1]	Y	Amazon	ldaps://ldap.stratuslab.eu:10636/
Web Server	DokuWiki [5]	Y	CNRS/LAL	http://stratuslab.eu/
Agenda Management	Indico [6]	N	CNRS/LAL	http://indico.lal.in2p3.fr/categoryDisplay.py?categId=131
Mailing Lists	Mailman [8]	N	UCM	stratuslab@dsa-research.org
Issue Tracking	JIRA [4]	Y	Amazon	http://jira.stratuslab.eu:8080/
Agile Tool Support	Greenhopper [3]	Y	Amazon	http://jira.stratuslab.eu:8080/
Code Management	Git [7]	Y	Amazon	https://code.stratuslab.eu/git/...
Continuous Integration	Hudson [11]	Y	GRNET	http://hudson.stratuslab.eu:8080/
Maven Repository	Nexus [13]	Y	CNRS/LAL	http://repo.stratuslab.eu:8081/
				http://repo.stratuslab.eu:8081/content/repositories/releases
				http://repo.stratuslab.eu:8081/content/repositories/snapshots
Package Repository	yum [12]	N	CNRS/LAL	http://yum.stratuslab.eu/

3 Metrics

The project defines a set of metrics to gauge its progress over its lifetime. The general philosophy for metrics within the project is

1. To verify that proposed metrics provide relevant information for the project,

2. To collect these measures automatically,

3. To ensure that data collection meets institutional requirements, and

4. To ensure that users and system administrators are aware of what information is collected.

The proposed metrics with and without specific targets are listed in Table 3.1 and Table 3.2, respectively. These metrics will be reported starting in PM4 (after the first StratusLab release) and be available from the project website. They will also be reported in the project's quarterly and periodic reports. Metrics related to the accounting system will require integration with the EGI infrastructure; consequently, reporting of these metrics may be delayed until that integration has been achieved.

Table 3.1: *Metrics with Targets*

Metric	WPs	Source	Y1 Target	Y2 Target
No. of people on StratusLab announcement list	WP2, WP3	Mailer	25	75
No. of people on StratusLab discussion list	WP2, WP3	Mailer	50	100
No. of sites running StratusLab distribution	WP4, WP5	Information System	5	10
No. of sites exposing the cloud API	WP4, WP5	Information System	0	5
No. of VOs served via StratusLab sites	WP2, WP3, WP4	Information System	10	30
No. of sci. disciplines served via StratusLab sites	WP2, WP3, WP4	Information System	3	7
No. of available appliances	WP5	Repository	5	15
Availability of sites	WP4, WP5	Accounting	80%	95%
Reliability of sites	WP4, WP5	Accounting	80%	95%

Table 3.2: *Metrics without Targets*

Metric	WPs	Source
No. of sprints	WP4	Tracker
No. of releases	WP4	Tracker
Delivered CPU	WP5	Accounting
Delivered CPU through cloud API	WP5	Accounting
Storage used	WP5	Accounting
Storage used through cloud API	WP5	Accounting
No. of appliance downloads	WP2, WP5	Repository
No. of views of website	WP3	Web server
No. of features (by state)	WP2, WP4	Tracker
No. of bugs (by state)	WP4	Tracker
No. of sites providing scale-out	WP4, WP6	Accounting
Fraction of resources by scale-out of a site	WP4, WP6	Accounting

References

[1] Apache Foundation. Apache Directory Service. http://directory.apache.org/.

[2] Apache Foundation. Maven. http://maven.apache.org/.

[3] Atlassian. Greenhopper. http://www.atlassian.com/software/greenhopper/.

[4] Atlassian. JIRA. http://www.atlassian.com/software/jira/.

[5] DokuWiki Community. DokuWiki. http://www.dokuwiki.org/.

[6] EU InDiCo Project. Indico. http://indico-software.org/.

[7] Git. Git. http://git-scm.com/.

[8] GNU. Mailman. http://www.gnu.org/software/mailman/.

[9] L. Lamport. *LaTeX, A Document Preparation System*. Addison-Wesley, second edition, 1994.

[10] Microsoft. PowerPoint. http://hadoop.apache.org/.

[11] Oracle. Hudson. http://hudson-ci.org/.

[12] Seth Vidal, Duke University. Yellowdog Updater Modified. http://yum.baseurl.org/.

[13] Sonatype, Inc. Nexus. http://nexus.sonatype.org/.