HiPC 2007 Poster

GARUDA Grid Integrated Testing Methodology Test Suite -GITeMS

VCV.Rao, K Shiva Prasad, Irshad Khan, Jyothi N K, Nisha A, Samrit M
{vcvrao, shivak, irshadk, jyothink, anisha, samritm}@cdac.in

National PARAM Super Computing Facility, Centre for Development of Advanced Computing (C-DAC), Pune University Campus, Ganesh Khind, Pune 410007, India

Abstract

We describe GARUDA (http://www.garudaindia.in/) grid integration test script that is used for testing grid programming environment of GARUDA and measure the grid middleware overheads. GARUDA Integrated Testing Methodology Suite (GITeMS) is a set of test suites that check the basic grid services, quantify the Grid middleware overheads and performance of grid probes. The objective is to execute test programs on integrated software and hardware infrastructure of grid computing which is a vital aspect building of GARUDA [5]. The test programs are based on programming languages such as C, C++, with Globus Toolkit as well as Perl CoG Kit, python CoG Kit and Java CoG Kit. The test script is executed on GARUDA sites and exposed to the application characteristics, as far as possible, independent of the Grid middleware incarnation used. GITeMS can be downloaded from C-DAC web-site http://www.cdac.in

1. Introduction

Grid computing infrastructure offers a wide range of distributed resources for execution of large-scale resource intensive applications. The testing of utilizing grid resources for managing, monitoring, charging the usage of these huge set of resources spread under varied administrative domains is quite challenging. The grid infrastructure required rigorous testing of operating systems and programming languages on various heterogeneous platforms [1-4,6,8]. GARUDA is a collaboration of scientific and technological researchers on a nation wide grid comprising of computational nodes, mass storage and scientific instruments. The Garuda network is a Layer 2/3 MPLS Virtual Private Network (VPN) connecting select institutions at 10/100 mbps with stringent quality and Service Level Agreements. GARUDA aims at strengthening and advancing scientific and technological excellence in the area of Grid and Peer-to-Peer technologies. To achieve its objective, GARUDA brings together a critical mass of well-established researchers, from 45 research laboratories and academic institutions of India.

2. An overview of GITeMS

The test suites are organized in *three* parts and each part has *different levels of complexity in testing the GARUDA services* as given in figure 1. The idea is to *provide a common ground* to test GARUDA [6] deployed grid; some of them monitor the operational issues of GARUDA sites, and others estimate the overheads associated with grid middleware, focusing on services at various layers. Some test suites focus on different grid services of GARUDA with emphasis on grid programming.

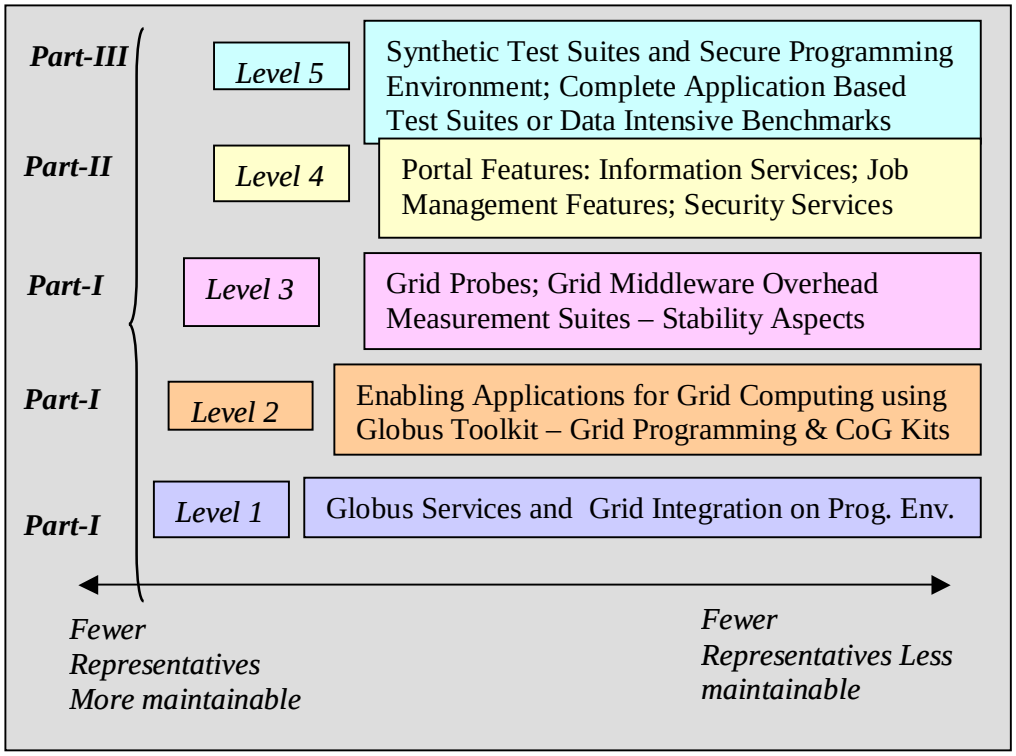

Fig. 1 Representation of test suites (Level. 1. Level 2.Level 3. Level 4. Level 5)

The test package is scalable and it is possible to vary the number of geographical sites to be used and the size of the test problem to be solved [15,16]. The test suite is also designed diversity of grid middleware, in which grid overheads is quantified. The other test suites named as synthetic *benchmarks codes*, which capture basic use-patterns of several applications across different scientific domains.

The test suites in part-I address several issues regarding basic grid services, grid programming environment and grid middleware overheads. Three different levels (Level-1, Level-2, Level-3) of test suites are included in part-I. In the part-II, the test features of grid portal, as given in Level-4 are highlighted. The Level-5 test suites in part-III cover synthetic application characteristics, grid enabled MPI programs, and data intensive benchmarks. The test suites in Level-1 address basic grid services and grid-programming environment such as basic grid services, interoperability test; basic grid programming environment test; and basic grid programming environment test. The Level-2 test suites focus on integrated test scripts to enable applications for grid computing using different programming languages. The test programs are based on programming languages such as C, C++, Java with Globus toolkit, Perl CoG, Python CoG and Java CoG. The Level-3 test suites quantify grid middleware overheads in which test programs such as *Sleep, Ping-Pong, Gather, Circular* probes have been executed to demonstrate the usability and robustness of GARUDA. The Level-4 test suites focus on testing grid portal features, which addresses many of the issues related to easy access to GARUDA resources and performing various grid operations such as portal environment set, disk space information as per job requirements, site information, interactive services, latency associated with portal system, batch-queuing job information, Wait time prediction, input/output operation, list of Job managers, compilation and execution; error messages, static information about resources,

interoperability test, and grid programming environment test. The Level-5 test suites measure communication overheads across various sites in a grid environment. The aim is to develop and execute a set of benchmarks, which measure the performance in grid environment in terms of useful metrics like turnaround time, and throughput. Synthetic tiny suites for secure grid programming environment for controlling the application process development are developed and the MPICH-G2 test suites environment in which the programs such as *Ping-Pong, Circular, Gather, Client Server Application, Topology discovery, Matrix Computations, Compute Intensive Grid Benchmarks:* (NAS Grid Benchmarks) are executed on GARUDA. The open source software such as GridBench and GRASP software have been used.

3. Description of GITeMS Level-2 & Level-3 Test Suites

The Level –2 test suites focus on integrated test scripts to enable applications for grid computing using programming languages with Globus toolkit. The languages such C, *C++, Perl, python*, and *Jav*a interfaces with the Globus toolkit play an important role to solve scientific applications in which the job completion is done from multiple distributed components using workflow concepts. The test suites are developed to focus on multi-tiered communication costs, and possibly heterogeneous computation speeds as the clusters at different locations can differ in their hardware and/or software characteristics. The suites can be executed on command-line from the client site and these suites can be executed for repeated number of iterations to address the stability aspects of grid infrastructure.

INPEAG-v1.0: The INPEAG-v1.0 (INtegrated Perl script to Enable Applications for Grid computing with Globus) Software is an integrated Perl script, which checks the Globus installation and support of grid services using *RSL* scripts. The objective is to check the basic grid capabilities such as valid grid proxy, mutual authentication, *GridFTP*, remote job submission, submission of job using *RSL* and discovery and monitoring of computing resources. The test suite focus on many issues that grid operational staff needs to be aware of when grid computing infrastructure is made available for end-users. The integration script, INPEAG-v1.0.pl integrates the test suite: *MDS–GRIS* (Monitoring and Discovery Service), *GRAM* (Globus Resource Allocation Manager), *GRAM-Batch*, *GridFTP*, *GridFTP-RSL*, *JOB MANAGER*, DUROC, and *RSL* (Resource Specification Language).

GEAGUL-C-v1.0 : The **`GEAGUL-C v1.0`** **(**Grid Software – Enabling Applications for Grid Computing Using GLobus and C-Language**)** is a set of test suite written in C-language using Globus APIs. The aim is to check the basic grid capabilities such as remote job submission, validation of proxy, mutual authentication etc. using rich set of Globus APIs, which mimic the various application characteristics. The test suite provide foundation for distributed workflow applications, which are first specified using a novel high-level abstract workflow language that shields the user from any grid middleware implementation or technology details. The abstract representation can be mapped to a concrete workflow that can be scheduled, deployed and executed on multiple grid sites.

GOPAEAG -v1.0 : The test suite **`GOPAEAG v1.0`** **(**Globus and Object Oriented Programming Approach to Enable Applications for Grid computing**)** is a set of test

programs written in *C++* language using Globus APIs. The test suites mimic the various application characteristics and integrate C++ applications with the components provided by Globus framework. The test suite serves as a preliminary user's guide to simplify the development of enabling grid applications using *Object Oriented* Programming. The objective is to check the basic grid capabilities such as remote job submission, validation of proxy, mutual authentication, GridFTP, GASS using rich set of Globus APIs, and perform computations on remote sites.

JAGEAG v1.0: The **JAGEAG- v1.0** (JAva CoG Kit and Globus to Enable Applications for Grid Computing) Software is a set of test programs using Java CoG Kit [7,11,13] and Globus toolkit to enable applications for grid computing. It supports rich set of Globus APIs and mimic the various applications. The **JAGEAG-v1.0** software is designed to enable applications for grid computing using Java CoG Kit and Globus toolkit. The set of programs support rich set of Globus packages and mimic the various application characteristics. The aim is to check the basic grid capabilities such as validation of grid proxy, mutual authentication, remote job submission using *RSL,* discovering and monitoring of computing resources, check the running of *HTTP, LDAP, MDS* and *GridFTP* server on remote sites. Figure 2 illustrates the integration of grid and commodity technologies.

PerlCEAG-v1.0: The **PerlCEAG-v1.0** is designed to enable applications for grid computing using Perl CoG Kit [10] and Globus toolkit which checks the Globus installation and support of grid services. The objective is to check the basic grid capabilities such as valid grid proxy, mutual authentication, *GridFTP*, remote job submission, submission of job using *RSL* and discovery and monitoring of computing resources and batch job submission. The **PerlCEAG-v1.0** depends on the access to Grid resources and the suites can be executed on command-line from the client system. **PerlCEAG-v1.0** test suite uses Globus components *GRAM* (Grid Resource Allocation Manager), *GridFTP, MDS , RSL*.

.

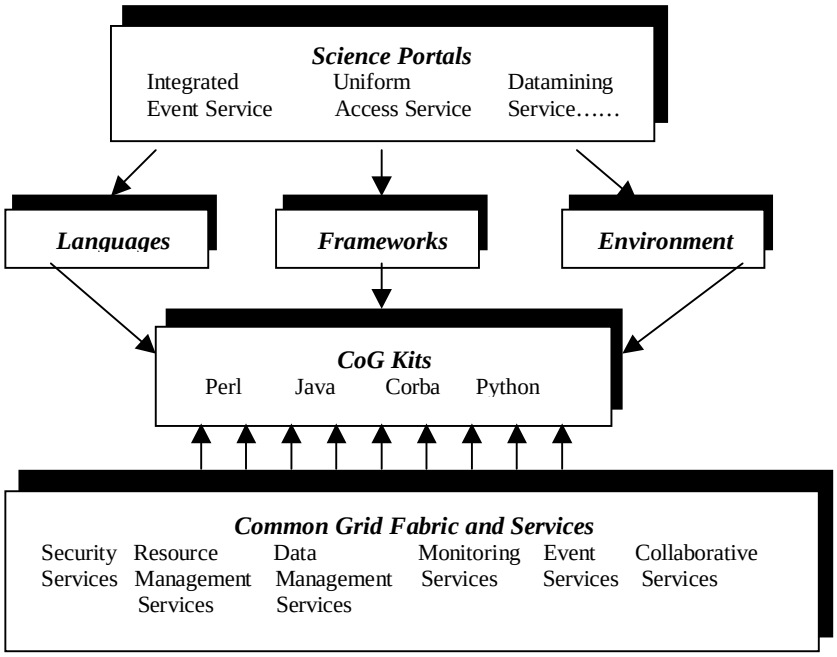

Figure 2: CoG Kits provide a mapping between computing languages, frameworks, and environments and Grid Services and fabric.

pyCEAG-v1.0: The **`pyCEAG-v1.0`** is designed to enable applications for Grid computing using Python CoG Kit [17] and Globus-4.0 which checks the Globus installation and support of various Grid services. The objective is to check the basic Grid capabilities such as valid *Grid proxy, mutual Authentication, GridFTP Services, GASS Services, Batch Job Submission*. The test cases such as *Circular data transfer*, *Gather data transfer* and matrix-matrix multiplication mimic the application characteristics and checks reliability of Grid from users perspective. The test suite provides foundation for the applicability of a grid environment to applications.

SPAGMOS-v1.0: The **SPAGMOS-v1.0** (Software Probes for Assessment of Grid Middleware OverheadS) software is a set of grid probes to estimate the grid middleware overheads such as inter-site communication for different message sizes, data transfer, basic rates of networks, and grid information service. The suites can be considered as grid probes (low level grid benchmarks), which model the communication overheads among the grid nodes and measure the time taken for different grid services. The probes can demonstrate the usability and robustness of the probes by testing in an automated fashion for many repetitions and investigate the extent to which our probes can provide insight into the grid stability, robustness, and performance. The GridFTP performance between GARUDA sites is given in the Fig. 3. The bandwidth performance results for data file of size 1 MB to 1 GB using multiple TCP Streams in parallel is obtained for various test runs on GARUDA.

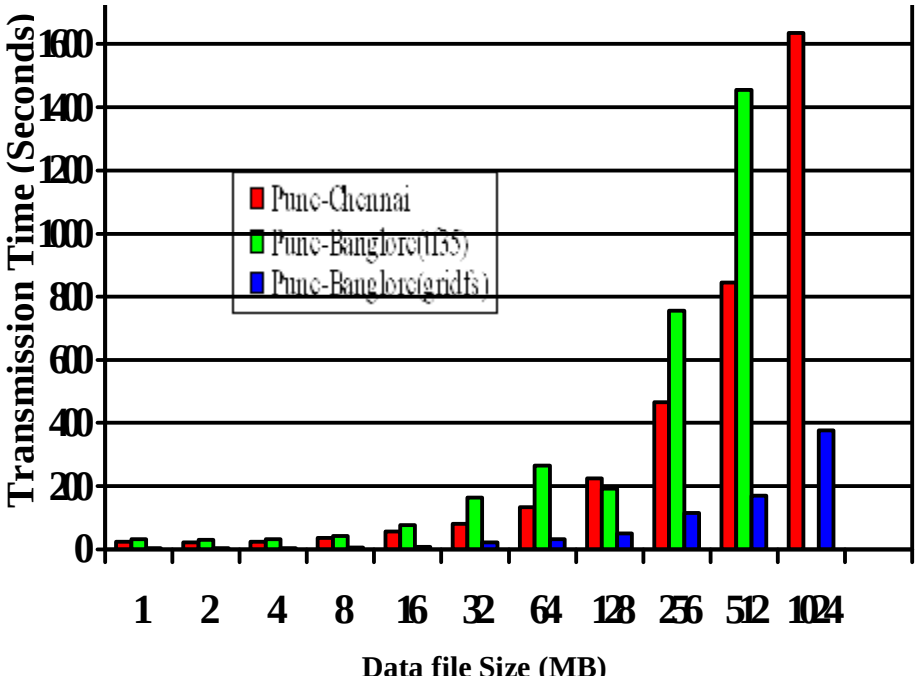

Fig. 3 Performance of GridFTP on GARUDA

4. References

1. I.Foster and C. Kesselman (eds) (**2004**) *The Grid Blueprint for a Future Computing Infrastructure*, 2nd edn, Morgan Kaufmann Publishers
2. I.Foster, C. Kesselman, and S.Tuecke (**2001**) "The Anatomy of the Grid: Enabling Scalable Virtual Organizations, " *International J. of Supercomputer Applications*, 15 (3), 200-222.

3. F.Berman, A Hey, and G. Fox (**2003**), *Grid Computing: Making The Global Infrastructure a* Reality, John Wiley & Sons Ltd.
4. The Globus Toolkit, http://www.globus.org/toolkit/about.html
5. GARUDA, The National Grid Computing Initiative, http://www.garudaindia.in
6. Global Grid Forum, http://www.Gridforum.org
7. CoG Kit Information, www.cogkit.org/
8. The TeraGrid Project: http ://www.teragrid.org
9. Java CoG Kit user manual, http://globus.org/cog/manual-user.pdf
10. The Perl Commodity Grid Toolkit, www.mcs.anl.gov/~gregor/papers/vonLaszewski--perl-cog.pdf
11. Java CoG Kit Tutorial, www-unix.mcs.anl.gov
12. Enabling Applications for Grid Computing with Globus, IBM Red Books; www.redbooks.ibm.com/redbooks/pdfs/sg246936.pdf
13. Gregot von Laszewski, Jarek Gawor, Peter Lane, Nell Rehn, and Mike Russel, *Features of the Java Commodity Grid Kit*; Concurrency and Computation: Practice and Experience; 2001
14. Gregor von Laszewski, Ian Foster and Jarek Gawor *CoG Kits : A Bridge between Commodity Distributed Computing and High-Performance Grids*
15. The UK Grid Integration Test Script – GITS http://www.soton.ac.uk/~djb1/gits.html
16. Grid Status Test: http://grid.ncsa.uiuc.edu/test/grid-status-test/tests.html
17. Python CoG Kit http://dsd.lbl.gov/gtg/projects/pyGlobus/