Monitoring User-level Functionality and Performance using Inca

Shava Smallen*

Abstract—**Inca is a user-level monitoring tool that has been used successfully in cyberinfrastructure projects such as the TeraGrid and other similar projects. Its flexible architecture enables multiple types of functionality and performance data to easily be collected about the systems it is monitoring. A unique centralized monitoring configuration approach makes Inca easy to setup and maintain on multiple systems and lends to consistent monitoring results. We are experimenting with using Inca in Cloud environments where users are interested in monitoring the functionality and performance of their personal Clouds as they are re-instantiated on different machines and locations. We believe Inca's flexibility, easy-to-use administrative interfaces, and other features make it a natural fit for Cloud environments.**

Index Terms— **Monitoring, Cyberinfrastructure, Cloud, Reliability, Performance**

I. INTRODUCTION

Cyberinfrastructure (CI) aggregates multiple complex and interdependent systems that span several administrative domains. This complexity poses challenges for both administrators who build and maintain CI resources and scientists who use them. These challenges and the cutting-edge nature of the infrastructure make it susceptible to failures, which frustrate and even dissuade users. In order to provide scientists with persistent and reliable CI, the user-level functionality and performance of the system must be monitored. Developed in 2003, *Inca* [1] is a tool that provides user-level monitoring for CI systems. It has proven to be a valuable tool for the TeraGrid [2] and eleven other production CI projects [3], and is currently monitoring over a hundred machines. Inca differs from other monitoring tools with its user-level monitoring approach, flexibility to collect multiple types of monitoring data, and full archiving support. Inca also uses a unique centralized monitoring configuration approach, which is essential for providing consistent monitoring results and a unified and coherent infrastructure for users.

Inca automates the installation, execution, and maintenance of a large variety and number of reporters (monitoring scripts) consistently across Grid resources. Monitoring results are collected, archived, and displayed through web status pages.
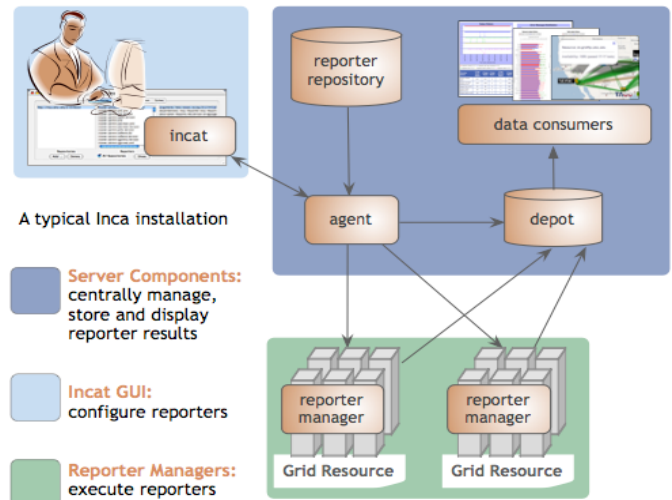
Figure 1: Inca architecture.

The core of the Inca architecture is formed by four server components: the agent, depot, data consumer, and reporter repository (see Figure 1). The agent controls the deployment and maintenance of the Inca installation, while the depot provides data storage. A reporter repository contains a set of tests and benchmarks, and the data consumer displays the collected results as web status pages. An additional Inca component called the reporter manager runs on each monitored resource to collect status and performance data. The final component of Inca is a GUI tool (incat) used by Inca administrators to configure and maintain their Inca deployments.

Recently, we have been exploring how to leverage Inca in virtual Cloud environments as part of the work we are doing in FutureGrid [4]. FutureGrid is an experimental testbed that will be used to develop and test novel approaches to parallel, grid, and cloud computing. It supports user "experiments" that can be run either on the bare hardware or on virtual machines using Cloud technologies Nimbus or Eucalyptus. In particular, we are interested in using Inca to validate a user's virtual environment started up by these tools both in terms of functionality and performance. For example, a user may want to know how the performance of their virtual cluster compares to a month earlier when they last instantiated it. We are also interested in using Inca to validate individual virtual machine images as well.

We propose a poster that describes the Inca architecture, its capabilities, and how it could be used in a Cloud environment
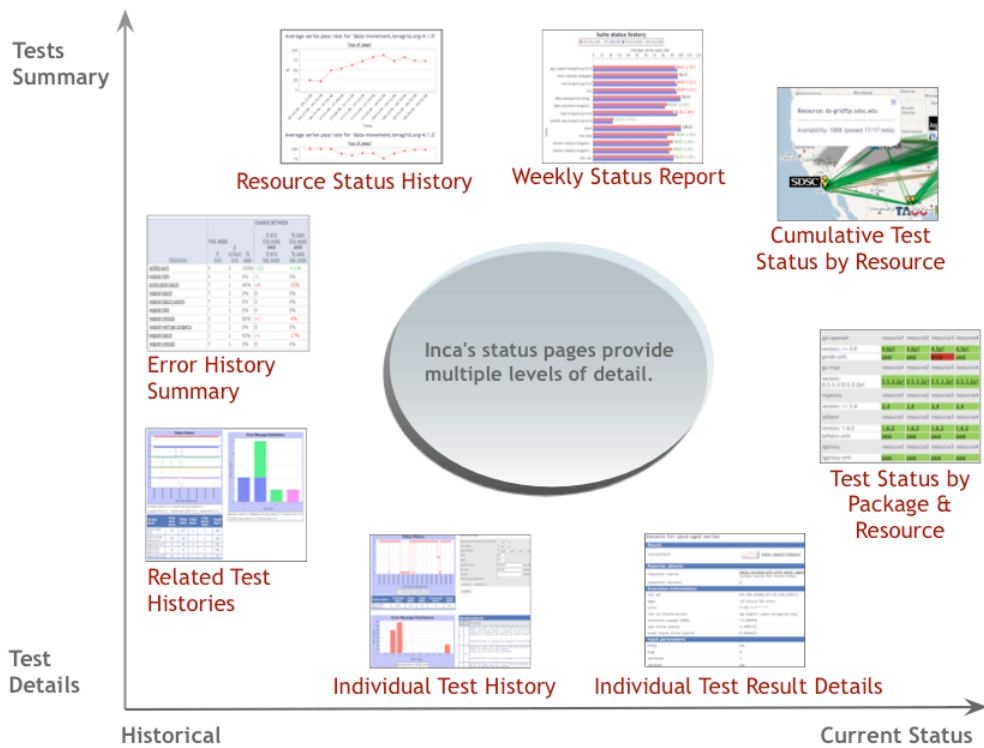
Figure 2: Shows the different types of views provided by the Inca status pages that vary by how much detailed data and historical data is displayed.

using our planned work on FutureGrid. The poster will include an architecture diagram similar to that shown in Figure 1, examples of its web interfaces as shown in Figure 2 and Figure 3, in addition to a use cases for leveraging Inca in FutureGrid.

REFERENCES

[1] S. Smallen, K. Ericson, J. Hayes, and C. Olschanowsky 2007. User-level grid monitoring with Inca 2. In Proceedings of the 2007 Workshop on Grid Monitoring (Monterey, California, USA, June 25 - 25, 2007). GMW '07. ACM, New York, NY, 29-38.
[2] The TeraGrid Project Web page. http://www.teragrid.org, 2010.
[3] The Inca Project Web page. http://inca.sdsc.edu, 2010.
[4] The FutureGrid Project Web page. http://futuregrid.org/, 2010
[5] The Nimbus Project Web page. http://www.nimbusproject.org/, 2010.
[6] The Eucalyptus Project Web page. http://www.eucalyptus.com/, 2010.
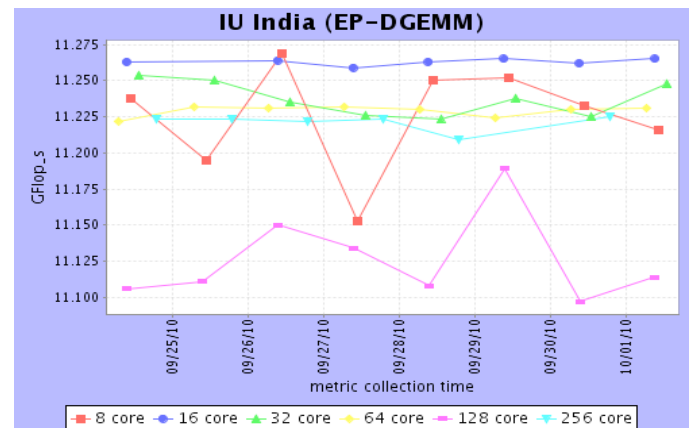[7] The HPC Challenge Project Web page. http://icl.cs.utk.edu/hpcc/, 2010.

Figure 3: Shows performance data collected from a run of the HPCC [7] benchmark on one of the FutureGrid machines at Indiana University.