Dynamic Provisioned Experiments in FutureGrid

Gregor von Laszewski[1], Geoffrey C. Fox, and the FutureGrid Team
[1] Corresponding Author
Pervasive Technology Institute, Indiana University
2729 E 10th St., Bloomington, IN 47408, U.S.A.

***Abstract*—FutureGrid provides novel computing capabilities enabling reproducible experiments while supporting dynamic provisioning. This paper describes the FutureGrid project and experiment abstractions to create and execute such experiments. The experiments consists of the activities performed by the various users of FutureGrid ranging from administrators, software developers and end users. To coordinate Experiments we have introduced the concepts of FG Projects that allow users to coordinate several experiments in order to generate a reproducible set of tasks and resource configurations. Additionally, the project abstraction is used to share not only the experiment setup, but also performance information for the specific instantiation of the experiment. This makes it possible to compare a variety of experiment setups and analyze the impact Grid and cloud software stacks have.**

I. Introduction

FutureGrid (FG) [1] provides computing capabilities that will enable researchers to tackle complex research challenges related to the use and security of grids and clouds. FG provides a significant and new experimental computing Grid and cloud test-bed to the research community, together with user support for third-party researchers conducting experiments on FutureGrid. The test-bed will make it possible for researchers to conduct experiments by submitting an experiment plan, which is then executed via a sophisticated workflow engine, preserving the provenance and state information necessary to reproduce the experiment. The test-bed includes a geographically distributed set of heterogeneous computing systems, a data management system that will hold both metadata and a growing library of software images necessary for cloud computing, and a dedicated network allowing isolated, secure experiments. The test-bed will support virtual machine-based environments, as well as operating systems on native hardware for experiments aimed at minimizing overhead and maximizing performance. The project partners will integrate existing open-source software packages to create an easy-to-use software environment that supports the instantiation, execution and recording of grid and cloud computing experiments. One of the goals of the project is to understand the behavior and utility of cloud computing approaches. Recently, cloud computing has become quite popular and a multitude of cloud computing middleware have been developed. However, it is not clear at this time which of these toolkits will become the users' choice toolkit. FG provides the ability to compare these frameworks with each other while considering real scientific applications. Hence, researchers will be able to measure the overhead of cloud technology by requesting linked experiments on both virtual and bare-metal systems, providing them valuable information that will help them decide which infrastructure suits them better and also help users that want to transition from one environment to the other.

The FG Team. The list of participants to build the FutureGrid capacity includes Indiana University, Purdue University, San Diego Supercomputer Center at University of California San Diego, University of Chicago/Argonne National Laboratory, University of Florida, University of Southern California Information Sciences Institute, University of Tennessee Knoxville, Texas Advanced Computing Center at University of Texas at Austin, University of Virginia, Center for Information Services, and GWT-TUD from Technische Universtität Dresden. However, users of FG do not have to be from these partner organizations. Furthermore, we hope that new organizations in academia and industry can partner with the project in the future.

II. FG Dynamic Provisioning

The goal of dynamic provisioning in FG is to partition a set of resources in an intelligent way that provides a user defined environment to any user that makes such a request. This entails a specific, specialized deployment which can allocate and deallocate resources in real-time. As such, customized environments need to be in place and be able to dynamically add and remove resources depending on the overall system load and utilization. Dynamic provisioning is used in several contexts as part of FG:

- **Dynamic Resource Assignment.** Resources in a cluster may be reassigned based on the anticipated user requirements, e.g. a server may be participating as part of an HPC application on the machine, but at a later time the server is removed from the HPC resource pool and included through dynamic provisioning into a Eucalyptus Cloud. Resources that are not used are in a "unused resource pool".
- **Execution-based Dynamic User Requested Resource Assignment.** At the time of the job execution, a system is provisioned that fulfills the user's need at runtime.
- **Queue-based Dynamic User Requested Resource Assignment.** Since the provisioning of images is time consuming, it is often possible to queue such jobs with the same image requirement in a queue and instantiate the provisioning before all jobs are executed which are belong to the queue.

This capability is unique and offers users a new perspective on exploring systems research within a Cloud or Grid computing deployment. In its current implementation of FutureGrid, the dynamic provisioning features are provided by a combination of using XCAT [2] and Moab [3]. As the term dynamic provisioning is not consistently used in the community, we use the term “raining” within the FutureGrid project as a description for placing an environment onto resources. The reason is that our use of dynamic provisioning goes beyond the services offered by common scheduling tools that provide such features. In fact we want our users to rain an HPC, a Cloud environment, or a virtual network onto our resources with little effort. Hence we will provide simple command line tools supporting this task. A recent example within FutureGrid is to “rain” a Hadoop environment defined by a user onto a given cluster. Instead of the user having to learn a complex set of commands that depend on intrinsic functions of the queuing system and low level support software, users can simply invoke a command:

```
fg-hadoop -n 8 -app myHadoopApp.jar ...
```

The dynamic provisioning and scheduling of the job is handled exclusively by the FG system. Users and administrators do not have to set up the Hadoop environment as it is done for them (see Figure 1).

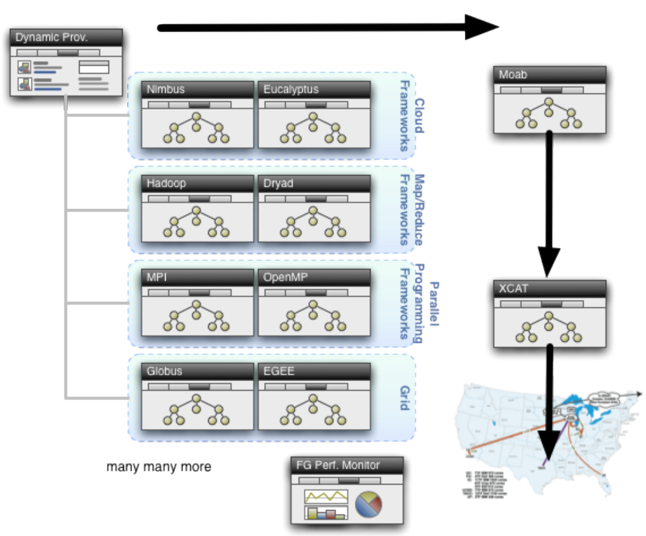

Fig. 1. Dynamic Provisioning in FG allows high level “raining” of systems onto the FG resources.

III. Project and Experiment Abstractions

Experiments are carried out using the scientific method to answer a question or investigate a problem [4]. As in physics, FG experiments typically contain one or more hypothesis that are supported by the experiment or disprove the hypothesis. They also include an experiment resource that is used to conduct the experiment. Proper recording of these activities not only allows the reproducibility of the experiment, but also the sharing of results within an interest group or the community. Moreover, an experiment resource can itself be a point of research or activities, that allow the creation of new experiments due to the sheer availability of the resource. This is a common model used in scientific discovery. For instance many astronomical discoveries would not have been possible without the invention of the telescope. FG provides such an elementary scientific instrument for system scientists. FG experiments require a sufficient description about the experiment so that a proper record useful for the community is preserved.

Activities within FutureGrid will be primarily experiment-based. These activities will be driven by steps that can be together classified as an experiment. Experiments may vary in complexity. They may include basic experiments, such as to utilize a particular pre-installed service and let a researcher debug an application interactively. They may also include more sophisticated experiments, such as to instantiate a particular environment and run a pre-specified set of tasks on the environment. We envision that a direct outcome of having such a experiment-centric approach will be the creation of a collection of software images and experimental data that provides a reusable resource for application and computational sciences. FutureGrid will thus enable grid researchers to conveniently define, execute, and repeat application or grid middleware experiments within interacting software “stacks” that are under the control of the experimenter. It will also allow researchers to leverage from previous experiences of other experimenters in setting up and configuring experiments, hence creating a community of users. FutureGrid will support these pre-configured experiment environments with explicit default settings so that researchers can quickly select an appropriate pre-configured environment and use it in their specific scenario.

IV. Conclusion

We will showcase two major components of FG namely dynamic provisioning and experiment management to support the concept of reproducable experiments. The combined features are unique and allow scientists to explore Grids and Clouds in new ways while being able to conduct performance experiments on the same hardware while being at the same time in control of the resource selection.

V. Acknowledgments

This document was developed with support from the National Science Foundation (NSF) under Grant No. 0910812 to Indiana University for ”FutureGrid: An Experimental, High-Performance Grid Test-bed.” Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] “Future grid web page,” Web Page, 2009. [Online]. Available: http://www.futuregrid.org
[2] “xcat extreme cloud administration toolkit.” [Online]. Available: http://xcat.sourceforge.net/
[3] “bcfg2.” [Online]. Available: http://www.adaptivecomputing.com/products/index.php
[4] Wikipedia, “Experiment,” Web Page. [Online]. Available: http://en.wikipedia.org/wiki/Experiment