GrADS Annual Report Year 3

II. Findings

During the reporting period (6/1/01-5/31/02), GrADS research has focused on six inter-institutional efforts: *Program Execution System (PES)*, *Program Preparation System (PPS) & Libraries, ScaLAPACK, Cactus, MacroGrid* & Infrastructure, and *MicroGrid.* The following sections summarize the findings of each subproject.

1 Program Execution System (PES)

The major findings of the GrADS efforts in the program execution system area are as follows:

- It is possible to build a generalized Grid scheduling system that effectively assigns Grid resources to a variety of applications.
- Different grid applications can have very different resource selection criteria and resource selection criteria may not be specifiable in a declarative form; therefore, a resource selection framework should provide for executable application-specific resource evaluation models.
- We have demonstrated that the enhanced, generic contract monitoring infrastructure can be used to detect contract violations with a wide set of applications and a diverse collection of performance data. Our tests with new approaches to application and system monitoring, based on compact application signatures and statistical sampling principles, have proven promising.
- We have demonstrated that an assembly-level approach to vectorization can succeed. Our prototype that analyzes Pentium 4 object code has produced 3x speedups on some loop kernels.
- We have developed new, fast techniques to reconstruct from object code many of the basic data structures needed for program analysis.

2 Program Preparation System (PPS) and Libraries

The major findings of the program preparation system effort are summarized as follows:

- To construct accurate, scalable, machine-independent models of application performance, detailed analysis of an application's instruction sequence is necessary, both to understand how computational kernels map to the resources of a target architecture and to understand the performance impact of exposed memory latency for a particular architecture and problem size.
- Automatic construction of mappers for programs written in MPI or other explicitly parallel languages can be based on clustering applied to the familiar task graph abstraction, which is used as a basis several major performance-modeling projects. Note that task graphs can be automatically constructed from program source, including programs written in HPF.
- We have developed techniques for low-level performance estimation. As one way of validating these models, we are using them to replace rule-of-thumb estimates in the implementation of classic optimizations.

3 ScaLAPACK

The major findings of the GrADS efforts related to ScaLAPACK are summarized as follows:

- High performance can be achieved on the Grid for numerical applications with a low overhead due to the GrADS framework.
- It has been demonstrated that different kinds of numerical applications can be integrated into the Grid framework with minimum effort.
- Rescheduling has been shown to be a feasible way to increase performance and throughput in the dynamic Grid environment.
- Global scheduling techniques (i.e., scheduling using simulated annealing) can be used to generate good schedules for a complex, changing environment such as a Computational Grid.
- The GrADSoft software framework can be extended and its interfaces can be defined to make it generic for multiple different applications.
- The GrADS-related technologies can be applied to create subprojects to enhance already existing numerical applications like LAPACK.

4 Cactus

We have demonstrated grid execution of the Cactus application using a performance evaluation and resource selection strategy based on an enhancement to the ClassAd system from Condor that permits the matching of sets of resources.

5 MacroGrid & Infrastructure

The MacroGrid and its associated web-based clearinghouse have been in continuous operation for the past two years. These services have matured considerably and have operated quite stably with far fewer disruptions of service during this period.

Main findings of the MacroGrid during this past year included:

- Fulfilled a critical role in development and testing of GrADS software.
- Provided an execution environment for testing GrADSoft components.
- Expanded the utility of the Virtual Organization concept and services.
- Expanded the role of information services within the GrADS software framework.

The communication channels and mechanisms put into place across the various cooperating institutions have stood us in good stead in the operation of the testbed. New applications such as the UCSD resource selector were developed and tested on the MacroGrid with relative ease. The successful testing of three distinct applications (Cactus, SCALAPACK and UCSD Resource Selector) on the testbed demonstrates the utility of the testbed as a program execution environment for the GrADSoft components.

The virtual organization structure provided by the MDS 2.0 model continues to be very effective in delivering information about the MacroGrid execution environment to both GrADS users and applications. Additional information providers are also adopting the VO concept for information delivery. An important example: The Network Weather Service has adopted the VO model for organizing and presenting NWS data.

We have found that the role of information services can be extended to provide additional capabilities required within the GrADSoft framework. For an application to run within the GrADSoft framework, additional pieces of information will be required to be available. These items could include location of parameter files, location of executables required for staging prior to execution, location of standard input/output/errors, list of target resources for executing binaries etc. We intend to extend the information services capabilities provided by MDS 2.0/2.1 and develop a GrADS Information Repository. This repository can be queried for appropriate information to make suitable decisions on resource selection and contract monitoring etc.

Furthermore, as we gained experience with GrADS-enabled applications running on the MacroGrid, it became clear that the performance of the GrADS Information Service (GrADS-IS) was often a critical factor in determining application performance. Analysis revealed that a new abstraction for dynamic managing performance data -- the VO-Grid -- was could better manage the interface between GrADSoft tools and the Grid fabric. Our subsequent development and deployment of VO-Grids for GrADS confirms the efficacy of the new abstractions and their ability to be integrated within the MacroGrid testbed.

At the same time, the VO-Grids interface is general enough to be useful to the wider community of Grid users. Having the MacroGrid environment to use as a test deployment facility has ensured that the results we have generated are applicable in a wide variety of Grid settings.

We formalized our findings on operating the information services for Virtual Organizations such as GrADS in the following technical papers.

- "Grid Information Services for Distributed Resource Sharing," K Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman. *Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10),* IEEE Press, August 2001.

- "Representing Dynamic Performance Information in Grid Environments with the Network Weather Service," M. Swany, R. Wolski, *Proc of the 2nd IEE/ACM International Symposium on Cluster Computing and the Grid*, IEEE Press, May, 2002.

6 MicroGrid

The major findings of the MicroGrid effort are summarized as follows:

- Filesystem emulation can be smoothly integrated with the MicroGrid structures and is a key element of modeling Grids and data Grids. We have incorporated these capabilities into the MicroGrid, providing a strong basis for modeling emerging use of Grids on large data systems.
- Scalable network emulation can be achieved, depending on innovative approaches to synchronization and topology-based load distribution. Scalable network emulation is a key problem for Grid modeling scaling, and we expect to demonstrate these capabilities in large-scale Grid experiments in the coming year.