II. Project Findings

During the reporting period (7/1/00-6/30/01), GrADS research has focused on six inter-institutional efforts: *Program Execution System (PES)*, *Program Preparation System (PPS)*, *ScaLAPACK Experiment*, *Cactus Experiment*, *MacroGrid*, and *MicroGrid*. The following sections summarize the findings of each subproject.

1. Program Execution System (PES)

We continue to evolve the PES components based on lessons learned in the ScaLAPACK and Cactus experiments, as well as through the ongoing revision of prototype implementations. In addition, we are building the foundation for upcoming design and development targets through a variety of basic research efforts.

The GrADSoft architecture has transitioned from a high-level design to an increasingly detailed specification, with prototype implementations in place for several components. We believe that an object-oriented, service-based approach will provide the most adaptable framework for a variety of applications.

Interfaces among applications, scheduling services, and information services have been defined and are currently in use with several functional schedulers. While minor modifications may still be necessary to accommodate unforeseen needs, we believe these interfaces will remain relatively stable.

The contract monitoring infrastructure has been exercised in a wide-area Grid with both developer supplied and application signature performance models. To date, contracts have been monitored on a task basis, and we have been able to detect unexpected behavior using the fuzzy logic decision procedures. We have shown that task contract output can be combined hierarchically to form a program-level contract monitor, but have not yet explored rule bases for this monitoring architecture. This exploration, interface hardening, and integration with PPS-supplied models will be addressed in the upcoming year.

The groundwork is in place for execution reconfiguration, and during the upcoming year we will connect the contract monitor output to the renegotiator to enable application adaptation. As adaptation occurs, and multiple GrADS-enabled applications are active at once, the G-commerce policy guidelines will become increasingly important.

2. Program Preparation System (PPS)

The major findings from the PPS project are summarized as follows:

- Construction of accurate Grid performance models within a configurable object program is essential to support the scheduling/service negotiation and contract monitoring process. These performance models may best be based on a strategy that uses preliminary executions as a basis for estimates.

- To support the program execution system, it is essential that a configurable object program provide facilities for tailoring the program to the available resources. This has led to the development of the Application Manager interface.
- High level programming interfaces based on scripting languages, ranging from simple to sophisticated should be used to hide the details of Grid implementation from the typical end user.
- Libraries should incorporate latency-tolerant algorithms and be parameterized to facilitate adaptivity to different Grid execution environments.

3. ScaLAPACK Experiment

The set of ScaLAPACK experiments reported was more challenging than originally anticipated. Part of the challenge stemmed from the fact that we had to coordinate a number of machines across different administrative domains. Part of the challenge was due to the varying degrees of maturity in the software and the sheer amount of software involved in getting the experiments in place and maintaining a workable configuration over a long period of time. Hopefully this situation will improve as the software matures, more sites engage in grid-based computing, and the software infrastructure is more widely used.

Part of the point of conducting these experiments was to show that using geographically distributed resources under a grid framework through the control of the library routine could lead to an improved time to solution for a user. As such the results, for this modest number of experiments, show that performing a Grid-based computation can be a reasonable undertaking. When solving dense matrix problems, we have $O(n^2)$ data to move and $O(n^3)$ operations to perform. So the fact that we are dealing with geographically distributed systems is not a major factor in performance when the data has to be moved across slow networks. If the problem characteristics were different, the situation might not be the same in terms of Grid-feasibility.

Future work will involve the development of a system that implements a migration system if the time to solution violates the performance contract, and a mechanism to deal with fault tolerance.

4. Cactus Experiment

We summarize the major findings resulting from the Cactus experiment to date, as follows:

- *Initial findings suggesting that Grid-enabled frameworks can allow significant applications to be executed efficiently in Grid environments*. The set of modules “thorns” developed for Cactus to support adaptation to heterogeneous resources, contract monitoring, migration, and the like have been proven effective in preliminary studies, albeit with considerable manual guidance. In particular, we have demonstrated that we can (a) run large Cactus computations across distributed heterogeneous resources and (b) achieve dynamic resource selection and migration.
- *Performance model for Cactus*. We have developed a detailed performance model for Cactus and validated this model in a variety of settings.
- *An initial instantiation of part of the GrADS architecture*. We have developed the interfaces, protocols, and components required to support dynamic resource discovery and reallocation.

These three developments put us in a position to be able to address our two overarching goals:

1. Exploring more quantitatively the effectiveness of our dynamic adaptation techniques and
2. Exploring the application of tools developed by other GrADS participants, initially, in particular, the contract monitoring techniques being developed at UIUC.

5. MacroGrid

The MacroGrid has been in continuous operation over the past year. A significant amount of effort was put into building communication channels and putting in place organization and procedures for testbed operation. Because of this, we were successful in both broadening the size of the testbed and the range of experiments that could be run. Progress was demonstrated in that bringing up the second application (Cactus) was significantly easier then the initial ScaLAPACK experiment.

The current testbed maintains information about resource and software availability. However, we do not have a rigorously defined notion of what it means for a resource to be MacroGrid compliant, nor do we have a means for checking compliance. This is particularly problematic in the case of installed software. We found that in practice this represents a significant obstacle to conducting MacroGrid experiments. An automated script verifying the compliance and notifying either the MacroGrid or local administrator of failure to comply would have been extremely beneficial. We plan to develop this capability in the coming year.

We have found the virtual organization structure provided by the MDS 2.0 model to be effective in delivering information about the MacroGrid execution environment to both GrADS users and applications. We believe that this will be a generally useful approach for managing the complexity of Grid environments. We plan to increase the use of MacroGrid information services in GrADS tools and services.

Due do some transient problems in the GrADS virtual organizational server, we found that reliability became an issue in testbed operation. These problems were overcome by the creation of a second GrADS server at UIUC. This server is not a mirror, but exploits the use of the dynamic registration protocol to create a second server with the exact same information content as the initial server at ISI. The creation of this second server has eliminated single point of failures within the MacroGrid.

Another challenge faced in the operation of the testbed was to get the system administrators to keep up-to-date information about software installations on each computer. While system administrators were quick to follow through with upgrades and patches, it was sometimes quite difficult to get them to update the registration information, which has to be done manually. Administrators also made mistakes in copying or customizing information into this process, introducing errors in the GIIS server. We are investigating methods to automatically search for software installations and insert appropriate information into the GrADS system-wide information servers. Software information is also presently not used by runtime tools. This is partly a communications issue, and will be addressed in the coming year.

Finally, operation of the MacroGrid has resulted in identifying some scalability issues associated with the technology used to implement the GrADS server. These have been addressed in the design of MDS 2.1, which we plan to roll out in the MacroGrid in the coming year.

6. MicroGrid

The major findings from the MicroGrid project are:

- Basic MicroGrid tools can achieve high fidelity simulation of Grid applications (NAS NPB, Cactus), delivering on both external (total run time) and internal (detailed application progress).
- Scalable network emulation is a key problem for Grid modeling scaling, and we have innovative approaches that we believe will lead to solutions.
- The MicroGrid tools are now ready for use with significantly larger applications, such as the ScaLAPACK and Cactus programs being used in other parts of the GrADS effort. We will exploit the adaptive versions of these programs as workloads as they become available.