Project Findings

During the initial nine-month reporting period (10/1/99-6/30/00), GrADS research has focused on five inter-institutional efforts: *Program Execution System (PES)* led by Fran Berman, *Program Preparation System (PPS)* led by Ken Kennedy, *Macro Testbed* led by Carl Kesselman, *MicroGrid* led by Andrew Chien, and *Applications* led by Dennis Gannon. A brief summary of the findings of each subproject is presented in the following sections.

1. Program Execution System (PES)

At present, we are building the initial system prototype. We have performed several experiments to ensure that MPICH-G works correctly across research sites and have been installing a uniform software base and developing the modules which form the execution scenario described under "Project Activities." In addition, we have been exploring various contract specification, runtime monitoring, and violation detection algorithms to identify the parameters that we can reliably measure and reason about. Our goal is to have the initial prototype running by the end of the summer and to be able to assess and evaluate findings from this work in the fall.

2. Program Preparation System (PPS)

Through our experimentation and analysis of the ScaLAPACK LU routine, it became clear that (1) it would be difficult to automatically derive a parameterizable DAG model (suitable for arbitrary problem sizes) that reflects the serialization among computational units, and (2) given such a model, it would be non-trivial to use it to decide how to map the computation onto resources to obtain good load balance over time. This led us to conclude that library or application designer knowledge of algorithmic serialization should be represented explicitly and used to guide program execution system decision making. Having found top-down conceptualization of a solution to these requirements very hard, we have shifted our short-term focus to developing actual interfaces to pass expert knowledge about a library routine to the GrADS components.

3. Macro Testbed

Our initial testbed deployment confirmed several early conjectures about the ease of integrating Grid software services, and exposed important requirements for the development environment that is necessary to support the GrADS effort. As an initial experiment, the GrADS project team has focused on developing and deploying an adaptive, Grid-enabled version of ScaLAPACK. To do so, Globus, MPICH-G, and the Network Weather Service had to be deployed, maintained, and integrated so that they worked seamlessly across GrADS execution sites. While other researchers have used these Grid services piecewise and in pairs, the macro-testbed requires them to work in unison while they are administered by separate organizations. Both Globus and the Network Weather Service have been designed to function across multiple administrative domains, but our experience in deploying them together to support GrADS confirms their

utility and compatibility. In addition, the ScaLAPACK experiment requires that the ScaLAPACK library work with MPICH-G. After an initial integration effort that focused mainly on configuration problems, ScaLAPACK and MPICH-G have been successfully integrated and deployed. The lesson learned from these two integration and deployment efforts are being incorporated into the next releases of Globus (1.1.3) and the Network Weather Service (beta-7). These new releases will result in a tighter integration of the Grid services that they provide.

The initial macro-testbed has exposed the need for a Grid software release and integration management policy. In response to the needs of GrADS researchers, Globus, Network Weather Service, MPICH-G, ScaLAPACK, AppLeS, and Autopilot researchers are modifying and enhancing their individual services and software components. Often, these enhancements must be deployed in order to enable testbed-wide functionality or stability. While we have been able to test the individual components piecewise, our initial experiences indicate that a release and testing plan will be necessary as the project progresses.

4. MicroGrid

An initial set of MicroGrid tools that allows researchers to run arbitrary Grid applications on virtual Grid resources has been designed, implemented, and validated. These tools enable the use of Globus applications without change by *virtualizing* the execution environment, providing the illusion of a virtual Grid. Thus, experimentation with a wide variety of existing Grid applications is feasible. Additionally, these tools manage heterogeneous models in the virtual grid, using a global virtual time model to preserve simulation accuracy. The MicroGrid also provides basic resource simulation models for computing, memory, and networking. These elements have been implemented and validated at three levels: micro-benchmarks, parallel benchmarks, and single application. Experiments on the NAS Parallel Benchmark suite ultimately matched within 1 to 12%, while maintaining high execution efficiency.

To achieve the larger vision of Grid simulation, significant advances in scalability, precision of modeling, and network traffic modeling must be achieved.

5. Applications

Two initial projects involving the Cactus code have been launched. The Illinois team has installed the Cactus Code on Solaris and on a Linux cluster of PCs. They have successfully tested two Cactus thorns on the cluster: Wavetoy and Einstein. Tests were conducted for both stand-alone and Globus MPICH modes, using various grid sizes and iterations to produce 3-D simulation data. In addition, they experimented with Autopilot sensors to capture the arrival rates of file read requests generated by Cactus Wavetoy. The have also used Autodriver, a graphical component of Autopilot that displays sensor/actuator contents, to visualize the read behavior of the application.

The second initial effort has involved the University of Chicago working with the Cactus team to design a web-portal for remote execution of the Cactus framework and resulting applications. This portal incorporates the Globus Security infrastructure and allows the authenticated use of the ability to configure a simulation, manage parameter settings, launch and control the application, and view output results, all from the web interface. The GrADS team will work closely with this portal group to integrate our tools with this web-framework.