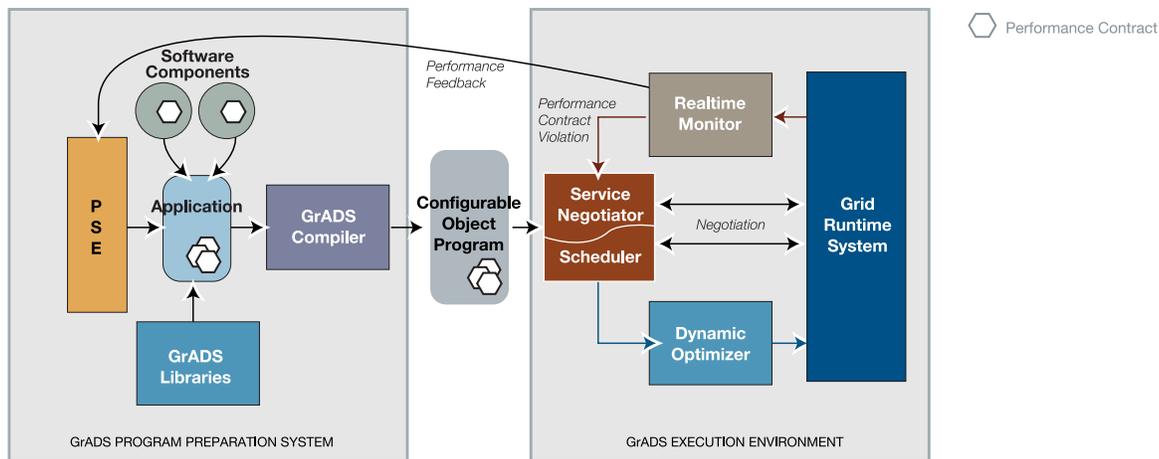


# GRADS

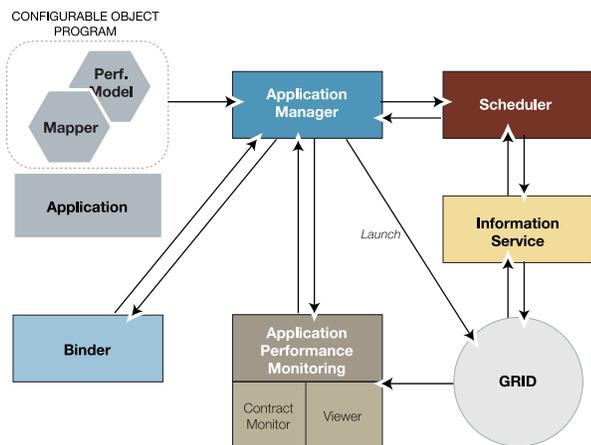
## GRID APPLICATION DEVELOPMENT SOFTWARE

The goal of the Grid Application Development Software (GRADS) project is to simplify distributed heterogeneous computing in the same way that the World Wide Web simplified information sharing over the Internet. The GRADS project will explore the scientific and technical problems that must be solved to make grid application development and performance tuning for real applications an everyday practice.

### GRADS ARCHITECTURE



### GRADSOFT PROGRAM EXECUTION



### APPLICATIONS RUNNING ON GRADSOFT

**ScaLAPACK LU Solver** – A highly developed SPMD code for solving dense systems of equations via LU factorization. GrADSoft uses a simulation model of the code to generate a near-optimal execution schedule.

**Cactus** - An application framework primarily intended for compute intensive high-energy physics and astrophysics applications.

**FastA** – Protein and genome sequence alignment application. GrADSoft schedules execution using distributed, possibly replicated databases at Grid locations.

**GrADSAT** - A parallel solver for satisfiability instances generated from digital circuits and other industrial applications. Using GrADSoft, a set of distributed resources is used to solve satisfiability instances faster.

**Smith-Waterman** - Protein and genome sequence alignment application used to find homologies within the entire database set (running all databases as queries against all others). Uses distributed adaptive techniques to ensure fault tolerance and earliest completion time.

**TOMCATV** - A vectorized mesh generation HPF application. GrADSoft uses a Mapper Builder Tool to map the HPF application onto a suitable set of Grid resources.

### PRINCIPAL INVESTIGATORS

RICE UNIVERSITY  
Ken Kennedy · Keith Cooper ·  
John Mellor-Crummey · Linda Torczon

UNIVERSITY OF CALIFORNIA - SANTA BARBARA  
Richard Wolski

UNIVERSITY OF ILLINOIS AT URBANA -  
CHAMPAIGN  
Daniel Reed

INDIANA UNIVERSITY  
Dennis Gannon

ARGONNE NATIONAL LAB / UNIVERSITY OF CHICAGO  
Ian Foster

UNIVERSITY OF SOUTHERN CALIFORNIA  
INFORMATION SCIENCES INSTITUTE  
Carl Kesselman

UNIVERSITY OF CALIFORNIA - SAN DIEGO  
Francine Berman · Andrew Chien

UNIVERSITY OF HOUSTON  
Lennart Johnsson

UNIVERSITY OF TENNESSEE  
Jack Dongarra

SPONSORED BY



NSF  
Next Generation Software (NGS)

GrADS

@ <http://www.hipersoft.rice.edu/grads/>

Advances in networking technologies will soon make it possible to use the global information infrastructure in a qualitatively different way--as a computational as well as an information resource. As described in the recent book "The Grid: Blueprint for a New Computing Infrastructure," this "Grid" will connect the nation's computers, databases, instruments, and people in a seamless web of computing and distributed intelligence, that can be used in an on-demand fashion as a problem-solving resource in many fields of human endeavor---and, in particular, for science and engineering.

The availability of Grid resources will give rise to dramatically new classes of applications, in which computing resources are no longer localized, but distributed, heterogeneous, and dynamic; computation is increasingly sophisticated and multidisciplinary; and computation is integrated into our daily lives, and hence subject to stricter time constraints than at present. The impact of these new applications will be pervasive, ranging from new systems for scientific inquiry, through computing support for crisis management, to the use of ambient computing to enhance personal mobile computing environments.

To realize this vision, significant scientific and technical obstacles must be overcome. Principal among these is usability. Because the Grid will be inherently more complex than existing computer systems, programs that execute on the Grid will reflect some of this complexity. Hence, making Grid resources useful and accessible to scientists and engineers will require new software tools that embody major advances in both the theory and practice of building Grid applications.

The goal of the Grid Application Development Software (GrADS) Project is to simplify distributed heterogeneous computing in the same way that the World Wide Web simplified information sharing over the Internet. To that end, the project is exploring the scientific and technical problems that must be solved to make it easier for ordinary scientific users to develop, execute, and tune applications on the Grid. In this paper we describe the vision and strategies underlying the GrADS project, including the base software architecture for Grid execution and performance monitoring, strategies and tools for construction of applications from libraries of Grid-aware components, and development of innovative new science and engineering applications that can exploit these new technologies to run effectively in Grid environments.

GrADS will foster research and technology transfer programs contributing to revolutionary new ways of utilizing the global information infrastructure as a platform for computation, changing the way scientists and engineers solve their everyday problems.

At the University of Tennessee the primary goals of our effort in algorithms and libraries are: 1) To develop a new generation of algorithms and software libraries needed for the effective and reliable use of (wide area) dynamic, distributed and parallel environments, and 2) To validate the resulting libraries and algorithms on important scientific and engineering applications.

To consistently obtain high performance in Grid environments will require advances in both algorithms and their supporting software. These same advances may help with other advanced computing environments, including distributed systems, heterogeneous systems, clusters of workstations, distributed shared memory machines, and the proposed petaflops architectures, will require advances in algorithms and supporting software. Some of the challenges in this arena have already been encountered. For example, to make effective use of current high-end machines, the software must manage both communication and the memory hierarchy. This problem has been attacked with a combination of compile-time and run-time techniques. On the Grid, the increased scale of computation, depth of memory hierarchies, range of latencies, and increased variability in the run-time environment will make these problems much harder.

To address these issues, we must rethink the way that we build libraries. The issues to consider include software architecture, programming languages and environments, compile versus run-time functionality, data structure support, and fundamental algorithm design. Among the challenges that we foresee are:

- Library software must support performance optimization and algorithm choice at run time.
- The architecture of software libraries must facilitate efficient interfaces to a number of languages, as well as effective support for the relevant data structures in those languages.
- New algorithmic techniques will be a prerequisite for latency tolerant applications.
- New scalable algorithms that expose massive numbers of concurrent threads will be needed to keep parallel resources busy and to hide memory latencies.