Software Technology for Problem Solving on Computational Grids

An Overview of the CGrADS STC Proposal

Ken Kennedy

Center for High Performance Software

Rice University

http://hipersoft.rice.edu/stc_site_visit/talks/CGrADSOverview.pdf

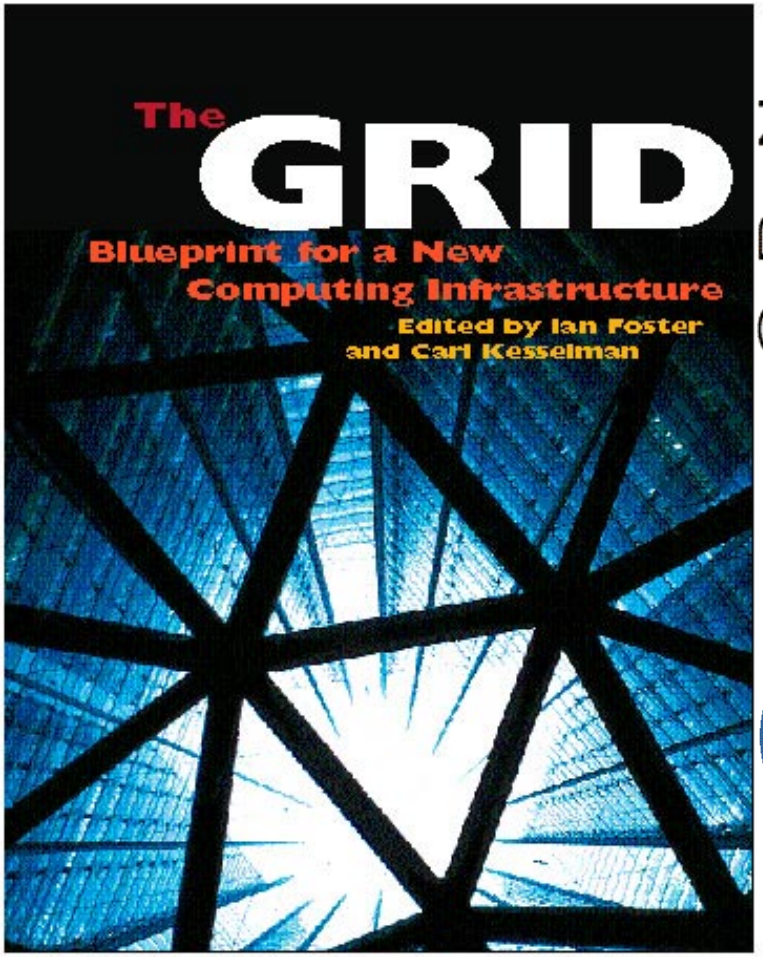
GrADS

Grid Application Development Software Project

Resources

- Site Visit Web Site
  - http://hipersoft.rice.edu/stc_site_visit/
  - Biographies of participants
    - With pictures
  - Viewable versions of slides
    - Printed versions in your packet
  - Letters of support
- GrADS Web Site
  - http://hipersoft.rice.edu/grads/
  - Planning reports and technical reports

Grids are "Hot"



GriPhyN
Data Intensive Science

NEESgrid

Computational
Data
Information
Access
Knowledge

DISCOM

SinRG

European GRID Forum

APGrid

GLOBAL GF

DataGRID

EUROGRID

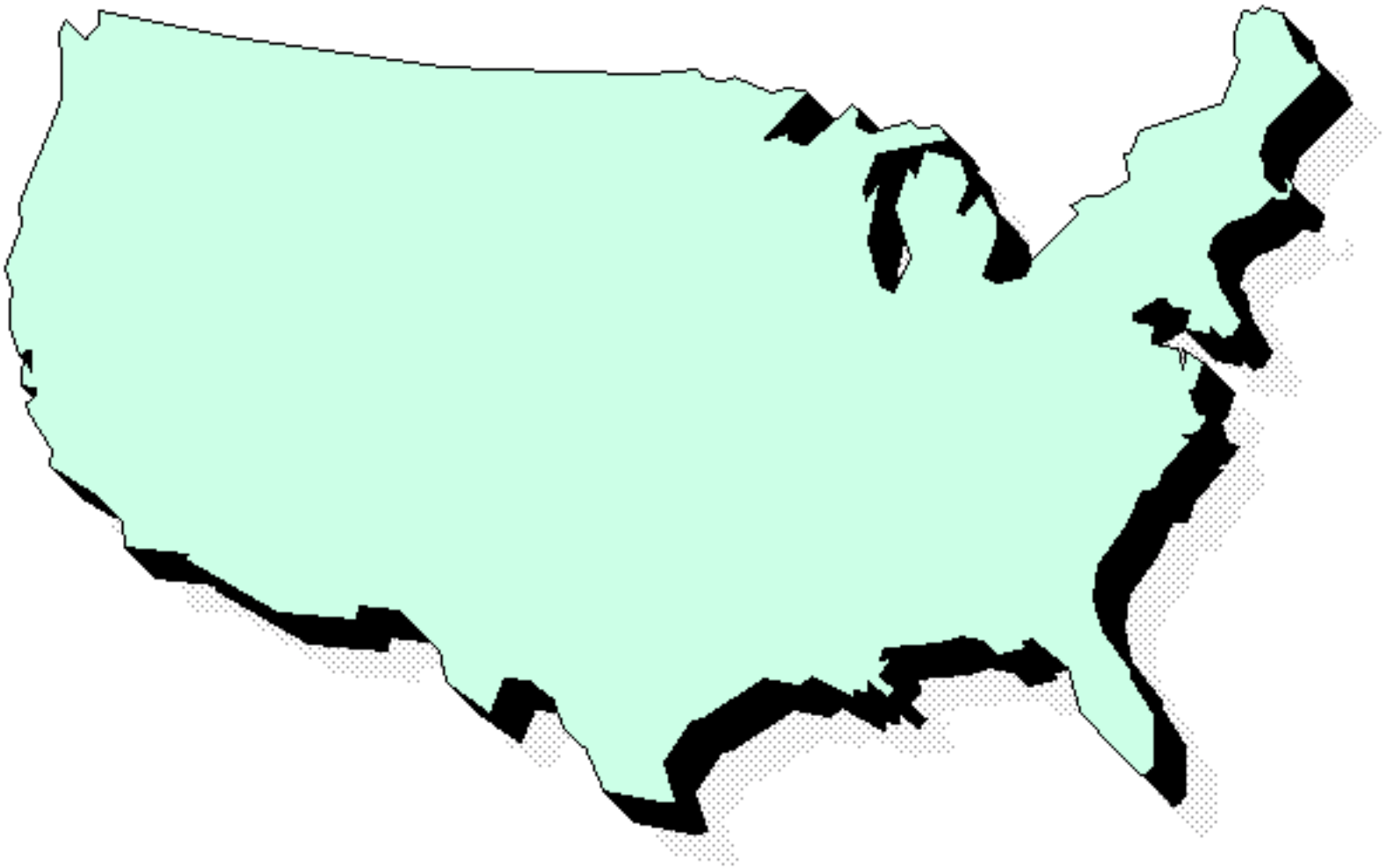
TeraGrid

GrADS
Grid Application Development Software Project

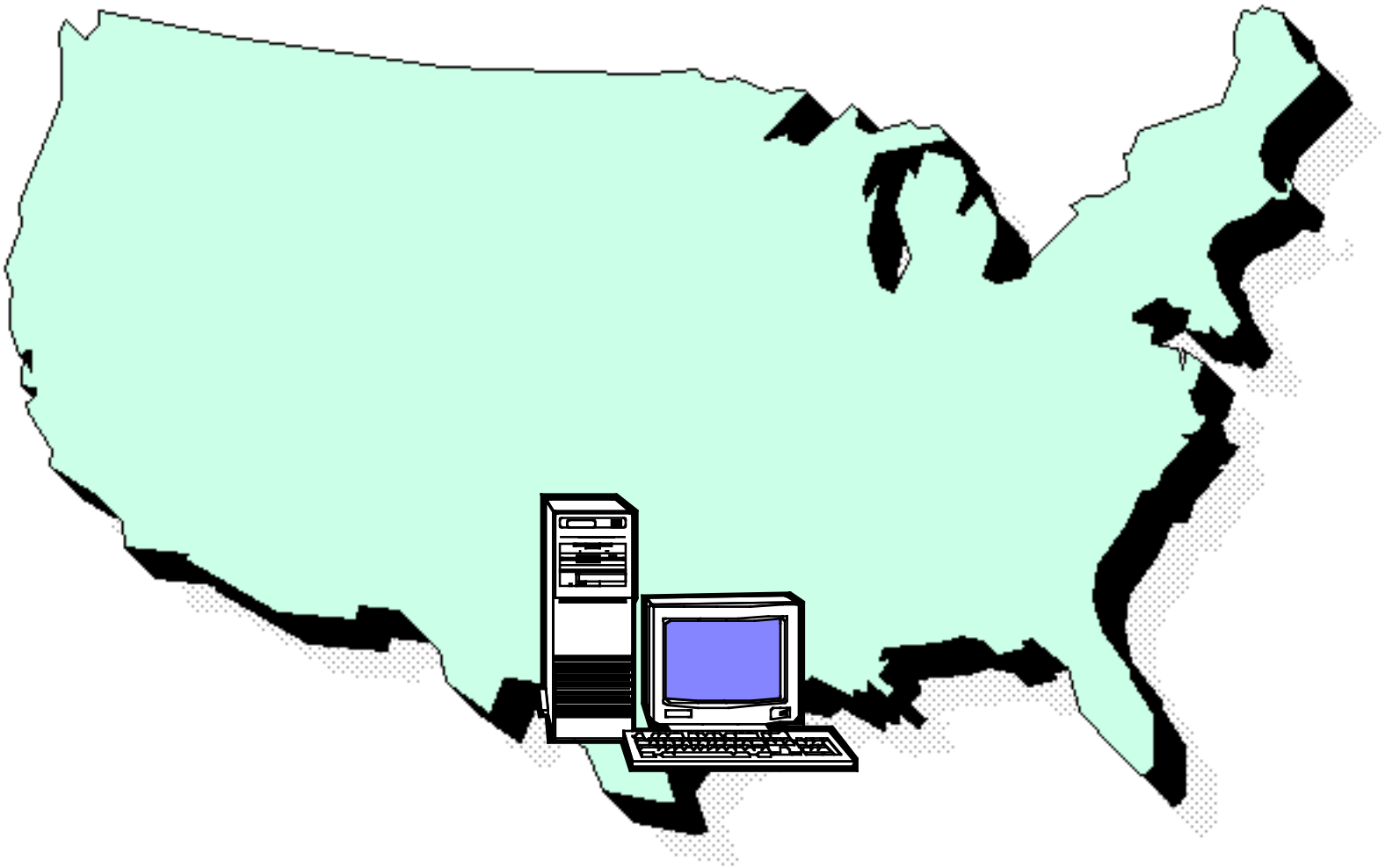
But What Are They?

- Collection of computing resources
  - —Varying in power or architecture
  - —Potentially dynamically varying in load
    - Unreliable?
  - —No hardware shared memory
- Interconnected by network
  - —Links may vary in bandwidth
  - —Load may vary dynamically
- Distribution
  - —Across room, campus, state, nation, globe
- Inclusiveness
  - —Distributed-memory parallel computer is a degenerate case

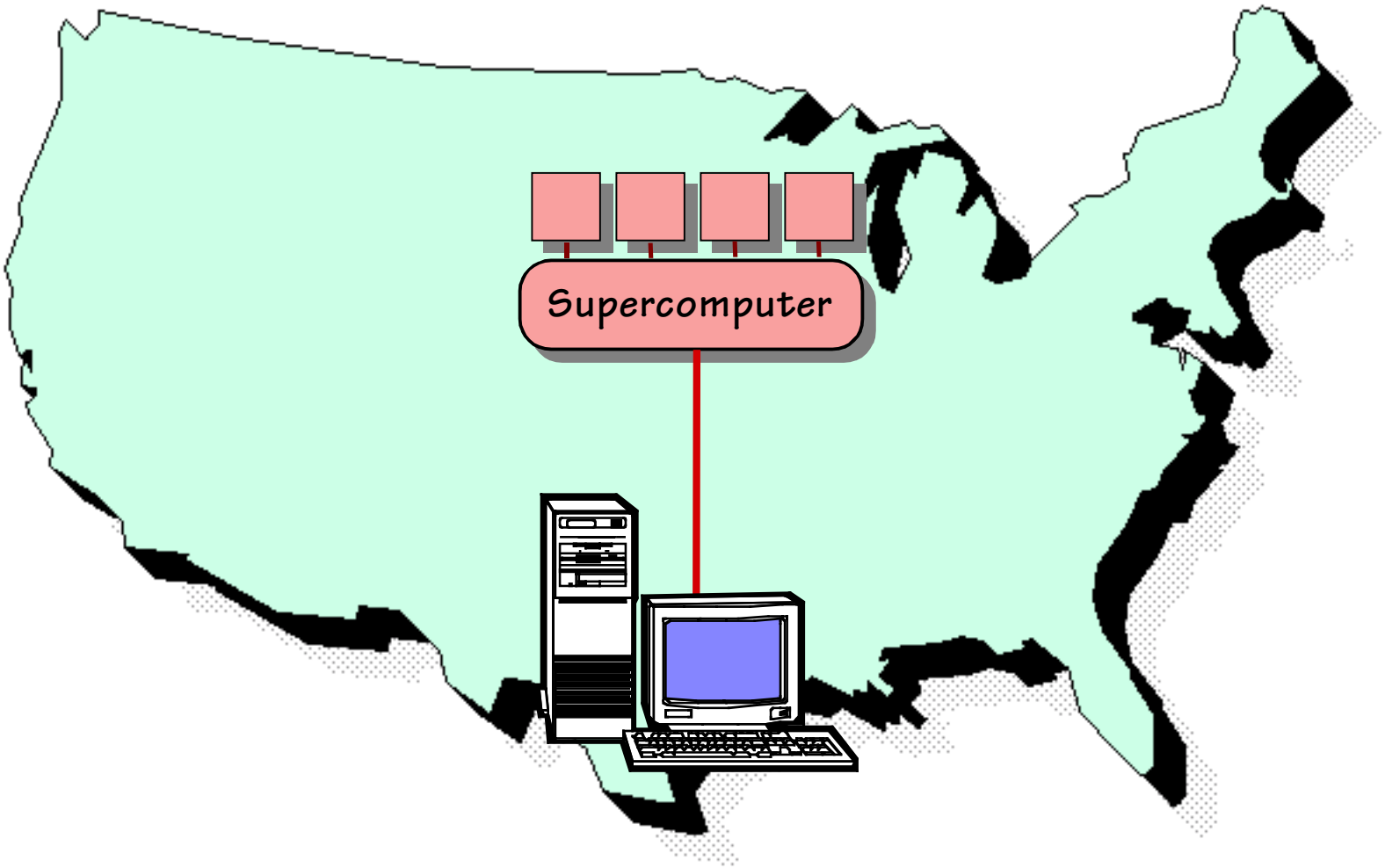
National Distributed Problem Solving



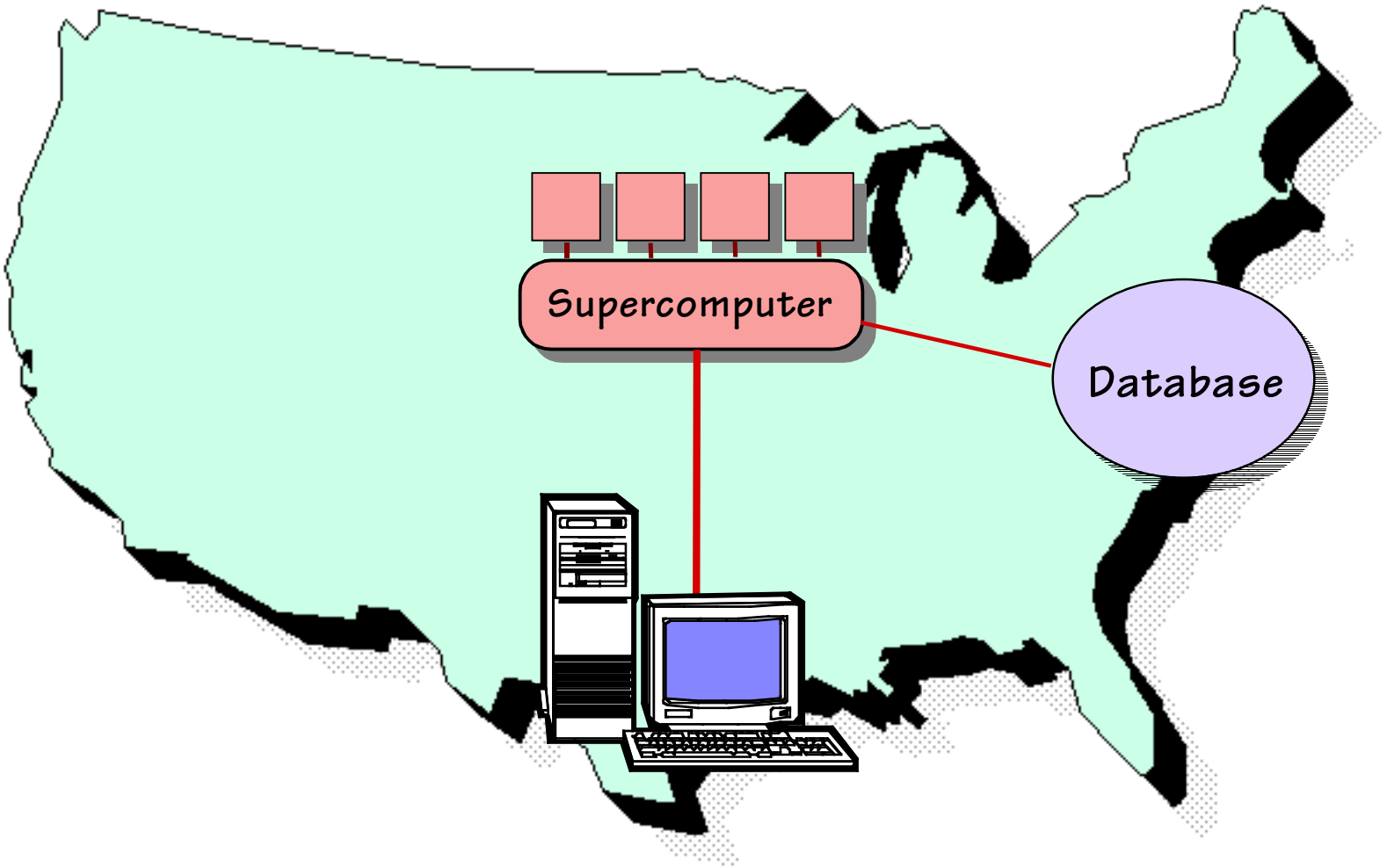
National Distributed Problem Solving



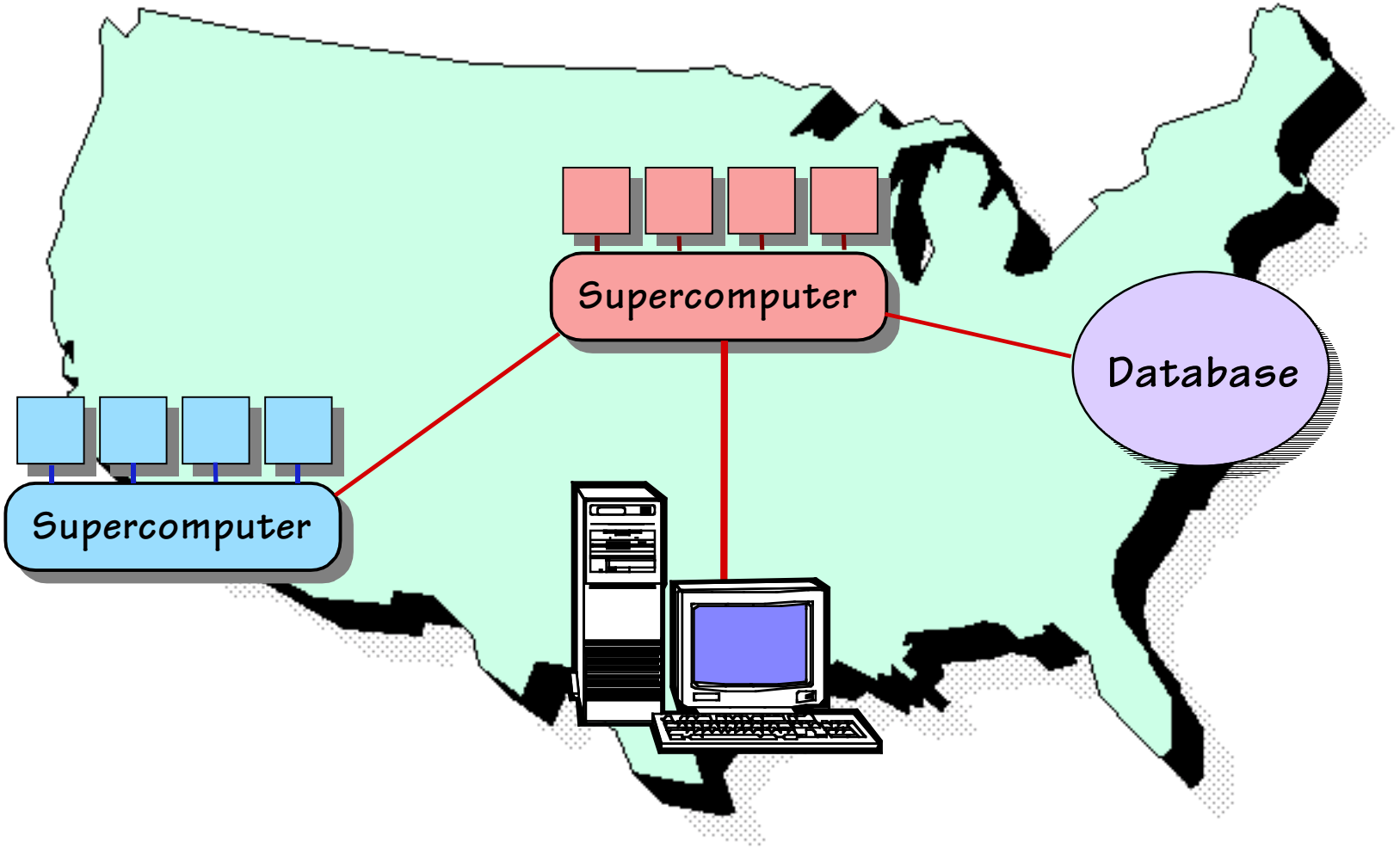
National Distributed Problem Solving



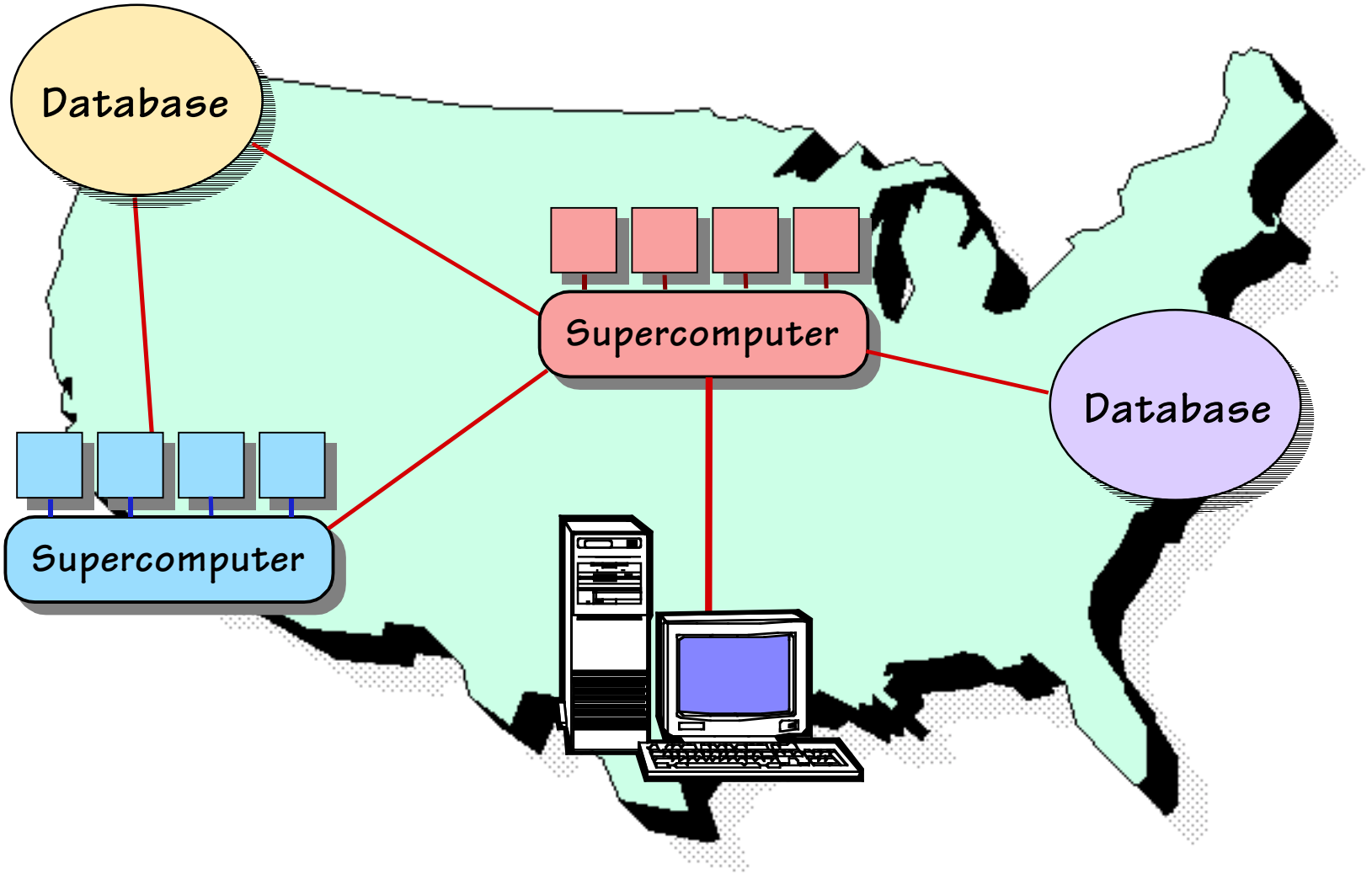
National Distributed Problem Solving



National Distributed Problem Solving



National Distributed Problem Solving



A Software Grand Challenge

- Software Support for Application Development on Grids
  - Goal: Make it easy to develop applications for and solve problems on the Grid
- Challenges:
  - Presenting a high-level application development interface
    - If programming is hard, the Grid will not not reach its potential
  - Designing and constructing applications for adaptability
  - Late mapping of applications to Grid resources
  - Monitoring and control of performance
    - When should the application be interrupted and remapped

Today: Globus

- Developed by Ian Foster and Carl Kesselman
  - Grew from the I-Way (SC-95)
- Basic Services for distributed computing
  - Resource discovery and information services
  - User authentication and access control
  - Job initiation
  - Communication services (Nexus and MPI)
- Applications are programmed by hand
  - Many applications
  - User responsible for resource mapping and all communication
    - Existing users acknowledge how hard this is

CGrADS Goal and Background

- Goal:
  - Design and build programming systems for the Grid that broaden the community of users who can develop and run applications in this complex environment
- The GrADS Project, sponsored by NSF NGS
  - Three-year effort to carry out initial studies on Grid programming systems (1999-2002)
    - Included all current CGrADS PIs + Dennis Gannon (Indiana)
    - http://hipersoft.cs.rice.edu/grads/
  - Developed concepts underlying the CGrADS vision
    - Initial application studies
    - Initial GrADSoft design

Research Strategy

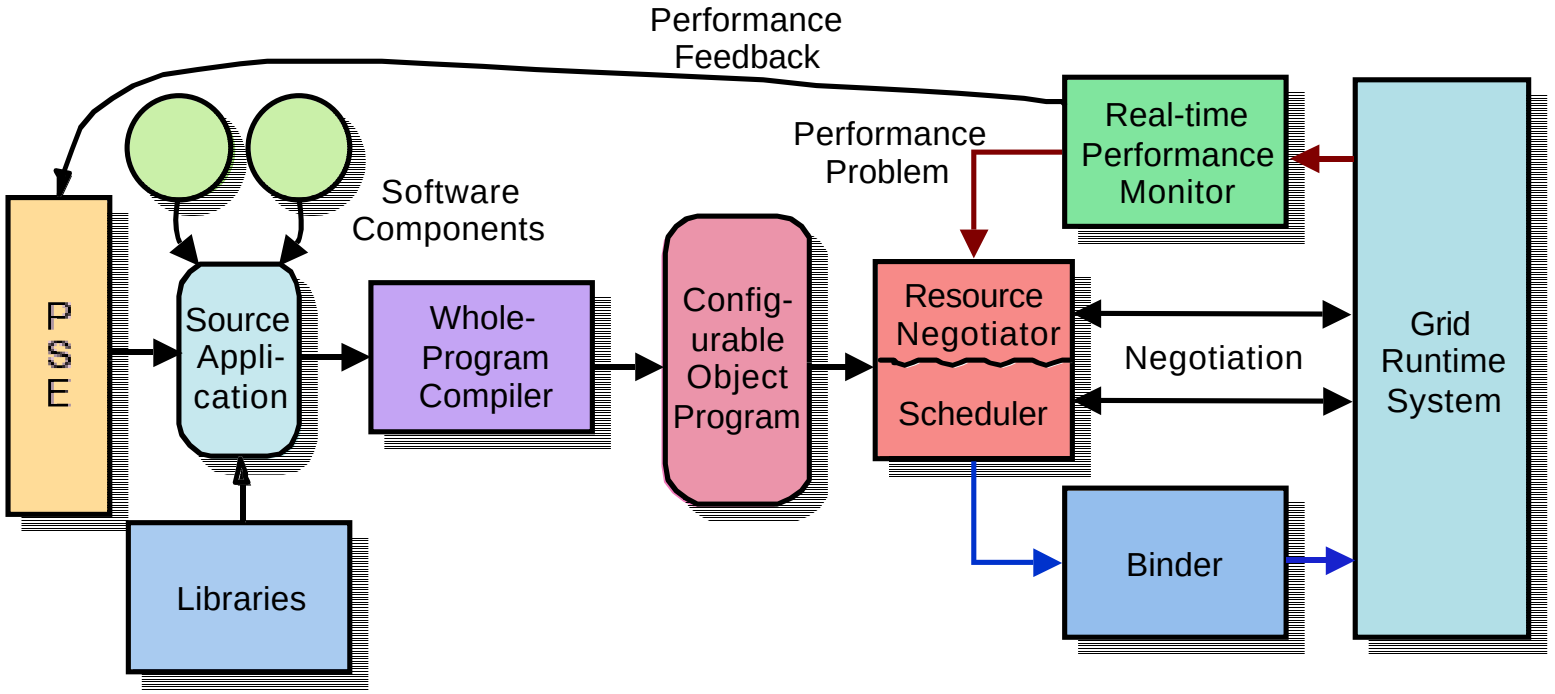
- Applications Studies
  - Prototype a series of applications using components of envisioned execution system
    - ScaLAPACK and Cactus demonstration projects
- Move from Hand Development to Automated System
  - Identify key components that can be isolated and built into a Grid execution system
    - e.g., prototype reconfiguration system
  - Use experience to elaborate design of software support systems
- Experiment
  - Use testbeds to evaluate results and refine design

Conclusions: What is Needed

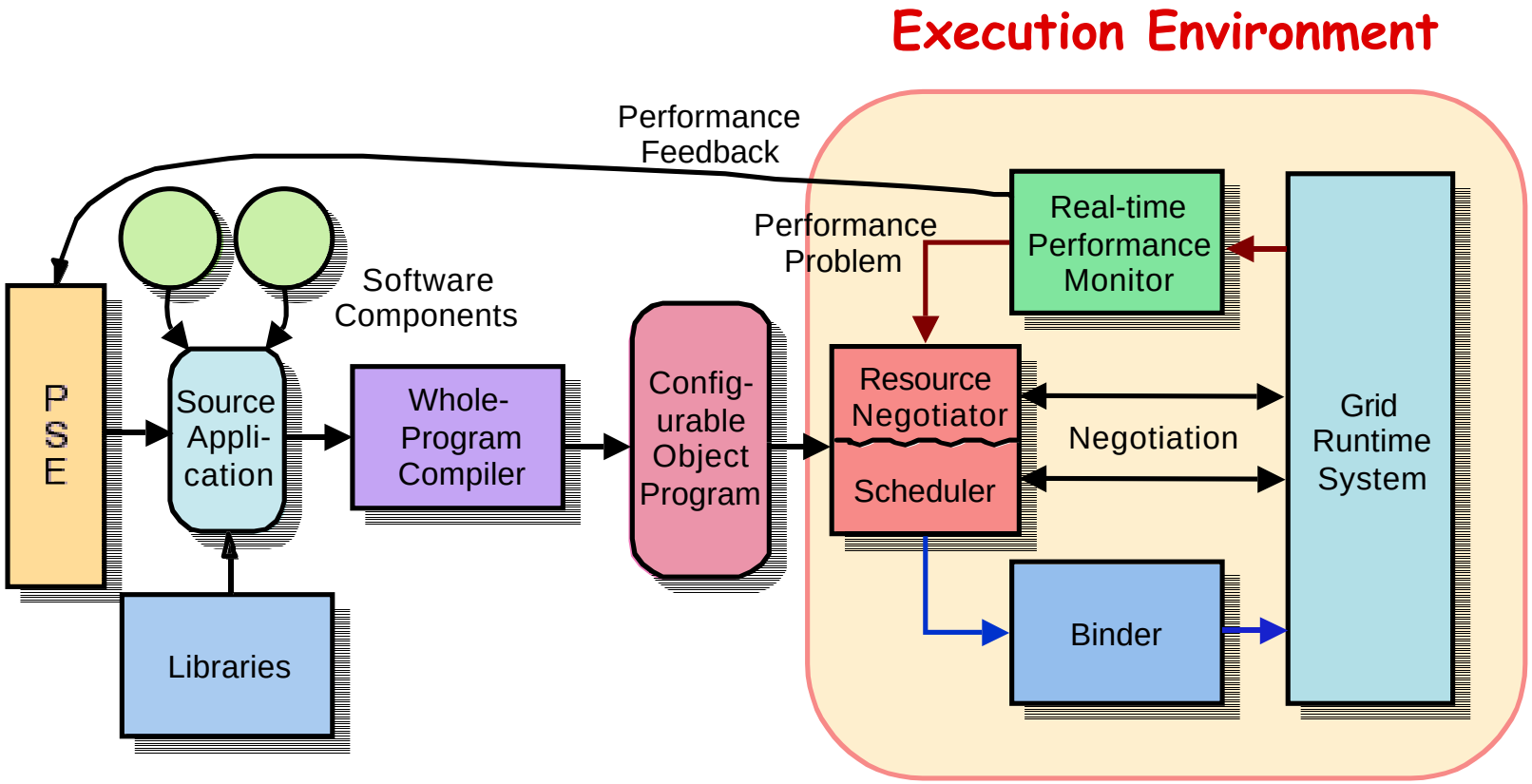
- Execution infrastructure for adaptive execution
  - Automatic resource location and execution initiation
  - Dynamic configuration to available resources
  - Performance monitoring and control strategies
    - deep integration across compilers, tools, and runtime systems
    - performance contracts and dynamic reconfiguration
- Abstract Grid programming models and easy-to-use programming interfaces
  - Problem-solving environments
- Robust reliable numerical and data-structure libraries
  - Predictability and robustness of accuracy and performance
  - Reproducibility and fault tolerance

GrADSoft Architecture

- Goal: reliable performance on dynamically changing resources



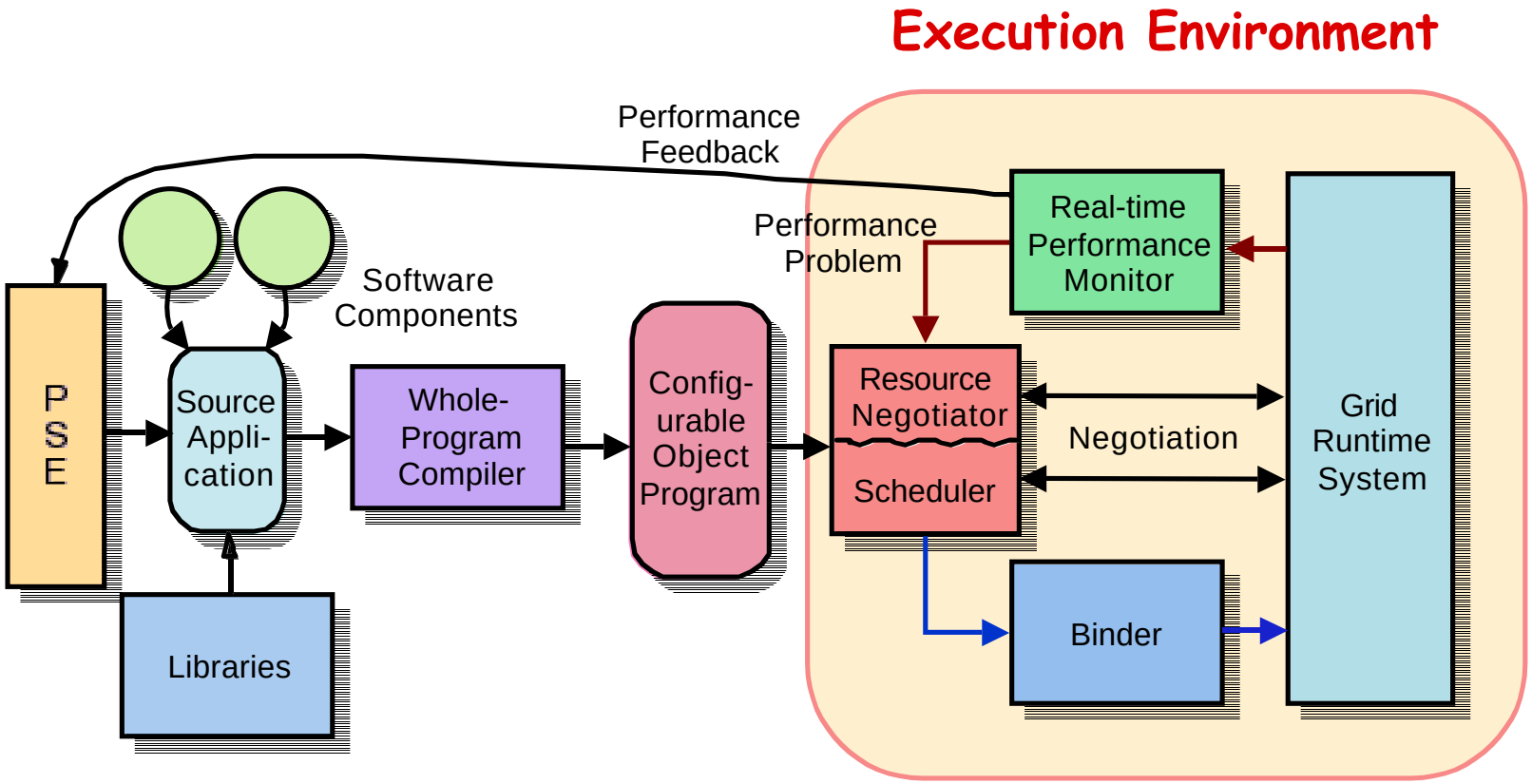
GrADSoft Architecture



Configurable Object Program

- Representation of the Application
  - Supporting dynamic reconfiguration and optimization for distributed targets; includes
    - Program intermediate code
    - Mapping strategy and performance model(s)
    - Historical information (run profile to now)
- Mapping Strategy
  - Defines required resources and affinities to specialized resources
  - Given a set of resources, maps computation to those resources
    - Optimal performance, given all requirements met
- Performance Model
  - Given a set of resources and mapping, estimates performance
  - Serves as objective function for Resource Negotiator/Scheduler

GrADSoft Architecture



Execution Cycle

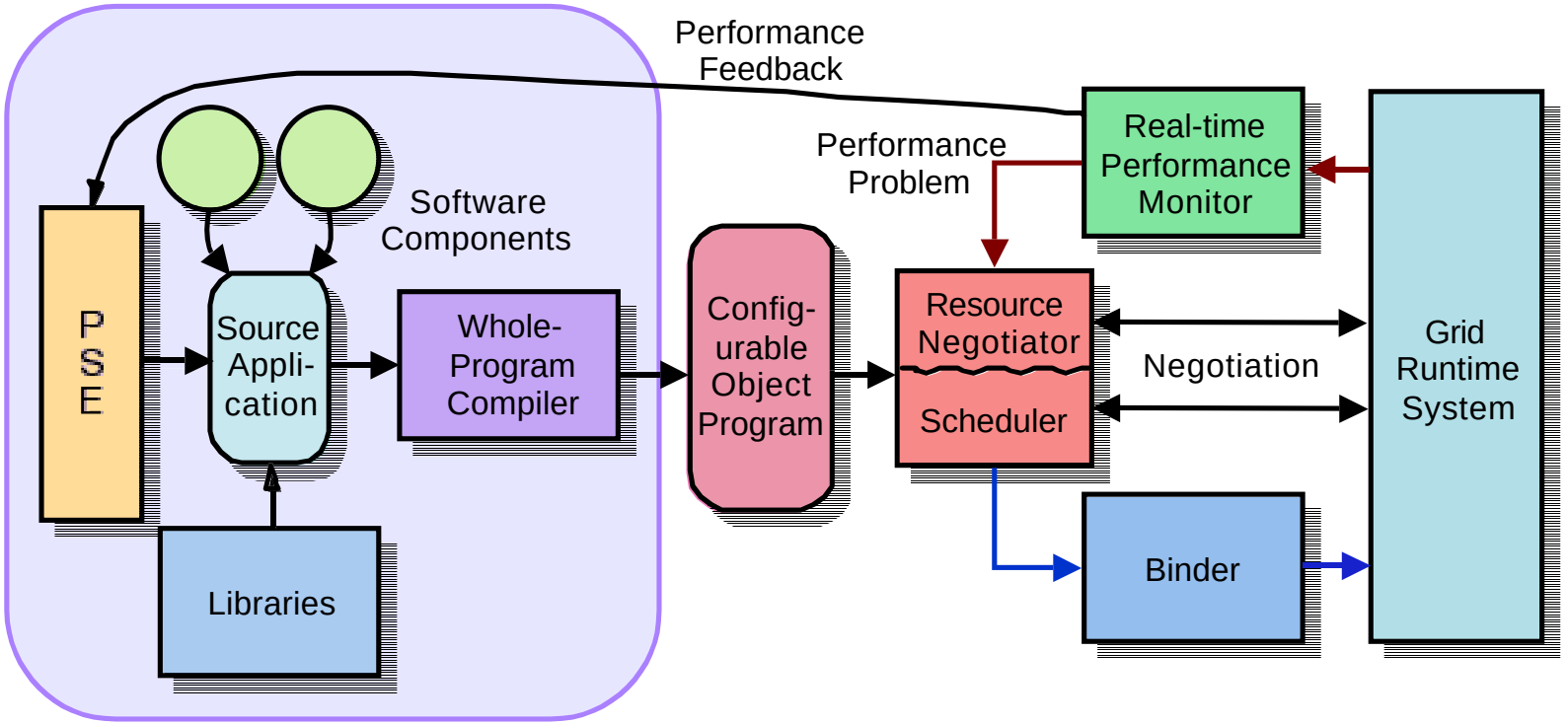
- Configurable Object Program is presented
  - Space of feasible resources must be defined
  - Mapping strategy and performance model provided
- Resource Negotiator solicits acceptable resource collections
  - Performance model is used to evaluate each
  - Best match is selected and contracted for
- Execution begins
  - Binder tailors program to resources
    - Carries out final mapping according to mapping strategy
    - Inserts sensors and actuators for performance monitoring
- Contract monitoring is performed continuously during execution
  - Soft violation detection based on fuzzy logic

Performance Contracts

- At the Heart of the GrADS Model
  - Fundamental mechanism for managing mapping and execution
- What are they?
  - Mappings from resources to performance
  - Mechanisms for determining when to interrupt and reschedule
- Challenge:
  - When should a contract be violated?
    - Strict adherence balanced against cost of reconfiguration

GrADSoft Architecture

Program Preparation System



Program Preparation System Overview

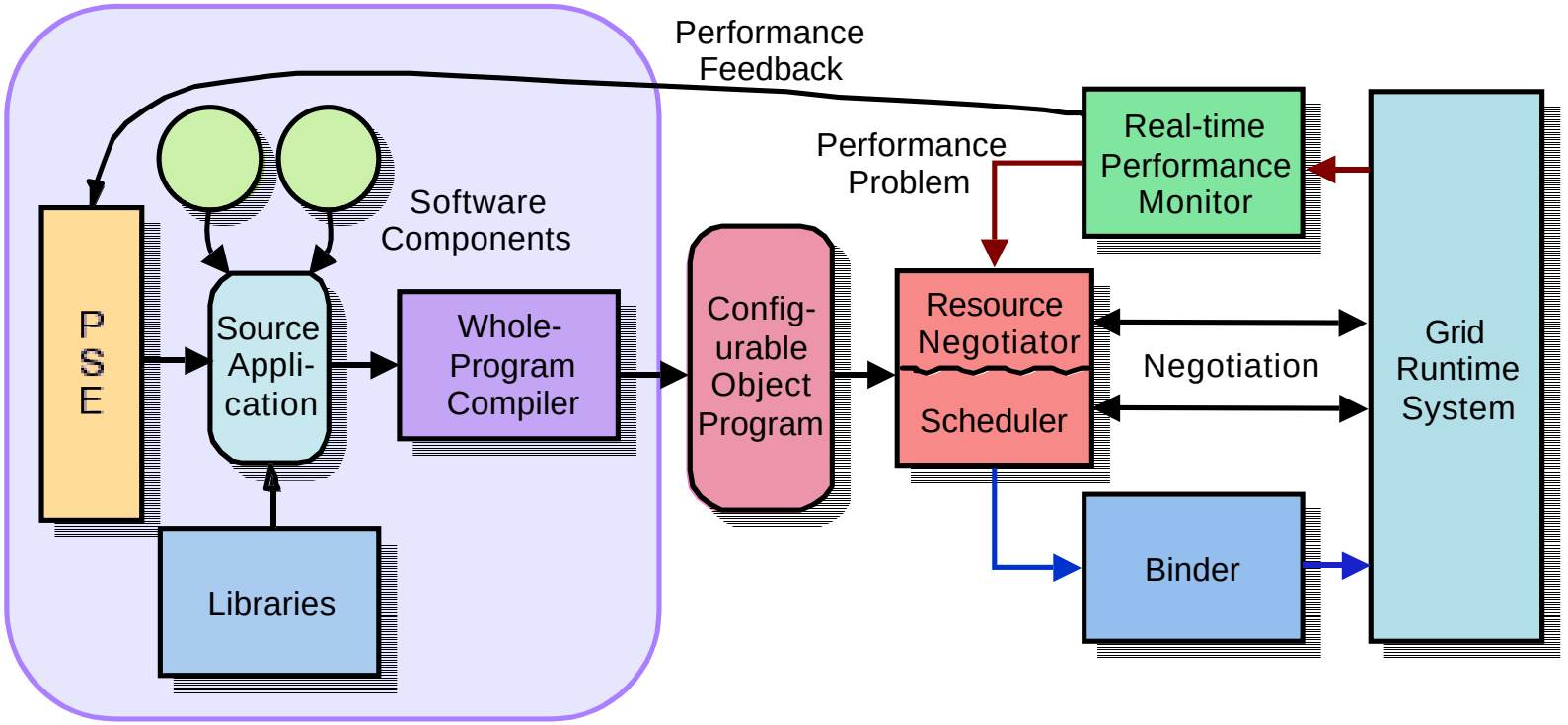
- Libraries Coded by Professionals:
  - Included mapping strategies and cost models
  - High level knowledge of integration strategies
- Whole Program Compiler:
  - Produces Configurable Object Program
    - Integrated mapping strategy and performance model
  - Performance enhanced through context-dependent variants
  - Context includes potential execution platform
- Binder:
  - Performs final pre-launch processing
  - Implements mapping and chooses machine-specific variants
  - Inserts performance monitoring sensors and actuators

Software Support for Adaptivity

- Needed: tools to support the construction of adaptive applications
- Performance modeling
  - Challenge: synthesis and integration of performance models
    - Combine expert knowledge, trial execution, and scaled projections
- Support for reconfiguation
  - Mapping and remapping
- Latency tolerance
- Fault tolerance

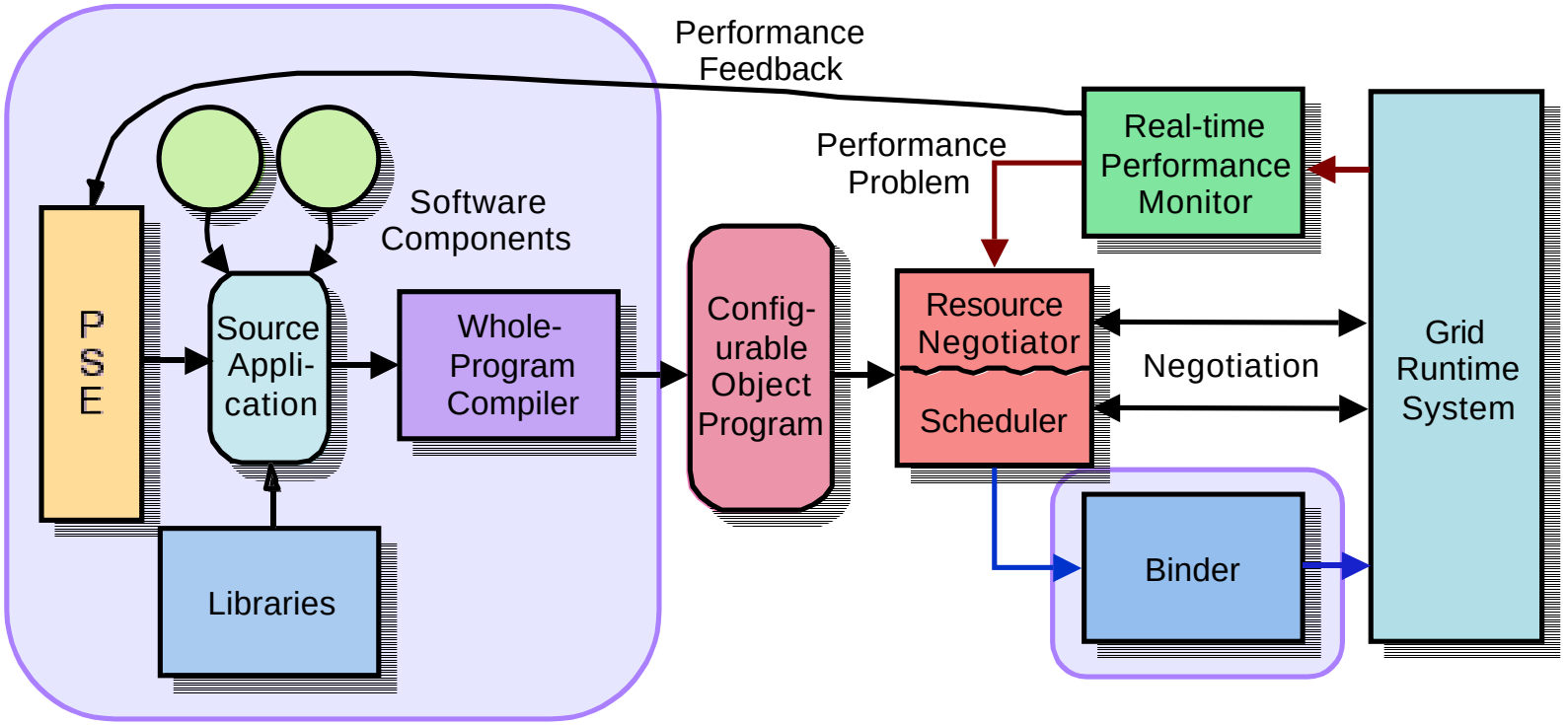
GrADSoft Architecture

Program Preparation System

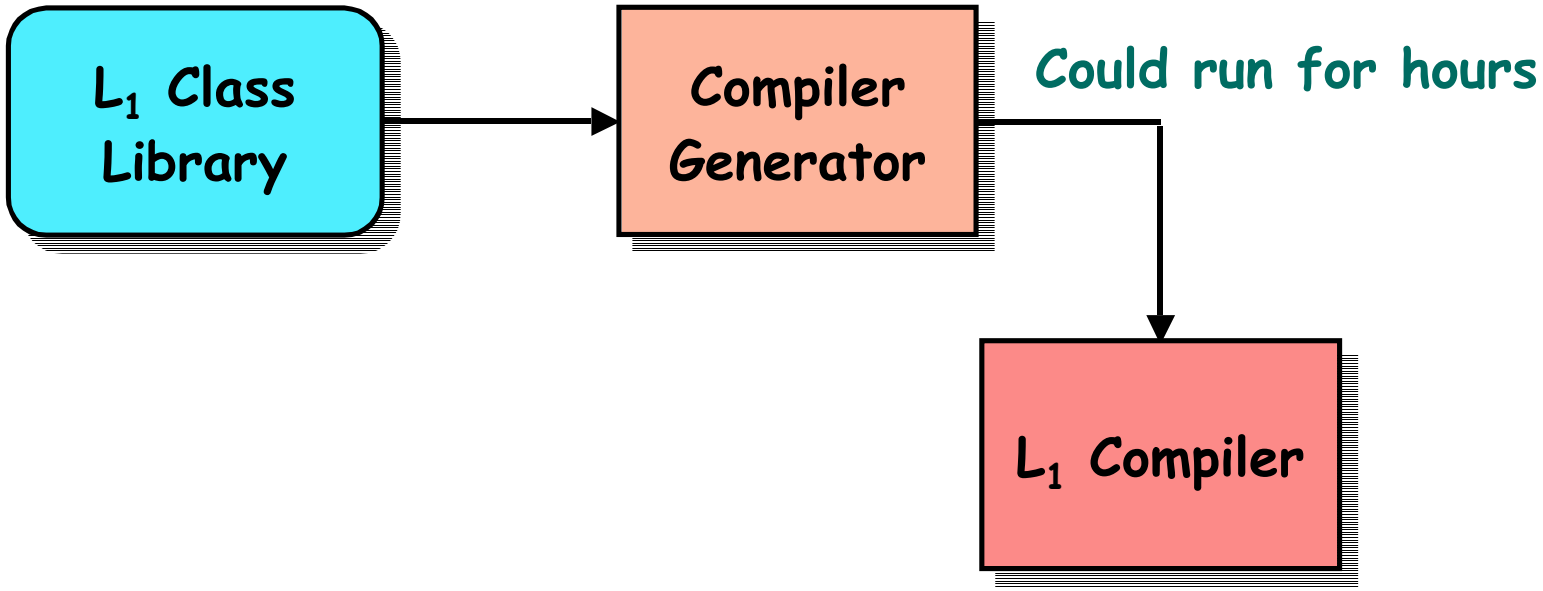


GrADSoft Architecture

Program Preparation System

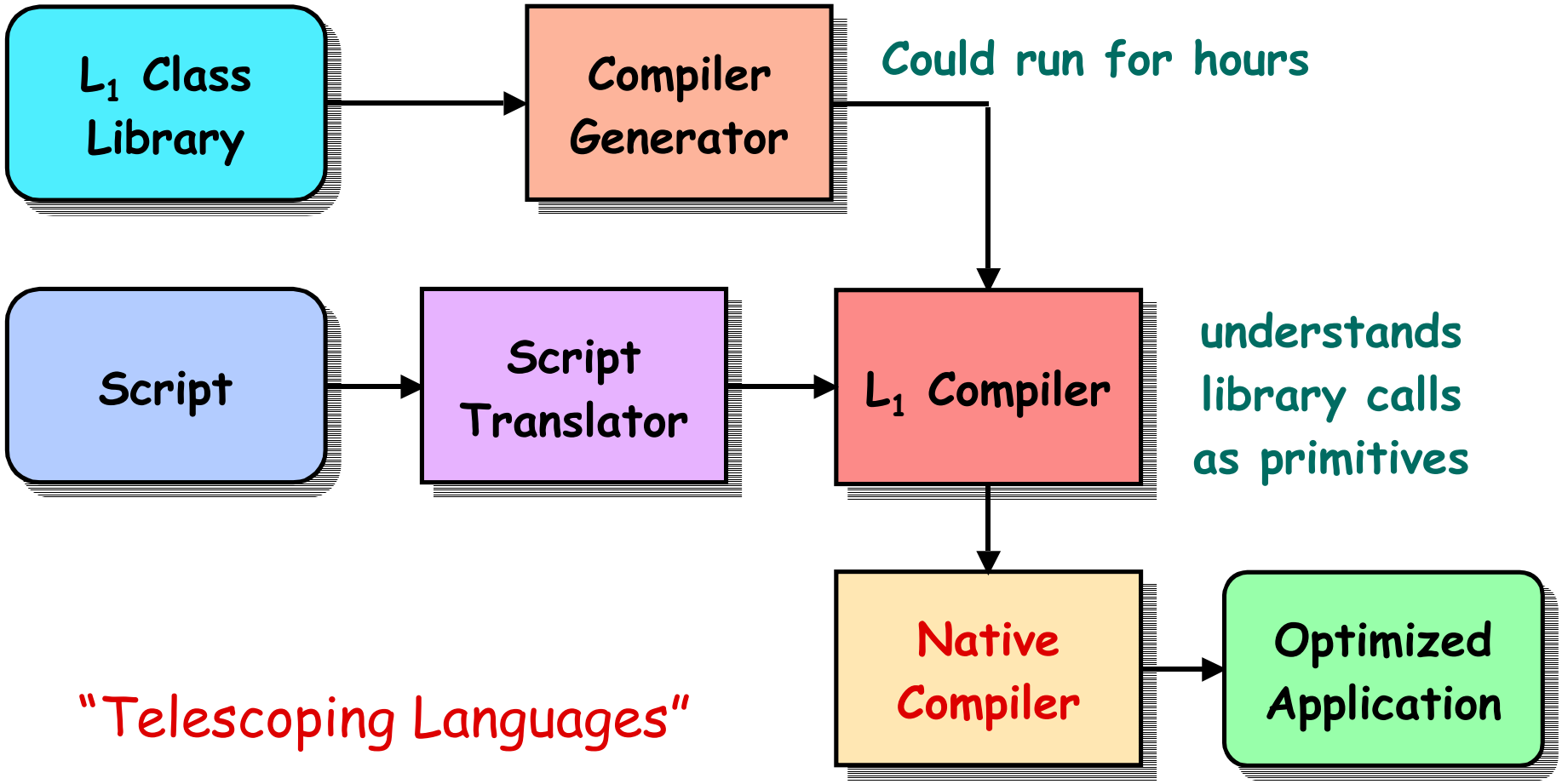


Automatic Generation of PSEs

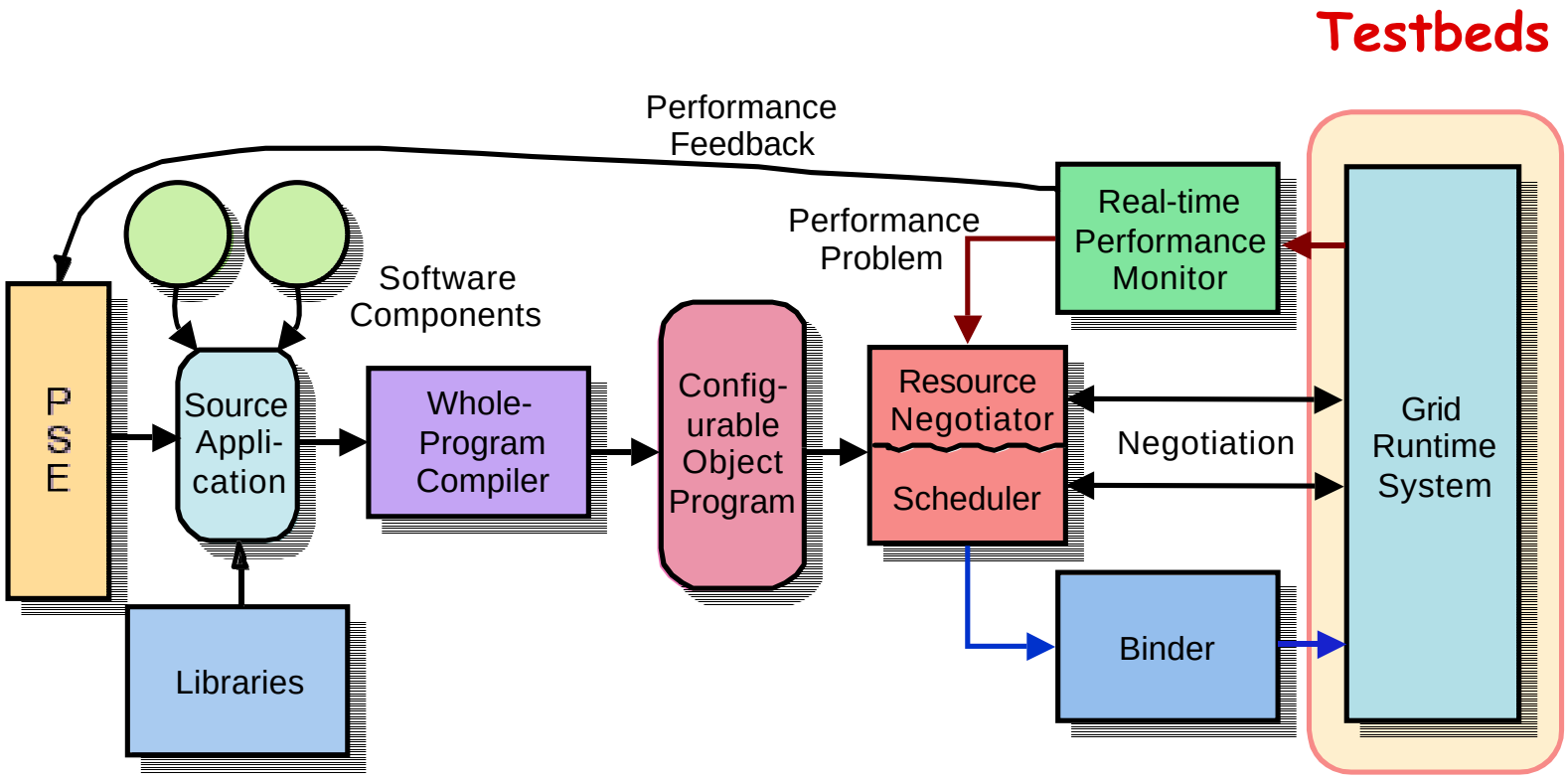


"Telescoping Languages"

Automatic Generation of PSEs



GrADSoft Architecture



Testbeds

- Goal:
  - Provide vehicle for experimentation with the dynamic components of the GrADS software framework
- MacroGrid (Carl Kesselman)
  - Collection of processors running Globus and GrADS middleware
    - Consistent software environment
  - At all 8 CGrADS sites + Indiana
    - Availability listed on web page
  - Permits experimentation with real applications
- MicroGrid (Andrew Chien)
  - Cluster of processors (currently Compaq Alphas and x86 clusters)
  - Runs standard Grid software (Globus, Nexus, GrADS middleware)
  - Permits simulation of varying loads and configurations
    - Stress GrADS components (Performance modeling and control)

Middleware Infrastructure for GrADS

- Empirical application studies for GrADS leverage successful Grid middleware efforts
  - Globus Meta Directory Service (MDS)
    - Scalable, distributed, robust system for cataloging and serving Grid resource data
  - The Network Weather Service (NWS)
    - High-performance, robust system for gathering Grid resource performance data, and making on-line forecasts of future performance levels
- GrADSoft tools consider these software systems to be Grid resources
  - Highly-available, high-performance, scalable
  - Basis of MacroGrid and MicroGrid

Knowledge Transfer and Partnerships

- Applications Partners
  - —Collaborative efforts to develop both applications and technology
    - not simply support for Grid application development
- Institutional Partners: PACIs and National Labs
  - —sources of application and technology partnerships
  - —deployment of software as appropriate
- Industrial Partners
  - —sources for new industrial applications
  - —deployment of software
  - —collaborations on research

Education and Outreach Programs

- Major Focus: increase participation of underrepresented minorities and women in science and technology
- Graduate Education
  - New courses incorporating Grid-oriented problem-solving
  - Exchange of graduate students across sites
- Undergraduate Education
  - New courses involving Grid topics
  - Undergraduate Research
    - Support communities to enhance retention
- K-12
  - Improve teacher training in IT
  - Pilot program on parent education

Management

- Integrated Research and Development Project
  - Goals include building two major infrastructures
    - Execution environment and Program Preparation System
  - Diverse collection of researchers with different backgrounds needed
  - Extensive planning required
- Organization
  - Executive Committee, External Advisory Committee, Staff
- Workshops
  - Three research workshops per year
    - Summary of progress, discussion of technical direction
  - Workshops include planning meeting by Executive Committee
- Documentation Series
  - Numbered design and planning documents

Why a Center?

- Integrated, Long-term Research Effort
  - Many researchers from different institutions and different academic backgrounds needed to address the problem
    - Focus will be needed to drive toward a common goal
- Requires Construction of Large Software Systems
  - Center coordination needed to integrate the many components
- Flexibility
  - To pursue new approaches as they emerge
- Focal Point for the Community
  - Many researchers must be drawn into the effort
- Intellectual Ferment
  - Enhances research, education, and knowledge transfer
- Critical Mass
  - To foster education and outreach programs

Research Talks

- Overview [Kennedy]
- Application Prototyping
  - Numerical Libraries on the Grid [Jack Dongarra]
  - Cactus and Other Applications [Ian Foster]
- Execution System
  - Execution System Overview [Fran Berman]
  - Performance Contracts and Monitoring [Dan Reed]
- Program Preparation System
  - Program Preparation System Overview [Keith Cooper]
  - Constructing Adaptive Grid Programs [John Mellor-Crummey]
- Testbeds
  - MacroGrid. MicroGrid, and Middleware [Carl Kesselman]

Summary

- Goal:
  - Design and build programming systems for the Grid that broaden the community of users who can develop and run applications in this complex environment
- Strategy:
  - Build an execution environment that automates the most difficult tasks
    - Maps applications to available resources
    - Manages adapting to varying loads and changing resources
  - Automate the process of producing Grid-ready programs
    - generate performance models and mapping strategies semi-automatically
  - Construct programs using high-level domain-specific programming interfaces