

# Programming Languages and Environments

**Challenges:**

# Programming Languages and Environments

## Challenges:

- Development of abstract Grid programming models
  - design of an implementation strategy for those models

# Programming Languages and Environments

## Challenges:

- Development of abstract Grid programming models
  - design of an implementation strategy for those models
- Development of easy-to-use programming interfaces
  - problem-solving environments

# Programming Languages and Environments

## Challenges:

- Development of abstract Grid programming models
  - design of an implementation strategy for those models
- Development of easy-to-use programming interfaces
  - problem-solving environments
- Compiler and language support for reliable performance
  - dynamic reconfiguration
  - optimization for distributed targets

# Programming Languages and Environments

## Challenges:

- Development of abstract Grid programming models
  - design of an implementation strategy for those models
- Development of easy-to-use programming interfaces
  - problem-solving environments
- Compiler and language support for reliable performance
  - dynamic reconfiguration
  - optimization for distributed targets
- Performance monitoring and control strategies
  - deep integration across compilers, tools, and runtime systems
  - performance contracts and dynamic reconfiguration

# Programming Languages and Environments

## Challenges:

- Development of abstract Grid programming models
  - design of an implementation strategy for those models
- Development of easy-to-use programming interfaces
  - problem-solving environments
- Compiler and language support for reliable performance
  - dynamic reconfiguration
  - optimization for distributed targets
- Performance monitoring and control strategies
  - deep integration across compilers, tools, and runtime systems
  - performance contracts and dynamic reconfiguration
- Robust reliable numerical and data-structure libraries
  - predictability and robustness of accuracy and performance
  - reproducibility, fault tolerance, and auditability

# Programming Models

- **Distributed Collection of Objects**
  - message passing for communication

# Programming Models

- **Distributed Collection of Objects**
  - message passing for communication
- **Problem-Solving Environment**
  - packaged components
  - graphical or scripting language for glue

# Programming Models

- **Distributed Collection of Objects**
  - message passing for communication
- **Problem-Solving Environment**
  - packaged components
  - graphical or scripting language for glue
- **Decomposition of Shared-Memory Programs**
  - language-based decomposition specification from programmer
  - parametrizable for reconfiguration
    - example: reconfigurable distributed arrays (DAGH)
  - implemented as distributed object collection

# Compilation Architecture

- **Challenge**

- dynamic, changing nature of target, difficult to manage by hand

# Compilation Architecture

- **Challenge**

- dynamic, changing nature of target, difficult to manage by hand

- **Solution**

- A new program preparation architecture

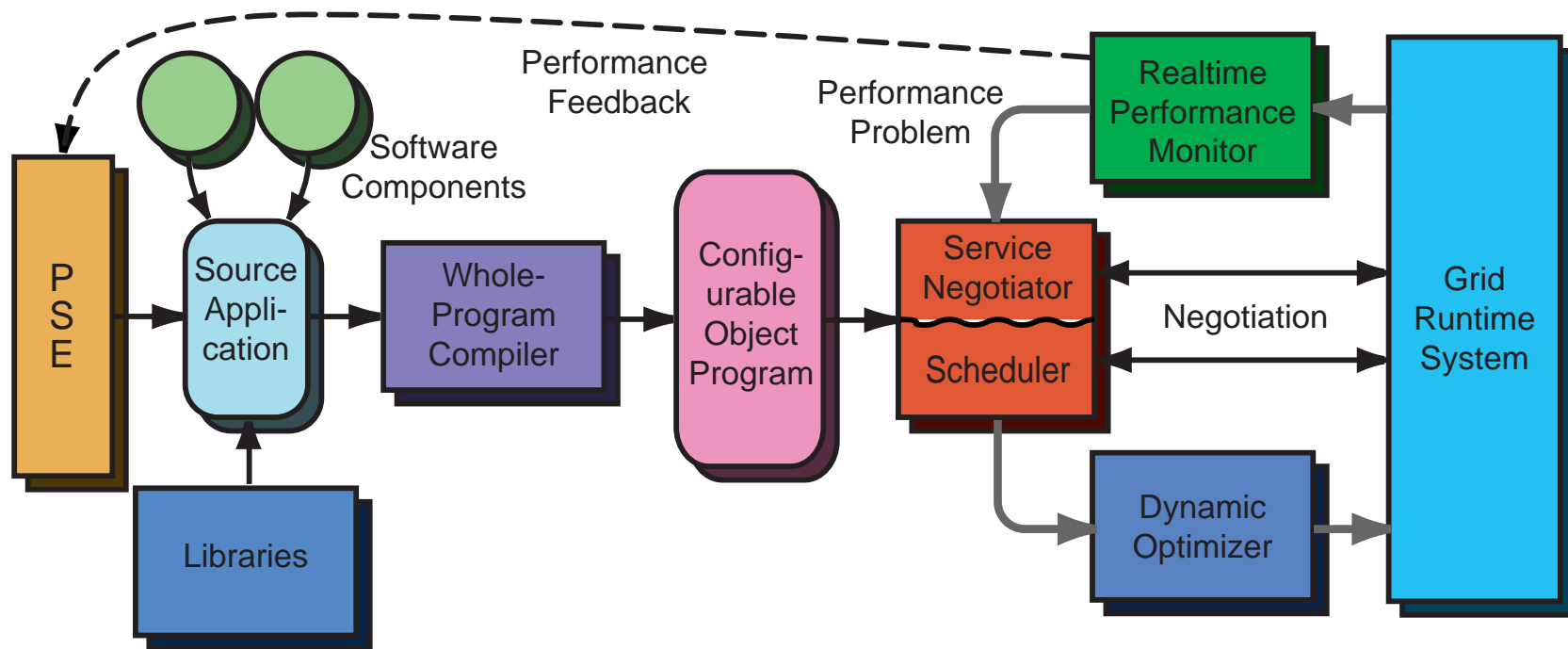
# Compilation Architecture

- **Challenge**

- dynamic, changing nature of target, difficult to manage by hand

- **Solution**

- A new program preparation architecture



# Research Strategies

- **Near-term**

- application experience to identify opportunities
- prototype reconfiguration system
  - with performance monitoring, without dynamic optimization
- preliminary research on robust, fault-tolerant libraries

# Research Strategies

- **Near-term**

- application experience to identify opportunities
- prototype reconfiguration system
  - with performance monitoring, without dynamic optimization
- preliminary research on robust, fault-tolerant libraries

- **Medium-term**

- experiment with high-level programming models
- preliminary dynamic optimization strategies
- performance contracts
- reconfigurable libraries

# Research Strategies

- **Near-term**

- application experience to identify opportunities
- prototype reconfiguration system
  - with performance monitoring, without dynamic optimization
- preliminary research on robust, fault-tolerant libraries

- **Medium-term**

- experiment with high-level programming models
- preliminary dynamic optimization strategies
- performance contracts
- reconfigurable libraries

- **Long-term**

- refinement of concept of configurable object program
- refinement of performance contracts
- tools for building and tuning reconfigurable programs