The Virtual Grid Application Development Software (VGrADS) Project

Ken Kennedy
Center for High Performance Software
Rice University

http://www.hipersoft.rice.edu/vgrads/
The VGrADS Team

- VGrADS is an NSF-funded Information Technology Research project

- Plus many graduate students, postdocs, and technical staff!
The VGrADS Vision: National Distributed Problem Solving

• Where We Want To Be
  — Transparent Grid computing
    - Submit job
    - Find & schedule resources
    - Execute efficiently

• Where We Are
  — Low-level hand programming

• What Do We Need?
  — A more abstract view of the Grid
    - Each developer sees a specialized “virtual grid”
  — Simplified programming models built on the abstract view
    - Permit the application developer to focus on the problem
The Original GrADS Vision

Program Preparation System

Execution Environment

Performance Feedback

Performance Problem

Libraries

Source Application

Software Components

Whole-Program Compiler

Configurable Object Program

Real-time Performance Monitor

Resource Negotiator

Scheduler

Negotiation

Binder

Grid Runtime System

Virtual Grid Application Development Software Project
Lessons from GrADS

• **Mapping and Scheduling for MPI Jobs is Hard**
  - Although we were able to do some interesting experiments

• **Performance Model Construction is Hard**
  - Hybrid static/dynamic schemes are best
  - Difficult for application developers to do by hand

• **Heterogeneity is Hard**
  - We completely revised the launching mechanisms to support this
  - Good scheduling is critical

• **Rescheduling/Migration is Hard**
  - Requires application collaboration (generalized checkpointing)
  - Requires performance modeling to determine profitability

• **Scaling to Large Grids is Hard**
  - Scheduling becomes expensive
VGrADS Virtual Grid Hierarchy
Virtual Grids and Tools

• Abstract Resource Request
  - Permits true scalability by mapping from requirements to set of resources
    - Scalable search produces manageable resource set
  - Virtual Grid services permit effective scheduling
    - Fault tolerance, performance stability

• Look-Ahead Scheduling
  - Applications map to directed graphs
    - Vertices are computations, edges are data transfers
  - Scheduling done on entire graph
    - Using automatically-constructed performance models for computations
    - Depends on load prediction (Network Weather Service)

• Abstract Programming Interfaces
  - Application graphs constructed from scripts
    - Written in standard scripting languages (Python, Perl, Matlab)
Virtual Grids

• **Goal:** Provide abstract view of grid resources for application use
  – Will need to experiment to get the right abstractions

• **Assumptions:**
  – Underlying scalable information service
  – Shared, widely distributed, heterogeneous resources
  – Scaling and robustness for high load factors on Grid
  – Separation of the application and resource management system

• **Basic Approach:**
  – Specify vgrid as a hierarchy of ...
    - Aggregation operators (ClusterOf, LooseBagOf, etc.) with ...
    - Constraints (type of processor, installed software, etc.) and ...
    - Application-based rankings (e.g. predicted execution time)
  – Execution system returns (candidate) vgrid, structured as request
  – Application can use as it sees fit, make further requests
Programming Tools

• **Collaborating on definition of the Virtual Grids interface**
  - Initial experiments based on GrADS infrastructure

• **Focus: Automating critical application-development steps**
  - Building workflow graphs
    - From Python scripts used by EMAN
  - Scheduling workflow graphs
    - Heuristics required (problems are NP-complete at best)
    - Good initial results if accurate predictions of resource performance are available (see EMAN demo)
  - Constructing of performance models
    - Based on loop-level performance models of the application
    - Requires benchmarking with (relatively) small data sets, extrapolating to larger cases
  - Initiating application execution
    - Optimize and launch application on heterogeneous resources
VGrADS Demos at SC04

- **EMAN - Electron Microscopy Analysis [Rice, Houston]**
  - 3D reconstruction of particles from electron micrographs
  - Workflow scheduling and performance prediction to optimize mapping

- **GridSAT - Boolean Satisfiability [UCSB]**
  - Classic NP-complete problem useful in circuit design and verification
  - Performance-based dynamic resource allocation and scheduling

---

**Figure 1:** Example of conflict analysis with learning and non-chronological backtracking