The Virtual Grid Application Development Software (VGrADS) Project

The VGrADS Team

http://vgrads.rice.edu/
VGrADS Goal: Distributed Problem Solving

• Where We Want To Be
  o Transparent computing
    - In an increasingly distributed space
    - Applications to HPC
    - Applications to cloud computing

• Where We Were (circa 2003)
  o Low-level hand programming
  o Programmer had to manage:
    - Heterogeneous resources
    - Scheduling of computation and data movement
    - Fault tolerance and performance adaptation

• What Progress Have We Made?
  o Separate application development from resource management
    - VGrADS provides a uniform “virtual grid” abstraction atop widely differing resources
  o Provide tools to bridge the gap
    - Scheduling, resource management, distributed launch, simple programming models, fault tolerance, grid economies
VGrADS Goal: Distributed Problem Solving

• Where We Want To Be
  o Transparent computing
    - In an increasingly distributed space
    - Applications to HPC
    - Applications to cloud computing

• Where We Were (circa 2003)
  o Low-level hand programming
  o Programmer had to manage:
    - Heterogeneous resources
    - Scheduling of computation and data movement
    - Fault tolerance and performance adaptation

• What Progress Have We Made?
  o Separate application development from resource management
    - VGrADS provides a uniform “virtual grid” abstraction atop widely differing resources
  o Provide tools to bridge the gap
    - Scheduling, resource management, distributed launch, simple programming models, fault tolerance, grid economies
Overview of SC08 Demo

• Building on SC04 - SC07 demonstrations
  o Gradually built up system to handle LEAD workflow
  o Previous years focused improved performance estimates, scheduling methods, fault tolerance
  o Use LEAD as an application driver

• New for SC08
  o VGrADS integrates HPC and cloud resources
    - Using TeraGrid (HPC), Amazon EC2 (cloud), Eucalyptus (cloud) resources
    - Using reservations, batch queues, and on-demand clouds
  o Scheduling for balancing deadlines, reliability, and cost
    - vgES supports search for best set of resources
    - Application-specific trade-offs of reliability, time, cost
  o Abstractions really do work!
Demo Architecture

LEAD System

Portal System

Creates workflows

Plans workflow

Manages workflow

Manages resources

VGrADS System

Globus

HPC and Cloud Resources

Executes components

HPC

Cloud

Executes components
Executing LEAD Workflow Sets

- Demonstrate planning and execution of LEAD workflow sets execution atop virtualized cloud and Grid resources.

- LEAD Workflow Orchestration schedules a set of independent workflows with characteristics
  - a deadline $D$ (e.g. 2 hours)
  - fraction $F$ such that at least $F$ of the workflows finish by the deadline (e.g. 3/8)

- Virtual Grid Execution System (vgES) provides an abstraction over batch and cloud systems including Amazon EC2 and Eucalyptus cloud sites.
LEAD System (without VGrADS)

Selected components

Research Prototype components

Executes components

HPC Resources
LEAD System (with VGrADS)

- **Portal**
- **LEAD System**
  - **Workflow Planner**
  - **DAG Scheduler**
  - **Execution Manager**
- **LEAD BPEL Workflow Engine**
- **Application Service**
- **Event Broker**
- **App. Factory**
- **Globus**
- **VGrADS System**
- **HPC and Cloud Resources**

Research Prototype components:
- Plans workflow
- Manages resources
- Executes components
VGrADS Components at SC08

- **Virtual Grid Execution System (vgES)**
  - Uses Amazon EC2 tools to interact with cloud resources
  - Uses QBETS and Globus to provision batch resources
  - Uses Personal PBS to control execution on batch resources
  - Provides a “resource gantt chart” view of resources to aid higher level workflow orchestration tool

- **Eucalyptus** - Developed for VGrADS
  - Implements cloud computing on Xen-enabled clusters
  - Open-source software infrastructure that is compatible with Amazon EC2
  - vgES “thinks” a Eucalyptus cloud is EC2

- **Fault Tolerance (FTR)**
  - Schedules a task to increase the probability of successful execution of a task up to a desired level, constrained by resource availability and application deadlines
VGrADS System

LEAD System

Plans workflow

Manages workflow

Virtual Grid Execution System

Fault tolerance
resource find & bind

Globus

HPC and Cloud Resources

Batch Systems

HPC

Cloud
LEAD / VGrADS Architecture

LEAD System
- Portal
- Workflow Planner
- DAG Scheduler
- Virtual Grid Execution System
  - Fault tolerance
  - resource find & bind
- Execution Manager
- Application Service
  - Event Broker
  - App. Factory
  - BQP / VAR
  - GT4, Personal PBS

VGrADS System

Globus

HPC and Cloud Resources
- UCSD
- Renci
- NCSA
- Amazon Webservices
- UCSD
- Eucalyptus
- UF
- Renci
And Now, On With the Demo
LEAD System Components

- **Workflow Orchestration**
  - Query vgES for resources
  - Phase 1: Schedule fraction F of workflows
  - Phase 2: Sort the scheduled tasks of fraction F by success probability. Compare scheduling from FTR and DAGscheduler on each DAG to make next orch. Decision
  - Phase 3: Try remaining FTR or remaining DAGs
  - Phase 4: For tasks below certain threshold, schedule on EC2

- **Execution Manager**
  - Early prototype for ordered execution of tasks on the slot based on the schedule determined by the orchestration.
Demo Vizualization Key

Resource Name

Workflow Color

Slot Width

Time

Task Status

Run

Done

Failed

mayhem

kittyhawk

VGrADS
Virtual Grid Application Development Software Project
VGrADS at SC08

• Booth Talks and Demos
  o When and where
    - Tuesday, 11:30am, RENCI booth (2633)
    - Wednesday, noon, GCAS booth (285)
    - Wednesday, 2:00pm, SDSC booth (568)
    - Wednesday, 4:00pm, RENCI booth (2633)

• Other talks
    - Tuesday, 2:30-3:00pm, Ballroom G
  o Paper: “Analysis of Application Heartbeats: Learning Structural and Temporal Features in Time Series Data for Identification of Performance Problems” by Emma Buneci and Dan Reed
    - Thursday, 2:30-3:00pm, Ballroom E
    - Thursday, 11:30-11:45am, Room 17A/B
The VGrADS Team

- Dan Reed
- Anirban Mandal
- Keith Cooper
- Ken Kennedy
- Charles Koelbel
- Richard Tapia
- Linda Torczon
- Carl Kesselman
- Andrew Chien
- Henri Casanova
- Fran Berman
- Rich Wolski
- Lennart Johnsson
- Anirban Mandal
- Jack Dongarra

- Plus many graduate students, postdocs, and technical staff!

Thanks to the National Science Foundation
Backup slides
VGrADS Big Ideas

• Virtualization of Resources
  o Application specifies required resources in Virtual Grid Definition language (vgDL)
    - Give me a tight bag of as many Opterons as possible
  o Virtual Grid Execution System (vgES) produces specific virtual grid matching specification
    - May involve adding layer of capabilities
    - vgES also reports properties of the virtual grid to application

• Generic In-Advance Scheduling of Application Workflows
  o Application includes performance models for all workflow nodes
    - Performance models automatically constructed
  o Software schedules applications onto virtual Grid, minimizing total makespan
    - Including both computation and data movement times
Virtual Grids (VGs)

• A Virtual Grid (VG) takes
  o Shared heterogeneous resources
  o Scalable information service

• and provides
  o A hierarchy of application-defined aggregations with constraints and rankings
  o E.g. Cluster of Opteron nodes

• Virtual Grid Execution System (vgES) implements VG
  o Brings order out of chaos

• Can be implemented on multiple platforms
  o Grids
  o Compute clouds
  o HPC
  o Manycore chips
Virtual Grids (VGs)

- A Virtual Grid (VG) takes
  - Shared heterogeneous resources
  - Scalable information service
- and provides
  - A hierarchy of application-defined aggregations with constraints and rankings
  - E.g. Cluster of Opteron nodes
- Virtual Grid Execution System (vgES) implements VG
  - Brings order out of chaos
- Can be implemented on multiple platforms
  - Grids
  - Compute clouds
  - HPC
  - Manycore chips
Virtual Grids (VGs)

• A Virtual Grid (VG) takes
  o Shared heterogeneous resources
  o Scalable information service

• and provides
  o A hierarchy of application-defined aggregations with constraints and rankings
  o E.g. Cluster of Opteron nodes

• Virtual Grid Execution System (vgES) implements VG
  o Brings order out of chaos

• Can be implemented on multiple platforms
  o Grids
  o Compute clouds
  o HPC
  o Manycore chips

vgDL Description

Resource Universe

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USERNAME</th>
<th>STATE</th>
<th>PROCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>364932</td>
<td>square</td>
<td>Running</td>
<td>200</td>
</tr>
<tr>
<td>364863</td>
<td>triangle</td>
<td>Running</td>
<td>4</td>
</tr>
<tr>
<td>364312</td>
<td>pentagon</td>
<td>Running</td>
<td>48</td>
</tr>
</tbody>
</table>

Successfully
Bound Candidates

Virtual Grid
VGrADS Tool Research

• Scheduling of workflow computations
  o Off-line look-ahead scheduling dramatically improves total time
  o Accurate performance models significantly affect quality of scheduling
  o Batch queue behavior can be predicted accurately enough for scheduling decisions

• Fault tolerance
  o Diskless checkpointing for linear algebra computations (application-specific)
  o Qualitative performance analysis for fault detection
  o Optimal checkpoint frequency for iterative applications
  o Availability prediction
VGrADS: Application-driven Research

- **SC04**
  - Executing the EMAN application
    - Incorporated performance models

- **SC05**
  - Executing the EMAN application
    - Used batch queue predictions (and performance models)

- **SC06**
  - Executing the LEAD application
    - Scheduled to vgES virtualized resource reservation slots (leveraging queue predictions and performance models)

- **SC07**: *Fault tolerance through virtual grid abstractions*
  - Executing the LEAD application
    - Incorporates reliability information (and above features)
    - Combine replication and migration of workflow nodes
The LEAD Vision: A Paradigm Shift

Dynamic Observations → Analysis/Assimilation → Prediction/Detection → Product Generation, Display, Dissemination

Models and Algorithms Driving Sensors

The CS challenge: Build cyberinfrastructure services that provide adaptability, scalability, availability, useability, and real-time response.

End Users

NWS
Private Companies
Students
Coordinated by VGrADS

Data Analysis & Simulations

Dynamic Observations

Provide data to

Coordinate Collection of

Predictions
6 Workflows on 7 Clusters