The Virtual Grid Application Development Software (VGrADS) Project

“VGrADS: Enabling e-Science Workflows on Grids and Clouds with Fault Tolerance”

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

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

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

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

- Built on previous SC demonstrations
  - Gradually built up system to handle LEAD workflow
  - Previous years focused improved performance estimates, scheduling methods, fault tolerance
  - Use LEAD as an application driver

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

- **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 (now Eucalyptus Inc.*)
  - 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
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.
Workflow Orchestration

- **Workflow Orchestration**
  - **Phase 1:** *Minimal Scheduling* - Schedule minimum fraction of workflows using simple probabilistic DAG scheduler
  - **Phase 2:** *Fault Tolerance Tradeoff* - Compare scheduling additional workflows with increasing fault-tolerance of one or more tasks of the scheduled workflows
  - **Phase 3:** *Additional Scheduling* - Use available slots for other scheduling
  - **Phase 4:** *EC2 Scheduling* - For tasks below certain threshold, schedule on EC2

- **Execution Manager**
  - A prototype for ordered execution of tasks on the slot based on the schedule determined by the orchestration.
Comparison of the LEAD-VGrADS collaboration system with cyberinfrastructure production deployments

Without VGrADS

User → Portal → Workflow Engine → Application Service → Execution Manager → Resources

- Globus
- EC2 interfaces

With VGrADS

User → Portal → Workflow Planner → Virtual Grid Execution System → Application Service → Execution Manager → Resources

- Globus
- EC2 interfaces

- Specific protocol based execution
- Standard execution
- Resource planning
- Query execution plan
- Resource binding

User
Example Scheduling of Workflows

(a) Example Workflow Set
need $F=2/3$

(b) Uncoordinated Schedule
Without VGrADS

(c) Coordinated Schedule
With VGrADS
Interaction of system components for resource procurement and planning
LEAD / VGrADS Architecture: Putting It All Together
Visualization Key

Resource Name

Workflow Color

Run

Done

Failed

Task Status

Slot Width

Time

Virtual Grid Application Development Software Project
Snapshot of Execution of 6 Workflows
6 Workflows on 7 Clusters
Infrastructure Timing Metrics

Binding time

Planning time line

Workflow execution time line
Fault Tolerance Exploration

Low Reliability

Medium Reliability

High Reliability
Conclusions

• VGrADS’ virtual grid abstraction simplifies
  o Programming grid and cloud systems for e-Science workflows
  o Managing QoS (performance and reliability)

• The VGrADS system unifies workflow execution over
  o batch queue systems (with and without advanced reservations), and
  o cloud computing sites (including Amazon EC2 and Eucalyptus)

• The system provides an enabling technology for executing deadline-driven, fault-tolerant workflows

• The integrated cyber-infrastructure from the LEAD and VGrADS system components provides a strong foundation for next-generation dynamic and adaptive environments for scientific workflows
Thank You


- Paper to be presented at 3:30 p.m., Thursday, Nov. 19 in Room E145 - 146

Thank you..