Responses to Questions

http://vgrads.rice.edu/site_visit/april_2005/slides/responses

vgES Accomplishments

- Design and Implementation of Synthetic Resource Generator for Grids which can generate Realistic Grid Resource Environments of Arbitrary Scale and Project forward in time (a few years)
- Study of Six major Grid Application to understand desirable Application Resource Abstractions and drive the design of vgES
- Complete Design and Implementation of initial vgDL Language which allows Application-level Resource Descriptions
- Complete Design and Implementation of Virtual Grid interface which provides an explicit resource abstraction, enabling application-driven resource management
- Design and Implementation of “Finding and Binding” Algorithms
  - Simulation Experiments demonstrate the effectiveness of “Finding and Binding” vs. Separate Selection in Competitive Resource Environments
- Design and Implementation of a vgES Research Prototype Infrastructure which
  - Realizes the Key Virtual Grid Ideas (vgDL, FAB, Virtual Grid)
  - Enables Modular Exploration of Research Issues by VGrADS Team
  - Enables Experimentation with Large Applications and Large-scale Grid Resources (Leverages Globus/Production Grids)

vgES Research Plans for FY06

- Dynamic Virtual Grid
  - Implement Dynamic Virtual Grid Primitives
  - Work with Fault Tolerance and Dynamic Workflow Applications to evaluate utility
- Experiments with Applications (EMAN, LEAD, and VDS)
  - Work with application teams on how to generate initial vgDL specs
  - Evaluate Selection and Binding for those applications
  - Experiment with Application Runs
  - Stretch to External Grid Resources
- Explore Relation of vgES with non-immediate Binding (Batch Schedulers, Advance Reservations, Glide-ins)
  - Characterization and Prediction, Reservation
  - Statistical Guarantees
  - Explore what belongs below/above VG Abstraction

vgES Research Plans for FY06 (cont.)

- Explore Efficient Implementation of Accurate Monitoring
  - Efficient compilation/implementation of custom monitors
  - Explore tradeoff of accuracy (flat) versus scalable (hierarchical)
  - Default and customizable expectations

Programming Tools Accomplishments

- Collaborated on development of vgDL
- Developed an application manager based on Pegasus
  - Supports application launch and simple fault tolerance
  - In progress: integration with vgES
  - Demonstrated on EMAN
- Developed and demonstrated whole-workflow scheduler
  - Papers have demonstrated effectiveness in makespan reduction
- Developed a performance model construction system
  - Demonstrated its effectiveness in the scheduler
- Applied the above technologies to EMAN
- Dynamic optimization
  - Brought LLVM in house and wrote new back-end components (Das Gupta, Eckhardt) that work across multiple ISAs.
  - Began work on a demonstration instance of compile-time planning and run-time transformation (Das Gupta)

Programming Tools Plans for FY06

- **Application management**
  - Generation of vgDL
  - Preliminary exploration of rescheduling interfaces
- **Scheduling**
  - Explore new “inside-out” whole-workflow strategies
  - Finish experiments on two-level scheduling and explore class-based scheduling algorithms
- **Improved performance models**
  - Handle multiple outstanding requests
  - Continued research on MPI applications
  - Explore new architectural features

More Programming Tools Plans for FY06

- **Preliminary handling of Python scripts**
  - Application of size analysis
  - Use in EMAN 2
- **Retargetable program representation**
  - Running demo of compile-time planning and run-time transformation (Das Gupta)
  - Reach point where LLVM is a functional replacement for GCC in the VGrADS build-bind-execute cycle

EMAN Accomplishments & Plans

- Accomplishments
  - Applied programming tools to bring up EMAN up on the VGrADS testbed
    - Developed floating-point model
    - Applied memory-hierarchy model
  - Demonstrated effectiveness of tools on second iteration of EMAN
    - In two weeks
  - Demonstrated scaling to significantly larger grids and problem instances
    - Larger than would have been possible using GrADS framework
- Plans for FY06
  - Explore EMAN 2 as a driver for workflow construction from scripts
  - Bring up EMAN 2 using enhanced tools
    - Test new inside-out scheduler on EMAN 2
  - Work with TIGRE funds to plan for EMAN challenge problem (3000 Opterons for 100 hours)
    - Use as success criterion for TIGRE/LEARN

LEAD, Scalability & Workflows Accomplishments

- LEAD workflow validation with vgDL/vgES
  - virtual grid design shaping
    - static and dynamic workflow feasibility assessment
  - Rice scheduler integration (with simplified models)
- NWS/HAPI software integration and extension
  - scalable sampling of health and performance data
    - vgES integration and access
- Qualitative classification methodology (Emma Buneci thesis)
  - measurement driven classification
    - behavioral classification and reasoning system
- New research group launched at UNC Chapel Hill
  - all new students, staff and infrastructure

LEAD, Scalability & Workflows Plans for FY06

- Monitoring scalability for virtual grids
  - performance and health monitoring
  - statistical sampling, failure classification and prediction
- Performability (performance plus reliability)
  - integrated specification and tradeoffs
  - reliability policy support
    - over-provisioning, MPI fault tolerance, restart
- Complex workflow dynamics and ensembles (LEAD driven)
  - research parameter studies (no real-time constraints)
  - weather prediction (real-time constraints)
- Behavioral application classification
  - validation of classification and temporal reasoning approach

Fault Tolerance Accomplishments

- GridSolve
  - Integrated into VGrADS framework
- Fault tolerant linear algebra algorithms
  - Use VGrADS vgDL and vgES to acquire virtual grid

Plans for FY06

- Fault Tolerant applications
  - Software to determine the checkpointing interval and number of checkpoint processors from the machine characteristics.
    - Use historical information.
    - Monitoring
    - Migration of task if potential problem
  - Local checkpoint and restart algorithm.
    - Coordination of local checkpoints.
    - Processors hold backups of neighbors.
  - Have the checkpoint processes participate in the computation and do data rearrangement when a failure occurs.
    - Use p processors for the computation and have k of them hold checkpoint.
  - Generalize the ideas to provide a library of routines to do the diskless check pointing.
  - Look at "real applications" and investigate "Lossy" algorithms.
- GridSolve integration into VGrADS
  - Develop library framework

VGrADS-Only Versus Leveraged

- **Rephrased Question: Which accomplishments and efforts were exclusive to VGrADS and which were based on shared funding?**

VGrADS-Generated Contributions

- Virtual Grid abstraction and runtime implementation
  - vgDL language for high-level, qualitative specifications
  - Selection/Binding algorithms and based on vgDL
  - vgES runtime system and API research prototype
- Scheduling
  - Novel, scalable scheduling strategies using the VG abstraction
- Resource Characterization and Monitoring
  - Batch-queue wait time statistical characterization
  - NWS “Doppler Radar” API
  - Application behavior classification study
- Applications
  - LEAD workflow / vgES integration
  - Pegasus / vgES integration
  - EMAN numerical performance modeling and EMAN / vgES integration
  - GridSolve / vgES integration
- Fault-tolerance
  - HAPI / vgES integration
- VGrADS testbed

Projects Used by VGrADS

- Grid middleware
  - Globus [NSF NMI, NSF ITR, DOE SIDAC]
  - Pegasus [NSF ITR]
  - DVC [NSF ITR]
  - NWS [NSF NGS, NSF NMI, NSF ITR]
  - GridSolve [NSF NMI]
- Fault-tolerance
  - FT-MPI [DOE MICS]
  - FT-LA (Linear Algebra) [DOE LACSI]
  - HAPI [DOE LACSI]
- Applications
  - EMAN application [NIH]
  - EMAN performance modeling [DOE LACSI]
  - GridSAT development [NSF NGS]
  - LEAD [NSF ITR]
- Infrastructure
  - Teragrid [NSF ETF]

Jointly Funded Projects

- Grid middleware
  - Globus [NSF NMI, NSF ITR, DOE SIDAC]
  - Pegasus [NSF ITR]
  - DVC [NSF ITR]
  - NWS [NSF NGS, NSF NMI, NSF ITR]
  - GridSolve [NSF NMI]
- Fault-tolerance
  - FT-MPI [DOE Harness project]
  - FT-LA (Linear Algebra) [DOE LACSI]
  - HAPI [DOE LACSI]
- Applications
  - EMAN application [NIH]
  - EMAN performance modeling [DOE LACSI]
  - GridSAT development [NSF NGS]
  - LEAD [NSF ITR]
- Infrastructure
  - Teragrid [NSF ETF]

Milestones and Metrics

Can you quantify the goals of this program? Can you update the milestones and provide quantitative measures?

- Milestones in the original SOW:
  - Year 1: Mostly achieved, some deferred, some refocused
  - Year 2: Good progress on relevant milestones
  - Later years: needs to be updated based on changing plans
- We will revise milestones for FY06 and update for later years annually. The plans provided on previous slides are a good start
- Question of quantification is a difficult one (several answers on subsequent slides)

Quantitative Metrics and Goals

- Increased capability provided by VGrADS can be quantified by increases in a number of dimensions
  - 1/(error in match of performance models)
  - # of workflow nodes
  - # of used resources
  - 1/(time to find and bind)
  - percentile quality of FAB results
  - # of resources in grid resource environment
  - # of nodes monitorable
  - Maximum size of computation completable

- If helpful, we can construct a metric which combines these to measure the advance in capabilities achieved by the project.

Role of Executive Committee Judgment

- We regularly re-evaluate progress and plans
  - executive committee discussion, based on broad input
  - feedback from application collaborators and national centers
  - GrADS experience showed periodic priority re-evaluation
- It would be counterproductive to replace this process by a purely quantitative measurement
- However, we can attempt to augment this process by quantitative measures where they make sense.

Application Drivers

How are the applications driving this, and how do they provide criteria for success? How do you know the application creators believe VGrADS is a contribution?

- Case studies for evaluating alternative research approaches
  - realistic resource and behavior cases
  - real costs and benefits for virtual grid techniques
    - virtual grid abstractions, scheduling overheads
    - efficacy of resource management decisions
- Reality of application benefits
  - domain scientists are investing time in the collaboration (for free)
  - adoption of VGrADS technologies by application groups
  - standards influence and adoption

Education, Outreach, & Training

Can you expand outreach at low cost beyond Rice?

- We hope to distribute CS-CAMP materials through the EPIC network
  - Agreement in principle with Ann Redelfs
  - Would require significant local funding to duplicate the program
- We will package & distribute the grid-oriented courseware
  - General education course + two graduate courses on the grid
  - May also fit into the EPIC distribution scheme
- We will apply for a supplement through NSF BPC this fall
  - Incarnation of CS-CAMP session in California
  - Online tutorial resources for using VGrADS tools (?)

Education, Outreach, & Training

Is further academic outreach possible, to allow other universities and institutions to benefit from this work?

- We will distribute our software stack in open-source form
  - Allow other research groups to use the tools, as they mature
  - To have impact, would need to develop an online tutorial and/or give a tutorial at a conference such as SC
    - Would require resources; potentially from BPC
- Materials from our courses are available on the web
  - Simplify adoption at new schools
  - Courses use VGrADS and GrADSoft tools

Research Impact

As the project moves forward in time with successes and failures, what are your plans to prioritize activities in the project? For example, how can you ensure impact of the research without hardening of research prototypes?

- We regularly re-evaluate progress and plans
  - executive committee discussion, based on broad input
  - feedback from application collaborators and national centers
  - GrADS experience showed periodic priority re-evaluation
- There are many impact mechanisms and metrics
  - educated, involved and well placed students
  - conference and journal publications
  - international visibility and community leadership
  - availability of prototype software and conference demonstrations
  - application group interactions
  - concept transfer to other funded projects
  - selected software integration with other activities
- We are continuing to seek funding sources for software packaging

Large Scale Experiments

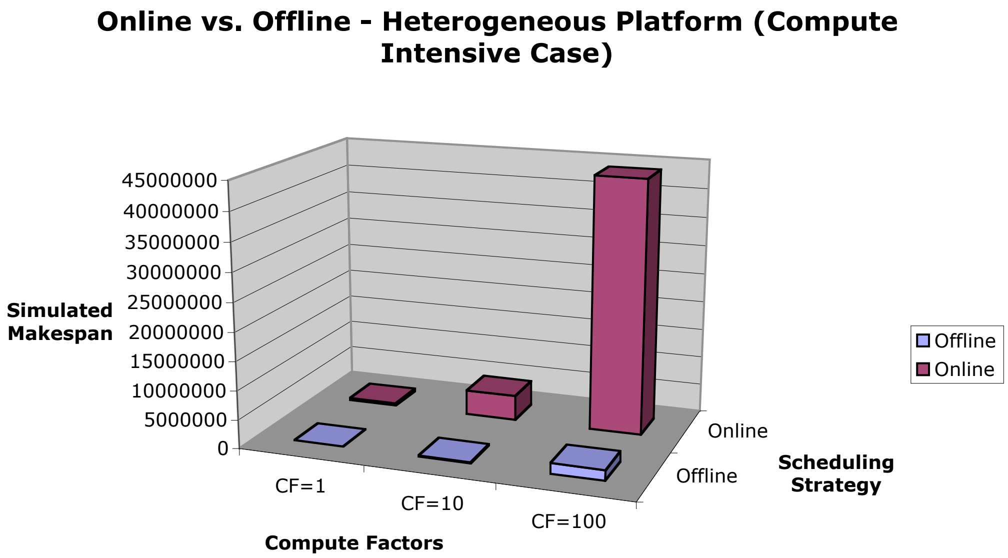
How scalable is this work? Has anything been done on large applications?

How about running on the TeraGrid?

- Large Runs
  - Montage Runs: 17,000 node Montage workflow, 100's of machines
    - Run on TeraGrid
  - Rice Scheduling Simulation: 500 node EMAN workflows, 5,000 processors
  - Scheduled 3029 Montage workflow steps using Rice scheduler (8 seconds)
    - Projected makespans 12,000 hours => 498 hours
- NWS demonstrated scalable for 10,000 Time Series (many nodes)
- Demonstrated Scalable Selection: 1,000,000 resource Grid Environments

Workflow Scheduling Results

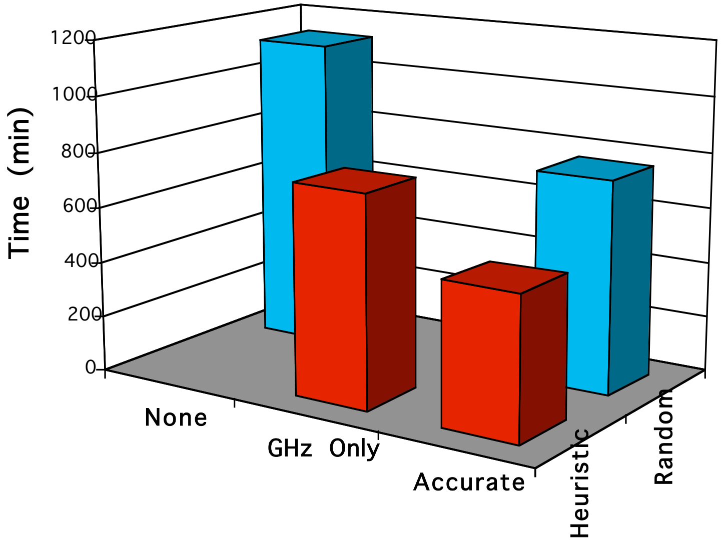
Dramatic makespan reduction of *offline* scheduling over *online* scheduling — Application: **Montage**



"Resource Allocation Strategies for Workflows in Grids"

CCGrid'05

Value of *performance models* and *heuristics* for offline scheduling — Application: **EMAN**



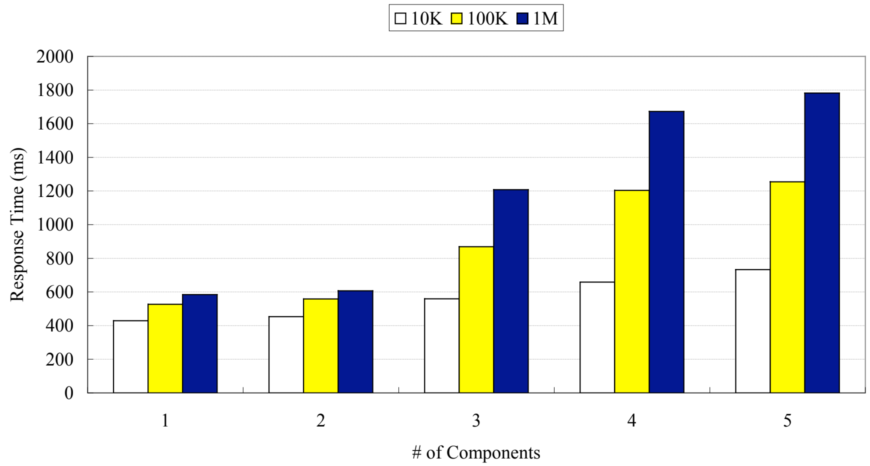
"Scheduling Strategies for Mapping Application Workflows onto the Grid"

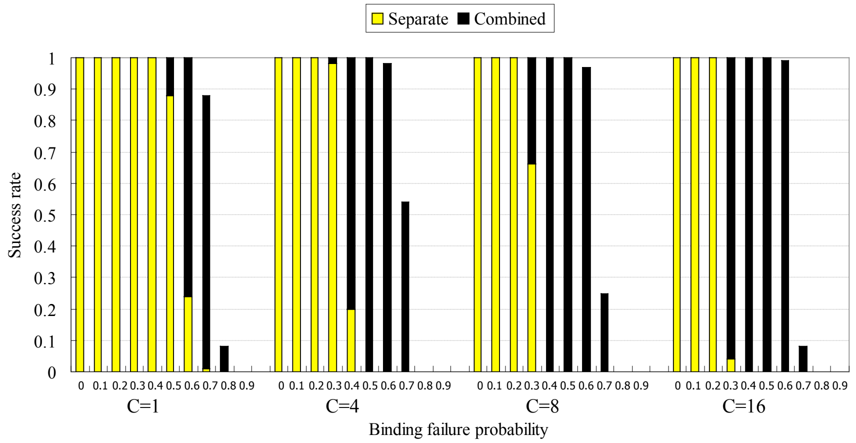
HPDC'05

Scheduling

- Scheduling for very large ensembles (10,000's of machines from vgES) and large workflows on modest ensembles of computers (from vgES)
  - Current: Quadratic in # of resources
- Research Question:
  - How to scale up to large numbers of resources and maintain quality of scheduling?
- Strategy:
  - Resource Classification and 2-level Scheduling strategy 1) over those Classes and 2) within the class
- Challenge Problem:
  - Schedule for TeraGrid on minimal number of nodes > any single resource

vgES Scalable Selection





- Results show vgFAB an return high quality resources in a few seconds; Variety of vgDL Complexities; Grid Environment of 1M distinct resources (SC05 Submission)
- Research Issue: How does this Scale with vgDL request complexity? What complexity do applications really need?
- Strategy: Determine what is Realistic vgDL
  - Work with Applications to develop realistic vgDL workload
  - Evaluate Scalable Selection with vgDL workloads

VG Binding, Operations, Information

- **Current: Localized vgES/vgFAB implementation – Limits # Resource Managers, Nodes**
  - Supports dozens of resource managers, maybe 1000's of Resources
- **Research Issue: How to scale up binding of large numbers of resources, VG's, Attribute Requests?**
- **Strategy: Distributed vgES/vgFAB Implementation**
  - Distributed Responsibility and Decision Making
  - Challenge to Maintain Quality of Results; Coherent Grid View

vgES Scalable Monitoring

- **Current:**
  - —HAPI scalable measurement and statistical sampling (health and performance); stratified sampling
  - —vgAgent information services gateway can trades freshness for scalability
- **Research Issues: What is tradeoff of accuracy versus scalability? What is the “right” hierarchical decomposition?**
- **Strategy: Compose a set of flat representations into a hierarchy**
  - —Derive equivalence classes of bags from virtualization
  - —Across hierarchy levels, topology reflects proximity