VGrADS Programming Tools Research: Vision and Overview

Ken Kennedy

Center for High Performance Software

Rice University

http://vgrads.rice.edu/site_visit/april_2005/slides/kennedy-tools

VGrADS
Virtual Grid Application Development Software Project

Programming Tools Contributors

- Rice University
  - Ken Kennedy, PI
  - Keith Cooper, Chuck Koelbel (co-PIs)
  - Rob Fowler, Mark Mazina, John Mellor-Crummey, Tim Harvey (faculty/staff)
  - Raj Bandyopadhyay, Adam Bordelon, Jason Eckhardt, Anshuman Das Gupta, Anirban Mandal, Gabriel Marin, Ryan Zhang (students)
- University of Houston
  - Lennart Johnson, co-PI
  - Nils Smeds (staff)
  - Bo Liu, Mitul Patel (students)
- University of California San Diego
  - Fran Berman, Henri Casanova, Andrew Chien (co-PIs)
  - Yang-Suk Kee, Ken Yocum (staff)
  - Jerry Chou, Richard Huang, Dennis Logothetis (students)
- University of California Santa Barbara
  - Rich Wolski (co-PI)
  - Graziano Obertelli (staff)
  - Dan Nurmi (student)
- University of North Carolina
  - Dan Reed (co-PI)
  - Lavayna Ramakrishnan (staff)
  - Emma Buneci, Min Lim (students)
- University of Tennessee
  - Jack Dongarra (co-PI)
  - Asim YarKhan (staff)
  - Zhiao Shi (student)
- University of Southern California
  - Carl Kesselman (co-PI)
  - Ewa Deelman, Gurang Mehta (staff)
  - Gurmeet Singh (student)

Note : Baylor College of Medicine collaborators not included

VGrADS
Virtual Grid Application Development Software Project

Programming Tools

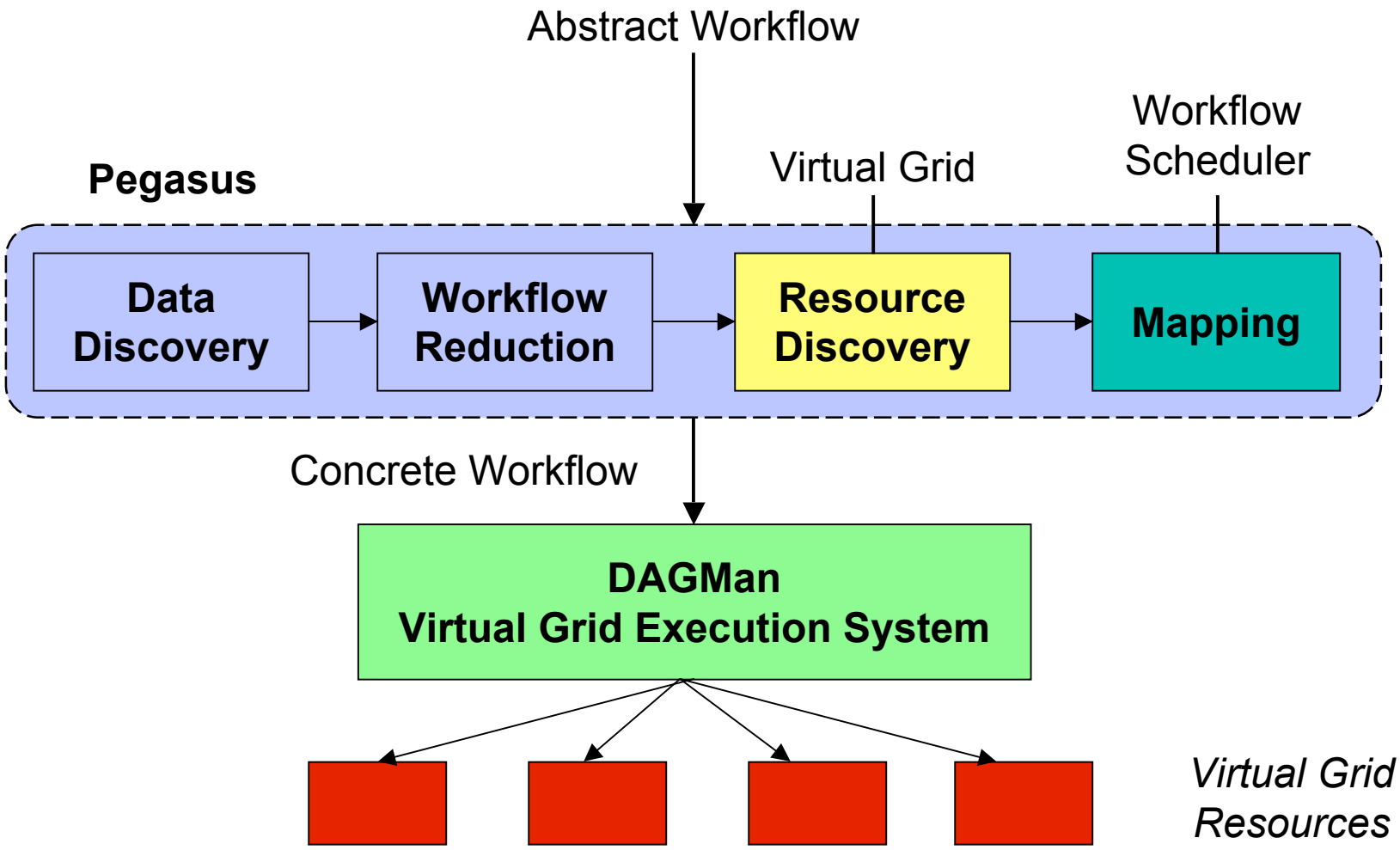
Focus: Automating critical application-development steps:

- Initiating and managing application execution
  - Optimize and launch application on heterogeneous resources
  - Support for fault tolerance and rescheduling/migration
- Scheduling application workflows
  - Whole-workflow scheduling using performance models
- Constructing performance models
  - Automatically from application binaries
    - Cross-platform modeling
- Building workflow graphs from high-level scripts
  - Examples: Python (EMAN), OGRE (LEAD), Matlab

Managing Application Execution

- Vision: Transparent to the User with Fault Tolerance
  - Binaries shipped or preinstalled
    - Reconfigured where necessary
  - Data moved to computations
  - Support for fault tolerance
  - Support for rescheduling, migration and restart
- Research
  - Separation of concerns between workflow management and virtual grid
  - Support for fault tolerance and rescheduling/migration
    - Checkpointing, load projection, performance modeling
  - Platform-independent application representation

Application Initiation and Management



Support for Fault Tolerance and Rescheduling

- Fault tolerance
  - Workflows: Recovery via Pegasus mechanisms
    - Issue: support for registration of completed steps in vgES
  - MPI steps: Disk-free checkpointing
    - Dongarra talk
  - Checkpoint scheduling (Nurmi poster ❶)

    ❶ "Optimal Checkpoint Scheduling using Automatic Resource Characterization" by Dan Nurmi (UCSB)
- Rescheduling
  - Mechanisms for monitoring and extending virtual grids in vgES
    - Under development (Huang poster ❷)
  - Issue: Determining whether rescheduling will be profitable
    - Models to project performance on current and alternative resources

❷ "Scheduling Compute Intensive Applications in Volatile, Shared Resource (Grid) Environments" by Richard Huang (UCSD)

Platform-Independent Applications

- Supporting a Single Application Image for Different Platforms
  - Translation and optimization tools that map the application onto the hardware in an effective and efficient way
    - At the high level, resource allocation and scheduling
    - At the low level, optimization, scheduling, and runtime reoptimization
  - We are working with the LLVM system (from Illinois)
    - Produces good code for a variety of hardware platforms
- Run-time Reoptimization
  - The Idea: In response to poor performance, reoptimize
  - The Vision: To reduce the runtime cost of reoptimization, move analysis and planning to compile time
    - Example: alternative blocking strategies for a loop nest, with shift triggered by excessive cache misses

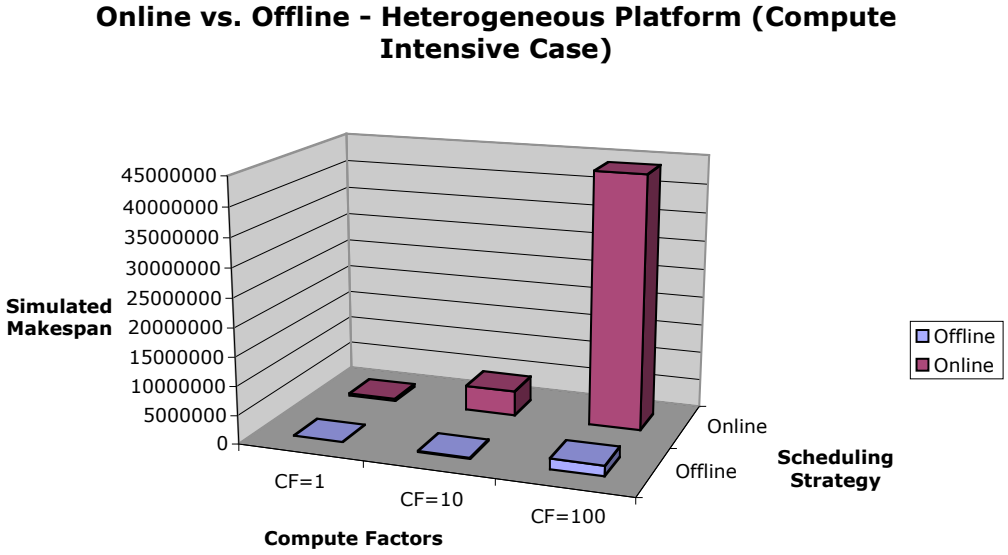
Scheduling Workflows

- Vision:
  - Application includes performance models for all workflow nodes
    - Performance models automatically constructed
  - Software schedules applications onto Grid in two phases
    - Virtual grid requirement and acquisition
    - Model based scheduling on the returned vGrid
- Research
  - Scheduling strategy: Whole workflow
    - Dramatic makespan reduction (see Mandal-Liu poster ❶)
  - Two-phase scheduling
    - Exploring trade-offs
    - Expectation: dramatic reduction in complexity with little loss in performance

❶ "Performance Model-Based Scheduling of EMAN Workflows" by Anirban Mandal (Rice) and Bo Liu (U Houston)

Workflow Scheduling Results

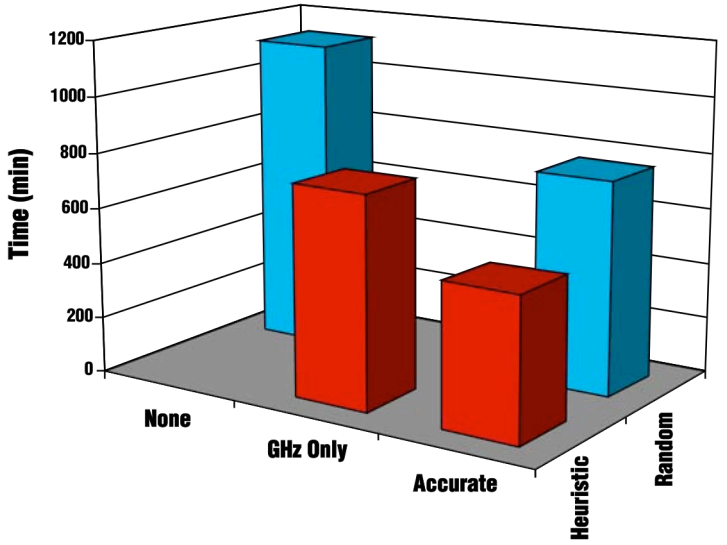
Dramatic makespan reduction of *offline* scheduling over *online* scheduling — Application: **Montage**



"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"

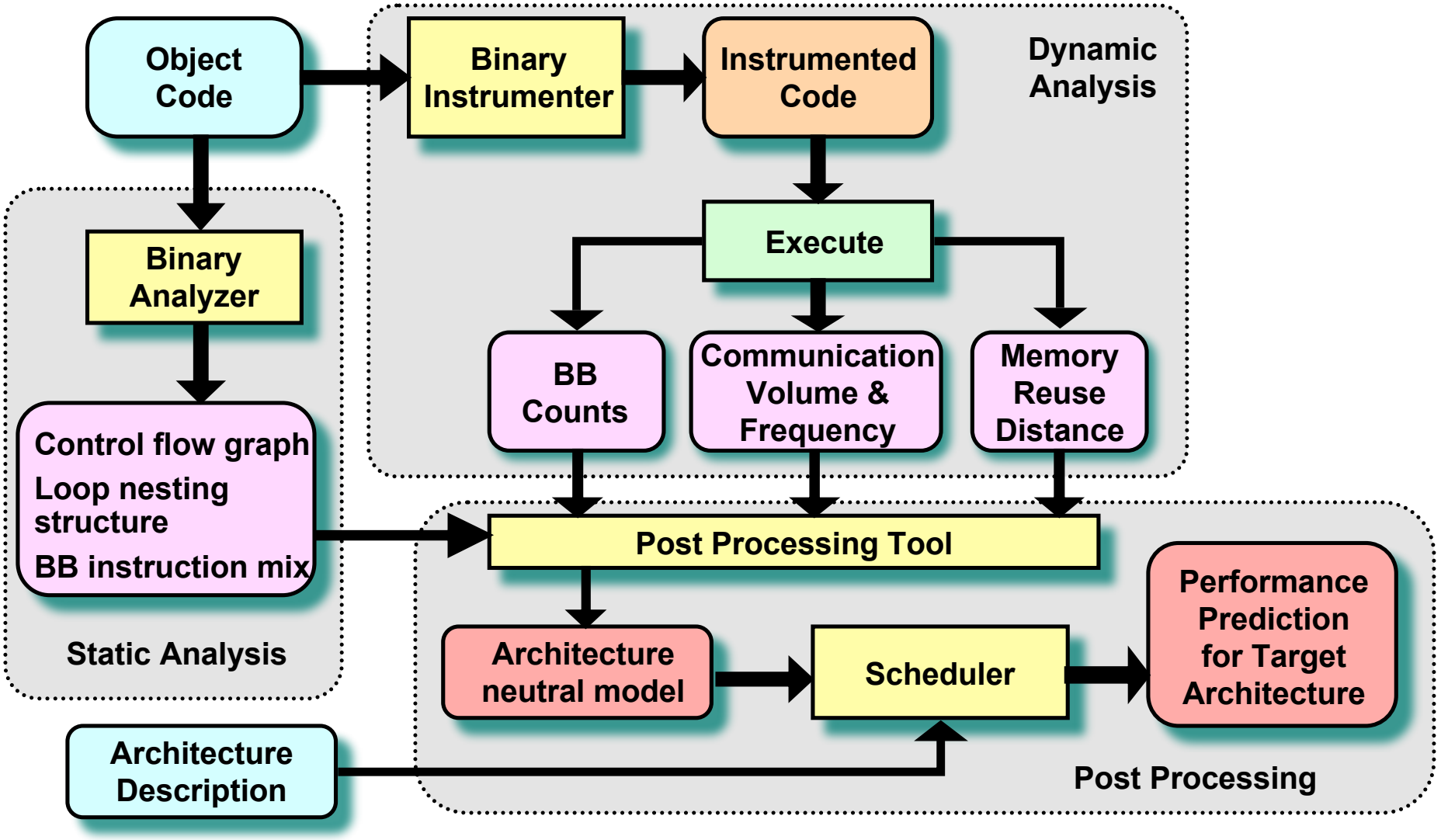
HPDC'05

VGrADS
Virtual Grid Application Development Software Project

Performance Model Construction

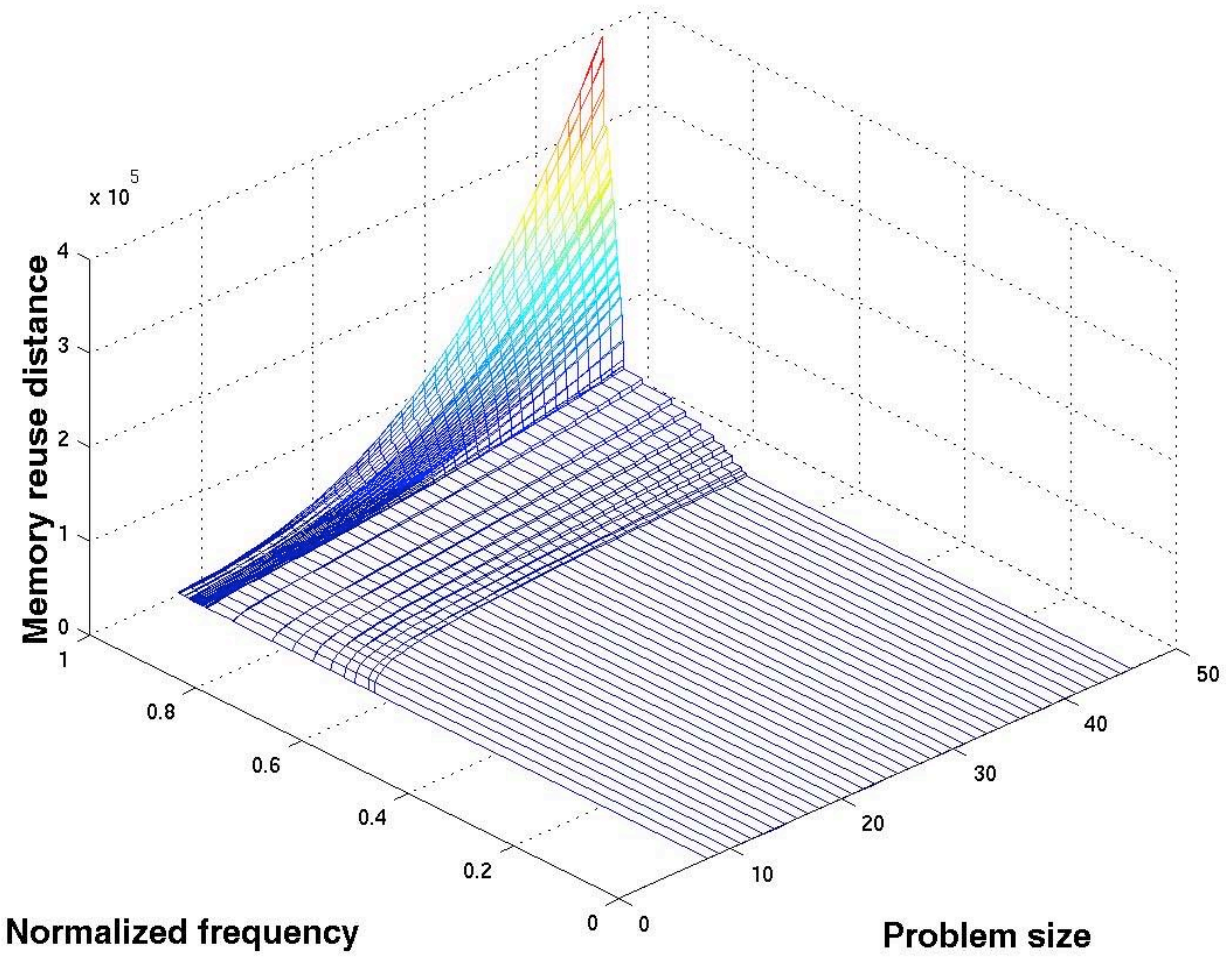
- Vision: Automatic performance modeling
  - From binary for a single resource and execution profiles
  - Generate a distinct model for each target resource
- Research
  - Uniprocessor modeling
    - Can be extended to parallel MPI steps
  - Memory hierarchy behavior
  - Models for instruction mix
    - Application-specific models
    - Scheduling
- Posters:
  1. "Performance Model-Based Scheduling of EMAN Workflows" by Anirban Mandal (Rice) and Bo Liu (U Houston)
  2. "Scalable Cross-Architecture Predictions of Memory Latency for Scientific Applications" by Gabriel Marin (Rice)

Performance Prediction Overview

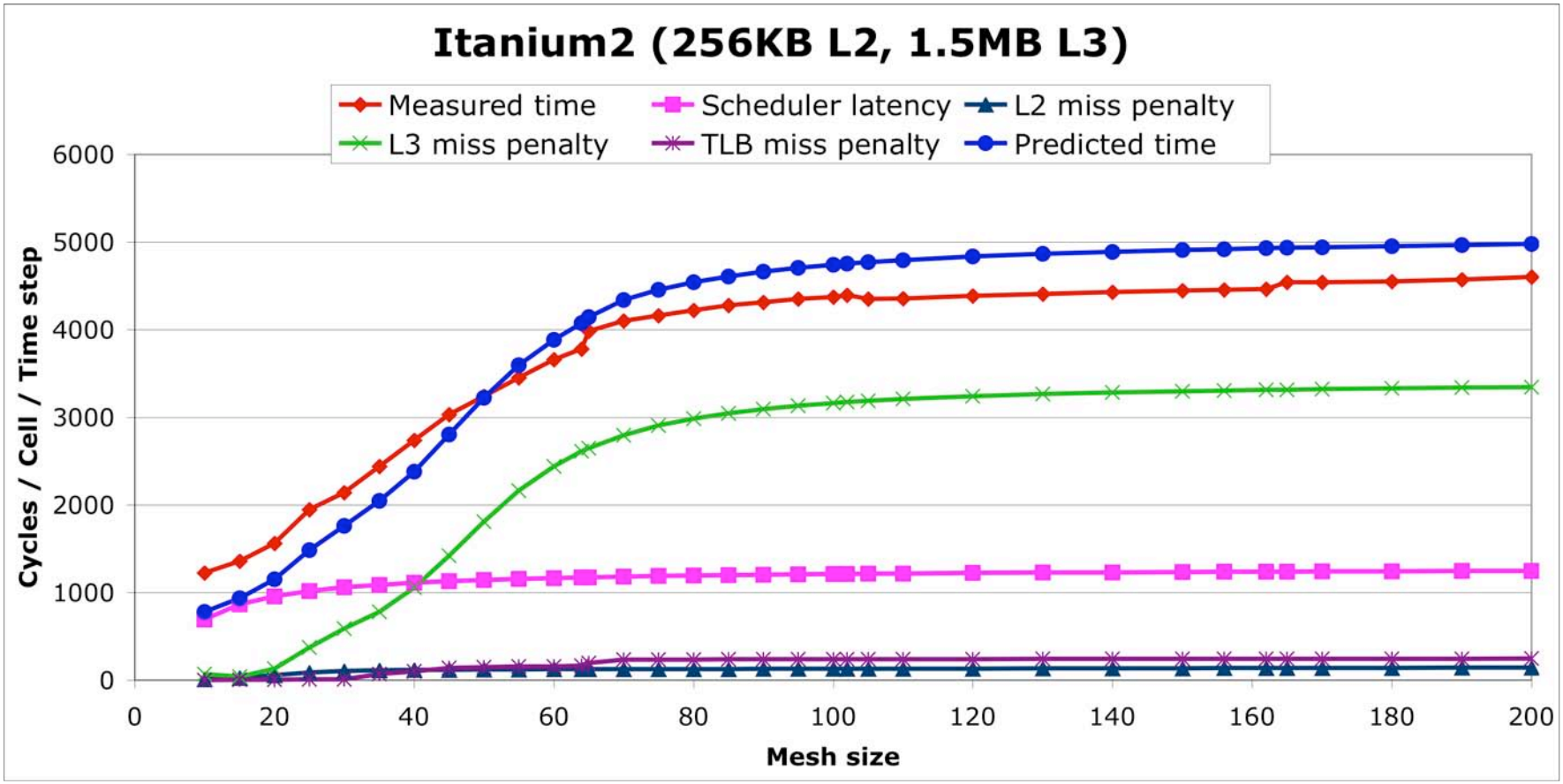


VGrADS
Virtual Grid Application Development Software Project

Modeling Memory Reuse Distance



Execution Behavior: NAS LU 2D 3.0



Building Workflow Graphs

- Vision:
  - Application developer writes in scripting language
    - Examples: Python (EMAN), OGRE (LEAD), Matlab, S-PLUS/R
    - Components represent applications
  - Software constructs workflow and data movement
- Research
  - Related project: Telescoping languages
    - Compilation of Matlab and R
    - Type analysis
    - Pre-optimization of components for different contexts
  - Plan: Harvest this work for Grid application development
    - Just getting started

Summary

- Making Grid Applications Easy to Develop
  - Abstract interfaces (e.g., scripts)
  - Effective (and easy) application scheduling
  - Automatic performance model construction
- Building on Virtual Grid Abstraction
  - Easy application launch, monitoring, and management
- Driven by Real Application Needs
  - Initial foci: EMAN, LEAD, and Montage