



# Feedback-Controlled, Virtualized Resource Sharing for Predictable eScience

Sang-Min Park  
Dept. of Computer Science,  
University of Virginia  
(Advisor: Prof. Marty Humphrey)



Computer Science  
*at the* UNIVERSITY of VIRGINIA



# Emerging HPC Applications

- **Dynamic Data Driven Applications**
  - Linked Environments for Atmospheric Discovery (LEAD)
  - Earthquake modeling
  - Medical modeling for patient-specific treatment
  - Coastal hazard prediction, disaster planning, etc
- **Predictability** is key to success!





# Why existing works fail

- Standard: Batch-mode, space-sharing
- Goal: predictable computation
- Existing Research
  - Advance reservation
  - Queue wait-time prediction

## Problem Statement:

Space-sharing and related provisioning model cannot respond to adaptive computing demand predictably





# Research Direction

- Time-sharing principle
  - Focus is to control job's running time (not waiting time)
- Virtualization
  - Exploit VM's isolation / run-time reconfiguration
- Formal Control Theory
  - Provable correctness on deadline guarantee





# Contributions

- Bring community the advantage of time-shared HPC infrastructure
- Introduce control theory as a systematic, practical tool for HPC researcher
- Solve limitation of distributed (meta) job scheduling (e.g., workflow)





# Key Metrics and Assumptions

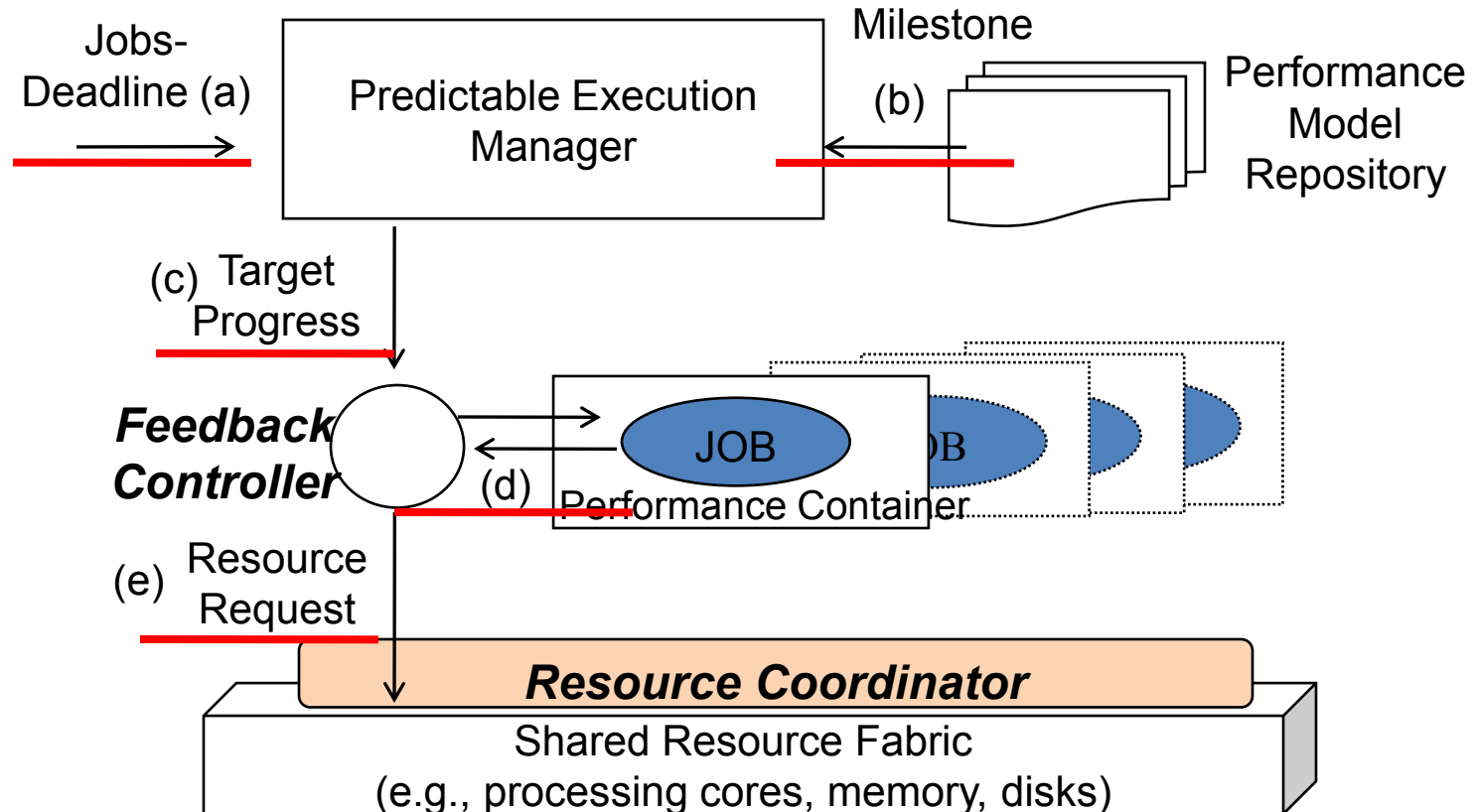
$$\frac{\textit{Milestone}}{\textit{Deadline}} = \textit{Progress}$$

- Milestone
  - Explicit overall computation
  - Examples
    - Total # of floating point operation
    - # of data file to process, # of query to process
- Progress
  - Computation in unit time (how fast)
  - Examples:
    - # floating point/sec, files/min., queries/min.



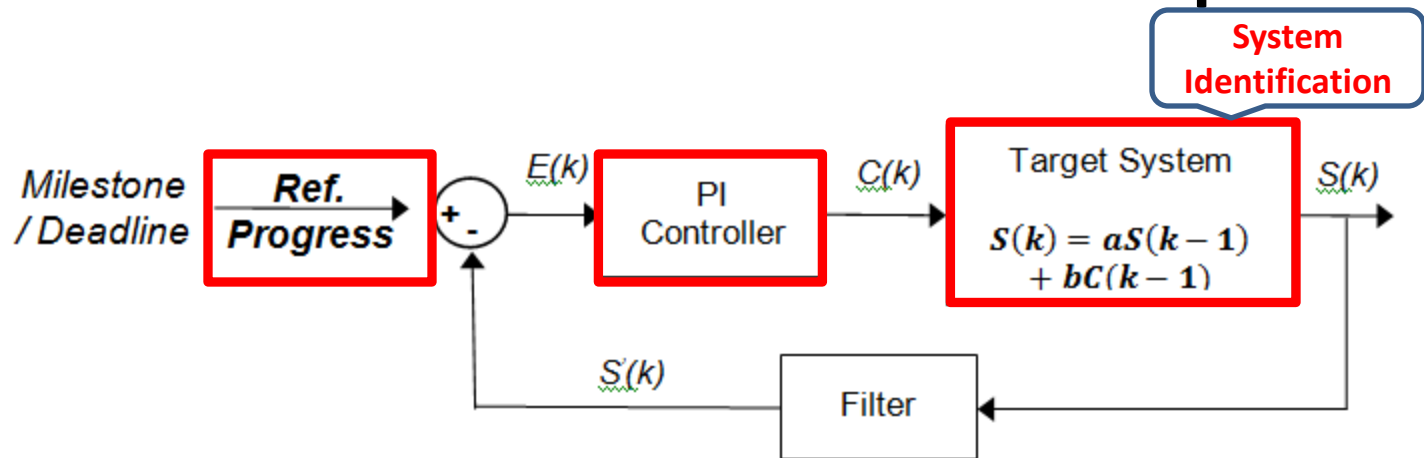


# Compute Throttling Framework





# Feedback Control Loop



- Closed-loop control copes with disturbances (e.g., model error, severe I/O): Cruise-control of computation
- Control theory provides provable properties of closed-loop system



# Early Results

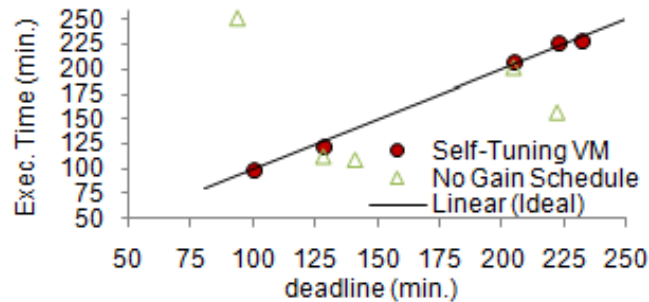


Figure 6(a) ADCIRC

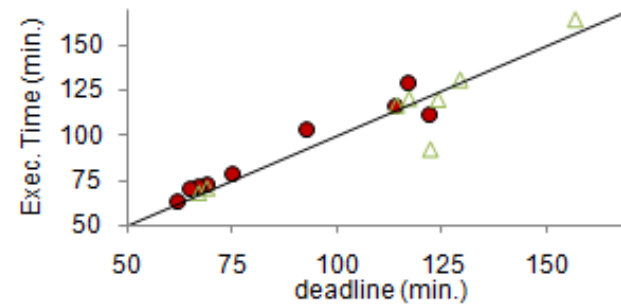


Figure 6(b) BLAST

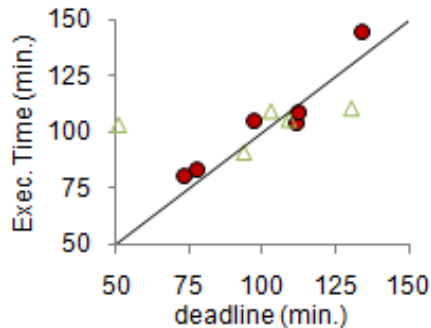


Figure 6(c) mProject

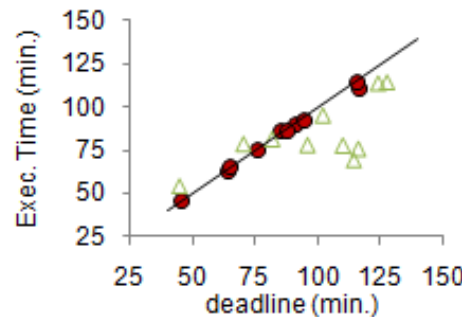


Figure 6(d) OpenLB

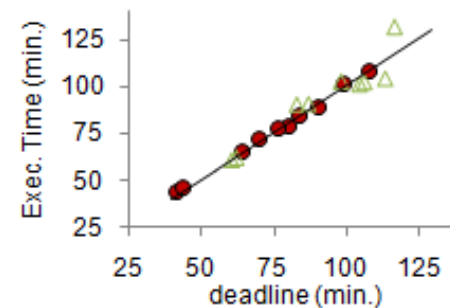


Figure 6(e) WRF

Meet the deadline with less than 3% errors  
in general while achieving > 90% utilization



# Research Components

- Feedback-controlled VM: SC'08 paper
- Self-tuning Controller: Make feedback control usable for ordinary scientists
- Scheduling scientific workflow:
  - IPDPS'08 paper: solving predictable communication problem
  - See how predictable comp.&comm. enables better workflow schedule





# Question and Feedback?

Sang-Min Park

sangmin@email.virginia.edu



Computer Science  
*at the* UNIVERSITY of VIRGINIA