On Dynamic Resource Management Mechanism using Control Theoretic Approach for Wide-Area Grid Computing

Hiroyuki Ohsaki

Graduate School of Information Science & Technology
Osaka University, Japan

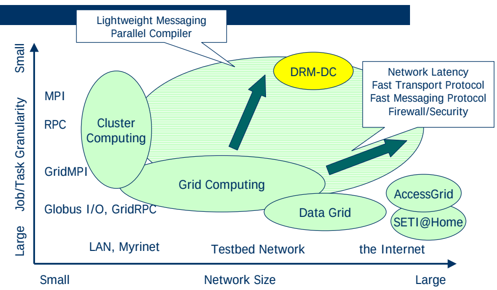
Contents

- Introduction
- Resources management in Grid computing
- Modeling as feedback control
- Dynamic resource management mechanism DRM-DC
- Performance evaluation using simulation
- Conclusion

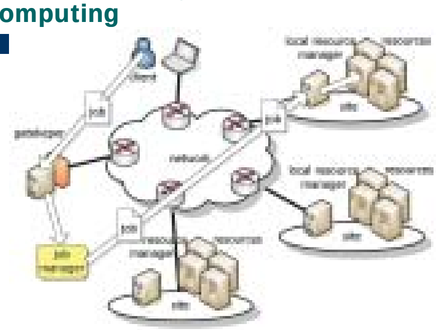
Background

- Emergence of Grid computing
  - Integrate **geographically distributed** resources
  - Enable large-scale scientific computations
- Problems in wide-area Grid computing
  - **Variation in the amount of available resources**
    - Resources are shared by multiple users
    - e.g., computing resources, network, disk space
  - **Transfer delay** between sites is non-negligible
    - Round-trip time might be > 100ms
- Dynamic resource management is required

Grid Computing: Current and Future



Resource Management in Grid Computing



GRAM (Grid Resource Allocation and Management)

- A resource management component of Globus toolkit
- GRAM building blocks
  - Client
    - Create job, and ask gatekeeper for its execution
  - Gatekeeper
    - Authenticate user, and create job manager
  - **Job manager**
    - Assign job to local resource manager
  - **Local resource manager**
    - Execute job using site's available resource

Requirements for Resource Management in Grid Computing

- Objectives
  - Adjust **the amount of jobs injected** into a site...
    - According to the amount of available resources
- Desired characteristics
  - Steady-state performance
    - **High resource utilization**, fast job execution
  - Transient-state performance
    - **Fast convergence**
  - Stability, robustness, flexibility...

Problems in Wide-Area Grid Computing

- **Variation in the amount of available resources**
  - Simple open-loop control is insufficient
  - **Closed-loop control** using feedback information is indispensable
- **Transfer delay** between sites is non-negligible
  - Difficult to realize stability and fast convergence
  - **Feedback delay** must be taken account of

Research Objectives

- Propose **dynamic resource management mechanism**
  - DRM-DC (Dynamic Resource Management with Delay Compensator)
  - Feedback-based control with **a delay compensator**
  - Realize high **steady-/transient-state performance**
- Performance evaluation of DRM-DC
  - Implement DRM-DC in Simgrid simulator
  - Comparison of DRM-DC with PI controller
  - Show DRM-DC's effectiveness in **WAN environment**

Modeling Assumptions

- **Parameter-sweep applications**
  - Job manager **always has jobs** to execute
    - Job = multiple tasks
  - **Task granularity is small**
    - i.e., task execution time << transfer delay
- **Transfer delay is constant**
  - Network between sites is not heavily congested

Modeling Site (Plant)

- Can be modeled as **a single queue**
  - Site processes assigned tasks one by one
    1. Receive tasks from job manager
    2. Queue them in **buffer** of local resource manager
    3. Process tasks in the buffer one by one
- Continuous-time system model
  - Input *u(t)*: task injection rate from job manager
  - Output *x(t)*: the number of tasks in the buffer

$$x(t) = \left[\int_0^t \{u(s - \tau(s)) - \mu(s)\} ds\right]^+$$

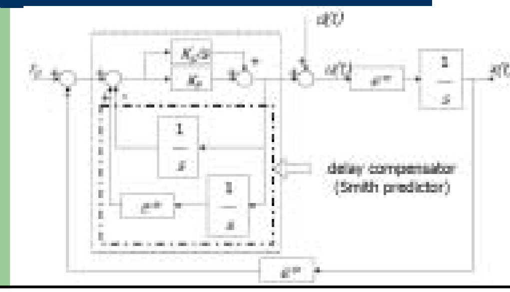
Modeling Job Manager (Controller)

- Can be modeled as **a controller**
  - Job manager assigns tasks to each site
    1. Receive **feedback information** from sites
    2. Adjust the amount of **task assignments** to sites
- Continuous-time system model
  - Input *x(t)*: the number of waiting tasks in site
  - Output *u(t)*: task injection rate to site

Dynamic Resource Management Mechanism DRM-DC

- DRM-DC (Dynamic Resource Management with Delay Compensator)
- **Controller running on job manager**
  - Input: the number of tasks waiting in site
  - Output: task injection rate to site
- DRM-DC control objective
  - Keep the number of tasks in the buffer **at target level**
    - **Avoid overload** and **realize high utilization**
- PI (Proportional Integral) controller with **a delay compensator (Smith Predictor)**

Block Diagram of DRM-DC



Performance Evaluation

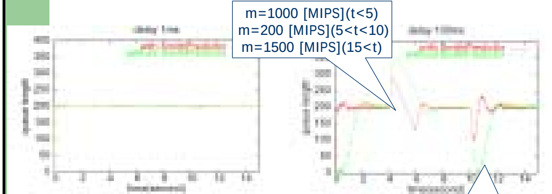
- Modified a discrete-time simulator Simgrid
  - Implement **discrete-time version of DRM-DC** (controls at every constant interval T)
- Simulation conditions
  - Parameter-sweep applications
  - Number of sites: 10
  - Network: **LAN** (1 [ms]), **WAN** (100 [ms])
- Performance metrics
  - **Queue dynamics** (i.e., the number of tasks in buffer)

Simulation Scenario: Parameter Configuration

- Simulation scenarios
  1. **Varying amount of available resources**
  2. Varying transfer delay between job manager and site

		LAN-S2	LAN-S20	WAN-S2	WAN-S20
task size S	[MI]	2	20	2	20
transfer delay τ	[ms]	1.0	1.0	100.0	100.0
control interval T	[ms]	2.0	20.0	2.0	20.0
target queue length n_1		200	20	200	20

Simulation Result (Queue Dynamics for Varying Available Resources)



Simulation Result (Rise Time, Overshoot, Settling Time)

		τ [ms]	S [MI]	rise time [s]	overshoot [%]	settling time [s]
LAN-S2	DRM-DC	1.0	2	0.016	0.5	0.012
LAN-S2	PI	1.0	2	0.016	0.5	0.008
LAN-S20	DRM-DC	1.0	20	6.4	5	8.8
LAN-S20	PI	1.0	20	8.8	9	9.8
WAN-S2	DRM-DC	100.0	2	0.20	46	0.11
WAN-S2	PI	100.0	2	1.6	79	1.7
WAN-S20	DRM-DC	100.0	20	0.16	55	8.3
WAN-S20	PI	100.0	20	1.62	95	1.6

DRM-DC shows significantly shorter rise time/settling time

DRM-DC shows smaller overshoot

Conclusion and Future Work

- Proposed **dynamic resources management mechanism**
  - Designed for **wide-area Grid computing**
  - Utilize **a delay compensator** (Smith predictor)
- Verified its effectiveness by simulation
  - Realize high steady-/transient-state performance
  - Effective **in WAN environment** with a large delay
- Future work
  - Improvement of stability, robustness, and flexibility
  - Implementation in Globus toolkit