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

Hiroyuki Ohsaki, Soushi Watanabe, and Makoto Imase
Department of Information Networking
Graduate School of Information Science and Technology, Osaka University
1-5, Yamadaoka, Suita, Osaka 565-0871, Japan
E-mail: {oosaki,s-watanb,imase}@ist.osaka-u.ac.jp

Abstract*— In recent years, Grid computing that integrates geographically distributed computing resources through communication networks captures the spotlight. Unlike parallel computing using conventional cluster computer systems, wide-area Grid computing must resolve the following issues for effectively utilizing geographically distributed computing resources. First, since computing resources are shared by multiple users, the amount of available resources in a site changes over time. Second, since sites are geographically distributed, the transfer delay between sites cannot be neglected for computing resource management. For utilizing computing resources effectively, the amount of jobs injected into a site must be dynamically controlled according to the dynamically changing amount of available resources in sites. However, the transfer delay of a network is significant, so that it is not trivial for a resource allocation controller to quickly and dynamically adopt to the change in the amount of available resources in sites. In this paper, a dynamic resources management mechanism for wide-area Grid computing called *DRM-DC (Dynamic Resource Management with Delay Compensator)* based on control theory is proposed. The main feature of our DRM-DC is that it realizes high steady state and transient state performance in wide-area Grid computing using a delay compensator called *Smith predictor*. Moreover, several discrete-time simulations using a Simgrid simulator are performed, and the effectiveness of our DRM-DC is demonstrated.

I. Introduction

In recent years, Grid computing, in which geographically distributed computer resources are used in an integrated manner through a network, has garnered a good deal of attention [1, 2]. Grid computing is an approach to use various computers' free resources that had been little used in the past. Grid computing using geographically separate computer resources in particular is called "wide-area Grid computing". With wide-area Grid computing, various problems occur, as opposed to parallel computation using conventional cluster computers.

The first problem is that the amount of available resources at a site varies with time. With Grid computing in general, resources are shared by multiple users, so the amount of available resources at a site varies with time. As an example, computer resources at a site change in accordance with the number of jobs performed at the same time. In addition, network resources change in accordance with the communication status of other users and disk resources change when information is saved or deleted by other users.

The second problem is that the sites are distributed over a wide geographic area, so the transfer delay between sites cannot be ignored. This is a problem particular to wide-area Grid computing. As an example, the round-trip time between sites in wide-area network is an extremely large value such as above 100 ms compared to a LAN environment like that with cluster computers. Thus, ascertaining the condition of free resources at each site in real-time is difficult.

Thus, wide-area Grid computing has problems in that the amount of available resources in varies with time and the network's transfer delay cannot be ignored. To make effective use of resources such as computer resources, the amount of jobs submitted to a site must be adjusted dynamically to match the amount of available resources that varies with time [3, 4]. However, the network's transfer delay is extremely long, so changes in available resources at a site cannot be followed with simple resource allocation control.

Thus, this paper proposes dynamic resource management with a delay compensator (DRM-DC) based on control theory. DRM-DC uses feedback control to follow variations in the amount of available resources at a site. In other words, it intends to effectively use resources at a site through feedback of a site's usage. Characteristics of the proposed DRM-DC are the achievement of high steady-state characteristics and transient-state characteristics in wide-area Grid computing with a substantial transfer delay through use of a delay compensator (a Smith predictor). A delay compensator is a technique that can counteract the effects of control delay (allowing control delay to be considered zero) even in conditions with a substantial control delay through use of a plant model [5]. Through use of a delay compensator, resources are prevented from being overloaded and left free and available effective use of resources at a site is encouraged even in wide-area Grid computing with a substantial transfer delay.

This paper models resource management in the Grid as a control system for a continuous-time system. A site sequentially processes a series of allocated tasks, so a site is modeled as a single queue. In reality, this is modeled with an integrator where input is the task arrival rate from a job manager and output is the number of tasks in the buffer for the site's local resource manager. In addition, the job manager adjusts the amount of tasks submitted to a site based on the feedback information from the site. Accordingly, the job manager is modeled with a proportional integral (PI) controller where input is feedback information from a site (the number of tasks in the buffer) and output is the task submission rate. This paper deals with an environment with a single resource manager and multiple sites, and the queue length (the number of tasks in the buffer) for each site's local resource manager is indicated as feedback information to the job manager. In addition, discrete time simulation is performed using the Simgrid simulator [6] and the effectiveness of the proposed DRM-DC is verified. Specifically, the proposed DRM-DC achieves satisfactory steady-state characteristics and transient-state characteristics in wide-area grid computing as indicated by comparison of the proposed DRM-DC and a PI controller not using a delay compensator.

The composition of this paper is as follows. First, Section II presents related research. Section III explains resource management in Grid computing. Section IV explains how resource management in Grid computing can be modeled as feedback control. Section V explains the algorithm for DRM-DC using a delay compensator as proposed by the authors. Section VI discusses verification of the effectiveness of the proposed resource management DRM-DC through simulation testing. Finally, Section VII describes this paper's conclusions and future topics for research.

II. Related research

This paper resolves problems in wide-area Grid computing — the amount of available resources varying with time and the inability to ignore the network's transfer delay — through use of a delay compensator in control theory. Attempts to resolve feedback control in a network with substantial transfer delay through a delay compensator are cited in the literature [7]. As an example, the literature [7] proposed a technique using a delay compensator (Smith predictor) for rate control in an ATM (Asynchronous Transfer Mode) network. The queue length (number of packets in a buffer) of a bottleneck router is used as feedback information and the queue length of the bottleneck router is controlled using simple proportional (P) control. In this instance, a Smith predictor was used to compensate for control delay due to the transfer delay of the links. However, there is a problem with this in that the controller is a simple P controller and steady-state deviation occurred (the queue length of the bottleneck router in a steady state does not match the target queue length, a control target). However, the DRM-DC proposed in this paper achieves satisfactory steady-state characteristics and transient-state characteristics in wide-area Grid computing through use of a PI controller with an integral element to resolve steady-state deviation.

III. Resource management in Grid computing

First, an overview of resource management in Grid computing is explained. Here, an explanation is given with component GRAM (Grid Resource Allocation and Management) for resource management as is included in the Globus Toolkit [8] as an example.

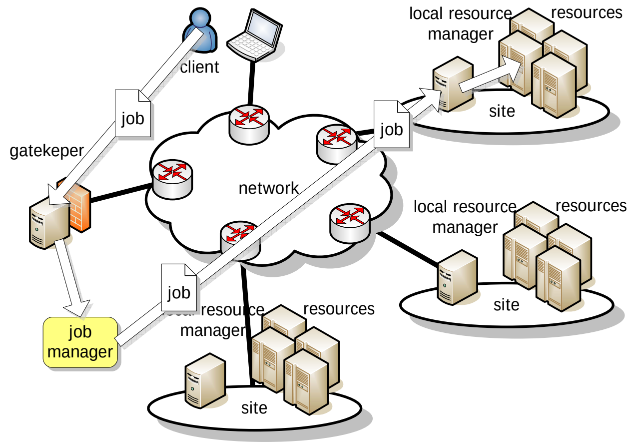

Fig. 1. Resource management in Grid computing

GRAM consists of a client, gatekeeper, job manager, local resource managers, and sites (Fig. 1). A site means the gathering of resources (e.g., cluster computers) assigned to a virtual organization. The flow of job execution in GRAM will now be explained. First, the client generates a job and requests that the job be executed though the gatekeeper. The gatekeeper authenticates the client and generates a job manager to actually manage execution of the job. The job manager allocates the job to local resource managers at each site and supervises the state of the job's execution. The local resource manager executes the assigned job using computer resources at the site and returns the executed results to the job manager.

The goal of resource management in Grid computing is dynamic adjustment of the amount of jobs submitted to a site to match time the amount of each site's available resources, which vary with time [3, 4]. In other words, this is control of which portion of a job is allocated to what site's local resource manager by the job manager generated by the gatekeeper based on the amount of each site's available resources. This paper deals with instances of executing a job consisting of numerous independent processes (called tasks) like those for a parameter search application [9]. In other words, it deals with instances where a job generated by a client consists of numerous tasks with a small degree of granularity.

Next, what characteristics are important in resource management in Grid computing will be discussed. First, a system's behavior in a stable state (steady-state characteristics) is probably important. Being able to maintain a high level of usage without having the site's resources overloaded or idle is preferable. In addition, having a short period of time from when a job is allocated to a site until the execution of the job is actually complete is preferable.

With Grid computing, the amount of a site's free resources varies with time. Accordingly, behavior until a system is stable (transient-state characteristics), or the rise time, is also important. In addition, control is performed based on the amount of a site's available resources in an environment with a substantial delay time, so whether or not a system is stable (stability) if some time passes will also be an important performance indicator.

Moreover, the occurrence of network device and computer faults is also possible in Grid computing. Accordingly, whether the system will operate normally even if a fault occurs (robustness) is important. As an example, being able to have a system operate normally and maintain stability even if the network delay increases temporarily due to a fault in a network device is preferable.

In Grid computing in general, factors such as a site's throughput and a network's delay time are not homogeneous. Accordingly, being able to achieve stability and robustness (flexibility) in a heterogeneous environment, i.e., under various parameter conditions, is preferable.

IV. Modeling of resources as feedback control

Below, how resource management in Grid computing is modeled as a feedback control is explained.

In Grid computing, a site's resources are shared by multiple users, so the amount of available resources varies with time [1, 2]. With resource management for Grid computing, predicting variations in the amount of available resources and scheduling accordingly are difficult. Accordingly, dynamic control to match variations in the amount of available resources must be performed. In other words, simple open-loop control is insufficient for resource management for Grid computing, and closed-loop control using feedback information from a site becomes essential.

In addition, site resources are distributed over a wide geographic area in wide-area Grid computing, so the transfer delay increases and transfer delay cannot be ignored in resource management. Because transfer delay is substantial, time is taken for the job manager to obtain a site's resource information and for a job allocated to a site by the job manager to arrive at the site. Accordingly, stability and robustness cannot be achieved with simple feedback control. A transfer delay like this must be dealt with by resource management for wide-area Grid computing, i.e., control must consider feedback delay.

Thus, resource management in Grid computing is modeled below as a control system for a continuous-time system.

A network consisting of multiple sites as in Fig. 1 is considered. Here, the focus is on a site. A site's available resources (the number of tasks that can be processed per unit of time) at time t are denoted as $\mu(t)$ and the transfer delay between the job manager and a site is denoted as $\tau(t)$. In addition, the task submission rate from the job manager to a site (the number of allocated tasks per unit of time) at time t is $u(t)$. The number of tasks (queue length) stored in the buffer of a site's local resource manager at time t is $x(t)$.

First, the site sequentially processes a series of allocated tasks, so the site can be modeled as a single queue. Tasks submitted by the job manager are temporarily stored in the buffer of the local resource manager. The local resource manager executes tasks stored in the buffer in sequence in accordance with the amount of the site's available resources. Accordingly, site can be modeled as a queue where the processing speed is $\mu(t)$. Accordingly, the site corresponds to a plant when considered from the perspective of feedback control. When site is modeled as a control system for a continuous-time system, input is the task arrival rate from the job manager to a site $u(t)$ and output (feedback information sent to the job manager) is the number of tasks in the buffer of the site's local resource manager $x(t)$, so it can be modeled with an integrator like that below.

$$x(t) = \left[\int_0^t (u(v-\tau(v)) - \mu(v))dv\right]^+ \quad (1)$$

Here, it is defined as $[x]^+ \equiv \max(0, x)$.

However, the job manager adjusts the amount of tasks submitted to a site based on the feedback information from a site. Accordingly, the job manager can be modeled as a controller where input is the feedback information from a site (the number of tasks in the buffer) $x(t)$ and output is the task submission rate to a site $u(t)$. Accordingly, the job manager corresponds to a controller that adjusts the amount of tasks submitted to a site when considered from the perspective of feedback control.

In other words, resource management in wide-area Grid computing results in a problem in that how should a job manager (controller) be designed under conditions where the amount of available resources varies with time and the network's transfer delay cannot be ignored. Thus, the next section discusses design of a controller DRM-DC that achieves high steady-state characteristics and transient-state characteristics in wide-area Grid computing with a substantial transfer delay through use of a delay compensator (Smith predictor).

V. DRM-DC

This section explains the algorithm for our DRM-DC using a delay compensator. DRM-DC is a controller that operates in the job manager and controls the task submission rate to the local resource manager based on the amount of a site's available resources. In general, multiple sites are managed by the job manager. Thus, DRM-DC performs feedback control independently for each site.

DRM-DC performs closed-loop feedback control with the number of tasks stored in the buffer of a site's local resource manager (queue length) as feedback information. The control target is keeping the queue length for the local resource manager to a fixed value. Thus, it is intended to prevent overload of resources at the site and improve usage of resources at the site. In addition, the time from when the local resource manager submits tasks until execution results for tasks are received by the job manager is reduced by keeping the queue length for the local resource manager short.

If resource management in Grid computing is regarded as feedback control, the plant are resources at a site and the control variable is the task submission rate from the job manager to a site. DRM-DC's problems of the amount of available resources varying with time and the inability to ignore the network's transfer delay are resolved through use

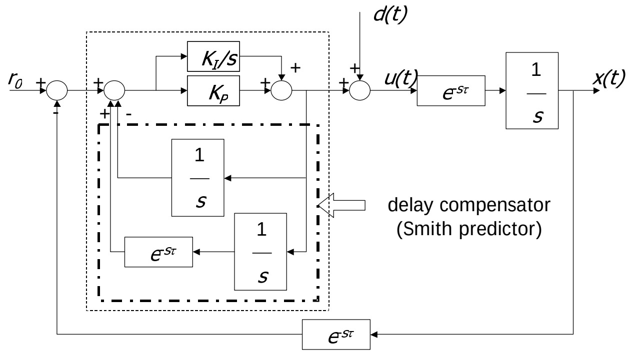

Fig. 2. Block diagram of DRM-DC (Dynamic Resource Management with Delay Compensator)

TABLE I
DEFINITION OF SYMBOLS

r_0	target queue length for the local resource manager
$x(t)$	queue length for the local resource manager at time t
$u(t)$	task submission rate to the local resource manager at time t
$d(t)$	disturbances at time t
K_P	control parameter for DRM-DC (gain for proportional element)
K_I	control parameter for DRM-DC (gain for integral element)
τ	transfer delay between the job manager and a site

of a delay compensator in control theory. DRM-DC prevents deterioration in steady-state characteristics by performing PI control [5] using a delay compensator.

This paper makes the following assumptions: (1) granularity of tasks is sufficiently small compared to the transfer delay between the job manager and a site, (2) propagation delay between the job manager and a site is known and does not vary with time, (3) a site's throughput (the number of tasks that can be processed per unit of time) is constant regardless of the number of tasks in the local resource manager's buffer, and (4) the job manager has jobs (consisting of multiple tasks) that must be constantly executed. A block diagram for DRM-DC is shown in Fig. 2. In addition, a definition of notation used in this paper is shown in Table I.

Next, to what the proposed DRM-DC extent fulfills important characteristics in resource management in Grid computing as mentioned in Section III will be discussed qualitatively. First, DRM-DC controls the number of tasks in the local resource manager's buffer at a fixed value. Accordingly, it can prevent overloading of resources intended for effective use of resources at a site and satisfactory steady-state characteristics can be expected. In addition, the time from when a job is allocated to a site until the execution of the job is actually complete can be kept short by appropriately configuring control target r_0.

The proposed DRM-DC follows changes in the amount of a site's available resources by using the number of tasks in the local resource manager's buffer as feedback information, so satisfactory transient-state characteristics can be expected. However, DRM-DC's effectiveness is largely dependent on the control parameters of the PI controller (K_P and K_I). However, DRM-DC is a PI controller using a delay compensator, so the system's stability can be increased by appropriately configuring control parameters (K_P and K_I).

Robustness and flexibility depend on the accuracy of parameters such as the model of the plant (sites) and transfer delay used by the delay compensator. In reality, an exact numerical model of sites cannot be obtained, and the transfer delay between the job manager and local resource manager can also cause network congestion and vary with time. Thus, to what extent error included in models of items to control and transfer delay has on the DRM-DC's robustness and flexibility must be evaluated. Thus, Section V describes validation of the DRM-DC's effectiveness though simulation testing.

VI. PERFORMANCE EVALUATION BY SIMULATION

This section describes performance evaluation on the proposed DRM-DC through simulation testing.

Simgrid [6], a discrete time simulator for the Grid, was revised and used in simulation testing. Simgrid is a simulator for master-worker applications. In DRM-DC simulation, the job manager is the master and sites (local resource manager and resources) are workers. DRM-DC is control of a continuous-time system, although Simgrid is a discrete time simulator. Thus, a discrete-time DRM-DC was implemented as the Simgrid master so as to perform control per a fixed control interval T. In addition, a FIFO queue was implemented for the buffer size as a Simgrid worker.

Here, 10 sites were managed by a job manager. The simulation was run with a LAN environment ($\tau = 1$ [ms]) and WAN environment ($\tau = 100$ [ms]) as the network environments. In addition, the amount of resources required for each task was equal. Assuming a parameter search application with tasks that the job manager must constantly execute, the floating-point operand required to execute each task (task size) S was 2 [million instructions (MI)] or 20 [MI]. For comparison, simulation of a PI controller not using a delay compensator was also run.

In simulation testing, queue length (the number of tasks in the buffer) for the local resource manager is focused upon as a performance indicator for DRM-DC. In particular, variations in the queue length over time, rise time, overshoot, and settling time are used as performance indicators [5]. Specifically, the time immediately after the start of the simulation from when the queue length is 10% of the queue length target value until it changed to 90% was measured as the rise time for the queue length. To what extent the maximum queue length surpassed the target value for the queue length was measured as overshoot. The time until the queue length stabilized to within 5% of the target value for the queue length was measured as the settling time. Simulation was performed for 15 [s]c and the aforementioned performance indicators were measured.

In this paper, simulation was run with respect to the following two scenarios.

1) Scenario 1: When the amount of a site's available resources varies with time

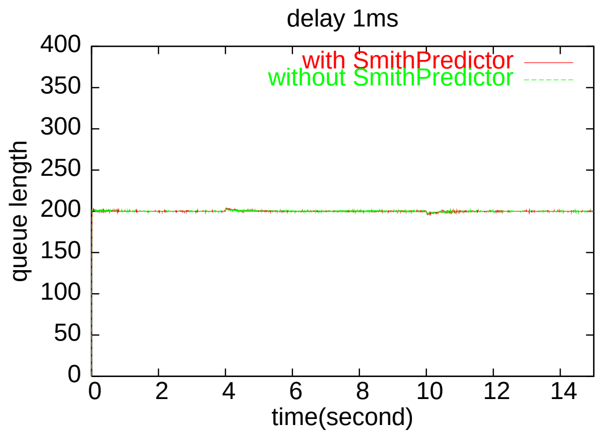

Fig. 3. Case LAN-S2: Queue dynamics for varying amount of available resources in LAN environment ($\tau = 1.0$ [ms], $S = 2$ [MI])

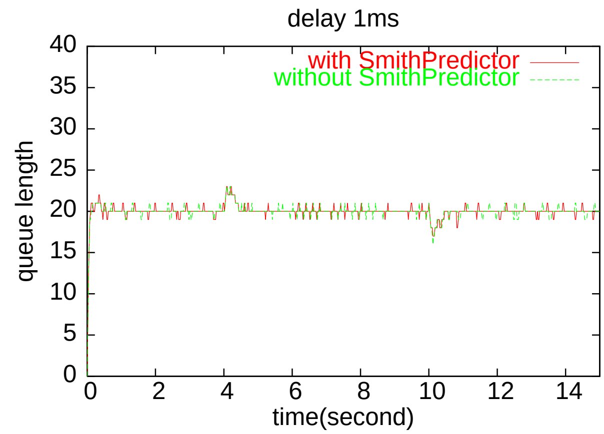

Fig. 4. Case LAN-S20: Queue dynamics for varying amount of available resources in LAN environment ($\tau = 1.0$ [ms], $S = 20$ [MI])

2) Scenario 2: When transfer delay between the job manager and a site varies with time

First, the results for when the amount of a site's available resources $\mu(t)$ varied with time (Scenario 1) are indicated. The parameter configuration used in Scenario 1 is shown in Table II. In Scenario 1, the amount of the site's available resources varied with time, and the simulation was run. Specifically, the amount of the site's available resources $\mu(t)$ is changed as follows (units are MIPS).

$$\mu(t) = \begin{cases} 1,000 & 0 \leq t < 4 \\ 200 & 4 \leq t < 10 \\ 1,500 & \text{otherwise} \end{cases}$$

The simulation results (rise time, overshoot, and settling time) for Scenario 1 are shown in Table III.

Variations in the queue length for the local resource manager in a LAN environment are shown in Figs. 3 (LAN-S2) and 4 (LAN-S20). Figure 3 (LANS2) is a result with fine granularity of tasks ($S = 2$ [MI]). Figure 4 (LAN-S20) is a result with rough granularity of tasks ($S = 20$ [MI]).

According to these figures, both PI control and DRM-DC displayed satisfactory performance in a LAN environment with slight transfer delay. In other words, the rise time for the queue length was sufficiently small and overshoot was also sufficiently small at less than 5%. In addition, settling time was also under 1 [s] (Table III). The amount of the site's available resources changed at $t = 4, 10$ [s], although control instantly followed these changes. When the task size was large at $S = 20$ [MI], performance indicators (rise time, overshoot, and settling time) deteriorated somewhat, although this is because of the increased task size and rough granularity of resource management control.

In a network with substantial transfer delay as in a WAN environment, however, performance deteriorated substantially with simple PI control. Variations in the queue length for the local resource manager in a WAN environment are shown in Figs. 5 (WAN-S2) and 6 (WAN-S20). Figure 5 (WAN-S2) is a result with fine granularity of tasks ($S = 2$ [MI]). Figure 6 (WAN-S20) is a result with rough granularity of tasks ($S = 20$ [MI]).

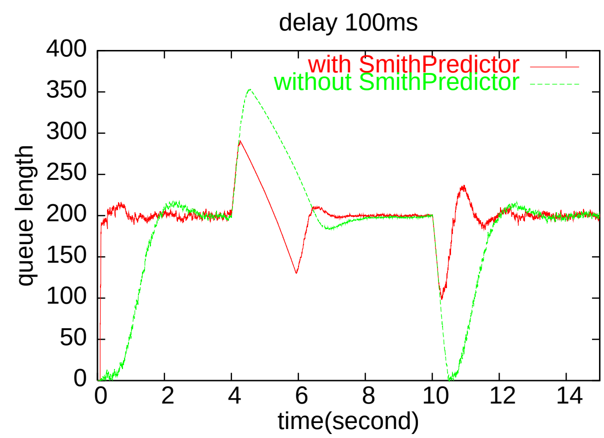

Fig. 5. Case WAN-S2: Queue dynamics for varying amount of available resources in WAN environment ($\tau = 100.0$ [ms], $S = 2$ [MI])

According to these figures, there is a substantial difference in transient-state characteristics of PI control and DRM-DC in a WAN environment with a substantial transfer delay. In particular, a remarkable difference appeared immediately after the amount of a site's available resources changed at $t = 4, 10$ [s]. Comparison of PI control and DRM-DC rise time and settling time indicates the exceptional superiority of DRM-DC transient-state characteristics (Table III). In addition, overshoot with DRM-DC was small compared to that with PI control. In addition, immediately after the amount of available resources increased (around $t = 11$ [s]), queue length became 0 with PI control and usage of resources decreased.

Based on these observations, DRM-DC using a delay compensator displays satisfactory steady-state characteristics and transient-state characteristics in a WAN environment with a substantial transfer delay in particular.

Next, results when transfer delay between the job manager and a site τ varied with time (Scenario 2) are indicated.

TABLE II

PARAMETER CONFIGURATION IN SCENARIO 1

		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 r_0		200	20	200	20

TABLE III

SIMULATION RESULTS IN SCENARIO 1

		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	0.4	5	0.8
LAN-S20	PI	1.0	20	0.4	5	0.8
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	0.3
WAN-S20	PI	100.0	20	1.62	85	1.6

TABLE IV

PARAMETER CONFIGURATION IN SCENARIO 2

		LAN-S20	WAN-S20
task size S	[MI]	20	20
transfer delay $\mu(t)$	[MI]	1000	1000
control interval T	[ms]	20.0	20.0
target queue length r_0		20	20

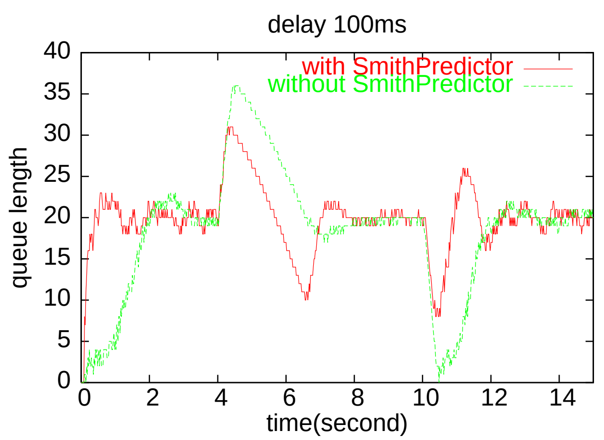

Fig. 6. Case WAN-S20: Queue dynamics for varying amount of available resources in WAN environment ($\tau = 100.0$ [ms], $S = 20$ [MI])

The parameter configuration used in Scenario 2 is shown in Table IV. In Scenario 2, the transfer delay between the job manager and a site varied with time, and the simulation was run. Specifically, transfer delay between the job manager and a site τ changed as follows.

$$\tau \leftarrow \begin{cases} \tau & 0 \leq t < 4 \\ 1.8\tau & 4 \leq t < 10 \\ 0.8\tau & \text{otherwise} \end{cases}$$

Simulation results (rise time, overshoot, and settling time) for Scenario 2 are shown in Table V.

Variations in the queue length for the local resource manager in a LAN environment are shown in Fig. 7 (LAN-S20). In addition, variations in the queue length for the local resource manager in a WAN environment are shown in Fig. 8 (WAN-S20). These simulations are both results with a rough granularity of tasks ($S = 20$ [MI]).

Focusing first on simulation results in a LAN environment with a slight transfer delay (Fig. 7 (LAN-S20)) indicates that both PI control and DRM-DC show satisfactory performance. Looking at results in Table V as well, PI control and DRM-DC have exactly the same performance with regard to the queue length's rise time, overshoot, and settling time.

In a WAN environment with a substantial transfer delay, however, DRM-DC show exceptional steady-state characteristics and transient-state characteristics. According to Fig. 8 (WAN-S20), DRM-DC quickly responds to variations in transfer delay compared to PI control. In addition, DRM-DC performance is superior to a large degree over PI control with regard to the rise time and settling time according to Table V. However, overshoot reached 100% for both PI control and DRM-DC. As is apparent from Fig. 8 (WAN-S20), queue length differs when transfer delay increases ($4 \leq t < 10$) and stability can not be achieved. The result is that overshoot is extremely large. To avoid this problem, the control parameters (K_P and K_I) can be configured to smaller values.

Based on the above results, DRM-DC using a delay compensator exhibit highly exceptional steady-state characteristics and transient-state characteristics in a WAN environment with a substantial transfer delay in particular.

TABLE V

SIMULATION RESULTS IN SCENARIO 2

		τ [ms]	S [MI]	rise time [s]	overshoot [%]	settling time [s]
LAN-S20	DRM-DC	1.0	20	0.06	5	0.08
LAN-S20	PI	1.0	20	0.06	5	0.08
WAN-S20	DRM-DC	100.0	20	0.1	100	0.24
WAN-S20	PI	100.0	20	0.52	100	0.7

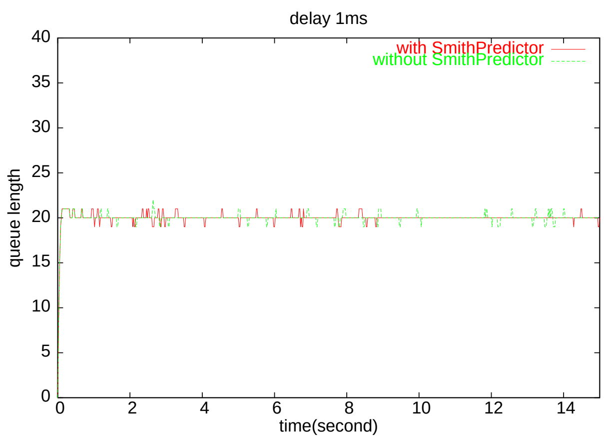

Fig. 7. Case LAN-S20: Queue dynamics for varying transfer delay in LAN environment ($\tau = 1.0$ [ms], $S = 20$ [MI])

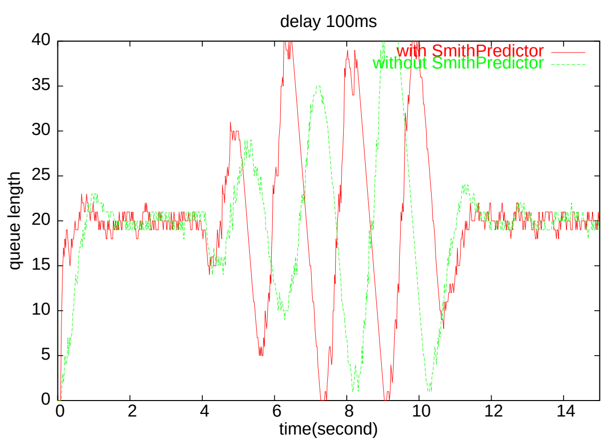

Fig. 8. Case WAN-S20: Queue dynamics for varying transfer delay in WAN environment ($\tau = 100.0$ [ms], $S = 20$ [MI])

VII. CONCLUSIONS AND FUTURE TOPICS

This paper proposed DRM-DC for wide-area Grid computing. Wide-area Grid computing has problems in that available resources vary with time and the network's transfer delay cannot be ignored. Thus, resource management to achieve high stability and transient-state characteristics through use of a delay compensator (a Smith predictor) in feedback control was proposed. Then, discrete time simulation was performed using a Simgrid simulator, and the effectiveness of the proposed DRM-DC was validated. The result was that, with respect to variations in site throughput, satisfactory results remained, and the effectiveness of the proposed format was clarified.

The DRM-DC proposed in this paper sought to improve steady-state characteristics and transient-state characteristics in particular as part of performance indicators for resource management explained in Section III. However, performance indicators of stability, robustness, and flexibility are also important in actual wide-area Grid computing. The DRM-DC used a delay compensator in feedback control to counteract the effects of transfer delay. However, the delay compensator's effectiveness is largely dependent on the accuracy of the plant model (sites) [5]. In reality, construction of an accurate numerical model of sites is difficult and modeling error must be allowed to some extent. Thus, there are plans to study techniques to improve the stability and robustness of the proposed DRM-DC in the future.

REFERENCES

[1] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*. San Francisco: Morgan Kauffman, 1999.

[2] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The physiology of the Grid: An open grid services architecture for distributed systems integration," Jan. 2002, available at http://www.globus.org/research/papers/ogsa.pdf.

[3] I. Foster, A. Roy, and V. Sander, "A quality of service architecture that combines resource reservation and application adaptation," in *Proceedings of the Eight International Workshop on Quality of Service (IWQOS 2000)*, June 2000, pp. 181–188.

[4] F. Berman, G. Fox, and T. Hey, Eds., *Grid Computing: Making the Global Infrastructure a Reality*, 2003, ch. Condor and the Grid, pp. 945–962.

[5] W. S. Levine, Ed., *The Control Handbook*. CRC Press, 1996.

[6] H. Casanova, "Simgrid: A toolkit for the simulation of application scheduling," in *Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001)*, May 2001.

[7] S. Mascolo, "Congestion control in high-speed communication networks using the Smith principle," *Automatica*, vol. 35, pp. 1921–1935, May 1999.

[8] Globus Project Team, "The Globus Alliance," http://www.globus.org/.

[9] H. Casanova, G. Obertelli, F. Berman, and R. Wolski, "The AppLeS parameter sweep template: User-level middleware for the Grid," in *Proceedings of Super Computing 2000*, Nov. 2000.