A Performance Evaluation of the Synchronized Provisioning with an Adaptive Buffer Resilience Scheme over Grid Networks

Byungsang Kim, *Student Member, IEEE*, and Chan-Hyun Youn, *Member, IEEE*

***Abstract*—For widely distributed data analysis applications over that run the Internet, both the instability of the data transfer time and the dynamics of data processing rate require a more sophisticated data provisioning scheme to maximize parallel efficiency, in particular, under conditions in which real-time and limited data buffer (storage) constraints are given. In this letter, we propose a synchronized data provisioning scheme that implicitly avoids the data buffer overflow as well as explicitly controls the data buffer underflow by optimally adjusting the buffer resilience. In order to guarantee the designated quality of service, we further exploit an adaptive buffer resilience control algorithm based on sample path analysis of the state of the data buffer and the demand queue. The simulation results show that the proposed scheme is suitably efficient to apply to an environment that can not postulate the stochastic characteristics of the data transfer time and data processing rate.**

***Index Terms*—Grid network, distributed data processing, parallel efficiency, data provisioning, buffer resilience.**

I. INTRODUCTION

On large-scale data analysis applications that run across widely distributed computing domains in the Grid, it is essential to provide application dataset continuously to all participating computing nodes according to their data processing rate so as to prevent idle processors [1]. In particular, on real-time data processing schemes which have emphasized the ability to replace data in a memory buffer for high availability rather than constantly fetching the data from slower storage, a more sophisticated data provisioning scheme is required, as the buffer capacity in such a case serves as a major constraint [2]. Furthermore, under conditions in which instability of the data transfer time and the dynamics of data processing rate are presented, keeping an optimal level of data holding level is a critical issue when seeking to reduce both memory usage and the number of idle processors.

In such a data provisioning model, most studies thus far have attempted only improve the end-to-end data transfer throughput without any consideration of the storage capacity or data processing rate [1][3]. Although the approaches to coordinate between data provisioning and processing rate have been developed in a dedicated resource model, most of them are based on centralized sender-driven data distribution models and, focus on global optimization under the deterministic assumptions that a sender has static knowledge about the data transfer time and the processing rate of all receiving nodes [2][4]. On the wide and large scale collective computing model however, the above approaches are not practical because shared resources show unpredictable or fluctuating state. Furthermore, the constraint of the limited storage undermines the robustness of the optimized solution. Moreover, the complexity of the global solution is in fact NP-hard [4].

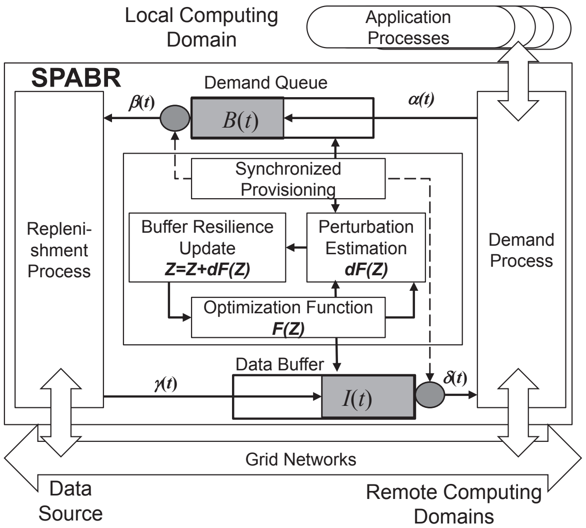


Fig. 1. Schematic diagram of the data provisioning service.

In this letter, we consider the data provisioning problem in terms of the receiver-driven decentralized model. We place a data provisioning service (DPS) on each receiver side (computing domain), which is responsible for storing a set of application data from the source and forwarding it into the local computing nodes as well as any remote computing domains. Within the DPS, we propose a more robust data provisioning method as a *synchronized provisioning with adaptive buffer resilience* (SPABR) scheme which serves implicitly to avoid the data buffer overflow while explicitly regulating the buffer underflow by optimally adjusting the buffer resilience. In order to guarantee the designated quality of service (the waiting time of the processors), we further exploit an adaptive buffer resilience control algorithm based on a sample path analysis without any knowledge of the stochastic characteristics of the data transfer time and the data processing rate.

II. DECENTRALIZED DATA PROVISIONING MODEL

Denoting a data object as a small part of the total dataset, which is a countable processing unit for application processes, as shown in Fig. 1, we model an arbitrary DPS as a discrete event system which has a finite data buffer (DB) and an infinite demand queue (DQ) with two stochastic processes: **(*i*) Data**

Manuscript received November 21, 2011. The associate editor coordinating the review of this letter and approving it for publication was I.-R. Chen.

B. Kim is with the Dept. of Information and Communications Engineering, KAIST, Daejeon, Rep. of Korea (e-mail: bs.kim@kaist.ac.kr).

C.-H. Youn is with the Dept. of Electrical Engineering, KAIST, Daejeon, Rep. of Korea (e-mail: chyoun@kaist.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2012.030512.112324

demand process - When an event arrives at the DQ, the DPS responses with a data object in the DB. If the DB is empty, the event should wait in the DQ until the DB has been filled. ***(ii)*** **Data replenishment process** - When the DPS sends a request event to the data source, a data object comes to the DB after the replenishment time has elapsed. With these two processes, the characteristics of the SPABR are described below in detail.

A. Synchronized Data Provisioning Scheme

In order to avoid the overflow of the DB, the SPABR synchronizes the two aforementioned processes, e.g., as soon as a demand process is served, the DPS triggers a replenishment process. Thus, after the replenishment time has elapsed, the number of data objects in the DB is restored in the previous state. Note that the DPS will never be operated if the DB is initially empty. Therefore, we introduce a buffer resilience (Z) which represents the number of reserved data objects in the DB. Before the DPS performs the synchronized data provisioning operation, the DPS determines Z and prepares the data objects up to Z using the replenishment process.

Let $I(t)$ and $B(t)$ be the number of reserved data objects in the DB and the number of idle processes in the DQ at t, respectively. Then, initially, $I(0) = Z$ and $B(0) = 0$. It is also possible to identify that $I(t) = Z + \gamma(t) - \delta(t)$, $B(t) = \alpha(t) - \beta(t)$, where $\gamma(t)$ and $\delta(t)$ are the arrival and the departure rate in the DB and $\alpha(t)$ and $\beta(t)$ are the arrival and the departure rate in the DQ over time $(0, t)$, respectively (see Fig. 1). Since these two departure processes are synchronized, $\beta(t)$ and $\delta(t)$ are always the same. This gives the following relationship:

$$I(t) - B(t) = Z + \gamma(t) - \alpha(t). \tag{1}$$

Since $I(t) > 0$ implies $B(t) = 0$, while, $B(t) > 0$ implies $I(t) = 0$ and considering both are non-negative integers, we obtain $I(t)$ and $B(t)$ from (1) as follows:

$$I(t) = \max[Z - N(t), 0], \tag{2}$$

$$B(t) = \max[N(t) - Z, 0], \tag{3}$$

where, we denote $N(t)$ as the *provisioning function* of the DPS, which is defined by the two aforementioned arrival rates:

$$N(t) = \alpha(t) - \gamma(t), \alpha(t) \geq \gamma(t), t \geq 0. \tag{4}$$

$N(t)$ is represented as a general queuing system of which the arrival and service patterns correspond to the data processing rate in the computing domains and the data transfer time from the data source. In the SPABR, the DB avoids the overflow problem if the buffer size is larger than Z since $I(t)$ is bounded to Z, as shown in (2). Furthermore, because $B(t)$ appears when $N(t) \geq Z$, it is possible to regulate $B(t)$ as well as $I(t)$ by adjusting Z under given $N(t)$.

B. Adaptive Buffer Resilience Control

Supposing that $E_I(z)$ and $E_B(z)$ are the average buffer usage and the average number of idle processes under $Z = z$, we formulate the cost minimized function to find the optimal buffer resilience as a weighted sum of two criteria:

$$F(z) = \upsilon E_B(z) + (1 - \upsilon) E_I(z), 0 \leq \upsilon \leq 1, \tag{5}$$

where υ denotes the relative weight of $E_B(z)$.

Assuming that the stochastic behavior of $N(t)$ is known and stationary, we can obtain the optimal buffer resilience that minimizes the cost function by $\partial F(z)/\partial z = 0$, as $F(z)$ is convex to z. However, it is difficult to be identify $N(t)$ practically. In addition, the steady state cannot easily be justified under the fluctuation of $N(t)$. For this reason, we develop an adaptive buffer resilience (ABR) control algorithm by estimating $\partial F(z, t)/\partial z$ based on a perturbation analysis of the observed sample path over a finite time area. This allows us to obtain n^{th} the optimal buffer resilience z_n^* through an iterative form as

$$z_{n+1}^* = z_n^* - \nu f_n(z_n^*), n = 0, 1, ... \tag{6}$$

where ν is a scale parameter for adjusting the gradient and $f_n(z^*)$ is the $\partial F_n(z^*)/\partial z$ for n^{th} iterations.

Through a perturbation analysis of both $I(t)$ and $B(t)$, we can obtain the derivative of $F_n(z^*)$. We apply the results in [5][6], in which the authors approximated the G/G/1 queuing model as a stochastic fluid model and derived the infinitesimal perturbation analysis (IPA) estimates of the occupancy of the buffer with respect to buffer size and its unbiasedness. Since $N(t)$ is independent of z, it turned out to be the sum of all intervals of $I(t) > 0$ periods such as

$$\frac{\partial E_I(z, t)}{\partial z} = \frac{1}{T} \sum_{j=0}^{J} \frac{\partial}{\partial z} \int_{\eta_j} (Z - N(t))\, dt = \frac{1}{T} \sum_{j=0}^{J} \eta_j. \tag{7}$$

where η_j is the j^{th} surplus period of the DB during $(0, T)$, $0 \leq j \leq J$, and $\eta_J \leq T$. Similarly, the derivative of $B(z, t)$ with respect to z over $(0, T)$ is

$$\frac{\partial E_B(z, t)}{\partial z} = \frac{1}{T} \sum_{k=0}^{K} \frac{\partial}{\partial z} \int_{\xi_k} (N(t) - Z) dt = -\frac{1}{T} \sum_{k=0}^{K} \xi_k. \tag{8}$$

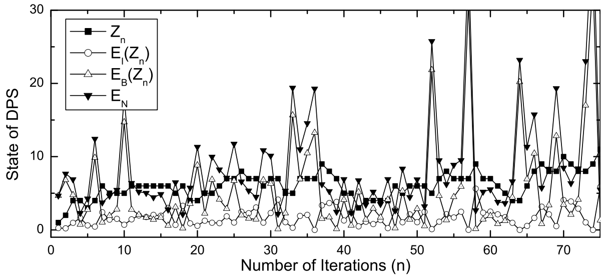
where ξ_k is the k^{th} surplus period of the DQ over $(0, T)$, $0 \leq k \leq K$, and $\xi_K \leq T$. Hence, the gradient for the cost function over [0,T] yields

$$f_n(z) = \frac{1}{T} \left(\upsilon \sum_{k=1}^{K} \xi_j - (1 - \upsilon) \sum_{j=1}^{J} \eta_j \right). \tag{9}$$

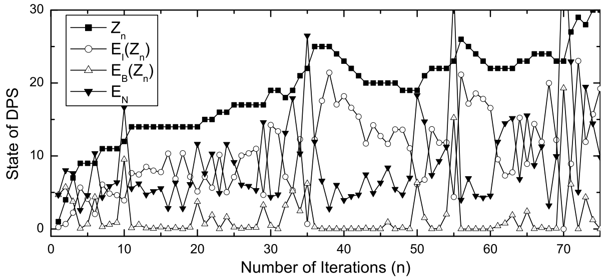
The result of (9) implies the simplicity of the ABR scheme, which only identifies the empty states of the DB and DQ in order to update the buffer resilience for each iteration.

III. PERFORMANCE EVALUATION

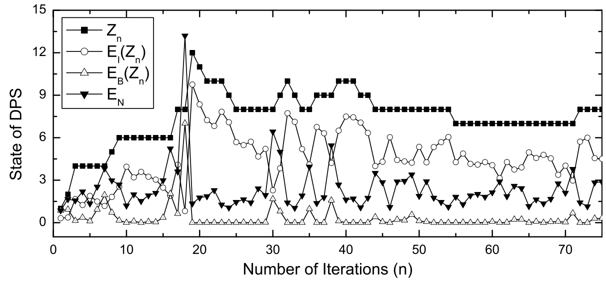
We evaluate the proposed ABR scheme using the SimJava modeler [8]. We set the initial parameters as Z_0=1, ν=5, and T= every 500 arrivals of the demands. Fig. 2 depicts the buffer resilience, the states of $I(t)$ and $B(t)$ with given $N(t)$ and υ. As shown in the plots, although the initial buffer resilience is fixed to one, each finds the optimal level of buffer resilience as the iterations are repeated. In particular, Fig. 2(a) and Fig. 2(b) compare different weighted factors on the same $N(t)$. The magnitude of the buffer resilience when $\upsilon = 0.9$ is higher than that when $\upsilon = 0.5$, resulting in different service quality levels. In similar, Fig. 2(a) and Fig. 2(b) compare different variations of $N(t)$ with the same weighted factor. Less variation is



(a) Provisioning time = $exp(9)$, processing time = $exp(10)$, $\upsilon = 0.5$



(b) Provisioning time = $exp(9)$, processing time= $exp(10)$, $\upsilon = 0.9$



(c) Provisioning time = $uni(9, 9)$, processing time = $uni(10, 10)$, $\upsilon = 0.9$

Fig. 2. Comparison of the size of buffer resilience and the occupancy of the DB and the DQ on each iteration with different $N(t)$ and υ.

presented, Less buffer resilience is required. The results show that the proposed ABR scheme provides a customized service quality based on the sample path analysis.

On the other hand, we compared the average costs, $F(Z)$ among the DPS with the ABR (proposed), the DPS without the ABR (non-ABR), and a case without DPS itself (non-DPS). For the non-ABR cases, the buffer resilience is fixed initially to one while that of the non-DPS case is zero since it has no space to store the data objects. As shown in Fig. 3, the proposed ABR scheme keeps the smallest values over all PPRs (average provisioning to processing time ratio) of $N(t)$. Meanwhile, the non-ABR cases are close to the minimum cost only at specific points according to their initial buffer resiliences such as Z=5 at (0.86, 0.88, 0.90), Z=10 at (0.92, 0.94), Z=15 at (0.94, 0.96), and Z=20 at 0.96, respectively. On the non-DPS case, it can not achieve the minimum costs at any of the PPRs. By minimizing the average cost, the proposed ABR scheme guarantee the optimal buffer resilience that satisfies synthetically both the data space and the waiting demands during run-time.

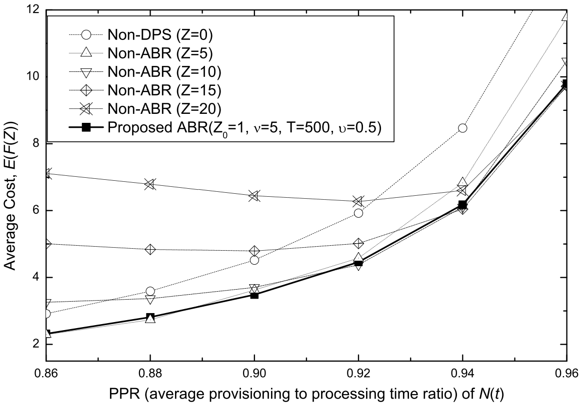


Fig. 3. Comparison of the $F(Z)$ among the proposed ABR case, the non-ABR cases, and the non-DPS case when the processing and the provisioning time are exponentially distributed.

IV. CONCLUSION

This letter presents a decentralized optimization model for an adaptive data provisioning scheme given a limited memory capacity and an unpredictable environment. The proposed SPABR makes it possible to avoid the data buffer overflow problem and to regulate the buffer underflow by optimally adjusting the resilience of the buffer, which satisfies both the buffer space and the service quality. The results show that the proposed scheme attempts to determine the optimal buffer resilience even when specific probability law of the provisioning function is not postulated. Future research will investigate the bursty types of provisioning time and processing rate. In particular, we are interested in nonlinear estimation techniques that are able to handle bursty traffic flows.

ACKNOWLEDGMENT

This research was equally supported by research and development programs through MEST/NRF (2011-0020522), MKE/KEIT(10039260), and MKE/KEIT(10040231).

REFERENCES

[1] J. Plank, A. Bassi, M. Beck, T. Moore, D. M. Swany, and R. Wolski, “Managing data storage in the network,” *IEEE Internet Computing*, vol. 5, no. 5, pp. 50–58, Sep. 2001.

[2] L Xiaolin, V. Bharadwaj, and C. C. Ko, “Divisible load scheduling on single-level tree networks with buffer constraints,” *IEEE Trans. Aerospace and Electron. Syst.*, vol. 36, no. 4, pp. 1298–1308, Oct. 2000.

[3] Y. Dengpan, Y. Esma, K. Sivakumar, R. Brandon, and K. Tevfik, “A data throughput prediction and optimization service for widely distributed many task computing,” *IEEE Trans. Parallel and Distrib. Syst.*, vol. 22, no. 6, June 2011.

[4] Y. Yang, H. Casanova, M. Drozdowski, M. Lawenda, and A. Legrand, “On the complexity of multi-round divisible load scheduling, Research Report 6096, INRIA, 2007.

[5] G. C. Christos, Y. Wardi, B. Melamed, G. Sun, and G. P. Christos, “Perturbation analysis for online control and optimization of stochastic fluid models,” *IEEE Trans. Automatic Control*, vol. 47, no. 8, pp. 1234–1248, 2002.

[6] Y. Zhao and B. Melamed, “IPA derivatives for make-to-stock production-inventory systems with backorders,” *Method. Comput. Appl. Probab.*, vol. 8, p. 191, 2006.

[7] A. John and J. G. Buzacott, *Stochastic Models of Manufacturing Systems*. Prentice-Hall, 1994.

[8] F. Howell and R. McNab, “SimJava: a discrete event simulation package for Java with applications in computer systems modelling,” ICWBMS, 1998.