Optimal Periodic Remapping of Bulk Synchronous Computations on Multiprogrammed Distributed Systems*

Ngo-Tai Fong, Cheng-Zhong Xu, and Le Yi Wang
Department of Electrical and Computer Engineering
Wayne State University, Detroit, MI 48202
Email: {ntfong, czxu, lywang}@ece.eng.wayne.edu

Abstract

For bulk synchronous computations that have non-deterministic behaviors, dynamic remapping is an effective approach to ensure parallel efficiency. There are two basic issues in remapping: when and how to remap. This paper presents a formal treatment of the first issue for dynamic computations with a priori known statistical behaviors. We have formulated the problem as two complement sequential stochastic optimization, with an objective of finding optimal remapping frequencies for a given tolerance of load imbalance on multiprogrammed distributed systems. We have developed analytical approaches to precisely characterize the transient statistical behaviors of the workload process and derived optimal remapping frequencies for various random workload change processes.

1 Introduction

A bulk synchronous computation proceeds in phases. During each phase, its processes perform calculations independently and then communicate new results with their data-dependent peers. Due to the need of synchronization between phases, the duration of a phase is determined by the slowest process. For bulk synchronous computations whose processes run in non-deterministic phase durations, it is highly desirable to re-distribute the workload of processing nodes at run-time [12].

Since dynamic remapping incurs non-negligible run-time overhead, a critical issue is when to remap so that the benefit from remapping will not be outweighed by its overhead. An important policy is periodic remapping. The "remap every k steps" policy has been applied to many parallel applications, in particular to those exhibiting gradual workload changes, due to its simplicity [8, 9, 10, 13]. The remapping frequency was often derived through experiments against the remapping periodicity in practice. The literature lacks formal analyses of the effect of the remapping frequency. Nicol and Saltz [10] modeled the issue of when to remap as a stochastic dynamic programming problem, assuming the workload of processors changes in independent and identical Markov death-birth processes. Due to the complexity of dynamic programming, their approach is limited to systems with very small number of workload states. In [5], we formulated the problem as a sequential stochastic optimization model from the perspective of individual applications. This paper extended the model to multiprogrammed distributed systems and formulated the problem as a complement model from the perspective of systems. It is known that general stochastic optimization approaches tend to reveal *asymptotic* or *stationary* properties of a random process. They were applied to predict the average execution time of the computations (without remapping) in the literature [1, 6, 7, 11]. However, the general optimization approaches are not readily applicable to the analysis of the effect of remapping because a remapping operation would be invoked anytime over the course of the computation. Based on order statistics and other stochastic optimization techniques, we developed optimization approaches to precisely characterize the *transient* statistic behaviors of the computation. We derived the optimal remapping frequencies for applications with various statistic behaviors on both homogeneous and heterogeneous systems.

Note that there were studies on scheduling for high performance computing on multiprogrammed distributed systems. Previous studies were primarily focused on extending coscheduling policies of shared memory machines onto multiprogrammed clusters. These coscheduling work on clusters basically answer the question of when to start the processes of a parallel job. Dynamic remapping complements these work by addressing the issue of load balancing among processors.

The remainder of the paper is organized as follows. Section 2 describes the computational models and formulates the problem. Sections 3, 4, and 5 deal with the optimization problem for homogeneous and heterogeneous platforms. Section 6 concludes this paper.

*This research is supported in part by NSF grants EIA-9729828 and ECS-9634375.

2 The Model

Consider dynamic computations in a parallel computer with N processing nodes. The processing nodes can be either homogeneous or heterogeneous in terms of their computational capacities. In the following, we first assume homogeneous systems. We will extend the analysis to heterogeneous environments in Section 5. Let t be a time variable, representing phase index of an adaptive bulk synchronous computation. We quantify the workload of processor i at time t by $w_i(t)$ in terms of the number of residing processes, $i = 1, 2, \cdots, N$. Let $z_i(t)$ denote the amount of workload generated or finished from $t-1$ to t. Let the vectors $\mathbf{w}(t) = (w_1(t), w_2(t), \cdots, w_N(t))$ and $\mathbf{z}(t) = (z_1(t), z_2(t), \cdots, z_N(t))$ denote the global workload distribution at certain time t and the workload change distribution from time $t-1$ to t, respectively. Then, the workloads at time t, without remapping, satisfy the following dynamic systems:

$$\mathbf{w}(t) = \mathbf{w}(t-1) + \mathbf{z}(t), \tag{1}$$

Assume processors initially have the equal workload at time $t = 0$ and the amount of workload change, $z_i(t)$, $i = 1, 2, \ldots, N$, are independent random variables with mean μ_i and variance σ_i^2. Note that modeling the workload (or workload change) of a processor by a random variable is commonplace in the performance evaluation literature [11, 10]. Most of their models assumed the random variables were independent and identically distributed (i.i.d.) with distributions like normal distributions and exponential distributions. By contrast, the model of workload change in Eq.(1) is distribution-free and hence features a characterization of general dynamic applications.

By the dynamic system in Eq.(1), it is expected that the processors' workload distribution will change with time and finally lead to a severely imbalance state. Since the duration of a phase is determined by the heavily loaded processors due to the need of barrier synchronization between phases, the overall system performance may deteriorate in time. The objective of remapping is to minimize the workload difference between processors. Since a remapping operation incurs significant run-time overhead, the adaptive computation cannot afford frequent remapping. It must tolerate certain degree of load imbalance so as to amortize the remapping cost. Our primary concern is to minimize the remapping frequency for a given tolerate.

Let $\bar{\mathbf{w}}(t) = (\bar{w}(t), \bar{w}(t), \cdots, \bar{w}(t))$, where $\bar{w}(t) = \sum_{i=1}^{N} w_i(t)/N$, denote the uniform workload distribution at time t. We define *normalized extreme workload difference* at time t as

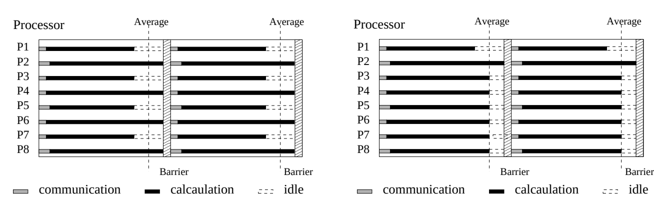
$$d(t) = \frac{E\left[\max_{i=1,2,\cdots,N} |w_i(t) - \bar{w}(t)|\right]}{E[\bar{w}(t)]}. \tag{2}$$

Throughout this paper, $E[\cdot]$ denotes the expected value of a random variable.

The term reflects the extra execution time of the most heavily loaded processor and the waiting time of the most lightly loaded processor, normalized with respect to the average load level. The normalized metric ensures that processors' workload changes at a comparable rate between phases. The first objective of this study is to find the maximum interval T for a given bound D of the $d(t)$. Since a remapping operation drive any load distribution to a uniform distribution, we consider a single period starting from a uniform distribution (t=0) to the time (t=T) when remapping becomes necessary. Computation periods separated by remapping operations may start with different workload mean $\bar{w}(0)$. The objective is then reduced to maximize T while keeping the workload difference bounded. Precisely, we represent the objective as the following stochastic optimization problem:

$$\mathcal{P}_I : \begin{cases} \text{maximize} & T \\ \text{subject to} & \begin{cases} \mathbf{w}(t) = \mathbf{w}(t-1) + \mathbf{z}(t), \\ d(t) \le D, \\ \mathbf{w}(t) \ge 0, \\ \text{for all } t = 1, 2, \cdots, T. \end{cases} \end{cases}$$

Notice that the objective of $\mathcal{P}_I$ is to optimize the performance from the perspective of individual applications. Its optimal solution may not necessarily lead to high efficient utilization of the available system resource on a multiprogrammed distributed system. Figure 1 shows two scenarios, where $w_{2i-1} = \bar{w} - D$ and $w_{2i} = \bar{w} + D$ for $1 \le i \le N/2$ in Figure 1(a) and $w_1 = \bar{w} - D$, $w_2 = \bar{w} + D$, and $w_i = \bar{w}$ for $3 \le i \le N$ in Figure 1(b). Clearly, both are optimal solutions in terms of the objective of $\mathcal{P}_I$. However, from the perspective of systems, Figure 1(b) is evidently more desirable because it allows more processors to be co-scheduled at a time for other parallel jobs.



(a) $w_{2i-1} = \bar{w} - D$ and $w_{2i} = \bar{w} + D$

(b) $w_1 = \bar{w} - D$, $w_2 = \bar{w} + D$, and $w_i = \bar{w}$

Figure 1: Two different workload distributions under objective of $\mathcal{P}_I$

To reflect the desirable properties of Figure 1(b), we define a term *normalized workload deviation* $v(t)$ at time t

to measure the normalized deviation of $\mathbf{w}(t)$ from $\bar{\mathbf{w}}(t)$, as follows:

$$v(t) = \frac{\sqrt{E\left[\|\mathbf{w}(t) - \bar{\mathbf{w}}(t)\|^2\right]}}{E[\bar{w}(t)]} = \frac{\sqrt{E\left[\sum_{i=1}^{N}(w_i(t) - \bar{w}(t))^2\right]}}{E[\bar{w}(t).]} \quad (3)$$

Correspondingly, the second objective of this study is to maximize time T for a given bound B of the normalized workload deviation $v(t)$. Specifically, we represent the objective as the following stochastic optimization problem:

$$\mathcal{P}_{II}: \begin{cases} \text{maximize} & T \\ \text{subject to} & \begin{cases} \mathbf{w}(t) = \mathbf{w}(t-1) + \mathbf{z}(t), \\ v(t) \le B, \\ \mathbf{w}(t) \ge 0, \\ \text{for all } t = 1, 2, \cdots, T. \end{cases} \end{cases}$$

By the objective of problem $\mathcal{P}_{II}$, the distribution of Figure 1(a) is more desirable. On the other hand, the objective of $\mathcal{P}_{II}$ should also be complemented by that of $\mathcal{P}_I$. For example, consider an extreme scenario where $w_1 = \bar{w} + (N-1)\delta$ and $w_i = \bar{w} - \delta$, for $2 \le i \le 8$. It can be seen that the scenario may exhibit a small workload deviation and a large extreme workload difference for a small δ and a large N.

In fact, we have the following relationships between the two objectives. The lemma reveals that remapping with respect to $v(t)$ is a conservative strategy from the viewpoint of $d(t)$.

Lemma 2.1

$$d^2(t) + \theta^2 \le v^2(t) \le N(d^2(t) + \theta^2), \quad (4)$$

where $\theta^2 = \frac{Var(\max_{i=1,2,\cdots,N}|w_i(t) - \bar{w}(t)|)}{(E[\bar{w}(t)])^2}$.

In the subsequent sections, we address the optimization problem for computations that exhibit different statistical behaviors.

3 Near-Optimal Remapping Frequency for Problem $\mathcal{P}_I$

First, we consider of parallel computations that exhibit distribution-free and i.i.d. random variables $z_i(\cdot)$ in the model of Eq. (1). We will derive a lower bound of the optimal remapping interval for a given bound of the normalized extreme workload difference. The bound is tight and nearly optimal in the case that the remapping interval becomes large enough.

For tractability, we approximate the optimization problem by decomposing the extreme workload difference $d(t)$ into a weak combination of $d_1(t)$ and $d_2(t)$:

$$d_1(t) = \frac{E\left[\max_{i=1,2,\cdots,N} w_i(t) - \bar{w}(t)\right]}{E[\bar{w}(t)]}, \quad (5)$$

$$d_2(t) = \frac{E\left[\bar{w}(t) - \min_{i=1,2,\cdots,N} w_i(t)\right]}{E[\bar{w}(t)]}. \quad (6)$$

Assume processors initially have the equal workload and their initial workloads $w_i(0) = w$. Assume $z_i(t)$ are i.i.d. random variables $z_i(t)$ of the same mean μ and variance σ^2. By Eq. (1), the workloads of N processors at any time t, $w_i(t)$ are i.i.d. random variables with the same mean $w + t\mu$ and variance $t\sigma^2$. Then, using a well-known result from order statistics [3], we have

$$E\Big[\max_{i=1,2,\cdots,N} w_i(t)\Big] \le w + t\mu + \frac{N-1}{\sqrt{2N-1}}\sqrt{t}\sigma. \quad (7)$$

It follows that

$$d_i(t) \le \frac{(N-1)\sqrt{t}\sigma}{\sqrt{2N-1}(w + t\mu)}, \quad i = 1, 2.$$

Let

$$\hat{d}(t) \equiv \frac{(N-1)\sqrt{t}\sigma}{\sqrt{2N-1}(w + t\mu)}. \quad (8)$$

For a given load imbalance bound D, we set $\hat{d}(t) \le D$ conservatively, instead of ensuring $d(t) \le D$. We are to find the maximum T^* such that Eq. (8) holds for $t = 1, 2, \cdots, T^*$. The T^* is a lower bound of the optimal interval for the problem $\mathcal{P}_{\mathcal{I}}$.

Figure 2 plots $\hat{d}(t)$ without remapping in the case $\mu \ne 0$. It reaches its maximum value $D^* = \frac{(N-1)\sigma}{2\sqrt{(2N-1)w\mu}}$ at the time of w/μ. Let $T_0 = w/\mu$. The figure shows that processors tend to arrive an equilibrium state statistically in the long run, while their workload difference is increasing at the beginning until the time of T_0. Suppose processors' initial load level is 100 and the mean of workload change each time step is 2. The most severe load imbalance occurs in 50 steps in statistics. Setting an appropriate bound D ensures processors workload difference won't exceed a certain level.

In [5], we proved that for a given load imbalance D and $D = \frac{D^*}{m}$ where $m = 1, 2, \ldots$, the lower bound of the optimal remapping interval $T^* = (2m^2 - 2m\sqrt{m^2-1} - 1)T_0$.

4 Optimal Remapping Frequency for Problem $\mathcal{P}_{II}$

This section addresses the issue of when to remap, subject to the second optimization constraints, for applications

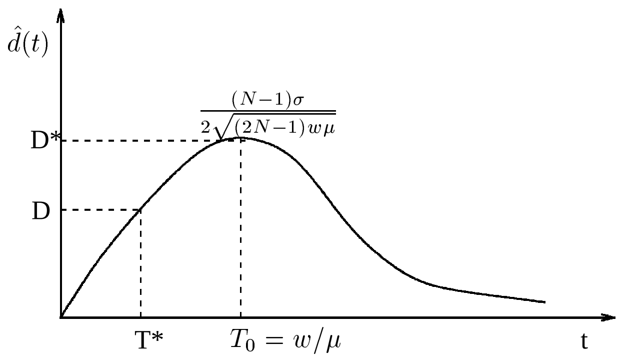


Figure 2: Illustration of workload difference: $\mu \neq 0$

where the workload change of processors $z_i(\cdot)$ are independent with mean μ_i and variance σ_i^2. We will precisely characterize the transient behaviors of random processes and derive optimal remapping frequencies for the most general computations that are of distribution-free workload changes.

4.1 Distribution-Free Workload Changes

Before deriving the optimal remapping interval for a given bound B, we first present an asymptotic value V_L of the workload deviation $v(t)$ as t gets large. The following lemma shows that it is the stationary workload deviation when the computations proceeds without remapping.

Lemma 4.1 *The normalized workload deviation function $v(t)$ of Eq. (3) is convergent to a asymptotic normalized workload deviation V_L as $t \to \infty$ and*

$$V_L \equiv \lim_{t\to\infty} v(t) = \frac{\sqrt{\sum_{i=1}^{N} \mu_i^2 - N\bar{\mu}^2}}{\bar{\mu}},$$

if $\bar{\mu} \neq 0$, where $\bar{\mu} = \sum_{i=1}^{N} \mu_i / N$.

For a given bound B, we are to find the maximum T^* such that $v(t) \leq B$. From Eq. (3), we obtain that

$$v(t) = \frac{\sqrt{(N-1)\bar{\sigma}^2 t + \left[\sum_{i=1}^{N} \mu_i^2 - N\bar{\mu}^2\right] t^2}}{w + t\bar{\mu}},$$

where $\bar{\sigma}^2 = \sum_{i=1}^{N} \sigma_i^2 / N$.

Let B^* be the maximum value of $v(t)$, as shown in Figure 3. It can ben shown that $v(t) = V_L$ at

$$T_1 = \frac{w^2 V_L^2}{(N-1)\bar{\sigma}^2 - 2w\bar{\mu}V_L^2},$$

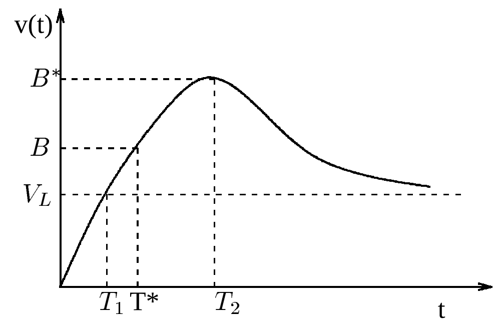


Figure 3: Illustration of normalized workload deviation

and that $v(t) = B^* = \frac{(N-1)\bar{\sigma}^2}{2\sqrt{w\bar{\mu}[(N-1)\bar{\sigma}^2 - w\bar{\mu}V_L^2]}}$ at

$$T_2 = \frac{w(N-1)\bar{\sigma}^2}{\bar{\mu}[(N-1)\bar{\sigma}^2 - 2w\bar{\mu}V_L^2]} = 2T_1 + \frac{w}{\bar{\mu}}.$$

For a given bound B between 0 and B^*, we summarize the results in the following theorem. The detailed proof can be seen in [4].

Theorem 4.1 *Assume the workload change of processors $z_i(\cdot)$, $i = 1, 2, \cdots, N$ are independent random variables with mean μ_i and variance σ_i^2, respectively. For a given small bound B of the normalized workload deviation, the optimal remapping interval for the problem $\mathcal{P}_{II}$ is*

1. *For a given B, $V_L < B < B^*$, there exits $m_1 > 1$ such that $B^2 = V_L^2 + \frac{1}{m_1}(B^{*2} - V_L^2)$. Then,*

$$T^* = (2k_1 - 1)T_2 - 2(k_1 - 1)T_1, \qquad (9)$$

where $k_1 = m_1 - \sqrt{m_1(m_1 - 1)}$.

2. *For a given B, $0 < B < V_L$, there exits $m_2 > 0$ such that $B^2 = V_L^2 - \frac{1}{m_2}(B^{*2} - V_L^2)$. Then,*

$$T^* = (2k_2 - 1)T_2 - 2(k_2 - 1)T_1, \qquad (10)$$

where $k_2 = \sqrt{m_2(m_2 + 1)} - m_2$.

Since the workload change $z_i(t), i = 1, 2, \cdots, N$, are assumed to be of distribution-free with different means and variances, T^* of Theorem 4.1 holds for most general classes of computations. In the case that $z_i(t)$ share the same mean μ and variance σ^2, the normalized workload deviation function is reduced to

$$v_0(t) = \frac{\sqrt{(N-1)\sigma^2 t}}{w + t\mu}, \qquad (11)$$

and $V_L = \lim_{t\to\infty} v_0(t) = 0$.

The maximum value of $v_0(t)$, B^*, is equal to $\sqrt{\frac{(N-1)\sigma^2}{4w\mu}}$ when $T_0 = w/\mu$. In comparison with $\hat{d}(t)$ in Eq.(8), it is interesting to see that

$$v_0(t) = \sqrt{\frac{2N-1}{N-1}}\hat{d}(t). \tag{12}$$

Recall that $\hat{d}(t)$ is a conservative index of extreme workload difference $d(t)$ due to the use of order statistics in analysis. Surprisingly, Eq.(7) increases the conservatility to the level of $v_0(t)$. We present the optimal remapping frequency with respect to $v_0(t)$ as follows.

Corollary 4.1 *Assume the workload change $z_i(\cdot)$, $i = 1, 2, \cdots, N$ are independent random variables with the same mean μ and variance σ^2. For a given small bound B of the normalized workload deviation, the optimal remapping interval for the problem $\mathcal{P}_{II}$ is*

1. *if $\mu = 0$,*

$$T^* = \left\lfloor \frac{w^2 B^2}{(N-1)\sigma^2} \right\rfloor; \tag{13}$$

2. *if $\mu \neq 0$, for a given $B = B^*/m$,*

$$T^* = (2m^2 - 2m\sqrt{m^2 - 1} - 1)T_0, \tag{14}$$

where $T_0 = w/\mu$ and $m = 1, 2, \ldots$.

4.2 Simulation Results

To illustrate the accuracy of the above estimate, we conducted an experiment to simulate the random processes with different distributions over 64 processors. Processors were assumed to be initially balanced with 100 workload units, unless otherwise specified. Each simulation data was an average of 400 replications.

The experiment assumed that processors change their workload units following a distribution function

$$z_i(t) = \begin{cases} 1, & \text{w. p.} \quad 0.25, \\ 0, & \text{w. p.} \quad 0.5, \\ -1, & \text{w. p.} \quad 0.25, \end{cases} \tag{15}$$

where w. p. means “with probability”. It is a typical death-birth Markov chain model. Similar distribution functions were also considered by other researchers [10].

Table 1 shows the simulation results from different bounds B of the normalized workload deviation, together with the corresponding theoretical results, T^*_{free} of Corollary 4.1. From the table, it can be seen that T^*_{free} perfectly matches the simulation results. The optimality of the estimation can also been seen from Figure 4 with different initial workload levels.

Table 1: Optimal remapping intervals for $\mathcal{P}_{II}$ with same mean and variance distributed workload changes

B	.10	.15	.20	.25	.30	.35	.40	.45	.50
T^*_{free}	3	7	12	19	28	38	50	64	79
T^*_{simu}	3	7	13	20	29	39	51	65	79

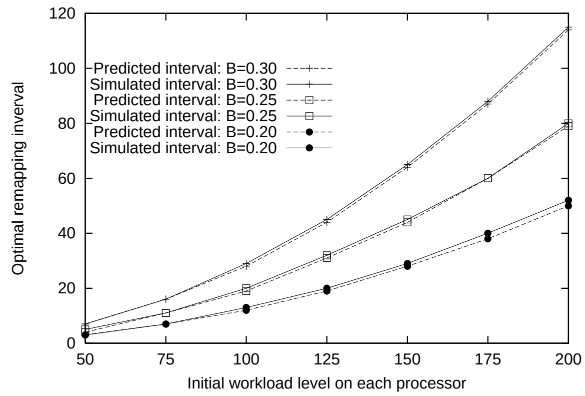


Figure 4: Optimal remapping interval with different initial workload levels

5 Heterogeneous Distributed Systems

Note that the above analysis assumed processors were homogeneous in their computational capacities and behaviors of the bulk synchronous computations were known in advance. In this section, we will extend the analytical framework to heterogeneous distributed systems by taking into account the processors' different computational capacities in remapping.

Let a constant vector $\mathbf{c} = (c_1, c_2, \cdots, c_N)$ denote the processors' computational capacities. We normalize workload distributions with respect to the capacity vector by setting

$$\mathbf{w}(t) = (\frac{w_1(t)}{c_1}, \frac{w_2(t)}{c_2}, \cdots, \frac{w_N(t)}{c_N})$$

and re-defining the uniform workload distribution at time t as

$$\bar{w}(t) = \frac{\sum_{i=1}^{N} w_i(t)/c_i}{N}.$$

Correspondingly, we define the normalized extreme workload difference at time t as

$$d(t) = \frac{E\left[\max_{i=1,2,\cdots,N} |w_i(t)/c_i - \bar{w}(t)|\right]}{E[\bar{w}(t)]}.$$

From the analysis in preceding sections, it can be easily seen that the major results in theorems hold for normalized workload levels if processors' workload change after normalization, $z_i(t)/c_i$ are identical. However, it may not be the case in general heterogeneous systems. As revealed in [2] through profiling NAS benchmark programs, running the same code with identical inputs on different machines may lead to execution time with different distributions. In the following, we will solve the generalized optimization problem by relaxing the distribution-specific assumptions.

Denote $\hat{w}_i(t)$ to be the accumulation of the normalized workload change at processor i from time 1 to time t. $\hat{w}_i(t) = \sum_{j=1}^{t} z_i(j)/c_i$. Let $Y = \max_{i=1,2,\cdots,N} \hat{w}_i(t)$. It is a random variable with the probability distribution function

$$F_Y^t(y) = \prod_{i=1}^{N} F_i^t(y), \tag{16}$$

where $F_i^t(\cdot)$ is the distribution function of $\hat{w}_i(t)$. The probability density function of the maximum, found by differentiating, yields

$$f_Y^t(y) = \sum_{j=1}^{N} f_i^t(y) \prod_{i=1, i\neq j}^{N} F_i^t(y), \tag{17}$$

where $f_i^t(\cdot)$ is the density function of $\hat{w}_i(t)$. Consequently, we have

$$E\Big[\max_{i=1,2,\cdots,N} \hat{w}_i(t)\Big] = \int_0^\infty y f_Y^t(y) dy \tag{18}$$

Let $h(t)$ denote the right hand side of Eq. (18). We obtain the following results.

Theorem 5.1 *Assume the normalized workload change $z_i(\cdot)$, $i = 1, 2, \cdots, N$ are independent random variables with different distributions. For a given bound D of the normalized extreme workload difference, the optimal remapping interval for the problem $\mathcal{P}_I$ is the solution T^* of the following inequality:*

$$\Big|\frac{h(t) - t\bar{\mu}}{w + t\bar{\mu}}\Big| \le D, \quad \textit{for } t = 1, 2, \cdots, T^*. \tag{19}$$

6 Summary

In summary, we have presented a formal treatment of the issue of when to invoke remapping operations during the execution of adaptive bulk synchronous computations. The objective of this study is to derive optimal remapping frequencies for a given tolerance of load imbalance. We have formulated the optimization problem as optimizing the remapping frequency while keeping the degree of load imbalance bounded by a constant. The degree of load imbalance is defined as normalized extreme workload difference between processors from the perspective of parallel jobs and normalized workload deviation from uniform distributions from the perspective of systems. Using order statistics theories and other stochastic optimization techniques, we have derived the optimal remapping frequencies for adaptive computations that exhibit various statistical behaviors. The analytical results have been shown accurate via simulations.

References

[1] V. D. Agrawal and S. T. Chakradhar. Performance analysis of synchronized iterative algorithms on multiprocessor systems. *IEEE Transactions on Parallel and Distributed Systems*, 3(6):739–746, November 1992.

[2] R. Armstrong, D. Hensgen, and T. Kidd. The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions. In *Proc. of the 7th Heterogeneous Computing Workshop*, 1998.

[3] H. A. David. *Order Statistics*. John Wiley, 2 edition, 1981.

[4] N.-T. Fong, C. Xu, and L. Wang. Optimal remapping of bullk synchronous computations on multiprogrammed distributed systems. Technical report, Dept. of ECE, Wayne State University, October 1999.

[5] N.-T. Fong, C. Xu, and L. Wang. Periodic remapping of non-deterministic bullk synchronous computations. In *Proc. of the 11th IASTED Int'l Conf. on Parallel and Distributed Computing and Systems*, pages 788–793, 1999.

[6] K. Li. Stochastic bounds for parallel program execution times with processor constrains. *IEEE Transactions on Computers*, 46:630–636, 1997.

[7] S. Madala and J. B. Sinclair. Performance of synchronous parallel algorithms with regular structures. *IEEE Transactions on Parallel and Distributed Systems*, 2(1):105–116, January 1991.

[8] B. Moon and J. Saltz. Adaptive runtime support for direct simulation monte carlo methods on distributed memory architectures. In *Scalable High-Performance Computing Conference*, pages 176–183, May 1994.

[9] D. Nicol and G. Ciardo. Automated parallelization of discrete state-space generation. *Journal of Parallel and Distributed Computing*, 47(2):153–167, December 1997.

[10] D. M. Nicol and J. H. Saltz. Dynamic remapping of parallel computations with varying resource demands. *IEEE Transactions on Computers*, 37(9):1073–1087, September 1988.

[11] G. D. Peterson and R. D. Chamberlain. Beyond execution time: expanding the use of performance models. *IEEE Parallel and Distributed Technology*, pages 37–49, Summer 1994.

[12] C. Xu and F. Lau. *Load Balancing in Parallel Computers: Theory and Practice*. Kluwer Academic Publishers, 1997.

[13] C. Xu, F. Lau, and R. Diekmann. Decentralized remapping of data parallel applications in distributed memory multiprocessors. *Concurrency: Practice and Experience*, pages 1351–1376, December 1997.