ACADEMIC PRESS

Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

J. Parallel Distrib. Comput. 63 (2003) 1036–1049

Journal of Parallel and Distributed Computing

http://www.elsevier.com/locate/jpdc

Optimal periodic remapping of dynamic bulk synchronous computations

Ngo-Tai Fong, Cheng-Zhong Xu,* and Le Yi Wang

Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA

Received 18 May 2000; revised 5 October 2002

Abstract

Dynamic remapping is critical to the performance of bulk synchronous computations that have non-deterministic behaviors because of the need of barrier synchronization between phases. There are two basic issues in remapping: when and how to remap. This paper presents a formal analysis of the issue of when to remap for dynamic computations with a priori known statistical behaviors, with an objective of finding optimal remapping frequencies for a given tolerance of load imbalance. The problem is formulated as two complement sequential stochastic optimization. Since general optimization techniques tend to reveal stationary properties of the workload process, they are not readily applicable to the analysis of the effect of periodic remapping. Instead, this paper develops new analytical approaches to precisely characterize the transient statistical behaviors of the workload process on both homogeneous and heterogeneous machines. Optimal remapping frequencies are derived for various random workload change processes with known or unknown probabilistic distributions. They are shown accurate via simulations.

Keywords: Bulk synchronous computations; Dynamic remapping; Load balancing; Stochastic optimization; Task assignment

1. Introduction

A fundamental issue in parallel computing is assignment, or mapping, of the workload to processing nodes of a parallel computer. The primary performance goals of mapping are to balance the workload among the processors, to minimize the interprocessor communication, and to reduce the run-time overhead of managing the assignment [9,28]. This can be accomplished statically or dynamically, depending upon the nature and predictability of the computation. Static assignment works for applications that have static and predictable computation and communication characteristics [9]. For applications that have unpredictable characteristics, dynamic scheduling must be used at run-time so as to adapt to the change of the workload. A special form of dynamic scheduling is *dynamic remapping*, the subject of this paper, which re-distributes the workload among processing nodes during the execution time.

In general, achieving the performance goals of mapping is intractable. However, a bulk synchronous computation lends itself to a fairly structured approach to mapping. Its processes proceed in phases that are separated by global synchronization points. During each phase, they perform calculations independently and then communicate new results with their data-dependent peers. Fig. 1 shows a typical scenario of the computation model with four processes running on different processors. The horizontal scale corresponds to computation phases; the vertical lines represent barrier synchronization operations which are due to begin. Because of the need of synchronization between phases, the duration of a phase is determined by the most heavily loaded processors, which in turn determines the total execution time of the computation. Consequently, mapping of a bulk synchronous computation can be reduced to load balancing at each phase.

Bulk synchronous computations may exhibit varying phase-wise computational requirements as the computation proceeds. This can occur in applications where the behavior of the physical objects being modeled changes with time. For example, a molecular dynamics program simulates the dynamic interactions among all atoms in a system of interest for a period of time. For each time

*Corresponding author.
E-mail addresses: ntfong@ece.eng.wayne.edu (N.-T. Fong), czxu@ece.eng.wayne.edu (C.-Z. Xu), lywang@ece.eng.wayne.edu (L.Y. Wang).

0743-7315/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2003.07.004

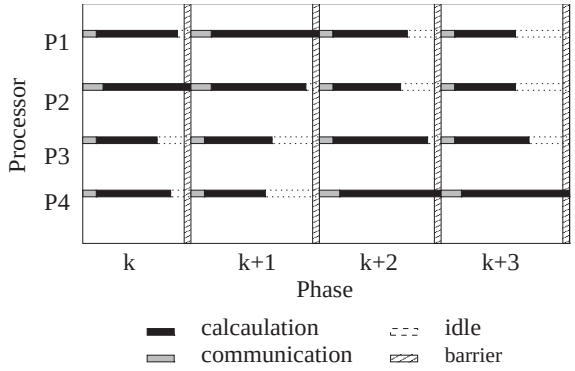


Fig. 1. An illustration of dynamic bulk synchronous computations.

step, the simulation calculates the forces between atoms, the energy of the whole structure, and the movements of atoms. Since atoms tend to move around, the amount of workloads with different parts of the system will change from one step to another as the atoms change their spatial positions. Dynamic computations can also appear in solution-adaptive problems. For example, a computational fluid dynamics code calculates the velocity and pressure of grid points for the purpose of deriving their structural and dynamic properties. In simulations that uses adaptive gridding to adjust the scale of resolution as the simulation progresses, computational workloads associated with different parts of a grid may change from phase to phase.

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. 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 [15,17,19,29]. 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. This paper provides some such analysis, assuming the workload change of a process is a random process with known or unknown probabilistic distributions. We formulated the problem as two complement sequential stochastic optimization model with different objectives and derived optimal remapping schedules for a given tolerance of load imbalance. (The non-zero tolerance of load imbalance is set according to the cost of remapping.) 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 [1,11,12,14,21]. 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.

The remainder of the paper is organized as follows. Section 2 presents a brief review of related work. Section 3 describes the computational model and formulates the problem. Sections 4 and 5 deal with the optimization problem subject to different constraints. Section 6 extends the analysis to heterogeneous platforms. The paper is concluded in Section 7 with remarks on future work.

2. Related work

Due to its popularity in many application areas, the dynamic bulk synchronous model has long been the subject of many researches. Marinescu and Rice quantitatively analyzed the effects of the load imbalance of a bulk synchronous computation, assuming the execution time of processes in an iteration are independent identical random variables [14]. Agrawal and Chakradhar modeled VLSI logic simulations as the dynamic bulk synchronous model [1]. By assuming the workload of a processor within a time step to be a binomial random variable, they derived an accurate estimate of the parallel simulation time. Madala and Sinclair derived asymptotic approximates and upper bounds for the average execution time of a bulk synchronous computation [12]. Similar results were obtained by Peterson and Chamberlain when they modeled a general class of discrete event simulation as the dynamic bulk synchronous model [21].

There were also studies that were intended to lessen the penalty due to synchronization and load imbalances via dynamic remapping [4,6–8,16,20,24,26,27,29]. Dynamic remapping strategies were demonstrated effective when they were applied to solution-adaptive CFD [8,27], multi-stage image understanding systems [4], and Monte Carlo dynamical simulations [7,26]. Dynamic remapping was shown important as well for applications that have no inherent synchronization requirements [17]. Nicol and Ciardo experimented with dynamic remapping at artifactual synchronization points created with real-time clocks. There were also time-efficient distributed remapping strategies that re-decompose the problem domain in parallel based on the previous distributions [29]. They are not sufficient to produce the highest quality partitions because of the lack of

global knowledge about the problem domain. The loss of their quality could be compensated for by its smaller run-time overhead in periodic remapping. Recently, Kwok et al. studied trade-offs between remapping complexity and quality [10]. They compared a dynamic approach using a fast on-line mapping heuristic and a semi-static strategy of dynamic invocation of off-line derived mappings and concluded the superiority of the semi-static approach.

Besides the issue of how to remap, another critical and complementary issue in remapping is when to invoke a remapping so that its performance gain will not be offset by its overhead. It can be either specified by the programmer in the form of a directive or be determined automatically by the compiler. Most of the existing policies are application-specific and heuristic. They were to perform static remapping algorithms periodically or in response to phase changes (e.g. grid refinement in the CFD code) as the computation proceeds. Nicol and Reynolds [18] modeled the remapping decision problem as a Markov decision process for computations where phases change with uncertainty and with radically different requirements. They determined the structure of the decision policy based on stochastic dynamic programming approaches.

For applications with gradually varying resource demands, Nicol and Saltz [19] proposed a simple provably good invocation policy, stop-at-rise (SAR). Moon and Saltz [15] applied the SAR invocation policy, coupled with an elegant chain-structured partitioner and a recursive coordinate bisection (RCB) partitioner, to three-dimensional direct Monte Carlo simulation methods. An alternative to the SAR is periodic remapping. Xu and Lau [29] and Nicol and Ciardo [17] experimented with periodic remapping policies against the remapping periodicity in parallel skeleton of images, WaTor simulation, and discrete state-space generations. Moon and Saltz [15] compared the periodic policy with the SAR strategy in their Monte Carlo simulations and found that the best fixed interval policy was able to deliver comparable performance to the SAR through the periodic policy. Nicol and Saltz [19] 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 this paper, we relax the assumption of independent and identical workload change to be independent with known or unknown distributions and present optimal or near-optimal remapping schedules in closed forms.

We note that this paper studies the periodic remapping policy, focusing on the impact of remapping frequencies. Most recently, we also investigated the general issue of dynamic remapping [31]. We modeled the bulk synchronous computations as an optimal stopping problem in stochastic control theory. By converting the stopping problem to an equivalent binary (i.e. remapping or not) decision over an infinity horizon, we derived conditions for the existence of optimal strategies and presented necessary and sufficient conditions for the optimality. Consequently, searches for the optimal controls were reduced to finding fixed points of a policy iteration equation. This iterative process converges at a geometric rate.

3. The model

Consider dynamic computations in a parallel computer with N processing nodes. The machine can be with either centralized memory or distributed memory organizations. Remapping is critical to the performance of the computation on centralized memory multiprocessors because parallel programs often rely on distributed data structures to exploit the data locality of caches and reduce the contention of access to shared variables. 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 6.

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, $i = 1, 2, \ldots, N$. Let $z_i(t)$ denote the amount of workload generated or finished from $t-1$ to t. Let the vectors $\mathbf{w}(t) = (w_1(t), w_2(t), \ldots, w_N(t))$ and $\mathbf{z}(t) = (z_1(t), z_2(t), \ldots, z_N(t))$ 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:

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

The distribution of the workload change $\mathbf{z}(t)$ has much bearing on the reward of remapping. Given some adaptive computations whose execution behavior is non-deterministic, it is possible for uniform distributions of $\mathbf{w}(t)$ in same phases become severely imbalance in the next phase. That is, for computations whose execution behavior is non-deterministic and non-predictable, there exists the possibility that a version with no balancing would outperform a dynamically balanced version, regardless of how well one can optimize the remapping procedure. Therefore, in order for remapping to be promising in leading to appreciable performance gains, the ideal arena in which to apply remapping would be the class of computations whose computational requirements vary gradually over time.

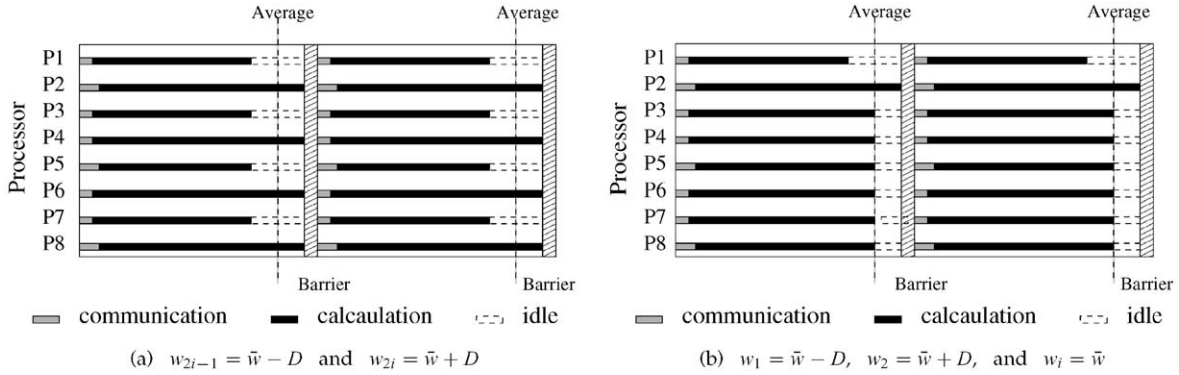


Fig. 2. Two different workload distributions under objective of $\mathscr{P}_1$. (a) $w_{2i-1} = \bar{w} - D$ and $w_{2i} = \bar{w} + D$. (b) $w_1 = \bar{w} - D$, $w_2 = \bar{w} + D$, and $w_i = \bar{w}$.

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 [21,19]. 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 tolerance.

Let $\bar{\mathbf{w}}(t) = (\bar{w}(t), \bar{w}(t), \ldots, \bar{w}(t))$, 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

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

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 change at a comparable rate between phases. average load level when a remapping occurs. The first objective of this study is to find the optimal 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:

$$\mathscr{P}_1: \begin{cases} \text{maximize} & T \\ \text{subject to} & \begin{cases} \mathbf{w}(t) = \mathbf{w}(t-1) + \mathbf{z}(t), \\ d(t) \leqslant D, \\ \mathbf{w}(t) \geqslant 0, \ \text{for all } t = 1, 2, \ldots, T. \end{cases} \end{cases}$$

Notice that the objective of $\mathscr{P}_1$ is to optimize the performance from the perspective of individual applications. Its optimal solution may not necessary lead to high efficient utilization of the available system resource. Fig. 2 shows two scenarios, where $w_{2i-1} = \bar{w} - D$ and $w_{2i} = \bar{w} + D$ for $1 \leqslant i \leqslant N/2$ in Fig. 2(a) and $w_1 = \bar{w} - D$, $w_2 = \bar{w} + D$, and $w_i = \bar{w}$ for $3 \leqslant i \leqslant N$ in Fig. 2(b). Clearly, both are optimal solutions in terms of the objective of $\mathscr{P}_1$. However, in practice, Fig. 2(b) is better than Fig. 2(a) because Fig. 2(b) tends to generate smaller degrees of load imbalance in subsequent phases. Note that a barrier operation in a large-scale distributed system takes substantial time even in the case that all processors arrive at the synchronization point simultaneously. The superiority of Fig. 2(b) is also due to its smaller cost for a barrier synchronization.

To reflect the desirable properties of Fig. 2(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:

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

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:

$$\mathscr{P}_{\mathrm{II}}: \begin{cases} \text{maximize} & T \\ \text{subject to} & \begin{cases} \mathbf{w}(t) = \mathbf{w}(t-1) + \mathbf{z}(t), \\ v(t) \leqslant B, \\ \mathbf{w}(t) \geqslant 0, \text{ for all } t = 1, 2, \ldots, T. \end{cases} \end{cases}$$

By the objective of problem $\mathscr{P}_{\mathrm{II}}$, the distribution of Fig. 2(a) is more desirable. On the other hand, the objective of $\mathscr{P}_{\mathrm{II}}$ should also be complemented by that of $\mathscr{P}_{\mathrm{I}}$. Fig. 3 shows an extreme scenario, in which $w_1 = \bar{w} + (N-1)\delta$ and $w_i = \bar{w} - \delta$, for $2 \leqslant i \leqslant 8$. 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 3.1.

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

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

Proof.

$$v^2(t) = \frac{E[\sum_{i=1}^{N}(w_i(t) - \bar{w}(t))^2]}{(E[\bar{w}(t)])^2} \\ = \frac{\sum_{i=1}^{N}E[(w_i(t) - \bar{w}(t))^2]}{(E[\bar{w}(t)])^2} \\ \leqslant \frac{\sum_{i=1}^{N}E[(\max_i |w_i(t) - \bar{w}(t)|)^2]}{(E[\bar{w}(t)])^2} \\ = N(d^2(t) + \theta^2).$$

Similarly,

$$v^2(t) \geqslant [d^2(t) + \theta^2]. \qquad \square$$

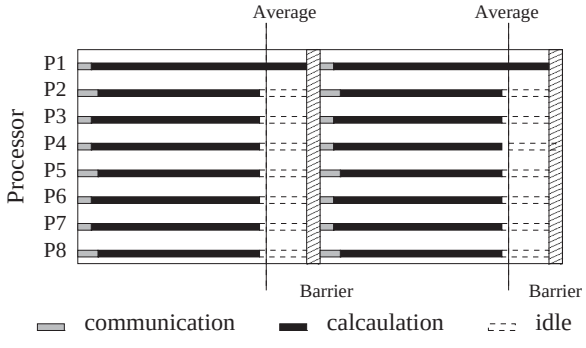


Fig. 3. An extreme workload distribution under objective of $\mathscr{P}_{\mathrm{II}}$.

In subsequent sections, we address these two optimization problems separately for computations that exhibit different statistical behaviors.

4. Optimal remapping frequency for problem $\mathscr{P}_{\mathrm{I}}$

First, we consider parallel computations that exhibit distribution-free and i.i.d. random variables $z_i(\cdot)$, $i = 1, 2, \ldots, N$ in the model of Eq. (1). The distribution-free random process model represents very general classes of bulk synchronous applications. 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 computations that are of Markovian or exponentially distributed random processes, their optimal remapping frequencies will be derived.

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

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

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

4.1. Distribution-free workload changes

To analyze the effect of periodic remapping on distribution-free workload processes, the only assumption we need to make is that the i.i.d. random variables $z_i(t)$, $i = 1, 2, \ldots, N$, are of the same mean μ and variance σ^2. It holds in most data parallel applications on homogeneous computer systems because processes on a portion of data domain normally exhibit the same computational characteristics as the overall computations. Assume processors initially have the equal workload and their initial workloads $w_i(0) = w$. By Eq. (1), the workload of processor i at time t is

$$w_i(t) = w_i(t-1) + z_i(t) = w + \sum_{j=1}^{t} z_i(j).$$

Then,

$$E[w_i(t)] = w + \sum_{j=1}^{t} E[z_i(j)] = w + t\mu,$$

$$Var(w_i(t)) = Var(w) + \sum_{j=1}^{t} Var(z_i(j)) = t\sigma^2.$$

Thus, the workloads of N processors at any time t, $w_1(t), w_2(t), \ldots, w_N(t)$, are i.i.d. random variables with the same mean $w + t\mu$ and variance $t\sigma^2$. Therefore, using well-known result from order statistics [5], we have

$$E\left[\max_{i=1,2,\ldots,N} w_i(t)\right] \leqslant w + t\mu + \frac{N-1}{\sqrt{2N-1}}\sqrt{t}\sigma. \tag{7}$$

Thus,

$$\begin{aligned} d_1(t) &= \frac{E[\max_{i=1,2,\ldots,N} w_i(t) - \bar{w}(t)]}{E[\bar{w}(t)]} \\ &= \frac{E[\max_{i=1,2,\ldots,N} w_i(t)] - E[\bar{w}(t)]}{E[\bar{w}(t)]} \\ &\leqslant \frac{w + t\mu + \frac{N-1}{\sqrt{2N-1}}\sqrt{t}\sigma - (w + t\mu)}{w + t\mu} \\ &= \frac{(N-1)\sqrt{t}\sigma}{\sqrt{2N-1}(w + t\mu)}, \end{aligned}$$

and likewise

$$d_2(t) \leqslant \frac{(N-1)\sqrt{t}\sigma}{\sqrt{2N-1}(w + t\mu)}.$$

For a given bound D of the normalized extreme workload difference, if

$$\hat{d}(t) \equiv \frac{(N-1)\sqrt{t}\sigma}{\sqrt{2N-1}(w + t\mu)} \leqslant D, \tag{8}$$

then, neither $d_1(t)$ or $d_2(t)$ is larger than D. Thus, our objective is to find the maximum T^* such that Eq. (7) holds for $t = 1, 2, \ldots, T^*$. The T^* is a lower bound of the optimal interval for the problem $\mathscr{P}_1$.

Fig. 4 plots $\hat{d}(t)$ without remapping in the case $\mu \neq 0$. It reaches its maximum value $D^* = \frac{(N-1)\sigma}{2\sqrt{(2N-1)w\mu}}$ at the time of $T_0 = w/\mu$. 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 w/μ. 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.

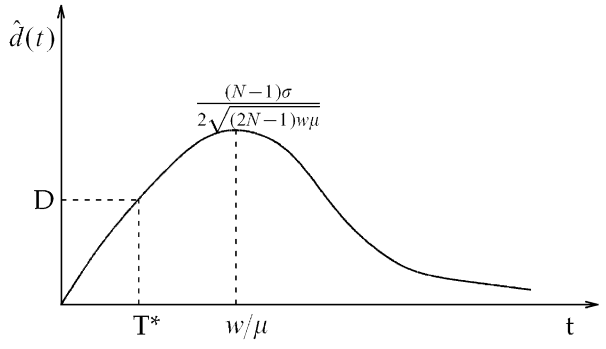


Fig. 4. Illustration of normalized extreme workload difference: $\mu \neq 0$.

We present the dependent relationship between the bound D and remapping frequency T^* in the following theorem.

Theorem 4.1. *Assume the workload change, $z_i(\cdot)$, $i = 1, 2, \ldots, N$ are i.i.d. arbitrarily distributed random variables with the same mean μ and variance σ^2. For a given small bound D of the normalized extreme workload difference, a lower bound of the optimal remapping interval for problem $\mathscr{P}_1$ is*

1. *if $\mu \neq 0$, for a given $D = \frac{D^*}{m}$, $m = 1, 2, \ldots$,*

$$T^* = (2m^2 - 1 - 2m\sqrt{m^2 - 1})T_0, \tag{9}$$

where $T_0 = w/\mu$;

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

$$T^* = \frac{(2N-1)D^2w^2}{(N-1)^2\sigma^2}. \tag{10}$$

Proof. From Eq. (7), if $\mu \neq 0$, we obtain that for a given small bound which is not larger than $\hat{d}^*(t)$,

$$t \leqslant \frac{(N-1)^2\sigma^2 - 2(2N-1)w\mu D^2 - (N-1)\sigma\sqrt{\Delta_1}}{2(2N-1)D^2\mu^2} \equiv t_1,$$

$$t \geqslant \frac{(N-1)^2\sigma^2 - 2(2N-1)w\mu D^2 + (N-1)\sigma\sqrt{\Delta_1}}{2(2N-1)D^2\mu^2} \equiv t_2,$$

where $\Delta_1 \equiv (N-1)^2\sigma^2 - 4(2N-1)w\mu D^2$. It can be shown that $0 \leqslant t_1 \leqslant t_2$. Thus,

$$\begin{aligned} T^* = t_1 &= \frac{(N-1)^2\sigma^2 - 2(2N-1)w\mu D^2 - (N-1)\sigma\sqrt{\Delta_1}}{2(2N-1)D^2\mu^2} \\ &= (2m^2 - 1 - 2m\sqrt{m^2-1})\frac{w}{\mu} \\ &= (2m^2 - 1 - 2m\sqrt{m^2-1})T_0. \end{aligned}$$

If $\mu = 0$, we directly solve Eq. (7) and obtain

$$T^* = \frac{(2N-1)D^2w^2}{(N-1)^2\sigma^2}. \qquad \square$$

Theorem 4.1 shows the optimal interval T^* is dependent on the bound of load imbalance degree D and processors' initial workload as well. The dependence on initial workload W is because $d(t)$ is defined as a relative workload difference to the workload mean. The impact of initial workload distribution cannot be

completely isolated due to the workload change $z_i(t)$ being relative to the workload mean. Theorem 4.1 also shows that the remapping interval T^* increases with bound D. That is, given a looser bound (possibly due to heavy overhead of remapping), computations can proceed without remapping for a longer time.

As t gets large, according to the central limit theorem of statistics [23], the N i.i.d. random variables $w_1(t), w_2(t), \ldots, w_N(t)$, would tend to become normally distributed with the same mean $w + t\mu$ and variance $t\sigma^2$, as t gets large. Combined with the extreme value theory [2], it follows that

$$E\left[\max_{i=1,2,\ldots,N} w_i(t)\right] \approx w + t\mu + \alpha(N)\sigma\sqrt{t}, \tag{11}$$

where

$$\alpha(N) = (2\ln N)^{1/2} - \frac{\ln\ln N + \ln 4\pi}{2(2\ln N)^{1/2}} + \frac{\gamma}{(2\ln N)^{1/2}}$$

and γ is Euler's constant $(0.5772\cdots)$. To ensure $d(t) \leqslant D$, we conservatively set

$$\tilde{d}(t) \equiv \frac{\alpha(N)\sqrt{t}\sigma}{w + t\mu} \leqslant D, \tag{12}$$

for a given bound D. It can be easily shown that $\tilde{d}(t)$ reaches its maximum value $D^* = \frac{\alpha(N)\sigma}{2\sqrt{w\mu}}$ at the time of $T_0 = w/\mu$. Therefore, we can derive a tighter bound for the remapping interval, as shown in Theorem 4.2.

Theorem 4.2. *Assume the workload change, $z_i(\cdot)$, $i = 1, 2, \ldots, N$ are i.i.d. arbitrarily distributed random variables with the same mean μ and variance σ^2. For a given loose bound D of the normalized extreme workload difference, a tight bound of the optimal interval for the problem $\mathscr{P}_{\mathrm{I}}$ is*

1. *if $\mu \neq 0$, for a given $D = \frac{D^*}{m}$, $m = 1, 2, \ldots$,*

$$T^* = (2m^2 - 1 - 2m\sqrt{m^2 - 1})T_0, \tag{13}$$

where $T_0 = w/\mu$;

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

$$T^* = \frac{D^2 w^2}{[\alpha(N)]^2\sigma^2}. \tag{14}$$

Note that the bound T^* in the above theorem is proven to be near optimal as the remapping interval becomes large enough due to the central limit theorem. However, how large the remapping interval should be for an accurate approximation is still open in theory. Simulations in Section 4.3 will show that its accuracy is sustained even in the case of frequent remapping with an interval of as small as 8 steps.

4.2. Exponentially distributed workload changes

Adaptive computations are often modeled by a Markovian birth-death process in the literature [19,22]. In general, we assume the workload of processors $z_i(\cdot)$ are i.i.d. random variables with an exponential distribution. Then, the total workload change of a processor at time t, i.e. the summation of $z_i(t)$, $\hat{w}_i(t) \equiv \sum_{j=1}^{t} z_i(j)$, is of gamma distribution, with the density function

$$f_i(x) = \begin{cases} \frac{\mu_i^{-t} x^{t-1} e^{-x/\mu_i}}{\Gamma(t)} & \text{if } x > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

where $\Gamma(t)$ is the gamma function, defined by $\Gamma(t) = \int_0^\infty s^{t-1}e^{-s}\,ds$ for any $t > 0$. Correspondingly, the probability distribution function of the random variable $\hat{w}_i^t$ is

$$F_i(x) = \begin{cases} 1 - e^{-x/\mu_i}\sum_{j=0}^{t-1}\frac{(x/\mu_i)^j}{j!} & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Using order statistics, we obtain the mean of the maximum of the above N independent gamma distributed random variables $\hat{w}_i(t)$ at time t for $i = 1, 2, \ldots, N$:

$$E\left[\max_{i=1,2,\ldots,N} \hat{w}_i(t)\right] = \int_0^\infty x\left\{\prod_{i=1}^{N} F_i(x)\right\}\sum_{i=1}^{N}\frac{f_i(x)}{F_i(x)}\,dx. \tag{17}$$

Let $g(t)$ denote the right-hand side of Eq. (15). Therefore,

$$\begin{aligned} d_1(t) &= \frac{E[\max_{i=1,2,\ldots,N} w_i(t) - \bar{w}(t)]}{E[\bar{w}(t)]} \\ &= \frac{w + E[\max_{i=1,2,\ldots,N}\hat{w}_i(t)] - E[\bar{w}(t)]}{E[\bar{w}(t)]} \\ &= \frac{w + g(t) - (w + t\bar{\mu})}{w + t\bar{\mu}} \\ &= \frac{g(t) - t\bar{\mu}}{w + t\bar{\mu}}, \end{aligned}$$

where $\bar{\mu} = \sum_{i=1}^{N}\mu_i/N$, and likewise

$$d_2(t) = -\frac{g(t) - t\bar{\mu}}{w + t\bar{\mu}}.$$

For a given bound D, our objective is to find the maximum T^* such that

$$d_i(t) \leqslant D \quad \text{for } i = 1, 2, \quad t = 1, 2, \ldots, T^*.$$

Thus, we have the following results.

Theorem 4.3. *Assume the workload change of processors $z_i(\cdot)$, $i = 1, 2, \ldots, N$ are i.i.d. random variables with an exponential distribution of mean μ_i and variance σ_i^2, respectively. For a given bound D of the normalized extreme workload difference, the optimal remapping interval for the problem $\mathscr{P}_{\mathrm{I}}$ is the solution T^* of the*

Table 1
Optimal remapping intervals for $\mathscr{P}_1$ with death–birth modeled workload changes

D	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.14
T^*_{free}	1	2	3	4	5	6	7	9	10	12
T^*_{normal}	8	12	17	22	28	34	42	50	58	68
T^*_{simu}	7	11	15	19	25	31	35	41	49	56

following inequality:

$$\left|\frac{g(t)-t\bar{\mu}}{w+t\bar{\mu}}\right| \leqslant D \quad \text{for } t=1,2,\ldots,T^*. \tag{18}$$

Although we cannot derive the optimal remapping interval T^* in a closed form due to the complexity of the term $g(t)$, its approximation can be easily obtained by numerical methods.

4.3. Simulation results

Note that T^* in Theorem 4.1 is a conservative estimate of the remapping interval subject to the optimization constraints of problem $\mathscr{P}_1$. To illustrate the accuracy of the estimate, we conducted two experiments 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 first experiment assumed that processors change their workload units following a distribution function

$$z_i(t)=\begin{cases}1, & \text{w.p. } 0.25,\\ 0, & \text{w.p. } 0.5,\\ -1, & \text{w.p. } 0.25,\end{cases} \tag{19}$$

where w.p. means "with probability". It is a typical death-birth Markov chain model. Similar distribution functions were also considered by other researchers [19].

Table 1 shows the simulation results for different bounds of normalized extreme workload difference. For comparison, the theoretical results, T^*_{free} of Theorem 4.1 (due to distribution-free workload changes), T^*_{normal} of Theorem 4.2, and the practical simulation results T^*_{simu}, are included. The table confirms the conservativity of T^*_{free}. By this estimate, the bulk synchronous computations would rely on much more than necessary remapping operations to ensure load balance. It is because the expected results do not assume any distribution information with the workload change. Although conservative, they serve as lower bounds of the remapping frequency. That is, without any knowledge about the workload change, periodic remapping at intervals of T^*_{free} guarantees the load imbalance degree to be bounded by a pre-defined threshold D.

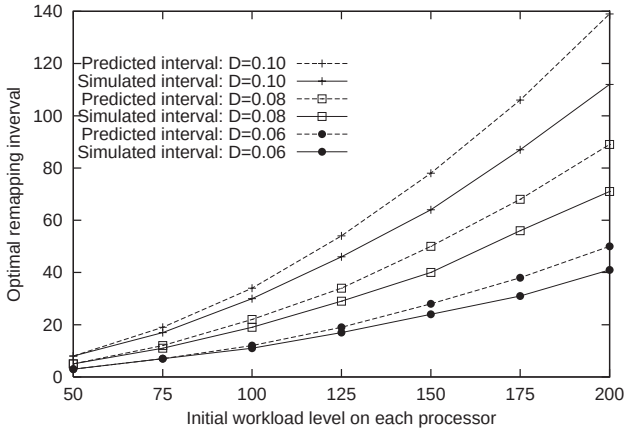


Fig. 5. Optimal remapping intervals with different initial workload levels.

By contrast, it can be seen that T^*_{normal} is quite accurate in all the test cases, even though T^*_{normal} is approximated for the case of large t in theory. For example, for a given $D=0.06$, T^*_{normal} requires a computation to perform remapping operations every 12 steps. It is close to the 11 steps obtained from simulations. It suggests that Theorem 4.2 be applicable to real situations.

Recall that Theorem 4.1 shows T^* depends on processors' initial workload, as well as the normalized extreme workload difference D. Fig. 5 confirms the results with plots of the optimal remapping interval T^*_{normal} versus different initial workload. In comparison with T^*_{simu}, the figure also shows that T^*_{normal} provides an optimistic estimate of the remapping interval. The actual remapping frequency should be slightly higher than estimations from the central limited theorem. Accuracy of the estimation drops with the increase of the initial workload because we assumed each processor changes its workload by a constant unit, independent of its actual workload at run-time. Treatment of situations where the workload change at each step is relative to workload distributions is beyond the scope of this paper and are to be investigated in the future.

In the second experiment, we assumed that processors changed their workload units according to an exponential distribution function with the same mean $\mu=0.5$. Table 2 presents the simulation results, together with the corresponding theoretical results, T^*_{exp} of Theorem 4.3. Evidently, T^*_{exp} is highly accurate, in particular in the

Table 2
Optimal remapping intervals for $\mathscr{P}_{\mathrm{I}}$ with exponentially distributed workload changes

D	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
T^*_{exp}	1	2	4	8	13	18	27	40	73
T^*_{simu}	1	2	4	8	14	20	30	45	76

case of small load imbalance tolerance (i.e. tight bound of extreme workload difference).

5. Optimal remapping frequency for problem $\mathscr{P}_{\mathrm{II}}$

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

5.1. 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. As shown in Fig. 6, it is the stationary workload deviation when the computations proceeds without remapping.

Lemma 5.1. *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*

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

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

Proof. According to Eq. (1), the workload of processor i at time t is

$$w_i(t) = w_i(t-1) + z_i(t) = w + \sum_{j=1}^{t} z_i(j).$$

Thus,

$$\begin{aligned} E[w_i(t)] &= w + \sum_{j=1}^{t} E[z_i(j)] \\ &= w + t\mu_i, \\ E[\bar{w}(t)] &= E\left[\sum_{i=1}^{N} w_i(t)/N\right] \\ &= \sum_{i=1}^{N} E[w_i(t)]/N = w + t\bar{\mu}, \end{aligned}$$

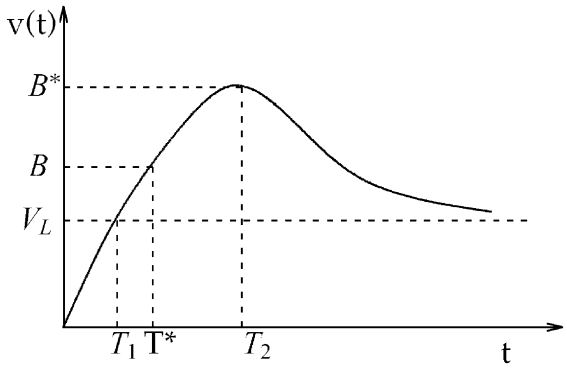


Fig. 6. Illustration of normalized workload deviation.

$$\begin{aligned} E[w_i^2(t)] &= Var(w_i(t)) + (E[w_i(t)])^2 \\ &= t\sigma_i^2 + (w + t\mu_i)^2, \\ E[\bar{w}^2(t)] &= Var(\bar{w}(t)) + (E[\bar{w}(t)])^2 \\ &= t\bar{\sigma}^2/N + (w + t\bar{\mu})^2. \end{aligned}$$

Hence,

$$\begin{aligned} &E\left[\sum_{i=1}^{N} (w_i(t) - \bar{w}(t))^2\right] \\ &= \sum_{i=1}^{N}\{E[w_i^2(t)] - 2E[w_i(t)\bar{w}(t)] + E[\bar{w}^2(t)]\} \\ &= (N-1)\bar{\sigma}^2 t + \left[\sum_{i=1}^{N}\mu_i^2 - N\bar{\mu}^2\right]t^2. \end{aligned}$$

It follows that

$$\begin{aligned} v(t) &= \frac{\sqrt{E[\sum_{i=1}^{N}(w_i(t) - \bar{w}(t))^2]}}{E[\bar{w}(t)]} \\ &= \frac{\sqrt{(N-1)\bar{\sigma}^2 t + [\sum_{i=1}^{N}\mu_i^2 - N\bar{\mu}^2]t^2}}{w + t\bar{\mu}}. \end{aligned} \tag{20}$$

Consequently,

$$V_{\mathrm{L}} = \lim_{t\to\infty} v(t) = \frac{\sqrt{\sum_{i=1}^{N}\mu_i^2 - N\bar{\mu}^2}}{\bar{\mu}}. \qquad \square$$

For a given bound B, we want to find the maximum T^* such that $v(t)\leqslant B$. Fig. 6 shows the workload change with time. Let B^* denote the maximum value of $v(t)$. We

can prove that

$$B^* = \frac{(N-1)\bar{\sigma}^2}{2\sqrt{w\bar{\mu}[(N-1)\bar{\sigma}^2 - w\bar{\mu}V_L^2]}},$$

and $v(t)$ reaches the maximum value B^* at time

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

Note that $v(t)$ increases to V_L at time

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

Consequently, we obtain the following theorem.

Theorem 5.1. *Assume the workload change of processors $z_i(\cdot)$, $i = 1, 2, \ldots, N$ 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 $\mathscr{P}_{II}$ is*

1.

$$T^* = \begin{cases} T_1 & \text{if } B = V_L, \\ T_2 & \text{if } B = B^*. \end{cases} \tag{21}$$

2. *For a given B, $0 < B < B^*$ and $B \neq V_L$, there exits $m \notin [0,1]$ such that $B^2 = V_L^2 + \frac{1}{m}(B^{*2} - V_L^2)$. Then,*

$$T^* = (2m - 2m\sqrt{1 - 1/m} - 1)T_2 - 2(m - m\sqrt{1 - 1/m} - 1)T_1. \tag{22}$$

Proof. For a given bound B, we want to find the maximum T^* which satisfies $v(t) \leqslant B$. It follows that for $t = 1, 2, \ldots, T^*$,

$$\left[B^2\bar{\mu}^2 - \left(\sum_{i=1}^{N} \mu_i^2 - N\bar{\mu}^2\right)\right]t^2 - [(N-1)\bar{\sigma}^2 - 2B^2 w\bar{\mu}]t + B^2 w^2 \geqslant 0. \tag{23}$$

In order to obtain real number solutions to the above inequality, the bound B must be set to be less than or equal to the maximum value of $v(t)$. For such a small bound, Eq. (23) leads to the following solutions:

1. if $B = B^*$, $t \leqslant T_2$;
2. if $B - V_L > 0$,

$$t \leqslant \frac{(N-1)\bar{\sigma}^2 - 2w\bar{\mu}B^2 - \sqrt{\Delta_3}}{2(B^2 - V_L^2)\bar{\mu}^2} \equiv T_3,$$

$$t \geqslant \frac{(N-1)\bar{\sigma}^2 - 2w\bar{\mu}B^2 + \sqrt{\Delta_3}}{2(B^2 - V_L^2)\bar{\mu}^2} \equiv T_4,$$

where $\Delta_3 \equiv [(N-1)\bar{\sigma}^2 - 2B^2 w\bar{\mu}]^2 + 4B^2 w^2[\sum_{i=1}^{N}\mu_i^2 - (N + B^2)\bar{\mu}^2]$, and it is easily shown that $0 \leqslant T_3 \leqslant T_4$;

3. if $B - V_L = 0$, $t \leqslant T_1$;
4. if $B - V_L < 0$, $T_4 \leqslant t \leqslant T_3$, where it is easily shown that $T_3 \geqslant 0$ and $T_4 \leqslant 0$.

Thus, the optimal remapping interval for the problem $\mathscr{P}_{II}$ is

$$T^* = \begin{cases} T_1 & \text{if } B = V_L, \\ T_2 & \text{if } B = B^*, \\ T_3 & \text{if } B \neq V_L,\ 0 < B < B^*. \end{cases} \tag{24}$$

For a given B, $0 < B < B^*$ and $B \neq V_L$, there exits $m \notin [0,1]$ such that $B^2 = V_L^2 + \frac{1}{m}(B^{*2} - V_L^2)$. In order to simplify the proof, let

$$A_1 \equiv (N-1)\bar{\sigma}^2 - w\bar{\mu}V_L^2, \quad A_2 \equiv (N-1)\bar{\sigma}^2 - 2w\bar{\mu}V_L^2.$$

Then,

$$T_1 = \frac{w^2 V_L^2}{A_2}, \quad T_2 = \frac{w(N-1))\bar{\sigma}^2}{\bar{\mu}A_2} \quad \text{and} \quad T_2 - 2T_1 = \frac{w}{\bar{\mu}}.$$

Thus,

$$T_3 = \frac{(N-1)\bar{\sigma}^2 - 2w\bar{\mu}B^2 - \sqrt{\Delta_3}}{2(B^2 - V_L^2)\bar{\mu}^2} = (2m - 2m\sqrt{1 - 1/m} - 1)T_2 - 2(m - m\sqrt{1 - 1/m} - 1)T_1.$$

Therefore, this theorem is proved. □

Since the workload change $z_i(t)$, $i = 1, 2, \ldots, N$, 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

$$v_1(t) = \frac{\sqrt{(N-1)\sigma^2 t}}{w + t\mu}, \tag{25}$$

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

The maximum value of $v_1(t)$, B^*, is equal to $\sqrt{\frac{(N-1)\sigma^2}{4w\mu}}$ when $T_0 = w/\mu$, as shown in Fig. 7. Comparing with $\hat{d}(t)$ in Eq. (7), it is interesting to see that

$$v_1(t) = \sqrt{\frac{2N-1}{N-1}}\,\hat{d}(t). \tag{26}$$

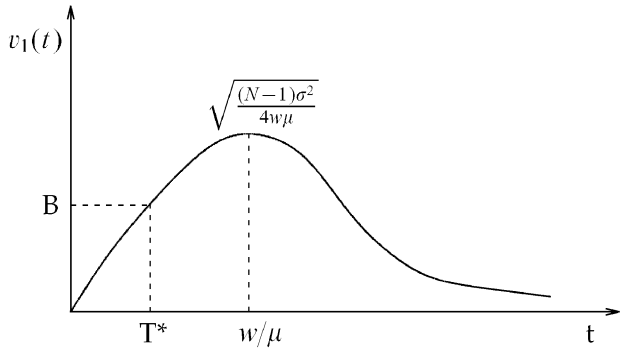


Fig. 7. Illustration of normalized workload deviation: same mean μ ($\mu \neq 0$) and variance σ^2.

Recall that $\hat{d}(t)$ is a conservative index of extreme workload difference $d(t)$ due to the use of order statistics in analysis. Eq. (21) shows that conservative results in terms of $\mathscr{P}_{\mathrm{I}}$ are corresponding to the optimal results in problem $\mathscr{P}_{\mathrm{II}}$. We present the optimal remapping frequency with respect to $v_1(t)$ as follows.

Corollary 5.1. *Assume the workload change $z_i(\cdot)$, $i = 1, 2, \ldots, N$ 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 $\mathscr{P}_{\mathrm{II}}$ is*

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

$$T^* = \frac{w^2 B^2}{(N-1)\sigma^2}; \tag{27}$$

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

$$T^* = (2m^2 - 1 - 2m\sqrt{m^2-1})T_0, \tag{28}$$

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

5.2. Simulation results

We conduced a similar experiment to the first one of the preceding section to simulate distribution-free multiple birth–death Markov chains on 64 processors. Assume workloads are initially balanced with 100 units. Table 3 shows the simulation results from different bounds B of the normalized workload deviation, together with the corresponding theoretical results, T^*_{free} of Corollary 5.1. 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 Fig. 8 with different initial workload levels.

By Lemma 3.1, we know the relationships between the extreme workload difference $d(t)$ and workload deviation $v(t)$. Comparing Table 3 with Table 1, we can see the remapping frequencies by the two objectives of $\mathscr{P}_{\mathrm{I}}$ and $\mathscr{P}_{\mathrm{II}}$ for given D and B are in agreement with the relationship uncovered in the lemma. For example, in the case of $D = 0.1$, the maximum remapping interval by Table 3 is 31 iteration steps. According to Lemma 3.1, the range of B corresponding to the D is approximately between 0.1 and 0.8 because $N = 64$. From Table 3, it can be seen their maximum remapping intervals are close to 31.

6. Remapping on 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. There were many recent studies on mapping and remapping on heterogeneous computing systems (see [10,13,25] for examples). Their focuses were mostly on the issue of how to map and remap. This section emphasizes on its complementary issue of when to remap.

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

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

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

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

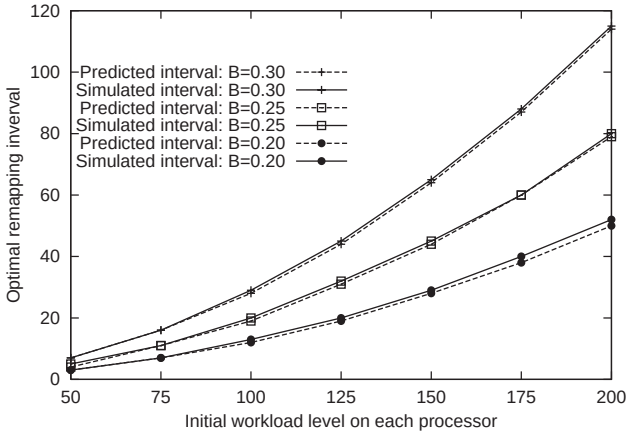


Fig. 8. Optimal remapping interval with different initial workload levels.

Table 3
Optimal remapping intervals for $\mathscr{P}_{\mathrm{II}}$ with same mean and variance distributed workload changes

B	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55
T^*_{free}	3	7	12	19	28	38	50	64	79	96
T^*_{simu}	3	7	13	20	29	39	51	65	79	97

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

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

and the normalized workload deviation at time t as

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

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 [3] through profiling NAS benchmark programs, running the same code with identical inputs on different machines may lead to execution time with different distributions. Given independent $z_i(t)$, the normalized workload change $z_i(t)/c_i$ are still independent. Then, the results for Problem $\mathscr{P}_{\mathrm{II}}$ in Section 5 still go through because of the distribution-free assumption on $z_i(t)$. In the following, we will solve the generalized problem $\mathscr{P}_{\mathrm{I}}$ 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 $Y = \max_{i=1,2,\ldots,N}\hat{w}_i(t)$. It is a random variable with the probability distribution function

$$F_Y^t(y) = \prod_{i=1}^{N} F_i^t(y), \tag{29}$$

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

$$f_Y^t(y) = \sum_{j=1}^{N} f_i^t(y) \prod_{i=1,\, i\neq j}^{N} F_i^t(y), \tag{30}$$

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

$$E\left[\max_{i=1,2,\ldots,N} \hat{w}_i(t)\right] = \int_0^{\infty} y f_Y^t(y)\, dy. \tag{31}$$

Notice that the density function $f_i^t(y)$ and distribution function $F_i^t(y)$ in Eq. (25) are presented in most general forms. Two examples of the functions are Eqs. (14) and (16) in Section 4.2. That is, when the normalized workload change $z_i(t)/c_i$ at different processors are i.i.d. random variables in an exponential distribution with different mean μ_i, the density and distribution functions $f_i^t(y)$ and $F_i^t(y)$ in Eq. (25) can be represented in the same closed forms as Eq. (14) and Eq. (16).

Let $\bar{\mu} = \sum_{i=1}^{N} \mu_i$, and $h(t)$ denote the right hand side of Eq. (26). In a similar way to the proof of Theorem 4.3, we obtain the following results.

Theorem 6.1. *Assume the normalized workload change $z_i(\cdot)/c_i$, $i = 1, 2, \ldots, N$ are independent random variables with different means μ_i and different distributions. For a given bound D of the normalized extreme workload difference, the optimal remapping interval for the problem $\mathscr{P}_{\mathrm{I}}$ is the solution T^* of the following inequality*:

$$\left|\frac{h(t) - t\bar{\mu}}{w + t\bar{\mu}}\right| \leqslant D \quad \textit{for } t = 1, 2, \ldots, T^*. \tag{32}$$

We remark that Theorem 6.1 is a generalization of Theorem 4.3 because Theorem 6.1 does not assumes any distribution information about the workload change $z_i(t)/c_i$ or its summation $\hat{w}_i(t)$. Theorem 4.3 holds in a special case that the workload changes $z_i(t)/c_i$ on different processors, $1 \leqslant i \leqslant N$, are i.i.d. random variables in exponential distributions. The second numerical experiment in Section 4.3 verifies the result in Theorem 4.3.

7. Concluding remarks

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 and normalized workload deviation from uniform distributions.

Unlike the general stochastic optimization approaches, which tend to reveal asymptotic or stationary properties of a random process, the analytical approaches we have developed are intended to capture the transient features of a random process because remapping operations are intended to be invoked at the beginning of the process. 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 solid via simulations.

Note that the analysis in this paper assumes statistical independence between processors' workloads for tractability. However, workloads in many real situations may be correlated. For example, in molecular dynamics applications, atoms tend to move around and conse-

quently the workload change between adjacent subdomains are highly related. Future work will be to extend the model to take into account the covariance of workload change $Cov[w_i, w_j]$ between processors i and j.

The periodic remapping policies reported in this paper assumed a priori knowledge about the workload change distributions and the cost for each remapping operation. Workload or execution time estimation on its own is an important topic. In [30], we derived tight bounds on execution time of bulk synchronous computations. Methodologies for determining phase-wise execution time distributions based on workload information in previous phases of a bulk synchronous computation need to be further studied. Future work will also include the estimation and/or measurement of remapping cost for a given remapping algorithm. Log P performance model provides a good way for such estimations.

Acknowledgments

We thank the anonymous referees for their reviews. This work was supported in part by the US National Science Foundation under Grants CCR-9988266 and ACI-0203592. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

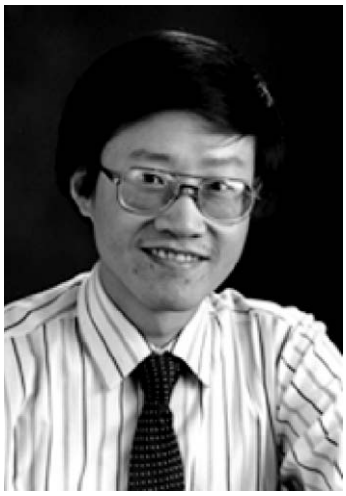
References

[1] V.D. Agrawal, S.T. Chakradhar, Performance analysis of synchronized iterative algorithms on multiprocessor systems, IEEE Trans. Parallel Distrib. Systems 3 (6) (1992) 739–746.
[2] A.H.-S. Ang, W.H. Tang, Probability Concepts in Engineering Planning and Design, Vol. II, Rainbow Bridge, John Wiley & Sons, 1984.
[3] R. Armstrong, D. Hensgen, T. Kidd, The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions, Proceedings of the Seventh Heterogeneous Computing Workshop, Orlando, FL, 1998.
[4] A.N. Choudhary, B. Narahari, R. Krishnamurti, An efficient heuristic scheme for dynamic remapping of parallel computations, Parallel Comput. 19 (1993) 621–632.
[5] H.A. David, Order Statistics, 2nd Edition, Wiley, New York, 1981.
[6] G. Edjlali, G. Agrawal, A. Sussman, J. Humphries, J. Saltz, Compiler and runtime support for programming in adaptive parallel environments, Sci. Programming 6 (1997) 215–227.
[7] R.V. Hanxleden, L.R. Scott, Load balancing on message passing architectures, J. Parallel Distrib. Comput. 13 (3) (1991) 312–324.
[8] J. De Keyser, D. Roose, Multigrid with solution-adaptive irregular grids on distributed memory computers, in: D.J. Evans, G.R. Joubert, H. Liddell (Eds.), Parallel Computing, Elsevier Science Publishers, Amsterdam, 1992, pp. 375–382.
[9] Y.-K. Kwok, I. Ahmad, Static scheduling algorithms for allocating directed task graphs to multiprocessors, ACM Comput. Surveys 31 (4) (1999) 406–471.
[10] Y.-K. Kwok, A. Maciejewski, H.J. Siegel, A. Ghafoor, I. Ahmad, Evaluation of a semi-static approach to mapping dynamic iterative tasks onto heterogeneous computing systems, Proceedings of the Fourth International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN'99), Freemantle, Australia, June 1999, pp. 204–209.
[11] K. Li, Stochastic bounds for parallel program execution times with processor constrains, IEEE Trans. Comput. 46 (1997) 630–636.
[12] S. Madala, J.B. Sinclair, Performance of synchronous parallel algorithms with regular structures, IEEE Trans. Parallel Distrib. Systems 2 (1) (1991) 105–116.
[13] M. Maheswaran, H. Siegel, A dynamic matching and scheduling algorithm for heterogeneous computing systems, Proceedings of the 7th Heterogeneous Computing Workshop, Orlando, FL, 1998, pp. 57–69.
[14] D.C. Marinescu, J.R. Rice, Synchronization and load imbalance effects in distributed memory multi-processor systems, Concurrency: Pract. Exper. 3 (6) (1991) 593–625.
[15] B. Moon, J. Saltz, Adaptive runtime support for direct simulation Monte Carlo methods on distributed memory architectures, Scalable High-Performance Computing Conference, Knoxville, TN, May 1994, pp. 176–183.
[16] J.E. Moreira, V.K. Naik, S.P. Midkiff, Dynamic data distribution and processor repartitioning for irregularly structured computation, J. Parallel Distrib. Comput. 50 (1998) 28–60.
[17] D. Nicol, G. Ciardo, Automated parallelization of discrete state-space generation, J. Parallel Distrib. Comput. 47 (2) (1997) 153–167.
[18] D.M. Nicol, P.F. Reynolds, Optimal dynamic remapping of data parallel computation, IEEE Trans. Comput. 39 (2) (1990) 206–219.
[19] D.M. Nicol, J.H. Saltz, Dynamic remapping of parallel computations with varying resource demands, IEEE Trans. Comput. 37 (9) (1988) 1073–1087.
[20] L. Oliker, R. Biswas, Efficient load balancing and data remapping for adaptive grid calculations, In Proceedings of SPAA'97, Newport, RI, 1997.
[21] G.D. Peterson, R.D. Chamberlain, Beyond execution time: expanding the use of performance models, IEEE Parallel Distrib. Technol. Vol. 2, 1994, pp. 37–49.
[22] Y. Rabani, A. Sinclair, R. Wanka, Local divergence of Markov chains and the analysis of iterative load-balancing schemes, Proceedings of the 39th Annual Foundations of Computer Science, Palo Alto, CA, 1998.
[23] S.M. Ross, Introduction to Probability Models, 6th Edition, Academic Press, New York, 1997.
[24] W. Shu, M.-Y. Wu, Runtime incremental parallel scheduling (RIPS) on distributed memory computers, IEEE Trans. Parallel Distrib. Systems 7 (6) (1996) 637–649.
[25] H. Topcuoglu, S. Hariri, M.-Y. Wu, Performance-effective and low-complexity task scheduling for heterogeneous computing, IEEE Trans. Parallel Distrib. Systems 13 (3) (2002) 260–274.
[26] J. Watts, S. Taylor, A practical approach to dynamic load balancing, IEEE Trans. Parallel Distrib. Systems 9 (3) (1998) 235–248.
[27] R.D. Williams, Performance of dynamic load balancing algorithms for unstructured mesh calculations, Concurrency: Practice Exper. 3 (5) (1991) 451–481.
[28] C. Xu, F. Lau, Load Balancing in Parallel Computers: Theory and Practice, Kluwer Academic Publishers, Dordrecht, 1997.
[29] C. Xu, F. Lau, R. Diekmann, Decentralized remapping of data parallel applications in distributed memory multiprocessors, Concurrency: Practice Exper. 9 (1997) 1351–1376.
[30] C. Xu, L. Wang, N.-T. Fong, Stochastic prediction of execution time for dynamic bulk synchronous computations, J. Supercomputing 21 (2002) 91–103.

[31] G. Yin, C. Xu, L. Wang, Optimal remapping in dynamic bulk synchronous computations via a stochastic control approach, IEEE Trans. Parallel Distrib. Systems 14 (1) (2003) 51–62.

Ngo-Tai Fong obtained the BSc degree in computational mathematics from Nanjing University in 1985, MSc degree in operations research from Shanghai University in 1988, and Ph.D. degree in systems engineering from the Chinese University of Hong Kong in 1996. After then, he has been a postdoctoral fellow in Chinese University and a research associate in Wayne State University. Since 2000, he has worked in Ford and its one of suppliers, CAEtech Inc., where he is currently a senior software consultant. His research interests are in optimal control, discrete-event manufacturing systems, decision support systems, and operations management and its web-based applications. Dr. Fong received the Second-Class Award in Progress in Science and Technology from Ministry of Machine and Electronics Industry of China in 1991. He was also awarded the Sino-British Fellowship from Chinese University in 1995.

Cheng-Zhong Xu is currently a tenured Associate Professor in the Department of Electrical and Computer Engineering and Director of Cluster and Internet Computing Laboratory of Wayne State University (Detroit, Michigan). He received the BS and MS degrees from Nanjing University in 1986 and 1989, respectively, and the Ph.D. degree from the University of Hong Kong, Hong Kong in 1993, all in Computer Science. He has published more than sixty papers in archival journals and refereed conference proceedings in the areas of distributed and parallel systems, high performance computing, mobile agent technology, and scalable Internet services. He is the co-author of the book "Load Balancing in Parallel Computers: Theory and Practice" (Kluwer Academic, 1997). He has served on technical program committees of numerous international conferences. He was a guest co-editor of a special issue on "Scalable Web Services and Architecture" in Journal of Parallel and Distributed Computing. He is a recipient of Year 2000 "Faculty Research Award", Year 2002 "President's Awards for Excellence in Teaching" and Year 2003 "Career Development Chair Award" of Wayne State University. He is a senior member of IEEE and a member of ACM.



Le Yi Wang received the Ph.D. degree in electrical engineering from McGill University, Montreal, Canada, in 1990. Since 1990, he has been with Wayne State University, Detroit, Michigan, where he is currently a Professor in the Department of Electrical and Computer Engineering. His research interests are in the areas of H-infinity optimization, complexity and information, robust control, time-varying systems, system identification, adaptive systems, hybrid and nonlinear systems, information processing and learning, as well as automotive, computer and medical applications of control methodologies.

Dr. Wang was awarded the Research Initiation Award in 1992 from the National Science Foundation. He also received the Faculty Research Award from Wayne State University, in 1992, and the College Outstanding Teaching Award from the College of Engineering, Wayne State University, in 1995. He was a keynote speaker in two international conferences. He serves on the IFAC Technical Committee on Modeling, Identification and Signal Processing. He was an Associate Editor of the IEEE Transactions on Automatic Control, and currently is an Editor of the Journal of System Sciences and Complexity and an Associate Editor of International Journal of Control and Intelligent Systems.