ICOT Technical Report: TR-0812

TR-0812

Isoefficiency Analysis of Processing Randomly Scattered Data in Parallel

by
K. Kimura & Ichiyoshi (MRI)

October, 1992

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191 ~ 5

Institute for New Generation Computer Technology

Iso-efficiency Analysis of Load Balancing with Randomly Scattered Data

KOUICHI KIMURA AND NOBUYUKI ICHIYOSHI

Institute for New Generation Computer Technology

1-4-28 Mita, Minato-ku, Tokyo 108, Japan

Abstract

Scalability is an essential notion in predicting the performance of large-scale parallel computation. Iso-efficiency analysis examines the scaling rule between the number of processors and the problem size that can guarantee the efficient use of processors. This paper presents an iso-efficiency analysis of load balancing strategies for problems that consist of many subproblems of unpredictable, different sizes. We assume that input data are randomly scattered over the subproblems according to a Poisson distribution; and that each subproblem is independently solved by the same polynomial-time algorithm of degree $d \geq 1$. The performance of a static load balancing strategy (assigning an equal number of subproblems to each processor all at once) is studied. The efficiency (the ratio of the average to the maximum load at each processor) is shown to converge to one for an increasing number of processors p, provided that there are $\omega(\log^d p)$ subproblems of average size $\Theta(1)$ at each processor. Hence, this load balancing strategy can be considered fairly scalable. A new comparison between polynomials of Poisson random variables in terms of convex ordering is developed and utilized in the proof.

1 Introduction

Recently, there has been growing interest in utilizing massively parallel computers for time-consuming tasks. In order to efficiently use parallel computers, we must confront matters that do not appear in sequential computation. Among others, *load balancing* is very important to all kinds of parallel machines, and deserves to be studied in a general formulation because of

its rather machine-independent nature[1].

Scalability is an essential notion in predicting the performance of large-scale parallel computation. *Iso-efficiency analysis* examines the scaling rule between the number of processors and the problem size that can guarantee the efficient use of processors[2] [9, 10, 11]. We define the efficiency, η, as the ratio of the average to the maximum load at each processor. Clearly, $0 \leq \eta \leq 1$ holds, and $\eta = 1$ if and only if a perfect balance of load is attained. A load balancing strategy is considered *scalable* if the problem size that can guarantee a constant efficiency grows slowly (polynomially) with an increasing number of processors, since this condition indicates that one can utilize a large number of processors efficiently for moderately large problems. In addition, the growth rate succinctly captures the characteristics of scalability.

In this paper, we present an iso-efficiency analysis of load balancing strategies for problems that consist of many subproblems of unpredictable, different sizes. We assume that input data are randomly scattered over the subproblems according to a Poisson distribution; and that each subproblem is independently solved by the same polynomial-time algorithm. For example, a problem concerning large geometric data (points, segments, etc.) in a planar region may consist of subproblems concerning those in the partitioned subregions. A problem of manipulating data in a hash table may also consist of subproblems of manipulating data in each slot [5].

One of the most naïve approaches to solving this kind of problem in parallel is to solve each subproblem at a different processor. This is referred to as the *single assignment strategy*. However, such an approach is likely to incur large load imbalances due to non-uniform subproblem sizes. For example, when each subproblem is solved by a linear-time algorithm, the load imbalance will occur exactly in proportion to the variation in subproblem size. Moreover, when each subproblem is solved by a polynomial-time algorithm of degree $d > 1$, unevenness in the subproblem size is translated into larger unevenness (according to d) in the load between processors, as we will see in Section 3.

A simple remedy for such a load imbalance is to assign a sufficient number of subproblems to each processor. This is referred to as the *multiple assignment strategy*. Intuitively speaking,

[1] In contrast, the overheads associated with interprocessor communications, which are another important concern in parallel processing, are likely to be affected by the characteristics of an individual machine such as the topology of interconnections and others [1].

[2] In general, the speedup attained by a parallel machine does not continue to increase linearly with an increasing number of processors but tends to saturate because there is only a limited degree of concurrency inherent in a problem. However, in many cases, one can obtain an almost linear speedup by solving increasingly large problems [9, 10, 11].

if many subproblems are assigned to each processor, they will consist of small and large subproblems and will, hence, yield a comparable load between processors on average; thus a better result may be expected. In Section 5, we investigate this approach and show that one can almost balance the load by assigning only a modest number of subproblems to each processor. Namely, $\omega(\log^d p)$ subproblems per processor is shown to be sufficient for p processors, provided that each subproblem has size $\Theta(1)$ on average and is solved by a polynomial-time algorithm of degree d.

A similar concept lies behind the principle of *scattered decomposition*, which has been successfully applied to balance the load in irregular matrix problems [2]. It divides a matrix regularly into many fine-grained pieces and distributes them to the processors cyclically.

These load balancing strategies are *static* in the sense that the assignment of subproblems to processors is determined prior to execution. Much work has also been done on *dynamic* load balancing strategies, where the assignment of subproblems to processors is adaptively determined at runtime [2, 3]. In general, the latter strategies outperform the former at the expense of extra overheads, and hence reveal different scalability characteristics [6, 8, 11].

Kruskal and Weiss [8] studied the performance of several dynamic strategies for allocating independent subtasks *on demand* to parallel processors. One of their remarkable conclusions was that allocating an equal number of subtasks to each processor all at once (hence, *not* on demand) provides almost optimal performance, on the assumption that the computation time of each subtask is an independent identically distributed (i.i.d.) random variable with increasing failure rate (IFR) [4]. In particular, they showed that the efficiency approaches one when there are $\omega(p \log p)$ subtasks for p processors and $p \to \infty$. Note that this strategy is the same as our multiple assignment strategy. We will extend their results to non-IFR cases. Briefly, although a polynomial time-complexity function translates the unevenness in the input data size into larger unevenness in its computation time, it does not affect the scalability seriously. (Note that a Poisson random variable has IFR, while its polynomial of degree $d > 1$ does not.)

Vitter and Flajolet [13] gave asymptotic estimates of the expected maximum bucket occupancy in hash tables based on the results by Kolchin et al. [7]. These estimates can be regarded as predicting the performance of load balancing with the single assignment strategy when each subproblem is solved by a linear-time algorithm. We will give a straightforward extension of their results to general cases where a polynomial-time algorithm of degree $d > 1$ is assumed.

Results on the performance of the multiple assignment strategy are derived from the results on the single assignment strategy. We can relate them by virtue of the notion of convex

ordering for random variables [12]. The essence of the relationship is expressed as follows. Let $X_1, X_2, \ldots, X_{r^d}$ be i.i.d. Poisson random variables with parameter $\alpha > 0$ and $\tilde{X}$ be a Poisson random variable with parameter $r\alpha$, where r and d are positive integers. Then,

$$(1.1) \qquad \sum_{1 \le i \le r^d} X_i(X_i - 1)(X_i - 2)\cdots(X_i - d + 1) \prec \tilde{X}(\tilde{X} - 1)(\tilde{X} - 2)\cdots(\tilde{X} - d + 1)$$

where $\prec$ denotes convex ordering. This is established in Section 5.

The rest of the paper is organized as follows. In Section 2, we formulate the load balancing issue and define the efficiency. Section 3 is concerned with the single assignment strategy. Section 4 provides some mathematical prerequisites for the following analysis. Section 5 is concerned with the multiple assignment strategy. Finally, Section 6 summarizes the results.

2 Load Balancing Model

In this section, we introduce a formal model that captures the performance of load balancing with the single and multiple assignment strategies.

Let p be the number of processors and suppose that a problem consisting of $m = pq$ subproblems is given, where $q = 1$ for the single assignment strategy and $q > 1$ for the multiple assignment strategy. Let $X_1, X_2, \ldots, X_m$ be the input data sizes of the subproblems, and assume that they are independent identically distributed (i.i.d.) Poisson random variables with parameter $\alpha > 0$. This implies that input data are randomly scattered over the subproblems. We also assume that each subproblem is independently solved by the same polynomial-time algorithm of degree $d \ge 1$. Let the time-complexity function be $c(X) = a_d X^d + a_{d-1} X^{d-1} + \cdots + a_1 X + a_0$ with $a_d > 0$.

Note that the average subproblem size is α and that the expected computation time for solving a subproblem is given by $c^*(\alpha)$, where $c^*(\cdot)$ is a polynomial defined by

$$(2.1) \qquad c(X) = a_d^* X^{(d)} + \cdots + a_1^* X^{(1)} + a_0^*, \qquad c^*(X) = a_d^* X^d + \cdots + a_1^* X + a_0^*$$

where $X^{(k)}$ denotes $X(X-1)\cdots(X-k+1)$.

Under the single assignment strategy ($q = 1$) or the multiple assignment strategy ($q > 1$), the load at the i-th processor is given by: $L_i = \sum_{(i-1)q < j \le iq} c(X_j)$ for $i = 1, 2, \ldots, p$. Hence, the average load is given by $(L_1 + \cdots + L_p)/p$ and the maximum load is given by $\max_{1 \le i \le p} L_i$. We

define the *efficiency*, η, by the ratio of their expectations:

$$\eta = \frac{E(L_1 + \cdots + L_p)}{p \cdot E\left(\max_{1 \le i \le p} L_i\right)} = \frac{q \cdot c^*(\alpha)}{E\left(\max_{1 \le i \le p} \sum_{j=1}^{q} c(X_{(i-1)q+j})\right)}. \tag{2.2}$$

Clearly, $0 < \eta \le 1$ holds, and $\eta = 1$ if and only if $L_1 = \cdots = L_p$, which means a perfect balance of load.

We will study the asymptotic behavior of η for an increasing number of processors p either with $q = 1$ and an increasing α at various rates (in the single assignment strategy in Section 3) or with a fixed α and an increasing q at various rates (in the multiple assignment strategy in Section 5).

Example 2.1 A *distributed hash table* is a parallel implementation of a hash table [5]. A hash table of size $m = pq$ is divided into p subtables of equal size q. These subtables are then allocated to p processors. Let h be a hash function taking values in $\{1, 2, \ldots, m\}$. An item x with hash value $h(x)$ is stored in the j-th bucket in the subtable at the i-th processor, where $i = \lfloor (h(x) + p - 1)/p \rfloor$ and $j = h(x) - (i-1)p$. Items in the same bucket are linked together into a chain. Now, consider inserting a batch of items in parallel into an initially empty distributed hash table, where a redundancy check must be performed against each item already inserted in the same bucket before each insertion. This checks whether the items coincide with each other or can be unified into a single item. Let X_{ij} be the final number of items in the j-th bucket inside the i-th subtable for $1 \le i \le p$ and $1 \le j \le q$. On the assumption that the hash function is ideally random, X_{ij}s are i.i.d. Poisson random variables with parameter $\alpha > 0$, where α denotes the average number of items per bucket. Let us define the computation time in terms of the number of redundancy checks. Assuming that all redundancy checks fail (i.e., there is no redundancy), we have the time-complexity function: $c(X) = X(X-1)/2 = X^{(2)}/2$. The parameter α is called the *load factor* and indicates the expected number of probes per unsuccessful search (or twice the number of probes per successful search) [13]; hence α should be kept small by taking a sufficiently large q. In particular, it is desirable to have $\alpha = \Theta(1)$ as $p \to \infty$.

3 Single Assignment Strategy

In this section, we study the balance of load between processors when a single subproblem is assigned to each processor. The input data sizes, $X_1, X_2, \ldots, X_p$, are assumed to be i.i.d.

Poisson random variables with parameter $\alpha > 0$. The implication of different assumptions on the growth rate of α for an increasing number of processors p is examined. The load at the i-th processor is given by $c(X_i)$, where $c(\cdot)$ is the time-complexity function of the algorithm being used. The next theorem estimates the expectation of the parallel computation time, $\max_{1\le i\le p} c(X_i)$. This is a straightforward extension of the result by Vitter and Flajolet [13] on the average maximum bucket occupancy in hash tables. Most of the essential points in this section can be found in [7, Chapter II, §6] by Kolchin *et al.*, where the asymptotic behavior of the distribution of the maximum bucket occupancy is studied.

Theorem 3.1 *Let $c : \boldsymbol{R}_+ \longrightarrow \boldsymbol{R}_+$ be a polynomial function of degree $d \ge 1$, and $X_1, \ldots, X_p$ be i.i.d. according to the Poisson distribution with parameter $\alpha > 0$, where $\boldsymbol{R}_+$ denotes the set of all non-negative real numbers. Then we have, as $p \to \infty$,*

$$E\{\max_{1\le i\le p} c(X_i)\} \sim \begin{cases} c(\log p/\log\log p) & \text{if } \alpha = \Theta(1), \\ c(b) \text{ with } b = \omega(\alpha) & \text{if } \alpha = o(\log p), \\ c(\alpha/\gamma) & \text{if } \alpha \sim x\log p \text{ for some } x > 0, \\ c(\alpha) & \text{if } \alpha = \omega(\log p), \end{cases} \tag{3.1}$$

where $\sim$ denotes asymptotic equivalence and $b = b(\alpha, p)$ is an integer larger than α and defined by

$$\frac{e^{-\alpha}\alpha^{b+1}}{(b+1)!} \le \frac{1}{p} < \frac{e^{-\alpha}\alpha^{b}}{b!}, \tag{3.2}$$

and $\gamma = \gamma(x)$ is the root of the equation

$$\gamma + x(\log\gamma - \gamma + 1) = 0 \tag{3.3}$$

in the interval $0 < \gamma < 1$. Moreover, when $\alpha = \omega(\log p)$,

$$\max_{1\le i\le p} c(X_i) - c(\alpha) \sim c'(\alpha)\sqrt{2\alpha\log p} \quad \textit{(in probability)}.$$

PROOF: For brevity, we write $c_n = c(n)$ and use the following notations within this proof.

$$c_n(\alpha) = \sum_{j=0}^{n} \frac{\alpha^j}{j!} \ : \text{ truncated exponential function up to the } n\text{-th term } \ (n = 0, 1, 2, \ldots),$$

$$\rho_n = \rho(\alpha, p, n) = P(\max_{1\le i\le p} X_i \le n) = \{e^{-\alpha}e_n(\alpha)\}^p \quad (n = 0, 1, 2, \ldots), \qquad \rho_{-1} = 0.$$

Then, we get an expression:

$$E\{\max_{1\le i\le p} c(X_i)\} = \sum_{n=0}^{\infty} c_n(\rho_n - \rho_{n-1}) = c_0 + \sum_{n=0}^{\infty}(c_{n+1} - c_n)(1 - \rho_n). \tag{3.4}$$

First we shall consider the case with $\alpha/\log p \to x$ for some $x \geq 0$, which includes the first three cases in (3.1). For a sufficiently large p, an integer $b = b(\alpha, p)$ satisfying $b > \alpha$ and condition (3.2) exists, since $e^{-\alpha}\alpha^n/n!$ is decreasing in $n \geq \alpha$ and $e^{-\alpha}\alpha^\alpha/\Gamma(\alpha+1) \sim 1/\sqrt{2\pi\alpha} = \omega(p^{-1})$. Note that $b \to \infty$ holds as $p \to \infty$. Applying Stirling's formula to $1 < pe^{-\alpha}\alpha^b/b! \leq (b+1)/\alpha$, taking the logarithm, and multiplying the result by $\alpha/(b\log p)$, we find that

$$0 \leq r + \frac{\alpha}{\log p}\left(\log r - r - \frac{\log\sqrt{2\pi b}}{b} + 1\right) \lesssim \left(\frac{\log(b+1)}{b} - \frac{\log\alpha}{b}\right)\frac{\alpha}{\log p},$$

where $r = \alpha/b$ and $\lesssim$ indicates that its left side is asymptotically not larger than its right side. From this estimate it follows that $r \to 0$ when $x = 0$ and that $r \to \gamma$ when $x > 0$, where γ is the root of the equation (3.3) in the interval $0 < \gamma < 1$.

Fixing an integer $k > 0$, we rewrite (3.4) into

$$E\{\max_{1\leq i\leq p} c(X_i)\} = A_1 + A_2 + A_3, \qquad A_1 = \sum_{n=0}^{b-k-1} c_n(\rho_n - \rho_{n-1}),$$

$$A_2 = \sum_{n=b-k}^{b+k-1} c_n(\rho_n - \rho_{n-1}) + c_{b+k}(1 - \rho_{b+k-1}), \qquad A_3 = \sum_{n\geq b+k}(c_{n+1} - c_n)(1-\rho_n),$$

and estimate each term.

Since

$$\rho_{b+k} = \left(1 - e^{-\alpha}\sum_{n>b+k}\frac{\alpha^n}{n!}\right)^p \geq \left(1 - \frac{e^{-\alpha}\alpha^{b+1}}{(b+1)!}\sum_{n=k}^{\infty} r^n\right)^p \geq \left(1 - \frac{r^k}{p(1-r)}\right)^p \geq 1 - \frac{r^k}{1-r},$$

we have

$$A_3 \leq \sum_{j=0}^{\infty} c_{b+k+j+1}\cdot\frac{r^{k+j}}{1-r}.$$

Hence, when $x = 0$, we obtain $A_3 = o(c_b)$ by taking $k = 1$. When $x > 0$, it follows that $A_3 \lesssim C\gamma^k c_b/(1-\gamma)^{d+2}$, where C is a constant depending only on d, the degree of polynomial $c(\cdot)$. Therefore, for arbitrary $\varepsilon > 0$, we have $A_3 \lesssim \varepsilon c_b$ as $p \to \infty$ by taking sufficiently large k.

We also have

$$\rho_{b-k-1} \leq \left(1 - e^{-\alpha}\frac{\alpha^{b-k}}{(b-k)!}\right)^p \leq \exp\left\{-\frac{pe^{-\alpha}}{b!}\cdot\frac{b(b-1)\cdots(b-k+1)}{\alpha^k}\right\}.$$

Hence, when $x = 0$, we obtain $\rho_{b-2} \leq e^{-1/r} \to 0$ by taking $k = 1$. When $x > 0$, we obtain $\rho_{b-k-1} \lesssim \exp(-1/\gamma^k) < \varepsilon$ by taking sufficiently large k for arbitrarily chosen $\varepsilon > 0$. Since $A_1 \leq \rho_{b-k-1}c_{b-k-1}$ and $A_2 \sim (1-\rho_{b-k-1})c_b$, it follows that $A_1 = o(c_b)$ and $A_2 \sim c_b$ when $x = 0$; and that $A_1 \lesssim \varepsilon c_b$ and $|c_b - A_2| \lesssim \varepsilon c_b$ when $x > 0$.

Thus, we obtain $E\{\max_{1\le i\le p} c(X_i)\} \sim c(b)$ when $\alpha/\log p \to x \ge 0$. As noted above, $b \sim \alpha/\gamma$ when $x > 0$. Using Stirling's formula, it is also easily verified that $b(\alpha, p) \sim \Gamma^{-1}(p) \sim \log p/\log\log p$ for $\alpha = \Theta(1)$, where Γ^{-1} denotes the inverse of Gamma function. These complete the proof for the first three cases in (3.1).

Next, let us consider the last case with $\alpha = \omega(\log p)$. By integration by parts, we have

$$1 - e^{-\alpha} e_n(\alpha) = \frac{e^{-\alpha}\alpha^{n+1}}{n!}\int_0^1 (1-\theta)^n e^{\alpha\theta} d\theta. \tag{3.5}$$

For $n > \alpha$, the integral on the right side is less than $1/(n-\alpha)$, since $(1-\theta)^n \le e^{-n\theta}$ for $0 \le \theta < 1$. Hence, we have

$$1 - \rho_n = 1 - \{e^{-\alpha} e_n(\alpha)\}^p \le p \cdot \{1 - e^{-\alpha} e_n(\alpha)\} < \frac{p e^{-\alpha}\alpha^{n+1}}{n!\cdot(n-\alpha)}.$$

In particular, for $n = \lfloor 2\alpha \rfloor$,

$$1 - \rho_{\lfloor 2\alpha\rfloor} \lesssim \frac{p e^{-\alpha}\alpha^{2\alpha+1}}{\alpha\Gamma(2\alpha)} \sim p\left(\frac{e}{4}\right)^{\alpha}\sqrt{\frac{\alpha}{\pi}},$$

and $1 - \rho_{\lfloor 2\alpha\rfloor + k} \lesssim p(e/4)^{\alpha}\sqrt{\alpha/\pi}\cdot 2^{-k}$ for $k \in \boldsymbol{N}$. Hence, we obtain

$$\sum_{n > \lfloor 2\alpha\rfloor} (c_{n+1} - c_n)(1-\rho_n) \lesssim p\left(\frac{e}{4}\right)^{\alpha}\sqrt{\frac{\alpha}{\pi}}\sum_{k=1}^{\infty} c_{\lfloor 2\alpha\rfloor + k + 1} 2^{-k} \to 0 \quad \text{as } p \to \infty. \tag{3.6}$$

Now, let κ be a positive constant, $\nu = \lfloor \alpha + \sqrt{2\alpha\kappa\log p}\rfloor$, and consider ρ_ν.

Lemma 3.1 *For any fixed $\kappa > 0$, we have $\log\log(1/\rho_\nu) = \{1 - \kappa + o(1)\}\log p$ as $p \to \infty$. In particular, $\rho_\nu \to 0$ for $0 < \kappa < 1$, and $\rho_\nu \to 1$ for $\kappa > 1$.*

PROOF: By Equation (3.5), we have

$$\log \rho_\nu = p \log\left\{1 - \frac{e^{-\alpha}\alpha^{\nu+1}}{\nu!}\int_0^1 (1-\theta)^\nu e^{\alpha\theta} d\theta\right\}.$$

For $\gamma = \alpha^{-1/2}(\log p)^{-1/3}$, since $(\nu - \alpha)\gamma \to \infty$ and $\nu\gamma^2 \to 0$ as $p \to \infty$, we have

$$\int_0^\gamma (1-\theta)^\nu e^{\alpha\theta} d\theta \sim \int_0^\gamma e^{-(\nu-\alpha)\theta} d\theta \sim \frac{1}{\nu - \alpha} \sim \frac{1}{\sqrt{2\alpha\kappa\log p}},$$

$$\int_\gamma^1 (1-\theta)^\nu e^{\alpha\theta} d\theta \le \int_\gamma^1 e^{-(\nu-\alpha)\theta} d\theta = \frac{e^{-(\nu-\alpha)\gamma} - e^{-(\nu-\alpha)}}{\nu - \alpha} = o\left(\frac{1}{\nu-\alpha}\right).$$

Hence

$$\frac{e^{-\alpha}\alpha^{\nu+1}}{\nu!}\int_0^1 (1-\theta)^\nu e^{\alpha\theta} d\theta \sim \frac{e^{\nu-\alpha}}{2\sqrt{\pi\kappa\log p}}\left(\frac{\alpha}{\nu}\right)^\nu,$$

and the logarithm of its dominant factor is

$$\nu - \alpha + \nu\log\frac{\alpha}{\nu} = -\frac{(\nu-\alpha)^2}{2\alpha} + O\left(\frac{(\nu-\alpha)^3}{\alpha^2}\right) = -\kappa\log p + O\left(\sqrt{\frac{\log p}{\alpha}}\right)\cdot\log p.$$

From these we obtain the claim. □

PROOF OF THEOREM 3.1 (CONTINUED): Taking $\kappa = 2$ and $\nu = \lfloor \alpha + 2\sqrt{\alpha \log p} \rfloor$, we have $\rho_\nu \to 1$ and $\rho_{\lfloor \alpha \rfloor} \to 0$ as $p \to \infty$ due to the above Lemma. Hence,

$$\sum_{n=\nu}^{\lfloor 2\alpha \rfloor} (c_{n+1} - c_n)(1 - \rho_n) \le c_{\lfloor 2\alpha \rfloor + 1}(1 - \rho_\nu) = o(c(\alpha)) \quad \text{as } p \to \infty, \tag{3.7}$$

$$\sum_{n=\lfloor \alpha \rfloor}^{\nu-1} (c_{n+1} - c_n)(1 - \rho_n) \le c_\nu - c_{\lfloor \alpha \rfloor} = O((\nu - \alpha)c'(\nu)) = o(c(\alpha)) \quad \text{as } p \to \infty, \tag{3.8}$$

$$\sum_{n=0}^{\lfloor \alpha \rfloor - 1} (c_{n+1} - c_n)(1 - \rho_n) \sim (c_{\lfloor \alpha \rfloor} - c_0)(1 - \rho_{\lfloor \alpha \rfloor}) \sim c(\alpha) \quad \text{as } p \to \infty. \tag{3.9}$$

Therefore, from (3.4), (3.6), (3.7), (3.8) and (3.9), we obtain

$$E\{\max_{1 \le i \le p} c(X_i)\} \sim c(\alpha) \quad \text{as } p \to \infty \text{ with } \alpha = \omega(\log p).$$

Moreover, for arbitrary $0 < \kappa_0 < 1 < \kappa_1$, we have $P(\alpha + \sqrt{2\alpha\kappa_0 \log p} \le \max_{1 \le i \le p} X_i \le \alpha + \sqrt{2\alpha\kappa_1 \log p}) \to 1$ as $p \to \infty$, by Lemma 3.1. Hence,

$$P\left(\sqrt{\kappa_0} < \frac{\max_{1 \le i \le p} c(X_i) - c(\alpha)}{c'(\alpha)\sqrt{2\alpha \log p}} < \sqrt{\kappa_1}\right) \to 1 \quad \text{as } p \to \infty \quad \text{for } 0 < \forall\kappa_0 < 1 < \forall\kappa_1.$$

This implies $\max_{1 \le i \le p} c(X_i) - c(\alpha) \sim c'(\alpha)\sqrt{2\alpha \log p}$ in probability. These complete the proof for the case with $\alpha = \omega(\log p)$. □

As an immediate corollary, we obtain the following result. This tells us, in terms of iso-efficiency, the scalability characteristic of load balancing with the single assignment strategy.

Corollary 3.1 *Under the assumptions of Theorem 3.1, we have, as* $p \to \infty$,

(i) $\eta \sim (\alpha \log\log p / \log p)^d$ *if* $\alpha = \Theta(1)$,

(ii) $\eta \to 0$ *if* $\alpha = o(\log p)$,

(iii) $\eta \to \gamma^d$ *if* $\alpha \sim x \log p$ *for some* $x > 0$, *where* $\gamma = \gamma(x)$ *is the root of* (3.3) *in* $0 < \gamma < 1$,

(iv) $\eta \to 1$ *if* $\alpha = \omega(\log p)$,

where η *is the efficiency defined by* (2.2).

Thus, the efficiency can be maintained in cases (iii) and (iv) for an increasing number of processors. In these cases, the overall input data size (problem size) amounts to $\Omega(p \log^d p)$ for an increasing p. Hence, the single assignment strategy can be considered to be fairly scalable. However, this strategy should not be applied to the distributed hash table (Example 2.1 with $q = 1$), because load factor α should be kept small and, hence, cases (iii) and (iv) are not useful. Its proper implementation is given by the multiple assignment strategy (Example 2.1 with $q \gg 1$), as we will see in Section 5.

4 Convex Ordering and Its Properties

In this section we prepare some mathematical prerequisites for the analysis in the next section. The definition and basic properties of convex ordering for random variables are briefly described. Although the first two propositions are general, the last proposition concerns a rather special situation which we will encounter in the next section. Within this section, $\mathcal{M}$ denotes the class of all non-negative random variables with finite expectations, and P^X denotes the distribution of $X \in \mathcal{M}$. We often write $X \vee Y = \max(X, Y)$.

Definition 4.1 For $X, Y \in \mathcal{M}$, we define $P^X \prec P^Y$ (or simply, $X \prec Y$), if and only if $E(X \vee c) \leq E(Y \vee c)$ for all $c \geq 0$.

The binary relation $\prec$ is called *convex ordering* [12]. Strictly speaking, it is defined on the class of probability distributions rather than on the class of random variables. However, for brevity, we will often write $X \prec Y$.

Convex ordering $\prec$ is weaker than stochastic inequality [12], as is shown in the next proposition. Let μ and ν be probability distributions over the set of non-negative numbers, $\boldsymbol{R}_+$. Then, μ is said to be *stochastically smaller* than ν, denoted $\mu <_{st} \nu$, if and only if their distribution functions, $\phi_\mu(t) = \mu([0,t])$ and $\phi_\nu(t) = \nu([0,t])$, satisfy $\phi_\mu(t) \geq \phi_\nu(t)$ for all $t \geq 0$.

Proposition 4.1 *Let $X, Y, Z, X_n, Y_n \in \mathcal{M}$, where $n = 1, 2, \ldots$.*

(i) *If $X \leq Y$ a.s. (almost surely), then $X \prec Y$.*

(ii) *If $X \prec Y$ and $Y \prec Z$, then $X \prec Z$.*

(iii) *If $X \prec Y$ and $Y \prec X$, then $P^X = P^Y$.*

(iv) *If $X \prec Y$, then $E(X) \leq E(Y)$.*

(v) *If $P^X <_{st} P^Y$, then $X \prec Y$.*

(vi) *Assume that P^{X_n} and P^{Y_n} converge to P^X and P^Y respectively as $n \to \infty$. If $X_n \prec Y_n$ for any n, then $X \prec Y$.*

PROOF: Since (i), (ii), (iv) and (vi) are trivial by definition, we shall only prove (iii) and (v). Let $\phi_X(t) = P(X \leq t)$ and $\phi_Y(t) = P(Y \leq t)$ be the distribution functions of X and Y respectively. And define $\psi_X(t)$ and $\psi_Y(t)$ by

$$\psi_X(t) = \int_t^\infty (x - t) d\phi_X(x), \qquad \psi_Y(t) = \int_t^\infty (y - t) d\phi_Y(y).$$

Then we have

$$\psi_X(+\infty) = 0, \qquad \psi_X'(t) = \phi_X(t) - 1, \qquad E(X \vee t) = t + \psi_X(t) \qquad (\forall t \geq 0),$$
$$\psi_Y(+\infty) = 0, \qquad \psi_Y'(t) = \phi_Y(t) - 1, \qquad E(Y \vee t) = t + \psi_Y(t) \qquad (\forall t \geq 0).$$

(iii) If $X \prec Y$ and $Y \prec X$, then $\psi_X \equiv \psi_Y$, $\phi_X \equiv \phi_Y$ and $P^X = P^Y$ follow.

(v) $P^X <_{st} P^Y$ implies $\phi_X \geq \phi_Y$. Hence $\psi_X \leq \psi_Y$ and $X \prec Y$ follow. □

Proposition 4.2 *Let $X, Y, Z \in \mathcal{M}$, and Z be independent of X and Y.*

(i) *If $X \prec Y$, then $X \vee Z \prec Y \vee Z$.*

(ii) *If $X \prec Y$, then $X + Z \prec Y + Z$.*

(iii) *If $X \prec Y$ and $YZ \in \mathcal{M}$, then $XZ \prec YZ$.*

(iv) *Let $f : \boldsymbol{R}_+^2 \longrightarrow \boldsymbol{R}_+$ be a Lebesgue measurable function and $f(X, z), f(Y, z) \in \mathcal{M}$ for $\forall z \geq 0$. If $f(X, z) \prec f(Y, z)$ for $\forall z \geq 0$, then $f(X, Z) \prec f(Y, Z)$.*

PROOF: Since (i), (ii) and (iii) follow from (iv), we shall only prove (iv). For arbitrary $c \geq 0$, since $f(X, z), f(Y, z) \in \mathcal{M}$ for $\forall z \geq 0$,

$$E\{f(X, Z) \vee c\} = \int_0^\infty E\{f(X, z) \vee c \mid Z = z\} P^Z(dz) = \int_0^\infty E\{f(X, z) \vee c\} P^Z(dz)$$
$$\leq \int_0^\infty E\{f(Y, z) \vee c\} P^Z(dz) = \int_0^\infty E\{f(Y, z) \vee c \mid Z = z\} P^Z(dz) = E\{f(Y, Z) \vee c\}.$$

Thus, we obtain $f(X, Z) \prec f(Y, Z)$. □

By repeatedly applying Propositions 4.1 (ii) and 4.2 (i), we obtain the following corollary. It shows that the convex ordering is useful in comparing the expectation of maxima.

Corollary 4.1 *Let $X_1, \ldots, X_n \in \mathcal{M}$ and $Y_1, \ldots, Y_n \in \mathcal{M}$ be i.i.d. according to μ and ν respectively. If $\mu \prec \nu$, then*

$$E(\max_{1 \leq i \leq n} X_i) \leq E(\max_{1 \leq i \leq n} Y_i).$$

The next proposition extracts the essence of comparison that will be made repeatedly in the next section.

Proposition 4.3 *Let $f : (\boldsymbol{R}_+^n)^r \longrightarrow \boldsymbol{R}_+$ be a measurable function such that*

$$f(\boldsymbol{x}, \ldots, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r) + f(\boldsymbol{y}, \ldots, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r)$$
$$\leq f(\boldsymbol{x}, \ldots, \boldsymbol{x}, \boldsymbol{x}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r) + f(\boldsymbol{y}, \ldots, \boldsymbol{y}, \boldsymbol{y}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r),$$

$$\max\{f(\boldsymbol{x}, \ldots, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r),\ f(\boldsymbol{y}, \ldots, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r)\}$$
$$\leq \max\{f(\boldsymbol{x}, \ldots, \boldsymbol{x}, \boldsymbol{x}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r),\ f(\boldsymbol{y}, \ldots, \boldsymbol{y}, \boldsymbol{y}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_r)\},$$

for $2 \le \forall k \le r$ *and* $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_r \in \boldsymbol{R}_+{}^n$. *And let* $g : (\boldsymbol{R}_+{}^n)^r \longrightarrow \boldsymbol{R}_+$ *be a symmetric measurable function, that is,*

$$g(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r) = g(\boldsymbol{x}_{\sigma_1}, \ldots, \boldsymbol{x}_{\sigma_r}) \quad \text{for } \forall \sigma = (\sigma_1, \ldots, \sigma_r) \in \mathcal{S}_r,\ \forall \boldsymbol{x}_1, \ldots, \boldsymbol{x}_r \in \boldsymbol{R}_+{}^n,$$

where $\mathcal{S}_r$ *denotes the symmetric group of degree* r. *And let* $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_r$ *be i.i.d. random vectors of dimension* n, *and* $E[f(\boldsymbol{X}_1, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_1)], E[g(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_r)] < +\infty$. *Then,*

$$(4.1)\quad f(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_r) + g(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_r) \prec f(\boldsymbol{X}_1, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_1) + g(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_r).$$

PROOF: By induction on k, it is sufficient to show that

$$\begin{aligned} &f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_1, \boldsymbol{X}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r) + g(\boldsymbol{X}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}, \boldsymbol{X}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r) \\ &\prec f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_1, \boldsymbol{X}_1, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r) + g(\boldsymbol{X}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}, \boldsymbol{X}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r) \end{aligned}$$

for each $k = 2, 3, \ldots, r$ and arbitrary $\boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-1}, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_r \in \boldsymbol{R}_+^n$. Since this reduces to (4.1) with $r = 2$, we may assume $r = 2$ from the beginning. Note that the assumption on f implies

$$f(\boldsymbol{x}, \boldsymbol{y}) \vee c + f(\boldsymbol{y}, \boldsymbol{x}) \vee c \le f(\boldsymbol{x}, \boldsymbol{x}) \vee c + f(\boldsymbol{y}, \boldsymbol{y}) \vee c \quad \text{for } \forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{R}_+^n,\ \forall c > 0.$$

Hence, for arbitrary $c \ge 0$,

$$\begin{aligned} & 2E[\{f(\boldsymbol{X}_1, \boldsymbol{X}_2) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c] \\ = \ & E[\{f(\boldsymbol{X}_1, \boldsymbol{X}_2) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c + \{f(\boldsymbol{X}_2, \boldsymbol{X}_1) + g(\boldsymbol{X}_2, \boldsymbol{X}_1)\} \vee c] \\ \le \ & E[\{f(\boldsymbol{X}_1, \boldsymbol{X}_1) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c + \{f(\boldsymbol{X}_2, \boldsymbol{X}_2) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c] \\ = \ & 2E[\{f(\boldsymbol{X}_1, \boldsymbol{X}_1) + g(\boldsymbol{X}_1, \boldsymbol{X}_2)\} \vee c]. \end{aligned}$$

Thus, the claim is established. □

As a special case of this proposition, we obtain the following.

Corollary 4.2 *Let* $X_1, \ldots, X_r \in \mathcal{M}$ *be i.i.d. and assume that* $Y_1, \ldots, Y_r, Z \in \mathcal{M}$ *are independent of* $X_1, \ldots, X_r$. *Then we have*

$$X_1 Y_1 + \cdots + X_r Y_r + Z \prec X_1(Y_1 + \cdots + Y_r) + Z.$$

5 Multiple Assignment Strategy

In this section, we investigate the balance of load between processors when a sufficient number of subproblems are assigned to each processor. We shall derive the results from those obtained in Section 3, where only a single subproblem is assigned to each processor. These two cases can, in fact, be related by virtue of convex ordering. The next theorem is the basis of this approach.

Theorem 5.1 *Let $\{X_i\}_{1\le i\le r^d}$ be i.i.d. Poisson random variables with parameter $\alpha > 0$, and $\tilde{X}$ be a Poisson random variable with parameter $r\alpha$, where $r \in \boldsymbol{N}$. Then we have*

$$\sum_{1\le i\le r^d} X_i^{(d)} \prec \tilde{X}^{(d)}.$$

This theorem is proved by approximating a Poisson variable by a sum of $\{0,1\}$-valued random variables and reducing the claim to its binary counterpart. The first lemma relates a polynomial of a Poisson variable to a polynomial of binary random variables in the limit. The following several lemmas examine the properties of the latter polynomial. Subsequently, the proof of the theorem will be given. Now, define a homogeneous polynomial function of degree d in dn variables, $\Delta^d : (\boldsymbol{R}_+^n)^d \longrightarrow \boldsymbol{R}_+$, by

$$\Delta^d(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_d) = \sum_{1\le j_1<\cdots<j_d\le n} x_{1j_1}\cdots x_{dj_d} \qquad \text{for} \quad \boldsymbol{x}_i = (x_{i1},\ldots,x_{in}) \qquad (1\le i\le d).$$

For brevity, we write $\Delta^d(\boldsymbol{x}) = \Delta^d(\boldsymbol{x},\ldots,\boldsymbol{x})$ and $\Delta^{d_1,d_2}(\boldsymbol{x},\boldsymbol{y}) = \Delta^{d_1+d_2}(\underbrace{\boldsymbol{x},\ldots,\boldsymbol{x}}_{d_1},\underbrace{\boldsymbol{y},\ldots,\boldsymbol{y}}_{d_2})$.

Lemma 5.1 *Let X be a Poisson random variable with parameter $\alpha > 0$. For $\forall n \ge d$, let $X_{n,1},\ldots,X_{n,n}$ be i.i.d. such that $P(X_{n,j}=1) = \alpha/n$ and $P(X_{n,j}=0) = 1-\alpha/n$, and $\boldsymbol{X}_n = (X_{n,1},\ldots,X_{n,n})$. Then*

$$X^{(d)} = d!\lim_{n\to\infty}\Delta^d(\boldsymbol{X}_n) \qquad (\textit{convergence in law}).$$

PROOF: Since $X_{n,i} = 0$ or 1 for each $1\le i\le n$,

$$(X_{n,1}+\cdots+X_{n,n})\sum_{\substack{i_1,\ldots,i_d\le n\\(\text{distinct})}} X_{n,i_1}\cdots X_{n,i_d} = \sum_{\substack{i_1,\ldots,i_{d+1}\le n\\(\text{distinct})}} X_{n,i_1}\cdots X_{n,i_{d+1}} + d\sum_{\substack{i_1,\ldots,i_d\le n\\(\text{distinct})}} X_{n,i_1}\cdots X_{n,i_d},$$

namely, $(X_{n,1}+\cdots+X_{n,n}-d)\cdot d!\cdot\Delta^d(\boldsymbol{X}_n) = (d+1)!\cdot\Delta^{d+1}(\boldsymbol{X}_n)$ holds. Hence, we obtain $(X_{n,1}+\cdots+X_{n,n})^{(d)} = d!\cdot\Delta^d(\boldsymbol{X}_n)$ by induction on d. Here the distribution of $X_{n,1}+\cdots+X_{n,n}$ converges to the Poisson distribution with parameter α due to the classical law of small numbers. Therefore, we obtain the desired result. □

Lemma 5.2 *Let $\{X_{ij}\}_{1\le i\le r,1\le j\le n}$ be $\{0,1\}$-valued i.i.d. and $\boldsymbol{X}_i=(X_{i1},\dots,X_{in})$ for $1\le i\le r$. Then, for $\forall d,s=0,1,2,\dots$, we have*

$$\sum_{i=1}^{r}\Delta^{d+1,s}(\boldsymbol{X}_i,\sum_{h=1}^{r}\boldsymbol{X}_h)\prec\Delta^{d,s+1}(\boldsymbol{X}_1,\sum_{h=1}^{r}\boldsymbol{X}_h). \tag{5.1}$$

PROOF: Since both sides of (5.1) are identical for $d=0$, we may assume that $d\ge 1$. For $k=0,1,\dots,n-1$, we define $\Delta_{k+1}^{d+1}:\boldsymbol{R}_+^n\times\boldsymbol{R}_+^n\times\boldsymbol{R}_+^n\longrightarrow\boldsymbol{R}_+$ by

$$\begin{aligned}\Delta_{k+1}^{d+1,s}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) &= \sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\le k+1}} x_{j_1}\cdots x_{j_{d+1}}z_{j_{d+2}}\cdots z_{j_{d+s+1}}\\ &\quad+\sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\ge k+2}} y_{j_1}\cdots y_{j_d}x_{j_{d+1}}z_{j_{d+2}}\cdots z_{j_{d+s+1}}\\ &\quad\text{for}\quad \boldsymbol{x}=(x_1,\dots,x_n),\quad \boldsymbol{y}=(y_1,\dots,y_n),\quad \boldsymbol{z}=(z_1,\dots,z_n).\end{aligned}$$

Fixing an arbitrary $1\le k<n$, we write $\hat{\boldsymbol{X}}_i=(X_{i1},\dots,X_{ik})$ for $1\le i\le r$. Taking an arbitrary $(x_{ij})_{1\le i\le r,\,k+1\le j\le n}\in\boldsymbol{R}_+^{r(n-k)}$, we define $a(\hat{\boldsymbol{X}}_1,\dots,\hat{\boldsymbol{X}}_r)$ by

$$a(\hat{\boldsymbol{X}}_1,\dots,\hat{\boldsymbol{X}}_r)-\sum_{i=1}^{r}\Delta_{k+1}^{d+1,s}(\bar{\boldsymbol{X}}_i,\bar{\boldsymbol{X}}_1,\sum_{h=1}^{r}\bar{\boldsymbol{X}}_h),$$

where $\bar{\boldsymbol{X}}_i=(X_{i1},\dots,X_{ik},x_{i\,k+1},\dots,x_{in})$ for $1\le\forall i\le r$. Since

$$\begin{aligned}\Delta_{k+1}^{d+1,s}(\bar{\boldsymbol{X}}_i,\bar{\boldsymbol{X}}_1,\sum_{h=1}^{r}\bar{\boldsymbol{X}}_h) &= \sum_{\ell=d+2}^{d+s+2}\sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\le k,\ j_{\ell-1}\le k<j_\ell}} X_{ij_1}\cdots X_{ij_{d+1}}Z_{j_{d+2}}\cdots Z_{j_{\ell-1}}z_{j_\ell}\cdots z_{j_{d+s+1}}\\ &\quad+\sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}=k+1}} X_{ij_1}\cdots X_{ij_d}x_{i\,k+1}z_{j_{d+2}}\cdots z_{j_{d+s+1}}\\ &\quad+\sum_{\ell=1}^{d+1}\sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\ge k+2,\ j_{\ell-1}\le k<j_\ell}} X_{1j_1}\cdots X_{1j_{\ell-1}}x_{1j_\ell}\cdots x_{1j_d}x_{ij_{d+1}}z_{j_{d+2}}\cdots z_{j_{d+s+1}},\\ &\text{where}\quad Z_j\overset{\text{def}}{=}\sum_{h=1}^{r}X_{hj}\ \ (1\le j\le k),\qquad z_j\overset{\text{def}}{=}\sum_{h=1}^{r}x_{hj}\ \ (k+1\le j\le n),\end{aligned}$$

we have $a(\hat{\boldsymbol{X}}_1,\dots,\hat{\boldsymbol{X}}_r)=f(\hat{\boldsymbol{X}}_1,\dots,\hat{\boldsymbol{X}}_r)+g(\hat{\boldsymbol{X}}_1,\dots,\hat{\boldsymbol{X}}_r)$, where

$$\begin{aligned}f(\hat{\boldsymbol{X}}_1,\dots,\hat{\boldsymbol{X}}_r) &= \sum_{i=1}^{r}\left(u_i\sum_{1\le j_1<\cdots<j_d\le k}X_{ij_1}\cdots X_{ij_d}\right)+\sum_{\ell=1}^{d+1}\left(v_\ell\sum_{1\le j_1<\cdots<j_{\ell-1}\le k}X_{1j_1}\cdots X_{1j_{\ell-1}}\right),\\ g(\hat{\boldsymbol{X}}_1,\dots,\hat{\boldsymbol{X}}_r) &= \sum_{\ell=d+2}^{d+s+2}\sum_{\substack{1\le j_1<\cdots<j_{d+s+1}\le n,\\ j_{d+1}\le k,\ j_{\ell-1}\le k<j_\ell}}\left\{\left(\sum_{i=1}^{r}X_{ij_1}\cdots X_{ij_{d+1}}\right)Z_{j_{d+2}}\cdots Z_{j_{\ell-1}}z_{j_\ell}\cdots z_{j_{d+s+1}}\right\},\end{aligned}$$

and u_i and v_ℓ are non-negative constants determined by k and $\{x_{pq}\}_{1<p<r,\,k+1\le q\le n}$ for each $1\le i\le r$ and $1\le\ell\le d+1$. Note that

$$\sum_{1\le j_1<\cdots<j_\ell\le k}X_{ij_1}\cdots X_{ij_\ell}=\binom{X_{i1}+\cdots+X_{ik}}{\ell}\qquad(1\le\forall i\le r,\ 0\le\forall\ell\le d+1),$$

since we have $X_{ij} = 0$ or 1 by assumption. Thus, we can apply Proposition 4.3 to obtain $a(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, \ldots, \hat{\boldsymbol{X}}_r) \prec f(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_1, \ldots, \hat{\boldsymbol{X}}_1) + g(\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, \ldots, \hat{\boldsymbol{X}}_r)$. Hence, by Proposition 4.2 (iv), we obtain

$$\sum_{i=1}^{r} \Delta_{k+1}^{d+1,s}(\boldsymbol{X}_i, \boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h) \prec \sum_{i=1}^{r} \Delta_k^{d+1,s}(\boldsymbol{X}_i, \boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h).$$

Since this holds for arbitrary $1 \leq k < n$, we obtain

$$\sum_{i=1}^{r} \Delta_n^{d+1,s}(\boldsymbol{X}_i, \boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h) \prec \sum_{i=1}^{r} \Delta_1^{d+1,s}(\boldsymbol{X}_i, \boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h).$$

This is the desired result. □

Lemma 5.3 *Let $\{X_{ij}\}_{1 \leq i \leq r^2, 1 \leq j \leq n}$ be $\{0,1\}$-valued i.i.d. and $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{in})$ for $1 \leq i \leq r^2$. Then, for $\forall d, s = 0, 1, 2, \ldots$, we have*

$$\sum_{i=1}^{r} \Delta^{d,s}(\boldsymbol{X}_i, \sum_{h=1}^{r} \boldsymbol{X}_{i+(h-1)r}) \prec \sum_{i=1}^{r} \Delta^{d,s}(\boldsymbol{X}_i, \sum_{h=1}^{r} \boldsymbol{X}_h).$$

PROOF: This claim is immediately established by repeatedly applying Corollary 4.2. Namely, in dominating the left side, we replace $X_{i+(h-1)r\,j}$ by $X_{h'j}$ for each $(i,h,j) \in \{1,\ldots,r\} \times \{1,\ldots,r\} \times \{1,\ldots,n\}$, where $1 \leq h' \leq r$ and $h' \equiv i + h - 1 \bmod r$. □

Lemma 5.4 *Let $\{X_{ij}\}_{1 \leq i \leq r^d, 1 \leq j \leq n}$ be $\{0,1\}$-valued i.i.d. and $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{in})$ for $1 \leq i \leq r^d$. Then we have*

$$\sum_{1 \leq i \leq r^d} \Delta^d(\boldsymbol{X}_i) \prec \Delta^d(\boldsymbol{X}_1 + \cdots + \boldsymbol{X}_r).$$

PROOF: By Lemma 5.2, we have

$$\sum_{i=1}^{r} \Delta^d(\boldsymbol{X}_i) \prec \Delta^{d-1,1}(\boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h).$$

Consider the sum of r such expressions as that on the right side with distinct indices. Successively applying Proposition 4.2 (i), Lemma 5.3 and Lemma 5.2 to the sum, we obtain

$$\sum_{i=1}^{r^2} \Delta^d(\boldsymbol{X}_i) \prec \sum_{i=1}^{r} \Delta^{d-1,1}(\boldsymbol{X}_i, \sum_{h=1}^{r} \boldsymbol{X}_{i+(h-1)r}) \prec \sum_{i=1}^{r} \Delta^{d-1,1}(\boldsymbol{X}_i, \sum_{h=1}^{r} \boldsymbol{X}_h) \prec \Delta^{d-2,2}(\boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h).$$

Applying again Proposition 4.2 (i), Lemma 5.3 and Lemma 5.2 to the sum of r such expressions as that on the rightmost side with distinct indices, and so on. We finally obtain

$$\sum_{i=1}^{r^d} \Delta^d(\boldsymbol{X}_i) \prec \Delta^{0,d}(\boldsymbol{X}_1, \sum_{h=1}^{r} \boldsymbol{X}_h) = \Delta^d(\boldsymbol{X}_1 + \cdots + \boldsymbol{X}_r).$$

This completes the proof. □

PROOF OF THEOREM 5.1: Let n be an arbitrary integer greater than d, and $(X_{inj})_{1\le i\le r^d,\,1\le j\le n}$ be i.i.d. random variables such that $P(X_{inj}=1)=\alpha/n$ and $P(X_{inj}=0)=1-\alpha/n$. We write $\boldsymbol{X}_{in}=(X_{in1},\ldots,X_{inn})$ for $1\le \forall i\le r^d$. By Lemma 5.4, we have $\sum_{1\le i\le r^d}\Delta^d(\boldsymbol{X}_{in}) \prec \Delta^d(\boldsymbol{X}_{1n}+\cdots+\boldsymbol{X}_{rn})$. Letting $n\longrightarrow\infty$, we obtain the desired result due to Lemma 5.1 and Proposition 4.1 (vi). □

Theorem 5.2 *Let $\{X_{ij}\}_{1\le i\le p,1\le j\le q}$ be i.i.d. Poisson random variables with parameter $\alpha>0$. For any fixed $\alpha>0$, according to different assumptions on the growth rate of $q=q(p)$ as $p\longrightarrow\infty$, we have:*

$$E\left(\max_{1\le i\le p}\sum_{1\le j\le q}X_{ij}^{(d)}\right) \lesssim \begin{cases} (\log p/\log\log p)^d & \text{if } q=\Theta(1),\\ b^d \text{ with } b=\omega(q^{1/d}) & \text{if } q=o(\log^d p),\\ \alpha^d q/\gamma^d & \text{if } \alpha^d q\sim x^d\log^d p \text{ for some } x>0, \end{cases}$$

$$E\left(\max_{1\le i\le p}\sum_{1\le j\le q}X_{ij}^{(d)}\right) \sim \alpha^d q \qquad \text{if } q=\omega(\log^d p),$$

where $b=b(\alpha q^{1/d},p)$ is an integer larger than $\alpha q^{1/d}$ and defined by

$$\frac{e^{-\alpha q^{1/d}}(\alpha q^{1/d})^{b+1}}{(b+1)!}\le\frac{1}{p}<\frac{e^{-\alpha q^{1/d}}(\alpha q^{1/d})^{b}}{b!},$$

and $\gamma=\gamma(x)$ is the root of the equation (3.3) in the interval $0<\gamma<1$.

PROOF: The estimates from above are obvious by Theorems 5.1 and 3.1 and by Corollary 4.1. In the last case, we also have $E\left(\max_{1\le i\le p}\sum_{1\le j\le q}X_{ij}^{(d)}\right)\ge E\left(\sum_{1\le j\le q}X_{1j}^{(d)}\right)\sim\alpha^d q$. Thus, the asymptotic equivalence holds. □

Now, let us discuss the scalability of load balancing with the multiple assignment strategy. As before, p denotes the number of processors and q denotes the number of subproblems per processor. Let X_{ij} be the size of the j-th subproblem assigned to the i-th processor for $1\le i\le p$ and $1\le j\le q$. These are assumed to be i.i.d. Poisson random variables with parameter α, where α denotes the average subproblem size. The load at the i-th processor is given by $\sum_{1\le j\le q}c(X_{ij})$ for $1\le i\le p$, where $c(\cdot)$ denotes the time-complexity function of the algorithm. The expectation of the maximum load among the processors can be immediately estimated by the above theorem as follows:

Corollary 5.1 *Let $c:\boldsymbol{R}_+\longrightarrow\boldsymbol{R}_+$ be a polynomial function of degree $d\ge 1$, and $\{X_{ij}\}_{1\le i\le p,1\le j\le q}$ be i.i.d. Poisson random variables with parameter $\alpha>0$. For any fixed $\alpha>0$, according to*

different assumptions on the growth rate of $q = q(p)$ *as* $p \to \infty$, *we have:*

$$E\left(\max_{1\le i\le p}\sum_{1\le j\le q} c(X_{ij})\right) \lesssim \begin{cases} c(\log p/\log\log p) & \text{if } q=\Theta(1),\\ c(b) \text{ with } b=\omega(q^{1/d}) & \text{if } q=o(\log^d p),\\ qc(\alpha/\gamma) & \text{if } \alpha^d q\sim x^d\log^d p \text{ for some } x>0,\end{cases}$$

$$E\left(\max_{1\le i\le p}\sum_{1\le j\le q} c(X_{ij})\right) \sim qc^*(\alpha) \qquad \text{if } q=\omega(\log^d p),$$

where $b = b(\alpha q^{1/d}, p)$ *and* $\gamma = \gamma(x)$ *are the same as in the preceding theorem and* $c^*(\cdot)$ *is a polynomial defined by* (2.1).

Therefore, we obtain the following result.

Corollary 5.2 *Under the assumptions of Corollary 5.1, we have, as* $p \to \infty$,
(i) $\eta \to 1$ *if* $\alpha^d q = \omega(\log^d p)$,
(ii) $\eta \gtrsim \gamma^d$ *if* $\alpha^d q \sim x^d \log^d p$ *for some* $x > 0$,
where η *is the efficiency defined by* (2.2), *and* $\gamma = \gamma(x)$ *is the root of the equation* (3.3) *in the interval* $0 < \gamma < 1$.

Thus, a constant efficiency is maintained for an increasing number of processors p, provided that there are $\Omega(p\log^d p)$ subproblems of average size $\Theta(1)$. For example, the distributed hash table in Example 2.1 works efficiently in parallel, provided that there are $\Omega(p\log^2 p)$ items. The size of the subtable at each processor should be $\Omega(\log^2 p)$ in order to keep the load factor α small.

In contrast, for a fixed $\alpha > 0$ and $q = o((\log p/\log\log p)^d)$, we have $\eta \to 0$ as $p \to \infty$, since

$$E\left(\max_{1\le i\le p}\sum_{1\le j\le q} c(X_{ij})\right) \ge E\left(\max_{\substack{1\le i\le p\\ 1\le j\le q}} c(X_{ij})\right) \sim c(\log p/\log\log p) = \omega(q)$$

due to Theorem 3.1. We conjecture that $\eta \to 0$ holds for $q = o(\log^d p)$ for $d > 1$ as well as for $d = 1$ (cf. Corollary 3.1 (ii)).

6 Conclusions

We have presented an iso-efficiency analysis of the load balance between p processors, when an equal number q of subproblems are independently solved at each processor by the same polynomial-time algorithm of degree d, on the assumption that the input data are randomly scattered over the subproblems according to Poisson distribution with parameter $\alpha > 0$. The

efficiency (the ratio of the average to the maximum load at each processor) is shown to be asymptotically at least γ^d, provided that $\alpha^d q \sim x^d \log^d p$ holds for a fixed $x > 0$ as $p \to \infty$. Here, $\gamma = \gamma(x)$ denotes the root of the equation $\gamma + x(\log \gamma - \gamma + 1) = 0$ in the interval $0 < \gamma < 1$. And one can take γ arbitrarily close to one by choosing a sufficiently large $x > 0$. Thus, this simple load balancing strategy is shown to be fairly scalable.

References

[1] W. Dally, "Network and Processor Architecture for Message-Driven Computers," in: R. Suaya and G. Birtwictle, ed., *VLSI and Parallel Computation*, Morgan Kaufmann, (1990).

[2] G. C. Fox, M. A. Johnson, G. A. Lyzenga, S. W. Otto, J. K. Salmon and D. W. Walker, *Solving Problems on Concurrent Processors*, vol.I, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[3] M. Furuichi, K. Taki, and N. Ichiyoshi, "A Multi-Level Load Balancing Scheme for OR-Parallel Exhaustive Search Programs on the Multi-PSI," *Proc. of the 2nd ACM SIGPLAN Symposium on Principles & Practice of Parallel Programming*, pp.50–59 (1990).

[4] D. P. Heyman and M. J. Sobel (eds.), *Handbooks in Operation Research and Management Science Volume 2: Stochastic Models*, North-Holland, (1990).

[5] N. Ichiyoshi and K. Kimura, *Asymptotic Load Balance of Distributed Hash Tables*, Proc. of the International Conference on Fifth Generation Computer Systems, Tokyo, Japan (1992), pp. 869–876.

[6] K. Kimura and N. Ichiyoshi, "Probabilistic Analysis of the Optimal Efficiency of the Multi-Level Dynamic Load Balancing Scheme," *Proc. of the 6th Distributed Memory Computing Conference*, (1991).

[7] V. F. Kolchin, B. A. Sevastyanov and V. P. Chistyakov, *Random Allocations*, Winston & Sons, Washington, 1978.

[8] C. P. Kruskal and A. Weiss, *Allocating Independent Subtasks on Parallel Processors*, IEEE Trans. on Software Engineering, vol. SE-11, no. 10 (1985), pp. 1001–1016.

[9] V. Kumar and A. Gupta, *Analysis of Scalability of Parallel Algorithms and Architectures: A Survey*, Proc. of International Conference on Supercomputing, 1991.

[10] V. Kumar and V. N. Rao, "Parallel depth-First Search, part II: Analysis," *International Journal of Parallel Programming*, vol. 16, no. 6 (1987), pp. 501–519.

[11] V. Kumar and V. N. Rao, "Load Balancing on the Hypercube Architecture," *Proc. of the 1989 Conference on Hypercubes, Concurrent Computers and Applications*, pp.603–608 (1989).

[12] D. Stoyan, *Comparison Methods for Queues and other Stochastic Models*, John Wiley & Sons, Chichester, 1983.

[13] J. S. Vitter and P. Flajolet, *Average-Case Analysis of Algorithms and Data Structures*, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol. A, Algorithms and Complexity*, North-Holland, Amsterdam (1990), pp. 431–524.