Isoefficiency Analysis of Hierarchical On-Demand Load Distribution*

KOUICHI KIMURA
kokimura@icot.or.jp

NOBUYUKI ICHIYOSHI
ichiyoshi@icot.or.jp

Institute for New Generation Computer Technology
1-4-28 Mita, Minato-ku, Tokyo 108, Japan

Abstract

This paper provides a probabilistic analysis of the scalability of load balancing techniques with on-demand distribution. A problem to be solved (or a task) is assumed to be composed of many independent subtasks that require different amounts of computation.

In the *single-level dynamic load balancing scheme*, one processor divides a given task into many subtasks, which are distributed to other processors on demand and executed independently. We introduce a formal model of its execution as a queuing system with multiple servers, and estimate the efficiency (speedup divided by the number of processors) taking account of the dividing costs and the load imbalance between processors due to the non-uniformness of subtasks.

These results are then applied to the analysis of scalability of the *multi-level dynamic load balancing scheme*, which is an iterated application of the single-level scheme in a hierarchical manner. And we show how the scalability is thereby improved over that in the single-level scheme.

Keywords: parallel processing, load balancing, efficiency, scalability, isoefficiency, queuing model.

*Some of the materials in this work were presented in a preliminary form at the 6th Distributed Memory Computing Conference (1991).

Notations and Conventions

Throughout this paper, we assume that all random variables are defined on a standard probability space $(\Omega, \mathcal{B}, P)$, where Ω is the base space, $\mathcal{B} \subset 2^{\Omega}$ is the σ-algebra of events, and P is the underlying probability measure. Upper-case letters represent real-valued random variables and lower-case letters represent real numbers, unless otherwise specified. Basically, we adopt frequently-used notations such as those in [4]. In particular, we use the following.

a.s. : almost surely

i.i.d. : independent identically distributed

$\boldsymbol{N}$: set of all positive integers

$\boldsymbol{Z}$: set of all integers

$\boldsymbol{R}$: set of all real numbers

$\boldsymbol{R}_+$: set of all non-negative real numbers

A^c : complement of set A

$A \setminus B = A \cap B^c$: difference of sets A and B

$a \vee b = \max(a, b)$: maximum of $a, b \in \boldsymbol{R}$

$a \wedge b = \min(a, b)$: minimum of $a, b \in \boldsymbol{R}$

$a_+ = \max(a, 0)$: positive part of $a \in \boldsymbol{R}$

$\lfloor x \rfloor$: largest integer not more than $x \in \boldsymbol{R}$

$\lceil x \rceil$: smallest integer not less than $x \in \boldsymbol{R}$

$a_p = O(b_p) \iff \limsup_{p \to \infty} |a_p|/b_p < \infty$

$a_p = o(b_p) \iff \lim_{p \to \infty} |a_p|/b_p = 0$

$a_p = \Omega(b_p) \iff \liminf_{p \to \infty} a_p/b_p > 0$

$a_p = \omega(b_p) \iff \lim_{p \to \infty} a_p/b_p = \infty$

$a_p = \Theta(b_p) \iff a_p = O(b_p)$ and $a_p = \Omega(b_p)$

$a_p \simeq b_p \iff \lim_{p \to \infty} a_p/b_p = 1$

$a_p \lesssim b_p \iff \limsup_{p \to \infty} a_p/b_p \leq 1$

$a_p \gtrsim b_p \iff \liminf_{p \to \infty} a_p/b_p \geq 1$

1_A : defining function of a set A *i.e.* $1_A(x) = 1$ $(x \in A)$, $1_A(x) = 0$ (otherwise)

$P(A) = P\{A\}$: probability of an event A

$E(X) = E\{X\}$: expectation of a random variable X

$E(X, A) = E(X \cdot 1_A)$: expectation of X over an event A

$V(X) = E(X^2) - \{E(X)\}^2$: variance of a random variable X

$P(A \mid \mathcal{C}) = P\{A \mid \mathcal{C}\}$: conditional probability of an event A under a condition $\mathcal{C}$

$E(X \mid \mathcal{C}) = E\{X \mid \mathcal{C}\}$: conditional expectation of X under a condition $\mathcal{C}$

$V(X \mid \mathcal{C}) = E(X^2 \mid \mathcal{C}) - \{E(X \mid \mathcal{C})\}^2$

$\mathrm{ess.sup}(X) = \inf\{a \in \boldsymbol{R} \mid X \leq a \text{ a.s. }\}$: essential supremum of a random variable X

P^X : distribution of a random variable X

$[X \mid \mathcal{C}]$: conditional distribution of X under a condition $\mathcal{C}$

$X \prec Y$ (or $P^X \prec P^Y$) : X is stochastically smaller than Y

$\phi * \psi$: convolution of distributions ϕ, ψ over $\boldsymbol{R}$

$\mathcal{F}[X](z) = E(e^{izX})$: Fourier transformation of a random variable X

$\mathcal{F} \vee \mathcal{G}$: σ-algebra generated by σ-algebras $\mathcal{F}, \mathcal{G}$ in Ω

$\bigwedge_s \mathcal{F}_s = \bigcap_s \mathcal{F}_s$: the greatest σ-algebra in Ω smaller than any σ-algebra in a family $\{\mathcal{F}_s\}$

$\sigma[X_1, \ldots, X_n]$: complete subσ-algebra of $\mathcal{B}$ generated by random variables $X_1, \ldots, X_n$

$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt$: gamma function

$B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} dx$: beta function

$C = 0.5772\ldots$: Euler's constant

1 Introduction

The purpose of parallel processing is to accelerate the execution of time-consuming tasks by utilizing a number of processors. *Efficiency*, defined as the speedup divided by the number of processors, indicates the performance of parallel processing. It depends not only on the algorithm itself, but also on the number of processors, the problem size (the amount of computation required by the best sequential algorithm for solving it), and other factors. In general, for a given task, the efficiency decreases as the number of processors increases, as Amdahl's law illustrates. Therefore, it is important to analyze how the efficiency depends on these factors. In particular, as increasingly larger-scale multiprocessors are now being developed [14], scalability analysis of efficiency is becoming more and more important.

Various measures of the scalability of a parallel algorithm have been proposed for different situations [9]. Among these, the notion of *isoefficiency* succinctly captures the characteristics of scalability of a parallel algorithm [10, 3]. The efficiency usually decreases with an increasing number of processors, but recovers again with larger problems. The *isoefficiency function* indicates how much the problem size should be increased with the number of processors so as to maintain a constant efficiency. A parallel algorithm with a slowly increasing isoefficiency function is supposed to be scalable.

Ideally, the efficiency should be one, however, it may deteriorate due to various reasons: the load imbalance between the processors, inter-processor communication latency, speculative computations, or other overheads associated with parallel execution. In particular, if the algorithm is composed of many parts requiring unpredictable amounts of computation, as is usual with many combinatorial search problems, load imbalance between the processors is likely to occur and may have a great influence on the efficiency. Thus the load balancing is one of the central issues of parallel processing.

Kruskal and Weiss [8] studied the load balance when independent subtasks are allocated to processors on demand, assuming that the subtask running times have IHR (increasing hazard rate). They employed a queueing model with null arrival intervals, and showed how the non-uniformness of subtasks affects efficiency. However, a subtask queue was assumed to be given at the beginning, and access times to the queue were incorporated into the subtask running times. Thus, the possible bottleneck at generating the subtasks, which might degrade the efficiency, was not taken into account.

Also, in a distributed computer system, load balancing or *load sharing* attempts to improve the performance when different nodes are heavily or lightly loaded. Wang and Morris [15] surveyed more than ten different strategies, each of which has been analytically or numerically studied using queueing models for several particular probability distributions. These are concerned with *stationary* behaviors when independent tasks keep arriving at different nodes in the system from outside. The total processing capacity and the fairness of service are primarily studied there. On the other hand, in this paper, we will study the parallel execution time of a single *finite* divisible task.

Furuichi *et al.* [2] proposed the *multi-level dynamic load balancing scheme* for a program on an MIMD machine, which requires many *independent* pieces of computation. They evaluated its performance for an exhaustive tree exploration using Multi-PSI [11], a distributed memory MIMD machine with 64 processors. Their basic strategy is to divide a given problem into mutually independent *subtasks*, and distribute them to other processors *on demand*. This on-demand distribution dynamically balances the load between the processors. They pointed out that tuning the granularity of subtasks is essential for high performance. However, when the number of processors increases, the number of subtasks should increase correspondingly. Therefore, if the division is entrusted to one processor (*the single-level dynamic load balancing scheme*), it will become a bottleneck. So, they proposed divid-

ing the problem iteratively in a hierarchical manner (*the multi-level dynamic load balancing scheme*). Their experiments show that the latter is in fact more "scalable" than the former.

The purpose of this paper is to theoretically investigate the scalability of these load balancing schemes. We define a formal model of the single-level dynamic load balancing as a queuing system with multiple servers, taking into account the positive finite dividing costs. Here the unpredictable amount of computation required by each subtask is probabilistically treated. We will estimate the efficiency and show how it depends on the granularity of subtasks. The results on the single-level scheme are then applied to the analysis of the multi-level scheme. Among others, we show:

1. With subtasks of random sizes, the efficiency is worse than that with subtasks of exactly the same size. The isoefficiency function in the former case is $\log p$ times larger than that in the latter case, where p is the number of processors.

2. The multi-level scheme is indeed more "scalable" than the single-level scheme. The isoefficiency function for the former has a smaller fractional order of p than the latter.

3. In the tree configuration of the processors in the multi-level scheme, a processor at a higher level should have a larger fan out degree than one at a lower level. Their ratio is of the order of a fractional power of $\log p$. In particular, the uniform tree configuration with the same degree at all levels is *not* optimal.

In this paper, we present our analysis in detail, where familiarity with basic notions from modern probability theory based on measure theory is assumed. Some of the material in this work appeared previously in concise terms in Ref. [7], where we made a compromise in rigidity and gave only intuitive proofs.

The rest of the paper is organized as follows. In Section 2, the single-level and multi-level dynamic load balancing schemes are described. In Section 3, a formal model of single level dynamic load balancing is defined as a queuing system. Its behavior is analyzed in Sections 4–6: for a special case with a uniform subtask size, a typical case with random subtask sizes, and general cases. In Section 7, we investigate the probability distribution of the parallel execution time with the single-level scheme, based on which the multi-level scheme is investigated in Section 8. Finally, concluding remarks are given in Section 9.

2 Single-Level and Multi-Level Load Balancing Schemes

In this section, we describe the single-level and multi-level dynamic load balancing schemes proposed by Furuichi, Taki and Ichiyoshi [2]. These schemes are applicable to a parallel program on an MIMD machine, which requires many *independent* pieces of computation. In particular, they applied these schemes to exhaustive tree exploration, and evaluated the performance using Multi-PSI [11], a distributed memory MIMD machine with 64 processors. Here, we intuitively discuss the performance these schemes in order to promote detailed analysis in subsequent sections.

2.1 Assumptions for problems

We assume that our target problem can be divided into many subproblems as follows:

(A-1) The subproblems can be solved independently of one another.

(A-2) The amount of computation required by each subproblem is unpredictable before it is solved.

These assumptions seem natural in the OR-parallel exhaustive search procedures for many combinatorial problems. For instance, consider an exhaustive search of a tree, which represents the search space of a combinatorial problem. This tree can be divided into many subtrees at any depth d, and the search of the entire tree will be reduced to the searches of these subtrees. The latter searches will be independent of one another, as long as we don't employ any special pruning strategies. Furthermore, as is inherent in combinatorial problems, the entire tree will be irregular and the size of each subtree will be unpredictable before we search it. Thus, both assumptions are satisfied.

Now, in general, assumption (A-1) implies that the problem can be solved efficiently in parallel — different processors can solve different subproblems simultaneously. However, assumption (A-2) makes it difficult to *statically* balance the load between the processors. This prompts us to employ a *dynamic* load balancing strategy.

In the following, we will refer to the problem to be solved as a *task*, and similarly, to a subproblem as a *subtask*.

2.2 On-demand load distribution (single-level load balancing scheme)

We consider one of the most naïve on-demand load distribution techniques: — given a task, one *producer* processor divides it into a number of mutually independent subtasks, which are transmitted to the *consumer* processors on demand and then executed.

This *on-demand load distribution* will dynamically balance the load between the processors. We refer to this load balancing strategy as *the single-level load balancing scheme* [2].

2.3 Tuning the granularity

In order to make this load balancing scheme work efficiently, we have to tune the granularity of the subtasks. With a few large subtasks, load imbalance between the consumer processors is likely to occur. As an extreme case, if the number of subtasks is less than the number of consumer processors, some of the processors will never be used. On the other hand, with many small subtasks, the producer is likely to become a bottleneck.

So we assume that:

(A-3) The granularity of the subtasks can be controlled.

Namely, we assume that the task can be divided into more subtasks of smaller sizes or less subtasks of larger sizes, at will.

For example, let us consider the OR-parallel exhaustive search of a tree again. Each subtask is the search of a subtree with its root at depth d. So, by choosing an appropriate depth d, we can control the granularity of the subtasks.

2.4 Bottleneck in speedup

This on-demand load distribution technique has the apparent drawback of not being scalable.

Suppose that the number of processors is increased. Then, as we just saw in the last subsection, the granularity of the subtasks should be tuned accordingly. In this case, we have to increase the number of subtasks so that they are a certain extent larger than the number of consumer processors.

But if we increase the number of subtasks, the producer will become a bottleneck, since it is in charge of producing all of these subtasks. Thus, efficiency will inevitably drop.

So, in order to improve the scalability, we should remove such a producer bottleneck.

2.5 Multi-level load balancing scheme

The *multi-level dynamic load balancing scheme* alleviates the producer bottleneck by hierarchical load distribution.

In the *2-level dynamic load balancing scheme*, we divide a given task at a *root producer* into many subtasks. They are distributed to the *second level producers* on demand, and are, then, divided again into smaller *subsubtasks*. These subsubtasks are further distributed to the *leaf consumers* on demand, and are finally carried out. For instance, consider the exhaustive search of a tree again. This search tree is divided into many subtrees at depth d_1 by the root producer, and each subtree is again divided into many subsubtrees at depth $d_2 (> d_1)$. Thus the search of the entire tree is reduced to the searches of these subsubtrees.

Schemes of more than two levels can be defined similarly. More precisely, it is defined by induction on the number of levels ℓ. For $\ell = 1$, the 1-level dynamic load balancing is nothing but the single-level dynamic load balancing. For $\ell \geq 1$, the $(\ell + 1)$-level dynamic load balancing is the single-level dynamic load balancing with each consumer being replaced by a number of processors, which execute each subtask in parallel using the ℓ-level dynamic load balancing scheme. By increasing the number of levels ℓ, we can improve the scalability, as we will see later.

3 Model of On-Demand Load Distribution

In this section, we introduce a formal model of the on-demand load distribution (parallel execution with the single-level load balancing scheme), which gives an expression for the parallel execution time. This is intended to capture the speedup deterioration due to the load imbalance. Here, we assume that the inter-processor communication latency is negligible, which is, hence, not incorporated into the model. We describe the basic assumptions in our analysis, and define several useful characteristics.

3.1 Expression for parallel execution time

In this subsection, we give an expression for the parallel execution time of a task assuming its division into subtasks.

Let p be the number of consumer processors and N the number of subtasks. For each $1 \leq n \leq N$, let R_n be the CPU time required for executing the n-th subtask with a single processor. We refer to R_n as the *size* of the n-th subtask. Let U_n be the CPU time for the producer processor to cut the n-th subtask from the whole task.

For each $1 \leq n \leq N$, let X_n and Y_n be the time when the execution of the n-th subtask starts and ends respectively at a consumer processor. Neglecting the inter-processor communication latency, the first p subtasks are executed at different consumers as soon as they are generated at the producer. Hence,

$$X_n = U_1 + \cdots + U_n, \qquad (1 \leq n \leq p).$$

According to assumption (A-1) in Section 2.1, we assume that no suspension occurs in executing each subtask. Hence, Y_n is simply given by:

$$Y_n = X_n + R_n \qquad (1 \le n \le p).$$

As soon as at least one of these p subtasks is completed, the corresponding consumer becomes free. So, if the next subtask is generated at the producer, it can readily be executed at the free consumer. Hence,

$$X_{p+1} = \max\{U_1 + \cdots + U_{p+1}, \min\{Y_1, \ldots, Y_p\}\}, \qquad Y_{p+1} = X_{p+1} + R_{p+1}.$$

Similarly, each of the remaining subtasks can be treated inductively. Namely, they can be executed as soon as (i) they are generated at the producer and (ii) at least one of the consumers becomes free, although at most $p-1$ of the preceding subtasks may still be in execution. Finally, the *parallel execution time*, denoted by T_p, is given by the maximum of $Y_1, \ldots, Y_n$ — when all of the subtasks are completed.

We summarize these in the following definition.

Definition 3.1 (i) A *division* of a task is specified by $(\{R_n\}_{n=1}^N, \{U_n\}_{n=1}^N)$ with $N \in \boldsymbol{N}$ and $R_n,\ U_n \in \boldsymbol{R}_+$ for each $1 \le n \le N$, where N represents the number of subtasks, U_n the CPU time for producing the n-th subtask, and R_n the *size* of the n-th subtask (CPU time required for execution).

(ii) For $1 < p \in \boldsymbol{N}$ and a division of a task specified by $(\{R_n\}_{n=1}^N, \{U_n\}_{n=1}^N)$, we define T_p, the *parallel execution time* with p consumers by:

$$T_p = \max_{1 \le n \le N} Y_n \tag{3.1}$$

where Y_n as well as X_n, O_n, Z_n are defined by induction on n.

$$\left\{ \begin{array}{rcll} O_n & = & \displaystyle\sum_{k=1}^{n} U_k & (1 \le n \le N) \\ X_n & = & O_n \vee Z_n & (1 \le n \le N) \\ Y_n & = & X_n + R_n & (1 \le n \le N) \\ Z_n & = & \left\{ \begin{array}{ll} \displaystyle\min_{1 \le i_1 < \cdots < i_{p-1} \le n-1} \ \max_{\substack{1 \le k \le n-1 \\ k \ne i_1, \ldots, i_{p-1}}} Y_k & (p < n \le N) \\ 0 & (1 \le n \le p). \end{array} \right. & \end{array} \right. \tag{3.2}$$

Here, O_n represents the birth time of the n-th subtask, X_n (Y_n) the start (end) of its execution period, and Z_n the time when at least one of the consumers becomes ready to execute the n-th subtask after completing the previously assigned subtask. Note that Z_n depends only on $Y_1, \ldots, Y_{n-1}$, which makes induction on n possible.

Equation (3.2) defines a *queuing system* with p servers, where R_n represents the service time for the n-th customer and U_n represents the interval of arrival between the $(n-1)$-th and n-th customers. Here, a subtask corresponds to a customer in the queueing system, and a consumer processor corresponds to a server. Hence, the production of a subtask corresponds to the arrival of a customer and the execution of a subtask corresponds to the service to a customer. Thus, the dynamic behavior of the on-demand load distribution can be naturally expressed as a queueing system with multiple servers.

3.2 Probabilistic assumptions

In terms of assumption (A-2) in Section 2.1, the exact values of N, R_n and U_n will not be known beforehand. One of the worthwhile approaches is the average-case analysis of algorithms initiated by R. Karp [6] and others. We suppose that a *problem instance* is given randomly from a *problem space* (a set of similar problems), and regard N, R_n and U_n as random variables. And we will be engaged in the average-case analysis over the problem space.

In this paper, we assume that the subtasks are probabilistically equivalent and that there is no correlation between them. Namely,

$$\{R_n\}_{n=1,2,\ldots} : \text{ i.i.d.}, \qquad \{U_n\}_{n=1,2,\ldots} : \text{ i.i.d.}, \qquad N, R_n, U_n \ (n = 1, 2, \ldots) : \text{ independent.}$$

If the subtask sizes were not probabilistically equivalent, namely, if we knew that some of the subtasks were expected to be larger than the others, they should be distributed in a different manner. If there were correlation between the subtask sizes, for example, if we could predict the size of a subtask based on that of another, we could balance the load better, based on such prediction. However, we will not discuss such cases here as these might suggest other more efficient load distribution strategies, which might be complicated, strongly problem-dependent, and hard to analyze in a general setting.

According to the analogy with a queuing system, we introduce several notations:

$$\frac{1}{\lambda} = E(U_n) \quad : \quad \text{average time for producing a subtask}$$

$$\frac{1}{\mu} = E(R_n) \quad : \quad \text{average subtask size}$$

In other words, λ represents the *production rate* of the subtasks per unit time and μ represents the *consumption rate* of the subtasks per unit time at each consumer processor. Reducing the scale of unit time, if necessary, we assume that $\lambda, \mu < 1$ (in fact, we will be interested in the situation where $\lambda, \mu \to 0$). We define ρ as the ratio of the production rate to the overall consumption rate of the subtasks:

$$\rho = \frac{\lambda}{\mu p}.$$

A small value of ρ implies fine-grained granularity, while a large value of ρ implies coarse-grained granularity. Intuitively, the optimal granularity of the subtasks seems to be specified by $\rho = 1$ (this is when the subtasks are being produced and consumed at the same speed). In later sections, we will see how the performance of the parallel execution depends on the value of ρ. These notations, λ, μ and ρ, are conventional in queuing theory.

We also write:

$$\nu = E(N) : \text{ average number of subtasks}$$

In terms of assumption (A-3) in Section 2.3, we assume that we can control the granularity of the subtasks in the *average* sense. Namely, we assume that we can control ν but not N itself.

We define the *size* of the task, denoted by T_1, as its *sequential execution time*, i.e., the CPU time required for executing the whole task with a single processor. We assume that it is given by the sum of the sizes of all the subtasks:

$$T_1 = R_1 + \cdots + R_N$$

We write:

$$t_1 = E(T_1) : \text{ average task size}$$

Accordingly, $t_1 = \nu/\mu$ holds regardless of the choice of the granularity.

On the other hand, we regard U_ns as the overheads associated with parallel processing, and assume that these are moderate in the following sense. Informally, producing a subtask is just computing an "address" which specifies the portion of the search space assigned to it. We assume that this address should be computed in a reasonable amount of time, *i.e.*, in a polynomial time of $\log p$, the description length of p. Namely, we assume:

$$\frac{1}{\lambda} = O((\log p)^{k'}) \quad \text{for some } k' \geq 0.$$

For example, in the exhaustive tree search, a subtask, i.e., a subtree with its root at depth d, can be specified by a path of length d from the root of the entire tree to its own root. If the nodes of the tree have bounded degrees, the "address" of a subtask can be written down in $O(d)$ time. We should choose $d = O(\log p)$ in order to produce a polynomial number of subtasks in p. Later we will see that the optimal granularity, in fact, takes a polynomially bounded number of subtasks in p (cf. (5.15), (6.20)).

Furthermore, we assume that the worst-case computation time for the "address" of a subtask,

$$\alpha \equiv \text{ess. sup } U_n$$

is not too diverse from the average-case computation time $1/\lambda$. If U_n's were the same for all n, the average-case and worst-case computation time would coincide: $\alpha = 1/\lambda$. However, we will assume a somewhat more relaxed condition:

$$\alpha\lambda = O(1).$$

Since we are interested in the scalability of the load balancing technique with an increasing number of processors, we should consider increasingly larger problems. However, the size of a task (problem), defined as the CPU time required for executing it on a single processor, is itself a random variable given by $T_1 = R_1 + \cdots + R_N$. So, we assume a family of problem spaces with different *average* sizes (t_1). A *family* corresponds to a general question, e.g., color the vertices of a graph so as to satisfy a certain condition. Each problem space may consist of problem instances with the same description length, e.g., graphs with the same number of edges. Problem instances with a larger description length will require a larger amount of computation on average, namely, they will have a larger average size, t_1. Thus, those with different description lengths will constitute a family of problem spaces with different average sizes.

In general, in order to exploit a larger number of processors efficiently, a correspondingly larger task should be treated that can compensate for the growing parallelization overheads. Namely, we will study the case with $t_1 \to \infty$ and $p \to \infty$. However, we assume that the average task size should not grow *too* fast, i.e., t_1 should be polynomially bounded by p, i.e.,

$$t_1 = O(p^k) \quad \text{for some } k > 1.$$

We summarize these assumptions in the following.

Assumption 3.1 We assume a family of independent random variables:

$$\mathcal{T}^{(t_1,\nu,p)} = (N^{(t_1,\nu)}, \{R_n^{(t_1,\nu)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu,p)}\}_{n=1}^{\infty})$$

parameterized by $t_1 > \nu > 1$ and $1 < p \in \boldsymbol{N}$ with

$$1 < t_1 \leq ap^k \quad \text{for some } a > 1,\ k > 2 \tag{3.3}$$

such that the following hold.

(i) $N^{(t_1,\nu)} \in \boldsymbol{N}$ a.s., and each of $\{R_n^{(t_1,\nu)}\}_{n=1}^{\infty}$ and $\{U_n^{(t_1,\nu,p)}\}_{n=1}^{\infty}$ are i.i.d. over $\boldsymbol{R}_+$.

(ii) $E(N^{(t_1,\nu)}) = \nu$.

(iii) $T_1^{(t_1)}$, defined by $T_1^{(t_1)} = R_1^{(t_1,\nu)} + \cdots + R_{N^{(t_1,\nu)}}^{(t_1,\nu)}$, does not depend on ν, and $E(T_1^{(t_1)}) = t_1$.

(iv) $1/\lambda^{(t_1,\nu,p)} \equiv E(U_n^{(t_1,\nu,p)})$ and $\alpha^{(t_1,\nu,p)} \equiv \text{ess. sup } U_n^{(t_1,\nu,p)}$ satisfy

$$1 < \frac{1}{\lambda^{(t_1,\nu,p)}} \leq a' \cdot (\log p)^{k'} \quad \text{for some } a' > 1,\ k' \geq 0 \tag{3.4}$$

$$\sup_{\nu,\, t_1} \alpha^{(t_1,\nu,p)} \lambda^{(t_1,\nu,p)} \leq a'' \quad \text{for some } a'' \geq 1 \tag{3.5}$$

We will refer to such $\mathcal{T}^{(t_1,\nu,p)}$ as a *division of a task* (of expected size t_1) between p consumers in granularity t_1/ν. For each $t_1 > 1$, $\mathcal{T}^{(t_1)} = \{\mathcal{T}^{(t_1,\nu,p)} \mid 1 < \nu < t_1 \leq ap^k,\ 1 < p \in \boldsymbol{N}\}$ represents a *divisible task* of expected size t_1. We will also refer to $\mathcal{T} = \{\mathcal{T}^{(t_1,\nu,p)} \mid 1 < \nu < t_1 \leq ap^k,\ 1 < p \in \boldsymbol{N}\}$ as a *family of divisible tasks*. For each (t_1, ν, p), the parallel execution time $T_p^{(t_1,\nu,p)}$ is defined by (3.1)-(3.2), and $\mu^{(t_1,\nu)} \equiv \nu/t_1 < 1$, $\rho^{(t_1,\nu,p)} \equiv \lambda^{(t_1,\nu,p)}/\mu^{(t_1,\nu)}p$. For brevity, we will often omit the superscripts and write: $N = N^{(t_1,\nu)}$, $R_n = R_n^{(t_1,\nu)}$, $\mu = \mu^{(t_1,\nu)}$ and so on.

Here we give two examples satisfying this assumption. These will be studied in Sections 4 and 5.

Example 3.1 (deterministic case) This is a case when all the random variables are distributed according to the delta distributions: for any $1 < p \in \boldsymbol{N}$ and $t_1 > \nu > 1$ with $\nu \in \boldsymbol{N}$ and (3.3),

$$T_1^{(t_1)} \equiv t_1, \qquad N^{(t_1,\nu)} \equiv \nu, \qquad R_n^{(t_1,\nu)} \equiv \frac{1}{\mu^{(t_1,\nu)}}, \qquad U_n^{(t_1,\nu,p)} \equiv \frac{1}{\lambda^{(t_1,\nu,p)}} \qquad (\nu = 1, 2, \ldots)$$

where $1/\mu^{(t_1,\nu)} = t_1/\nu$ and $\lambda^{(t_1,\nu,p)}$ is any constant satisfying (3.4). This case corresponds to a *uniform* program, in which we know the exact size of a given task and can divide it into an arbitrary number of subtasks of exactly the same size in a constant time for each.

Example 3.2 (exponential case) This is a case when the task (subtask) size is distributed according to the exponential distribution, and the number of subtasks is distributed according to the geometric distribution — for any $t_1 > \nu > 1$,

$$P(x < T_1^{(t_1)} \leq x + dx) = \frac{1}{t_1} \exp(-\frac{x}{t_1})dx \quad \text{for } \forall x \geq 0 \qquad \text{(exponential distribution with mean } t_1\text{)}$$

$$P(x < R_n^{(t_1,\nu)} \leq x + dx) = \mu e^{-\mu x} dx \quad \text{for } \forall x \geq 0,\ \forall n \qquad \text{(exponential distribution with mean } 1/\mu\text{)}$$

$$P(N^{(t_1,\nu)} = n) = \frac{1}{\nu}\left(1 - \frac{1}{\nu}\right)^{n-1} \quad \text{for } \forall n = 1, 2, \ldots \qquad \text{(geometric distribution with mean } \nu\text{)}$$

where $\mu = \mu^{(t_1,\nu)} = \nu/t_1$. For example, if a subtask consists of a simple loop and there is a constant probability of termination at each iteration, the size (execution time) of a subtask is distributed exponentially. The above condition for $R_n^{(t_1,\nu)}$ is its continuous variant, which is referred to as *Markov service time* in queueing theory. Condition (iii) in Assumption 3.1 is verified by the following proposition.

Proposition 3.1 *Let N and R_n $(n = 1, 2, \ldots)$ be independent random variables such that N is distributed according to the geometric distribution with mean $\nu > 1$ and each R_n is distributed according to the exponential distribution with mean $\sigma > 0$. Then $R_1 + \cdots + R_N$ is distributed according to the exponential distribution with mean $\nu\sigma > 0$.*

PROOF: Applying Fourier transformation, we obtain

$$\mathcal{F}[R_1 + \cdots + R_N](z) = E((1 - i\sigma z)^{-N}) = \sum_{n=1}^{\infty} \frac{1}{\nu} \cdot (1 - \frac{1}{\nu})^{n-1} \cdot (1 - i\sigma z)^{-n} = \frac{1}{1 - i\nu\sigma z}. \blacksquare$$

3.3 Performance measures and other technical notions

In this subsection, we define several useful characteristics and notions. We define the *mean execution time* as $t_p = E(T_p)$ and adopt as a proper definition of the *mean speedup*

$$s_p = \frac{t_1}{t_p} \ : \ \text{"collective" mean speedup}$$

instead of the usual *arithmetic* mean speedup $E(T_1/T_p)$. In general, a "collective" mean represents a *weighted* mean of ratios according to their denominators and is calculated by dividing the sum of their numerators by the sum of their denominators. For instance, suppose that a task is chosen at random uniformly from the problem space with $T_1 \in \{t_1^{(1)}, \ldots, t_1^{(I)}\}$ and $T_p \in \{t_p^{(1)}, \ldots, t_p^{(I)}\}$. Then,

$$s_p = \frac{t_1^{(1)} + \cdots + t_1^{(I)}}{t_p^{(1)} + \cdots + t_p^{(I)}} = \sum_{i=1}^{I} \frac{t_p^{(i)}}{t_p^{(1)} + \cdots + t_p^{(I)}} \cdot \frac{t_1^{(i)}}{t_p^{(i)}}$$

where $t_1^{(i)}/t_p^{(i)}$ is the speedup for the i-th problem instance and is weighted in proportion to its parallel execution time $t_p^{(i)}$.

We usually have $0 < s_p < p$, and $s_p = p$ in the ideal case. Normalizing the speedup, we define:

$$\eta = \frac{s_p}{p} = \frac{t_1}{pt_p} \quad : \quad \text{mean efficiency} \tag{3.6}$$

$$\eta_{\max} = \sup_{\nu} \eta(p, t_1, \nu) \quad : \quad \text{maximal mean efficiency.} \tag{3.7}$$

Thus, we usually have $0 < \eta < 1$, and $\eta = 1$ in the ideal case. The *maximal mean efficiency* $\eta_{\max}$ is the mean efficiency when we choose the optimal granularity for a given p and t_1.

Finally, we define a technical notion that will be required in the proofs in later sections.

Definition 3.2 Let $T^{(t_1,\nu,p)} = (N^{(t_1,\nu)}, \{R_n^{(t_1,\nu)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu,p)}\}_{n=1}^{\infty})$ be as in Assumption 3.1. For each $t \geq 0$, we define $\mathcal{F}_t = \mathcal{F}_t^{(t_1,\nu,p)}$, *information up to time* t, by

$$\mathcal{F}_t = \bigwedge_{s>t} \bar{\sigma}[X_n \wedge s,\ Y_n \wedge s,\ Z_n \wedge s \mid n = 1, 2, \ldots]$$

where X_n, Y_n and Z_n are those defined by (3.2).

Note that $\{\mathcal{F}_t\}_{t\geq 0}$ is a complete and right-continuous increasing family of σ-algebras on $(\Omega, \mathcal{B}, P)$. Each of X_n, Y_n and Z_n for any n is a stopping time (Markov time) with respect to $\{\mathcal{F}_t\}_{t\geq 0}$. Strictly speaking, $\mathcal{F}_t$ represents the *information available to the consumers up to time* t. In general, O_n is *not* a stopping time with respect to $\{\mathcal{F}_t\}_{t\geq 0}$.

For any stopping time S, we refer to $\mathcal{F}_S$ as the *information up to time* S, where

$$\mathcal{F}_S = \{A \in \mathcal{B} \mid A \cap \{S \leq t\} \subset \mathcal{F}_t \quad (\forall t \geq 0)\}.$$

4 Deterministic Case

In this section, we investigate the efficiency of on-demand load distribution (single-level load balancing) in the deterministic case, in which a task can be divided into subtasks of exactly the same size in a constant time for each (cf. Example 3.1). In this case, all the random variables are constants (distributed according to the delta distributions):

$$T_1 \equiv t_1, \qquad N \equiv \nu, \qquad U_n \equiv \frac{1}{\lambda}, \qquad R_n \equiv \frac{1}{\mu} \qquad (n = 1, 2, \ldots) \tag{4.1}$$

with $t_1 > \nu > 1$, $\nu \in \boldsymbol{N}$, $0 < \lambda < 1$ and $\mu = \nu/t_1$. The deterministic case corresponds to $D/D/s$ with $s > 1$ in queuing theory.

Throughout this section, we assume that $\mathcal{T}^{(t_1,\nu,p)} = (N^{(t_1,\nu)}, \{R_n^{(t_1,\nu)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu,p)}\}_{n=1}^{\infty})$ satisfies Assumption 3.1 and (4.1) for each t_1, ν and p. We call such $\mathcal{T} = \{\mathcal{T}^{(t_1,\nu,p)}\}$ a *family of divisible tasks of deterministic type*. We will constantly use notations O_n, X_n, Y_n, Z_n and T_p defined by (3.1)-(3.2). The efficiency η is defined in (3.6).

Theorem 4.1 *Let $\mathcal{T}$ be a family of divisible tasks of deterministic type. Then, we have $\eta_{\max} \le 1/2$ for $\lambda t_1 \le p^2$, and $1 + \dfrac{p^2}{\lambda t_1} \le \dfrac{1}{\eta_{\max}} \le 1 + \dfrac{p^2}{\lambda t_1} \cdot (1 - \dfrac{1}{p})^{-1}$ for $\lambda t_1 > p^2$.*

PROOF: According to (3.2), the subtasks are assigned to the consumers circularly since U_n and T_n are constants independent of n. Note that each consumer completes each subtask in time $1/\mu$, and that each consumer can get a new subtask in time p/λ after it got the last subtask. We firstly establish several basic estimates of T_p for (i) $0 < \rho < 1$ and (ii) $\rho \ge 1$ separately.

(i) When $0 < \rho < 1$, we have $1/\mu < p/\lambda$. This implies that each consumer, after completing its last subtasks, always waits for a while $(p/\lambda - 1/\mu)$ for a new subtask to come. Accordingly, the ability of the producer is relatively poor, to which the overall execution time is sensitive:

$$T_p = O_N + R_N = \frac{\nu}{\lambda} + \frac{1}{\mu} = \frac{\nu}{\lambda} + \frac{t_1}{\nu} \tag{4.2}$$

(ii) On the other hand, when $\rho \ge 1$, we have $1/\mu \ge p/\lambda$. This implies that each subtask always waits for a while $(1/\mu - p/\lambda)$ for its turn to be carried out, until the consumer in charge completes the previously assigned subtask. Therefore, the ability of the consumers is relatively poor, to which the overall execution time is sensitive. Now let

$$\nu = kp + r, \qquad k \ge 0, \qquad 1 \le r \le p, \qquad k, r \subset \boldsymbol{Z}$$

According to (3.2), the last subtask is carried out by the r-th consumer as its $(k+1)$-th job, and

$$T_p = Y_{kp+r} = O_r + \sum_{j=0}^{k} R_{jp+r} = \frac{r}{\lambda} + \frac{k+1}{\mu} = \frac{r}{\lambda} + \frac{kp}{\mu p} + \frac{1}{\mu}$$

Since $\lambda \ge \mu p$, we have

$$\frac{kp+r}{\lambda} + \frac{1}{\mu} \le T_p \le \frac{kp+r}{\mu p} + \frac{1}{\mu}$$

$$\therefore \quad \frac{\nu}{\lambda} + \frac{t_1}{\nu} \le T_p < \frac{t_1}{p} + \frac{t_1}{\nu} \tag{4.3}$$

And also

$$T_p = \frac{r}{\lambda} + (\frac{\nu - r}{p} + 1) \cdot \frac{1}{\mu} = \frac{\nu}{\mu p} + \frac{p - r}{\mu p} + \frac{r}{\lambda} \ge \frac{\nu}{\mu p} + \frac{p}{\lambda} \tag{4.4}$$

Now assume that $\lambda t_1 \le p^2$. By (4.2) and (4.3), in either case of (i) or (ii) above, we have

$$T_p \ge \frac{\nu}{\lambda} + \frac{t_1}{\nu} \ge 2\sqrt{\frac{t_1}{\lambda}} \qquad \therefore \ \frac{1}{\eta} = \frac{pT_p}{t_1} \ge \frac{2p}{\sqrt{\lambda t_1}} \ge 2$$

Assume that $\lambda t_1 > p^2$ throughout the rest of this proof. First we give a lower bound of $1/\eta_{\max}$. When $0 < \rho < 1$, according to (4.2), T_p is a convex function of $\nu > 0$, and takes a minimum value at $\nu_0 = \sqrt{\lambda t_1}$. Since $\nu = \lambda t_1/\rho p \ge \lambda t_1/p > \nu_0$, we have

$$T_p \ge \frac{1}{\lambda} \cdot \frac{\lambda t_1}{p} + t_1 \cdot \frac{p}{\lambda t_1} \qquad \therefore \ \frac{1}{\eta} \ge 1 + \frac{p^2}{\lambda t_1}$$

On the other hand, when $\rho \ge 1$, (4.4) immediately gives $1/\eta \ge 1 + p^2/\lambda t_1$.

Finally, we give an upper bound of $1/\eta_{\max}$. By choosing $\nu = \lfloor \lambda t_1/p \rfloor$, we have $\nu > 1$ and $\rho = \lambda t_1/\nu p \ge 1$. Hence by (4.3)

$$\frac{1}{\eta} \le 1 + \frac{p}{\nu} \le 1 + \frac{p^2}{\lambda t_1 - p} \le 1 + \frac{p^2}{\lambda t_1} \cdot (1 - \frac{1}{p})^{-1} \ \blacksquare$$

As one can see in the proof of this theorem, the efficiency is maximized when $\nu = \lfloor \lambda t_1/p \rfloor$, which implies $\rho \approx 1$, i.e., the production and consumption rate of the subtasks should be almost equal. This is intuitively convincing. The optimal efficiency is asymptotically given by

$$\frac{1}{\eta_{\max}} \simeq 1 + \frac{p^2}{\lambda t_1} \qquad \text{as} \quad p \to \infty \quad \text{with} \quad t_1 > \frac{p^2}{\lambda} \tag{4.5}$$

This expression succinctly describes how the efficiency η depends on the number of consumers p, the production rate λ, and the overall task size t_1. In particular, the task size t_1 should be $\Omega(p^2/\lambda)$ as $p \to \infty$, in order to maintain a constant efficiency. Its principal factor is given by p^2 because of the reasonable cost in producing the subtasks as in (3.4).

5 Exponential Case

In this section we investigate the efficiency of the on-demand load distribution (single-level load balancing) in the exponential case (cf. Example 3.2). We assume that the task (subtask) size is distributed according to the exponential distribution and that the number of subtasks is distributed according to the geometric distribution:

$$\begin{cases} P(x < T_1 \le x + dx) = \dfrac{1}{t_1} \exp(-\dfrac{x}{t_1}) dx & \text{for } \forall x \ge 0 \\ P(x < R_n \le x + dx) = \mu e^{-\mu x} dx & \text{for } \forall x \ge 0,\ \forall n = 1, 2, \ldots \\ P(N = n) = \dfrac{1}{\nu} \left(1 - \dfrac{1}{\nu}\right)^{n-1} & \text{for } \forall n = 1, 2, \ldots \end{cases} \tag{5.1}$$

with $t_1 > \nu > 1$ and $\mu = \nu/t_1$.

In this case, although both of the subtask size and the number of subtasks are random, they satisfy the so-called *Markov property*. For any $t > 0$ and $n = 1, 2, \ldots$, the distribution of $R_n - t$ under the condition $R_n > t$ does not depend on t, and is the same as that of R_n itself. For any $k = 1, 2, \ldots$, the distribution of $N - k$ under the condition $N > k$ does not depend on k, and is the same as that of N itself. Namely,

$$P(x < R_n - t \le x + dx \mid R_n > t) = \mu e^{-\mu x} dx \qquad \text{for } \forall t \ge 0,\ \forall x \ge 0,\ \forall n = 1, 2, \ldots$$

$$P(N - k = n \mid N > k) = \frac{1}{\nu}\left(1 - \frac{1}{\nu}\right)^{n-1} \qquad \text{for } \forall k = 1, 2, \ldots, \forall n = 1, 2, \ldots$$

These properties make the analysis exceptionally easy. The exponential case corresponds to $GI/M/s$ with $s > 1$ in queuing theory.

Throughout this section we assume that $T^{(t_1,\nu,p)} = (N^{(t_1,\nu)}, \{R_n^{(t_1,\nu)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu,p)}\}_{n=1}^{\infty})$ satisfies Assumption 3.1 and (5.1) for each t_1, ν and p. We call such $\mathcal{T} = \{T^{(t_1,\nu,p)}\}$ a *family of divisible tasks of exponential type*. We will constantly use notations O_n, X_n, Y_n, Z_n and T_p defined by (3.1)-(3.2). The efficiency η is defined in (3.6).

5.1 Lower estimate of the efficiency

In this subsection we give a lower estimate of the efficiency in the exponential case. Let us define $W_n = X_n - O_n$, *waiting time* of the n-th subtask for the start of its execution.

Lemma 5.1 *There exists $\{V_n\}_{n=1}^{\infty}$ such that*

(i) $\{V_n\}_{n=1}^{\infty}$: *i.i.d. according to the exponential distribution with mean $1/\mu p$*

(ii) $\{U_n\}_{n=1}^{\infty}$, $\{V_n\}_{n-1}^{\infty}$: *independent*

(iii) $W_1 = 0, \qquad 0 \le W_{n+1} \le \max(W_n + V_n - U_{n+1}, 0)$ *a.s.* $(\forall n \in \boldsymbol{N})$

PROOF: Note that $Z_{n+1} > X_n$ occurs if and only if each of the p consumers is still executing one of the first n subtasks at time $X_n + 0$. We first make an observation on $Z_{n+1} - X_n$ when it is positive.

Assume that $\mathcal{F}_{X_n}$, the information up to time X_n, is given and that $Z_{n+1} > X_n$ has occurred with the subtasks of order $k_1, \ldots, k_p$ being executed at time $X_n + 0$ for $1 \le k_1 < \cdots < k_p \le n$. Let a_i be the elapsed time in executing the k_i-th subtask up to time X_n for each $i = 1, 2, \ldots, p$. Then we have $R_{k_1} > a_1, \ldots, R_{k_p} > a_p$ and $Z_{n+1} - X_n = \min\{R_{k_1} - a_1, \ldots, R_{k_p} - a_p\}$. Hence $Z_{n+1} - X_n$ is distributed according to the exponential distribution with mean $1/\mu p$ owing to the Markov property.

Now, let $\{\tilde{V}_n\}_{n=1}^{\infty}$ be i.i.d. according to the exponential distribution with mean $1/\mu p$, independent of $\{U_n\}_{n=1}^{\infty}$ and $\{R_n\}_{n=1}^{\infty}$. For each n, let $\mathcal{G}_n = \mathcal{F}_{X_n} \vee \bar{\sigma}[\tilde{V}_1, \ldots, \tilde{V}_{n-1}]$, and define V_n by

$$V_n = (Z_{n+1} - X_n) \cdot 1_{Z_{n+1} > X_n} + \tilde{V}_n \cdot 1_{Z_{n+1} \le X_n}$$

From the above observation, (ii) follows immediately. We also have:

(a) V_n is $\mathcal{G}_{n+1}$-measurable.

(b) V_n is independent of $\mathcal{G}_n$.

(c) V_n is distributed according to the exponential distribution with mean $1/\mu p$.

Indeed, (a) follows immediately from $Z_{n+1} \le X_{n+1}$. For $\forall x \ge 0$ and $\forall A \in \mathcal{G}_n$, we have

$$\begin{aligned}
& P(\{V_n \ge x\} \cap A) \\
= \; & P(\{Z_{n+1} - X_n \ge x\} \cap A \cap \{Z_{n+1} > X_n\}) + P(\{\tilde{V}_n \ge x\} \cap A \cap \{Z_{n+1} \le X_n\}) \\
- \; & E(P(Z_{n+1} - X_n \ge x \mid \mathcal{G}_n) \cdot 1_A \mid Z_{n+1} > X_n) \cdot P(Z_{n+1} > X_n) \\
& + P(\tilde{V}_n \ge x) \cdot P(A \cap \{Z_{n+1} \le X_n\}) \\
& \quad (\because) \; \{Z_{n+1} > X_n\} \text{ and } A \text{ are } \mathcal{G}_n\text{-measurable and } \tilde{V}_n \text{ is independent of } \mathcal{G}_n. \\
= \; & E(P(Z_{n+1} - X_n \ge x \mid \mathcal{F}_{X_n}) \cdot 1_A \mid Z_{n+1} > X_n) \cdot P(Z_{n+1} > X_n)
\end{aligned}$$

$$
\begin{aligned}
&+ e^{-\mu p x} \cdot P(A \cap \{Z_{n+1} \le X_n\}) \\
= \;& E(e^{-\mu p x} \cdot 1_A \mid Z_{n+1} > X_n) \cdot P(Z_{n+1} > X_n) + e^{-\mu p x} \cdot P(A \cap \{Z_{n+1} \le X_n\}) \\
= \;& e^{-\mu p x} \cdot P(A)
\end{aligned}
$$

This implies (b) and (c). From (a), (b) and (c), we obtain (i). Finally, according to (3.2), we have

$$
W_{n+1} = \begin{cases} \max(W_n + Z_{n+1} - X_n - U_{n+1}, 0) & (Z_{n+1} > X_n) \\ \max(W_n - U_{n+1}, 0) & (Z_{n+1} \le X_n) \end{cases}
$$

from which (iii) follows. ∎

Let us define $\{\tilde{W}_n\}_{n=1}^{\infty}$ and $\{S_n\}_{n=1}^{\infty}$ by

$$
\tilde{W}_1 = 0, \qquad \tilde{W}_{n+1} = \max(\tilde{W}_n + V_n - U_{n+1}, 0) \qquad (n = 1, 2, \ldots)
$$

$$
S_1 = 0, \qquad S_n = \sum_{k=1}^{n-1} (V_k - U_{k+1}) \qquad (n = 1, 2, \ldots)
$$

Owing to the preceding lemma, we obtain

$$
0 \le W_n \le \tilde{W}_n, \qquad S_n \le \tilde{W}_n \qquad \text{a.s.} \qquad (n = 1, 2, \ldots)
$$

by induction on n. Hence we have $\tilde{W}_{n+1} \le \tilde{W}_n + V_n - U_{n+1} + U_{n+1} \cdot 1_{S_{n+1} < 0}$ a.s., and

$$
E(\tilde{W}_{n+1}) \le E(\tilde{W}_n + V_n - U_{n+1}) + E(U_{n+1}, S_{n+1} < 0) \le E(\tilde{W}_n) + E(V_n - U_{n+1}) + \alpha P(S_{n+1} < 0)
$$

$$
\therefore \; E(\tilde{W}_n) \le E(S_n) + \alpha \sum_{k=1}^{n} P(S_k < 0)
$$

Now assume that $\rho > 1$. For each $n > 1$,

$$
E(S_n) = (n-1)\left(\frac{1}{\mu p} - \frac{1}{\lambda}\right) = \frac{(n-1)(\rho - 1)}{\lambda} > 0
$$

$$
V(S_n) \le (n-1)\left(\frac{1}{\mu^2 p^2} + \alpha^2\right) = \frac{(n-1)(\rho^2 + \alpha^2 \lambda^2)}{\lambda^2}
$$

and by Chebyshev's inequality

$$
P(S_n < 0) \le \frac{\rho^2 + \alpha^2 \lambda^2}{(n-1)(\rho - 1)^2}
$$

Therefore,

$$
E(W_n) \le E(\tilde{W}_n) \le \frac{\rho - 1}{\lambda} \cdot (n-1) + \frac{\alpha(\rho^2 + \alpha^2 \lambda^2)}{(\rho - 1)^2} \{1 + \log(n-1)\} \qquad (1 < \forall n \in \boldsymbol{N})
$$

$$
\therefore \; E(W_N) \le \frac{\rho - 1}{\lambda} \cdot (\nu - 1) + \frac{\alpha(\rho^2 + \alpha^2 \lambda^2)}{(\rho - 1)^2} \{1 + \log(\nu - 1)\}
$$

Let K be the number of consumers executing one of the first N subtasks at time $X_N + 0$. By Proposition 5.1 below,

$$
E(T_p - X_N \mid K = q) \le \frac{1}{\mu} \cdot \left\{ \log(q+1) - \frac{1}{2(q+1)} + C + 2 \right\} \qquad (q = 0, 1, \ldots, p)
$$

$$
\therefore \; E(T_p - X_N) \le \frac{1}{\mu} \cdot \left\{ \log(p+1) - \frac{1}{2(p+1)} + C + 2 \right\}
$$

where $C = 0.5772\ldots$ denotes Euler's constant. From these estimates, it follows that:

$$\begin{aligned} E(T_p) &\le E(O_N) + E(W_N) + E(T_p - X_N) \\ &\le \frac{\nu}{\lambda} + \frac{\rho - 1}{\lambda} \cdot (\nu - 1) + \frac{\alpha(\rho^2 + \alpha^2\lambda^2)}{(\rho - 1)^2}\{1 + \log(\nu - 1)\} + \frac{1}{\mu} \cdot \{\log(p+1) + C + 2\} \\ &\le \frac{\rho\nu}{\lambda} + \frac{\rho p}{\lambda}\{\log(p+1) + C + 2\} + \frac{\alpha(\rho^2 + \alpha^2\lambda^2)}{(\rho - 1)^2}(1 + \log\nu) \end{aligned}$$

$$\therefore \ \frac{1}{\eta} \le 1 + \frac{\rho p^2}{\lambda t_1} \cdot \left\{ \log(p+1) + C + 2 + \frac{\alpha\lambda(\rho^2 + \alpha^2\lambda^2)}{\rho(\rho - 1)^2 p} \cdot \left(1 + \log\frac{\lambda t_1}{\rho p}\right) \right\}$$

Thus we obtain the next theorem.

Theorem 5.1 *Let $\mathcal{T}$ be a family of divisible tasks of exponential type. When $\rho > 1$, we have*

$$\frac{1}{\eta} \le 1 + \frac{\rho p^2}{\lambda t_1} \cdot \left(\log(p+1) + C + 2 + \frac{c_0\rho}{(\rho - 1)^2} \cdot \frac{\log p}{p} \right)$$

where c_0 is a constant depending only on the constants a, a', a'', k, k' appeared in (3.3)-(3.5).

Proposition 5.1 *Let $R_1, R_2, \ldots, R_p$ be i.i.d. according to the exponential distribution with mean $\sigma > 0$. Then we have $E(\max_{1\le i\le p} R_i) = \sigma g(p)$, where*

$$g(p) \stackrel{\text{def}}{=} -\Gamma'(1) + \frac{\Gamma'(p+1)}{\Gamma(p+1)} = \log(p+1) + C + 2 - \frac{1}{(p+1)\theta_p} \qquad (1 < \exists\theta_p < 2) \tag{5.2}$$

and $\Gamma(\cdot)$ denotes the gamma function.

PROOF: For simplicity we may assume $\sigma = 1$. The probability distribution function of $\max_{1\le i\le p} R_i$ is given by

$$\varphi(x) = P(\max_{1\le i\le p} R_i \le x) = \left(\int_0^x e^{-t}dt\right)^p = (1 - e^{-x})^p \qquad (\forall x \ge 0)$$

and,

$$E(\max_{1\le i\le p} R_i) = \int_0^\infty x\, d\varphi(x) = p\int_0^\infty xe^{-x}(1 - e^{-x})^{p-1}dx = -p\int_0^1 (1 - y)^{p-1}\log y\, dy = -p\,\frac{\partial}{\partial q}B(p, q)\Big|_{q=1}$$

where $B(\cdot,\cdot)$ denotes the beta function. Hence, we obtain (cf. [1])

$$E(\max_{1\le i\le p} R_i) = -\Gamma'(1) + \frac{\Gamma'(p+1)}{\Gamma(p+1)} = C + 2 + \log(p+1) - \frac{1}{2(p+1)} - \int_0^\infty \frac{2t}{t^2 + (p+1)^2} \cdot \frac{dt}{e^{2\pi t} - 1}$$

and the last term is estimated as

$$0 \le \int_0^\infty \frac{2t}{t^2 + (p+1)^2} \cdot \frac{dt}{e^{2\pi t} - 1} \le \frac{1}{\pi}\int_0^\infty \frac{dt}{t^2 + (p+1)^2} \le \frac{1}{2(p+1)}$$

These complete the proof. ∎

5.2 Upper estimate of the efficiency

In this subsection we give an upper estimate of the efficiency in the exponential case. We consider the case with infinite production rate ($\lambda = +\infty$) and estimate its parallel execution time $\tilde{T}_p$, which never exceeds T_p.

We define $\tilde{T}_p$ by

$$\tilde{T}_p \equiv \tilde{T}_p(R_1, \ldots, R_N) = \max_{1 \le n \le N} \tilde{Y}_n \tag{5.3}$$

where $\{\tilde{Y}_n\}_{n=1}^{\infty}$ as well as $\{\tilde{X}_n\}_{n=1}^{\infty}$ are defined by induction on n:

$$\begin{cases} \tilde{Y}_n &= \tilde{X}_n + R_n \qquad (n \ge 1) \\ \tilde{X}_n &= \begin{cases} \min\limits_{1 \le i_1 < \cdots < i_{p-1} \le n-1} \max\limits_{\substack{1 \le k \le n-1 \\ k \ne i_1, \ldots, i_{p-1}}} \tilde{Y}_k & (n > p) \\ 0 & (1 \le n \le p) \end{cases} \end{cases} \tag{5.4}$$

Note that (3.1) and (3.2) reduce to (5.3) and (5.4) when $U_n \equiv 0$ for all $n \ge 1$, i.e., when $\lambda = +\infty$. By induction on n, we obtain:

$$\tilde{X}_n \le X_n, \qquad \tilde{Y}_n \le Y_n, \qquad \tilde{T}_p \le T_p \qquad \text{a.s.} \tag{5.5}$$

Lemma 5.2 *For any $n = 1, 2, \ldots,$*

$$E(\tilde{T}_p \mid N = n) = \frac{1}{p} \cdot E(T_1 \mid N = n) + \frac{1}{\mu} \cdot g(n \wedge p) - \frac{n \wedge p}{\mu p}$$

PROOF: Owing to the Markov property and Proposition 5.1, we have $E(T_1 \mid N = n,\ \tilde{X}_{n-p+1} = t) = pt + p/\mu$ and $E(\tilde{T}_p \mid N = n,\ \tilde{X}_{n-p+1} = t) = t + g(p)/\mu$ for any $n \ge p$ and $t > 0$. Namely,

$$E(T_1 \mid N = n) = p \cdot E(\tilde{X}_{n-p+1} \mid N = n) + \frac{p}{\mu} \qquad (n \ge p)$$

$$E(\tilde{T}_p \mid N = n) = E(\tilde{X}_{n-p+1} \mid N = n) + \frac{1}{\mu} \cdot g(p) \qquad (n \ge p)$$

And for $1 \le n < p$, we have $E(T_1 \mid N = n) = n/\mu$ and $E(\tilde{T}_p \mid N = n) = g(n)/\mu$. Combining these results, we obtain

$$E(\tilde{T}_p \mid N = n) = E(\tilde{L} \mid N = n) + \frac{1}{\mu} \cdot g(n \wedge p) \qquad (\forall n \in \boldsymbol{N}) \tag{5.6}$$

$$E(T_1 \mid N = n) = p \cdot E(\tilde{L} \mid N = n) + \frac{n \wedge p}{\mu} \qquad (\forall n \in \boldsymbol{N}) \tag{5.7}$$

where we define:

$$\tilde{L} \stackrel{\text{def}}{=} 1_{N \ge p} \cdot \tilde{X}_{N-p+1} \tag{5.8}$$

Eliminating the term $E(\tilde{L} \mid N = n) < +\infty$ from the above equations, we obtain the desired results. In fact, since $\tilde{L} \le \tilde{X}_{N-p+1}$, we have $E(\tilde{L}) \le (\nu - p + 1)/\mu < +\infty$. And $P(N = n) = \frac{1}{\nu}(1 - \frac{1}{\nu})^{n-1} > 0$ for $\forall n \in \boldsymbol{N}$. Hence $E(\tilde{L} \mid N = n) < +\infty$. ∎

Lemma 5.3 *If $\nu \ge p$, we have*

$$E(\log(N \wedge p + 1)) \ge \log(p + 1) - 1 \tag{5.9}$$

$$E(g(N \wedge p)) \ge (1 - \frac{1}{p})\log(p + 1) - \frac{1}{p + 1} + C + 1 \tag{5.10}$$

PROOF: (5.9) follows directly from (5.1) as

$$\begin{aligned} E(\log(N \wedge p+1)) &= \sum_{n=1}^{p} \frac{1}{\nu}(1-\frac{1}{\nu})^{n-1}\log(n+1) + (1-\frac{1}{\nu})^{p}\log(p+1) \\ &= \frac{1}{\nu}\sum_{n=1}^{p}(1-\frac{1}{\nu})^{n-1}\log\frac{n+1}{p+1} + \log(p+1) \\ &\geq \frac{1}{p}\sum_{n=1}^{p}\log\frac{n}{p} + \log(p+1) \\ &\geq \int_{0}^{1}\log x\,dx + \log(p+1) \;=\; \log(p+1) - 1 \end{aligned}$$

By (5.1) and (5.2), we have

$$E(g(N \wedge p)) \geq E(\log(N \wedge p+1)) + C + 2 - \frac{1}{p+1} - \sum_{n=1}^{p}\frac{1}{\nu}(1-\frac{1}{\nu})^{(n-1)}\frac{1}{n+1}$$

where the last term is estimated as

$$0 \leq \sum_{n=1}^{p}\frac{1}{\nu}(1-\frac{1}{\nu})^{(n-1)}\frac{1}{n+1} \leq \frac{1}{p}\sum_{n=1}^{p}\frac{1}{n+1} \leq \frac{1}{p}\log(p+1)$$

Thus we obtain (5.10). ∎

Lemma 5.4 *If $\nu \geq p$, we have*

$$\frac{1}{\eta} \geq 1 + \frac{\rho p^2}{\lambda t_1}\left(\log(p+1) + C - \frac{1}{p}\log(p+1) - \frac{1}{p+1}\right) \tag{5.11}$$

PROOF: From (5.5) and Lemmas 5.2 and 5.3, we obtain

$$E(T_p) \geq E(\tilde{T}_p) \geq \frac{t_1}{p} + \frac{1}{\mu}\left\{(1-\frac{1}{p})\log(p+1) - \frac{1}{p+1} + C + 1\right\} - \frac{1}{\mu}$$

Multiply the both sides by p/t_1, we obtain (5.11). ∎

Lemma 5.5 *For any $n = 1, 2, \ldots$, we have*

$$E(T_p \mid N = n) \geq \frac{n-p+1}{\lambda} + \frac{g(n \wedge p)}{\mu} \tag{5.12}$$

PROOF: When $N = n \geq p$, we have $T_p \geq \max\{O_k + R_k \mid n-p+1 \leq k \leq n\} \geq O_{n-p+1} + \max\{R_k \mid n-p+1 \leq k \leq n\}$. Hence

$$E(T_p|N=n) \geq \frac{n-p+1}{\lambda} + \frac{g(p)}{\mu} \qquad (n \geq p)$$

Similarly, when $N = n < p$, we have $T_p \geq \max\{O_k + R_k \mid 1 \leq k \leq n\} \geq O_1 + \max\{R_k \mid 1 \leq k \leq n\}$. Hence

$$E(T_p|N=n) \geq \frac{1}{\lambda} + \frac{g(n)}{\mu} \qquad (1 \leq n < p)$$

Combining these results, we obtain (5.12). ∎

Theorem 5.2 *Let $\mathcal{T}$ be a family of divisible tasks of exponential type. Then the efficiency η is bounded from above as follows.*

(a) *For $1 < \nu < p$, we have $\eta \le 1/(\log p + C + 1/2)$.*

(b) *For $\nu \ge p$ and $\rho > 1$, we have*

$$\frac{1}{\eta} \ge 1 + \frac{\rho p^2}{\lambda t_1}\left(\log(p+1) + C - \frac{1}{p}\log(p+1) - \frac{1}{p+1}\right)$$

(c) *For $\nu \ge p$, $0 < \rho \le 1$, and $\lambda t_1 \ge w(p)$, we have*

$$\frac{1}{\eta} \ge 1 + \frac{p^2}{\lambda t_1}\left(\log(p+1) + C - 1 - \frac{1}{p}\log(p+1)\right)$$

(d) *For $\nu \ge p$, $0 < \rho \le 1$, and $\lambda t_1 < w(p)$, we have $\eta \le (\sqrt{5}-1)/2 = 0.6180\ldots$.*

Here we define $w(p)$ by

$$w(p) = p^2\left(\log(p+1) + C - \frac{1}{p}\log(p+1) - \frac{1}{p+1}\right)$$

PROOF: (a) For $1 < \nu < p$, let $q = \lfloor \nu \rfloor$. By (5.5), (5.6) and (5.2),

$$E(T_p \mid N = n) \ge E(\tilde{T}_p \mid N = n) \ge \frac{g(n \wedge p)}{\mu} \ge \frac{g(n \wedge q)}{\mu} \ge \frac{1}{\mu}\cdot\left(\log(n \wedge q + 1) + C + \frac{3}{2}\right)$$

Hence by (5.9),

$$E(T_p) \ge \frac{1}{\mu}\cdot\left(\log(q+1) + C + \frac{1}{2}\right) \qquad \therefore\ \frac{1}{\eta} \ge \frac{p}{\nu}\cdot\left(\log\nu + C + \frac{1}{2}\right) \ge \log p + C + \frac{1}{2}$$

since $\nu^{-1}\cdot(\log\nu + C + 1/2)$ is a monotonically decreasing function of $1 \le \nu < p$.

(b) This is already established in Lemma 5.4.

(c) From Lemma 5.5 and (5.10), we obtain

$$E(T_p) \ge \frac{\nu - p + 1}{\lambda} + \frac{1}{\mu}\cdot\left\{(1 - \frac{1}{p})\log(p+1) - \frac{1}{p+1} + C + 1\right\}$$

$$\therefore\ \frac{1}{\eta} \ge \frac{1}{\rho} + \frac{1}{\lambda t_1}\left\{-p^2 + p + \rho\cdot w(p)\right\}$$

where $w(p)$ is defined above. The right-hand side in the last inequality is a convex function of $0 < \rho \le 1$ and non-increasing at $\rho = 1$ since $\lambda t_1 \ge w(p)$. Hence it is non-increasing in $0 < \rho \le 1$ and we obtain the desired result by estimating its value at $\rho = 1$.

(d) Let $\nu \ge p$, $\lambda t_1 \le w(p)$, and $\omega = (\sqrt{5}-1)/2$. If $\rho \ge \omega$, by Lemma 5.4, we have $1/\eta \ge 1 + \rho \ge 1 + \omega = 1/\omega$. On the other hand, if $\rho < \omega$, we have $E(T_p \mid N = n) \ge E(O_N \mid N = n) = n/\lambda$. Hence $E(T_p) \ge \nu/\lambda$ and $1/\eta \ge 1/\rho > 1/\omega$. Thus we obtain $\eta \le \omega$ in either case. ∎

According to Theorems 5.1 and 5.2(b), we have for $\rho > 1$,

$$\begin{aligned} 1 + \frac{\rho p^2}{\lambda t_1}\left(\log(p+1) + C - \frac{1}{p}\log(p+1) - \frac{1}{p+1}\right) &\le \frac{1}{\eta} \\ &\le 1 + \frac{\rho p^2}{\lambda t_1}\cdot\left(\log(p+1) + C + 2 + c_1\cdot\frac{1+\rho^2}{\rho(\rho-1)^2}\cdot\frac{\log p}{p}\right) \end{aligned} \tag{5.13}$$

Hence we obtain the asymptotic behavior of the efficiency for $\rho > 1$,

$$\frac{1}{\eta} \simeq 1 + \frac{\rho p^2}{\lambda t_1} \log p \qquad \text{as} \qquad p \to \infty \tag{5.14}$$

This expression succinctly describes how the efficiency η depends on the number of consumers p, the production rate λ, and the overall task size t_1. In particular, the task size t_1 should be $\Omega(p^2 \log p/\lambda)$ as $p \to \infty$, in order to maintain a constant efficiency. Its principal factor is given by p^2 because of the reasonable cost in producing the subtasks as in (3.4).

Now, compare this result with that in the deterministic case. An extra factor of $\log p$ appears here, which stems from the load imbalance between the consumers due to the diversified subtask sizes. (In the deterministic case, all the subtasks have the same size.) Thus the scalability in the exponential case is poorer than that in the deterministic case by this factor.

According to (5.14), the asymptotic efficiency is improved by decreasing ρ to 1. However, for a finite p, the efficiency may be degraded by letting $\rho \to 1$ because of the growth of the non-leading term as can be seen in the right-most side of (5.13). On the other hand, by letting $\rho < 1$, the asymptotic efficiency can never be improved as shown in Theorem 5.2(c) and (d). These results are reasonable because ρ represents the ratio of the production rate to the overall consumption rate of the subtasks. More precisely, $\rho \simeq 1$ is preferable, since it indicates that the ability of the producer and the consumers are comparable and that neither will become a bottleneck. However, when the number of consumers (and also subtasks) is small, the statistical irregularity becomes prominent with some of the consumers idling even if $\rho \simeq 1$. So, we should assume a small margin in the producer's ability, i.e., $\rho > 1$.

Finally, we note a simple relation between the efficiency and the average number of subtasks ν. Given a task, we can specify the granularity of subtasks either by ρ or by ν. Since $\mu = \nu/t_1$ and $\rho = \lambda/\mu p$ as defined in Assumption 3.1, we obtain from (5.14),

$$\frac{1}{\eta} \simeq 1 + \frac{p}{\nu} \log p \qquad \text{i.e.,} \qquad \nu \simeq \frac{\eta}{1-\eta} \cdot p \log p \qquad \text{as} \quad p \to \infty \tag{5.15}$$

In particular, we should have $\nu = \Theta(p \log p)$ as $p \to \infty$, in order to maintain a constant efficiency. Note that (5.15) holds *only when* $\rho > 1$. For large ν with $\rho < 1$, it may not hold. So ρ is a more fundamental parameter than ν.

6 General Case

6.1 Additional assumption for the general case

In this section we investigate the efficiency of the on-demand load distribution (single-level load balancing) in a general case. However, in order to guarantee a reasonable efficiency, we need to make an additional assumption.

Remember that, once a subtask is entrusted to one consumer processor, it is never shared with other processors no matter how long its execution may last. So a subtask should be "steadily" executed, otherwise a reasonable efficiency will never be attained. To be more specific, we assume that the expectation of the remaining execution time (*life expectancy*) of a subtask should never blow up: for each $n = 1, 2, \ldots$,

$$E(R_n - x \mid R_n > x) \le \frac{1}{\mu}, \qquad E\{(R_n - x)^2 \mid R_n > x\} = O(\frac{1}{\mu^2}) \qquad \text{for } \forall x \ge 0$$

as $p \to \infty$ and $\mu \to 0$. The first inequality means that the life expectancy of a subtask, $E(R_n - x \mid R_n > x)$, should never exceed its initial life expectancy, $1/\mu$. The second expression means that the dispersion of the life expectancy of a subtask should remain the same order throughout. These will be shown to give a sufficient condition for a reasonable efficiency when the granularity of the subtask is properly tuned.

For example, in the deterministic case, the life expectancy decreases by the amount of elapsed time: we have $E(R_n - x \mid R_n > x) = 1/\mu - x$ and $E\{(R_n - x)^2 \mid R_n > x\} = (1/\mu - x)^2$ for $0 \le \forall x \le 1/\mu$. This is the "steadiest" case.

In the exponential case, the life expectancy remains constant: we have $E(R_n - x \mid R_n > x) \equiv 1/\mu$ and $E\{(R_n - x)^2 \mid R_n > x\} \equiv 2/\mu^2$ for $\forall x \ge 0$. This is the marginal case.

On the contrary, when the size of a subtask is distributed too divergently, this condition may be violated and the efficiency may be poor. For example, suppose that most of the subtasks are small (of size a) and only a few of them are exceptionally large (of size $A \gg a$). Then the few subtasks that have survived the initial small period (a) will have a much longer life expectancy ($A - a$) than the overall initial life expectancy ($\approx a$). They will incur a fatal load imbalance and the efficiency will be damaged. A remedy for such cases might be to subdivide the subtasks that have been found to be exceptionally large. Such a load balancing scheme is beyond the scope of discussion here.

Thus, in order to guarantee a reasonable efficiency, we should assume that the subtask sizes are not too diverse. The above condition in terms of life expectancy indirectly limits the diversity of the subtask size. However, for the sake of the applicability to the analysis in later sections, we will slightly relax this condition. Let $T = \{(N^{(t_1,\nu)}, \{R_n^{(t_1,\nu)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu,p)}\}_{n=1}^{\infty})\}$ be a family of divisible tasks. We assume the following throughout the rest of this paper.

Assumption 6.1 (moderate diversity in the subtask size) There exit constants, b and $b' > 0$, such that, for any $t_1 > \nu > 1$ with $\mu^{(t_1,\nu)} = \nu/t_1$,

$$E(R_n^{(t_1,\nu)} - x \mid R_n^{(t_1,\nu)} > x) \le \frac{1}{\mu^{(t_1,\nu)}} \left(1 + \frac{b}{1 + |\log \mu^{(t_1,\nu)}|}\right) \qquad \text{for } \forall x \ge 0,\ n = 1, 2, \ldots \tag{6.1}$$

$$E\{(R_n^{(t_1,\nu)} - x)^2 \mid R_n^{(t_1,\nu)} > x\} \le \frac{b'}{(\mu^{(t_1,\nu)})^2} \qquad \text{for } \forall x \ge 0,\ n = 1, 2, \ldots \tag{6.2}$$

In the theory of reliability engineering [13], several *aging* notions are defined, which are closely related to our assumption above. Let T be a nonnegative random variable, representing a lifetime of a device. $\lambda(t) = \lim_{\varepsilon \downarrow 0} P(t < T \le t + \varepsilon \mid T > t)/\varepsilon$ is called the *hazard* (or *failure*) *rate function*. T is said to have *increasing hazard rate* (IHR) or *increasing failure rate* (IFR) if $\lambda(t)$ is nondecreasing. In this case, $E(T - t \mid T > t)$ is nonincreasing in t, and T is said to have *decreasing mean residual life* (DMRL). Note that the first condition in the above assumption, (6.1), is implied by DMRL. Moreover, if $R_n^{(t_1,\nu)}$ has IHR and its coefficient of variation (ratio of the standard deviation to the mean) is bounded, (6.2) as well as (6.1) holds.

6.2 Lower estimate of the efficiency

In this subsection, we show that the efficiency in the general case on Assumption 6.1 is not worse than that in the exponential case when $\rho > 1$.

Let $T = \{(N^{(t_1,\nu)}, \{R_n^{(t_1,\nu)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu,p)}\}_{n=1}^{\infty})\}$ be a family of divisible tasks satisfying Assumption 6.1. O_n, X_n, Y_n, Z_n, T_p are those defined in (3.1)-(3.2).

In order to obtain an upper bound of the average parallel execution time t_p, we introduce a "synchronized" parallel execution model, in which the progress of execution is rather easy to estimate. Namely, we consider a "synchronized" parallel execution: all the consumers are synchronized so that either all of them are "busy" executing the subtasks or all of them stay "idle" with possible suspended subtasks. More precisely, we define the following by induction on n.

$$
\left\{
\begin{array}{l}
\bar{X}_n = O_p, \qquad \bar{Z}_n = 0, \qquad (1 \le \forall n \le p) \\
\bar{K}_{p,i} = i, \qquad \bar{D}_{p,i} = R_i \qquad (1 \le \forall i \le p) \\
\bar{J}_n = \min\{k \mid 1 \le k \le p,\ \bar{D}_{n,k} = \min\limits_{1 \le i \le p} \bar{D}_{n,i}\} \qquad (\forall n \ge p) \\
\bar{K}_{n+1,i} = \begin{cases} \bar{K}_{n,i} & (i \ne \bar{J}_n) \\ n+1 & (i = \bar{J}_n) \end{cases} \qquad \text{for } \forall n \ge p \text{ and } 1 \le \forall i \le p \\
\bar{Z}_{n+1} = \bar{X}_n + \bar{D}_{n,\bar{J}_n} \qquad (\forall n \ge p) \\
\bar{X}_{n+1} = O_{n+1} \vee \bar{Z}_{n+1} \qquad (\forall n \ge p) \\
\bar{D}_{n+1,i} = \begin{cases} \bar{D}_{n,i} - \bar{D}_{n,\bar{J}_n} & (i \ne \bar{J}_n) \\ R_{n+1} & (i = \bar{J}_n) \end{cases} \qquad \text{for } \forall n \ge p \text{ and } 1 \le \forall i \le p
\end{array}
\right. \tag{6.3}
$$

Each of these random variables represents the following.

$\bar{X}_n$: start of the execution period of the n th subtask

$\bar{Z}_n$: time when a consumer becomes ready to execute the n-th subtask

$\bar{K}_{n,i}$: order of the subtask being executed at the i-th consumer at time $\bar{X}_n + 0$

$\bar{D}_{n,i}$: remaining execution time of the $\bar{K}_{n,i}$-th subtask at time $\bar{X}_n + 0$

$\bar{J}_n$: order of the consumer that first completes the subtask resumed at time $\bar{X}_n$

Clearly, we have $0 \le Z_1 \le X_1 \le Z_2 \le X_2 \le \cdots \le Z_n \le X_n \le Z_{n+1} \le \cdots$. Each (Z_n, X_n) is a "idle" period and each (X_n, Z_{n+1}) a "busy" period. The next lemma shows that the progress in the synchronized parallel execution never goes ahead of the that in the parallel execution described in Definition 3.1(ii).

Lemma 6.1 (a) *For $\forall n \ge p$ and $1 \le \forall k \le n$, we have*

$$
Y_k \le \begin{cases} \bar{X}_n + \bar{D}_{n,i} & \text{if } 1 \le \exists i \le p \text{ such that } k = \bar{K}_{n,i} \\ X_n & \text{if } k \notin \{\bar{K}_{n,1}, \cdots, \bar{K}_{n,p}\}) \end{cases}
$$

(b) *For $\forall n \in \boldsymbol{N}$, we have $Z_n \le \bar{Z}_n$ and $X_n \le \bar{X}_n$.*

PROOF: (a) The claim is trivial for $n = p$, since $k = \bar{K}_{n,k}$ and $Y_k = O_k + R_k \le O_p + R_k = \bar{X}_p + \bar{D}_{p,k}$ for each $1 \le k \le p$. We will use induction on n. Assume that the claim is satisfied for some $n \ge p$. According to the value of $k = 1, 2, \ldots, n+1$, one of the following four cases occurs.

(i) When $k = \bar{K}_{n,i}$ and $i \ne \bar{J}_n$ for some $1 \le i \le p$, we have $\bar{K}_{n+1,i} = k$ and $\bar{X}_{n+1} + \bar{D}_{n+1,i} \ge \bar{Z}_{n+1} + \bar{D}_{n+1,i} = \bar{X}_n + \bar{D}_{n,i} \ge Y_k$.

(ii) When $k = \bar{K}_{n,\bar{J}_n}$, we have $k \notin \{\bar{K}_{n+1,1}, \cdots, \bar{K}_{n+1,p}\}$ and $\bar{X}_{n+1} \ge \bar{Z}_{n+1} = \bar{X}_n + \bar{D}_{n,\bar{J}_n} \ge Y_k$.

(iii) When $k = n+1$, we have $k = \bar{K}_{n+1,\bar{J}_n}$. By the definition of Z_n in (3.2) and by the induction hypothesis, we obtain

$$
Z_{n+1} \le \max\{Y_k \mid 1 \le k \le n \text{ with } k \notin \{\bar{K}_{n,1}, \cdots, \bar{K}_{n,p}\} \text{ or } k = \bar{K}_{n,\bar{J}_n}\} \le \bar{X}_n + \bar{D}_{n,\bar{J}_n} = \bar{Z}_{n+1}
$$

Hence, we have $X_{n+1} = O_{n+1} \vee Z_{n+1} \leq O_{n+1} \vee \bar{Z}_{n+1} = \bar{X}_{n+1}$ and $Y_{n+1} = X_{n+1} + R_{n+1} \leq \bar{X}_{n+1} + R_{n+1} = \bar{X}_{n+1} + \bar{D}_{n+1,J_n}$.

(iv) When $1 \leq k \leq n$ and $k \notin \{K_{n,1}, \cdots, K_{n,p}\}$, we have $k \notin \{\bar{K}_{n+1,1}, \cdots, \bar{K}_{n+1,p}\}$ and $Y_k \leq \bar{X}_n \leq \bar{X}_{n+1}$.

(b) For $1 \leq n \leq p$, the claim is trivial. For $n > p$, the claim is already shown in (iii) above. ∎

Owing to this lemma, we may concentrate on estimating the progress of the synchronized parallel execution. Now, let us express $\bar{X}_n$ and $\bar{Z}_n$ in terms of the lengths of the busy and idle periods. For $\forall n \geq p$, we define:

$$H_{n+1} = \bar{D}_{n,\bar{J}_n} = \bar{Z}_{n+1} - \bar{X}_n \tag{6.4}$$

$$I_{n+1} = (O_{n+1} - \bar{Z}_{n+1})_+ = \bar{X}_{n+1} - \bar{Z}_{n+1} \tag{6.5}$$

These represent the length of a busy period and an idle period respectively. Clearly, for $\forall n > p$, we have:

$$\bar{X}_n = O_p + \sum_{k=p+1}^{n} H_k + \sum_{k=p+1}^{n} I_k \tag{6.6}$$

$$\bar{Z}_n = O_p + \sum_{k=p+1}^{n} H_k + \sum_{k=p+1}^{n-1} I_k \tag{6.7}$$

Note that the length of an idle period does not exceed the generation time of a subtask: $I_n \leq U_n$. In fact, we have $\bar{Z}_n \geq \bar{X}_{n-1} \geq O_{n-1}$, and $I_n = (O_n - \bar{Z}_n)_+ \leq O_n - O_{n-1} = U_n$.

Now, we will proceed to estimate the lengths of the busy periods. For this purpose, we consider the parallel execution with an infinite production rate ($\lambda = +\infty$), in which no idle period occurs (cf. Section 5.2). More precisely, we define the following by induction on n.

$$\left\{\begin{array}{l} \tilde{X}_n = 0 \qquad (1 \leq \forall n \leq p) \\ \tilde{K}_{p,i} = i, \qquad \tilde{D}_{p,i} = R_i \qquad (1 \leq \forall i \leq p) \\ \tilde{J}_n = \min\{k \mid 1 \leq k \leq p, \ \tilde{D}_{n,k} = \min\limits_{1 \leq i \leq p} \tilde{D}_{n,i}\} \qquad (\forall n \geq p) \\ \tilde{K}_{n+1,i} = \left\{\begin{array}{ll} \tilde{K}_{n,i} & (i \neq \tilde{J}_n) \\ n+1 & (i = \tilde{J}_n) \end{array}\right. \qquad \text{for } \forall n \geq p \text{ and } 1 \leq \forall i \leq p \\ \tilde{X}_{n+1} = \tilde{X}_n + \tilde{D}_{n,\tilde{J}_n} \qquad (\forall n \geq p) \\ \tilde{D}_{n+1,i} = \left\{\begin{array}{ll} \tilde{D}_{n,i} - \tilde{D}_{n,\tilde{J}_n} & (i \neq \tilde{J}_n) \\ R_{n+1} & (i = \tilde{J}_n) \end{array}\right. \qquad \text{for } \forall n \geq p \text{ and } 1 \leq \forall i \leq p \end{array}\right. \tag{6.8}$$

Each of these random variables represents the following:

$\tilde{X}_n$: start of the execution period of the n-th subtask

$\tilde{K}_{n,i}$: order of the subtask being executed at the i-th consumer at time $\tilde{X}_n + 0$

$\tilde{D}_{n,i}$: remaining execution time of the $\tilde{K}_{n,i}$-th subtask at time $\tilde{X}_n + 0$

$\tilde{J}_n$: order of the consumer that first completes the subtask resumed at time $\tilde{X}_n$

Note that, when $\lambda = +\infty$, (6.3) reduces to (6.8) with $\tilde{Z}_n = 0$ $(1 \leq n \leq p)$ and $\tilde{Z}_n = \tilde{X}_n$ $(n > p)$. We also define $\tilde{Y}_n = \tilde{X}_n + R_n$, the end of the execution period of the n-th subtask, for each $n = 1, 2, \ldots$, and

$$\tilde{\mathcal{F}}_t = \bigwedge_{s>t} \bar{\sigma}[\tilde{X}_n \wedge s, \ \tilde{Y}_n \wedge s \mid n = 1, 2, \ldots] : \text{ information up to time } t \quad (\forall t \geq 0) \tag{6.9}$$

Note that $\{\tilde{\mathcal{F}}_t\}_{t\geq 0}$ is a complete and right-continuous increasing family of σ-algebras on $(\Omega, \mathcal{B}, P)$. For $\forall n \in \boldsymbol{N}$, each of $\tilde{X}_n$ and $\tilde{Y}_n$ is a stopping time (Markov time) with respect to $\{\tilde{\mathcal{F}}_t\}_{t\geq 0}$. Strictly speaking, $\tilde{\mathcal{F}}_t$ represents the *information available at the consumers up to time t*. In general, O_n is *not* a stopping time with respect to $\{\tilde{\mathcal{F}}_t\}_{t\geq 0}$.

Comparing (6.8) with (6.3), we have: $\tilde{K}_{n,i} = \bar{K}_{n,i}$, $\tilde{D}_{n,i} = \bar{D}_{n,i}$, $\tilde{J}_n = \bar{J}_n$, and $\tilde{D}_{n,\tilde{J}_n} = H_{n+1}$ for any $n \geq p$ and $1 \leq i \leq n$. Hence we have

$$\tilde{X}_n = \sum_{k=p+1}^{n} H_k \qquad (\forall n > p) \tag{6.10}$$

Now, define M_n, the *average load per consumer for the first* $(n-1)$ *subtasks*, by

$$M_n = \frac{1}{p}\sum_{k=1}^{n-1} R_k \tag{6.11}$$

In general, M_n is *not* a stopping time with respect to $\{\tilde{\mathcal{F}}_t\}_{t\geq 0}$. Note that the start of the execution period of the n-th subtask does not exceed M_n, namely,

$$\tilde{X}_n \leq M_n \ \text{ a.s.} \qquad (\forall n = 1, 2, \ldots) \tag{6.12}$$

In fact, for $n \leq p$, this is trivial. When $n > p$, all of the p consumers are busy executing the subtasks of order up to $n-1$ throughout time $\tilde{X}_n$, hence, $p\tilde{X}_n \leq R_1 + \cdots + R_{n-1}$.

Lemma 6.2 *For* $\forall n > p$, *we have*

$$E(M_n \quad \tilde{X}_n \mid \tilde{\mathcal{F}}_{\tilde{X}_n}) \leq \frac{p-1}{\mu p}\left(1 + \frac{b}{1+|\log \mu|}\right) \tag{6.13}$$

$$V(M_n - \tilde{X}_n \mid \tilde{\mathcal{F}}_{\tilde{X}_n}) \leq \frac{b'\rho^2}{\lambda^2}(p-1) \tag{6.14}$$

where $b, b' > 0$ *are the constants appeared in* (6.1) *and* (6.2).

PROOF: Let $n > p$ and assume that the information up to time $\tilde{X}_n$ is given, i.e., consider under the condition $\tilde{\mathcal{F}}_{\tilde{X}_n}$. Since the values of $\tilde{K}_{n-1,1}, \ldots, \tilde{K}_{n-1,p}$ and $\tilde{J}_{n-1}$ are uniquely determined by the information $\tilde{\mathcal{F}}_{\tilde{X}_n}$, we write $\{k_1, \ldots, k_{p-1}\} = \{\tilde{K}_{n-1,1}, \ldots, \tilde{K}_{n-1,p}\} \backslash \{\tilde{J}_{n-1}\}$ with $1 \leq k_1 < \cdots < k_{p-1} \leq n-1$. Let a_i denote the elapsed time in executing the k_i-th subtask up to the time $\tilde{X}_n$, which is also determined by $\tilde{\mathcal{F}}_{\tilde{X}_n}$, i.e., $a_i = \tilde{X}_n - \tilde{X}_{k_i}$. Then we have $p \cdot (M_n - \tilde{X}_n) = \sum_{i=1}^{p-1}(R_{k_i} - a_i)$, where $R_{k_1} - a_1, \ldots, R_{k_{p-1}} - a_{p-1}$ are independent and non-negative. Hence, by (6.1) and (6.2), we obtain:

$$p \cdot E(M_n - \tilde{X}_n \mid \tilde{\mathcal{F}}_{\tilde{X}_n}) = \sum_{i=1}^{p-1} E(R_{k_i} - a_i \mid R_{k_i} > a_i) \leq \frac{p-1}{\mu}\left(1 + \frac{b}{1+|\log \mu|}\right)$$

$$p^2 \cdot V(M_n - \tilde{X}_n \mid \tilde{\mathcal{F}}_{\tilde{X}_n}) = \sum_{i=1}^{p-1} V(R_{k_i} - a_i \mid R_{k_i} > a_i) \leq \frac{b'(p-1)}{\mu^2} = \frac{b'\rho^2 p^2 (p-1)}{\lambda^2} \ \blacksquare$$

Now, let us define, for $\forall n > p$,

$$L_n = \tilde{X}_n \wedge \left(M_n - \frac{b+1}{\mu}\right) \tag{6.15}$$

where $b > 0$ is the constant appeared in (6.1). In general, L_n is not a stopping time either.

Lemma 6.3 *For $\forall n > p$, we have*

$$E(M_n - L_n) \le \frac{\tilde{b}\rho p}{\lambda}, \qquad V(M_n - L_n) \le \frac{b'\rho^2(p-1)}{\lambda^2}$$

where $\tilde{b} = 2(b+1)$ and $b, b' > 0$ are the constants appeared in (6.1) and (6.2).

PROOF: By the definition of L_n and (6.12), we have

$$M_n - L_n = (M_n - \tilde{X}_n) \vee \frac{b+1}{\mu} \le M_n - \tilde{X}_n + \frac{b+1}{\mu}$$

And $E(M_n - \tilde{X}_n) \le (b+1)/\mu$ follows from Lemma 6.2. Hence, $E(M_n - L_n) \le 2(b+1)/\mu \le 2(b+1)\rho p/\lambda$, which establishes the first claim.

Let us define: $A_n = E(M_n - \tilde{X}_n \mid \tilde{\mathcal{F}}_{\tilde{X}_n})$. for each $n > p$. By (6.12) and Lemma 6.2, we obtain

$$0 \le A_n \le \frac{1+b}{\mu}, \qquad E[(M_n - \tilde{X}_n - A_n)^2 \mid \tilde{\mathcal{F}}_{\tilde{X}_n}] \le \frac{b'\rho^2(p-1)}{\lambda^2}$$

Observe that

$$\begin{aligned} \tilde{X}_n < M_n - \frac{b+1}{\mu} \quad &\Rightarrow \quad 0 < M_n - \frac{b+1}{\mu} - L_n \le M_n - A_n - \tilde{X}_n \\ \tilde{X}_n \ge M_n - \frac{b+1}{\mu} \quad &\Rightarrow \quad M_n - \frac{b+1}{\mu} - L_n = 0 \end{aligned}$$

Hence we obtain:

$$E\left\{ \left(M_n - \frac{b+1}{\mu} - L_n \right)^2 \,\middle|\, \tilde{\mathcal{F}}_{\tilde{X}_n} \right\} \le \frac{b'\rho^2(p-1)}{\lambda^2}$$

$$\therefore \; V(M_n - L_n) \le E\left\{ \left(M_n - \frac{b+1}{\mu} - L_n \right)^2 \right\} \le \frac{b'\rho^2(p-1)}{\lambda^2} \blacksquare$$

Lemma 6.4 *For $\forall n > p$,*

$$E(L_n) \ge \frac{\rho}{\lambda}(n - \tilde{b}p - 1), \qquad V(L_n) \le \frac{2b'\rho^2}{\lambda^2}(n + p - 2)$$

PROOF: Observe that:

$$E(M_n) = \frac{1}{p} \cdot \frac{n-1}{\mu} = \frac{\rho}{\lambda}(n-1), \qquad V(M_n) \le \frac{1}{p^2}\frac{b'}{\mu^2}(n-1) = \frac{b'\rho^2}{\lambda^2}(n-1)$$

From these and Lemma 6.3, we obtain the desired results. ∎

From Lemma 6.4, we obtain

$$E(O_n - L_n) \le \frac{n}{\lambda} - \frac{\rho}{\lambda}(n - \tilde{b}p - 1) = \frac{1-\rho}{\lambda}\left\{ n - \frac{\rho}{\rho - 1}(\tilde{b}p + 1) \right\} \le -\frac{\rho}{\lambda}\frac{1}{\rho - 1}(n - n_0)$$

$$V(O_n - L_n) = V(O_n) + V(L_n) \le \left(\alpha^2 + \frac{2b'\rho^2}{\lambda^2} \right) n + \frac{2b'\rho^2}{\lambda^2}(p-2)$$

where we define:

$$n_0 = \left\lceil \frac{\rho}{\rho - 1}(\tilde{b}p + 1) \right\rceil \tag{6.16}$$

Therefore, Chebyshev's inequality gives, for $\forall n > n_0$,

$$\begin{aligned} P(O_n - L_n \geq 0) &\leq \left(\frac{\lambda}{\rho-1}\right)^2 \cdot \frac{1}{(n-n_0)^2} \cdot \left\{ \left(\alpha^2 + \frac{2b'\rho^2}{\lambda^2}\right) n + \frac{2b'\rho^2}{\lambda^2}(p-2) \right\} \\ &\leq \frac{1}{(\rho-1)^2} \left\{ (\alpha^2\lambda^2 + 2b'\rho^2) \left(\frac{1}{n-n_0} + \frac{n_0}{(n-n_0)^2} \right) + \frac{2b'\rho^2(p-2)}{(n-n_0)^2} \right\} \end{aligned}$$

Thus we obtain:

$$(6.17) \quad \sum_{k=n_0+1}^{n} P(L_n < O_n) \leq \frac{1}{(\rho-1)^2} \left\{ (\alpha^2\lambda^2 + 2b'\rho^2) \left(1 + \log(n-n_0) + \frac{\pi^2}{6} n_0 \right) + \frac{\pi^2}{3} b'\rho^2(p-2) \right\}$$

Lemma 6.5 *For $\forall n \geq 1$,*

$$E(\bar{X}_n) \leq \frac{n_0}{\lambda} + \frac{\rho}{\lambda}(n-1) + \frac{\alpha}{(\rho-1)^2} \left\{ (\alpha^2\lambda^2 + 2b'\rho^2)(\log n + \frac{\pi^2}{6} n_0 + 1) + \frac{\pi^2}{3} b'\rho^2(p-2) \right\}$$

PROOF: For $n \leq p$, we have $\bar{X}_n = O_p$ and $E(\bar{X}_n) = p/\lambda \leq n_0/\lambda$; hence the claim is trivial. So we assume $n > p$. It follows from (6.6) that

$$E(\bar{X}_n) = E(O_p) + E\Big(\sum_{k=p+1}^{n} H_k \Big) + \sum_{k=p+1}^{n} E(I_k)$$

Here the first term in the right side is equal to p/λ. By (6.10), (6.12) and (6.11), the second term is bounded as

$$E\Big(\sum_{k=p+1}^{n} H_k \Big) = E(\bar{X}_n) \leq E(M_n) \leq \frac{n-1}{\mu p} = \frac{\rho(n-1)}{\lambda}$$

So we will concentrate our attention on the last term. For $p < \forall n \leq n_0$, we have $\sum_{k=p+1}^{n} E(I_k) \leq \sum_{k=p+1}^{n} E(U_k) = (n-p)/\lambda \leq (n_0-p)/\lambda$; hence the claim is trivial. So we assume that $n > n_0$. For $\forall k > n_0$, observe that

(i) $E(I_k) \leq \alpha P(I_k > 0)$, since $0 \leq I_n < U_n \leq \alpha$ a.s.

(ii) $I_k > 0 \Leftrightarrow \bar{Z}_k < O_k$ by (6.5).

(iii) $\bar{Z}_k \geq \bar{X}_k \geq L_k$ by (6.7), (6.10) and (6.15).

Hence we have $E(I_k) \leq \alpha P(L_k < O_k)$ for each $k > n_0$, and

$$\sum_{k=p+1}^{n} E(I_k) \leq \frac{n_0 - p}{\lambda} + \alpha \sum_{k=n_0+1}^{n} P(L_k < O_k) \qquad (\forall n > n_0)$$

where the last term is estimated in (6.17). Thus we obtain the desired result. ∎

Lemma 6.6

$$E(X_N) \leq \frac{\rho\nu}{\lambda} + \frac{n_0}{\lambda} + \frac{\alpha}{(\rho-1)^2} \left\{ (\alpha^2\lambda^2 + 2b'\rho^2) \left(\log\nu + \frac{\pi^2}{6} n_0 + 1 \right) + \frac{\pi^2}{3} b'\rho^2(p-2) \right\}$$

PROOF: This claim follows immediately from Lemmas 6.1(b) and 6.5.

Theorem 6.1 *Let T be a family of divisible tasks satisfying Assumptions 3.1 and 6.1. When $\rho > 1$, the efficiency η satisfies*

$$(6.18) \quad \frac{1}{\eta} \leq 1 + \frac{\rho p^2}{\lambda t_1} \left\{ \log(p+1) + \frac{c_0\rho^3}{(\rho-1)^3} \right\}$$

where $c_0 > 0$ is a constant depending only on a, a', a'', k, k' in (3.3)-(3.5) and b, b' in (6.1), (6.2).

PROOF: Let $n \in N$ and assume that the information up to time X_n is given, i.e., consider under the condition $\mathcal{F}_{X_n}$. Let $k_1, \ldots, k_q$ be the orders of the subtasks which are being executed at time $X_n + 0$, and $a_1, \ldots, a_q$ be the elapsed time in the respective execution up to time X_n, where $1 \le q \le p$. Then

$$\begin{aligned} E\left(\max_{1\le j\le n} Y_j \;\middle|\; \mathcal{F}_{X_n}\right) &= X_n + E\left\{\max_{1\le i\le q}(R_{k_i} - a_i) \;\middle|\; \mathcal{F}_{X_n}\right\} \\ &= X_n + E\left\{\max_{1\le i\le q}(R_{k_i} - a_i) \;\middle|\; R_{k_i} > a_i \;\; (1 \le \forall i \le q)\right\} \end{aligned}$$

Since, for each $1 \le i \le q$, we have:

$$E(R_{k_i} - a_i - a \mid R_{k_i} - a_i > a) \le \frac{1}{\mu}\left(1 + \frac{b}{1 + |\log \mu|}\right) \qquad (\forall a \ge 0)\,,$$

we obtain by the following Proposition 6.1,

$$E\left(\max_{1\le j\le n} Y_j \;\middle|\; \mathcal{F}_{X_n}\right) \le X_n + \frac{1}{\mu}\left(1 + \frac{b}{1 + |\log \mu|}\right) \cdot g(q)$$

Therefore,

$$E(T_p) = E\left(\max_{1\le j\le N} Y_j\right) \le E(X_N) + \frac{\rho p}{\lambda}\left(1 + \frac{b}{1 + |\log \mu|}\right)\{\log(p+1) + C + 2\}$$

where the first term in the rightmost side is estimated in Lemma 6.6. Hence

$$\begin{aligned} \frac{1}{\eta} \;\le\; & 1 + \frac{p^2}{\lambda t_1}\cdot\frac{n_0}{p} + \frac{p^2}{\lambda t_1}\cdot\frac{\alpha\lambda}{(\rho-1)^2}\cdot\left\{(\alpha^2\lambda^2 + 2b'\rho^2)\left(\frac{\log\nu}{p} + \frac{\pi^2}{6}\frac{n_0}{p} + \frac{1}{p}\right) + \frac{\pi^2}{3}b'\rho^2\right\} \\ & + \frac{\rho p^2}{\lambda t_1}\left(1 + \frac{b}{1 + |\log\mu|}\right)\{\log(p+1) + C + 2\} \end{aligned}$$

where we have $n_0/p \le (\bar{b}+1)\rho/(\rho-1) + 1$ due to (6.16); also $\alpha\lambda \le a''$, $\nu = \lambda t_1/\rho p \le t_1/p \le a p^{k-1}$, and $1/\mu = \rho p/\lambda > p$ due to Assumption 3.1. Thus we obtain the desired result. ∎

Proposition 6.1 *Let $\sigma > 0$ and $X_1, \ldots, X_p$ be positive i.i.d. such that $E(X_i - x \mid X_i > x) \le \sigma$ hold for any $x \ge 0$ and $1 \le i \le p$. Then we have $E(\max_{1\le i\le p} X_i) \le \sigma g(p)$, where $g(p)$ is the function defined by* (5.2).

PROOF: Let $Y_1, \ldots, Y_p$ be i.i.d. according to the exponential distribution with mean σ, and independent of $X_1, \ldots, X_p$. Since $E(Y_i - x \mid Y_i > x) = E(Y_i) = \sigma$ holds for any $x \ge 0$ and $1 \le i \le p$, it follows from the following Proposition 6.2 that

$$E(X_1 \vee \cdots \vee X_q \vee Y_{q+1} \vee \cdots \vee Y_p) \le E(X_1 \vee \cdots \vee X_{q-1} \vee Y_q \vee \cdots \vee Y_p) \qquad (1 \le \forall q \le p)$$

Hence $E(\max_{1\le i\le p} X_i) \le E(\max_{1\le i\le p} Y_i)$, and we obtain the desired result by Proposition 5.1. ∎

Proposition 6.2 *Let X, Y, Z be positive random variables such that Z is independent of (X, Y), $E(Z) < \infty$, and $E(X - x \mid X > x) \le E(Y - x \mid Y > x) < \infty$ for any $x \ge 0$. Then we have $E(X \vee Z) \le E(Y \vee Z)$.*

PROOF: Let $\varphi(t)$ be the distribution function of Z, i.e., $\varphi(t) = P(Z \le t)$. Then

$$E(X \vee Z) = \int_0^\infty \{t + E(X - t, X > t)\}d\varphi(t), \qquad E(Y \vee Z) = \int_0^\infty \{t + E(Y - t, Y > t)\}d\varphi(t)$$

So it is sufficient to show that $F(t) \le G(t)$ for any $t \ge 0$, where we define: $F(t) \equiv \log E(X - t, X > t)$ and $G(t) \equiv \log E(Y - t, Y > t)$. We have $E(X - t \mid X > t) = -1/F'(t)$ and $E(Y - t \mid Y > t) = -1/G'(t)$, since

$$E(X - x, X > x) = \int_x^\infty P(X > t)dt\,, \qquad E(X - x \mid X > x) = \frac{\displaystyle\int_x^\infty P(X > t)dt}{P(X > x)} \qquad (\forall x \ge 0)$$

Therefore the hypothesis implies that $F'(t) \le G'(t)$ for any $t \ge 0$. Hence $F(t) - G(t) \le F(0) - G(0) = \log E(X) - \log E(Y) \le 0$ for any $t \ge 0$, as desired. ∎

According to Theorem 6.1, we have for $\rho > 1$,

$$\frac{1}{\eta} \lesssim 1 + \frac{\rho p^2}{\lambda t_1} \log p \qquad \text{as} \qquad p \to \infty \tag{6.19}$$

This expression gives a succinct (pessimistic) estimate of the efficiency η, given the number of consumers p, the production rate λ, and the overall task size t_1. In particular, we can guarantee at least a constant efficiency when $t_1 = \Omega(p^2 \log p/\lambda)$ as $p \to \infty$. Its principal factor is given by p^2 because of the reasonable cost in producing the subtasks as in (3.4).

In the exponential case, this estimate (6.19) is critical, as (5.14) shows. On the other hand, in the deterministic case, it is pessimistic only by a factor of $\log p$, as (4.5) shows.

Finally, we note that the expected number of subtasks ν does not grow too fast when we increase the number of consumers p and the expected task size t_1 maintaining the constant efficiency η. In fact, (6.19) implies, for $\rho > 1$,

$$\frac{1}{\eta} \lesssim 1 + \frac{p}{\nu} \log p \qquad \text{i.e.,} \qquad \nu \lesssim \frac{\eta}{1 - \eta} \cdot p \log p \qquad \text{as} \quad p \to \infty \tag{6.20}$$

In particular, we can take $\nu = O(p \log p)$ as $p \to \infty$ while maintaining at least a constant efficiency (cf. Section 3.2).

7 Inheritance of the Moderate Diversity

In this section we investigate the distribution of the parallel execution time with the on-demand load distribution (single-level load balancing scheme). Assuming that the number of the subtasks, as well as the size of each subtask, satisfies the moderate diversity condition, we show that both of the task size and the parallel execution time satisfy the condition, too. In a word, the moderate diversity is inherited from the subtasks by the task size and the parallel execution time. These results will be used in the subsequent section dealing with the multi-level load balancing scheme, which hierarchically adopts the on-demand load distribution technique.

7.1 Assumption on the distribution of the number of subtasks

In this subsection, we introduce the moderate diversity condition for the number of the subtasks, similar to that for the size of each subtask in Section 6.1. Let $\mathcal{T} = \{(N^{(t_1,\nu)}, \{R_n^{(t_1,\nu)}\}_{n=1}^\infty, \{U_n^{(t_1,\nu,p)}\}_{n=1}^\infty)\}$ be a family of divisible tasks. We assume the following throughout the rest of this paper.

Assumption 7.1 (moderate diversity in the number of subtasks) There exit constants, b'' and $b''' > 0$, such that, for any $\nu > 1$,

$$E(N^{(t_1,\nu)} - n \mid N^{(t_1,\nu)} > n) \le \nu \left(1 + \frac{b''}{1 + \log \nu}\right) \qquad \text{for } \forall n = 0, 1, 2, \ldots \tag{7.1}$$

$$E\{(N^{(t_1,\nu)} - n)^2 \mid N^{(t_1,\nu)} > n\} \le b'''\nu^2 \qquad \text{for } \forall n = 0, 1, 2, \ldots \tag{7.2}$$

This condition for $N^{(t_1,\nu)}$ has precisely the continuous counterpart for $R_n^{(t_1,\nu)}$ in Assumption 6.1. So we will refer to either of these conditions as the *moderate diversity* condition whether it continuous or discrete.

Clearly, this assumption is satisfied either in the deterministic case or in the exponential case. In fact, we have $E(N - n \mid N > n) = \nu - n$ and $E\{(N - n)^2 \mid N > n\} = (\nu - n)^2$ in the former case; $E(N - n \mid N > n) \equiv \nu$ and $E\{(N - n)^2 \mid N > n\} \equiv \nu^2(2 - 1/\nu)$ in the latter case.

7.2 Moderate diversity of the task size

In this subsection we show that the entire task size inherits the moderate diversity from the subtasks.

Lemma 7.1 *Let $\{R_n\}_{n=1}^{\infty}$ be nonzero and nonnegative i.i.d. with*

$$E(R_n - x \mid R_n > x) \le v_1\,, \qquad E\{(R_n - x)^2 \mid R_n > x\} \le v_2 \qquad (\forall x \ge 0,\ \forall n \in \boldsymbol{N})$$

and N be a $\boldsymbol{N}$-valued random variable independent of $\{R_n\}_{n=1}^{\infty}$ with

$$E(N - n \mid N > n) \le s_1\,, \qquad E\{(N - n)^2 \mid N > n\} \le s_2 \qquad (\forall n = 0, 1, 2, \ldots)$$

Then $T_1 = \sum_{n=1}^{N} R_n$ satisfies

$$E(T_1 - x \mid T_1 > x) \le s_1 v_1\,, \qquad E\{(T_1 - x)^2 \mid T_1 > x\} \le s_2 v_1^2 + s_1 v_2 \qquad (\forall x \ge 0)$$

PROOF: Take an arbitrary $x \ge 0$ and define $K = \min\{k \in \boldsymbol{N} \mid \sum_{n=1}^{k} R_n > x\}$. For any $k \in \boldsymbol{N}$ and $\{r_n\}_{n=1}^{k-1} \in \boldsymbol{R}_+^{k-1}$ with $\sum_{n=1}^{k-1} r_n \le x$, we have

$$\begin{aligned}
& E\{T_1 - x \mid R_n = r_n \ \ (1 \le \forall n < k),\ \ K = k,\ \ T_1 > x\} \\
= \ & E\left\{ \sum_{n=1}^{k-1} r_n + R_k + \sum_{n=k+1}^{N} R_n - x \ \middle|\ R_n = r_n \ \ (1 \le \forall n < k),\ \ R_k > x - \sum_{n=1}^{k-1} r_n\,,\ \ N \ge k \right\} \\
= \ & E\left(R_k - x + \sum_{n=1}^{k-1} r_n \ \middle|\ R_k > x - \sum_{n=1}^{k-1} r_n \right) + E\left(\sum_{n=k+1}^{N} R_n \ \middle|\ N \ge k \right) \\
\le \ & v_1 + E\{\ (N - k)v_1 \ \mid \ N > k - 1\ \} \\
\le \ & s_1 v_1
\end{aligned}$$

Therefore, $E(T_1 - x \mid K = k,\ \ T_1 > x) \le s_1 v_1$ for any $k \in \boldsymbol{N}$ and $x \ge 0$. Since $T_1 > x$ implies $K < +\infty$, we obtain $E(T_1 - x \mid T_1 > x) \le s_1 v_1$, which establishes the first claim.

Similarly, for any k and $\{r_n\}_{n=1}^{k-1}$ with $\sum_{n=1}^{k-1} r_n \le x$, we have

$$\begin{aligned}
& E\{\ (T_1 - x)^2 \ \mid \ R_n = r_n \ \ (1 \le \forall n < k)\,,\ \ K = k\,,\ \ T_1 > x\ \} \\
= \ & E\left\{ \left(\sum_{n=1}^{k-1} r_n + R_k + \sum_{n=k+1}^{N} R_n - x \right)^2 \ \middle|\ R_n = r_n \ \ (1 \le \forall n < k)\,,\ \ R_k > x - \sum_{n=1}^{k-1} r_n\,,\ \ N \ge k \right\}
\end{aligned}$$

$$
\begin{aligned}
&= E\left\{ \left(R_k - x + \sum_{n=1}^{k-1} r_n \right)^2 \;\middle|\; R_k > x - \sum_{n=1}^{k-1} r_n \right\} + E\left\{ \left(\sum_{n=k+1}^{N} R_n \right)^2 \;\middle|\; N \geq k \right\} \\
&\quad + 2E\left(R_k - x + \sum_{n=1}^{k-1} r_n \;\middle|\; R_k > x - \sum_{n=1}^{k-1} r_n \right) \cdot E\left(\sum_{n=k+1}^{N} R_n \;\middle|\; N \geq k \right) \\
&\leq v_2 + E\{\, (N-k)v_2 + (N-k)(N-k-1)v_1^2 \mid N \geq k \,\} + 2v_1 \cdot E\{\, (N-k)v_1 \mid N \geq k \,\} \\
&= (v_2 - v_1^2) \cdot E(N - k + 1 \mid N > k-1) + v_1^2 \cdot E\{\, (N-k+1)^2 \mid N > k-1 \,\} \\
&\leq s_1 v_2 + s_2 v_1^2
\end{aligned}
$$

Therefore, $E\{(T_1 - x)^2 \mid K = k,\ T_1 > x\} \leq s_2 v_1^2 + s_1 v_2$ for any $k \in \boldsymbol{N}$ and $x \geq 0$. Hence, $E\{(T_1 - x)^2 \mid T_1 > x\} \leq s_2 v_1^2 + s_1 v_2$, which establishes the second claim. ∎

The next theorem shows that the task size satisfies the moderate diversity condition similar to (6.1) and (6.2).

Theorem 7.1 *Let $\mathcal{T}$ be a family of divisible tasks satisfying Assumptions 3.1, 6.1, and 7.1. For any $t_1 > 1$, the task size $T_1^{(t_1)}$ satisfies*

$$
E(T_1^{(t_1)} - x \mid T_1^{(t_1)} > x) \leq t_1 \cdot \left(1 + \frac{\hat{b}}{1 + \log t_1} \right) \qquad (\forall x \geq 0) \tag{7.3}
$$

$$
E\{(T_1^{(t_1)} - x)^2 \mid T_1^{(t_1)} > x\} \leq \hat{b}' t_1^2 \qquad (\forall x \geq 0) \tag{7.4}
$$

where $\hat{b}$ and $\hat{b}'$ are constants depending only on $a, a', a'', k, k', b, b', b'', b'''$ appeared in (3.3)-(3.5), (6.1), (6.2), (7.1) *and* (7.2).

PROOF: By Assumption 3.1(iii) and Lemma 7.1, we have, for any $t_1 > \nu > 1$ and $x \geq 0$,

$$
E(T_1 - x \mid T_1 > x) \leq \nu \left(1 + \frac{b''}{1 + \log \nu} \right) \cdot \frac{1}{\mu} \left(1 + \frac{b}{1 + |\log \mu|} \right)
$$

where $\mu = \nu / t_1$. Choosing $\nu = \sqrt{t_1}$, we establish the first claim. Similarly, for any $t_1 > \nu > 1$ and $x \geq 0$,

$$
E\{(T_1 - x)^2 \mid T_1 > x\} \leq b''' \nu^2 \cdot \frac{1}{\mu^2} \left(1 + \frac{b}{1 + |\log \mu|} \right)^2 + \nu \left(1 + \frac{b''}{1 + \log \nu} \right) \cdot \frac{b'}{\mu^2} \leq \{ b'''(1+b)^2 + (1+b'')b' \} t_1^2
$$

This establishes the last claim. ∎

7.3 Moderate diversity of the parallel execution time

In this subsection we show that the parallel execution time of the entire task inherits the moderate diversity from the subtasks when $\rho > 1$. For this purpose, we study the distribution of $T_p - t$, the *remaining execution time*, given that the execution is not completed up to time t, i.e., $T_p > t$. We assume $\rho > 1$ throughout this subsection.

Take an arbitrary $t > 0$ and assume that the information (more precisely, available at the consumers) up to time t is given and that $T_p > t$ holds. Namely, we consider under the condition $\mathcal{F}_t$ and $T_p > t$. P', E', and V' denote the conditional probability, expectation, and variation, respectively, under this condition. Given $\mathcal{F}_t$, we know the exact values of the following:

q_0 : number of the subtasks being executed at time t

$k_1 < \cdots < k_{q_0}$: orders of the subtasks being executed at time t

a_i : elapsed time in executing the k_i-th subtask up to time t $(1 \le i \le q_0)$

k_0 : the number of the subtasks that are already completed at time t

These concern the behavior of the consumers up to time t. On the other hand, we do not know the current state of the producer, in particular, the order of the subtask in which the producer is currently engaged. Hence it should be treated as a random variable: $S = \min\{s \in \boldsymbol{N} \mid O_s > t\}$.

Let us define $\tau_p : \boldsymbol{N} \times \boldsymbol{R}_+^\infty \times \boldsymbol{R}_+^\infty \longrightarrow \boldsymbol{R}_+$ by $\tau_p(n, \{r_i\}_{i=1}^\infty, \{u_i\}_{i=1}^\infty) = \max_{1 \le i \le n} y_i$, where y_i as well as o_i, x_i and z_i are determined by induction on i:

$$
(7.5) \qquad \left\{ \begin{array}{l} o_i = \displaystyle\sum_{j=1}^{i} u_j\,, \qquad x_i = o_i \vee z_i\,, \qquad y_i = x_i + r_i \qquad (i \ge 1) \\ z_i = \left\{ \begin{array}{ll} \displaystyle\min_{1 \le j_1 < \cdots < j_{p-1} \le i-1} \ \max_{\substack{1 \le h \le i-1 \\ h \ne j_1, \cdots j_{p-1}}} y_h & (i > p) \\ 0 & (1 \le i \le p) \end{array} \right. \end{array} \right.
$$

Then we have $T_p = \tau_p(N, \{R_n\}_{n=1}^\infty, \{U_n\}_{n=1}^\infty)$, according to (3.1) and (3.2) in Definition 3.1. Similarly, the remaining execution time is given by $T_p' \equiv T_p - t = \tau_p(N', \{R_n'\}_{n=1}^\infty, \{U_n'\}_{n=1}^\infty)$, where

$$
N' = N - k_0, \qquad R_n' = \left\{ \begin{array}{ll} R_{k_n} - a_n & (1 \le n \le q_0) \\ R_{n+k_0} & (n > q_0) \end{array} \right. , \qquad U_n' = \left\{ \begin{array}{ll} 0 & (1 \le n < S - k_0) \\ O_S - t & (n = S - k_0) \\ U_{n+k_0} & (n > S - k_0) \end{array} \right.
$$

Note that $T' \equiv (N', \{R_n'\}_{n=1}^\infty, \{U_n'\}_{n=1}^\infty)$ does *not* satisfy Assumption 3.1; in particular, each of $\{R_n'\}_{n=1}^\infty$ and $\{U_n'\}_{n=1}^\infty$ is i.i.d. just except for the first several members. However, T' admits a similar analysis as in Section 6.2. We give necessary modifications in the following.

We define $\Xi_p : (\boldsymbol{R}_+^\infty)^2 \longrightarrow (\boldsymbol{R}_+^\infty)^4$ by $\Xi_p(\{r_i\}, \{u_i\}) = (\{o_i\}, \{x_i\}, \{y_i\}, \{z_i\})$ with (7.5), and O_n', X_n', Y_n', Z_n' $(n = 1, 2, \ldots)$ by $\Xi_p(\{R_i'\}, \{U_i'\}) = (\{O_i'\}, \{X_i'\}, \{Y_i'\}, \{Z_i'\})$. Just as in the beginning of Section 6.2, we introduce the "synchronized" parallel execution of T', where the progress of execution is rather easy to estimate. More precisely, we define $\bar{X}_n'$, $\bar{Z}_n'$, $\bar{K}_{n,i}'$, $\bar{D}_{n,i}'$ and $\bar{J}_n'$ by induction on n:

$$
(7.6) \qquad \left\{ \begin{array}{l} \bar{X}_n' = O_p', \qquad \bar{Z}_n' = 0, \qquad (1 \le \forall n \le p) \\ \bar{K}_{p,i}' = i, \qquad \bar{D}_{p,i}' = R_i' \qquad (1 \le \forall i \le p) \\ \bar{J}_n' = \min\{k \mid 1 \le k \le p,\ \bar{D}_{n,k}' = \displaystyle\min_{1 \le i \le p} \bar{D}_{n,i}'\} \qquad (\forall n \ge p) \\ \bar{K}_{n+1,i}' = \left\{ \begin{array}{ll} \bar{K}_{n,i}' & (i \ne \bar{J}_n') \\ n+1 & (i = \bar{J}_n') \end{array} \right. \quad \text{for } \forall n \ge p \text{ and } 1 \le \forall i \le p \\ \bar{Z}_{n+1}' = \bar{X}_n' + \bar{D}_{n,\bar{J}_n'}' \qquad (\forall n \ge p) \\ \bar{X}_{n+1}' = O_{n+1}' \vee \bar{Z}_{n+1}' \qquad (\forall n \ge p) \\ \bar{D}_{n+1,i}' = \left\{ \begin{array}{ll} \bar{D}_{n,i}' - \bar{D}_{n,\bar{J}_n'}' & (i \ne \bar{J}_n') \\ R_{n+1}' & (i = \bar{J}_n') \end{array} \right. \quad \text{for } \forall n \ge p \text{ and } 1 \le \forall i \le p \end{array} \right.
$$

and

$$
H_{n+1}' = \bar{D}_{n,\bar{J}_n'}' - \bar{Z}_{n+1}' - \bar{X}_n', \qquad I_{n+1}' = (O_{n+1}' - \bar{Z}_{n+1}')_+ = \bar{X}_{n+1}' - \bar{Z}_{n+1}', \qquad M_n' = \frac{1}{p} \sum_{k=1}^{n-1} R_k'
$$

for each $n \geq p$. Then we have (cf. Lemma 6.1(b), (6.6), (6.7) with its subsequent remark),

$$X'_n \leq \bar{X}'_n, \qquad Z'_n \leq \bar{Z}'_n, \qquad I'_n \leq U'_n$$

$$\bar{X}'_n = O'_p + \sum_{k=p+1}^{n} H'_k + \sum_{k=p+1}^{n} I'_k, \qquad \bar{Z}'_n = O'_p + \sum_{k=p+1}^{n} H'_k + \sum_{k=p+1}^{n-1} I'_k$$

In order to estimate the busy period length, H'_k, we consider the parallel execution with an infinite production rate ($\lambda = +\infty$), in which no idle period occurs (cf. Section 6.2). More precisely, we define $\tilde{X}'_n, \tilde{\mathcal{F}}'_t$ and others by (6.8) and (6.9) with R'_n, U'_n instead of R_n, U_n ($n = 1, 2, \ldots$). Then we have, for each $n > p$,

$$\tilde{X}'_n = \sum_{k=p+1}^{n} H'_k \leq M'_n \ \text{ a.s.} \qquad \text{(cf. (6.10),(6.12))} \tag{7.7}$$

and besides (cf. Lemma 6.2)

$$E'(M'_n - \bar{X}'_n \mid \tilde{\mathcal{F}}'_{\bar{X}'_n}) \leq \frac{p-1}{\mu p}\left(1 + \frac{b}{1 + |\log \mu|}\right), \qquad V'(M'_n - \bar{X}'_n \mid \tilde{\mathcal{F}}'_{\bar{X}'_n}) \leq \frac{b'\rho^2}{\lambda^2}(p-1)$$

where $b, b' > 0$ are the constants appeared in (6.1) and (6.2). We also define, for each $n > p$,

$$L'_n = \tilde{X}'_n \wedge \left(M'_n - \frac{b+1}{\mu}\right) \tag{7.8}$$

Then we have, for each $n > p$,

$$E'(M'_n - L'_n) \leq \frac{\tilde{b}\rho p}{\lambda}, \qquad V'(M'_n - L'_n) \leq \frac{b'\rho^2(p-1)}{\lambda^2} \qquad \text{(cf. Lemma 6.3)}$$

where $\tilde{b} > 0$ is a constant defined by $\tilde{b} = 2(b+1)$. Since $E'(M'_n) \geq E(\frac{1}{p}\sum_{k=p+1}^{n-1} R_k) = \rho(n-p-1)/\lambda$, we have, for each $n > p$,

$$E'(L'_n) \geq \frac{\rho}{\lambda}(n - \tilde{b}'p - 1), \qquad V'(L'_n) \leq \frac{2b'\rho^2}{\lambda^2}(n+p-2) \qquad \text{(cf. Lemma 6.4)}$$

where $\tilde{b}' = \tilde{b} + 1 = 2b + 3$. Therefore, we have, for each $n > p$,

$$E'(O'_n - L'_n) \leq \alpha + \frac{n-1}{\lambda} - \frac{\rho}{\lambda}(n - \tilde{b}'p - 1) \leq -\frac{\rho - 1}{\lambda}(n - n_1)$$

$$V'(O'_n - L'_n) = V'(O'_n) + V'(L'_n) \leq \left(\alpha^2 + \frac{2b'\rho^2}{\lambda^2}\right) n + \frac{2b'\rho^2}{\lambda^2}(p-2)$$

where we define

$$n_1 = \left\lceil \frac{\rho(\tilde{b}'p + 1) + \alpha\lambda - 1}{\rho - 1} \right\rceil \tag{7.9}$$

By Chebyshev's inequality, we have, for each $n > n_1$,

$$\begin{aligned} P'(O'_n - L'_n \geq 0) &\leq \left(\frac{\lambda}{\rho - 1}\right)^2 \frac{1}{(n - n_1)^2} \cdot \left\{\left(\alpha^2 + \frac{2b'\rho^2}{\lambda^2}\right) n + \frac{2b'\rho^2}{\lambda^2}(p-2)\right\} \\ &\leq \frac{1}{(\rho-1)^2}\left\{(\alpha^2\lambda^2 + 2b'\rho^2)\left(\frac{1}{n - n_1} + \frac{n_1}{(n - n_1)^2}\right) + \frac{2b'\rho^2(p-2)}{(n-n_1)^2}\right\} \end{aligned}$$

Thus we obtain:

$$\sum_{k=n_1+1}^{n} P'(L'_n < O'_n) \leq \frac{1}{(\rho-1)^2}\left\{(\alpha^2\lambda^2 + 2b'\rho^2)\left(1 + \log(n - n_1) + \frac{\pi^2}{6}n_1\right) + \frac{\pi^2}{3}b'\rho^2(p-2)\right\} \tag{7.10}$$

Lemma 7.2 (cf. Lemma 6.5) *For each $n > q_0$,*

$$(7.11)\quad E'(\bar{X}'_n) \le \alpha + \frac{n_1}{\lambda} + \frac{\rho}{\lambda}(n-q_0) + \frac{\rho p}{\lambda}\left(1+\frac{b}{1+|\log\mu|}\right) + \frac{\alpha}{(\rho-1)^2}\left\{(\alpha^2\lambda^2+2b'\rho^2)\left(\log(n-q_0)+\frac{\pi^2}{6}n_1+1\right)+\frac{\pi^2}{3}b'\rho^2(p-2)\right\}$$

PROOF: For $q_0 < n \le p$, we have $\bar{X}'_n = O'_p$ and $E'(\bar{X}'_n) \le \alpha + (p-1)/\lambda \le \alpha + n_1/\lambda$; hence the claim is trivial. For $n > n_1$, we have

$$E'(\bar{X}'_n) = E'(O'_p) + E'\left(\sum_{k=p+1}^{n} H'_k\right) + \sum_{k=p+1}^{n_1} E'(I'_k) + \sum_{k=n_1+1}^{n} E'(I'_k)$$

Here the sum of the first and third terms is bounded by $\alpha + (n_1-1)/\lambda$, since $O'_p + \sum_{k=p+1}^{n_1} I'_k \le O'_p + \sum_{k=p+1}^{n_1} U'_k = O'_{n_1}$ a.s. Owing to (7.7), the second term is bounded as

$$E'\left(\sum_{k=p+1}^{n} H'_k\right) \le E'(M'_n) \le \frac{1}{\mu}\left(1+\frac{b}{1+|\log\mu|}\right) + \frac{n-p-1}{\mu p} < \frac{\rho(n-q_0)}{\lambda} + \frac{\rho p}{\lambda}\left(1+\frac{b}{1+|\log\mu|}\right)$$

So we should concentrate our attention on the last term. For each $k > n_1$, we have:

(i) $E'(I'_k) \le \alpha P'(I'_k > 0)$, since $0 \le I'_n \le U'_n \le \alpha$ a.s.

(ii) $I'_k > 0 \Leftrightarrow \bar{Z}'_k < O'_k$

(iii) $\bar{Z}'_k \ge \bar{X}'_k \ge L'_k$

These imply $E'(I'_k) \le \alpha P'(L'_k < O'_k)$. Therefore the last term above can be estimated using (7.10), and we obtain the desired result for $n > n_1$. The case with $p < n \le n_1$ can be treated similarly. ∎

Lemma 7.3 (cf. Theorem 6.1) *There exists a constant $c_0 > 1$ depending only on a, a', a'', k, k' in (3.3)-(3.5) and b, b', b'', b''' in (6.1), (6.2), (7.1), (7.2) such that:*

$$E'(T'_p) \le \frac{t_1}{p}\left(1+\frac{b''}{1+\log\nu}\right) + \frac{\rho p}{\lambda}\left(\log(p+1)+\frac{c_0\rho^3}{(\rho-1)^3}\right)\left(1+\frac{b}{1+|\log\mu|}\right)$$

PROOF: Note that N' is not less than q_0 by the definition. We first consider the case with $N' > q_0$. Proceeding as in the proof of Theorem 6.1, we have

$$E'(T'_p \mid N' > q_0) \le E'(\bar{X}'_{N'} \mid N' > q_0) + \frac{\rho p}{\lambda}\left(1+\frac{b}{1+|\log\mu|}\right)\{\log(p+1)+C+2\}$$

Substitute n by N' in (7.11) and take the expectation under the condition $N' > q_0$ (as well as $T_p \ge t$ and $\mathcal{F}_t$). Then we have

$$\begin{aligned} & E'(\bar{X}'_{N'} \mid N' > q_0) \\ \le\ & \alpha + \frac{n_1}{\lambda} + \frac{\rho}{\lambda}E'(N'-q_0 \mid N' > q_0) + \frac{\rho p}{\lambda}\left(1+\frac{b}{1+|\log\mu|}\right) \\ & + \frac{\alpha}{(\rho-1)^2}\left\{(\alpha^2\lambda^2+2b'\rho^2)\left(\log(E'(N'-q_0 \mid N' > q_0)+\frac{\pi^2}{6}n_1+1\right)+\frac{\pi^2}{3}b'\rho^2(p-2)\right\} \end{aligned}$$

where $E'(N'-q_0 \mid N' > q_0) = E(N-k_0-q_0 \mid N > k_0+q_0) \le \nu\{1+b''/(1+\log\nu)\}$ due to (7.1).

On the other hand, when $N' = q_0$, we have $T'_p = \max\{R'_1, \ldots, R'_{q_0}\}$. Hence, by Proposition 6.1,

$$E'(T'_p \mid N' = q_0) \leq \frac{\rho p}{\lambda}\left(1 + \frac{b}{1 + |\log \mu|}\right)\{\log(p+1) + C + 2\}$$

From these observations, irrespective of whether N' is greater than or equal to q_0, we obtain

$$\begin{aligned} E'(T'_p) \;\leq\; & \alpha + \frac{n_1}{\lambda} + \frac{\rho\nu}{\lambda}\left(1 + \frac{b''}{1+\log\nu}\right) \\ & + \frac{\alpha}{(\rho-1)^2}\left\{(\alpha^2\lambda^2 + 2b'\rho^2)\left(\log\nu + \frac{b''}{1+\log\nu} + \frac{\pi^2}{6}n_1 + 1\right) + \frac{\pi^2}{3}b'\rho^2(p-2)\right\} \\ & + \frac{\rho p}{\lambda}\left(1 + \frac{b}{1+|\log\mu|}\right)\{\log(p+1) + C + 3\} \\ \leq\; & \frac{t_1}{p}\left(1 + \frac{b''}{1+\log\nu}\right) + \frac{p}{\lambda}\left[\rho\left(1 + \frac{b}{1+|\log\mu|}\right)\{\log(p+1) + C + 3\} + \frac{n_1 + \alpha\lambda}{p}\right. \\ & \left. + \frac{\alpha\lambda}{(\rho-1)^2}\left\{(\alpha^2\lambda^2 + 2b'\rho^2)\left(\frac{\log\nu}{p} + \frac{\pi^2}{6}\cdot\frac{n_1}{p} + \frac{b''+1}{p}\right) + \frac{\pi^2}{3}b'\rho^2\right\}\right] \end{aligned}$$

where we have $(n_1 + \alpha\lambda)/p \leq (\bar{b}' + a'' + 1)\rho/(\rho-1) + 1$ and $\nu = \mu t_1 < t_1 \leq ap^k$ owing to (3.3), (3.5) and (7.9). Thus we obtain the desired result. ∎

Lemma 7.4

$$E'\{(T''_p)^2\} \leq 4\{(a'')^2 + b'\}\frac{b'''\nu^2 + 2(1+b'')p\nu + p^2}{\mu^2 p^2} + \frac{2b'}{\mu^2}\left[\{\log(p+1) + C + 2\}^2 + \frac{\pi^2}{6}\right]$$

PROOF: Take an arbitrary $n \in N$, and consider under the condition $\mathcal{F}'_{X'_n}$, information up to time X'_n. Let $k'_1, \ldots, k'_q$ be the orders of the subtasks which are being executed at time $X'_n + 0$, and $a'_1, \ldots, a'_q$ be the elapsed time in the respective execution up to time X'_n, where $1 \leq q \leq p$. Then we have $\max_{1\leq j\leq n} Y'_j = X'_n + \max_{1\leq i\leq q}(R'_{k'_i} - a'_i)$, and

$$E'\left\{\max_{1\leq j\leq n}(Y'_j)^2 \,\middle|\, \mathcal{F}'_{X'_n}\right\} \leq 2(X'_n)^2 + 2E'\left\{\max_{1\leq i\leq q}(R'_{k'_i} - a'_i)\}^2 \,\middle|\, R'_{k'_i} > a'_i \quad (1 \leq \forall i \leq q)\right\}$$

Since $E'\{(R'_{k'_i} - a'_i - a)^2 \mid R'_{k'_i} - a'_i > a\} \leq b'/\mu^2$ for any $a \geq 0$ and $1 \leq i \leq q$, we have, owing to the following Proposition 7.2,

$$E'\left\{\max_{1\leq i\leq q}(R'_{k'_i} - a'_i)^2 \,\middle|\, R'_{k'_i} > a'_i \quad (1 \leq \forall i \leq q)\right\} \leq \frac{b'}{\mu^2}h(q) \leq \frac{b'}{\mu^2}h(p)$$

where $h(p)$ is a function defined by (7.21).

When $n > p$, we have $X'_n \leq \bar{X}'_n = O'_p + \sum_{k=p+1}^{n} I'_k + \sum_{k=p+1}^{n} H'_k \leq O'_p + \sum_{k=p+1}^{n} U'_k + \bar{X}'_n \leq \alpha n + M'_n$, hence $E'\{(X'_n)^2\} \leq 2\alpha^2 n^2 + 2E'\{(M'_n)^2\}$, and

$$E'\{(M'_n)^2\} = \frac{n-1}{p^2}E'\left\{\sum_{k=1}^{n-1}(R'_k)^2\right\} \leq \frac{b'(n-1)^2}{\mu^2 p^2}$$

Therefore, since $T'_p = \max_{1\leq k\leq N'} Y'_k$, we obtain

$$E'\{(T'_p)^2 \mid N' > p\} \leq 4\left(\alpha^2 + \frac{b'}{\mu^2 p^2}\right)E'(N'^2 \mid N' > p) + \frac{2b'}{\mu^2}h(p)$$

where $E'(N'^2 \mid N' > p) = E\{(N - k_0)^2 \mid N > k_0 + p\} \leq b'''\nu^2 + 2(1+b'')p\nu + p^2$ by (7.1) and (7.2). On the other hand, for $q_0 \leq \forall n \leq p$, we have $X'_n \leq \bar{X}'_n \leq O'_p \leq \alpha p$ a.s., and similarly,

$E'\{(T'_p)^2 \mid N' = n\} \le 2\alpha^2 p^2 + \frac{2b'}{\mu^2} h(p)$. Therefore, irrespective of whether $N' > p$ or not, we obtain a common upper bound:

$$E\{(T'_p)^2\} \le 4\left(\alpha^2 + \frac{b'}{\mu^2 p^2}\right)\{b'''\nu^2 + 2(1+b'')p\nu + p^2\} + \frac{2b'}{\mu^2} h(p)$$

where $\alpha \le a''/\lambda \le a''/\mu p$. This completes the proof. ∎

Theorem 7.2 *Let $\mathcal{T}$ be a family of divisible tasks satisfying Assumptions 3.1, 6.1, and 7.1. When $\rho > 1$ and the task is as large as*

$$t_1 \ge \frac{\rho p^2}{\varepsilon\lambda}\left\{\log(p+1) + \frac{c^\circ \rho^3}{(\rho-1)^3}\right\} \tag{7.12}$$

for some $0 < \varepsilon < 1$, we have $1/\eta \le 1 + \varepsilon$ and

$$E(T_p - t \mid T_p > t) \le \frac{(1+\varepsilon)t_1}{p}\left\{1 + \frac{c}{1 + \log \frac{t_1}{p}}\right\} \qquad (\forall t \ge 0) \tag{7.13}$$

$$E((T_p - t)^2 \mid T_p > t) \le \frac{c' t_1^2}{p^2} \qquad (\forall t \ge 0) \tag{7.14}$$

where $c^\circ, c, c' > 1$ are constants depending only on $a, a', a'', k, k', b, b', b'', b'''$ appeared in (3.3)-(3.5), (7.15)-(7.16) and (7.1)-(7.2).

PROOF: Theorem 6.1 immediately gives the estimate of η. According to Lemmas 7.3 and 7.4,

$$E'(T'_p) \le \frac{t_1}{p}\left(1 + \frac{b''}{1+\log\nu}\right) + \frac{\varepsilon t_1}{p}\left(1 + \frac{b}{1 + |\log\mu|}\right)$$

$$E'\{(T'_p)^2\} \le 4\{(a'')^2 + b'\}\frac{b'''\nu^2 + 2(1+b'')p\nu + p^2}{\mu^2 p^2} + \frac{2b'}{\mu^2}\left[\{\log(p+1) + C + 2\}^2 + \frac{\pi^2}{6}\right]$$

where we have:

$$\nu = \frac{\lambda t_1}{\rho p} \ge \frac{p}{\varepsilon} > p, \quad \frac{1}{\mu} = \frac{\rho p}{\lambda} > p, \quad \frac{t_1}{p} \le a p^{k-1}, \quad \frac{1}{\mu}\log(p+1) = \frac{\rho p}{\lambda}\log(p+1) \le \frac{\varepsilon t_1}{p} < \frac{t_1}{p}$$

Therefore, we obtain (7.13) and

$$E'\{(T'_p)^2\} \le \frac{t_1^2}{p^2}\left[4\{(a'')^2 + b'\}(b''' + 2b'' + 3) + 2b'\left\{(C+3)^2 + \frac{\pi^2}{6}\right\}\right]$$

hence (7.14). ∎

Note that (7.12) implies $t_1/p \le E(T_p) \le (1+\varepsilon)t_1/p$ owing to Theorem 6.1. Hence T_p satisfies the moderate diversity condition in a slightly weakened sense. Namely, the expectation of the residual is bounded by the initial expectation multiplied by $(1+\varepsilon)$.

Now, assuming the moderate diversity in the slightly weakened sense from the beginning, we show that the similar results are still obtained.

Assumption 7.2 (weak moderate diversity in the subtask size) There exist constants, $\kappa \ge 1$ and $b, b' > 0$, such that, for any $t_1 > \nu > 1$ with $\mu^{(t_1,\nu)} = \nu/t_1$,

$$E(R_n^{(t_1,\nu)} - x \mid R_n^{(t_1,\nu)} > x) \le \frac{\kappa}{\mu^{(t_1,\nu)}}\left(1 + \frac{b}{1 + |\log\mu^{(t_1,\nu)}|}\right) \qquad \text{for } \forall x \ge 0,\ n = 1, 2, \ldots \tag{7.15}$$

$$E\{(R_n^{(t_1,\nu)} - x)^2 \mid R_n^{(t_1,\nu)} > x\} \le \frac{b'}{(\mu^{(t_1,\nu)})^2} \qquad \text{for } \forall x \ge 0,\ n = 1, 2, \ldots \tag{7.16}$$

Corollary 7.1 *Let $\mathcal{T}$ be a family of divisible tasks satisfying Assumptions 3.1, 6.1, and 7.2. When $\rho > 1$ and the task is as large as*

$$t_1 \geq \frac{\rho p^2}{\varepsilon\lambda}\left\{\log(p+1) + \frac{c^\circ\rho^3}{(\rho-1)^3}\right\}, \tag{7.17}$$

for some $0 < \varepsilon < 1$, we have:

$$\frac{1}{\eta} \leq 1 + \frac{\kappa\rho p^2}{\lambda t_1}\left\{\log(p+1) + \frac{c^\circ\rho^3}{(\rho-1)^3}\right\} \leq 1 + \kappa\varepsilon \tag{7.18}$$

$$E(T_p - t \mid T_p > t) \leq \frac{(1+\kappa\varepsilon)t_1}{p}\left(1 + \frac{c}{1+\log\frac{t_1}{p}}\right) \qquad (\forall t \geq 0) \tag{7.19}$$

$$E\{(T_p - t)^2 \mid T_p > t\} \leq \frac{c' t_1^2}{p^2} \qquad (\forall t \geq 0) \tag{7.20}$$

where $c^\circ, c, c' > 1$ are constants depending only on $a, a', a'', k, k', b, b', b'', b'''$ appeared in (3.3)-(3.5), (7.15)-(7.16) and (7.1)-(7.2).

PROOF: These are direct extension of Theorems 6.1 and 7.2, and can be proved similarly, where (6.15) should be modified into $L_n = \tilde{X}_n \wedge \{M_n - \kappa(b+1)/\mu\}$ and likewise for (7.8). ∎

Proposition 7.1 *Let $R_1, R_2, \ldots, R_p$ be i.i.d. according to the exponential distribution with mean $\sigma > 0$. Then we have $E\{\max_{1\leq i\leq p}(R_i)^2\} = \sigma^2 h(p)$, where*

$$\begin{aligned} h(p) &\stackrel{\text{def}}{=} (C+2)^2 + \frac{\pi^2}{6} + 2(C+2)\frac{\Gamma'(p+1)}{\Gamma(p+1)} - \frac{\Gamma''(p+1)}{\Gamma(p+1)} + 2\left(\frac{\Gamma'(p+1)}{\Gamma(p+1)}\right)^2 \\ &= \{\log(p+1) + C + 2\}^2 + \frac{\pi^2}{6} - \frac{\theta_p}{p+1}\log(p+1) - \frac{(C+2)\theta_p + 1}{p+1} + \frac{\theta_p'}{(p+1)^2} \end{aligned} \tag{7.21}$$

for some $1 < \theta_p < 2$ and $-3/4 < \theta_p' < 1/2$.

PROOF: For simplicity we may assume $\sigma = 1$. Let $\varphi(x)$ denote the probability distribution function of $\max_{1\leq i\leq p} R_i$. Then as in the proof of Proposition 5.1, we have

$$\begin{aligned} E\left\{\max_{1\leq i\leq p}(R_i)^2\right\} &= \int_0^\infty x^2\, d\varphi(x) = p\int_0^\infty x^2 e^{-x}(1-e^{-x})^{p-1}dx = p\int_0^1 (1-y)^{p-1}(\log y)^2\, dy \\ &- p\,\frac{\partial^2}{\partial q^2}B(p,q)\Big|_{q=1} = \Gamma''(1) - 2\Gamma'(1)\frac{\Gamma'(p+1)}{\Gamma(p+1)} - \frac{\Gamma''(p+1)}{\Gamma(p+1)} + 2\left(\frac{\Gamma'(p+1)}{\Gamma(p+1)}\right)^2 = h(p) \end{aligned}$$

where $B(\cdot,\cdot)$ denotes the beta function and $\Gamma(\cdot)$ the gamma function. Here we have $\Gamma'(1) = -C - 2$ and $\Gamma''(1) = (C+2)^2 + \pi^2/6$, where C denotes Euler's constant, and (cf. [1])

$$\frac{\Gamma'(x)}{\Gamma(x)} = \log x - \frac{1}{2x} - 2\int_0^\infty \frac{1}{x^2+t^2}\cdot\frac{t\,dt}{e^{2\pi t}-1}$$

$$\frac{\Gamma''(x)}{\Gamma(x)} - \left(\frac{\Gamma'(x)}{\Gamma(x)}\right)^2 = \frac{1}{x} + \frac{1}{2x^2} + 4\int_0^\infty \frac{x}{(x^2+t^2)^2}\cdot\frac{t\,dt}{e^{2\pi t}-1}$$

As shown in the proof of Proposition 5.1, we have

$$0 < \int_0^\infty \frac{1}{x^2+t^2}\cdot\frac{t\,dt}{e^{2\pi t}-1} < \frac{1}{2x}$$

and also

$$0 < \int_0^\infty \frac{x}{(x^2+t^2)^2} \cdot \frac{t\,dt}{e^{2\pi t}-1} < \frac{1}{2\pi}\int_0^\infty \frac{x}{(x^2+t^2)^2}dt = \frac{1}{2\pi x^2}B\left(\frac{3}{2},\frac{1}{2}\right) = \frac{1}{8x^2}$$

Hence, for some $1 < c_1 = c_1(x) < 2$ and $1 < c_2 = c_2(x) < 2$, we have

$$\frac{\Gamma'(x)}{\Gamma(x)} = \log x - \frac{c_1}{2x}, \qquad \frac{\Gamma''(x)}{\Gamma(x)} - \left(\frac{\Gamma'(x)}{\Gamma(x)}\right)^2 = \frac{1}{x} + \frac{c_2}{2x^2}$$

Substituting these into the above expression, we obtain the desired result. ∎

Proposition 7.2 *Let $\gamma > 0$ and $X_1, \dots, X_p$ be positive i.i.d. such that $E\{(X_i - x)^2 \mid X_i > x\} \le \gamma$ holds for any $x \ge 0$ and $1 \le i \le p$. Then we have $E\{\max_{1\le i\le p}(X_i)^2\} \le \gamma h(p)$, where $h(p)$ is the function defined by* (7.21).

PROOF: Let $Y_1, \dots, Y_p$ be i.i.d. according to the exponential distribution with mean $\sqrt{\gamma}$, and independent of $X_1, \dots, X_p$. Since $E(X_i - x \mid X_i > x) \le [E\{(X_i - x)^2 \mid X_i > x\}]^{1/2} \le \sqrt{\gamma}$ holds for any $x \ge 0$ and $1 \le i \le p$, it follows from the following Proposition 7.3 that

$$E\{(X_1 \vee \cdots \vee X_q \vee Y_{q+1} \vee \cdots \vee Y_p)^2\} \le E\{(X_1 \vee \cdots \vee X_{q-1} \vee Y_q \vee \cdots \vee Y_p)^2\} \qquad (1 \le \forall q \le p)$$

Hence $E\{\max_{1\le i\le p}(X_i)^2\} \le E\{\max_{1\le i\le p}(Y_i)^2\}$, and we obtain the desired result by Proposition 7.1. ∎

Proposition 7.3 *Let X, Y, Z be positive random variables such that Z is independent of (X, Y), $E(Z^2) < \infty$, and*

$$E(X - x \mid X > x) \le E(Y - x \mid Y > x) < \infty \qquad (\forall x \ge 0)$$

$$E\{(X - x)^2 \mid X > x\} \le E\{(Y - x)^2 \mid Y > x\} \le \gamma \qquad (\forall x \ge 0)$$

for some $\gamma > 0$. Then we have $E(X^2 \vee Z^2) \le E(Y^2 \vee Z^2)$.

PROOF: Let $\psi(t)$ be the distribution function of Z, i.e., $\psi(t) = P(Z \le t)$. Then we have

$$E(X^2 \vee Z^2) = \int_0^\infty [t^2 + 2tE(X - t, X > t) + E\{(X - t)^2, X > t\}]d\psi(t)$$

$$E(Y^2 \vee Z^2) = \int_0^\infty [t^2 + 2tE(Y - t, Y > t) + E\{(Y - t)^2, Y > t\}]d\psi(t)$$

where $E(X - t, X > t) \le E(Y - t, Y > t)$ as shown in the proof of Proposition 6.2. So it is sufficient to show that $F(t) \le G(t)$ for any $t \ge 0$, where we define: $F(t) \equiv E\{(X - t)^2, X > t\}$ and $G(t) \equiv E\{(Y - t)^2, Y > t\}$. Note that

$$F(t) = 2\int_t^\infty (s - t)P(X > s)ds, \qquad G(t) = 2\int_t^\infty (s - t)P(Y > s)ds$$

In fact, using the distribution function of X, $\varphi(t) = P(X \le t)$, we have

$$F(t) = \int_t^\infty d\varphi(s)\int_0^{s-t} 2u\,du = \int_0^\infty 2u\,du\int_{u+t}^\infty d\varphi(s) = \int_0^\infty 2uP(X > u + t)du.$$

And

$$F'(t) = -2\int_t^\infty P(X > s)ds, \qquad G'(t) = -2\int_t^\infty P(Y > s)ds,$$

$$F''(t) = 2P(X > t) \geq 0, \qquad G''(t) = 2P(Y > t) \geq 0,$$

hence

$$E\{(X-t)^2 \mid X > t\} = \frac{2F(t)}{F''(t)}, \qquad E\{(Y-t)^2 \mid Y > t\} = \frac{2G(t)}{G''(t)}.$$

Therefore the hypothesis implies $F(t)/F''(t) \leq G(t)/G''(t)$ for any $t \geq 0$; namely, $F'(t)G(t) - F(t)G'(t)$ is non-decreasing in $t \geq 0$. We also have $F'(t)G(t) - F(t)G'(t) \to 0$ as $t \to +\infty$, since $F'(+\infty) = G'(+\infty) = 0$ and $\sup_{t\geq 0} F(t)$, $\sup_{t\geq 0} G(t) < +\infty$. In fact,

$$F'(t) = -2\int_t^\infty ds \int_s^\infty d\varphi(u) = -2\int_t^\infty d\varphi(u) \int_t^u ds = -2\int_t^\infty (u-t)d\varphi(u)$$

$$\therefore \ |F'(t)| \leq 2\int_t^\infty u\, d\varphi(u) = 2E(X, X > t) \to 0 \qquad \text{as } t \to 0$$

and $0 \leq F(t) = E\{(X-t)^2 \mid X > t\} \cdot P(X > t) \leq \gamma$. Therefore we have $F'(t)G(t) - F(t)G'(t) \leq 0$ for any $t \geq 0$; namely, $\log F(t) - \log G(t)$ is non-increasing in $t \geq 0$. Thus we obtain $F(t) \leq G(t)$ for any $t \geq 0$, since $F(0) = E(X^2) \leq E(Y^2) = G(0)$ according to the hypothesis. ∎

8 Multi-Level Dynamic Load Balancing Scheme

In this section, we investigate the performance of the multi-level dynamic load balancing scheme described in Section 2.5, which is an iterative application of the single-level scheme in a hierarchical manner. The results on the latter scheme obtained so far in the previous sections will be iteratively applied.

8.1 Definitions of the Model

In this subsection, we introduce a formal model of the parallel execution with the multi-level load balancing scheme, which gives an expression for the parallel execution time. The model is constructed from that of the single-level scheme defined in Section 3, and hence covers similar factors, e.g., the "producer bottleneck" and the non-uniformness of the subtasks, but not the inter-processor communication latency. As before, we assume a family of problem spaces with different average sizes, from one of which a task (problem) is chosen at random. A task is assumed to be divisible into many independent subtasks at the first level, each of which is again assumed to be divisible into many independent subtasks at the second level, and so on. Besides, we assume that, at each level, the average subtask size can take any continuous values, i.e., a (sub)task admits any (fine or coarse) grained (sub)division as we like. However, in real cases, the available granularity may be restricted by the discrete nature of a given problem. Hence the results obtained for this model should be viewed as optimistic bounds for real cases. In order to generally guarantee positive results, we assume the moderate diversity condition at each level (cf. Section 6.1). More precisely, we give the following definitions.

Definition 8.1 We define $\mathcal{T}^{(t_1;\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)}$, a $(\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)$-*division* of a task of expected size t_1, by induction on ℓ.

(i) For $\ell = 0$, $\mathcal{T}^{(t_1)} = T_1^{(t_1)}$ is a *null division* of a task of expected size t_1, if and only if $T_1^{(t_1)}$ is a non-negative random variable with mean $t_1 > 1$, satisfying the moderate diversity condition:

$$E(T_1^{(t_1)} - x \mid T_1^{(t_1)} > x) \leq t_1\left(1 + \frac{b}{1+\log t_1}\right) \qquad \text{for } \forall x \geq 0,\ n = 1,2,\ldots \qquad \text{(cf. (7.3))} \tag{8.1}$$

(8.2) $E\{(T_1^{(t_1)} - x)^2 \mid T_1^{(t_1)} > x\} \leq b' t_1^2 \quad \text{for } \forall x \geq 0,\ n = 1,2,\ldots$ (cf. (7.4))

$T_1^{(t_1)}$ is also referred to as the *size* of $\mathcal{T}^{(t_1)}$ itself.

(ii) For $\ell \geq 1$, $\mathcal{T}^{(t_1;\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)} = (N^{(t_1,\nu_1)}, \{\mathcal{T}_n^{(t_1/\nu_1;\nu_2,\ldots,\nu_\ell;p_2,\ldots,p_\ell)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu_1,p_1)}\}_{n=1}^{\infty})$ is a $(\nu_1,\ldots,\nu_\ell; p_1,\ldots,p_\ell)$-*division of a task* of expected size t_1, if and only if the following hold.

1. For each n, $\mathcal{T}_n^{(t_1/\nu_1;\nu_2,\ldots,\nu_\ell;p_2,\ldots,p_\ell)}$ is a $(\nu_2,\ldots,\nu_\ell;p_2,\ldots,p_\ell)$- (null, when $\ell = 1$) division of a task of expected size t_1/ν_1.
2. $\mathcal{T}_n^{(t_1/\nu_1;\nu_2,\ldots,\nu_\ell;p_2,\ldots,p_\ell)}$ are probabilistically equivalent for all n. Namely, each of them consists of independent random variables whose distribution does not depend on n.
3. $(N^{(t_1,\nu_1)}, \{T_{1,n}^{(t_1/\nu_1)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu_1,p_1)}\}_{n=1}^{\infty})$ is a division of a task (of expected size t_1) between p_1 consumers in granularity t_1/ν_1, satisfying Assumptions 3.1, 6.1 and 7.1, where $T_{1,n}^{(t_1/\nu_1)}$ is the size of $\mathcal{T}_n^{(t_1/\nu_1;\nu_2,\ldots,\nu_\ell;p_2,\ldots,p_\ell)}$.

In particular, the *size* of $\mathcal{T}^{(t_1;\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)}$ is defined by $T_1^{(t_1)} = T_{1,1}^{(t_1/\nu_1)} + \cdots + T_{1,N^{(t_1,\nu_1)}}^{(t_1/\nu_1)}$, which does not depend on ν_1. We assume that the constants $a, a', a'', k, k', b, b', b'', b'''$ appeared in (3.3)-(3.5), (6.1)-(6.2), (7.1)-(7.2) and (8.1)-(8.2) are chosen large enough so that they are the same for all the level from 0 to ℓ.

Note that a (ν_1, p_1)-division of a task of expected size t_1 is nothing but a division of the task (of expected size t_1) between p_1 consumers in granularity t_1/ν_1 defined in Section 3.2; i.e., $\mathcal{T}^{(t_1;\nu_1;p_1)} = \mathcal{T}^{(t_1,\nu_1,p_1)}$. For each $t_1 > 1$,

$$\mathcal{T}^{(t_1)} = \{\mathcal{T}^{(t_1;\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)} \mid \text{a division of a task with } \nu_1,\ldots,\nu_\ell > 1,\ 1 < p_1,\ldots,p_\ell \in N\}$$

represents an *ℓ-level divisible task* of expected size t_1. We will refer to

$$\mathcal{T} = \{\mathcal{T}^{(t_1;\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)} \mid \text{a division of a task with } t_1 > 1,\ \nu_1,\ldots,\nu_\ell > 1,\ 1 < p_1,\ldots,p_\ell \in N\}$$

as a *family of ℓ-level divisible tasks*.

Definition 8.2 Let $\mathcal{T}^{(t_1;\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)}$ be a $(\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)$-division of a task of expected size t_1. We define $T_{(p_1,\ldots,p_\ell)}^{(t_1;\nu_1,\ldots,\nu_\ell)}$ by induction on ℓ, the *parallel execution time* of $\mathcal{T}^{(t_1;\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)}$ by the ℓ-level dynamic load balancing scheme, more precisely, the *parallel execution time* of a task of expected size t_1 by the ℓ-level dynamic load balancing scheme with $(\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)$-division.

(i) For $\ell = 0$, we define it as the task size itself, $T_1^{(t_1)}$.

(ii) For $\ell \geq 1$, $T_{(p_1,\ldots,p_\ell)}^{(t_1;\nu_1,\ldots,\nu_\ell)} = \tau_{p_1}(N^{(t_1,\nu_1)}, \{T_{(p_2,\ldots,p_\ell),n}^{(t_1/\nu_1;\nu_2,\ldots,\nu_\ell)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu_1,p_1)}\}_{n=1}^{\infty})$, where $\tau_{p_1}(\cdot,\cdot,\cdot)$ is the function defined in Section 7.3, and each $T_{(p_2,\ldots,p_\ell),n}^{(t_1/\nu_1;\nu_2,\ldots,\nu_\ell)}$ is the parallel execution time of $\mathcal{T}_n^{(t_1/\nu_1;\nu_2,\ldots,\nu_\ell;p_2,\ldots,p_\ell)}$ by the $(\ell-1)$-level dynamic load balancing scheme. Here p_1 is referred to as the *degree of the root producer*.

This definition indicates that the root producer at the top level, $p_1 \cdots p_{j-1}$ subproducers at j-th level for each $1 < j < \ell$, and $p_1 \cdots p_\ell$ consumers at the bottom level constitute a tree of processors. We will refer to $(p_1,\ldots,p_\ell)$ as the *processor configuration*.

For brevity, we often write $T_{(p_1,\ldots,p_\ell)}^{(t_1;\nu_1,\ldots,\nu_\ell)} = T_p^{(\ell)}$ with $p = p_1 \cdots p_\ell$, and refer to it as the *parallel execution time by the ℓ-level dynamic load balancing scheme with p consumers*, implicitly assuming an

expected size $t_1 > 1$ and a $(p_1, \ldots, p_\ell; \nu_1, \ldots, \nu_\ell)$-division. We also define the *expected parallel execution time:* $t_p^{(\ell)} = E(T_p^{(\ell)})$, the *expected speed-up:* $s_p^{(\ell)} = t_1/t_p^{(\ell)}$, and the *expected efficiency:* $\eta = t_1/pt_p^{(\ell)}$.

By definition, a multi-level divisible task consists of a number of subtasks, each of which again consists of a number of subtasks, and so on. We will also refer to these descendants as *subtasks*. More precisely, we give the following definition.

Definition 8.3 Let $\mathcal{T}^{(t_1;\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)}$ be a $(\nu_1, \ldots, \nu_\ell; p_1, \ldots, p_\ell)$-division of a task of expected size t_1. We write $t_1^{(j)} = t_1/(\nu_1 \cdots \nu_{j-1})$ for $1 \le j \le \ell + 1$, and define a *subtask at the j-th level* by induction on j.

(i) For $j = 0$, we define the subtask at the 0-th level by the task itself: $\mathcal{T}^{(t_1;\nu_1,\ldots,\nu_\ell;p_1,\ldots,p_\ell)}$.

(ii) For $1 \le j \le \ell$, we write, for each $n_1, \ldots, n_{j-1}$,

$$\mathcal{T}^{(t_1^{(j)};\nu_j,\ldots,\nu_\ell;p_j,\ldots,p_\ell)}_{n_1,\ldots,n_{j-1}} = (N^{(t_1^{(j)},\nu_j)}_{n_1,\ldots,n_{j-1}}, \{\mathcal{T}^{(t_1^{(j+1)};\nu_{j+1},\ldots,\nu_\ell;p_{j+1},\ldots,p_\ell)}_{n_1,\ldots,n_j}\}_{n_j=1}^{\infty}, \{U^{(t_1^{(j)},\nu_j,p_j)}_{n_1,\ldots,n_j}\}_{n_j=1}^{\infty})$$

and refer to each $\mathcal{T}^{(t_1^{(j+1)};\nu_{j+1},\ldots,\nu_\ell;p_{j+1},\ldots,p_\ell)}_{n_1,\ldots,n_j}$ as the $(n_1, \ldots, n_j)$-*subtask* at the j-th lev elIn particular, we refer to the ℓ-th level as the *leaf level*.

Note that each subtask at the j-th level is itself a $(\ell - j)$-level divisible task, more precisely, a $(\nu_{j+1}, \ldots, \nu_\ell; p_{j+1}, \ldots, p_\ell)$-division of a task of expected size $t_1^{(j)}$. Therefore its notation is consistent with Definition 8.1. Moreover, since $(n_1, \ldots, n_j)$-subtask at the j-th level is probabilistically equivalent to each other for all $n_1, \ldots, n_j$ (cf. Definition 8.1 (ii) 2.), we will often write $\mathcal{T}^{(t_1^{(j+1)};\nu_{j+1},\ldots,\nu_\ell;p_{j+1},\ldots,p_\ell)}$ and similarly $U^{(t_1^{(j)},\nu_j,p_j)}$. We will constantly use the following notations:

$$\begin{aligned}
\lambda_j = \lambda_j^{(t_1^{(j)},\nu_j,p_j)} = 1/E(U^{(t_1^{(j)},\nu_j,p_j)}) &\quad : \quad \textit{production rate} \text{ of a producer at the } j\text{-th level} \\
\mu_j = 1/E(T^{(t_1^{(j+1)};\nu_{j+1},\ldots,\nu_\ell)}_{(p_{j+1},\ldots,p_\ell)}) &\quad : \quad \textit{consumption rate} \text{ of a "consumer" at the } j\text{-th level} \\
\rho_j = \lambda_j/\mu_j p_j &\quad : \quad \textit{ratio of production/consumption rate} \text{ at the } j\text{-th level} \\
\eta_j = \mu_{j-1}\nu_j/\mu_j p_j &\quad : \quad \textit{efficiency} \text{ at the } j\text{-th level}
\end{aligned}$$

where $T^{(t_1^{(j+1)};\nu_{j+1},\ldots,\nu_\ell)}_{(p_{j+1},\ldots,p_\ell)}$ is the parallel execution time of $\mathcal{T}^{(t_1^{(j+1)};\nu_{j+1},\ldots,\nu_\ell;p_{j+1},\ldots,p_\ell)}$ by the $(\ell - j)$-level dynamic load balancing scheme. Note that ρ_j and η_j are consistent with the definitions in Section 3.3. In fact, for any fixed $n_1, \ldots, n_{j-1}$, regard $\mathcal{T}^{(t_1^{(j)};\nu_j,\ldots,\nu_\ell;p_j,\ldots,p_\ell)}_{n_1,\ldots,n_{j-1}}$ as a "task", which consists of "subtasks": $\mathcal{T}^{(t_1^{(j+1)};\nu_{j+1},\ldots,\nu_\ell;p_{j+1},\ldots,p_\ell)}_{n_1,\ldots,n_j}$ with $n_j = 1, 2, \ldots$. And consider the parallel execution by the single-level dynamic load balancing scheme with p_j "consumers", each of which is equivalent to the $p_{j+1} \cdots p_\ell$ consumers with the $(\ell - j)$-level dynamic load balancing scheme. Then the mean "subtask" size is $1/\mu_j$, the mean "task" size is ν_j/μ_j, the mean parallel execution time with p_j "consumers" is $1/\mu_{j-1}$. Hence we obtain the above expressions for ρ_j and η_j. Finally, note that:

$$(8.3) \qquad t_p^{(\ell)} = \frac{1}{\mu_0}, \qquad t_1 = \frac{\nu_1 \cdots \nu_\ell}{\mu_\ell}, \qquad \eta = \eta_1 \ldots \eta_\ell, \qquad t_1^{(j)} = \frac{\nu_j \cdots \nu_\ell}{\mu_\ell} \quad (1 \le j \le \ell + 1)$$

8.2 Isoefficiency Analysis

In this section, we show that the scalability of the multi-level dynamic load balancing scheme is improved by increasing the number of levels, in particular, better than that of the single-level scheme. For this purpose, we investigate the *isoefficiency function* [9] -- how much the task size should be

increased with the number of consumer processors so as to maintain a constant efficiency. However, as one can see in (8.3), in order to achieve a good efficiency η, we should properly tune the processor configuration and the granularity of the subtasks at each level so as to have η_j large for all j. We will tune these from the leaves to the root.

Let $\mathcal{T}$ be a family of ℓ-level divisible tasks with $1 \le \ell \in \boldsymbol{N}$ and $k > \ell + 1$, where k is the constant appeared in (3.3). In order to avoid being annoyed with pathological cases, we assume that, at each level j, the production rate, $\lambda_j^{(t_1,\nu,p)}$, is monotone decreasing with t_1, ν and p; and rather smoothly depends on them, i.e.,

$$\frac{\lambda_j^{(t_1',\nu',p')}}{\lambda_j^{(t_1,\nu,p)}} \to 1 \quad \text{as} \quad t_1,\ \nu,\ p \to +\infty \quad \text{with} \quad \frac{t_1'}{t_1},\ \frac{\nu'}{\nu},\ \frac{p'}{p} \to 1 \tag{8.4}$$

Take $0 < \varepsilon < 1$ and $\rho_* > 1$ arbitrarily. For any $t_1^{(\ell+1)} \equiv 1/\mu_\ell > 2\rho_* a'(\log 2)^{k'}$, the average subtask size at the leaf level, we define $\nu_j, p_j, t_1^{(j)}, \mu_{j-1}$ for each $1 \le j \le \ell$ by induction on j according to the next lemma. Here a' and k' are the constants appeared in (3.4).

Lemma 8.1 *Let $1 \le j \le \ell$ and assume that $\nu_i, p_i, t_1^{(i)}, \mu_{i-1}$ for each $i = j+1, j+2, \ldots, \ell$ have been chosen according to this lemma with larger j. If $t_1^{(\ell+1)}$ is large enough, there exist constants, $c_j^\circ, c_j, c_j' > 1$ such that the following hold. Take the largest $1 < p_j \in \boldsymbol{N}$ satisfying*

$$\rho_j \equiv \frac{\lambda_j}{\mu_j p_j} \ge \rho_* \quad \text{where} \quad \lambda_j \equiv \lambda_j^{(t_1^{(j)},\nu_j,p_j)}, \quad t_1^{(j)} = \nu_j t_1^{(j+1)}, \quad \nu_j = \frac{p_j}{\varepsilon}\{\log(p_j+1) + c_j^\circ\} \tag{8.5}$$

and define $\mu_{j-1} = 1/E(T_{(p_j,\ldots,p_\ell)}^{(t_1^{(j)};\nu_j,\ldots,\nu_\ell)})$. Then we have $\eta_j \equiv \mu_{j-1}\nu_j/\mu_j p_j \ge (1-\varepsilon)/(1-\varepsilon^{\ell-j+2})$ and

$$E\left(T_{(p_j,\ldots,p_\ell)}^{(t_1^{(j)};\nu_j,\ldots,\nu_\ell)} - x \;\middle|\; T_{(p_j,\ldots,p_\ell)}^{(t_1^{(j)};\nu_j,\ldots,\nu_\ell)} > x\right) \le \frac{1-\varepsilon^{\ell-j+2}}{(1-\varepsilon)\mu_{j-1}}\left(1 + \frac{c_j}{1+|\log\mu_{j-1}|}\right) \quad (\forall x \ge 0) \tag{8.6}$$

$$E\left\{\left(T_{(p_j,\ldots,p_\ell)}^{(t_1^{(j)};\nu_j,\ldots,\nu_\ell)} - x\right)^2 \;\middle|\; T_{(p_j,\ldots,p_\ell)}^{(t_1^{(j)};\nu_j,\ldots,\nu_\ell)} > x\right\} \le \frac{c_j'}{\mu_{j-1}^2} \qquad (\forall x \ge 0) \tag{8.7}$$

Moreover, c_j°, c_j, c_j' depend only on ℓ, ρ_* and $a, a', a'', k, k', b, b', b'', b'''$ appeared in (3.3)-(3.5), (7.15)-(7.16) and (7.1)-(7.2). Finally, note that

$$\frac{\rho_{j-1}p_{j-1}}{\lambda_{j-1}} = \frac{1}{\mu_{j-1}} = \frac{\nu_j}{\eta_j\mu_j p_j} \ge \frac{\rho_j p_j\{\log(p_j+1)+c_j^\circ\}}{\varepsilon\lambda_j} \ge \frac{\rho_j p_j}{\varepsilon\lambda_j} - \frac{1}{\varepsilon\mu_j} \tag{8.8}$$

PROOF: Since (8.8) is trivial, we will establish the rest of the claims by induction on j. We first consider the case with $j = \ell$. The existence of p_ℓ follows from the fact that $\rho_\ell > \rho_*$ for $p_\ell = 2$ and that $\rho_\ell \to 0$ as $p_\ell \to \infty$. Since $t_1^{(\ell)} = (\rho_\ell p_\ell^2/\varepsilon\lambda_\ell)\{\log(p_\ell+1) + c_\ell^\circ\}$ and $1/\lambda_\ell \le a'(\log p_\ell)^{k'}$, we have $t_1^{(\ell)} = o(p_\ell^k)$ and hence (3.3) at the ℓ-th level. Therefore Theorem 7.2 establishes $1/\eta_\ell \le 1+\varepsilon$ and (8.6)-(8.7) with $c_\ell^\circ = c^\circ\rho_*^3/(1-\rho_*)^3$, $c_\ell = c$, and $c_\ell' = c'$, where c°, c, c' are the constants appeared in (7.12)-(7.14).

Now we consider a general case with $1 \le j < \ell$, assuming that the above claims are valid for larger j. The existence of p_j can be similarly established since $\mu_j \le \mu_{j+1} \le \mu_\ell$ by (8.8). Since

$$t_1^{(j)} - \frac{\nu_j\cdots\nu_\ell}{\mu_\ell} \le \frac{\rho_\ell p_\ell}{\lambda_\ell}\prod_{i=j}^{\ell}\frac{\rho_i p_i}{\varepsilon\lambda_i}\left\{\log\left(\frac{\rho_i p_i}{\lambda_i}+1\right)+c_i^\circ\right\} \le \frac{\rho_j p_j}{\lambda_j}\left[\frac{\rho_j p_j}{\varepsilon\lambda_j}\left\{\log\left(\frac{\rho_j p_j}{\lambda_j}+1\right)+c_j^\circ\right\}\right]^{\ell-j+1}$$

by (8.5), (8.8) and $1/\lambda_j \le a'(\log p_j)^{k'}$, we have $t_1^{(j)} = o(p_j^k)$ and hence (3.3) at the j-th level. Induction hypothesis (8.6)-(8.7) for $j+1$ implies that a subtask at the j-th level satisfies (7.15)-(7.16) with $\kappa = (1-\varepsilon^{\ell-j+1})/(1-\varepsilon)$. Therefore Corollary 7.1 establishes $1/\eta_\ell \le (1-\varepsilon^{\ell-j+2})/(1-\varepsilon)$ and (8.6)-(8.7) with $c_j^\circ = \max\{c_{j+1}^\circ, c^\circ \rho_*^3/(1-\rho_*)^3\}$, $c_j = c$, and $c_j' = c'$, where c°, c, c' are the constants appeared in (7.17)-(7.20)). ∎

Lemma 8.2 *For any $1 \le j \le \ell$,*

$$\frac{\nu_1 \cdots \nu_\ell}{\mu_\ell} \le \prod_{i=1}^{j} \frac{p_i\{\log(p_i+1)+c_i^\circ\}}{\varepsilon} \cdot \left(\frac{\rho_j p_j}{\lambda_j}\right)^{\frac{j}{\ell}} \cdot \prod_{i=j+1}^{\ell} \frac{\rho_i^{\frac{i}{\ell}} p_i^{1+\frac{i}{\ell}} \{\log(p_i+1)+c_i^\circ\}^{1-\frac{i-1}{\ell}}}{\varepsilon^{1-\frac{i-1}{\ell}} \lambda_i^{\frac{i}{\ell}}}$$

In particular,

$$t_1 = \frac{\nu_1 \cdots \nu_\ell}{\mu_\ell} \le \frac{(\rho_1 \cdots \rho_\ell)^{\frac{1}{\ell}} p^{\frac{\ell+1}{\ell}} (\log p + c_*)^{\frac{\ell+1}{2}}}{\varepsilon^{\frac{\ell+1}{2}} (\lambda_1 \cdots \lambda_\ell)^{\frac{1}{\ell}}}$$

where c_* is a constant depending only on ℓ, ρ_* and $a, a', a'', k, k', b, b', b'', b'''$ appeared in (3.3)-(3.5), (7.15)-(7.16) and (7.1)-(7.2).

PROOF: Since the second claim immediately follows from the first claim with $j = 1$, we shall prove the first claim by induction on j. When $j = \ell$, the first claim is trivial by (8.5) and the definition of ρ_ℓ. Assume that the first claim holds for some $1 < j \le \ell$. We have

$$\frac{p_j\{\log(p_j+1)+c_j^\circ\}}{\varepsilon} \left(\frac{\rho_j p_j}{\lambda_j}\right)^{\frac{j}{\ell}} \le \left(\frac{\rho_{j-1} p_{j-1}}{\lambda_{j-1}}\right)^{\frac{j-1}{\ell}} \frac{\rho_j^{\frac{j}{\ell}} p_j^{1+\frac{j}{\ell}} \{\log(p_j+1)+c_j^\circ\}^{1-\frac{j-1}{\ell}}}{\varepsilon^{1-\frac{j-1}{\ell}} \lambda_j^{\frac{j}{\ell}}}$$

by (8.8). Hence we obtain the first claim also for $j-1$. ∎

Theorem 8.1 *Let $\mathcal{T}$ be a family of ℓ-level divisible tasks with $1 \le \ell \in \boldsymbol{N}$ and $k > \ell+1$, where k is the constant appeared in (3.3). For any $0 < \varepsilon < 1$, $\rho_* > 1$ and $1 < p_* \in \boldsymbol{N}$, there exit $t_1 > 1$ and $\Delta \equiv (\nu_1, \ldots, \nu_\ell; p_1, \ldots, p_\ell)$ with $\nu_j > 1$, $1 < p_j \in \boldsymbol{N}$, $p \equiv p_1 \cdots p_\ell > p_*$ such that the following hold.* (i) *When a task of expected size t_1 is solved in parallel by the ℓ-level dynamic load balancing scheme with Δ-division, the expected efficiency η is larger than $1-\varepsilon$.* (ii) *When p grows, t_1 increases as slowly as*

$$t_1 \le \frac{(\rho_1 \cdots \rho_\ell)^{\frac{1}{\ell}} p^{\frac{\ell+1}{\ell}} (\log p + c_*^\circ)^{\frac{\ell+1}{2}}}{(\varepsilon/\ell)^{\frac{\ell+1}{2}} (\lambda_1 \cdots \lambda_\ell)^{\frac{1}{\ell}}} \tag{8.9}$$

where λ_j denotes the production rate of a producer at the j-th level, with $1 < 1/\lambda_j \le a'(\log p_j)^{k'}$; ρ_j denotes the ratio of the production/consumption rate at the j-th level, which is larger than ρ_* and approaches to ρ_* as $p \to \infty$; and c_*° is a constant depending only on $a, a', a'', k, k', b, b', b'', b'''$ appeared in (3.3)-(3.5), (7.15)-(7.16) and (7.1)-(7.2). (iii) *Moreover, the parallel execution time $T_p^{(\ell)}$ satisfies the moderate diversity condition:*

$$E(T_p^{(\ell)} - x \mid T_p^{(\ell)} > x) \le \frac{t_1}{(1-\varepsilon)p}\left(1 + \frac{c_*}{1+\log\frac{t_1}{p}}\right) \qquad (\forall x \ge 0) \tag{8.10}$$

$$E\{(T_p^{(\ell)} - x)^2 \mid T_p^{(\ell)} > x\} \le \frac{c_*' t_1^2}{p^2} \qquad (\forall x \ge 0) \tag{8.11}$$

where $c_*, c_*' > 1$ are also constants depending only on $a, a', a'', k, k', b, b', b'', b'''$.

PROOF: For arbitrary $t_1^{(\ell+1)} \equiv 1/\mu_\ell > 2\rho_* a'(\log 2)^{k'}$, define $\nu_j, p_j, t_1^{(j)}, \mu_{j-1}$ for each $1 \le j \le \ell$ by induction on j according to Lemma 8.1 with ε/ℓ instead of ε. Then we have $\eta = \eta_1 \cdots \eta_\ell > (1-\varepsilon/\ell)^\ell > 1-\varepsilon$ and (8.10)-(8.11) owing to Lemma 8.1. Besides, Lemma 8.2 implies (8.9). By letting $t_1^{(\ell+1)} \equiv 1/\mu_\ell \to \infty$, we have $p \to \infty$, in particular, $p > p_*$. Finally, $\rho_j \to \rho_*$ as $p \to \infty$ follows from the assumption (8.4). ∎

Remark 8.1 In the above proof, we may take any large ν_j with $\nu_j \ge (p_j/\varepsilon)\{\log(p_j+1) + c_j^o\}$, as long as $t_1^{(j)} \equiv \nu_j t_1^{(j+1)} \le a p_j^k$ is not violated (cf. (3.3)). Then the claims in Lemma 8.1 still hold. In particular, the expected efficiency η is larger than $1-\varepsilon$ for a subtask of expected size somewhat larger than t_1.

This theorem implies that the multi-level dynamic load balancing scheme is *indeed* more scalable than the single-level one in the sense of isoefficiency. The isoefficiency function of the ℓ-level dynamic load balancing scheme is $O(p^{(\ell+1)/\ell}(\log p)^{(\ell+1)/2}/\lambda_*)$, where $\lambda_* = (\lambda_1 \cdots \lambda_\ell)^{1/\ell}$ is the geometric mean of the production rates through ℓ levels and $1/\lambda_* = O((\log p)^{k'})$ by assumption. Its principal factor $p^{1+1/\ell}$ is decreasing in ℓ. Scalability is thus improved with the number of levels ℓ.

8.3 Asymptotic analysis of the processor configuration

In this subsection, we investigate the processor configuration that is employed in Theorem 8.1, which makes the multi-level dynamic load balancing scheme fairly scalable as the number of consumer processors increases. More precisely, we study how fast the degree of root producer, p_1, should increase as the number of consumers, p, increases.

Let $\mathcal{T}$ be a family of ℓ-level divisible tasks as above, and take arbitrary $0 < \varepsilon < 1$ and $\rho_* > 1$. For any large $t_1^{(\ell+1)} \equiv 1/\mu_\ell$, define $\nu_j, p_j, t_1^{(j)}, \mu_{j-1}$ for each $1 \le j \le \ell$ by induction on j according to Lemma 8.1, and let $p = p_1 \cdots p_\ell$ and $t_1 = \nu_1 \cdots \nu_\ell/\mu_\ell$. For brevity, we write:

$$\bar{p}_j = \frac{1}{\mu_j}, \quad \pi_j = \frac{\log(p_j+1) + c_j^o}{\varepsilon} \quad (1 \le j \le \ell); \qquad \kappa_j = \frac{\rho_{j-1}}{\rho_* \eta_j} \quad (1 < j \le \ell)$$

where c_j^o is he constant appeared in Lemma 8.1, and η_j is the efficiency at the j-th level. Note that, for each $1 < j \le \ell$, we have:

(8.12) $$\bar{p}_j \pi_j \le \bar{p}_{j-1} \le \kappa_j \bar{p}_j \pi_j$$

In fact, the first inequality immediately follows from (8.8). By (8.5), we also have:

$$\bar{p}_{j-1} = \frac{\rho_{j-1}\bar{p}_{j-1}}{\lambda_{j-1}} \le \frac{\rho_{j-1}}{\rho_* \mu_{j-1}} = \frac{\rho_{j-1}\nu_j}{\rho_* \eta_j \mu_j p_j} = \frac{\rho_{j-1}\pi_j}{\rho_* \eta_j \mu_j} = \kappa_j p_j \pi_j$$

Lemma 8.3 *For each $1 \le j \le \ell$, we have*

(8.13) $$\bar{p}_j \cdots \bar{p}_\ell \prod_{i=j+1}^{\ell} \pi_i^{\ell-i+1} \le \bar{p}_j^{\ell-j+1} \le \bar{p}_j \cdots \bar{p}_\ell \prod_{i=j+1}^{\ell} (\kappa_i \pi_i)^{\ell-i+1}$$

PROOF: We use induction on j. When $j = \ell$, the claim is trivial. Assume that the claim is true for some $1 < j \le \ell$. We have $\bar{p}_j^{\ell-j+1}\pi_j^{\ell-j+1} \le \bar{p}_{j-1}^{\ell-j+1} \le \kappa_j^{\ell-j+1}\bar{p}_j^{\ell-j+1}\pi_j^{\ell-j+1}$ by (8.12). Applying the induction hypothesis to the both ends of this expression, and multiplying p_{j-1} to each side, we obtain the desired result for $j-1$. ∎

Theorem 8.2 *Under the same condition as in Theorem 8.1, we can take* $\Delta \equiv (\nu_1, \ldots, \nu_\ell; p_1, \ldots, p_\ell)$ *such that all the claims in Theorem 8.1 hold (in particular,* $\eta > 1 - \varepsilon$*) and*

$$\frac{p^{\frac{1}{\ell}}(\log p)^{\frac{\ell-1}{2}}}{\varepsilon^{\frac{\ell-1}{2}}(\lambda_1 \cdots \lambda_\ell)^{\frac{1}{\ell}}} \lesssim \frac{p_1}{\lambda_1} \lesssim \frac{e^{\frac{\ell}{2}} p^{\frac{1}{\ell}}(\log p)^{\frac{\ell-1}{2}}}{\varepsilon^{\frac{\ell-1}{2}}(\lambda_1 \cdots \lambda_\ell)^{\frac{1}{\ell}}} \quad \text{as} \quad p \to \infty \tag{8.14}$$

where $x_p \lesssim y_p$ *indicates that:* $\limsup_{p\to\infty} x_p/y_p \le 1$.

PROOF: Take $\Delta \equiv (\nu_1, \ldots, \nu_\ell; p_1, \ldots, p_\ell)$ as in the proof of Theorem 8.1. Then Lemma 8.3 with $j = 1$ implies that

$$\bar{p}_1 \cdots \bar{p}_\ell \prod_{i=2}^{\ell} \pi_i^{\ell-i+1} \le \bar{p}_1' \le \bar{p}_1 \cdots \bar{p}_\ell \prod_{i=2}^{\ell} (\kappa_i \pi_i)^{\ell-i+1}$$

Since $\rho_j \to \rho_*$ as $p \to \infty$ and $1 \le 1/\eta_j \le 1 + \varepsilon/\ell$, we have $1 \lesssim \kappa_j \lesssim 1 + \varepsilon/\ell$. Hence, (8.12) implies $\log p_{j-1} \simeq \log p_j$ since $\bar{p}_j = \rho_j p_j/\lambda_j$ and $1/\lambda_j \le a'(\log p_j)^{k'}$. Here $\simeq$ indicates that the ratio of the both sides converges to one as $p \to \infty$. Therefore, we have $\pi_j \simeq (\log p)/\varepsilon$ and $\bar{p}_j \simeq \rho_* p_j/\lambda_j$ for each $1 \le j \le \ell$ in the above expression. (Note that we are using ε/ℓ instead of ε in defining π_j.) Thus this expression gives the desired estimate. ∎

Finally, we intuitively discuss the plausibility of the above processor configuration. In the uniform tree structure of processors, we would have $p_1 = \cdots = p_\ell = p^{1/d}$. However, the number of immediate descendants a producer can afford (degree of the producer) is clearly proportional to its production rate. Hence it is natural for p_1/λ_1 and $p^{1/\ell}/(\lambda_1 \cdots \lambda_\ell)^{1/\ell}$ to appear in the above expression (8.14), where $p^{1/\ell} = (p_1 \cdots p_\ell)^{1/\ell}$ is the geometric mean of the degree of producers through ℓ levels and $(\lambda_1 \cdots \lambda_\ell)^{1/\ell}$ is the geometric mean of the production rate.

Even if $\lambda_1 = \cdots = \lambda_\ell$, the uniform tree with $p_1 = \cdots = p_\ell$ does not provide us with the best scalability. For example, consider the case with two levels: $\ell = 2$, $\lambda_1 = \lambda_2 = \lambda_*$, $p_1 = p_2 = \sqrt{p}$. We will use the notations defined in Section 8.2 such as $t_1^{(j)}$, μ_j, η_j and ρ_j. According to Theorem 6.1, we can take $t_1^{(2)} = \Theta(p \log p/\lambda_*)$ while maintaining $\eta_2 = \Omega(1)$. This implies $1/\mu_1 = \Theta(\sqrt{p} \log p/\lambda_*)$. Note that this is *more* than enough to ensure $\rho_1 = \lambda_*/\mu_1\sqrt{p} > 1$ and to avoid the root producer bottleneck. Proceeding as in Section 6.2, we can take $\nu_1 = \Theta(\sqrt{p} \log p)$ while maintaining $\eta_1 = \Omega(1)$ as well (cf. (6.20)). Therefore we obtain the isoefficiency function $t_1 = \nu_1 t_1^{(2)} = O(p^{3/2}(\log p)^2/\lambda_*)$ on the uniform tree structure. This is worse than $t_1 = O(p^{3/2}(\log p)^{3/2}/\lambda_*)$ implied by Theorem 8.1.

In fact, (8.14) shows that the degree of the root producer has an extra factor of $(\log p)^{(\ell-1)/2}$. And, in general, the degree of a producer at a higher level is likewise larger than that at a lower level, which gives a tree bushy near the root. Hence, in terms of the ratio of the production/consumption rate, the higher the level, the lower the consumption rate, i.e., the larger granularity. This is an accumulated effect of the fact that, at each level, a sufficient number of "subtasks" are employed so as to compensate for the load imbalance.

9 Conclusions

We investigated the efficiency of the single-level and multi-level dynamic load balancing schemes for a program that requires many independent pieces of computation (subtasks) of non-uniform size. A queueing model was introduced in order to analyze how the efficiency is affected by both the load imbalance due to the non-uniform subtask sizes and the possible bottleneck at feeding many processors

with the subtasks. The *moderate diversity condition* on the non-uniformness was described, which guarantees a reasonable efficiency of these schemes. Intuitively, it requires that a subtask should be *steadily* executed, i.e., its remaining execution time should not be expected to blow up. Several bounds of the efficiency were obtained under this condition, which concisely express how the efficiency (η) depends on the production rate of the subtasks (λ), the amount of overall computation (t_1), the number of consumer processors (p), and the number of levels (ℓ). In particular, we showed how the multi-level scheme improves the scalability over the single-level scheme in terms of isoefficiency function [9].

On the other hand, these load balancing schemes are not powerful for a program that is too non-uniform to satisfy our condition, as we intuitively discussed earlier. More elaborated load balancing schemes should be devised in this case. And it would be worthwhile to perform similar probabilistic analysis of them.

So far, we have not considered inter-processor communication latency, speculative computation [16], or other overheads associated with parallel execution. How the efficiency suffers from these factors should be treated in future works.

Acknowledgments

We would like to thank Vipin Kumar for taking part in valuable discussions during his stay at ICOT in September 1990.

Appendix A. NBU Model

In this paper, we introduced the moderate diversity condition in order to guarantee a reasonable efficiency. As mentioned earlier in Section 6, it is closely related to several *aging* notions in the theory of reliability engineering, e.g., NBU, IHR, DMRL [13]. In this appendix, we discuss a model of divisible task based on NBU notion, which is in fact subsumed in our earlier model. The inheritance property, which was crucial for treating the multi-level schemes (cf. Section 8), can be shown rather easily in this restricted model. This appendix gives alternative proofs to those in Section 7.

A.1 Definition of NBU model

In this subsection, we define a model of divisible task based on NBU notion. NBU is weaker than IHR (IFR) mentioned in Section 6.1. A nonnegative random variable T is said to be NBU (*new better than used*) if and only if $[T - t \mid T > t] \prec T$ for any $t \geq 0$. Here $[T - t \mid T > t]$ denotes the distribution of $T - t$ under the condition $T > t$, and $\prec$ denotes the *stochastic inequality* [5]. Namely, for any real-valued random variables X and Y, X is *stochastically smaller* than Y (denoted $X \prec Y$ or $P^X \prec P^Y$) if and only if $E\{f(X)\} \leq E\{f(Y)\}$ for any bounded nondecreasing real-valued function f. Similarly, we will say that an $\boldsymbol{N}$-valued random variable N is NBU if and only if $[N - n \mid N > n] \prec N$ for any $n \in \boldsymbol{N}$.

Let $\mathcal{T} = \{(N^{(t_1,\nu)}, \{R_n^{(t_1,\nu)}\}_{n=1}^{\infty}, \{U_n^{(t_1,\nu,p)}\}_{n=1}^{\infty})\}$ be a family of divisible tasks satisfying Assumption 3.1. We define $T_1^{(t_1)}$, $\lambda^{(t_1,\nu,p)}$, $\alpha^{(t_1,\nu,p)}$, $\mu^{(t_1,\nu)}$ and $\rho^{(t_1,\nu,p)}$ as in Section 3.2. Instead of the moderate diversity condition (Assumptions 6.1 and 7.1), we assume the following condition throughout the rest of this appendix.

Assumption A.1 (NBU model) $\mathcal{T}$ satisfies both of the following conditions.

(i) $N^{(t_1,\nu)}$, $R_n^{(t_1,\nu)}$ and $U_n^{(t_1,\nu,p)}$ are NBU for any $1 < \nu < t_1$, $1 < p \in \boldsymbol{N}$ and $n \in \boldsymbol{N}$.

(ii) (boundedness of coefficients of variation) There exist constants b and b' such that $E\{(R_n^{(t_1,\nu)})^2\} \leq b/(\mu^{(t_1,\nu)})^2$ and $E\{(N^{(t_1,\nu)})^2\} \leq b'\nu^2$ hold for any $1 < \nu < t_1$ and $n \in \boldsymbol{N}$.

Note that this assumption is stronger than Assumptions 6.1 and 7.1. In fact, (i) implies (6.1) and (7.1); and (ii) together with (i) implies (6.2) and (7.2). As before, we will often omit the superscripts for brevity, and write: $N = N^{(t_1,\nu)}$, $R_n = R_n^{(t_1,\nu)}$, $\mu = \mu^{(t_1,\nu)}$ and so on. We also define T_p, $t_p \equiv E(T_p)$, $\mathcal{F}_t$, X_n and others as in Section 3.

A.2 Inheritance of NBU by task size

In the NBU model, the task size also satisfies conditions similar to (i) and (ii) in Assumption A.1. Namely, we have the next theorem.

Theorem A.1 *Let $\mathcal{T}$ be a family of divisible tasks satisfying Assumptions 3.1 and A.1. Then*

(i) *The task size $T_1^{(t_1)}$ is NBU for any $t_1 > 1$.*

(ii) *$E\{(T_1^{(t_1)})^2\} \leq (b + b' - 1)t_1^2$ for any $t_1 > 1$.*

PROOF: Take an arbitrary $x \geq 0$ and define $K = \min\{k \in \boldsymbol{N} \mid \sum_{n=1}^{k} R_n > x\}$. For any $k \in \boldsymbol{N}$ and $\{r_n\}_{n=1}^{k-1} \in \boldsymbol{R}_+^{k-1}$ with $\sum_{n=1}^{k-1} r_n \leq x$, we have

$$
\begin{aligned}
& [T_1 - x \mid R_n = r_n \ \ (1 \leq \forall n < k), \ \ K = k, \ \ T_1 > x] \\
= & \left[\sum_{n=1}^{k-1} r_n + R_k + \sum_{n=k+1}^{N} R_n - x \ \middle| \ R_n = r_n \ \ (1 \leq \forall n < k), \ \ R_k > x - \sum_{n=1}^{k-1} r_n, \ \ N \geq k \right] \\
= & \left[R_k - x + \sum_{n=1}^{k-1} r_n \ \middle| \ R_k > x - \sum_{n=1}^{k-1} r_n \right] * \left[\sum_{n=k+1}^{N} R_n \ \middle| \ N \geq k \right]
\end{aligned}
$$

where $*$ denotes the convolution of distributions over $\boldsymbol{R}$. This is stochastically smaller than $P^{R_k} * [\sum_{n=k+1}^{N} R_n \mid N \geq k] \prec [\sum_{n=1}^{N-k+1} R_n \mid N - k + 1 > 0]$, since R_k is NBU and $\{R_n\}_{n=1}^{\infty}$ is i.i.d. This is, in turn, stochastically smaller than $\sum_{n=1}^{N} R_n = T_1$, because N is NBU and independent of $R_n \geq 0$. Therefore,

$$
[T_1 - x \mid R_n = r_n \ \ (1 \leq \forall n < k), \ \ K = k, \ \ T_1 > x] \prec T_1
$$

for arbitrary $\{r_n\}_{n=1}^{k-1} \in \boldsymbol{R}_+^{k-1}$ with $\sum_{n=1}^{k-1} r_n \leq x$. Hence $[T_1 - x \mid K = k, \ T_1 > x] \prec T_1$. Since $T_1 > x$ implies $K < +\infty$ a.s., we obtain $[T_1 - x \mid T_1 > x] \prec T_1$ for any $x \geq 0$, which establishes (i). Moreover, owing to Assumption A.1 (ii), we have

$$
E\{(T_1)^2 \mid N = n\} = E\{(R_1 + \cdots + R_n)^2\} = nE\{(R_1)^2\} + (n^2 - n)\{E(R_1)\}^2 \leq \frac{n^2 + (b-1)n}{\mu^2}
$$

and $E\{(T_1)^2\} \leq \{b'\nu^2 + (b-1)\nu\}/\mu^2 \leq (b + b' - 1)t_1^2$, which establishes (ii). ∎

A.3 Inheritance of NBU by parallel execution time

In the NBU model, the parallel execution time also satisfies conditions similar to (i) and (ii) in Assumption A.1. Namely, we have the next theorem.

Theorem A.2 *Let $\mathcal{T}$ be a family of divisible tasks satisfying Assumptions 3.1 and A.1. Then*

(i) *The parallel execution time $T_p^{(t_1,\nu,p)}$ is NBU for any $1 < \nu < t_1$ and $1 < p \in \boldsymbol{N}$.*

(ii) *There exists a constant $c > 0$ such that $E\{(T_p^{(t_1,\nu,p)})^2\} \leq c(1+\rho^{-2})(t_p)^2$ for any $\rho > 0$, $1 < \nu < t_1$ and $1 < p \in \boldsymbol{N}$ with $t_1 \geq (\rho p^2/\lambda)\log(p+1)$.*

PROOF: Assume that $\mathcal{F}_t$, the information up to time t, is given and that $T_p > t$ and $N = n$, for some $t > 0$ and $n \in \boldsymbol{N}$. Define k_0, τ_p, R_i' and U_i' ($\forall i \in \boldsymbol{N}$) as in the beginning of Section 7.3. Then we have $T_p - t = \tau_p(n-k_0, \{R_n'\}_{i=1}^{\infty}, \{U_i'\}_{i=1}^{\infty})$ under the condition: $\mathcal{F}_t$, $T_p > t$, $N = n$. Since R_is are NBU and i.i.d., we have $[R_i' \mid \mathcal{F}_t,\ T_p > t,\ N = n] \prec R_i$. By definition, τ_p is monotone in its components:

$$n \leq n',\ \ r_i \leq r_i',\ \ u_i \leq u_i'\ \ (\forall i \in \boldsymbol{N}) \quad \Longrightarrow \quad \tau_p(n, \{r_i\}_{i=1}^{\infty}, \{u_i\}_{i=1}^{\infty}) \leq \tau_p(n', \{r_i'\}_{i=1}^{\infty}, \{u_i'\}_{i=1}^{\infty})$$

Therefore, we have

$$[T_p - t \mid \mathcal{F}_t,\ T_p > t,\ N = n] \prec \tau_p(n-k_0, \{R_i\}_{i=1}^{\infty}, \{U_i\}_{i=1}^{\infty}) = [T_p \mid N = n-k_0] \prec [T_p \mid N = n]$$

and hence $[T_p - t \mid T_p > t,\ N = n] \prec [T_p \mid N = n]$. Since $N < +\infty$ a.s., we have $[T_p - t \mid T_p > t] \prec T_p$ for any $t > 0$, which establishes (i).

Now, we shall prove (ii). Let X_N be the start of the execution period of the last subtask (cf. Definition 3.1(ii)). Then the total amount of computation time of p consumers up to time X_N is at most $R_1 + \cdots + R_{N-1}$, and that of idle time is at most $p(U_1 + \cdots + U_N)$. Hence we have $X_N \leq U_1 + \cdots + U_N + (R_1 + \cdots + R_{N-1})/p$. Therefore, $(T_p)^2 \leq 3(U_1 + \cdots + U_N)^2 + 3(R_1 + \cdots + R_{N-1})^2/p^2 + 3(T_p - X_N)^2$, and hence

$$E\{(T_p)^2 \mid N = n\} < 3\alpha^2 n^2 + 3(n-1)^2/\mu^2 p^2 + 3E\{(T_p - X_N)^2 \mid N = n\}$$

Since R_is are NBU and i.i.d., we have $T_p - X_N \prec \max\{R_1, \ldots, R_p\}$. Hence, by Proposition 7.2 and Assumption A.1 (i) (ii), we have $E\{(T_p - X_N)^2\} \leq bh(p)/\mu^2$, where $h(p)$ is the function defined by (7.21). Therefore, $E\{(T_p)^2 \mid N = n\} \leq 3\alpha^2 n^2 + 3(n-1)^2/\mu^2 p^2 + 3h(p)/\mu^2$, and hence

$$E\{(T_p)^2\} \leq 3b'\nu^2 \left(\alpha^2 + \frac{1}{\mu^2 p^2}\right) + \frac{3}{\mu^2}\left[\{\log(p+1) + C + 2\}^2 + \frac{\pi^2}{6}\right]$$

again by Assumption A.1 (ii). Here we have

$$1 < \nu = \mu t_1, \qquad \frac{t_1}{p} \leq t_p, \qquad \alpha \leq \frac{a''}{\lambda} = \frac{a''}{\rho\mu p}, \qquad \frac{1}{\mu}\log(p+1) = \frac{\rho p}{\lambda}\log(p+1) \leq \frac{t_1}{p}$$

Thus we obtain (ii). ∎

References

[1] L. V. Ahlfors, *Complex Analysis*, McGraw-Hill (1966).

[2] 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).

[3] A. Gupta and V. Kumar, "The Scalability of FFT on Parallel Computers," *Proc. of the Frontiers 90 Conference on Massively Parallel Computation* (1990).

[4] K. Ito (ed.), *Encyclopedic Dictionary of Mathematics*, MIT Press, 2nd ed., (1987).

[5] T. Kamae, U. Krengel and G.L.O'Brien, "Stochastic Inequalities on Partially Ordered Spaces," *The Annals of Probability*, Vol.5, No.6, pp.899-912, (1977).

[6] R. M. Karp, "The Probabilistic Analysis of Some Combinatorial Search Algorithms," in J. F. Traub (ed.), *Algorithms and Complexity: New Directions and Recent Results*, Academic Press, pp.1–19 (1976).

[7] 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*, pp.145–152, (1991).

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

[9] V. Kumar and A. Gupta, "Analysis of Scalability of Parallel Algorithms and Architectures: A Survey," *to appear in the 1991 International Conference on Supercomputing* (1991).

[10] 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).

[11] K. Nakajima, Y. Inamura, N. Ichiyoshi, K. Rokusawa and T. Chikayama, "Distributed Implementation of KL1 on the Multi-PSI/V2", *Proc. 6th Int. Conf. on Logic Programming* (1989).

[12] N. U. Prabhu, *Queues and Inventories*, John Wiley (1965).

[13] M. Shaked and J. G. Shanthikumar, "Reliability and Maintainability," in D. P. Heyman and M. J. Sobel (ed.), *Stochastic Models*, Handbooks in Operations Research and Management Science, Vol.2, North-Holland, pp.653–713 (1990).

[14] P. C. Treleaven, "Parallel Architecture Overview," *Parallel Computing* 8, pp.71–83 (1988).

[15] Y.-T. Wang and R. J. T. Morris, "Load Sharing in Distributed Systems," *IEEE Trans. Comput.*, vol. C-34, no. 3, pp. 204–217 (1985).

[16] F. W. Burton, "Speculative Computation, Parallelism, and Functional Programming," *IEEE Trans. Comput.*, vol. C-34, no. 12, pp. 1190–1193 (1985).