The Price of Synchrony

Peter J. Downey

TR 96-09

ABSTRACT

Loops that synchronize parallel processors at the end of each iteration are compared with loops that do not synchronize their iterations. In the presence of data dependencies, loop synchronization cannot always be removed—the purpose here is to estimate the additional costs incurred when synchronization is necessary.

Suppose there are n parallel processors each executing k iterations. Under the assumption that each iteration of the loop body runs for a time controlled by an independent identically distributed random variable X with mean μ and variance σ^2, it is shown here that the ratio of the expected time taken with synchronized loops to the expected time taken with unsynchronized loops is asymptotically

$$\frac{\mathbf{E}X_{(n)} + \varepsilon(n)}{\mu} \qquad (n, k \to \infty) \tag{$*$}$$

where $\mathbf{E}X_{(n)}$ is the expected maximum of n independent random variables with with the distribution of X, and $\varepsilon(n)$ is the (deterministic) time to fork n processes. Since $\mathbf{E}X_{(n)}$ grows with n in a way that can be estimated for many particular distributions, the ratio ($*$) yields an estimate of the relative time penalty incurred by loop synchronization and the sensitivity of this penalty as the degree of parallelism n increases.

November 20, 1996

Department of Computer Science

The University of Arizona

Tucson, Arizona 85721

The Price of Synchrony

1. Introduction

Large scale scientific computations performed on parallel processors often contain parallel iterations, in which the same body of code is repeatedly executed on each of the processors, and in which the number of iterations on each processor is very large. Depending on the specific application, the iterations may be synchronized or unsynchronized. In *synchronized* iteration, iteration number i must terminate on all of the processors before iteration number $i + 1$ can begin on any processor. In *unsynchronized* iteration, iterations can proceed in order but independently on each processor, with the only synchronization occurring after all parallel processors have finished their iterations.

This paper compares the running time of synchronized iteration to that of unsynchronized iteration under simple probabilistic assumptions about the running time of iteration bodies. It assesses the time penalties incurred by synchronization, and indicates the main determinants affecting such time penalties. Results are asymptotic—valid only for large numbers of iterations across a large number of processors.

Let there be n identical processors $P_1, P_2, \cdots, P_n$ each repeatedly executing a body of code B for a total of k iterations. Depending on the particular processor and the iteration number, this code will give rise to computations of various lengths (depending on differing conditional paths taken by varying initial data, differing machine environments, etc.). Let us denote by $\mathsf{X}_j^{(i)}$ the computation done by machine j $(1 \le j \le n)$ at iteration i $(1 \le i \le k)$.

In an unsynchronized loop, processor P_j performs the sequence of computations

$$\mathsf{X}_j^{(1)}, \mathsf{X}_j^{(2)}, \cdots, \mathsf{X}_j^{(k)} \quad ,$$

with no pause between successive computations. In the synchronized loop, P_j performs the same sequence of computations; however, all stage i computations on each of the processors

$$\mathsf{X}_1^{(i)}, \mathsf{X}_2^{(i)}, \cdots, \mathsf{X}_n^{(i)}$$

are constrained to begin simultaneously. This introduces overheads in both synchronizing the start of the n

parallel computations, as well as in awaiting the termination of the n parallel computations from the previous ($i-1$th) iteration.

To describe the differences between these two loop structures in terms of a (typical) parallel programming language, we use pseudocode taken from [Ber97]. If `B` is the repetitive body of code performed at each iteration, then the synchronized loop corresponds to the code

Synchronized:

```
for i := 1 to k do
    for 1 ≤ j ≤ n do in parallel
        B
    end in parallel
endfor
```

where the parallel **for** loop is the inner loop. The unsynchronized loop corresponds to the dual loop structure

Unsynchronized:

```
for 1 ≤ j ≤ n do in parallel
    for i := 1 to k do
        B
    endfor
end in parallel
```

in which the parallel **for** loop is the outer loop. The code body `B` is in general parameterized by both the processor number j as well as the iteration variable i.

These two programs are in general not equivalent. The question of whether the synchronized version can be replaced by the unsynchronized version while preserving equivalence can be answered only by reference to the particular structure of the iteration body `B`. For example, if `B` makes reference to data computed by earlier iterations on machine j only, then the unsynchronized version can be used. Generally, cross-processor dependencies will force the synchronized loop to be employed. Many algorithms based on the parallel random access machine (*PRAM*) computational model [Ber97] require that loops be synchronized.

In this paper, we study the relative efficiency of these two loop organizations in isolation from the question of whether the more efficient version can actually be used correctly. This serves to compare the relative cost of using synchronization should it be necessary, and to assess the improvement obtained when using unsynchronized loops can be correctly used.

1.1 Timing Model

To model the running time for these two loop structures, let us denote the execution time for computation $\mathsf{X}_j^{(i)}$ by $X_j^{(i)}$. In this paper, $X_j^{(i)}$ is assumed to be a random variable. In the model discussed here, we assume that the distribution of these random variables is known, but that the particular amount of time taken by processor P_j on its ith iteration—$X_j^{(i)}$—is not known deterministically. We therefore seek to understand how running time depends upon the *ensemble* of times, and characteristics of this ensemble, rather than upon particular instances of execution.

One obvious influence on the total time of a parallelized iterative process comes from execution time uncertainties. Successive executions of the code body can vary widely and unpredictably in running time; but the iteration body can be characterized by a distribution of running time rather than by a single quantity. The performance in time of these loops is clearly influenced by the characteristics of this distribution, such as its mean and standard deviation. In addition, when loops are run in parallel, one is forced to consider not just sums of uncertain time durations, but also the extreme (maximum) running time of a number uncertain time durations.

For the moment, let us ignore any time overheads necessary to *initiate* (or *fork*) n computations on processors $P_1, \cdots, P_n$. Such overheads will occur whenever a **for** $\cdots$ **do in parallel** statement is encountered. These overheads will be added during the analysis in Section 2 below. With this momentary simplification, the total time spent in executing the synchronized loop can be depicted in Figure 1.1 below.

The vertical lines in these figures depict synchronization barriers. The unsynchronized loop model is depicted in Figure 1.2.

1.2 Model Assumptions

In Section 2, we compare the expected values of S_n, the running time of the synchronized loop, with U_n, the running time of the unsynchronized loop. Here we collect the assumptions made in the analysis. Some assumptions are made in order to make the analysis tractable, and some are made to focus attention on asymptotic conditions of interest—where both the number of processors n and the number of iterations k are very large.

- There are k iterations of the loop performed in parallel on n processors.

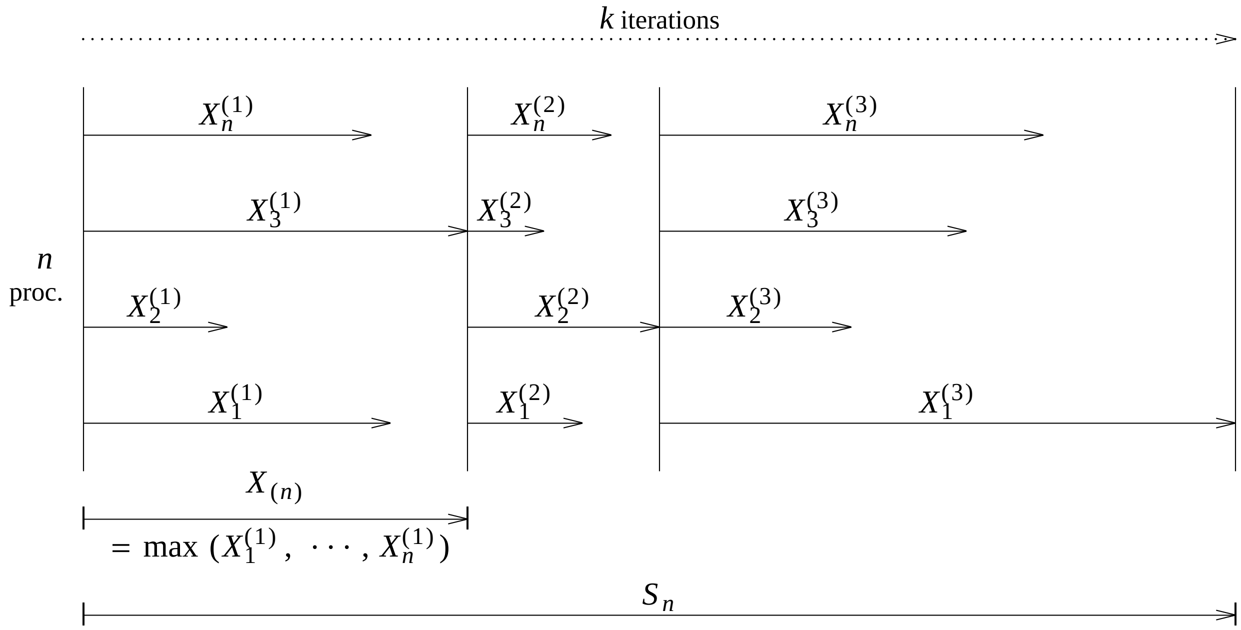

Figure 1.1: Synchronized Loops. Time for k iterations on n processors, with synchronization after each parallel loop body. Vertical lines are barriers at which synchronization occurs. In this illustration, there are $n = 4$ processors and $k = 3$ iteration stages.

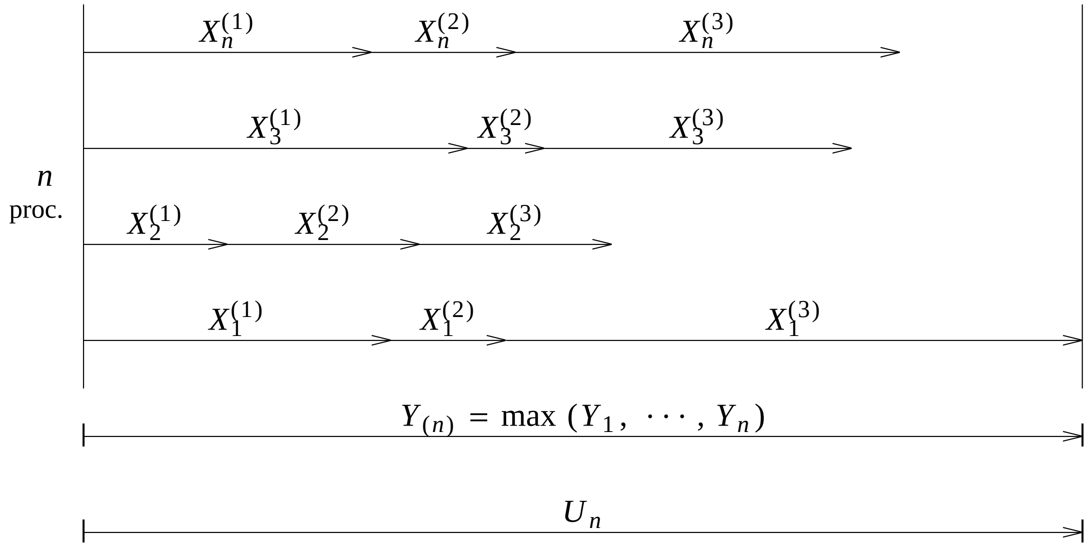

Figure 1.2: Unsynchronized Loops. Time for k iterations on n processors, with no synchronization of loops across processors. Vertical lines are barriers at which synchronization occurs. Random variables Y_i are sums of k random variables, each distributed as X, e.g., $Y_1 = X_1^{(1)} + X_1^{(2)} + X_1^{(3)} + \cdots$. In this illustration, there are $n = 4$ processors and $k = 3$ iterations.

- Task times $X_j^{(i)}$ $(1 \le j \le n,\ 1 \le i \le k)$ are independent, identically distributed random variables with a common distribution $F(x) = \mathbf{P}[X \le x]$.

- Distribution F has finite variance σ^2. The mean of F will be denoted μ.
- The time to fork n parallel tasks is a deterministic quantity $\varepsilon(n)$. In general it is will be a slowly increasing function of the number of processors. We will assume that it increases at most linearly in n:

$$\varepsilon(n) = O(n) \qquad (n \to \infty) .$$

- As the number k of iterations and the number of processors n gets large, we assume that the numbers of processors used grows more slowly than the number of iterations performed:

$$\frac{n}{k} \to 0 \qquad (n, k \to \infty) .$$

  This assumption assures that a large number of iterations are performed on each processor, no matter how many processors are available.

2. Analysis

2.1 Synchronized Loop

The completion time S_n (Figure 1.1) of the synchronized loop is the sum of k stages. Each stage completes in a time distributed as $X_{(n)}$, the maximum of n i.i.d. random task times with distribution F, plus the time $\varepsilon(n)$ needed to fork these n tasks. This yields an expectation of

$$\mathbf{E}S_n = k \cdot \mathbf{E}X_{(n)} + k \cdot \varepsilon(n) . \qquad (2\text{-}1)$$

2.2 Unsynchronized Loop

The completion time U_n (Figure 1.2) is a maximum of n sums, plus the initial fork time. Thus

$$U_n = \max(Y_1, Y_2, \cdots, Y_n) + \varepsilon(n) , \qquad (2\text{-}2)$$

where each of the Y_i are independent, and each is the sum of k i.i.d. variates with distribution F. That is:

$$Y_i =_d X_1 + X_2 + \cdots + X_k . \qquad (2\text{-}3)$$

Since F has finite variance, then for large k, each Y_i converges in distribution to a normal random variable with the appropriate mean and variance:

$$Y_i \xrightarrow{d} N(k \cdot \mu, k \cdot \sigma^2) \qquad (k \to \infty) . \qquad (2\text{-}4)$$

Therefore for large k, U_n is asymptotically the maximum of n normals with the distribution (2-4), plus a term that accounts for the forking overhead.

From the asymptotic theory of extremes of the normal distribution [Res87], it is known that, if $Z_1, Z_2, \cdots, Z_n$ are independent normals with mean μ_Z and standard deviation σ_Z, then the expectation of the maximum grows like $O(\sqrt{\ln n})$. More precisely:

$$\mathbf{E}Z_{(n)} = \mu_Z + \sigma_Z \cdot \sqrt{2 \cdot \ln n} + o(1) \qquad (n \to \infty) \ . \tag{2-5}$$

Applying this result to (2-2) and (2-4) results in

$$\begin{aligned} \mathbf{E}U_n &= \mathbf{E}Y_{(n)} + \varepsilon(n) \\ &= k\mu + k^{1/2}\sigma\sqrt{2 \ln n} + \varepsilon(n) + o(1) \qquad (n, k \to \infty) \ , \end{aligned} \tag{2-6}$$

where the $o(1)$ term reflects the normal approximation in (2-4) as well as the approximation to the extreme in (2-5).

2.3 The Ratio

The relative speed-up obtained by removing synchronization on each iteration is, using (2-1) with (2-6),

$$\begin{aligned} \frac{\mathbf{E}S_n}{\mathbf{E}U_n} &= \frac{k \cdot \mathbf{E}X_{(n)} + k \cdot \varepsilon(n)}{k\mu + k^{1/2}\sigma\sqrt{2 \ln n} + \varepsilon(n)} + o(1) \\ &= \frac{\mathbf{E}X_{(n)} + \varepsilon(n)}{\mu + \sigma\sqrt{2(\ln n)/k} + \varepsilon(n)/k} + o(1) \qquad (n, k \to \infty) \ , \end{aligned} \tag{2-7}$$

Since $n = o(k)$ by assumption, we have that $(\ln n)/k = o(1)$. Since $\varepsilon(n) = O(n)$ by assumption, we have also that $\varepsilon(n)/k = O(n/k) = o(1)$. Thus (2-7) simplifies to:

$$\frac{\mathbf{E}S_n}{\mathbf{E}U_n} = \frac{\mathbf{E}X_{(n)} + \varepsilon(n)}{\mu} + o(1) \qquad (n, k \to \infty) \ , \tag{2-8}$$

The result in (2-8) can be interpreted as indicating that the relative time dilation, or ‘‘slow-down’’, produced by synchronizing each iteration consists of the sum of a forking overhead factor ($\varepsilon(n)/\mu$) and a joining overhead factor ($\mathbf{E}X_{(n)}/\mu$). For large k, each group of n tasks out of the overall total of $k \cdot n$ tasks is slowed by this amount due to synchronization. The expression in (2-8) is thus a relative dilation factor of $\mathbf{E}S_n$ over $\mathbf{E}U_n$.

As an example, if X is normal with mean μ and variance σ^2, then $\mathbf{E}X_{(n)} = \mu + \sigma \cdot \sqrt{2 \cdot \ln n} + o(1)$ and we obtain from (2-8) the ratio $\mathbf{E}S_n / \mathbf{E}U_n = 1 + (\sigma/\mu)\sqrt{2 \ln n} + \varepsilon(n)/\mu + o(1)$. The latter expression shows that the joining overhead term is directly affected by the coefficient of variation (σ/μ) of the distribution of X, as well as rising slowly with increases in degree n of parallelism.

3. Growth of the Extreme $\mathbf{E}X_{(n)}$

This section reviews results on the rate of growth of the expected extreme $\mathbf{E}X_{(n)}$ in n. Depending upon what is known about the distribution of X, various asymptotic estimates or bounds can be provided. The most general result is [Dow95] that $\mathbf{E}X_{(n)}$ is a monotone increasing and concave function of n and that $\mathbf{E}X_{(n)} = o(n)$ provided $\mathbf{E}X$ is finite [Dow90a].

More can be said, however. Better bounds on the growth of $\mathbf{E}X_{(n)}$ with n allow us to use (2-8) to quantify the size of this ratio. If, as assumed in this paper, X has a finite variance, then $\mathbf{E}X_{(n)} = o(n^{\frac{1}{2}})$. These and other bounds on $\mathbf{E}X_{(n)}$ are summarized in Table 1 below. First we review definitions of regular variation used in many of the results in the table.

Regularly Varying Functions. Regularly varying functions are those that scale homogeneously for large argument. In this paper we are interested in distribution functions $F(t)$ for which the complementary d.f. $1 - F(t)$—denoted here by $\bar{F}(t)$—is a regularly varying function with negative index. However, we will begin with the general definition of a regularly varying function, and define rapid variation, index of variation, and classes of random variables related to these concepts.

DEFINITION. A positive measurable function $f : (0, \infty) \to (0, \infty)$ is *regularly varying* at infinity if for all $\lambda > 1$, the limit

$$\lim_{x \to \infty} \frac{f(\lambda x)}{f(x)} \tag{3-1}$$

exists and is in $(0, \infty)$. f is *rapidly varying* if the limit exists and is 0 or ∞.

The fundamental result about regular variation [Bin87, Theorem 1.4.1] is that *if* the (finite or infinite) limit (3-1) exists for all $\lambda > 1$, then there is an extended real number ρ, $-\infty \le \rho \le \infty$ such that

$$\forall \lambda > 0 \qquad \lim_{x \to \infty} \frac{f(\lambda x)}{f(x)} = \lambda^{\rho} . \tag{3-2}$$

This ρ is called the *exponent* or *index of variation*. If (3-2) holds, so that f is regularly varying with index ρ, we write $f \in R_{\rho}$. Thus with the understanding $\lambda^{-\infty} = 0$ and $\lambda^{\infty} = \infty$, $R_{-\infty}$ and R_{∞} are the rapidly varying functions.

Functions like $x^a + \sin x$ where $a > 0$ are in R_a. Functions like $\exp(-x^k)$, $k > 0$ belong to $R_{-\infty}$ and their reciprocals belong to R_{∞}. Functions that fail to have a well-defined limit (finite or infinite) in (3-2) are neither regularly varying nor rapidly varying. Examples are $\sin x$ and $e^{-\lfloor x \rfloor}$. By historical convention, the class R_0 is called the *slowly varying* functions. It includes, for example functions like $(\ln x)^{\alpha}$ for any $\alpha \geq 0$, and their reciprocals.

From the results above, it is evident that $f \in R_{\rho}$ if and only if there is some slowly varying function l such that $f(x) = x^{\rho} \cdot l(x)$. It is easy to see that if $f \in R_{\rho}$ and $g \sim f$ then $g \in R_{\rho}$. Thus asymptotic equality $\sim$ is the natural equivalence relation on the class of regularly varying functions.

In applications to random variables, we say that $X =_d F$ is a *regularly varying random variable of index* $-\alpha$, for some $\alpha \geq 0$, provided $\overline{F}(t) \in R_{-\alpha}$. In this case we write $X \in R_{-\alpha}$. For example, if $X =_d F(t)$ where $F(t) = 1 - (\ln t)^{-1}$ for $t \in [e, \infty)$, then $X \in R_0$.

With these preliminaries, results about the behavior of $\mathbf{E}X_{(n)}$ for large n are summarized in Table 1 below.

Asymptotic Bounds for $n \to \infty$		
d.f. hypotheses	$\mathbf{E}X_{(n)}$ *bound*	*Note*
$\mathbf{E}X < \infty$	$o(n)$	1
$\mathbf{E}X^p < \infty$ $(p > 1)$	$o(n^{1/p})$, $\mu + \| X - \mu \|_p \cdot n^{1/p}$	2
$X \in R_{-\alpha}$ $(\alpha > 1)$	$\sim \Gamma(1 - \alpha^{-1}) c_X(n)$ $(c_X(n) \in R_{1/\alpha})$	3
$X \in R_{-\infty}$	$\sim c_X(n)$ $(c_X(n) \in R_0)$	4
$X \in \mathcal{E}^1$	$O(\ln n)$	5
$X \in \mathcal{E}^\beta$	$O(\ln n)^\beta$	6

Notes:

(1). An example d.f. in this class is $\bar{F}(t) = t^{-1}(\ln t)^{-2}$ for $t \geq e$.

(2). See [Arn85] and [Dow90a]. The o bound assumes a fixed parent d.f. F that is independent of n, while the (weaker) bound allows F to depend on n, but has the advantage of explicit coefficients. Here $\| Y \|_p := (\mathbf{E}[Y^p])^{1/p}$ is the L^p norm.

(3). The characteristic maximum function $c_X(n)$ is (roughly) the solution to $\bar{F}(x) = n^{-1}$; see [Res87]. An example is the empirical distribution of Unix process times [Lel86], found to be $\bar{F}(t) = 0.241 t^{-1.122}$. This yields $c_X(n) = 0.281 n^{0.891}$ and so $\mathbf{E}X_{(n)} \sim 0.281 n^{0.891}$. As another example, if $\bar{F}(t) = t^{-2}(\ln t)^{-1}$ for $t \geq e$, then $\mathbf{E}X_{(n)} \sim \sqrt{\pi n / \ln n}$.

(4). Any d.f. in the domain of attraction $\mathcal{D}(\Lambda)$ of the double exponential extreme value distribution belongs to $R_{-\infty}$ [Res87]. For example, the exponential d.f. $\bar{F}(t) = e^{-t/\mu}$ yields $\mathbf{E}X_{(n)} \sim \mu \ln n$.

(5). See [Dow90b]. $\mathcal{E}^1$ is the class of random variables dominated in convex ordering [Ros83, Sto83] by some exponentially distributed variate. $\mathcal{E}^1$ is a very large class of random variables, including all those [Dow90b] that are Coxian or of Phase (*PH*) type, those with bounded mean residual life, those that are New Better Than Used in Expectation (*NBUE*), and all subclasses [Ros83] of the latter, such as Increasing Failure Rate (*IFR*) and Increasing Likelihood Ratio (*ILR*) variates.

(6). $\mathcal{E}^\beta$ is the class of variates X^β for some X in $\mathcal{E}^1$. An example is the Weibull $\bar{F}(t) = \exp(-\lambda t^{1/\beta})$.

Table 1. Summary of $\mathbf{E}X_{(n)}$ Bounds

References

[Arn85] B.C. ARNOLD, 1985. *p*-norm bounds on the expectation of the maximum of a possibly dependent sample, *J. Multivariate Anal. 17*, 316-332.

[Ber97] K.A. BERMAN and J.L. PAUL. 1997. *Fundamentals of Sequential and Parallel Algorithms*, PWS Publishing Company, Boston, MA.

[Bin87] N.H. BINGHAM, C.M. GOLDIE and J.L. TEUGELS, 1987. *Regular Variation*, volume 27 of *The Encyclopedia of Mathematics and Its Applications*, Cambridge University Press, Cambridge, England.

[Dow90a] P.J. DOWNEY, 1990. Distribution-free bounds on the expectation of the maximum with scheduling applications, *Operations Research Letters 9*, 189-201.

[Dow90b] P.J. DOWNEY and R.S. MAIER, 1991. Stochastic orderings and extreme moment growth, Technical Report 90-9, Department of Computer Science, University of Arizona, Tucson, AZ.

[Dow91] P.J. DOWNEY and R.S. MAIER, 1991. Stochastic orderings and the mean growth of extremes, Technical Report 91-13, Department of Computer Science, University of Arizona, Tucson, AZ.

[Dow93] P.J. DOWNEY and R.S. MAIER, 1993. Orderings arising from expected extremes, with an application. *Stochastic Inequalities*, IMS Lecture Notes - Monograph Series, vol. 22, Institute of Mathematical Statistics, Hayward, CA, pp. 66-75.

[Dow95] P.J. DOWNEY, 1995. Bounds and approximations for overheads in the time to join parallel forks, *ORSA Journal on Computing 7*, 125-139.

[Lel86] W.E. LELAND and T.J. OTT, 1986. Load-balancing heuristics and process behavior, *Performance Evaluation Review 14,* 1(May 1986), 54-69.

[Res87] S.I. RESNICK, 1987. *Extreme Values, Regular Variation and Point Processes*, Springer-Verlag, New York, NY.

[Ros83] S.M. ROSS, 1983. *Stochastic Processes*, Wiley, New York, NY.

[Sto83] D. STOYAN, 1983. *Comparison Methods for Queues and Other Stochastic Models*, Wiley, New York, NY.