
Bounds and Approximations for Overheads in the Time to Join Parallel Forks

Peter J. Downey

TR 92-10

ABSTRACT

This paper studies the effects of overheads in massively parallel processing. The execution times for tasks on individual processors are modeled as independent and identically but arbitrarily distributed random variables. The time to execute a process fork is assumed to be distributed exponentially. The main result bounds (in expectation) the overhead time in forking a large number of tasks across n machines and then waiting for the join event. The model used is appropriate for massive parallelism (when n is large): in fact the bound serves as a heavy traffic limit approached as $n \to \infty$ and for task times that are large in comparison to the time to execute a fork. In this model, the expected total time to reach the final join consists of a forking overhead that grows linearly with the number of processors n, a time for parallel execution of tasks that decreases in n, and finally a synchronization delay that is a concave sublinear function of $\rho = \mathbf{E}X / \mathbf{E}A$, which is the ratio of expected task time to the expected time needed to fork a new process. This overhead function is typically no worse that $o(\rho)$. An interesting aspect of the analysis is that the original problem reduces to a previously studied problem in queueing theory: estimating total delay in an infinite-server resequencing system. The new results thus provide new bounds and heavy-traffic approximations in the theory of $M/G/\infty$ resequencing queues.

April 2, 1992
Revised August 7, 1992

Department of Computer Science

The University of Arizona

Tucson, Arizona 85721

Bounds and Approximations for Overheads in the Time to Join Parallel Forks

1. Introduction

The most common parallel programming paradigm used in applications is known as *fork and join* or *fork and wait.* Processing begins in a single control thread, when a parent process creates *n* child processes; the parent and child processes each continue to execute the same body of code, resuming after the return point of the process creation call (a `fork()` system call) within their process. Each child process executes some task code, using knowledge of its own process identifier to schedule itself correctly and to access the correct partition of problem data. When a child finishes its task, it terminates with an `exit()` system call. The parent busy-waits until all child processes have exited, thence resuming a single thread of control. A generic example of this paradigm is illustrated in Figure 1.1, as it might appear in a typical Unix system with `fork()`, `wait()` and `exit()` system calls. Difficulties such as error returns and the handling of signals during a process `wait()` have been ignored in this simplified example.

Since `fork()` and `wait()` are low-level synchronization primitives, they need not be used in the restrictive way illustrated in Figure 1.1. However, this disciplined pattern of use, or "paradigm", is commonly seen in both the organization of operating systems code as well as in scientific applications [Bra89]. One important reason for the ubiquity of this pattern comes from the presence in high-level programming languages of the parallel execution construct

`cobegin` `task(1)` ‖ `task(2)` ‖ $\cdots$ ‖ `task(n)` **`coend`**.

A construct like this is used to specify concurrent execution of the given tasks in a wide variety of languages, including Algol 68, CSP and SR [And91]. Such a parallel construct is typically *implemented* through *n* `fork()` calls by the parent process, which then busy-waits for all *n* children to call `exit()` [And91]. This implementation of **`cobegin`** through system calls is structurally similar to Figure 1.1.

This paper examines the performance of the fork and wait paradigm (and therefore the performance of **`cobegin`**) on a multiprocessor with *ample servers*, i.e., the assumption is made that each `fork()` creates a new process running on a newly available processor. Since there is no contention for the use of a processor, no process will ever be required to queue, and queue waiting time will play no role in contributing to systems overhead. This liberal idealization is made since it is our purpose to study the effect of overheads arising from forking and synchronization alone, and to understand how these overheads trade against the benefit of parallel execution of tasks. Other models [Bac85, Bac89b, Kim89] have studied queueing delays caused by forking across a fixed number of processors. We also make the assumption that the `task(i)` represent the performance of independent, non-interfering and non-communicating processes.

An *overhead* results whenever any processor executes instructions that are not part of the `task()` code; i.e., any instructions not accounted for in the sequential execution

```
task(1);task(2); ... ;task(n);
```

We also use "overhead" to refer to the time delays suffered in executing these instructions.

Overheads in parallel programming are of two kinds: explicit and implicit. An overhead is *explicit* if it occurs even when the code is executed on a single processor without the presence or influence of other concurrent processes. For example, each `fork()` call executed by the parent in Figure 1.1 is an explicit overhead, as well as each of the `if` tests executed by parents and children. The time to perform a `wait()` call in the absence of any living children is also an explicit overhead; only the time spent busy waiting inside the call is due to implicit overhead.

```
main ()
{
    int n, j, id, status;

    ... computation in parent process ...

/*  create n additional processes to perform task(1),...,task(n).
    Assign id = 0 to the parent (caller of the main routine)
    and assign id=1,2,..., n to each new child process.
    Uses system call fork(), which creates one additional
    process, returning 0 to the new child and an arbitrary integer
    process-id to the calling parent     */

    id = 0;    /* parent */

    for (j = 1; j <= n; j++)
        if (fork() == 0 ) {
            id = j;
            break; /* child proceeds immediately to task */
        }

    /* child with id>=1 performs task(id)  */

    if (id != 0)  task(id);

    /* children terminate on return from task(id).
        parent spins on system call wait(&status),  a call that
        returns -1 only when all n children have died   */

    if (id == 0 )  /* parent waits for all children to terminate */
        while ( wait(&status) != -1 );
    else
        exit(0); /* child terminates immediately */

     ... computation in parent process ...
}

void task(int id)
{
     ... work to be done by process indexed  id  ...
}
```

Figure 1.1: The generic *fork and wait* paradigm in C with Unix system calls.

An overhead is *implicit* if caused by interference or intercommunication between concurrently executing processes. For example the time spent by the parent process busy-waiting inside the `wait()` calls of the `while` loop is one kind of implicit overhead called *synchronization overhead*. Other possible sources of implicit overhead are communication delays or queueing delays spent competing for an available processor; these kinds of implicit delay will not occur in our ample server model. All implicit delays result from the effects of interacting processes and are the most difficult to assess, since they are system-wide effects and cannot be directly computed from the program code and knowledge of instruction timings. Implicit overheads caused by queueing delay are the major subject of performance analysis on uniprocessors. Implicit overheads due to synchronization delay make performance analysis on multiprocessors interestingly different and challenging. The major focus of this work is to assess implicit synchronization delays for the fork and wait paradigm, and their contribution to the completion time.

The significant explicit overheads in Figure 1.1 arise from the `fork()` calls. If a `fork()` call actually creates a new process on a new processor, rather than reallocating an existing one, the call might require several milliseconds on a typical 1 MIP (million instructions per second) processor [Bra89]. Experiments on the Sequent Symmetry multiprocessor indicate that process creation takes 13-14 milliseconds. Process creation involves such substantial overhead from the need to manage memory tables and create a new address space. If all needed processes are pre-created and "parked" while busy-waiting until dispatched on a `fork()` request, the call can return in tens of microseconds [Bra89]. Sequent experiments indicate that 280-300 microseconds suffice for parked processor allocation. This is still a substantial overhead, equivalent to hundreds of instructions on a 1 MIP processor. The other explicit overheads in Figure 1.1 are trivial by comparison, and will be assumed either negligible or absorbed into other variables of our model.

1.1. Deterministic Model

We begin by modeling the total workload deterministically, assuming that the execution time of `task(i)` is X_i, and that the time to execute the ith call to `fork()` is A_i, where $i = 1, 2, \cdots, n$. The time to execute all other instructions in Figure 1.1 will be assumed negligible, or to be absorbed into the A_i or X_i intervals. The total time to complete the parallel construct, also known as the *time to join*, is the time interval in the parent process from the beginning of the `for` loop until the busy-waiting `while` loop is exited. We denote this period of time T_n. A time-space diagram illustrating the dependence of T_n upon the given data is provided in Figure 1.2.

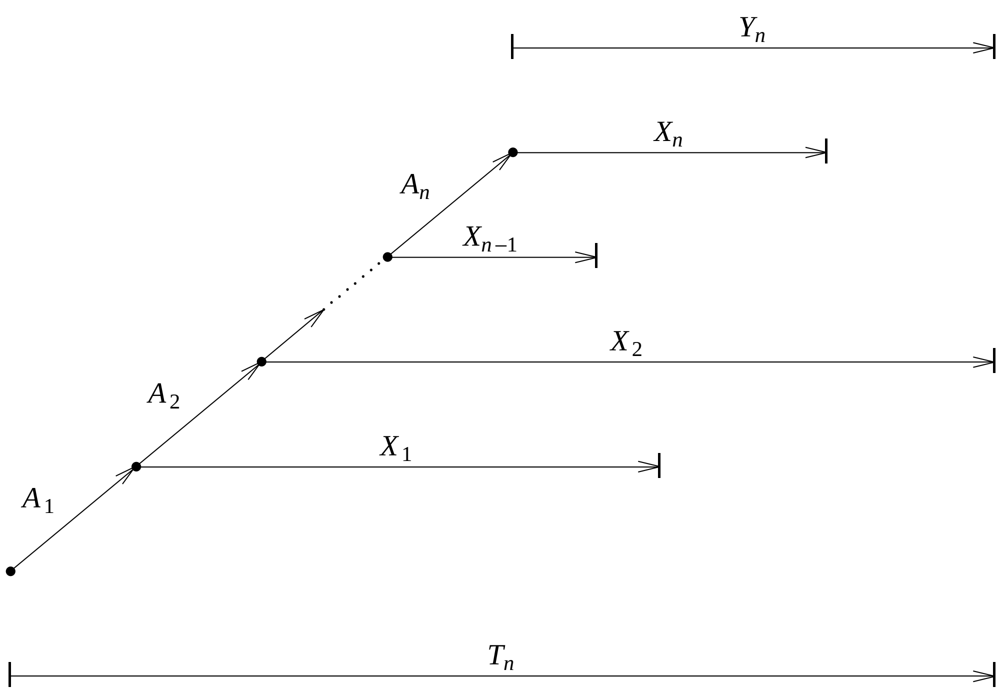

Figure 1.2: Time to join T_n with n forked child processes. The parent process runs along the top edge of the figure. Y_n is the end-to-end resequencing delay (Section 3).

The times $X_1, \cdots, X_n$ making up the workload are regarded in a deterministic model not as a sample, but as a complete description of a finite population of size n. For this workload, we define the population mean

$$\bar{m} := \frac{1}{n}\sum_{i=1}^{n} X_i \tag{1.1}$$

and the population variance

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{m})^2 \ . \tag{1.2}$$

Here s is the standard deviation of the workload population.

How does T_n depend upon the workload characteristics? Naturally its exact value will depend upon the relative sizes of overheads and task times associated with particular processors. To provide useful generalizations, we look for bounds on T_n that are representable in terms of simple workload characteristics such as $\bar{m}$ and s.

Instantaneous Forking. To see the source of and effect of implicit overheads alone, we first assume that $A_i = 0$ for all i. Then it is evident that

$$T_n = X_{(n)} \tag{1.3}$$

where $X_{(n)} := \max(X_1, \ \cdots \ , X_n)$ is the maximum of the workload population.

In the case of perfect balancing of tasks across the n processors, we would have all X_i equal to $\bar{m}$, so that $s = 0$ and $T_n = \bar{m}$. Define the speed-up ratio by

$$S(n) := \frac{\sum_{i=1}^{n} X_i}{T_n} \ .$$

Only in the perfect balance case can a perfect speed-up of $S(n) = n$ occur.

But perfect balance is not achieved in practice; there is always some variation in the population. Samuelson's [Sam68] famous result is that, in a finite population of n values $X_1, \ \cdots \ , X_n$, no value can lie more than $\sqrt{n-1}$ standard deviations from the population mean:

$$\max_{1 \le i \le n} |X_i - \bar{m}| \le s\sqrt{n-1} \ . \tag{1.4}$$

Half a dozen different proofs of this pleasant and extremely general inequality are provided in [Arn89]. Applied to the values in (1.3), Samuelson's inequality yields the bound

$$T_n \le \bar{m} + s\sqrt{n-1} \ . \tag{1.5}$$

This bound is tight for any fixed constants $\bar{m}$ and s, since equality is achieved for the worst-case data set $X_n = \bar{m} + s\sqrt{n-1}$ and $X_{n-1} = X_{n-2} = \ \cdots \ = X_1 = \bar{m} - 1/\sqrt{n-1}$.

The interpretation of (1.5) is that for all workloads with a fixed $\bar{m}$ and s, the time to join T_n can grow as the square root of the number of processors, but no worse. Here the effect of load imbalance in contributing to synchronization delays is evident. Even assuming a fixed average task time and fixed non-zero variability in the workload, implicit overheads can grow as the degree of concurrency of the program grows—in the worst case. Removal of explicit overheads does not eliminate synchronization delay, which depends upon imbalance s and the number of processors n. In the worst case, speed-up can be as low as

$$S(n) = \frac{n\bar{m}}{\bar{m} + s\sqrt{n-1}} \sim \frac{\bar{m}}{s}\sqrt{n} \ , \quad n \to \infty \ ;$$

this implies a processor efficiency $E(n) := S(n)/n$ which declines as $O(n^{-1/2})$.

Explicit Overheads. Next we include the explicit overhead times. Define

$$\bar{a} := \frac{1}{n}\sum_{i=1}^{n} A_i \ ,$$

with $\bar{a} > 0$, to be the average of the fork times in the workload population. Then reference to Figure 1.2 makes clear that

$$T_n = \max(A_1 + X_1, \ A_1 + A_2 + X_2, \ \cdots \ , A_1 + A_2 + \cdots + A_n + X_n) \ . \tag{1.6}$$

In the best case, this yields $T_n = \bar{a}n$. In the worst case the final child forked will be given the longest task, so that

$$T_n \le n\bar{a} + X_{(n)} \ . \tag{1.7}$$

For a perfectly balanced workload this is $n\bar{a} + \bar{m}$, but in the extremal case over all workloads with fixed $\bar{m}$ and s, using (1.4) again gives

$$T_n \le n\bar{a} + \bar{m} + s\sqrt{n-1} \ . \tag{1.8}$$

The bound is tight for the extremal workload example given above. The speed-up in the worst case is

$$S(n) = \frac{\bar{m}n}{\bar{a}n + \bar{m} + s\sqrt{n-1}} = \frac{\bar{m}}{\bar{a}} \left(1 - \frac{s}{\sqrt{n}} + O(\frac{1}{n}) \right) , \qquad n \to \infty \ . \tag{1.9}$$

Even in the best case, speed-up is $S(n) = \bar{m}/\bar{a}$. Achievable speed-up is therefore severely limited by the explicit cost of forking, and to first order is insensitive to the number of processors used as $n \to \infty$. Workload imbalance effects and the influence of n appears in the second order term. Since efficiency declines as $E(n) = O(1/n)$, there is little motivation for using large numbers of processors.

1.2. Stochastic Model

The general deterministic bound in (1.8) is an unsatisfactory model for understanding the performance of real computation for several reasons.

- While (1.8) does provide a robust bound, there is no way to argue how tight the bound is: there are no notions of equilibrium or heavy traffic available in a deterministic model.
- The bound is pessimistic in that it is achieved only under extreme worst-case assumptions. To achieve the bound, assignment of tasks to processors must be unlucky, and the workload population must actually change with each value of n, rather than comprise an unbiased sample from a fixed parent population. The bound (1.8) is therefore the product of a "malevolent adversary" assumption rather than the result of a "disinterested adversary" assumption about the source of workloads. In this sense, it is a game-theoretic bound rather than a decision-theoretic bound [Fer67], in which the opposing player is an intelligent optimizing agent rather than "nature".
- The deterministic nature of the bound prevents us from understanding how T_n is affected by other characteristics of the workload, such as the distribution of long tasks. It therefore prevents useful generalizations from being drawn by classifying workload properties.

The above drawbacks are ameliorated by the use of a *stochastic workload model*. In such a model $X_1, \cdots, X_n$ (and $A_1, \cdots, A_n$) are not themselves a population, but are *sampled* from a fixed, underlying population distribution of work assumed to exist *a priori*. The sample values are assumed to be *independent*. We again look for bounds (random variables or their expectations) that depend on simple descriptive parameters of the underlying distribution.

The main result of the paper, Theorem 6.1, gives an estimate for the expected value $\mathbf{E}T_n$ of the random variate T_n in a stochastic model in which the task times X_i are from a virtually arbitrary distribution with finite variance, and in which forking times A_i are independent and exponentially distributed. The estimate of Theorem 6.1 is both an upper bound for $\mathbf{E}T_n$, and a bound which is approached closely from below in heavy traffic conditions, i.e., when both n and the ratio $\mathbf{E}X/\mathbf{E}A$ are large.

Organization of the paper is as follows. Most of the terminology and background results needed in the sequel are laid out in Section 2, including information about expected extremes which turn out to play a large role in the proof. Section 3 describes how the gist of the problem—estimation of Y_n—can be reduced to the well-studied problem of calculating end-to-end delay in a queueing system with resequencing. Section 3 contains a useful internal monotonicity property regarding end-to-end delay in $GI/G/\infty$ resequencing systems. Section 5 contains a new result (Theorem 5.4) giving a heavy traffic bound and approximation for expected end-to-end delay in an $M/G/\infty$ resequencing system. This result is proved using an interesting asymptotic estimate of independent interest (Theorem 5.3). Section 6 assembles the work of preceding sections into the final result (Theorem 6.1). In Section 7 we interpret the resulting estimate to see its implications on speed-up and on the trade-off between process granularity and number of processors used.

2. Preliminaries

This section summarizes definitions and properties of the major tools used in the sequel. It may be passed over at first reading and referred back to as the occasion requires.

2.1. Notation

Unless otherwise indicated, all random variables are non-negative.

If random variables X and Y are identically distributed, we write $X =_d Y$. Similarly, we use the shorthand $X =_d F$ to mean that X has distribution function F. Where there may be confusion, we denote the distribution function $\mathbf{P}[Z \le x]$ of a random variable Z by a subscript: $F_Z(x)$. The complementary distribution function $1 - F_Z(x)$ of Z is denoted $\bar{F}_Z(x)$. The (generalized) inverse of $\bar{F}_Z(x)$ is defined by

$$\bar{F}_Z^{\leftarrow}(y) = \inf \{x : \bar{F}_Z(x) \le y\} .$$

Expectations are denoted $\mathbf{E}$. A frequently used fact is that for a non-negative random variable $Z =_d F$

$$\mathbf{E}Z^p = \int_0^\infty x^p \, dF(x) = p\int_0^\infty x^{p-1} \bar{F}(x) dx . \tag{2.1}$$

When it is necessary to emphasize the d.f. with respect to which an expectation is taken, a subscript is employed. For example, $\mathbf{E}_Z(X + Z)$ is equivalent to the random variable $\mathbf{E}Z + X$.

Two standard notions of stochastic convergence [Chu74] are used in the sequel. a sequence of random variables $X_n =_d F_n(x)$ is said to *converge in distribution* to a random variable $X =_d F(x)$ provided that for all x

$$\lim_{n \to \infty} F_n(x) = F(x) .$$

If this is the case we write $X_n \xrightarrow{D} X$. We say that X_n *converges in moment* to X provided $\lim_{n \to \infty} \mathbf{E}X_n = \mathbf{E}X$. This is often written $\mathbf{E}X_n \to \mathbf{E}X$.

2.2. The Renewal Process and Equilibrium

Let $\{X_i : i = 1, 2, \cdots \}$ be a sequence of nonnegative independent random variables with $X_i =_d X$ having common d.f. F. To avoid trivialities, assume that $F(0) < 1$. We interpret X_i as the time between the $(i-1)$th and ith epochs (renewals) of a stochastic process. For convenience, the time origin is considered a renewal. The X_i are the renewal periods. Assume $\mu = \mathbf{E}X < \infty$. The time epoch of the $(n+1)$st renewal (cf. Figure 2.1) is defined as S_n, where

$$S_0 = 0 ,$$

$$S_n = \sum_{i=1}^{n} X_i , \quad n \ge 1 .$$

The biased random walk $\{S_n\}$ defines the *renewal process* induced by F. The *renewal counting process* is defined as the number of renewals in $[0, t]$: $N(t) = \inf \{n : S_n > t\}$.

The remaining time in the renewal period $(S_{N(t)-1}, S_{N(t)}]$ intercepted by t is defined as:

DEFINITION. The *excess* of the renewal process at t is the random variable $X(t) = S_{N(t)} - t$.

The distribution of $X(t)$ can be complex to describe for arbitrary t. However, as $t \to \infty$ a fundamental result of renewal theory provides that, assuming X is non-lattice and has finite mean, $X(t)$ converges in distribution to a very simple *equilibrium excess* variate X^*.

DEFINITION. A random variable is called a *lattice* if it only takes on integral multiples of some nonnegative number p, called the *period*.

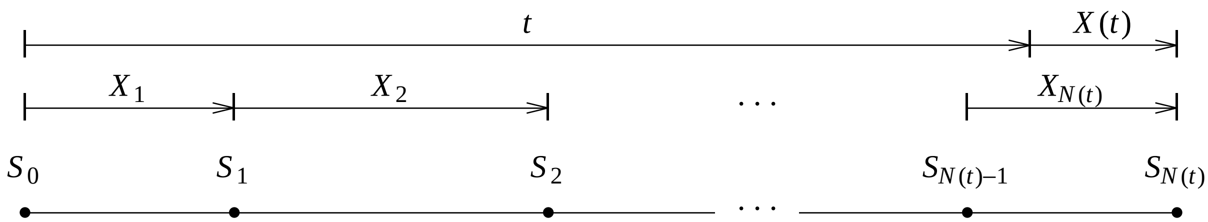

Figure 2.1: The renewal process and the excess $X(t)$ at time t.

If X has a density, it cannot be a lattice.

PROPOSITION 2.1 [Cin75, Ros83]. If the renewal period $X =_d F$ is not a lattice and $\mathbf{E}X < \infty$ then $X(t) \xrightarrow{D} X^*$ as $t \to \infty$, where

$$F_{X^*}(x) \;=\; \frac{\int_0^x \overline{F_X}(u)\,du}{\mathbf{E}X}\;. \tag{2.2}$$

Distributional convergence does not imply convergence in moment. However $\mathbf{E}X(t) \to \mathbf{E}X^*$ if and only if X has a finite second moment, and in that case it follows from (2.2) that

$$\mathbf{E}X^* = \frac{\mathbf{E}[X^2]}{2\,\mathbf{E}X}\;. \tag{2.3}$$

2.3. Stochastic Orderings

A number of useful notions of ordering among random variables occur in the literature [Sto83, Bac89a]. The two found most useful are defined below.

DEFINITION. Let X and Y be non-negative random variables. The ordering $X \le_d Y$, pronounced X is *stochastically less than* Y, and the ordering $X \le_c Y$, pronounced X is *less variable than* Y, are defined by

$$X \le_d Y \quad \Leftrightarrow \quad \overline{F_X}(t) \le \overline{F_Y}(t) \quad \text{for all} \quad t \ge 0\,, \tag{2.4}$$

$$X \le_c Y \quad \Leftrightarrow \quad \int_t^\infty \overline{F_X}(x)dx \le \int_t^\infty \overline{F_Y}(x)dx \quad \text{for all} \quad t \ge 0\,,$$

where the latter ordering is defined only if the expectations $\mathbf{E}X$ and $\mathbf{E}Y$ exist.

It is immediate that

$$X \le_d Y \;\Rightarrow\; X \le_c Y \;\Rightarrow\; \mathbf{E}[X^p] \le \mathbf{E}[Y^p] \tag{2.5}$$

for all $p \ge 1$, provided the expectations exist.

PROPOSITION 2.2 [Ros83]. If X_1, X_2, ..., X_n are independent and Y_1, Y_2, ..., Y_n are independent, and $X_i \le_d$ [resp. $\le_c$] Y_i, $1 \le i \le n$, then

$$g(X_1, \cdots, X_n) \;\le_d \text{[resp. } \le_c \text{]}\; g(Y_1, \cdots, Y_n)$$

whenever g is increasing [resp. increasing and convex].

2.4. Extremes

If $X_1, \cdots, X_n$ are iid random variables with d.f. $F(x)$, we write $X_{(n)}$ for their extreme $\max(X_1, \cdots, X_n)$. Then the d.f. of $X_{(n)}$ is $F(x)^n$. For any $p > 0$, if $\mathbf{E}X^p < \infty$, then $\mathbf{E}X_{(n)}^p < \infty$.

For any non-negative random variable X with finite expectation, the expected extreme is:

$$\mathbf{E}X_{(n)} = \int_0^\infty x\, nF(x)^{n-1}\, dF(x) \;=\; \int_0^\infty (1 - F^n(x))\, dx\;. \tag{2.6}$$

This expression for $\mathbf{E}X_{(n)}$ can be naturally extended to any real $n \in [0, \infty)$; each such integral exists since $\mathbf{E}X < \infty$. Note that $\mathbf{E}X_{(0)} = 0$, since $F^0(x)$ is the unit step at zero. With this real extension, the overall qualitative behavior of the expected extreme is reported below: $\mathbf{E}X_{(n)}$ is increasing and concave in n.

Let $\mathbf{D}f(n)$ represent differentiation with respect to real argument n.

DEFINITION [Wid71]. A real function $f(n)$ is *completely monotone* in argument n on the interval (a, ∞), written $f \in$ c.m.(a, ∞), if it is infinitely differentiable on (a, ∞) and for all $k \geq 0$

$$(-1)^k \mathbf{D}^k f(n) \geq 0 ,$$

for all $n \in [a, \infty)$. If in addition $f(a+) < \infty$, then $f \in$ c.m.$[a, \infty)$.

THEOREM 2.3. Assume $\mathbf{E}X < \infty$. Then $\mathbf{DE}X_{(n)}$ is c.m.$(0, \infty)$, implying that $\mathbf{E}X_{(n)}$ is monotone non-decreasing and concave for $n > 0$.

Proof: Let $h(n) := \mathbf{DE}X_{(n)}$. By differentiating the last integral in (2.6) repeatedly with respect to n we have that for $k \geq 0$ and for $n \in (0, \infty)$

$$(-1)^k \mathbf{D}^k h(n) = (-1)^k \mathbf{D}^{k+1} \mathbf{E}X_{(n)} = \int_0^\infty F^n (-\ln F)^{k+1} \, dx \geq 0 , \qquad (\dagger)$$

which establishes the complete monotonicity of $h(n)$ for $n \in (0, \infty)$, provided that the integrals on the right (one for each k) are uniformly convergent for all n in an interval $[\delta, \infty)$, for each positive $\delta > 0$. To establish uniform convergence, pick some $\delta \in (0, 1)$. We will show that for all $k \geq 0$, and for each $n \in [\delta, \infty)$, each integral is bounded by a convergent integral that is independent of n. Now since $F^\delta \geq F^n$ for $n \geq \delta$

$$\int_0^\infty F^n (-\ln F)^{k+1} \, dx \leq \int_0^\infty F^\delta (-\ln F)^{k+1} \, dx = \frac{1}{\delta^{k+1}} \int_0^\infty F^\delta (-\ln F^\delta)^{k+1} \, dx .$$

This integral is unform for all $n \in [\delta, \infty)$. It remains to show that the last integral above is convergent. Define the d.f. $G := F^\delta$ and for convenience define $m := k+1$. We must show that the integral

$$\int_0^\infty G(-\ln G)^m \, dx$$

is convergent for all integer $m \geq 1$. A little manipulation shows that this is the same integral as

$$m^m \int_0^\infty [\, G^{1/m} (-\ln G^{1/m}) \,]^m \, dx$$

It is elementary to see that

$$\forall\, u \in (0, 1] \qquad u(-\ln u) \leq (1-u) .$$

Applying this inequality to the preceding integral gives

$$\int_0^\infty G(-\ln G)^m \, dx \leq m^m \int_0^\infty (1 - G^{1/m})^m \, dx \leq m^m \int_0^\infty (1 - G^{1/m}) \, dx = m^m \int_0^\infty (1 - F^{\delta/m}) \, dx .$$

Now since $\delta < 1$, $1 - F^{\delta/m} \leq 1 - F$, and the last integral above is bounded by

$$m^m \int_0^\infty (1 - F) \, dx = m^m \cdot \mathbf{E}X .$$

By assumption, this is convergent. To sum up, we have shown that for any $\delta \in (0, 1)$, then for all $n \in [\delta, \infty)$ and any $k \geq 0$

$$\int_0^\infty F^n (-\ln F)^{k+1} \, dx \leq \frac{(k+1)^{k+1}}{\delta^{k+1}} \cdot \mathbf{E}X .$$

Since the integrals in question are convergent, and what is more converge uniformly for n in intervals of the form $[\delta, \infty)$, differentiation under the integrals (†) is justified for any n in the union of these intervals, namely $(0, \infty)$. Since each integrand is non-negative, complete monotonicity in (†) is assured. □

Obviously $\mathbf{E}X_{(n)} \le \mathbf{E}X \cdot n$, but sharper results exist that bound the rate of increase of $\mathbf{E}X_{(n)}$ as a concave function of n:

PROPOSITION 2.4 [Arn89, Dow90]. If $\mathbf{E}X^p < \infty$ for some $p > 0$, then $\mathbf{E}X_{(n)} = o(n^{1/p})$.

2.5. Regular Variation

To a great extent the asymptotic behavior of $\mathbf{E}X_{(n)}$ for large n is controlled by the rate of decay of the upper tail of F_X. One method for classifying upper tails of d.f.s is to use the notions of regular and rapid variation. Regularly varying functions are those that scale homogeneously for large argument.

DEFINITION. A measurable function $U : \mathbb{R}^+ \to \mathbb{R}$ is *regularly varying* at infinity if for all $\lambda > 1$, the limit

$$\lim_{x \to \infty} \frac{U(\lambda x)}{U(x)} \tag{2.7}$$

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

The fundamental result about regular variation [Bin87] is that if the limit (2.7) exists for all $\lambda > 1$, then there is a real number α, $0 \le \alpha \le \infty$ such that

$$\forall \lambda > 1 \qquad \lim_{x \to \infty} \frac{U(\lambda x)}{U(x)} = \lambda^{\alpha} . \tag{2.8}$$

This α is called the *exponent* or *index of variation*. If (2.8) holds, we write $U \in R_\alpha$. Thus $R_{-\infty}$ and R_∞ are the rapidly varying functions. Functions like $\exp(-x^k)$, $k > 0$ belong to $R_{-\infty}$ and their reciprocals belong to R_∞.

By historical convention, the class R_0 is called the *slowly varying* functions. It includes, for example functions like $(\ln x)^k$ for any k. From the results above, it is evident that $U \in R_\alpha$ if and only if there is some slowly varying L such that $U(x) = x^{\alpha} \cdot L(x)$.

Let $F(x)$ be a d.f. Then its complementary d.f. $\bar{F}(x)$ is monotone non-increasing. If $\bar{F}$ is slowly, regularly or rapidly varying, then it has index that is 0, negative or $-\infty$. This property of a d.f. controls the behavior of the extreme moment by the following result.

DEFINITION. For a random variable X, the *characteristic maximum* function is defined as follows:

$$c_X(n) = \bar{F}_X^{\leftarrow}(n^{-1}) . \tag{2.9}$$

The significance of the characteristic maximum function is given by the following

PROPOSITION 2.5 [Pic68]. Let $X =_d F$ be a non-negative random variable with $\bar{F} \in R_{-\alpha}$, $1 < \alpha \le \infty$. Then $\mathbf{E}X_{(n)} < \infty$ and

$$\mathbf{E}X_{(n)} \sim \Gamma(1 - \alpha^{-1}) \cdot c_X(n) \qquad , \ n \to \infty . \tag{2.10}$$

Gnedenko [Gne43] showed that in the regular ($\alpha < \infty$) case, $X_{(n)}/c_X(n) \xrightarrow{D} \Phi_\alpha$ where the limit Φ_α has the d.f. $\exp(-x^{-\alpha})$. No general limit distribution result holds for the $R_{-\infty}$ case, though an analogous result holds for certain subclasses [Gne43, Res87].

3. Reduction to a Resequencing System

Consider a sequence of customers $1, 2, \cdots n, \cdots$ arriving at a service system. The interarrival period between customers $n-1$ and n is described by the amount of time A_n, where A_1 is the interval from the time origin to the first arrival; for convenience we define $A_0 := 0$. Once arrived, no particular order need be maintained among customers within the system; they may be disordered by, e.g., overtaking or parallel service. Customer n is after arrival delayed by some time D_n, termed the *disordering delay*. This delay may arise from the need for service, from time spent waiting in service queues, or from a combination of both. After completion of the delay, customer n is held in a *resequencing buffer* until customers 1, 2, $\cdots$, $n-1$ have departed from the buffer, at which time n departs from the resequencing system. Thus customer departures from the resequencing buffer and hence from the system occur in the same order as customer arrivals. The time spent by n between arrival and departure instants is called the *end-to-end delay* Y_n. The time spent by n waiting in the resequencing buffer is the *resequencing delay* $R_n := Y_n - D_n$. See Figure 3.1.

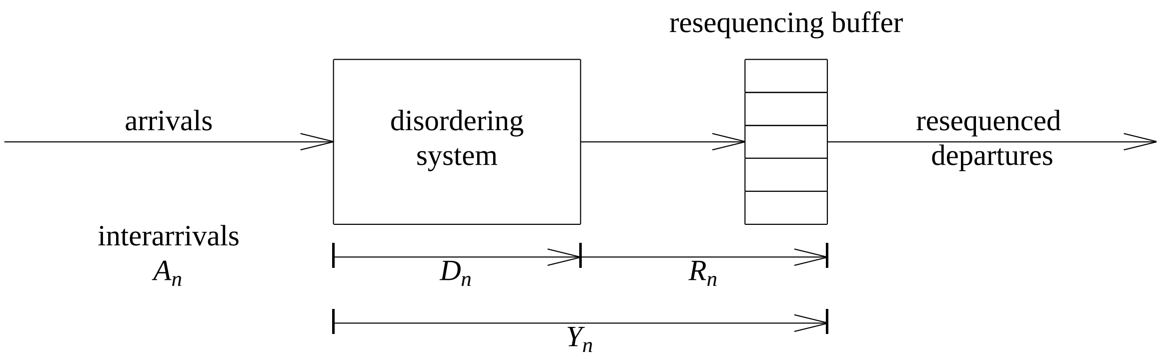

Figure 3.1. The general resequencing system.

A fundamental result in resequencing systems is the sample path recurrence describing Y_n.

LEMMA 3.1 [Bac84 , Var87].

$$\begin{aligned} &Y_0 = 0 , \\ &Y_{n+1} = \max(X_{n+1},\ Y_n - A_{n+1}) . \end{aligned} \tag{3.1}$$

Proof: Let a_{n+1} denote the arrival instant of customer $n+1$. This customer cannot leave until customer n (and hence all its predecessors) has departed, which will not occur until $a_n + Y_n = a_{n+1} + Y_n - A_{n+1}$. This customer also cannot leave until its own service delay is complete, i.e., until $a_{n+1} + X_{n+1}$. Taking the largest of these two times shows that the departure moment is $a_{n+1} + \max(X_{n+1},\ Y_n - A_{n+1})$. $\square$

We may now relate the time to join T_n to the end-to-end delay Y_n. Refer to Figure 1.2. If we identify task time X_n with the disordering delay in a resequencing system, the end-to-end delay Y_n emerges as the quantity upon which analysis of T_n depends. The associated R_n describes the component of T_n due to synchronization overhead.

THEOREM 3.2. Let X_n be the disordering delays in a general resequencing system with end-to-end delay Y_n. Then the time to join can be expressed as

$$T_n = A_1 + \cdots + A_n + Y_n . \tag{3.2}$$

Proof: By induction on n. From (1.6) or Figure 1.2 it is apparent that the following recurrence holds for T_n:

$$\begin{aligned} &T_0 = 0 , \\ &T_{n+1} = \max(X_{n+1} + A_1 + \cdots + A_{n+1},\ T_n) , \quad n \ge 0 . \end{aligned} \tag{3.3}$$

Clearly $T_1 = X_1 + A_1 = Y_1 + A_1$, so the base of (3.2) is established. For the induction step, assume that $T_n = a_n + Y_n$, where we define $a_n := A_1 + \cdots + A_n$. Then by (3.3)

$$\begin{aligned} T_{n+1} &= \max(X_{n+1} + a_n + A_{n+1},\ T_n) = \max(X_{n+1} + a_n + A_{n+1},\ a_n + Y_n) \\ &= a_n + A_{n+1} + \max(X_{n+1},\ Y_n - A_{n+1}) = a_{n+1} + Y_{n+1}, \end{aligned}$$

where the last equality follows from Lemma 3.1 setting $D_n := X_n$. This establishes the induction step. $\square$

The remainder of the analysis focuses upon deriving the distribution of Y_n, along with bounds and approximations for it, under simplifying assumptions about the structure of the disordering delays and interarrival distribution.

4. The $GI/G/\infty$ Resequencing System

We now make the added assumption that customers arrive at a service system having an arbitrary number of processors (an *ample servers* or *pure delay* system). The interarrival periods A_n are assumed to be independent and identically distributed (iid) with $A_n =_d A$. Customer n is dispatched upon arrival to a new server without queueing and is delayed by some time X_n in service. The service periods X_n are assumed iid with $X_n =_d X$. The sequences $\{A_n\}$ and $\{X_n\}$ are assumed to be independent sequences. Therefore the disordering delay $D_n = X_n$ in this special case is independent of the arrival pattern.

The next result shows that the $GI/G/\infty$ system's end-to-end delay has an *internal monotonicity property*: Y_n is monotone non-decreasing in n with respect to stochastic ordering.

THEOREM 4.1. In a $GI/G/\infty$ resequencing system,

$$\forall n \geq 0 \qquad Y_n \leq_d Y_{n+1} \ . \tag{4.1}$$

Proof: Since there is no queueing in an infinite server system, delays are independent of arrivals. Since the arrival sequence and disordering delay sequence are independent, the set of random variables

$$\{X_{i+1},\ Y_i,\ A_{i+1}\} \tag{4.2}$$

is mutually independent for every i.

The proof proceeds by induction on n. It is clear that the basis holds since $Y_1 = X_1 \geq_d 0 = Y_0$. As induction hypothesis, assume that for some $n \geq 0$

$$Y_n \geq_d Y_{n-1} \ .$$

Since $-A_{n+1} =_d -A_n$ and $+$ is a increasing function, we have by independence (4.2) and Proposition 2.2 that

$$Y_n - A_{n+1} \geq_d Y_{n-1} - A_n \ .$$

By (4.2) the set of random variables $\{X_{i+1},\ Y_i - A_{i+1}\}$ is independent for every i. Since $X_{n+1} =_d X_n$ by hypothesis, another use of Proposition 2.2 implies, along with Lemma 3.1, that

$$Y_{n+1} = \max(X_{n+1},\ Y_n - A_{n+1}) \geq_d \max(X_n,\ Y_{n-1} - A_n) = Y_n \ .$$

This completes the induction. □

For a $GI/G/\infty$ resequencing system, define the traffic intensity by $\rho := \mathbf{E}X/\mathbf{E}A$. In [Bac84] it is shown that, for any finite ρ, the end-to-end delay is stable. That is, Y_n converges in distribution to a proper (almost surely finite) random variable Y^* which represents the equilibrium end-to-end delay. Therefore, Y^* acts as an upper stochastic bound for Y_n, a bound which is approached as $n \to \infty$. This stability result along with the preceding theorem yields immediately:

COROLLARY 4.2. In a $GI/G/\infty$ resequencing system, with $A_n =_d A$ and $X_n =_d X$, if $\mathbf{E}A > 0$ and $\mathbf{E}X < \infty$, Y_n converges in distribution from below to an equilibrium random variable Y^*:

$$\forall n \geq 0 \qquad Y_n \leq_d Y^* \ , \tag{4.3}$$

and $Y_n \xrightarrow{D} Y^*$, $\qquad n \to \infty$.

Next we consider the expectation of the end-to-end delay. It is possible for the sequence Y_n to converge to an rv that does not have a first moment. If Y^* has a finite expectation, however, the sequence of expectations approaches it in limit from below

COROLLARY 4.3. In a $GI/G/\infty$ resequencing system, with $A_n =_d A$ and $X_n =_d X$, if $\mathbf{E}A > 0$ and $\mathbf{E}X < \infty$, then

$$\forall n \geq 0 \qquad \mathbf{E}Y_n \leq \mathbf{E}Y^* \ . \tag{4.4}$$

If in addition $\mathbf{E}Y^* < \infty$, then

$$\mathbf{E}Y_n = \mathbf{E}Y^* + o(1) \qquad \text{as } n \to \infty \ . \tag{4.5}$$

Proof: Since $\mathbf{E}X$ is finite, so is $\mathbf{E}X_{(n)}$ for all n. An easy induction using Lemma 3.1 establishes that $Y_n \le_d X_{(n)}$, so that $\mathbf{E}Y_n \le \mathbf{E}X_{(n)}$; thus all expectations $\mathbf{E}Y_n$ are finite. It follows by Theorem 4.1 and Corollary 4.2 that $\mathbf{E}Y_n \le \mathbf{E}Y_{n+1} \le \mathbf{E}Y^*$. If in addition $\mathbf{E}Y^*$ is finite, then the sequence $\mathbf{E}Y_n$ is uniformly bounded and $Y_n \xrightarrow{D} Y^*$. Under these conditions it is well-known [Chu74, Theorem 4.5.2] that convergence in distribution implies convergence in moment. This establishes (4.5). □

5. End-to-End Delay in the $M/G/\infty$ Resequencing System

As a final assumption in the development of the model, we assume that the interarrival rv A is distributed exponentially with mean a:

$$\overline{F_A}(x) = e^{-x/a} \ .$$

The disordering subsystem is then an $M/G/\infty$ queueing system. Harrus and Plateau [Har82] solved for the distribution of resequencing delay and various other parameters for this resequencing system. Their work built on an earlier solution of the $M/M/\infty$ case by Kamoun, Kleinrock and Muntz [Kam81]. For a generalization of this model, where output from the $M/G/\infty$ resequencing system is fed into a single-server queue with general service time, Bacelli, Gelenbe and Plateau [Bac84] gave conditions for stability at equilibrium, and derived closed-form solutions for the Laplace transform of the equilibrium end-to-end delay.

In this section we provide a simple, direct argument establishing the distribution of Y^* for the $M/G/\infty$ resequencing system. While this result follows from the results of Harrus and Plateau [Har82], our restatement here shows that Y^* can be regarded as a mixture of independent extreme order statistics, and allows simple expressions for the moments to be derived. It is therefore of benefit in applications to the moments of the join time T_n.

Below we write $X^*_{(n)}$ for $(X^*)_{(n)}$, i.e., the maximum of n iid variates with equilibrium distribution $=_d X^*$.

THEOREM 5.1. In a $M/G/\infty$ resequencing system, with $\rho := \mathbf{E}X/\mathbf{E}A$

$$F_{Y^*}(x) = \sum_{n=0}^{\infty} e^{-\rho} \frac{\rho^n}{n!} F_{X^*}(x)^n \cdot F_X(x) \ , \tag{5.1}$$

that is, Y^* is a mixture of maxima

$$Y^* =_d \sum_{n=0}^{\infty} e^{-\rho} \frac{\rho^n}{n!} \max(X^*_{(n)},\ X) \ . \tag{5.2}$$

Proof: Condition on a random (Poisson) arrival at equilibrium that finds n customers in the system. Under the conditioning, this arrival intercepts n independent equilibrium renewal processes with renewal period $=_d X$. Since the arrival is random, standard renewal theory considerations imply that the time to completion of a renewal period on the ith server is given by an equilibrium excess rv X^*_i with d.f. (2.1). Thus the end-to-end delay of the new arrival is

$$\max(X,\ X^*_1,\ \cdots\ ,\ X^*_n) \ ,$$

where the subscripted variables are iid $=_d X^*$. Conditioned on n in the system at arrival, the total delay is thus distributed as $\max(X^*_{(n)},\ X)$. Removing the conditioning, using the well-known $M/G/\infty$ equilibrium state distribution [Gro85] yields the mixture in (5.2). In terms of d.f.s, (5.1) is equivalent. □

Harrus and Plateau [Har82] do not directly derive an expression of end-to-end delay Y^*. Instead they focus upon the equilibrium resequencing delay $R^* = Y^* - X$. Suitably rearranged, their result [Har82, equation (11)] is:

$$F_{R^*}(z) = \sum_{n=0}^{\infty} e^{-\rho} \frac{\rho^n}{n!} \int_0^{\infty} F_{X^*}(z+x)^n \cdot dF_X(x) \ .$$

Our (5.1) is not directly derivable from this result, since R^* and X, the service time of an arriving customer, are dependent. One cannot simply convolve F_{R^*} with F_X. A simple proof of the above expression for F_{R^*} can be given as follows. Let R^*_n be the resequencing delay experienced by an equilibrium arrival C finding n customers in the system at arrival. Conditioned on C having service time x, it follows that

$$R_n^* \le z \quad \Leftrightarrow \quad \max(X_1^*, \ \cdots \ , X_n^*) \le x + z \ ,$$

because the n customers present at the arrival of C must finish before C can depart. Using this event definition allows calculation of the d.f. of R_n^*, which is just the integral in the above equation. Removal of the conditioning on n completes the result.

We turn now to deriving an expression for the expectation of end-to-end delay.

THEOREM 5.2. In a $M/G/\infty$ resequencing system with $\mathbf{E}X^2 < \infty$, let $\rho := \mathbf{E}X/\mathbf{E}A$. Then $\mathbf{E}Y^*$ is finite and

$$\mathbf{E}Y^* = \sum_{n=0}^{\infty} e^{-\rho} \frac{\rho^n}{n!} \mathbf{E}X_{(n)}^* \ + \ \mathbf{E}A \cdot (1 - e^{-\rho}) \ . \tag{5.3}$$

Proof: Finiteness of $\mathbf{E}X^2$ implies that X^* has a finite expectation; hence $\mathbf{E}X_{(n)}^*$ is finite for all n. By (5.2) we have

$$\mathbf{E}Y^* = \sum_{n=0}^{\infty} e^{-\rho} \frac{\rho^n}{n!} \mathbf{E} \max(X_{(n)}^*, \ X) \ . \tag{5.4}$$

Since the expectations on the right are finite and $O(n)$, finiteness of $\mathbf{E}Y^*$ is guaranteed. First we show that

$$\mathbf{E} \max(X_{(n)}^*, \ X) = \mathbf{E}X_{(n)}^* \ + \ \frac{\mathbf{E}X}{n+1} \ . \tag{5.5}$$

To see this, write $Z := \max(X_{(n)}^*, \ X)$. Then

$$\overline{F_Z}(x) = 1 - F_{X^*}(x)^n \cdot F_X(x) = 1 - F_{X^*}(x)^n + F_{X^*}(x)^n \cdot \overline{F_X}(x) \ . \tag{5.6}$$

From (2.1) it follows that the d.f. of X^* has a derivative

$$\mathbf{D}F_{X^*}(x) = \overline{F_X}(x) / \mathbf{E}X \ .$$

So (5.6) can be expressed as

$$\overline{F_Z}(x) = 1 - F_{X^*}(x)^n + \frac{\mathbf{E}X}{n+1} \cdot \mathbf{D}F_{X^*}(x)^{n+1} = \overline{F_{X_{(n)}^*}}(x) + \frac{\mathbf{E}X}{n+1} \cdot \mathbf{D}F_{X_{(n+1)}^*}(x) \ . \tag{5.7}$$

Integrating (5.7) over $(0, \ \infty)$ results in

$$\mathbf{E}Z = \mathbf{E}X_{(n)}^* \ + \ \frac{\mathbf{E}X}{n+1} \cdot \int_0^{\infty} dF_{X_{(n+1)}^*}(x) = \mathbf{E}X_{(n)}^* \ + \ \frac{\mathbf{E}X}{n+1} \ .$$

This establishes (5.5). Putting (5.5) into (5.4) and summing $e^{-\rho}\rho^n/n! \cdot \mathbf{E}X/(n+1)$ results in (5.3). $\square$

Thus the end-to-end delay has behavior controlled by the expected extreme of equilibrium excesses. In fact, there is a very simple upper bound for (5.3) which is asymptotically approached from below in heavy traffic (i.e., as $\rho \to \infty$). First we need a general property of the expected maximum of a Poisson-distributed number of terms: such an extreme behaves like an expected maximum of ρ terms, where ρ is the Poisson mean.

THEOREM 5.3. For any non-negative random variable Z having a non-trivial d.f. and such that $\mathbf{E}Z < \infty$, define

$$I(\rho) := \sum_{n=0}^{\infty} e^{-\rho} \frac{\rho^n}{n!} \mathbf{E}Z_{(n)} \ . \tag{5.8}$$

Note that $I(\rho) = \mathbf{E}_N[\mathbf{E}_Z Z_{(N)}]$ where N has a Poisson distribution with rate ρ. Then

(a) $I(\rho)$ is strictly increasing in ρ and

$$\forall \rho \ge 0 \quad I(\rho) \le \mathbf{E}Z_{(\rho)} \ . \tag{5.9}$$

(b) $I(\rho)$ approaches $\mathbf{E}Z_{(\rho)}$ in the limit:

$$I(\rho) = \mathbf{E}Z_{(\rho)} + o(1)\ , \qquad \rho \to \infty\ . \tag{5.10}$$

Proof: (a). By Theorem 2.3 $\mathbf{E}Z_{(n)}$ is concave for $n > 0$, and we have by Jensen's inequality [Chu74, p. 47] that for all $\rho > 0$

$$I(\rho) = \mathbf{E}_N[\mathbf{E}_Z Z_{(N)}] \le \mathbf{E}_Z Z_{(\mathbf{E}N)} = \mathbf{E}_Z Z_{(\rho)} = \mathbf{E}Z_{(\rho)}\ .$$

This bound trivially holds for $\rho = 0$. Write F for the d.f. of Z and assume $F(0) < 1$. Direct differentiation then yields $I'(\rho) > 0$. This establishes the bound and monotone property.

(b). Define $f(n) = \mathbf{E}Z_{(n)}/n$. Then

$$I(\rho) = \rho \sum_{n=0}^{\infty} e^{-\rho} \frac{\rho^n}{n\,!} f(n+1) = \rho \sum_{n=0}^{\infty} e^{-\rho} \frac{\rho^n}{n\,!} \int_0^{\infty} yF(y)^n\, dF(y)$$

or after interchanging the integral and sum

$$I(\rho) = \rho \int_0^{\infty} y e^{-\rho \bar{F}}\, dF\ .$$

Define

$$\iota(\rho) := I(\rho)/\rho = \int_0^{\infty} y e^{-\rho \bar{F}}\, dF\ . \tag{5.11}$$

It is immediate that

$$\mathbf{D}^k \iota(\rho) = (-1)^k \int_0^{\infty} y e^{-\rho \bar{F}} \bar{F}^k\, dF\ , \tag{5.12}$$

provided that the integrals (5.12) converge uniformly for $\rho \ge 0$. Uniform convergence is obvious since each integral in (5.12) is dominated for all ρ by

$$\int_0^{\infty} y\, dF\ ,$$

which converges since $\mathbf{E}Z < \infty$.

The following two Lemmata explore the asymptotic behavior of these derivatives of ι.

LEMMA 5.3.1. For all integer $k \ge 1$,

$$\int_0^{\infty} e^{-\rho \bar{F}} \bar{F}^k\, dy = o\left(\frac{1}{\rho^{k-1}}\right)\ . \tag{5.13}$$

Proof of Lemma: Break the integral into the sum of an integral over $(0, \ln\rho]$ and a second over $(\ln\rho, \infty)$. To bound the first integral, use the fact that

$$\forall\, u \ge 0 \qquad e^{-u} u^k \le e^{-k} k^k\ , \tag{5.14}$$

so that

$$e^{-\rho \bar{F}} \bar{F}^k \le \frac{e^{-k} k^k}{\rho^k}\ .$$

Thus

$$\int_0^{\ln\rho} e^{-\rho \bar{F}} \bar{F}^k\, dy \le \frac{e^{-k} k^k}{\rho^k} \int_0^{\ln\rho} dy = \frac{e^{-k} k^k}{\rho^k} \ln\rho = o\left(\frac{1}{\rho^{k-1}}\right)\ .$$

To bound the second integral, apply (5.14) again to get

$$e^{-\rho \bar{F}} \bar{F}^{k-1} \le \frac{e^{-(k-1)} (k-1)^{k-1}}{\rho^{k-1}}\ ,$$

so that

$$\int_{\ln\rho}^{\infty} e^{-\rho\bar{F}}\bar{F}^k\,dy \le \frac{e^{-(k-1)}(k-1)^{k-1}}{\rho^{k-1}}\int_{\ln\rho}^{\infty}\bar{F}\,dy = o(\frac{1}{\rho^{k-1}}) \ .$$

The last integral is $o(1)$ because $\ln\rho \to \infty$ and $\mathbf{E}Z < \infty$ by assumption.

Since both integrals are $o(\rho^{-(k-1)})$, Lemma 5.3.1 follows. □

LEMMA 5.3.2. For all integer $k \ge 0$,

$$\mathbf{D}^k\iota(\rho) = (-1)^k\int_0^{\infty} y e^{-\rho\bar{F}}\bar{F}^k\,dF = o(\frac{1}{\rho^k}) \ . \qquad (5.15)$$

Proof of Lemma: By induction on k.

Basis: From (5.9) in part (a) of the present Theorem, we conclude that $\iota(\rho) \le \mathbf{E}Z_{(\rho)}/\rho$. Since $\mathbf{E}Z$ is finite, $\mathbf{E}Z_{(\rho)} = o(\rho)$ by Proposition 2.4. Thus $\iota(\rho) = o(1)$.

Step: Assume the result is true for $k-1 \ge 0$ and consider the kth derivative. Using integration by parts on the integral in question,

$$\mathbf{D}^k\iota(\rho) = \int_0^{\infty} y e^{-\rho\bar{F}}(-\bar{F})^k\,dF = \frac{1}{\rho}\int_0^{\infty} y(-\bar{F})^k e^{-\rho\bar{F}}(-\rho\,d\bar{F}) = \ = -\frac{1}{\rho}\int_0^{\infty} e^{-\rho\bar{F}}d[y(-\bar{F})^k]$$

$$= -\frac{1}{\rho}\int_0^{\infty} e^{-\rho\bar{F}}(-\bar{F})^k\,dy + \frac{(-1)^k k}{\rho}\int_0^{\infty} y e^{-\rho\bar{F}}\bar{F}^{k-1}\,dF = \frac{(-1)^{k-1}}{\rho}\int_0^{\infty} e^{-\rho\bar{F}}\bar{F}^k\,dy - \frac{k}{\rho}\mathbf{D}^{k-1}\iota(\rho) \ .$$

Applying Lemma 5.3.1 to the first term and the induction hypothesis to the second term, we conclude that

$$\mathbf{D}^k\iota(\rho) = \frac{(-1)^{k-1}}{\rho}\cdot o(\frac{1}{\rho^{k-1}}) - \frac{k}{\rho}\cdot o(\frac{1}{\rho^{k-1}}) = o(\frac{1}{\rho^k}) \ .$$

This completes the proof of Lemma 5.3.2. □

We turn now to the proof of (5.10). In view of the definitions of $\iota(\rho)$ and $f(n)$, (5.10) is equivalent to the assertion $\iota(\rho) - f(\rho) = o(\rho^{-1})$, or equivalently

$$\iota(\rho+1) - f(\rho+1) = o(\rho^{-1}) \ . \qquad (5.16)$$

Lemma 5.3.2 and Taylor's theorem imply that $\iota(\rho+1) = \iota(\rho) + o(\rho^{-1})$. Therefore (5.10) and hence (5.16) is equivalent to

$$\iota(\rho) - f(\rho+1) = o(\rho^{-1}). \qquad (5.17)$$

In integral form, (5.17) is equivalent to

$$\int_0^{\infty} y e^{-\rho\bar{F}}\,dF - \int_0^{\infty} y F^{\rho}\,dF = o(\frac{1}{\rho}) \ . \qquad (5.18)$$

The remainder of the argument is devoted to establishing (5.18).

Let $J(\rho)$ denote the left hand side of (5.18). Then

$$J(\rho) = \int_0^{\infty} y\{\, e^{-\rho\bar{F}} - F^{\rho}\,\}\,dF \ .$$

A well-known bound is

$$\forall\, 0 \le z \le x \qquad 0 \le e^{-z} - (1 - z/x)^x \le \frac{z^2 e^{-z}}{x} \ . \qquad (5.19)$$

Using (5.19) with z set to $\rho\bar{F}$ and x set to ρ yields

$$0 \le e^{-\rho\bar{F}} - F^{\rho} \le \rho\bar{F}^2 e^{-\rho\bar{F}} \ .$$

Applying this bound in the integrand of $J(\rho)$ results in the bounds

$$0 \le J(\rho) \le \int_0^\infty y\rho \bar{F}^2 e^{-\rho \bar{F}}\, dF = \rho \mathbf{D}^2 \iota(\rho) .$$

Since $\mathbf{D}^2\iota(\rho) = o(\rho^{-2})$ by Lemma 5.3.2, we can conclude that $J(\rho) = o(\rho^{-1})$, and the objective set in equation (5.18) is attained. This completes the proof of Theorem 5.3. □

Asymptotic estimates similar to (5.10) have been proven for more general Poisson-weighted sums of regularly varying sequences (not involving extremes) in [Bin83, Teu77].

The previous two results can be combined to give a heavy traffic bound for expected end-to-end resequencing delay.

THEOREM 5.4 (*$M/G/\infty$ Heavy-Traffic Bound*). In an $M/G/\infty$ resequencing system, with non-lattice service random variable X and exponential interarrival random variable A, let $\rho := \mathbf{E}X/\mathbf{E}A$. Assume $\mathbf{E}X^2 < \infty$ and let $\mathbf{E}Y^*(\rho)$ denote the end-to-end delay at equilibrium.

(a) $\mathbf{E}Y^*(\rho)$ is a strictly increasing function of ρ, and

$$\forall \rho \ge 0 \qquad \mathbf{E}Y^*(\rho) \le \mathbf{E}X^*_{(\rho)} + \mathbf{E}A\cdot(1 - e^{-\rho}) . \tag{5.20}$$

(b) The bound is approached as the heavy-traffic limit, i.e.,

$$\mathbf{E}Y^*(\rho) = \mathbf{E}X^*_{(\rho)} + \mathbf{E}A + o(1) , \qquad \rho \to \infty . \tag{5.21}$$

Proof: Since $\mathbf{E}X^2 < \infty$, we have that $\mathbf{E}X^* < \infty$. Because X is non-lattice, X^* cannot have a trivial d.f. Apply Theorem 5.3 to the sum in Theorem 5.2 with Z set to X^*. □

The following specific examples will give a feeling for the quality of the general bounds in Theorem 5.4 by comparison with exact calculations.

EXAMPLE 5.5. (*Exponential Delays*.) The end-to-end delay in an $M/M/\infty$ system in which X is assumed to be $EXP(1/\mathbf{E}X)$ has been solved [Kam81] to give

$$\mathbf{E}Y^*(\rho) = \mathbf{E}X\cdot \sum_{n=0}^{\infty} e^{-\rho} \frac{\rho^n}{n!} H_{n+1}$$

where $H_n = \sum_1^n i^{-1}$ is the nth harmonic number. It is shown in [Cho90] that this is equivalent to

$$\mathbf{E}Y^*(\rho) = \mathbf{E}X\cdot \mathrm{Ein}(\rho) + \mathbf{E}A\cdot(1 - e^{-\rho}) ,$$

where the *exponential integral* [Abr68, p. 228] is defined by

$$\mathrm{Ein}(z) = \int_0^z (1 - e^{-t}) \frac{dt}{t} .$$

It is known that [Abr68, §5.1.39]

$$\mathrm{Ein}(z) = \ln z + \gamma + \frac{e^{-z}}{z}\cdot\left(1 + O(z^{-1})\right) , \qquad z \to \infty$$

where γ is Euler's constant. Therefore, to exponentially small terms in ρ we have essentially an exact solution for the equilibrium end-to-end delay in an $M/M/\infty$ resequencing system:

$$\mathbf{E}Y^*(\rho) = \mathbf{E}X\cdot \ln\rho + \mathbf{E}X\cdot\gamma + \mathbf{E}A + O(e^{-\rho}) , \qquad \rho \to \infty . \tag{5.22}$$

How does the exact expression (5.22) compare with the bounds given by Theorem 5.4? The bound will yield essentially the $O(\ln\rho)$ and $O(1)$ terms in the expansion (5.22), which can be seen as follows. In the exponential case, $X^* =_d X$ and the expected extreme of n exponentials is well-known [Res87] to be $\mathbf{E}X^*_{(n)} = \mathbf{E}X\cdot H_n$. H_n can be extended to the reals via the *digamma* or *psi* function $\psi(z) := \mathbf{D}\ln\Gamma(z)$; the relationship [Abr68, §6.3.2, 6.3.18] is

$$H_z := \psi(z+1) + \gamma = \ln z + \gamma + O(z^{-1}) , \qquad z \to \infty .$$

Theorem 5.4(a) then yields by (5.20) the bound

$$\mathbf{E}Y^{*}(\rho) \le \mathbf{E}X\cdot(\ln\rho + \gamma) + \mathbf{E}A + O(\rho^{-1}) \ .$$

This bound agrees with the exact value (5.22) to terms of order $O(\rho^{-1})$. It is more than a bound, however, as (5.21) implies

$$\mathbf{E}Y^{*}(\rho) = \mathbf{E}X\cdot(\ln\rho + \gamma) + \mathbf{E}A + o(1) \ , \qquad \rho \to \infty \ .$$

This asymptotic estimate agrees up to $o(1)$ terms with the exact value. □

EXAMPLE 5.6. (*Pareto Delays*.) This example discusses end-to-end delay in an $M/PAR(\beta)/\infty$ system in which X is assumed to have a Pareto distribution with parameter β:

$$F_X(x) = 1 - (1+x)^{-\beta} \ , \qquad \beta > 0 \ .$$

A $PAR(\beta)$ rv has finite pth moments only for $p \in [0, \beta)$; we confine attention to the case $\beta > 1$. Straightforward calculation (as found in [Res87] for example) establishes that for $\beta > 1$

$$\mathbf{E}X_{(n)} = \Gamma(1-\beta^{-1})\cdot n^{1/\beta} - 1 + o(1) \ .$$

Using (2.1) one directly shows that if X is $PAR(\beta)$ then X^{*} is $PAR(\beta-1)$. Assuming $\beta > 2$ gives a finite moment $\mathbf{E}X^{*} < \infty$. Direct calculation from (5.3) establishes the exact value of the expected end-to-end delay up to $o(1)$ terms:

$$\mathbf{E}Y^{*}(\rho) = \Gamma(1-(\beta-1)^{-1})\cdot n^{1/(\beta-1)} - 1 + \mathbf{E}A + o(1) \qquad (\beta > 2) \ . \tag{5.23}$$

By comparison with this directly calculated value, the general bound of Theorem 5.4 produces a result in agreement in the first term. To apply Theorem 5.4, note that $\overline{F_{X^*}} \in R_{-(\beta-1)}$ since F_{X^*} is a $PAR(\beta-1)$ distribution. By Proposition 2.5

$$\mathbf{E}X^{*}_{(n)} \sim \Gamma(1-(\beta-1)^{-1})\cdot c_{X^*}(n) \qquad , \ n \to \infty \ .$$

where the characteristic maximum function is defined in (2.9). Since the d.f. of X^{*} is known, we can directly calculate the characteristic maximum as

$$c_{X^*}(n) = n^{1/(\beta-1)} - 1 \ .$$

Applying (5.21) gives

$$\mathbf{E}Y^{*}(\rho) \sim \Gamma(1-(\beta-1)^{-1})\cdot\rho^{1/(\beta-1)} \qquad , \ \rho \to \infty \ . \tag{5.24}$$

The estimate (5.24) agrees with the exact result (5.23) in leading term.

As a specific example, for the case $X =_d PAR(3)$, application of Theorem 5.4 results in $\mathbf{E}Y^{*}(\rho) \sim \sqrt{\pi\rho}$. Theorem 5.4 will not apply to the case $X =_d PAR(2)$, since this has a finite mean but no finite second moment, and so $\mathbf{E}X^{*}$ and $\mathbf{E}Y^{*}$ are undefined.

6. The Expected Time to Join

Having reduced the problem of calculating the time to join T_n to that of calculating end-to-end resequencing delay, we now apply the results derived concerning the latter quantity to the task of bounding and approximating $\mathbf{E}T_n$.

THEOREM 6.1. Let task times be distributed independently according to a random variable X, and let fork times be distributed independently according to random variable A. Assume that A is distributed exponentially and that X is non-lattice with $\mathbf{E}X^2 < \infty$. Let $\rho := \mathbf{E}X/\mathbf{E}A$ and let $T_n(\rho)$ be the time to join defined by (1.6).

(a) $\mathbf{E}T_n(\rho)$ is increasing with n and

$$\forall\,\rho,\ n \qquad \mathbf{E}T_n(\rho) \le \mathbf{E}A\cdot n \ + \ \mathbf{E}X^{*}_{(\rho)} \ + \ \mathbf{E}A\cdot(1-e^{-\rho}) \ , \tag{6.1}$$

(b) The bound is approached in the limit for large n and ρ, i.e.

$$\mathbf{E}T_n(\rho) = \mathbf{E}A\cdot n + \mathbf{E}X^{*}_{(\rho)} + \mathbf{E}A + o(1), \quad \text{as both } n,\ \rho \to \infty \ . \tag{6.2}$$

where the $o(1)$ term goes to zero as both n and ρ go to infinity.

Proof: By Theorem 3.2, $\mathbf{E}T_n(\rho) - \mathbf{E}A \cdot n = \mathbf{E}Y_n$. For ρ large and fixed, Corollary 4.3 establishes that this difference is increasing with n and has bound $\mathbf{E}Y^*(\rho)$. As n becomes sufficiently large, Corollary 4.3 shows that the difference $\mathbf{E}Y^*(\rho) - \mathbf{E}Y_n$ becomes arbitrarily small, say smaller than $1/\rho$. Now Theorem 5.4 provides estimates for $\mathbf{E}Y^*(\rho)$ that have error terms dominating $1/\rho$, and which are valid for sufficiently large ρ. Thus (6.2) is valid if both ρ and n are sufficiently large. □

7. Conclusion

For the model of massive parallelism developed in Section 1, equation (6.2) shows that, under mild assumptions regarding the d.f. of the task time X, and assuming exponential forking times with mean $\mathbf{E}A$, the expected overall time to a join event is approximated up to vanishingly vanishing terms by

$$\mathbf{E}T_n = (n+1) \cdot \mathbf{E}A + \mathbf{E}X^*_{(\rho)} + o(1) . \tag{7.1}$$

This approximation is valid when n is large (large workload) and when ρ is large. Since $\rho = \mathbf{E}X/\mathbf{E}A$, ρ will be large in situations where the average overhead for a single fork call is small in relation to the running time of a task that is forked (large granularity). Therefore the heavy traffic assumptions represent an operating region of considerable interest.

The forking overhead $n \cdot \mathbf{E}A$ in (7.1) is unsurprising, arising from the "linear forking" paradigm in which a single parent forks all child processes. The remaining terms are of greater interest. The extra $\mathbf{E}A$ can be interpreted up to small terms as $\mathbf{E}X/\mathbf{E}[N+1]$, where N is the Poisson distribution of the number of tasks in the system at equilibrium (cf. equation (5.5)); here $\mathbf{E}N = \rho$. The remaining contributions to $\mathbf{E}T_n$ beyond the forking overheads include both expected service as well as expected synchronization delays, and can be re-expressed as:

$$\mathbf{E}T_n - n \cdot \mathbf{E}A = \mathbf{E}X^*_{(\mathbf{E}N)} + \mathbf{E}X/\mathbf{E}[N+1] + o(1) . \tag{7.2}$$

This sum can be interpreted as follows. At an equilibrium forking of a new task, on the average there are $\mathbf{E}N$ processors busy in parallel, each executing for a residual task time distributed as X^*. Therefore the total time summed across all active $\mathbf{E}[N+1]$ processors that elapses before the new task completes is $\mathbf{E}[N] \cdot \mathbf{E}X^*_{(\mathbf{E}N)} + \mathbf{E}X$. Dividing this total system time by $\mathbf{E}[N+1]$ gives the average time experienced by the new task at equilibrium. The resulting quotient agrees with (7.2) up to terms of order $o(1)$, as may be verified by dividing through by $\mathbf{E}[N+1]$ and using the fact that $\mathbf{E}X^*_{(\mathbf{E}N)}/\mathbf{E}N = o(1)$.

As Proposition 2.4 indicates, $\mathbf{E}X^*_{(\rho)}$ is a slowly growing function of ρ, no worse than $o(\rho)$ as long as $\mathbf{E}X^2 < \infty$. For many types of distribution function, it is at most polylogarithmic. Discussion of the slow growth of expected extremes occurs in [Dow90, Dow91].

We can apply (7.1) to the estimation of speed-up in expectation achievable as $n \to \infty$. It is instructive to compare the result with that from the deterministic "worst-case" model of (1.9). For purposes of comparison, let us use $\mu := \mathbf{E}X$ for expected task length and $a := \mathbf{E}A$ to denote the expected time of a single fork call. Then if μ/a is assumed large enough so that our estimates apply:

$$S(n) = \frac{\mu n}{an + \mathbf{E}X^*_{(\mu/a)} + a + o(1)} = \frac{\mu}{a} \cdot \left[1 - \frac{\mathbf{E}X^*_{(\mu/a)} + a}{n} + o\left(\frac{1}{n}\right) \right] \qquad n \to \infty . \tag{7.3}$$

The speed-up estimate for the stochastic model agrees in first term with that of the deterministic model (1.9) in both the best and worse cases. This is again unsurprising because of the linear growth in fork costs. The next highest order correction term decays faster in n than the analogous term in (1.9); as in the deterministic case, characteristics of the task population distribution of X as well as the value of n influence this term. However the coefficient of this term increases with μ/a, suggesting a trade-off between n and ρ.

To explore this trade-off, note that $\rho = \mu/a$, and that in practice value of a may be taken as fixed. We therefore wish to study the trade-off between μ and n, both of which may be assumed to be large. Thus μ may be thought of as a measure of the expected size of a "granule" of work dispatched to each of n processors. Suppose that the total amount of "useful work" to be accomplished is W where $W = n \cdot \mu$. How should W be allocated between n and μ? To answer this question for an illustrative example, suppose that for each choice of μ, task time X has a Pareto distribution of shape 3 with mean

μ, so that

$$\overline{F_X}(x) = \frac{(2\mu)^3}{(2\mu+x)^3} .$$

Then a simple calculation of the equilibrium excess X^* shows that

$$\overline{F_{X^*}}(x) = \frac{(2\mu)^2}{(2\mu+x)^2} ,$$

which is a Pareto of shape 2 with mean 2μ. (The mean of X^* is larger than that of X because X has a decreasing failure rate d.f.). From the calculations of Example 5.6, we have in this case, since X^* is $PAR(2)$ but scaled by mean 2μ:

$$\mathbf{E}X^*_{(n)} = 2\mu\cdot(\sqrt{\pi n} - 1 + o(1)) \qquad n \to \infty .$$

Applying (7.1) we have for this example

$$\mathbf{E}T_n = na + 2\mu\cdot\sqrt{\pi\mu/a} - 2\mu + a + o(1) \qquad \mu,\ n \to \infty .$$

If we use the relation $W = n\mu$ this is

$$\mathbf{E}T_n = \frac{aW}{\mu} + 2(\frac{\pi}{a})^{1/2}\cdot\mu^{3/2} - 2\mu + a + o(1) \qquad \mu,\ W \to \infty . \tag{7.4}$$

Since the first term decreases in μ and the the second increases, there is an optimum value of μ, say $\hat{\mu}$, and an optimum number of processors $\hat{n} = W/\hat{\mu}$, that minimizes $\mathbf{E}T_n$ for this example. Minimization of the right side of (7.4) yields

$$\hat{\mu} = (a^3/9\pi)^{1/5}\cdot W^{2/5} + O(W^{1/5}) \qquad W \to \infty .$$

and

$$\hat{n} = (9\pi/a^3)^{1/5}\cdot W^{3/5} + O(W^{2/5}) \qquad W \to \infty .$$

We see that for this particular type of task time distribution, the optimum number of processors to allocate grows as $W^{3/5}$ in the workload W. An attempt to use more (or fewer) processors will result in a greater total expected time to join. The total expected time to join at this optimum operating point is

$$\mathbf{E}T_{\hat{n}} = 5(a^2\pi/9)^{1/5}\cdot W^{3/5} + O(W^{2/5}) \qquad W \to \infty .$$

It is interesting to note the effect of the overhead parameter a in the above quantities. An increase in a will dramatically favor larger granularity $\hat{\mu}$ over the number $\hat{n}$ of processors used.

Similar calculations can be made for any random variable for which an approximate expression for $\mathbf{E}X^*_{(n)}$ is available. Qualitatively, the results are similar to the conclusions in the above example, though the exact trade-off between granularity and processors will differ in details.

There are several open problems suggested by the results of this paper. Several directions are possible in generalizing Theorem 6.1. The most interesting is to handle less costly forking paradigms, in which the original parent gives rise to a tree of processes, each of which forks a fixed number of children. This would give a leading term to $\mathbf{E}T_n$ of order $\Theta(\ln n)$; computing the analogue of "end-to-end delay" would appear difficult in this case.

Another generalization of the present model looks at more general distributions for the forking time A. Such a generalization would affect both the distribution of the equilibrium number of customers N as well as the role played by X^*, which will no longer measure the residual task time seen by an arrival, since arrivals no longer randomly sample the equilibrium state. Results will involve the non-equilibrium excess $X(t)$.

Finally, we may wish to restrict the number of available processors to a total of m, running several tasks sequentially on one machine when $n > m$. Analysis of this case would relate closely to the study of $GI/G/m$ resequencing queues [Bac89b]. Here bounding results involving stochastic inequalities are likely to be the most successful.

Acknowledgement

I am grateful to Robert Maier for several suggestions that considerably improved the statement and proof of Theorem 5.3.

8. References

[Abr68] Abramowitz, M. and I.A. Stegun (eds). *Handbook of Mathematical Functions*. Washington: National Bureau of Standards (1968).

[And91] Andrews, G.R. *Concurrent Programming*. Redwood City, CA: Benjamin-Cummings, 1991.

[Arn89] Arnold, B.C. and N. Balakrishnan. *Relations, Bounds and Approximations for Order Statistics*, Lecture Notes in Statistics No. 53. New York: Springer Verlag (1989).

[Bac84] Baccelli, F., E. Gelenbe and B. Plateau. An end-to-end approach to the resequencing problem. *J. ACM 31*, 3 (Jul 1984), 474-485.

[Bac85] Baccelli, F., and A.M. Makowski. Simple computable bounds for the fork-join queue. *Proc. 19th Annual Conf. Information Sciences and Systems*, The Johns Hopkins University, Baltimore, MD (Mar 1985), 536-441.

[Bac89a] Baccelli, F. and A.M. Makowski. Queueing models for systems with synchronization constraints. *Proc. IEEE 77*, 1 (Jan 1989), 138-161.

[Bac89b] Baccelli, F., A.M. Makowski, and A. Shwartz. The fork-join queue and related systems with synchronization constraints: stochastic ordering and computable bounds. *Adv. Appl. Prob. 21*, (1989), 629-660.

[Ben78] Bender, C.M. and S.A. Orszag. *Advanced Mathematical Methods for Scientists and Engineers*. New York: McGraw-Hill (1978).

[Bin83] Bingham, N.H. and J. Hawkes. On limit theorems for occupation times. In J.F.C. Kingman and G.E.H. Reuter, *Probability, Statistics and Analysis*, London Math. Soc. Lect. Note Ser. 79, Cambridge University Press (1983), 46-62.

[Bin87] Bingham N.H., C.M. Goldie and J.L. Teugels. *Regular Variation*, volume 27 of *The Encyclopedia of Mathematics and Its Applications*, Cambridge University Press, Cambridge, 1987.

[Bra89] Brawer, S. *Introduction to Parallel Programming*. New York: Academic Press (1989).

[Cho90] Chowdhury, S. Analysis of queueing systems requiring resequencing of customers. TR 90-26, Department of Computer Science, University of Arizona, August 1990.

[Cin75] Cinlar, E. *Introduction to Stochastic Processes*. Englewood Cliffs: Prentice-Hall (1975).

[Chu74] Chung, K.L. *A Course in Probability Theory, 2nd Ed.* New York: Academic Press (1974).

[Dow90] Downey, P.J. Distribution-free bounds on the expectation of the maximum with scheduling applications. *Operations Research Letters 9* (1990), 189-201.

[Dow91] Downey, P.J. and R.S. Maier. Stochastic orderings and the mean growth of extremes. TR 91-13, Department of Computer Science, University of Arizona, April 1991.

[Fer67] Ferguson, T.S. *Mathematical Statistics: A Decision Theoretic Approach*. New York: Academic Press (1967).

[Gne43] Gnedenko, B.V. Sur la distribution limite du terme d'une série aléatoire. *Ann. Math. 44*, (1943),423-453.

[Gro85] Gross, D. and G.M. Harris. *Fundamentals of Queueing Theory, 2nd Ed.* New York: Wiley, 1985.

[Har82] Harrus, G. and B. Plateau. Queueing analysis of a reordering issue. *IEEE Trans. on Software Engineering SE-8*, 2(Mar 1982), 113-123.

[Kam81] Kamoun, F., L. Kleinrock and R. Muntz. Queueing analysis of the ordering issue in a distributed database concurrency control mechanism. *Proc. 2nd International Conf. on Distributed Computing Systems*, Versailles, France, IEEE (April 1981).

[Kim89] Kim, C. and A.K. Agrawala. Analysis of the fork-join queue. *IEEE Trans. on Computers 38*, 2(Feb 1989), 250-255.

[Kle76] Kleinrock, L. *Queueing Systems, Vol. II: Computer Applications*. New York: Wiley (1976).

[Pic68] Pickands, J. Moment convergence of sample extremes. *Ann. Math. Statist. 39*, (1968), 881-889.

[Res87] Resnick, S.I. *Extreme Values, Regular Variation and Point Processes*. New York: Springer-Verlag (1987).

[Ros83] Ross, S.M. *Stochastic Processes*. New York: Wiley (1983).

[Sam68] Samuelson, P.A. How deviant can you be? *J. Amer. Statist. Assoc. 63*, (1968), 1522-1525.

[Sto83] Stoyan, D. *Comparison Methods for Queues and Other Stochastic Models*. New York: Wiley (1983).

[Teu77] Teugels, J.L. On the rate of convergence of the maximum of a compound Poisson process. *Bull. Soc. Math. Belg., Ser B 29*, II(1977), 205-216

[Var87] Varma, S. Some problems in queueing systems with resequencing. Electrical Engineering Technical Report TR 87-192, University of Maryland, College Park (1987).

[Wid71] Widder, D.V. *An Introduction to Transform Theory*. New York: Academic (1971).