ON THE THEORY OF DISTRIBUTED PROCESSING

Leonard Kleinrock

March 1985
CSD-850009

ON THE THEORY OF DISTRIBUTED PROCESSING*

LEONARD KLEINROCK
University of California, Los Angeles

ABSTRACT

We consider a distributed processing environment in which a total processing capacity is split into smaller processing units (of the same total capacity), which collectively process a stream of jobs. We study the performance ratio T *(the mean response time seen by jobs in this distributed environment) with* T_0 *(the mean response time seen by a job when it is processed in a centralized environment by a single processor). The most general configuration studied is that of a series-parallel topology. In particular, we consider* m *parallel chains, the kth of which contains* n_k *processors in series, each of capacity* C_k *operations/second. We assume that jobs arrive at the kth chain from a Poisson source at rate* λ_k *jobs/second and that each job requires an exponentially distributed number of operations from each processor. We find, for the symmetric system* ($n_k = n$, $\lambda_k = \lambda/m$), *that* $T(\rho)/T_0(\rho) = mn$. *For the general system (arbitrary* n_k*) but with equal loading on each series chain, we show that* $T(\rho)/T_0(\rho) = \sum_1^m n_k$.

We find the optimal distribution of traffic among the chains; one property of this solution is that some of the series chains carry zero traffic. When we optimize the capacity assignment, we find that $\min n_k \leq T(\rho)/T_0(\rho) \leq \sum n_k$. *When we do the joint optimization, we find that* $T(\rho)/T_0(\rho) = \min n_k$. *Distributed processing increases the mean response time! Lastly, we discuss the effect of processor costs on system performance.*

1. Introduction

We have seen a steady progression from 8-bit processors to 16-bit processors and, more recently, to 32-bit processors. The price/performance profile has been improving dramatically due to the unprecedented revolution in the integrated chip technology. As a result, computer system designers are now considering distributed processing configurations with hundreds, thousands and possibly hundreds of thousands of processors. Many interconnection and structural schemes have been proposed (ranging from precisely planned iterative arrays to randomly configured systems) in an attempt to harness the combined power of these processors in various kinds of applications.

How are we to measure the performance of a distributed processing system? Perhaps the most traditional performance profile is the mean response time (T) versus the system throughput (ρ). It is a normalized version of this measure that we adopt in this paper. Specifically, we will evaluate the ratio $T(\rho)/T_0(\rho)$ where

$T(\rho) =$ mean response time for a *distributed* processing configuration (omitting any delays due to interprocessor communication),

and

$T_0(\rho) =$ mean response time for the same job stream when it is processed by a single *centralized* processor of the same capacity as that contained in the total distributed system.

*This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-82-C-0064.

One would hope that the advantage of distributed processing would manifest itself by producing a ratio such that $T(\rho)/T_0(\rho) << 1$. For the model developed below, we show that this is definitely *not* the case and, in fact, $T(\rho)/T_0(\rho) \geq 1$. The key assumption which leads to this (apparently negative) result is that the collection of processors in the distributed system has the *same total capacity* as that of the single processor to which it is being compared. What this comparison ignores is the relative *cost* of "small" versus "large" processors. Specifically, it is currently true that a 10 MIP (millions of instructions per second) processor costs far more than ten times that of a 1 MIP processor. Thus, a more meaningful comparison would be to compare a distributed system to a centralized system of the *same cost*. We approach that problem in Section 5 below. Beyond response time, throughput and cost, there are other (more intangible) performance measures one should really introduce in any such evaluation. For example, one should consider system reliability, flexibility, availability, simplified climate control, user friendliness, etc. as factors favoring distributed systems; on the negative side, one might include increased memory requirements, inter-processor communication, etc. We do not consider these features in this paper.

2. The Model

We consider a distributed processing environment in which a total processing capacity, C (operations/second), is split into a number of smaller processing units (of the same total capacity), which collectively process a stream of jobs.

The most general configuration studied in this paper is that of a series-parallel topology as shown in Figure 2.1 below. In particular, we consider m parallel chains, the kth of which contains n_k processors in series, each of capacity C_k operations/second, such that

$$C = \sum_{k=1}^{m} n_k C_k \tag{2.1}$$

(and, at no loss of generality, we assume that $C_1 \geq C_2 \geq \cdots \geq C_m$). We assume that jobs arrive at the kth chain from a Poisson source at rate λ_k jobs/second such that

$$\lambda = \sum_{k=1}^{m} \lambda_k \tag{2.2}$$

When a job passes down the kth chain, we assume it receives $1/\mu n_k$ operations, on the average, from each of the n_k processors in that chain (i.e., a pipelined processor configuration); thus, each job receives a total number of operations, whose mean is $1/\mu$ operations/job. We further make the independence assumption [KLEI 64], i.e., that the service received at each processor is an i.i.d. exponential random variable with mean $1/\mu n_k C_k$ seconds. In all cases, we assume that

$$\rho_k = \lambda_k/\mu n_k C_k < 1 \tag{2.3}$$

so, each processor is a stable queueing system.

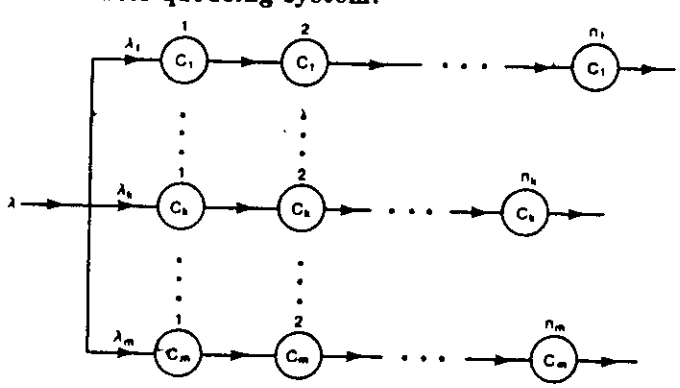

Figure 2.1: The General Series-Parallel Topology

3. Performance Evaluation

For the centralized processor, we have an M/M/1 queueing system for which the mean response time is [KLEI 75, Eq.(3.26)]

$$T_0(\rho) = \frac{\bar{x}}{1-\rho} \tag{3.1}$$

where $\bar{x} =$ E [service time] $= \dfrac{1}{\mu C}$ and $\rho = \lambda \bar{x} = \dfrac{\lambda}{\mu C}$. Thus,

$$T_0(\rho) = \frac{1}{\mu C - \lambda} \tag{3.2}$$

For the distributed system shown in Figure 2.1, we have [KLEI 76, Eq.(5.17)]

$$T(\rho) = \sum_{k=1}^{m} \frac{\lambda_k}{\lambda} T_k \tag{3.3}$$

where $T_k =$ E [response time to pass down the k^{th} parallel chain]. Since the k^{th} chain contains n_k processors in series, we have

$$T_k = n_k T^{(k)} \tag{3.4}$$

where $T^{(k)}$ is the mean response time for *each* processor in the kth branch. Each of these processors behaves as an M/M/1 queue with input rate λ_k, mean number of operations $1/\mu n_k$ and capacity C_k operations/second. Thus, $T^{(k)}$ has the same form as given in Eq.(3.1) with $\bar{x}=1/\mu n_k C_k$ and $\rho=\lambda_k \bar{x}$. Thus,

$$T^{(k)} = \frac{1/\mu n_k C_k}{1 - \lambda_k/\mu n_k C_k} = \frac{1}{\mu n_k C_k - \lambda_k}$$

From this last equation and Eqs.(3.3) and (3.4), we get

$$T(\rho) = \sum_{k=1}^{m} \frac{\lambda_k}{\lambda} \cdot \frac{n_k}{\mu n_k C_k - \lambda_k} \tag{3.5}$$

This is the governing behavior of our general series-parallel topology.

Finally, we have the general form of our performance measure:

$$\frac{T(\rho)}{T_0(\rho)} = \sum_{k=1}^{m} \frac{\lambda_k n_k}{\lambda} \cdot \frac{\mu C - \lambda}{\mu n_k C_k - \lambda_k} \tag{3.6}$$

This last may also be rewritten as

$$\frac{T(\rho)}{T_0(\rho)} = \sum_{k=1}^{m} n_k \frac{\rho_k/(1-\rho_k)}{\rho/(1-\rho)} \tag{3.7}$$

where $\rho_k = \lambda_k/\mu n_k C_k$ and $0 \leq \rho_k < 1$.

Let us now evaluate $T(\rho)/T_0(\rho)$ for some interesting, special cases:

THE PURE TANDEM

This system is depicted in Figure 3.1 and has the following parameter values: $m=1$, $n_1=n$, $\lambda_1=\lambda$, $1/\mu_1=1/n\mu$ and $C_1=C/n$. Substituting into Eq.(3.6), we have

$$\frac{T(\rho)}{T_0(\rho)} = \frac{\lambda n}{\lambda}\cdot\frac{\mu C-\lambda}{\mu n\frac{C}{n}-\lambda} = n \tag{3.8}$$

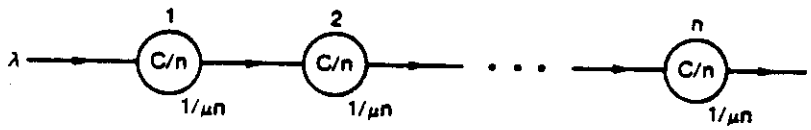

Figure 3.1: The Pure Tandem System

THE PURE PARALLEL SYSTEM

This system is depicted in Figure 3.2 and has the following parameter values: $n_k=1$, $\lambda_k=\lambda/m$, $1/\mu_k=1/\mu$, $C_k=C/m$ for $k=1,2,\ldots,m$. Substituting into Eq.(3.6), we have

$$\frac{T(\rho)}{T_0(\rho)} = \sum_{k=1}^{m}\frac{\lambda/m}{\lambda}\,\frac{\mu C-\lambda}{\mu\frac{C}{m}-\frac{\lambda}{m}} = m \tag{3.9}$$

THE SYMMETRIC SERIES-PARALLEL SYSTEM

This system is depicted in Figure 3.3 and has the following parameter values: $n_k=n$, $\lambda_k=\lambda/m$, $1/\mu_k=1/n\mu$, $C_k=C/mn$ for $k=1,2,\ldots,m$. Substituting into Eq.(3.6), we have

$$\frac{T(\rho)}{T_0(\rho)} = \sum_{k=1}^{m}\frac{\lambda n/m}{\lambda}\cdot\frac{\mu C-\lambda}{\frac{\mu nC}{mn}-\frac{\lambda}{m}} = nm \tag{3.10}$$

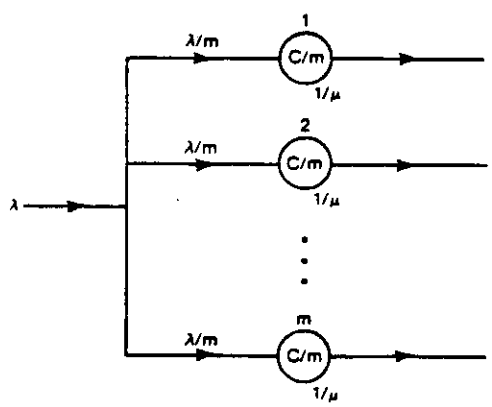

Figure 3.2:
The Pure Parallel System

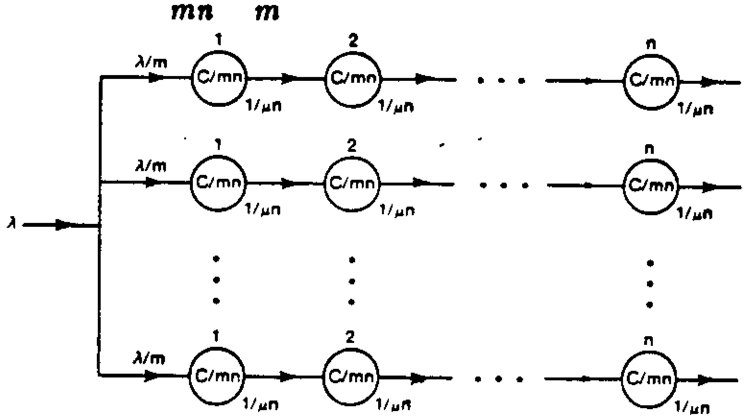

Figure 3.3:
The Symmetric Series-Parallel System

THE SERIES-PARALLEL SYSTEM WITH UNIFORM TRAFFIC

This system is depicted in Figure 2.1 and has the following parameter values: $\lambda_k=\lambda/m$, $1/\mu_k=1/n_k\,\mu$, $C_k=C/mn_k$ for $k=1,2,\ldots,m$. Substituting into Eq.(3.6), we have

$$\frac{T(\rho)}{T_0(\rho)} = \sum_{k=1}^{m} \frac{\lambda n_k/m}{\lambda} \cdot \frac{\mu C-\lambda}{\mu \frac{C}{m} - \frac{\lambda}{m}} = \sum_{k=1}^{m} n_k \tag{3.11}$$

Thus, we see, for *all* of the simple configurations, that things seem to be getting *worse* as we add more distributed processors! Indeed, we show below that, no matter how one selects λ_k and C_k, subject to constraint Eqs.(2.1, 2.2 and 2.3), one must have

$$\frac{T(\rho)}{T_0(\rho)} \geq \min_k n_k \geq 1 \tag{3.12}$$

4. Optimizing the Distributed System

In the previous section, we evaluated the behavior of the general series-parallel system. In this section, we consider some *optimization* problems, where we allow ourselves to adjust λ_k and/or C_k (subject to constraints 2.1, 2.2 and 2.3) in order to minimize $T(\rho)/T_0(\rho)$.

Before we proceed with these optimization problems, let us establish certain convexity properties of our objective function Eq.(3.6), which we repeat here.

$$\frac{T(\rho)}{T_0(\rho)} = \sum_{k=1}^{m} \frac{\lambda_k n_k}{\lambda} \cdot \frac{\mu C-\lambda}{\mu n_k C_k-\lambda_k} = \frac{\mu C-\lambda}{\lambda} \sum_{k=1}^{m} U_k \tag{4.1}$$

where

$$U_k = \frac{\lambda_k n_k}{\mu n_k C_k-\lambda_k} \tag{4.2}$$

We shall now prove that U_k is a convex function of its various arguments; then, since our performance ratio as given in Eq.(4.1) is a sum of the terms U_k, we shall have established that the performance ratio itself is a convex function of the system parameters (a sum of convex functions is convex). First we prove that U_k is a convex function of the traffic parameter λ_k by twice differentiating, namely,

$$\frac{d^2 U_k}{d\lambda_k^2} = \frac{2\mu n_k^2 C_k}{(\mu n_k C_k-\lambda_k)^2} \tag{4.3}$$

This second derivative is clearly greater than zero, thereby establishing the convexity of U_k with respect to λ_k. Thus, our objective function is a convex function of the traffic parameters λ_k. We now repeat the procedure with respect to the capacity, C_k. Twice differentiating, we obtain

$$\frac{d^2 U_k}{dC_k^2} = \frac{2\mu^2 \lambda_k n_k^3}{(\mu n_k C_k-\lambda_k)^3} \tag{4.4}$$

Since $\rho_k = \lambda_k/\mu n_k C_k < 1$ (i.e., we are assuming a stable system), then we have that the second derivative given in Eq.(4.4) is greater than zero, and this establishes the convexity of our performance ratio with respect to the capacity parameters C_k.

The set of optimization problems we wish to address may be identified by referring to Figure 4.1. In that figure, box A refers to a system with an arbitrary set of parameters for which no optimization has been performed; thus, the performance of box A is given by Eq.(3.6).

The first problem we address is to hold C_k and n_k fixed and optimally select the traffic pattern, λ_k, in order to minimize our performance ratio $T(\rho)/T_0(\rho)$. This leads us to box B. The second problem we solve is to hold λ_k and n_k fixed and optimally assign the total channel capacity C among the set of distributed processors, once again minimizing the performance ratio $T(\rho)/T_0(\rho)$, which leads us to box C. To reach box D, which is the joint optimization of both λ_k and C_k, we may either move from box B by optimizing the capacity or move from box C by optimizing the traffic. As we show below, both solutions are identical.

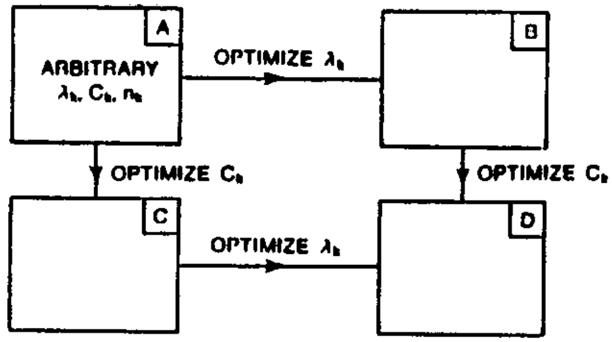

Figure 4.1

We begin with the problem characterized in box B, namely,

$$\underset{\lambda_k}{Minimize} \quad \frac{T(\rho)}{T_0(\rho)}$$

subject to Eq.(2.2) for a given set C_k, n_k and μ. Clearly, this is equivalent to minimizing $T(\rho)$ since $T_0(\rho)$ is a constant for this problem. Thus, we set up the Lagrangian

$$G = T(\rho) + \beta \left(\sum_{k=1}^{m} \lambda_k - \lambda \right) \tag{4.5}$$

where β is the Lagrangian undetermined multiplier. Forming the partial derivative of G with respect to λ_k, setting it equal to zero for each k and solving for λ_k, we obtain

$$\lambda_k = \mu n_k C_k - \sqrt{-\mu / \lambda \beta}\, n_k \sqrt{C_k} \tag{4.6}$$

In order to determine the unknown constant, namely, $\sqrt{-\mu / \lambda \beta}$, we simply form the constraint condition, Eq.(2.2), by summing Eq.(4.6), equating it to λ and then solving for the unknown constant. The result of this evaluation yields, finally,

$$\lambda_k = \mu n_k C_k - \mu C (1-\rho) \frac{n_k \sqrt{C_k}}{\sum_{i=1}^{m} n_i \sqrt{C_i}} \tag{4.7}$$

Rewriting this in terms of ρ_k, we have (for $C_k > 0$)

$$\rho_k = 1 - \frac{C(1-\rho)}{\sqrt{C_k}} \cdot \frac{1}{\sum_{i=1}^{m} n_i \sqrt{C_i}} \tag{4.8}$$

Either Eq.(4.7) or (4.8) gives the solution for the optimal assignment of traffic to the generalized series-parallel distributed processing network. However, we observe that the constraints on ρ are $0 \leq \rho_k < 1$. Clearly, the upper constraint is satisfied as long as $\rho < 1$, a condition we have assumed. It is the lower constraint which can be violated if, indeed, the capacity, C_k, is too small. That is, we must take care that our optimization problem does not try to place a *negative* flow λ_k on the kth parallel chain! Since we know that the objective function is a convex function of the flows, then when λ_k tries to go negative, the optimization procedure will force us to keep it at its boundary value of zero. Thus, the solution for λ_k given in Eq.(4.7) will hold only if the following condition is satisfied (C^*, ρ^* and k^* defined below):

$$\sqrt{C_{k^*}} \geq \frac{C^*(1-\rho^*)}{\sum_{i=1}^{k^*} n_i\sqrt{C_i}} \tag{4.9}$$

Any chain k for which condition (4.9) is not satisfied will lead to an optimum flow of value $\lambda_k = 0$. Since we have assumed that the capacities, C_k, are monotonically decreasing with the index k, then we may define k^* as the maximum value of k for which condition (4.9) holds. For $k = k^*+1,..., m$, the solution in Eq.(4.6) does not apply, but due to the convexity of our function with respect to λ_k, we recognize that λ_k must lie at the boundary of our constraint space, thereby yielding the solution $\lambda_k = 0$ for $k = k^*+1,..., m$. Returning once again to Eq.(4.6), which is now valid only for $k = 1,2,...,k^*$, we must re-evaluate the undetermined constant $\sqrt{-\mu / \lambda \beta}$; we do this again by forming the constraint Eq.(2.2), but we now recognize that we must sum not over all k but only over the range $k = 1,2,...,k^*$. The result of this operation, along with our earlier observation, yields the final result for our traffic assignment, namely,

$$\lambda_k = \begin{cases} \mu n_k C_k - \mu C^*(1-\rho^*)\dfrac{n_k\sqrt{C_k}}{\sum_{i=1}^{k^*} n_i\sqrt{C_i}} & k = 1,2,...,k^* \\ 0 & k = k^*+1, \ldots, m \end{cases} \tag{4.10}$$

where we have defined the following quantities

$$C^* = \sum_{k=1}^{k^*} n_k C_k \tag{4.11}$$

$$\rho^* = \lambda/\mu C^* \tag{4.12}$$

In order to determine the optimum performance ratio, we now substitute our optimized traffic assignment as given in Eq.(4.10) into the performance equation, Eq.(3.6), to yield

$$\left.\frac{T(\rho)}{T_0(\rho)}\right|_{\text{opt } \lambda_k} = \frac{\mu\rho^*(1-\rho)}{\lambda\rho(1-\rho^*)}\left[\sum_{k=1}^{k^*} n_k\sqrt{C_k}\right]^2 - \frac{N^*(1-\rho)}{\rho} \tag{4.13}$$

where we have defined N^* to be the total number of processors in our distributed system which carry a non-zero traffic, namely,

$$N^* = \sum_{k=1}^{k^*} n_k \tag{4.14}$$

Eq.(4.13) is the solution to the optimization problem represented by box B.

The optimization problem we just solved is, clearly, the solution to the Flow Assignment (FA) problem described in [KLEI 76], except that here we have a special topology to work with; so, we are able to develop an explicit analytic solution rather than the usual algorithmic solution (e.g., the Flow Deviation (FD) method in [KLEI 76, Sect.5.8]). We observe that $l^{(k)}$, the "length" (defined for the FD method) of the kth parallel chain is

$$l^{(k)} = \begin{cases} \dfrac{1}{\lambda}\left[\dfrac{\sum_{i=1}^{k^*} n_i\sqrt{C_i}}{C^*(1-\rho^*)}\right]^2 & k = 1,2,...,k^* \\ \dfrac{1}{\lambda\mu C_k} & k = k^*+1, \ldots, m \end{cases} \tag{4.15}$$

It can be seen from Eq.(4.9) that $l^{(k)}$ for $k = k^*+1,\ldots, m$ is strictly greater than $l^{(k)}$ for $k = 1,2,\ldots,k^*$. Thus, each of the zero-traffic parallel chains ($k>k^*$) is *longer* than each of the positive-traffic chains ($k\leq k^*$), which (by the FD method) implies that no traffic should be deviated onto the zero-traffic chains. Further, since each of the positive-traffic chains has the same "length," then no traffic should be deviated among those chains. Thus, our solution is in complete agreement with the FD method.

Now, let us proceed from box A to box C, namely, let us assume that the traffic λ_k remains fixed and that our objective is to optimally assign the total capacity, C, among the set of processors, that is, we must $\underset{C_k}{Minimize}\ \ T(\rho)/T_0(\rho)$ subject to constraint Eq.(2.1) and, of course Eq.(2.3). The appropriate Lagrangian in this case is

$$G = T(\rho) + \beta\,(\sum_{k=1}^{m} n_k C_k - C) \tag{4.16}$$

Again, due to the convexity of our performance ratio with respect to C_k, we may form $\partial G \,/\, \partial C_k = 0$ for all k in order to determine the optimal value for C_k. Carrying out this differential, evaluating the undetermined multiplier β by forming the constraint as given in Eq.(2.1) and solving for C_k, we obtain the optimal capacity assignment as

$$C_k = \frac{\lambda_k}{\mu n_k} + \frac{C(1-\rho)\sqrt{\lambda_k/n_k}}{\sum_{i=1}^{m}\sqrt{\lambda_i n_i}} \,. \tag{4.17}$$

As long as $\rho < 1$, then the optimal capacity assignment given in Eq.(4.17) always holds, as opposed to the traffic assignment problem where we encountered negative flows. If we now plug this optimized capacity assignment into our performance ratio as given in Eq.(3.6), we then obtain the minimized response time ratio, namely,

$$\left.\frac{T(\rho)}{T_0(\rho)}\right|_{\text{opt } C_k} = \left(\sum_{k=1}^{m} \sqrt{\lambda_k n_k/\lambda}\right)^2 \tag{4.18}$$

Note the similarity of this capacity optimization procedure and solution to that given in Eq.(5.28) of [KLEI 76]. The optimum performance profile for box C is that given in Eq.(4.18).

Let us now further optimize the solution in Eq.(4.18) with respect to the traffic flows λ_k, namely,

$$\underset{\lambda_k}{Minimize} \qquad \left.\frac{T(\rho)}{T_0(\rho)}\right|_{\text{opt } C_k} \tag{4.19}$$

The solution procedure here is almost identical to that given in Exercise 5.5 of [KLEI 76] and leads to the following traffic assignment:

$$\lambda_k = \begin{cases} \lambda & k=k_0 \\ 0 & k\neq k_0 \end{cases} \tag{4.20}$$

where k_0 is such that $n_{k_0} \leq n_k$ for all k; in the case where more than one value of k qualify for the value k_0, then selecting any one of them will yield the same optimum solution. We are assuming, in this double optimization problem, that each time the λ_k assignment changes, then the optimized capacity assignment follows that change (i.e., through Eq.(4.17); thus, the solution to problem (4.19) will yield a traffic pattern where all the traffic travels down the minimum length parallel chain, and that chain will contain the total capacity, C, uniformly spread out over n_{k_0} processors. In this case, we have now arrived at a pure tandem, whose solution is known from Eq.(3.8); so, the performance ratio which applies to box D, arrived at through this capacity and then traffic assignment optimization (i.e., box A to box C to box D), gives

$$\left.\frac{T(\rho)}{T_0(\rho)}\right|_{\text{opt } C_k,\lambda_k} = n_{k_*} \tag{4.21}$$

On the other hand, we could have begun with the solution from box B and then done an optimum capacity assignment to arrive at box D, a procedure which we now describe. Thus, we carry out the following minimization

$$\underset{C_k}{\mathit{Minimize}} \qquad \left.\frac{T(\rho)}{T_0(\rho)}\right|_{\text{opt } \lambda_k} \tag{4.22}$$

Of course, this result must lead to a performance ratio as given in Eq.(4.21). The approach here is not quite as simple as going from box C to box D. In particular, the problem comes about from the way in which we progressed from Eq.(4.7) to Eq.(4.10), namely, we required $\lambda_k = 0$ for $k = k^*+1, \ldots, m$ in order to prevent those values of λ_k from going negative. However, now that we allow ourselves to optimize the capacity to match the traffic flow (i.e., we will move from box B to box D), we may take the liberty of ignoring this non-negativity constraint, optimizing the capacity and then checking, after this capacity optimization, to see that the non-negativity constraint is satisfied by the λ_k's. Thus, for a given set, C_k, and chain lengths, n_k, we accept that the optimum traffic assignment is given by Eq.(4.7) for all k. Given this assignment for λ_k, we will now move from box B to box D by optimizing the capacity relative to the λ_k found from Eq.(4.7). But, of course, we have already carried out the capacity optimization in moving from box A to box C, and the solution is given in Eq.(4.17). Let us evaluate this optimized capacity assignment by substituting λ_k from Eq.(4.7) into our expression for C_k in Eq.(4.17), namely,

$$C_k = C_k - \frac{C(1-\rho)\sqrt{C_k}}{\sum_{i=1}^{m} n_i\sqrt{C_i}} + \frac{C(1-\rho)y_k}{\sum_{i=1}^{m} y_i n_i} \tag{4.23}$$

where

$$y_k = \left(\frac{\lambda_k}{n_k}\right)^{1/2} = \left(\mu C_k - \frac{\mu C(1-\rho)\sqrt{C_k}}{\sum_{i=1}^{m} n_i\sqrt{C_i}}\right)^{1/2} \tag{4.24}$$

Cancelling C_k on both sides of Eq.(4.23) and solving, we obtain

$$\sqrt{C_k} = \left(\sum_{i=1}^{m} n_i\sqrt{C_i}\right)\frac{y_k}{\sum_{i=1}^{m} y_i n_i} \tag{4.25}$$

Observing that y_k is a function of C_k, what are the solutions to this set of equations ($k = 1,2,\ldots,m$)? It can be shown that there are exactly m solution sets to this equation, each of the form

$$C_k = \begin{cases} C/n_{k'} & k=k' \\ 0 & k\neq k' \end{cases} \tag{4.26}$$

where the m solutions correspond to the m possible values for k', namely, $k' = 1,2,\ldots,m$. Plugging any one of these solutions for C_k into Eq.(4.24), we find

$$y_k = \begin{cases} \sqrt{\lambda/n_k} & k=k' \\ 0 & k\neq k' \end{cases}$$

It is now easy to verify that Eq.(4.26) satisfies Eq.(4.25). Each of these m solutions corresponds to a pure tandem, for which the performance ratio is given in Eq.(3.8), namely,

$$\frac{T(\rho)}{T_0(\rho)} = n_{k'} \tag{4.27}$$

Among the m possible solutions, that which minimizes this performance ratio is clearly the case $k' = k_0$ where, we recall, k_0 is defined as that chain with the minimum number of series processors, namely, $n_{k_0} \leq n_k$ for all k. This then yields the optimum performance profile for box D (by travelling from box A to box B to box D), namely,

$$\left.\frac{T(\rho)}{T_0(\rho)}\right|_{\text{opt } \lambda_k, C_k} = n_{k_0} \tag{4.28}$$

Thus, we see that the double optimization, in both cases, has led to a solution where all the traffic and all the capacity are placed on that chain with the fewest number of nodes in series. We now observe that the solution in box B allows us to put tight upper and lower bounds on our performance profile for any capacity-optimized series-parallel system. The general form for $T(\rho)/T_0(\rho)$ is given in Eq.(4.18). If we let $a_k = \sqrt{\lambda_k/\lambda}$ and $b_k = \sqrt{n_k}$, then we may apply the Cauchy-Schwartz inequality (see Eq.(4.59) in [KLEI 75]), namely,

$$\left(\sum_{k=1}^{m} a_k\, b_k\right)^2 \leq \sum_{k=1}^{m} a_k^2 \sum_{k=1}^{m} b_k^2 \tag{4.29}$$

to obtain

$$\left.\frac{T(\rho)}{T_0(\rho)}\right|_{\text{opt } C_k} \leq \sum_{k=1}^{m} n_k \tag{4.30}$$

Furthermore, since $n_k \geq n_{k_0}$, then $(\sum\sqrt{\lambda_k\, n_k/\lambda})^2 \geq n_{k_0}(\sum\sqrt{\lambda_k/\lambda})^2$. Now, since $\lambda_k/\lambda \leq 1$, it follows that $\sqrt{\lambda_k/\lambda} \geq \lambda_k/\lambda$, and so, $(\sum\sqrt{\lambda_k/\lambda})^2 \geq (\sum\lambda_k/\lambda)^2 = 1$. Thus we have shown that Eq.(4.18) is lower bounded by n_{k_0}, that is,

$$n_{k_0} \leq \left.\frac{T(\rho)}{T_0(\rho)}\right|_{\text{opt } C_k} \leq \sum_{k=1}^{m} n_k \tag{4.31}$$

From Eq.(4.28), we see that this lower bound is, indeed, achieved in box D. (Note also that the upper bound was achieved in the series-parallel system with uniform traffic (Eq.(3.11)).

If one had the further freedom to select the length of this minimum-length chain, then it is clear that one would select $n_{k_0} = 1$ to obtain the absolutely optimum performance, which corresponds to our centralized system! Thus, we see that, so far as *this* performance profile is concerned, distributed processing can only hurt us.

5. Cost Considerations

As promised, we now return to certain issues involving cost. In particular, we were disappointed to find that distributed processing offers no advantage as far as our performance ratio is concerned, and we argued earlier that this negative result is due largely to the assumption that the single, centralized processor of capacity C costs the same as a distributed system whose total capacity, C_{dist}, is equal to C. However, as we pointed out earlier, there is, basically, a dis-economy in scale for the cost of processor capacity due to the integrated chip revolution, and it is this fact that has the potential for "saving" distributed processing so far as our model is concerned. In order to introduce this factor into our analysis, we will find how much more capacity we must place in the distributed system relative to the centralized system in order that $T(\rho)/T_0(\rho) = 1$; i.e., we shall adjust C_{dist} so that the distributed system and the centralized system give the same mean response times.

Let us consider the symmetric series-parallel system, introducing the two different capacities. Re-deriving Eq.(3.11), we find

$$\frac{T(\rho)}{T_0(\rho)} = mn\frac{\mu C-\lambda}{\mu C_{dist}-\lambda} \tag{4.32}$$

Forcing this ratio to be unity, we find that the ratio of capacities in the two systems must be

$$\frac{C_{dist}}{C} = mn-\rho(mn-1) \tag{4.33}$$

where, once again, $\rho = \lambda/\mu C$. We note that this ratio of capacities is dependent upon the centralized system's load, ρ. In particular, we notice that $\lim_{\rho \to 0} C_{dist}/C = mn$ and also $\lim_{\rho \to 1} C_{dist}/C = 1$; i.e., at light load we require an increase in capacity equal to the number of processors in the distributed system, whereas at heavy load we need only an infinitesimal extra capacity (due to the steep slope of the mean response time near saturation).

6. Conclusions

We focussed on the ratio of response time of a distributed system to that of a centralized system as our performance measure. We found that distributed processing can never improve that ratio unless we introduce a dis-economy of scale in the cost of processing. We solved a number of related optimization problems, finding the optimum traffic assignment and the optimum capacity assignment, and when both those assignments are optimized jointly, we were led back to the centralized system. A number of extensions to this model are currently being investigated.

REFERENCES

[KLEI 64] Kleinrock, L., *Communication Nets; Stochastic Message Flow and Delay*, McGraw-Hill Book Company, New York, 1964. (Out of Print) Reprinted by Dover Publications, 1972.

[KLEI 75] Kleinrock, L., *Queueing Systems, Volume I: Theory*, Wiley Interscience, New York, 1975.

[KLEI 76] Kleinrock, L., *Queueing Systems, Volume II: Computer Applications*, Wiley Interscience, New York, 1976.

Presented at the
Twenty-Second Annual Allerton Conference
on
Communication, Control, and Computing
October 3, 1984