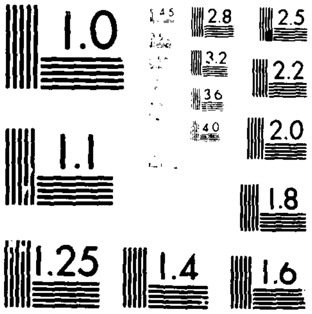
AD-A083 734 MARYLAND UNIV COLLEGE PARK COMPUTER SCIENCE CENTER F/G 12/1
THE EFFECT OF THE FUTURE IN WORK DISTRIBUTION.(U)
FEB 80 G RICART, A K AGRAWALA AFOSR-78-3654
UNCLASSIFIED CSC-TR-868 AFOSR-TR-80-0310 NL

1 OF 1
AD
A083734

END
DATE
FILMED
6-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

1807-1856-1920

UNIVERSITY OF MARYLAND
COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND
20742

Technical Report TR-868   February 1980
AFOSR78-3654[illegible]

THE EFFECT OF THE FUTURE IN WORK DISTRIBUTION

Glenn Ricart
Ashok K. Agrawala

This research was supported in part by the Air Force Office of Scientific Research under grant AFOSR 78-3654[illegible]. Assistance was also provided through the National Institutes of Health.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (19) (18) AFOSR-TR- 80-0310
2. GOVT ACCESSION NO. AD-A083734
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) THE EFFECT OF THE FUTURE IN WORK DISTRIBUTION
5. TYPE OF REPORT & PERIOD COVERED Interim
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Glenn Ricart, Ashok K. Agrawala
8. CONTRACT OR GRANT NUMBER(s) (15) AFOSR-78-3654
9. PERFORMING ORGANIZATION NAME AND ADDRESS
University of Maryland
Computer Science Center
College Park, MD 20742
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F (16) 2304 (17) A2
11. CONTROLLING OFFICE NAME AND ADDRESS
Air Force Office of Scientific Research/NM
Bolling AFB, Washington, DC 20332
12. REPORT DATE (11) Feb[redacted]80
13. NUMBER OF PAGES 25
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 29 402 11
15. SECURITY CLASS. (of this report) UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.
(14) CSC-TR-869
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
(9) Technical repts.
18. SUPPLEMENTARY NOTES
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A controller is considered which routes arrivals among several servers of different speeds. A decision which sends work to the server which will complete it soonest does not optimize the average completion time (mean flow time) because it doesn't take into account the impact of the decision on future arrivals. This impact on future arrivals, the "future effect", can be significant at high arrival rates. An estimate of the size of the future effect is derived and controllers which take it into account in routing decisions can reduce the average completion time to near optimum. The effect is most pronounced when the →

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 403018

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract cont.

service requirements for arrivals are nearly constant, server speeds are markedly different, and the arrival rate is close to the system's capacity. A controller considering the future effect will more heavily weigh a potential server's backlog than the arrival's service time when making a routing decision.

Abstract

A controller is considered which routes arrivals among several servers of different speeds. A decision which sends work to the server which will complete it soonest does not optimize the average completion time (mean flow time) because it doesn't take into account the impact of the decision on future arrivals. This impact on future arrivals, the "future effect", can be significant at high arrival rates. An estimate of the size of the future effect is derived and controllers which take it into account in routing decisions can reduce the average completion time to near optimum. The effect is most pronounced when the service requirements for arrivals are nearly constant, server speeds are markedly different, and the arrival rate is close to the system's capacity. A controller considering the future effect will more heavily weight a potential server's backlog than the arrival's service time when making a routing decision.

TABLE OF CONTENTS

1. Introduction

The functional distribution of components in computing networks brings with it the problem of distributing work among interchangeable servers. In this paper, a centralized controller (C) is imagined which distributes work among multiple servers N_i.

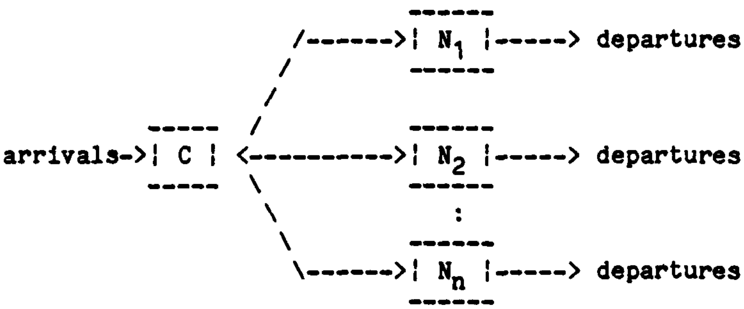


Figure 1.

Each arrival must be immediately dispatched by C to one of the servers; there it may be queued if the server is not free. The servers are work-conserving and operate at fixed speeds.

The goal of C is to distribute work in such a way that the average time to complete service (mean flow time) is minimized. This is equivalent to minimizing the total time in system for the arriving tasks.

The nature of the problem varies with the amount of knowledge that C is presumed to have available for making decisions. For example, if there is no feedback from the servers, the controller C must make its decisions based only upon its knowledge of arrival instants and its own memory of past routing[1].

In this research we permit C to be cognizant of all knowledge in the system except for future arrival instants. In particular, C knows the number of servers, their speeds, their current backlogs, its past routing decisions, and the mean arrival rate. These assumptions correspond to a realistic system with a very good information gathering system.

For the purposes of study, arrivals are generated by a Poisson process. The service times for the arrivals may be constant or variable. Three servers are utilized; to show the effect of varying server speeds, they are given speed ratios of 4:2:1. The mean service time of a task is normalized to 100.0 on the speed 1 server. (The speed 4 server processes the average task in 25.0.)

While this problem is similar to the Multiple Producer / Multiple Consumer Problem [8], it is simplified by assuming that there are no communications delays between the producers and consumers to impede information flow.

The goal of minimizing the average completion time (mean flow time) in a system with servers of different speeds has been previously considered in the case that all work is available at time 0 [2] [7]. An optimal algorithm to minimize mean flow time [4] when the servers are of fixed speed ratio operates by assigning tasks to the server which will complete them soonest.

3. Distributing Work with Varying Server Speeds

When the arrival rate to a system like that shown in Figure 1 is so low that the controller C often can choose between completely idle servers, it will usually choose the fastest server since the work will be completed there the soonest. Only after queueing delays accumulate at the fastest server will the controller C wish to send work to a slower but idle server. As a result, the servers are not used equally. The fastest server performs the bulk of the work at very low arrival rates. If arrivals increase to approach the capacity of the system, the controller C is forced to distribute work in proportions equal to the speeds of the servers in order to find enough capacity to process the arrivals.

Given the arrival rate and server capacities, Buzen and Chen [3] have computed the probabilistic fractions of the arrival rate that should be sent to each of the servers for Poisson arrivals and service times to minimize average completion time.

3. Task Completion Time

An obvious approach to the problem posed in section 1 is to have the controller C route a new arrival to the server where it will be completed the soonest. This strategy is optimal in the case that all work arrives at the controller at time 0 [4].

In making this decision, the controller C must test each server by adding the server's current backlog to the service time required for the new arrival at that server. The smallest total shows the server at which the new arrival will be completed the soonest.

Because the completion time of each arrival is being minimized, it is tempting to think that the average completion time for the system is minimized. This is not true!

This discrepancy between local optimization and global optimization was investigated by studying a complete transcript of arrival instants, routing decisions, and resultant completion times in a simulation. The effect of a routing decision was found to extend beyond the completion time of the work routed; it impacted subsequent or future arrivals. With high arrival rates the primary effect of the routing decision was on the completion time of subsequent arrivals.

4. The Future Effect

Imagine two servers. The faster completes work in 10 units of time. The slower one takes 20. Picture the situation using the time line or modified Gantt chart presented below. Arrivals are shown on the top line. The time spent in service is shown on the middle line. The completion times are summarized on the third line.

```
Arrivals:    1  2   3                    4      5
Server 1: ___11111111112222222222333333333344444444445555555555___
Tot Time:            10        17        23        12        15

Arrivals:
Server 2: ________________________________________________________
Tot Time:
Avg Time: -
```

Average completion time: 15.4

Consider the effect on average completion time of placing arrival number 2 with the slower server. The situation would have been:

```
Arrivals:    1      3                    4      5
Server 1: ___11111111113333333333________44444444445555555555___
Tot Time:            10        13                 10        13
Avg Time: 11.5

Arrivals:       2
Server 2: ______22222222222222222222____________________________
Tot Time:                         20
Avg Time: 20.0
```

Average completion time: 13.2

The average completion time has been reduced significantly. In fact, even if the slower server were only one-third as fast, the overall average would have been improved.

A controller minimizing the completion time of each arrival would never have moved arrival number 2, however. Its completion time at the fast server is only 17...less than the 20 it experiences at the slower server.

The controller may, however, attempt to move arrival number 3 since its completion time at the fast server is 23, higher than an alternative server with completion time 20.

```
Arrivals:    1  2                        4      5
Server 1: ___11111111112222222222________44444444445555555555___
Tot Time:            10        17                10        13

Arrivals:           3
Server 2: __________33333333333333333333________________________
Tot Time:                             20
```

Average completion time: 14.0

Even though the pattern of busy time at the fastest server is identical to the previous case, moving arrival 3 does not reduce the average completion time as much as moving arrival 2.

It is also interesting to note that the average completion time is improved by allowing the fastest server to go idle. A policy which attempts to keep the fastest server busy does not result in lowest average completion time. This is usually true only in deadline scheduling and flowshop scheduling [5].

To optimize the assignment of the arrivals given in these examples, the total cost of an assignment on current and future completion times must be considered.

For example, the cost of assigning arrival 2 to server 1 is:

waiting time of 7
+ service time of 10
+ delay of 10 to arrival 3
+ delay of 2 to arrival 4
+ delay of 2 to arrival 5
= 31

Of this total of 31 time units, only 17 were seen by controller C as direct costs when arrival number 2 was routed. The other 14 were the effect of the decision on future arrivals. The total cost of assigning it to server 2 is only 20. This arrival pattern is optimized by moving arrival 2 to the slower server due to the "future effect".

A controller attempting to optimize its routing of arrivals must consider both the apparent cost of routing and the "future effect" cost in choosing a server. The apparent cost of routing is visible and easily determined. The "future effect" cost is more elusive.

5. Impact of the Future Effect

The magnitude of the future effect can be studied if a single server is considered in an M/G/1 environment. The expected cost of an extra arrival to an existing arrival pattern (its contribution to overall system delay) can be found analytically. It is the sum of the following:

1. The service time of the extra arrival. The service time is dependent on the speed of the server.

2. If the server is busy when the arrival occurs, the order of service can be permuted to consider the new arrival to preempt the server and go into service immediately [9]. The work in the remainder of the current busy period is delayed by the service time of the new arrival. The future effect cost is the service time of the new arrival multiplied by the number of tasks yet to complete in the original busy period which was interrupted.

3. If the server is busy when the arrival occurs, the existing busy period is extended by the length of the new service time and may bump into a subsequent busy period. All members of the subsequent busy period are delayed, but not by the entire amount of the new arrival's service time. The subsequent busy period, if delayed, may also move back and bump into another following busy period, and so on. The result is one much larger busy period composed of the smaller, individual busy periods which have been coalesced.

4. If the server is not busy when the arrival occurs, a succeeding arrival which begins a new busy period may be delayed if it arrives before service is completed to the extra arrival. Therefore, the recursive coalescing of busy periods discussed in the previous paragraph may occur.

The costs may be quantified as follows:

Let L = arrival rate to the server
w = the backlog of work at the server
b = the mean service time
r = rho, the utilization of the server
x = the service time of the new arrival

Then the cost of item 1 above is simply

$$x \tag{1}$$

The cost of item 2 above is the average remaining number of tasks in the existing busy period. Following Kleinrock's sub-busy period analysis [6] this is of size

$$\frac{w}{b*(1-r)}$$

and multiplying by the cost (x) yields a total cost for this step of

$$\frac{x*w}{(1-r)*b} \tag{2}$$

It is not necessary to condition term (2) based on the probability of a busy server because it contributes nothing when the server is idle (since $w=0$ in that case).

Item 3 above occurs when the server is busy upon arrival, but item 4 is of the same size and occurs when the server is not busy upon arrival. The additional delay to subsequent busy periods may be calculated recursively. The expected cost to the first subsequent busy period is

$$F(x,L) = \text{Integral from 0 to } x \text{ of } (x-z)e^{-Lz}L \, dz$$

Here z can be imagined to be the size of the existing gap between busy periods and x is the amount of time by which the first busy period is extended. Factor (x-z) is the delay to the following busy period, e^{Lz} is the probability of no arrivals for period of time z, and L is the probability of arrival during interval dz.

The result is the average amount by which a subsequent busy period is moved back (if any). The second subsequent busy period will be coalesced only if F(x,L) is larger than the idle time period preceding it. The average amount that the second subsequent busy period is delayed is F(F(x,L),L). The third subsequent busy period is delayed on the average F(F(F(x,L),L),L) and so on. Each busy period in the future has an average of 1/(1-r) customers (M/G/1). So the total cost due to items 3 and 4 is

$$\frac{F(x,L)}{(1-r)} + \frac{F(F(x,L),L)}{(1-r)} + \frac{F(F(F(x,L),L),L)}{(1-r)} + \ldots \qquad (3)$$

The total cost of an extra task is the sum of terms (1-3):

$$x + \frac{x^{*}w}{b^{*}(1-r)} + \frac{F(x,L)}{(1-r)} + \frac{F(F(x,L),L)}{(1-r)} + \frac{F(F(F(x,L),L),L)}{(1-r)} + \ldots \qquad (4)$$

Only term (2) depends on the size of the existing backlog at a server; it causes the total cost to rise linearly with existing backlog. Terms (1) and (3) define a fixed cost; term (1) is the known cost and term (3) is the future fixed cost.

6. Controllers Considering the Future Effect

A controller which takes the future effect into account in routing arrivals uses (4) to evaluate the additional cost of assigning new work to a particular server. Work is routed to the server where it will have the least additional total cost. Note that the directly observable costs (w and x) are taken into consideration but do not appear as terms by themselves.

While (4) is exact for an M/G/1 situation, the arrival pattern to a particular server for the situation pictured in Figure 1 will not have independent arrivals. After an arrival has been routed, a closely following arrival is not as likely to be routed to the same server. Nevertheless, the terms developed for M/G/1 can be used to approximate the total system impact. The results of the next section show that this approximation yields excellent results.

7. Evaluation

7.1. Method

A simulation experiment was carried out using SIMULA on a DECsystem-10 for 24,000 arrivals. Assignment to servers with speed ratios 4:2:1 was carried out by a controller which was programmable to test the different algorithms. The principal result of each simulation run was a system completion time averaged over all arrivals.

The controller algorithms simulated were:

1. Distribute probabilistically in proportion to server speeds.

2. Distribute probabilistically according to the Buzen-Chen fractions. (Used only for exponential service times.)

3. Send the work to the server which will complete it soonest.

4. Send the work to the server at which it will have the least total impact as judged by (4).

5. Send the work to the server at which the following simplified estimate of the impact is minimized:

$$x + \frac{w}{1-r}$$

   This is approximately equivalent to terms (1) and (2).

6. Send the work to the server where it actually has the least total impact. This is accomplished by allowing an all-powerful observer to manipulate both past and future assignments until the overall system completion time is minimized.

All of the algorithms are feasible solutions to the problem posed in section 1 except for the last one which produces an ideal solution.

Simulations were conducted for arrival rates which represented .1 to .9 of the total system capacity in steps of .1. Fixed service times were used.

7.2. Results

The average time to completion is plotted against rho, the fraction of system capacity represented by the arrival rate, in Figure 2. Each line is marked with a number representing the controller algorithm used for that set of simulation points.

The upper dotted line for algorithm "1" is the average time to completion when sending probabilitistically proportional to server

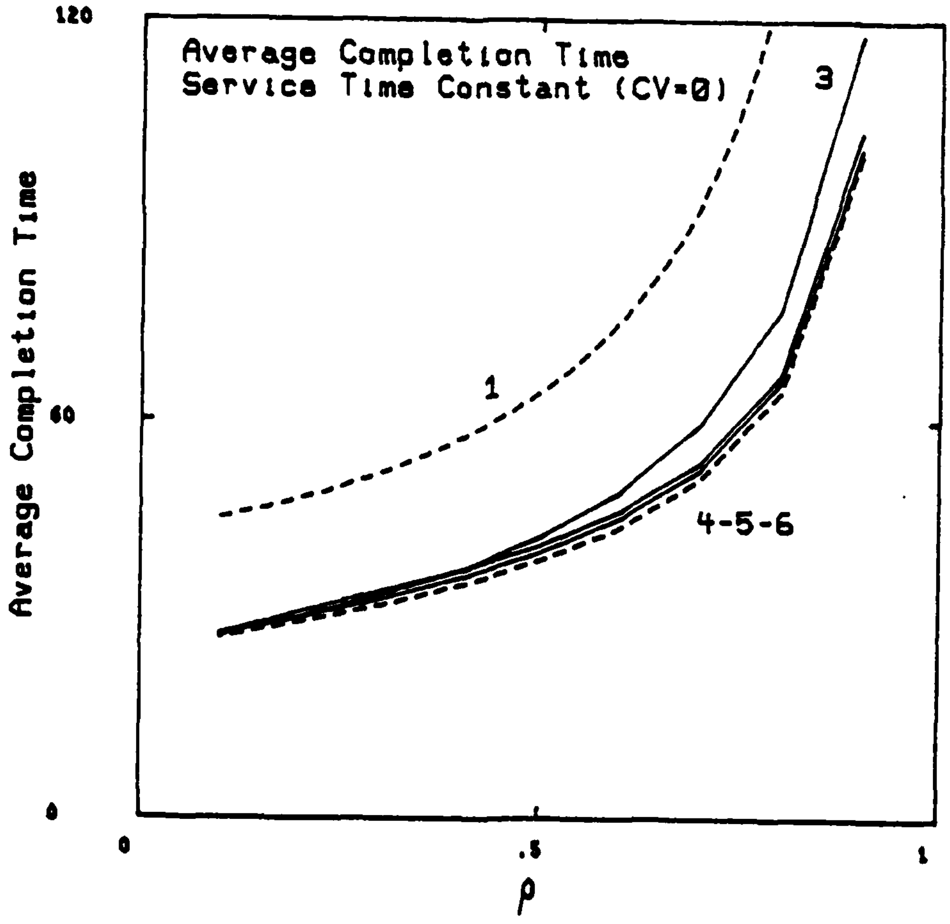


Figure 2

speed. It is markedly poor at all arrival rates because it does not use dynamic information on the state of backlogs to route work and will send new work to the slowest server even if the fastest one is idle.

The lower dotted line is the ideal average time to completion curve from algorithm 6.

The solid line for algorithm "3" shows the average time to completion when sending work to the server at which it will be completed soonest. At low arrival rates it performs well, but as the arrival rate rises its failure to take future arrivals into account results in higher than necessary average completion times.

Algorithms "4" and "5" take the future effect cost into account and have lower average completion times than algorithm "3". The choices made by "4" and "5" are so good that their delay curves lie very close to the ideal curve.

7.3. Discussion

Taking future costs into account reduces average completion time. The difference is more pronounced at higher arrival rates when the density of future arrivals is higher.

Controller algorithm 4 uses (4) to estimate the total additional increase in system completion time caused by routing the current arrival. To see how well (4) approximates the actual increase in completion times the simulation program was modified to record the extra impact on completion times due to each arrival. The data was collected into buckets according to the value of w at the time the controller decision was made. The average increases in completion times for work routed to the fastest server are shown as triangles in Figure 3 at rho=.7. For comparison, the impact predicted by (4) is shown as a straight line. The relationship is good and shows that the "future effect" cost is about 3 times as large as the observable cost of a routing decision.

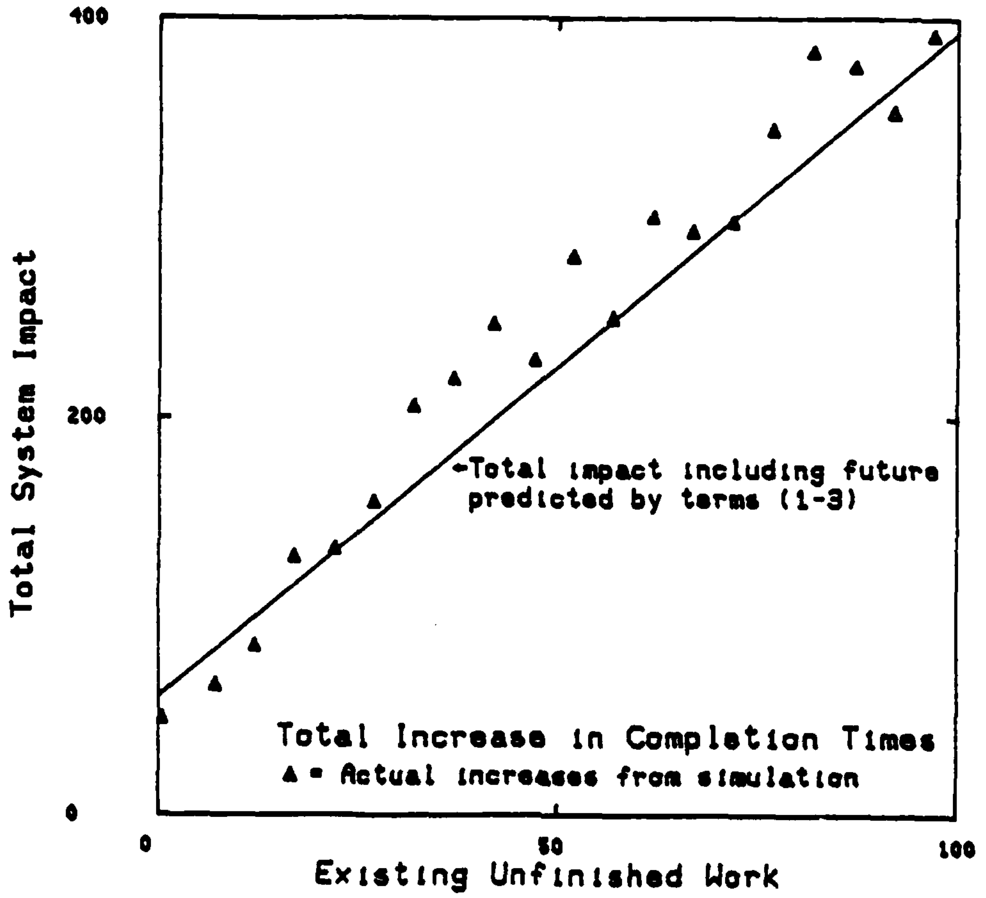


Figure 3

8. Limitations

Taking the future effect into account will not improve the average completion time under all circumstances.

8.1. Equal Speed Servers

In the case of equal speeds for the servers, the service time at all servers will be identical and terms (1) and (3) will yield identical values at all servers and may be disregarded. Therefore the least total effect will be produced by sending to the server which minimizes term (2) which can be considered a coefficient times the existing backlog. Sending to the server with lowest backlog is therefore the ideal strategy even considering the future effect in the case of equal speed servers.

8.2. Varying Service Times

If the service times are drawn from a distribution with considerable variance, the future effect is not as important as one having to do with the service time. In a loaded system the controller tends to equalize the sum of average service time, backlog, and future effect across all of the servers[1]. If a new arrival has a much shorter than usual service requirement, it will usually be sent to a slower server. To see why this is so, consider the sum given above. When it is approximately equal across servers, the average service time is

---

[1] This sum is the result of (4). The algorithm sends to the server which has the smallest value. This procedure tends to equalize the sum across servers.

largest at the slowest server (by definition), and the backlog and future effect must be larger at the fastest server (since the total is about the same). As a result, a low service requirement arrival will tend to look better on the slower servers since the backlog and future effect are comparatively small and the service time is a small fraction of the total.

On the other hand, a large service requirement arrival usually is routed to the fastest server because the service time at a slower server would overshadow the larger backlog and future effect at the fastest server.

To check this quantitatively, the ideal distribution scheme was used to route arrivals with service times drawn from an exponential distribution (rho=.9). The table below shows the fraction of work routed to each server (server 1 is the fastest, 3 the slowest).

Service Times	Fraction routed to server 1	2	3
Lowest 5%	.28	.35	.37
All	.50	.31	.19
Highest 5%	.72	.22	.06

The mean service times per server are correspondingly altered. The mean service time of work directed to the fastest server is 1.19 times the mean for all arrivals. The same factor for the medium speed server is .90 and for the slowest server it is only .67.

This tendency to distribute work according to service time further undermines the independent arrival assumption used by algorithm 4.

Algorithm 3 is sensitive to work distribution according to service time and is not substantially improved by considering the future costs. See Figure 4 which shows Algorithms 3, 4, 5, and 6 giving nearly identical performance. Algorithm 2 (Buzen-Chen fractions) is not as good as the others since it does not take current backlogs into account.

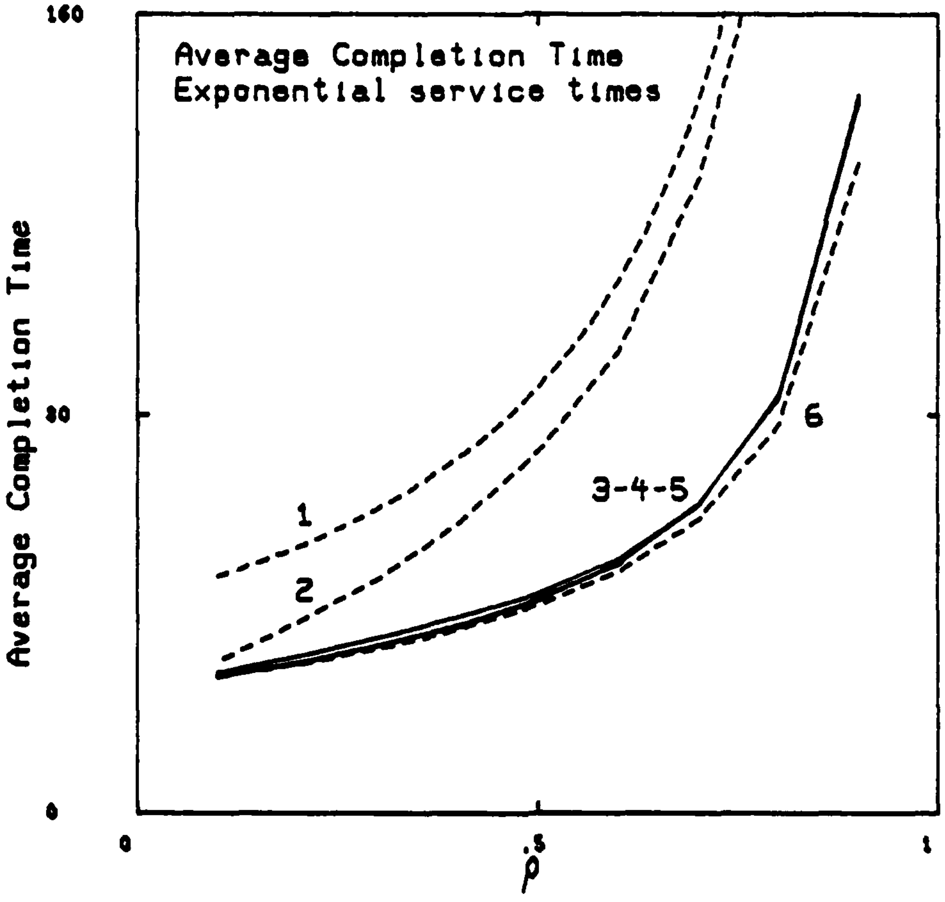


Figure 4

19. Weighting Unfinished Work

The additonal cost of an arrival can be broken into terms (1) and (3) which are constant for a constant service time and arrival rate, and term (2) which depends upon the existing backlog. Therefore the cost including the future effect is a_1x+a_2w where a_1 and a_2 are constants which depend upon the arrival rate and x is fixed for a given server. In algorithm 3 (send work to server with soonest completion time), the cost also takes this form but with $a_1=1$ and $a_2=1$. The future effect therefore modifies the relative weights to be given to the service time and unfinished work at a server.

These weights for algorithm 4 are:

rho	a_1	a_2	Ratio a_2/a_1
.1	1.054	1.111	1.054
.2	1.118	1.250	1.118
.3	1.198	1.429	1.192
.4	1.303	1.667	1.279
.5	1.448	2.000	1.381
.6	1.664	2.500	1.502
.7	2.023	3.333	1.648
.8	2.740	5.000	1.825
.9	4.889	10.000	2.046

The result of taking the future into account is to weight the amount of unfinished work more heavily than the service time. The exact amount of the extra weighting is not overly important. Algorithm 5 which sets a_1 to 1 and computes a_2 as in the table above gives average completion times almost indistinguishable from those of Algorithm 4.

20. Conclusion

In a work distribution model with servers of different speeds, the average completion time is not optimized by sending each task to the server at which it will be completed the soonest. The impact of any decision on future arrivals must be taken into account.

The information available to a controller may be categorized as static or dynamic, and "past" or "future". Static information is known by the controller and will not change. The static information includes the mean arrival rate and the server speeds. Dynamic information changes with time; dynamic information such as current backlogs must be exploited to achieve system goals. The "past" information is that body of knowledge of what has happened in the past. With Poisson arrivals, the complete state of past dynamic knowledge can be summarized for operational purposes by giving the current server backlogs. There is no "future" dynamic information...it is unknown. But there is an important piece of "future" static information: the mean arrival rate.

A successful controller must consider (a) The past dynamic information summarized by the current backlogs, (b) The current arrival and its requirements, and (c) The best projection of the future available from the "future" static information.

If no quantitative picture of the future is available, the future cost of a decision cannot be computed. But few arrivals are the "last" arrival and a controller can generally assume that the future arrival rate and pattern will be approximated by the recent past. This is sufficient future information to allow a controller to estimate the probabilistic costs of its decisions on future arrivals.

A controller's best estimate of the future effect impact will in general be conditioned on all of the "past" information available. Even if the future effect has no simple relationship to this "past" information it may be estimated using historical information. Using some estimate will usually be better than no estimate, and the system is likely to be insensitive to all but gross errors.

REFERENCES

1. Agrawala, A. K., Tripathi, S., and Ricart, G., "Adaptive Routing *Using a Virtual Waiting Time Technique*", *University of Maryland* Technical Report TR- , November 1979.

2. Bruno, J., Coffman, E.G. Jr., and Sethi, R., Scheduling Tasks to Reduce Mean Finishing Time, Comm. ACM 17, 7, July 1974, pp. 382-387.

3. Buzen, J.P., and Chen, Peter P.S., Optimal Load Balancing in Memory Hierarchies, Information Processing 74, North Holland, pp. 271-275.

4. Clark, Douglas, Scheduling Independent Tasks on Non-Identical Parallel Machines to Minimize Mean Flow-Time, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa., June 1974.

5. Gonzalez, Mario J. Jr., Deterministic Processor Scheduling, Computing Surveys, Vol. 9, No. 3, Sep. 1977, pp. 173-203.

6. Kleinrock, Leonard, Queueing Systems, Vol I: Theory, John Wiley and Sons, Inc., New York, 1975, pp. 206-223.

7. Liu, Jane W.S., and Yang, Ai-Tsung, Optimal Scheduling of Independent Tasks on Heterogeneous Computing Systems, Proc. ACM National Conf. 1974, Vol 1, ACM, N.Y., 1974, pp. 38-45.

8. Ricart, Glenn, and Agrawala, A.K., Some Strategies for the Multiple Producer / Multiple Consumer Problem, Technical Report TR-816, University of Maryland Computer Science Department, October 1979.

9. Takàcs, L., Introduction to the Theory of Queues, Oxford University Press, New York, 1962