Sādhanā, Vol. 15, Parts 4 & 5, December 1990, pp. 381–395. © Printed in India.

Task allocation in multiserver systems – A survey of results

ANURAG KUMAR

Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India

Abstract. Jobs consisting of one or more tasks arrive to a system comprising several servers, each with its own queue. Each task requires a single service at any of the servers, and a job completes service when all its constituent tasks have been serviced. Such models arise in the performance modelling of distributed computing systems, computer communication networks, and manufacturing systems. We survey the literature on this class of models.

We classify jobs as being of one of three types: single tasks, multitask with precedence constraints, and multitask stream jobs. After surveying the optimal allocation problem for single task jobs, we discuss the results on the performance analysis and optimal allocation of tasks for the other two job types.

Keywords. Control of queues; customer allocation; routing; load balancing.

1. Introduction

This paper is a summary of published, and a few to be published, results pertaining to customer allocation in single station queueing systems with multiple servers. Customers or jobs, each comprising one or more tasks, arrive to the system. Each task requires a single service from a server, after receiving which, it departs. The service of a job is complete after all its constituent tasks have been served. Although servers may have different speeds, each server can serve any task, and tasks do not inherently have any preference to be served at any particular server. We study the effect on job sojourn time of various policies for allocating tasks to servers.

Figure 1 is a schematic representation of the general model described above. Our description has left unspecified the number and location of the queues, the allocation strategy of customers to queues, and the strategy for allocating servers to service the queued customers. Thus this general model gives rise to several subclasses of models depending on the choices we make among one of the many alternatives we have in the general model.

1.1 *Job type*

Each job may consist of a *single task*. Each task is given one indivisible/uninterruptible service by a server. After completion of this one task the job departs. In this case the job sojourn time is the sojourn time of the task. This is the classical problem, and

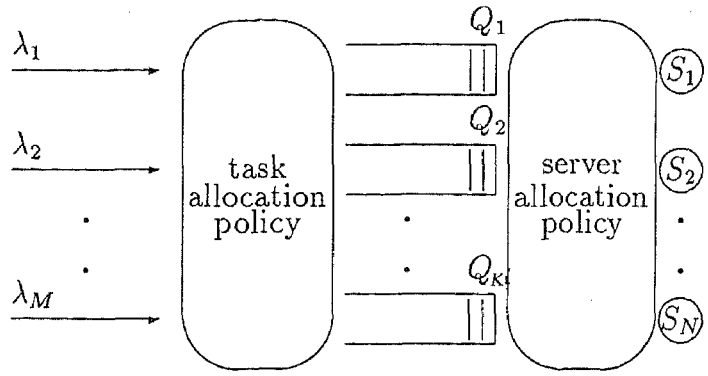

Figure 1. Schematic of the general problem of task allocation to multiple servers.

has received much attention in the literature. Alternatively, each job may comprise *several tasks* (in general, a random number of tasks) that have some precedence relation between themselves. A job is completed when all the tasks in it have been completed. All the tasks may arrive together, in which case the job sojourn time is the time between the arrival epoch of the job and the departure epoch of the last task. Such a model arises in parallel processing and manufacturing systems (Baccelli & Makowski 1990). Lastly, the tasks in a job may arrive in a stream, and must depart in the same order in which they arrived. Here we will only be concerned with the task sojourn times, since job sojourn time has no useful meaning in this case. Such jobs arise as models of virtual circuits in a computer communication network (Jean-Marie 1988, pp. 75–88; Kuri 1990).

1.2 *Queueing and service strategy*

The number of queues K is, in general, less than, equal to, or greater than M. The servers may be *statically allocated* to queues, i.e., each server serves customers from only one queue. Alternatively, the servers may not be attached to queues, but there is a *static policy* for the allocation of servers to queues, e.g., cyclic service. Finally the assignment of servers to queues may be *dynamic*, in the sense that system state information may be used to decide which queue(s) to serve next (Lin & Kumar 1987).

The focus of this paper is on the class of models depicted in figure 2. Server $i, 1 \leqslant i \leqslant N$, is statically assigned to queue $i, 1 \leqslant i \leqslant N$. This is a commonly occurring model, and our own work on this model has been motivated by models for multicomputer systems, and of communication links emanating from a packet switch. In the latter case a "job" corresponds to a virtual circuit which is a stream of packets (tasks).

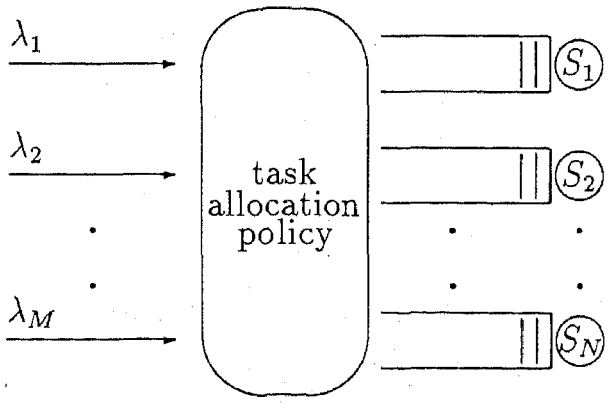

Figure 2. Schematic of the class of problems surveyed in this paper.

1.3 *Task allocation strategy*

The task allocation strategy may be *static*, which means that no system state information, nor any record of past allocations, is used to decide which queue to allocate a task to. Probabilistic allocation is a static allocation strategy. *Semidynamic* allocation strategies do not use dynamic information of system state, but do keep track of past allocations to decide where to put the next task (Agrawala & Tripathi 1981; Yum 1981). Cyclic (or round-robin) allocation belongs to this class of strategies. Finally, a *dynamic* allocation strategy uses system state information when allocating a task. Information that may be used includes queue lengths, work in the system, residual service (Weber 1978; Ephremides *et al* 1980).

In this paper we will consider all the types of task allocation strategies discussed above. We will, however, not permit "jockeying" and preemption. Tasks will be irrevocably assigned to a queue, and will be served to completion, without preemption, by the server at that queue.

The rest of the paper is organised as follow. In § 2 we discuss the classical problem in which each job has a single task. In § 3 we present the more recent results for the class problems with several tasks per job, all of which arrive together. In the context of a continuous arrival stream of such jobs, the only precedence constraints that seem to have been considered in the literature are the fork-join constraints, that give rise to the class of fork-join queueing models. In § 4 we turn our attention to jobs that consist of a stream of tasks that must depart in sequence. We conclude in § 5 with a discussion of some problems for further research.

2. Single task jobs

There are N servers, each with its own queue. Several job streams arrive to the system. Some of the streams may be dedicated to certain servers and others may be allocable. This class of models also arises in the study of load sharing (or load balancing) in computer systems. We begin by assuming that the servers have the same service rates.

We start with a fundamental result that bounds the performance of this class of models.

2.1 *A general lower bound*

If the service times across all the arriving jobs are independent and identically distributed (i.i.d.), then it is proved in Wolff (1977) that the system shown in figure 3

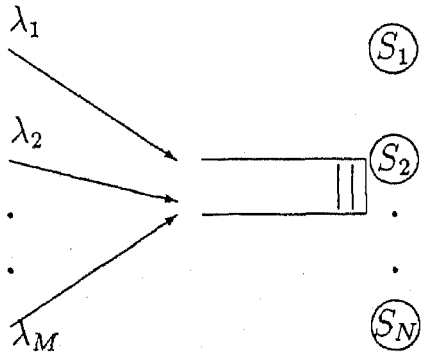

Figure 3. The lower bound system (G/GI/N).

(the G/GI/N system) is a lower bound to any system in the class shown in figures 1, or 2, in the sense of the following theorem

Define $Q(t)$ = queue length at time t for the G/GI/N system and $Q'(t)$ = queue length at time t for any other queueing and allocation policy.

Theorem 1. *For all $t \geqslant 0$, and for all $k \geqslant 0$, $P(Q(t) \geqslant k) \leqslant P(Q'(t) \geqslant k)$, i.e., $Q'(t) \geqslant_{st} Q(t)$, where $\geqslant_{st}$ is read as "stochastically greater than or equal to" (i.e., the stochastic ordering).*

Proof. The proof (Wolff 1977) uses the coupling argument (see Kemae *et al* 1977, Smith & Whitt 1981; Ross 1983). Two new queueing processes with queue lengths $\tilde{Q}(t)$ and $\tilde{Q}'(t)$ are constructed by sampling the service times of tasks as they enter service. Thus on each sample path ω, the nth task to enter service in both systems has the same service time. It is shown that $\tilde{Q}(t)(\omega) \leqslant \tilde{Q}'(t)(\omega)$ for all $t \geqslant 0$, and all ω. Owing to i.i.d. service times (and the task selection policy not depending on the service times of queued customers) $\{\tilde{Q}(t), t \geqslant 0\}$ has the same law as $\{Q(t, t \geqslant 0\}$, and $\{\tilde{Q}'(t), t \geqslant 0\}$ has the same law as $\{Q'(t), t \geqslant 0\}$. This completes the coupling argument and the stochastic comparison holds. □

If a stationary queue length distribution exists in each case, with EQ and EQ' denoting the limiting time average queue lengths, it follows by Little's Law that $EW \leqslant EW'$.

It is intuitively clear, and this is also the key point in the proof in Wolff (1977), that the G/GI/N system yields the best queueing performance since it does not allow a situation where there is a task waiting at one queue and an idle server is available at another queue. In a typical distributed system (one of the motivations for the class of models under study here), owing to communication delays, the G/GI/N model is inapplicable since this would require that the idleness of a server be communicated to all the queues instantaneously. Recall, further, that in this survey we are only considering models in which tasks are irrevocably assigned to queues, i.e., there is no "jockeying". Models that allow task migration have been studied by Eager *et al* (1986) and Mirchandaney *et al* (1989). If communication delays are small compared to task service times then it may be possible to approach the G/GI/N performance.

That the policy for allocating tasks to queues, as the tasks arrive, can have a significant effect on delay performance is demonstrated by the following simple example. Consider the model in figure 2 with $N = 2$ and a single Poisson arrival process of tasks, with rate λ, each requiring service with an exponential distribution with mean μ^{-1}, at either server. Assume that $\rho = \lambda/2\mu = 0{\cdot}9$, and $\mu = 1$. The M/M/2 ideal yields a mean task sojourn time of 4·26, whereas cyclic (alternating) allocation yields 6·81, and random allocation, with probability $\frac{1}{2}$ to each server, yields 9·00.

2.2 *Dynamic policies*

The most well-known references (Winston 1977; Weber 1978; Ephremides 1980) concerning dynamic policies all conclude that, by one criterion or another, the "Join-the-Shortest-Queue" discipline is optimal. We discuss the result of Weber (1978).

There are N identical servers, the tasks arrive according to an arbitrary process and have i.i.d. service times with non-decreasing hazard rate, i.e., if B denotes the service time random variable, $B(\cdot)$ denotes the distribution and $b(\cdot)$ the density, then $b(t)/(1 - B(t))$ is nondecreasing in $t, t \geqslant 0$. An arriving task can observe the state

$(Q_1, Q_2, \ldots, Q_N; X_1, \ldots, X_N)$, where $Q_i, 1 \leqslant i \leqslant N$, is the number of customers waiting in queue i, and X_i is the amount of service already rendered to the customer in service, with $X_i = 0$ if the server i is idle. The task allocation policy S^* is such that if an arriving task finds the state $(q_1, q_2, \ldots, q_N; x_1, \ldots, x_N)$ then S^* assigns the task to queue k, where $k = \arg\min_{1 \leqslant i \leqslant N}(q_i EB + E(B - x_i | B > x_i))$, i.e., the policy assigns the task to the queue with the least expected work. Let $D(t)$ be the departure process of the system. Denoting by $P^*_{(\mathbf{q}_0, \mathbf{x}_0)}$ (respectively $P_{(\mathbf{q}_0, \mathbf{x}_0)}$) the probability measures under policy S^* (respectively S) and initial state $(\mathbf{q}_0, \mathbf{x}_0)$, the following result is obtained.

Theorem 2. *$P^*_{(\mathbf{q}_0, \mathbf{x}_0)}(D(t) \geqslant k) \geqslant P_{(\mathbf{q}_0, \mathbf{x}_0)}(D(t) \geqslant k)$, i.e., the number of departures till time t is stochastically greater under S^* than under S.*

Proof. See Weber (1978). □

Letting $M(t)$ denote the number of tasks in the *system* at time t, and observing that $M(t) = M(0) + A(t) - D(t)$ where $A(t)$ is the arrival process, we have the following result.

COROLLARY.

(i) *$P^*_{(\mathbf{q}_0, \mathbf{x}_0)}(M(t) \geqslant k) \leqslant P_{(\mathbf{q}_0, \mathbf{x}_0)}(M(t) \geqslant k)$ i.e., the number in the system at time t is stochastically smaller with S^* than with any other policy.*
(ii) *If limiting distributions exist for $M(t)$ under each policy and are denoted M^* and M, then $M^* \leqslant_{st} M$, where $\leqslant_{st}$ reads "stochastically less than or equal to".*
(iii) *If ET^* and ET denote the mean task sojourn times then $ET^* \leqslant ET$.*

Proof. (i) is obvious, and (ii) follows from the weak convergence property of stochastic ordering (Stoyan 1983); (iii) is an immediate consequence of Little's Law. □

Observe that when the system is restricted to exponential service times (but an arbitrary arrival process) then the policy S^* becomes the join-the-shortest-queue (JSQ) policy. Winston (1977), assuming Poisson arrivals and exponential service times, shows that JSQ is optimal in the sense of stochastically maximising the discounted number of departures till time t(i.e., $\int_0^t e^{-\alpha u}\, dD(t)$ where $\alpha > 0$). For the special case of two queues, Ephremides *et al* (1980) assume exponential service times, and arbitrary arrivals, to show that JSQ is optimal for minimising the expected time till the system becomes empty, if the arrival stream is stopped at some time t. The latter paper also considers an important variation on the information structure of the problem. It is shown that if queue lengths are not observable but are known to be equal at the beginning, then the policy of alternately assigning arrivals to queues is optimal for minimising the same cost function.

Whitt (1986) provides a counterexample that shows that the nondecreasing failure rate condition in theorem 2 is necessary. Whitt considers a service time distribution that has a point mass of $1 - \varepsilon$ at 0 and a point mass of ε at n, where ε is small and n is large. This distribution clearly does not have nondecreasing failure rate (see Wolff 1989, p. 480). Whitt argues that for this service time distribution, a policy which follows JSQ if either queue is empty or if the queue length difference is less than or equal to 1 but assigns the task to the longer queue if both queues are nonempty and the difference is 2 or more, is better than pure JSQ. Suppose the queues start out

empty and consider the first time that both servers become busy. At this time both servers are serving tasks whose service times are n, and there is one task in each queue. Eventually the task at the head of one of the queues will complete, and with a high probability the queue will empty out. Arrivals will now join this queue until an arrival with nonzero service time occupies the server. The queue length difference, has a high probability now of being more than 1. It is now beneficial for the next arrival to join the longer queue since the task at the head of this queue will finish earlier than the task in the other queue. Since the other tasks queued up in the longer queue will, with a large probability, have zero service time, the new arrival will enter service earlier if it joins the longer queue. Whitt provides the "epsilonics" to tighten these arguments.

All the results discussed so far in this section have been for identical servers. Clearly if there are two queues and one server is much faster than the other, then, until the queue at the faster server builds up beyond a certain level, tasks should not be assigned to the slower server. Hajek (1984) considers a general model of two interacting service stations of which the following model is a special case. Tasks arrive in a Poisson process, and join one of two queues; at the first queue the service time distribution is exponential with rate μ_1 and at the other the distribution is exponential with rate μ_2. Let $Q_1(t)$ and $Q_2(t)$ denote the number of tasks in queue 1 and queue 2. Two cost functions are considered: Discounted weighted queue length average,

$$E\int_0^\infty e^{\alpha t}(c_1 Q_1(t) + c_2 Q_2(t))\,dt,$$

and long run time average of weighted queue lengths,

$$\lim_{T\to\infty}\frac{1}{T}E\int_0^T ((c_1 Q_1(t) + c_2 Q_2(t))\,dt,$$

where $\alpha > 0$ is the discount factor, and $c_1 > 0, c_2 > 0$ are cost rates.

In either case it is established that the optimal strategy is of *switch-over type*, as stated in the following theorem.

Theorem 3. *There is a nondecreasing function* s: $\{0,1,2,\ldots\}\to\{0,1,2,\ldots\}$, *such that the optimal strategy sends a customer arriving at* t *to queue 1 if* $Q_2(t)\geqslant s(Q_1(t))$, *and to queue 2 otherwise.*

Proof. See Walrand (1988). □.

This result demonstrates the form of the optimal policy, but the switching curve is difficult to compute (of course, if $\mu_1 = \mu_2$ then the earlier results show that $s(i) = i, i \geqslant 0$). As a practical heuristic, for the case with nonidentical servers, Yum & Schwartz (1981) discuss the Join-Biased-Queue (JBQ) rule. For example if $\mu_1 > \mu_2$ there is an integer Δ such that an arrival is sent to queue 2 only if $Q_1 > Q_2 + \Delta$. Using Markov chain analysis the optimal Δ is found for several examples.

Another class of results has to do with the heavy traffic behaviour of queueing systems subject to the optimal allocation policies discussed in this section. It is shown that suitably time scaled and normalised versions of the queue length processes converge weakly in heavy traffic to a certain diffusion process (namely, reflected Brownian motion) (see Foschini 1977, Foschini & Salz 1978, and Reiman 1983 for

the original results, and Flores 1990 for a survey). We do not give the results here but merely state them informally. The heavy traffic limit corresponds to considering a sequence of queueing systems (indexed by n) with the difference between the total arrival rate and total service rate going to zero as $O(1/n^{\frac{1}{2}})$. For two queues with renewal inputs and identical servers, and JSQ allocation, the time scaled and normalised queue length processes for the two queues are considered. It is shown that in the heavy traffic limit the difference between the normalised queue lengths goes to zero in probability, and the sum of the normalised queue lengths converges weakly to the same diffusion limit as the GI/GI/2 system. These results suggest that under heavy traffic JSQ tends to equalise queue lengths, and the total number in the system behaves just like the lower bound GI/GI/2 system (see § 2·1). These insights into the behaviour of the JSQ policy have been used in Nelson & Philips (1990) to develop simple and accurate approximate analysis techniques for queues with JSQ allocation. The difficulty of exact analysis is well-known (see Flatto & McKean 1977).

2.3 *Semidynamic policies*

These are policies that do not use system state information but do keep a record of their past decisions. Consider the allocation of tasks arriving in a Poisson stream of rate λ, to two queues with identical i.i.d. service times at each. If instantaneous state information is not to be used, two simple allocation policies come to mind. Assign the tasks to each queue with probability $\frac{1}{2}$, or assign the tasks alternately to the two queues. In the former case each queue becomes an M/G/1 queue and in the latter case each queue is an E_2/G/1 queue, and in both cases the mean interarrival time is $2/\lambda$. Owing to the fact that the E_2 distribution is smaller in the convex ordering sense than the exponential distribution with the same mean, it follows (Wolff 1989) that the waiting time with round-robin allocation is smaller in the convex ordering sense. In particular, round-robin allocation gives smaller mean delay.

As has already been discussed in § 2·2 it is shown by Ephremides *et al* (1990) that for identical exponential servers, with equal initial queue lengths, alternating allocation is optimal. This result is extended by Walrand (1988) to multiple queues and the cost function $E\{\int_0^\infty e^{-\alpha t}\Sigma_{i=1}^N Q_i(t)\mathrm{d}t\}$.

For servers with unequal service rates it is still an open question as to what the optimal semidynamic allocation policy is. One possibility is to find the best probabilistic allocation policy (this might, for example, yield $u_1 = \frac{2}{3}, u_2 = \frac{1}{3}$, i.e., allocate an arrival to queue 1 with probability $\frac{2}{3}$ and to queue 2 with probability $\frac{1}{3}$) and then implement these allocation ratios cyclically (in the example, route customers according to the pattern $1, 1, 2, 1, 1, 2, 1, 1, 2, \ldots$). It is shown in Agrawala & Tripathi (1981) by a counterexample that this, in general, does not yield an optimal cyclic policy in the sense of minimising the mean task delay.

Hajek (1985) considers the following problem. Tasks arrive as a renewal point process to a queue where they require exponentially distributed i.i.d. service. Only a fraction $0 < p < 1$ has to be admitted into the queue. A selection sequence is a 0–1 valued sequence $\{r_k, k \geqslant 1\}$ such that $\lim_{n\to\infty}(1/n)\Sigma_{k=1}^n r_k = p$, and task k is accepted into the queue if $r_k = 1$. Hajek introduces the regular sequence $r_k^* = [(k+1)p] - [kp]$, where $[\cdot]$ denotes "greatest integer less than or equal to". Observe that for rational $p = m/d$ the sequence has period d and there are exactly m 1's in each period. The following theorem is proved.

Theorem 4. *The selection sequence* $\{r_k^*\}$ *minimises* $\lim_{n\to\infty}(1/n)E\Sigma_{k=1}^{n}Q(t_k)$ *over all selection sequences (where* $\{t_k, k \geqslant 1\}$ *are the external arrival epochs before selection/rejection).*

Proof. See Hajek (1985). □

Observe that if the arrival process is Poisson then the cost function in theorem 4 is the same as the time average queue length, and hence by Little's Law $\{r_k^*\}$ also minimises mean delay. For non-Poisson arrivals, the cost function is the average number of customers seen by all arrivals (whether they are accepted or not), and thus minimising it does not, in general, minimise the mean delay of accepted customers. The import of this cost function in these cases is not clear.

2.4 *Static policies*

In static policies, no information is used about system state or past allocations. Probabilistic policies that base the allocation probabilities on a priori distributional information are static policies. The most well-studied problem in this class is the probabilistic allocation of tasks arriving in a Poisson stream and requiring general i.i.d. service. The servers have possibly unequal service rates. See Chow & Kohler (1979), de Souza e Silva & Gerla (1984), Tantawi & Towsley (1984), Ni & Hwang (1985) and Bonomi & Kumar (1990). In each case the cost function is the mean task sojourn time over all task streams. Since probabilistic splitting retains the Poisson nature of the arrivals, the mean task sojourn time can be written down explicitly. Minimisation of this over the splitting probabilities is a nonlinear programming problem that can be solved with the Lagrange multiplier technique. Explicit results can be obtained and are given, for example, in Bonomi & Kumar (1990).

In addition to obtaining the explicit solution of the mean delay minimisation problem, it is also shown in Bonomi & Kumar (1990) that the minimisation problem is equivalent to least squares idle time balancing of the queues. This result is then used to develop an adaptive algorithm for on-line learning of the optimal splitting probabilities, via a stochastic approximation technique.

3. Multitask jobs with precedence constraints

The basic system structure is still that shown in figure 2, except that now each job comprises a batch of tasks with precedence constraints between them. Figure 4 shows a batch of 7 tasks $\{T_1, T_2, \ldots, T_7\}$ with precedence constraints between them; for example, the directed edges between T_3 and T_6 and T_4 and T_6 indicate that processing of T_6 cannot begin until T_3 and T_4 are fully processed. This relation is symbolically denoted by $T_3 \succcurlyeq T_6$ and $T_4 \succcurlyeq T_6$. Generically a precedence relation is denoted by Γ. It is assumed that the graph representation of this partial order (as in figure 4) yields a *directed acyclic graph.*

A job is complete only when all its constituent tasks have been served. Thus job sojourn time is the time between the arrival of the job and the completion of processing of the last task in the job to be served.

Essentially all the work done on this problem has been for the following subclass of models. The job interarrival times are i.i.d., and the ith job brings a random batch

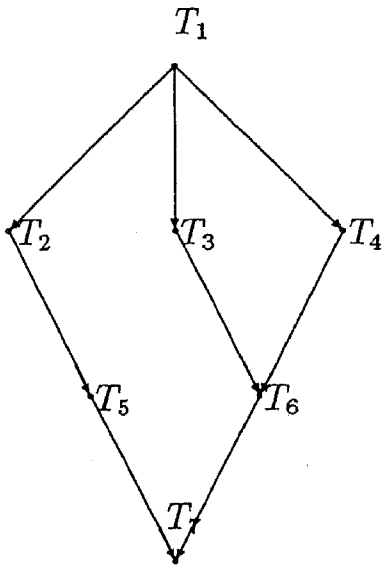

Figure 4. A directed acyclic graph representation of precedence constraints between tasks in a job.

of tasks of size C_k, where the $\{C_k, k \geq 1\}$ form an i.i.d. sequence. The "precedence" constraints are simple; all the tasks can be processed independently without having to wait for each other, but the job finishes processing only after all its tasks have been fully served. This is the class of *fork-join* models. Most of the results in the literature are for the special case in which $C_k = N$ (the number of servers) for all $k \geq 1$, and upon arrival, one task is assigned to each of the N queues. See Flatto & Hahn (1984), Nelson & Tantawi (1988) and Baccelli *et al* (1985, 1989). These papers essentially concern themselves with exact or approximate analysis of the fork-join queue. Recently some of these results have been extended to nondeterministic task batch sizes (see Shorey 1990). The latter work also studies various task allocation policies, and the effect of task batch size distributions.

In this survey we shall consider the special case of renewal job arrival epochs, i.i.d. task batch sizes, i.i.d. task service times, and servers with identical service rates. An exact analysis of this problem is extremely difficult even for the simplest case of two queues, Poisson arrivals, and exponential service times (Flatto & Hahn 1990). Subsequently, researchers have limited themselves to finding bounds and approximations.

As before the performance measure of interest is job sojourn time. The following stability condition assures the existence of a limiting distribution for the sojourn time.

Let B denote the task service time random variable, and λ the job arrival rate.

Theorem 5. *There exists a task service discipline such that the fork-join system is stable, if* $\lambda EC\,EB < N$.

Proof. The result (Shorey 1990), in fact, holds for the general system with arbitrary precedence relation Γ_k for the batch C_k. Since Γ_k is a directed acyclic graph, there exists a $\bar{\Gamma}_k$ that is a directed chain, and each precedence relation in Γ_k is contained in $\bar{\Gamma}_k$. If we serve the tasks in C_k as if they had the precedence relation $\bar{\Gamma}_k$, then we just have a GI/GI/N queue with traffic intensity $\lambda ECEB$. This queue is stable under the stated condition. Hence, we have a task service discipline that under the given condition renders the system stable. □

First consider the case with $C_k = N, \forall k \geqslant 1$, and one arriving task is allocated to each queue. Assuming that the stability condition holds, each queue is now a stable GI/GI/1 queue. Denote by $W^{(i)}(t)$ the stationary waiting time random variable at the ith queue, $1 \leqslant i \leqslant N$. Then the stationary job sojourn time T is given by

$$T = \max_{1 \leqslant i \leqslant N} (W^{(i)} + B^{(i)}),$$

where $B^{(i)}$, $1 \leqslant i \leqslant N$ are i.i.d., with the distribution of B, the task service time.

The distributions of $W^{(i)}$, $1 \leqslant i \leqslant N$ can be determined explicity for many special cases, but the difficulty is that $\{W^{(i)}, 1 \leqslant i \leqslant N\}$ are not mutually independent. Bounds, however, can be developed as follows. First replace the arrival process of jobs with a process with deterministic interarrival times equal to $1/\lambda$ (where λ is the arrival rate). Let $\mathbf{W}^{(i)}, 1 \leqslant i \leqslant N\}$ denote the steady-state waiting time random variables for the new system. It is clear that $\{\mathbf{W}^{(i)}, 1 \leqslant i \leqslant N\}$ are mutually independent. Let

$$\mathbf{T} = \max_{1 \leqslant i \leqslant N} (\mathbf{W}^{(i)} + B^{(i)}).$$

Further, let $\{\bar{W}^{(i)}, 1 \leqslant i \leqslant N\}$ be mutually independent with $\bar{W}^{(i)}$ and $W^{(i)}$ having the same distribution for each i. Let

$$\bar{T} = \max_{1 \leqslant i \leqslant N} (\bar{W}^{(i)} + B^{(i)}).$$

Theorem 6. $\mathbf{T} \leqslant_c T \leqslant_{st} \bar{T}$, *where* $\leqslant_c$ *denotes the convex increasing ordering.*

Proof. The lower bound intuitively (Baccelli *et al* 1989) states that "determinism minimises queueing delay". The upper bound uses the fact (proven in Baccelli *et al* 1989) that $(W^{(i)}, 1 \leqslant i \leqslant N)$ are associated (see Barlow & Proschan 1975). □

The significance of this theorem is that it yields computable bounds to the mean of T, since on each side we have the maximum of independent random variables.

The upper bound has been extended in Shorey (1990) to random batch size C, and probabilistic allocation of tasks to queues (i.e., there are probabilities $p_i, 1 \leqslant i \leqslant N$, $0 \leqslant p_i \leqslant 1$, $\Sigma_{1=i}^{N} p_i = 1$, such that an arriving batch is multinomially partitioned over the N queues using these probabilities). Let $\{C^{(i)}, 1 \leqslant i \leqslant N\}$ be the subbatches allocated to the queues. The main difficulty in extending theorem 6 is that the subbatches are dependent, in general. Note that if C is Poisson then the subbatches are independent and theorem 6 applies. Let $W^{(i)}$ denote the stationary waiting time at the ith queue, let $D^{(i)}$ denote the total service time of the subbatch assigned to the ith queue, and let $C^{(i)}$ denote the subbatch size assigned to the ith queue. Then the job sojourn time is given by

$$T = \max_{1 \leqslant i \leqslant N} (W^{(i)} + D^{(i)}) I_{\{C^{(i)} > 0\}},$$

where $I_{\{\cdot\}}$ is the indicator of $\{.\}$. In the expression for T, each term inside the maximum is multiplied by $I_{\{C^{(i)}>0\}}$ since a possibly empty batch may be assigned to a queue. Let $\{X^{(i)}, 1 \leqslant i \leqslant N\}$ be mutually independent, such that $X^{(i)}$ has the same distribution as $(W^{(i)} + D^{(i)}) I_{\{C^{(i)}>0\}}$. Let $\bar{T} = \max_{1 \leqslant i \leqslant N} X^{(i)}$.

Theorem 7. *If C has a geometric distribution then* $T \leqslant_{st} \bar{T}$.

Proof. The proof (Shorey 1990) rests on the result that the components of a multinomial partition of a geometric random variable are associated. □

Numerical results presented in Shorey (1990) show that the upper bound to ET yielded by theorem 7 is also a good approximation to ET. It is found that for the same mean batch size, a geometric batch size distribution yields larger mean delay than a Poisson batch size. This may be attributed to the larger variability in the geometric distribution.

Some simple, albeit loose lower bounds for the mean job delay in the general fork-join delay model are also given in Shorey (1990).

Simulations have been used to compare three task allocation strategies: probabilistic (as discussed above), cyclic (tasks are assigned to queues in a round-robin fashion), and JSQ (tasks arriving in a batch are successively assigned to the shortest queue). In each example studied it is found that the JSQ yields the least mean job delay, and the probabilistic strategy, the most. Thus the ordering observed in § 2 seems to be preserved, but we do not have a proof for this, nor is it known whether JSQ is still optimal in any sense.

The simulations yielded an interesting observation. When JSQ as just described (JSQ by task) was compared with JSQ by batch (i.e., the entire arriving batch of tasks is put into the shortest queue), it was found that whereas at light and moderate loads JSQ by task performed better, at heavy loads the performance of the two became equal. This has important implications since, in practice, JSQ by task requires interprocessor communication overheads, which we do not incorporate in our analysis. Thus at heavy loads it may be better to switch the allocation policy to JSQ by batch. This simulation result may be explained in terms of the heavy traffic results for JSQ discussed in § 2·2.

4. Stream jobs

A stream job comprises a sequence of tasks that arrive over time. Each task can be served independently at any of the N servers, and hence upon arrival it may be allocated to any of the N queues. The constraint is that the tasks must depart the system in sequence. Consequently, there is a need for a *resequencing buffer* following the servers (see figure 5). A stream job models packet arrivals in a virtual circuit in a computer network.

Very few results are available on this problem. Most of the available results concern a single stream job extending over infinite time, i.e., a stream of arrivals that must be served by any of the servers and must depart in sequence. Such models have been studied in Varma (1987), Jean-Marie (1988), Gün & Jean-Marie (1990) and Jean-Marie & Gün (1990). Recently some results have been obtained for the more realistic problem of a sequence of stream jobs each of finite extent; it is only necessary to maintain order between tasks belonging to a particular job (Kuri 1990). Further, as in § 3, most of the work concerns itself with the analysis of task delays under various assumptions and allocation policies. Some results regarding "good" allocation policies are discussed in Kuri & Kumar (1991).

Jean-Marie (1988) considers a problem with two independent Poisson streams of

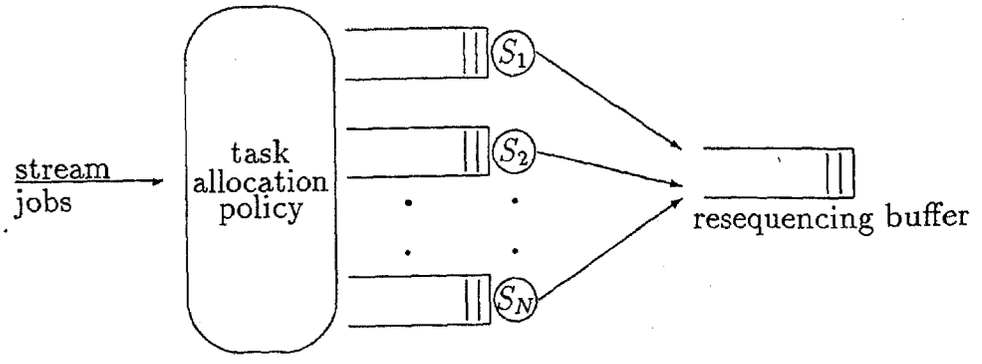

Figure 5. Stream jobs require a resequencing buffer.

tasks, that arrive to two queues with equal speed servers. The tasks in both the streams have i.i.d. service requirements with an exponential distribution. A task from stream 1 is allocated to queue 1 with probability p_1, and a task from stream 2 is allocated to queue 1 with probability p_2. The mean sojourn time, over all tasks of either stream, is derived as a function of p_1 and p_2. The problem of finding (p_1, p_2) that minimises this mean sojourn time is then studied.

The as yet unpublished results in Varma (1987), Gün & Jean-Marie (1990) and Jean-Marie (1990) are reported on in a survey article (Baccelli & Makowski 1990). These studies provide stochastic comparison results that help in bounding the delay performance of a single task stream that must be allocated to N queues, and must depart in order. For example, similar in spirit to the result in §2·1, it is shown that the delay through N G/G/1 queues is stochastically lower bounded by a system in which there are an infinite number of servers (the G/G/∞ system).

Motivated by performance comparison between virtual circuit and datagram packet networks (Tanenbaum 1988), some simple models are studied in Kuri (1990). A job (or call) has a holding time that is exponentially distributed with parameter μ; during a call, tasks (or packets) arrive at the rate λ. Each packet can be put into one of two queues with identical service rates; the packets have an exponential service time distribution with mean 1. Packets within a call must emerge from the queueing system in sequence. After the completion of a call another call arrives immediately. It is easy to see that the aggregate packet arrival process is Poisson with rate λ.

Several strategies are studied. The entire call may be allocated to one queue (this is the virtual circuit case), or packets from calls may be assigned to either queue upon arrival (the datagram case). For each case three allocation strategies are studied: probabilistic, alternating and join-the-shortest packet-queue. It is found in every case that, in spite of resequencing delays, datagram networks perform better in the sense of lower mean packet delay. Further JSQ yielded the smallest mean packet delay in both the virtual circuit and datagram cases.

Thus purely from a queueing delay point of view, resequencing delay in the case of stream jobs do not outweigh the gain due to load balancing obtained by splitting the job over several queues.

The question then arises as to the best allocation strategy for tasks in a stream job. Some progress has been made on this problem and is reported in Kuri & Kumar (1991). A single stream job of infinite extent is considered, i.e., tasks arrive in a stream and must depart in sequence. There are N equal-speed servers, each with its own queue. Each task must be allocated to a queue upon arrival, and is served to completion at that queue, in an FCFS manner. For a general arrival process, the special case of *deterministic service times* is considered (i.e. all tasks have the same fixed service time

at any server). Let d'_n denote the departure epoch of the nth arriving task with an arbitrary task allocation policy, and d_n denote the departure epoch of the nth arriving task with the join-the-shortest-work-queue (JSWQ) policy (i.e., an arriving task joins the queue with the least work). Note that d'_n and d_n include resequencing delay. We assume that all servers have zero work at time 0.

Theorem 8. *On every sample path,* $\forall n, d'_n \geqslant d_n$.

Proof. Denoting by c'_n, c_n the service departure epochs of the nth arriving task, by γ'_n, γ_n the nth service departure epochs, by γ''_n the nth service departure epoch in the G/D/N system,

$$d'_n = \max_{1 \leqslant i \leqslant N} c'_n \geqslant \gamma'_n \geqslant \gamma''_n = \gamma_n = c_n = d_n,$$

where the first relation is by definition, the second is obvious, the third follows from Wolff (1977) and the remaining from results in Kuri & Kumar (1991). □

Let $\tilde{d}_n$ denote the departure epoch of the nth arriving task, in the system in which tasks are allocated cyclically, or in a round-robin fashion, to the N queues.

Theorem 9. *On every sample path,* $\forall n, \tilde{d}_n = d_n$, *i.e., departures with cyclic allocation occur at the same epochs as with JSWQ.*

Proof. See Kuri & Kumar (1991). □

The last theorem asserts the optimality of the cyclic rule for deterministic service times. We note that JSQ may be strictly worse in the sense of sample-path-wise comparison.

Our discussion and results thus imply the following more general fact. Suppose tasks are arriving in several streams of finite or infinite extent; the tasks in a stream may all arrive instantaneously (batch arrivals), or in smaller batches, or spread over time. Only the tasks within a stream need to depart in sequence. If the service times of all the tasks are deterministic and identical, then the policy of mixing all the streams into one stream, and allocating tasks in this stream in a cyclic fashion, is sample-path-wise optimal.

5. Final remarks

It is clear from the foregoing that there are several interesting problems in this class of models that deserve further study. The JSQ result is quite general, but, as a practical matter, the scheduler that allocates tasks to queues might not be able to observe the instantaneous queue lengths. It will be interesting to understand the effect of delayed information on the optimality of JSQ. Hajek's (1985) result on the splitting of a renewal process by regular sequences needs to be extended to obtain policies for sojourn time optimal semidynamic routing to multiple queues.

The analysis of task allocation strategies for multitask jobs is still a topic of current research, particularly when the jobs do not have the simple deterministic fork-join

structure with one task being allocated to each queue. The problem of good allocation policies in the multitask case has received very little attention.

References

Agrawala A, Tripathi S 1981 On the optimality of semidynamic routing schemes. *Inf. Process. Lett.* 13: 20–22

Baccelli F, Makowski A M 1990 Synchronization in queuing systems: In *stochastic analysis of computer and communication system* (ed.) H Takagi (Amsterdam: North Holland/Elsevier Science Publishers)

Baccelli F, Makowski A M, Schwartz A 1985 Simple computable bounds and approximations for the fork-join queue. *Proc. Int. Seminar on Computer Networking and Performance Evaluation*

Baccelli F, Massey W A, Towsley D 1989 Acyclic fork-join queuing networks. *J. Assoc. Comput. Mach.* 36: 615–642

Barlow R E, Proschan F 1975 *Statistical theory of reliability and life testing: Probability models* (New York: Holt, Rinehart and Winston)

Bonomi F, Kumar A 1990 Adaptive optimal load balancing in a non-homogeneous multiserver system with a central job scheduler. *IEEE Trans. Comput.* 39: 1232–1250

Chow Y C, Kohler W 1979 Models of dynamic load balancing in a heterogeneous multiple processor system. *IEEE Trans. Comput.* 28: 354–361

de Souza e Silva E, Gerla M 1984 Load balancing in distributed systems, with multiple classes and site constraints. *Performance 84* 17–33

Eager D L, Lazowska E D, Zahorjan J 1986 Adaptive load sharing in homogeneous distributed systems *IEEE Trans. Software Eng.* 12: 662–675

Ephremides A, Varaiya P, Walrand J 1980 A simple dynamic routing problem: *IEEE Trans. Autom. Control.* 25: 690–693

Flatto L, Hahn S 1984 Two parallel queues created by arrivals with two demands I: *SIAM J. Appl. Math.* 44: 1041–1053

Flatto L, McKean H P 1977 Two queues in parallel. *Commun. Pure Appl. Math.* 30: 255–263

Flores C 1990 Diffusion approximations for computer communications networks: In *Stochastic analysis of computer and communication systems* (ed.) H Takagi (Amsterdam: North Holland/Elsevier Science)

Foschini G J 1977 On heavy traffic diffusion analysis and dynamic routing in packet switched networks. In *Computer performance* (eds) K M Chandy, M Reiser (Amsterdam: North Holland)

Foschini G J, Salz J 1978 A basic dynamic routing problem and diffusion. *IEEE Trans. Commun.* 26: 320–327

Gün L, Jean-Marie A 1990 Resequencing in parallel M/G/1 queues with Bernoulli loading. *Oper. Res.* (submitted)

Hajek B 1984 Optimal control of two interacting service stations. *IEEE Trans. Autom. Control.* 29: 491–499

Hajek B 1985 Extremal splitting of point processes. *Math. Oper. Res.* 10: 543–556

Jean-Marie A 1988 Load balancing in a system of two queues with resequencing. *Performance '87* 75–88

Jean-Marie A, Gün L 1990 Asymptotic results for parallel queues with resequencing. *J. Assoc. Comput. Mach.* (submitted)

Kemae T, Krengel U, O'Brien G L 1977 Stochastic inequalities on partially ordered spaces. *Ann. Probab.* 5: 899–912

Kuri J 1990 *Comparative performance evaluation of routing strategies in connection-oriented and connectionless data networks.* M E thesis, Electrical Communication Engineering Department, Indian Institute of Science, Bangalore

Kuri J, Kumar A 1991 On the optimal allocation of customers that must depart in sequence. *Oper. Res. Lett.* (submitted)

Lin W, Kumar P R 1987 Optimal control of a queuing system with two heterogeneous servers. *IEEE Trans. Autom. Control.* 29: 696–703

Mirchandaney R, Towsley D, Stankovic J A 1989 Analysis of the effects of delays on load sharing. *IEEE Trans. Comput.* 38: 1513–1525

Nelson R D, Philips T K 1990 An approximation for the mean response time for shortest queue routing with general interarrival and service times. IBM Research Report, RC-15429 (#68659)

Nelson R, Tantawi A N 1988 Approximate analysis of fork join synchronisation in parallel queues. *IEEE Trans. Comput.* 37: 739–743

Ni L M, Hwang K 1985 Optimal load balancing in a multiple processor system with multiple job classes. *IEEE Trans. Software Eng.* 11: 491–496

Reiman M I 1983 Some diffusion approximation with state space collapse. *Modelling and performance evaluation methodology* (eds) F Baccelli, G Fayolle (Lecture Notes in Control and Inf. Science) Vol. 60. (Berlin: Springer Verlag)

Ross S 1983 *Stochastic processes* (New York: John Wiley)

Shorey R 1990 *Performance analysis and scheduling of stochastic fork-join jobs in a multicomputer system.* M Sc (Eng). thesis, Electrical Communication Engineering Department, Indian Institute of Science, Bangalore

Smith D R, Whitt W 1981 On the efficiency of shared resources in queuing systems. *Bell Syst. Tech. J.* 60: 39–56

Stoyan D 1983 *Comparison methods for queues and other stochastic models* (New York: John Wiley)

Tanenbaum A 1988 *Computer networks*, 2nd edn (Englewood Cliffs, NJ: Prentice Hall)

Tantawi A N, Towsley D 1984 Optimal load balancing in distributed computer systems. Tech. Rep. RC-10346, IBM T J Watson Research Center

Varma S 1987 *Some problems in queuing systems with resequencing*, M S thesis, Electrical Engineering Dept., Univ. of Maryland, College Park, Maryland; Also Tech. Rep. TR-87-192, Systems Res. Center, Univ. of Maryland, College Park.

Walrand J 1988 *An introduction to queuing networks* (Englewood Cliffs, NJ: Prentice Hall)

Weber R R 1978 On the optimal assignment of customers to parallel queues. *J. Appl. Probab.* 15: 406–413

Whitt W 1986 Deciding which queue to join: Some counterexamples. *Oper. Res.* 34: 55–62

Winston W 1977 Optimality of the shortest line discipline. *J. Appl. Probab.* 14: 181–189

Wolff R W 1977 An upper bound for multi-channel queues. *J. Appl. Probab.* 14: 884–888

Wolff R W 1989 *Stochastic modelling and the theory of queues* (Englewood Cliffs, NJ: Prentice Hall)

Yum T S 1981 The design and analysis of a semidynamic deterministic routing rule: *IEEE Trans. Commun.* 29: 498–504

Yum T S, Schwartz M 1981 The join-biased queue rule and its application to routing in computer communication networks. *IEEE Trans. Commun.* 29: 505–511