Scheduling Flows with Unknown Sizes: Approximate Analysis*

Extended Abstract †

Liang Guo Ibrahim Matta
Computer Science Department
Boston University
{guol, matta}@cs.bu.edu

1. INTRODUCTION

Previous job scheduling studies indicate that providing rapid response to interactive jobs which place frequent but small demands, can reduce the overall system average response time [1], especially when the job size distribution is skewed (see [2] and references therein). Since the distribution of Internet flows is skewed, it is natural to design a network system that favors short file transfers through service differentiation. However, to maintain system scalability, detailed per-flow state such as flow length is generally not available inside the network. As a result, we usually resort to a threshold-based heuristic to identify and give preference to short flows. Specifically, packets from a new flow are always given the highest priority. However, the priority is reduced once the flow has transferred a certain amount of packets.

In this paper, we use the MultiLevel (ML) feedback queue [3] to characterize this discriminatory system. However, the solution given in [3] is in the form of an integral equation, and to date the equation has been solved only for job size distribution that has the form of mixed exponential functions. We adopt an alternative approach, namely using a conservation law by Kleinrock [1], to solve for the average response time in such system. To that end, we approximate the average response time of jobs by a linear function in the job size and solve for the stretch (service slowdown) factors. We show by simulation that such approximation works well for job (flow) size distributions that possess the heavy-tailed property [2], although it does not work so well for exponential distributions.

Due to the limited space available, in Section 2 we briefly describe the queueing model and summarize our approximation approach to solving for the average response time of the M/G/1/ML queueing system. We conclude our paper in Section 3.

2. THE M/G/1/ML QUEUEING SYSTEM

We use the MultiLevel (ML) processor sharing queueing model to describe a discriminatory service system made of a finite number of classes, without knowledge on input job sizes. In such a system, the priority of a job depends on the amount of service already received by the job. For a set of predefined thresholds $0 = b_0 < b_1 < ... < b_M = \infty$, once the job has received more than b_{i-1} units of service, its priority is reduced to i. Jobs of the same class are served by a processor-sharing scheduler, while jobs with different priorities can be served by a priority queueing (PRIO) algorithm or by a (weighted) Discriminatory Processor Sharing (DPS) algorithm. With the DPS scheduling algorithm, if there are N_i jobs of class i present, $i = 1, 2, ..., M$, then the service rate received by class-i jobs depends on class-i weight g_i and is given by:

$$r_i = \frac{g_i N_i}{\sum_{j=1}^{M} g_j N_j} C \qquad i = 1, 2, ..., M$$

C denotes the total service rate. We refer to the first system as the ML-PRIO queue, and the second as the ML-DPS queue. In the special case of 2 job classes ($M = 2$) and $g_1 = \infty$, the two systems are equivalent. We assume jobs arrive according to a Poisson process with total rate λ, and the service distribution is arbitrary. Thus, the queueing system is referred to as M/G/1/ML-SR system, where SR can be either PRIO or DPS.

Our goal is to solve for the *average (expected) response time* for jobs which require x total service, denoted by $T(x)$. For convenience, we define $T_i(x_i)$ as the expected time to serve x_i units of size while the job is at class i. Thus, we have, for $b_{i-1} < x \leq b_i$,

$$T(x) = \sum_{j=1}^{i-1} T_j(b_j - b_{j-1}) + T_i(x - b_{i-1})$$

To date, we are only able to obtain the exact solution to $T(x)$ for the special case of $M = 2$ in the form of integral equations and a closed-form solution may only exist for specific distributions. Approximation is needed to attack more general cases.

2.1 Approximation Approach to Solve for $T(x)$

The main idea of our analysis is to apply the *Conservation Law for Work-conserving Time-Shared Systems*, proposed and proven by Kleinrock ([1], pages 197-199). Formally, it states the following:

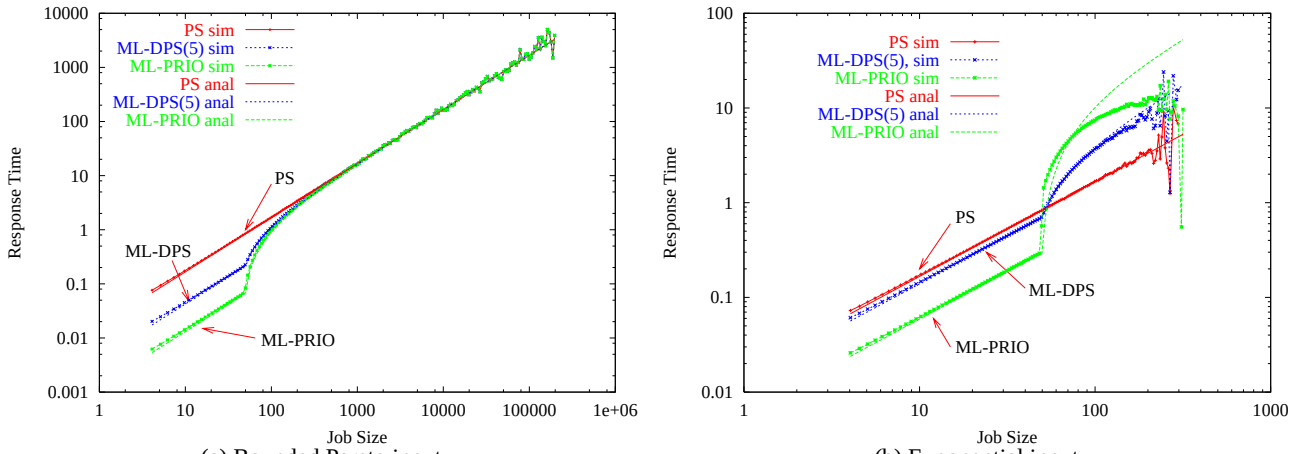
Theorem 1. **Kleinrock's Conservation Law for Time-Shared Systems**. *For any M/G/1 system and any work-conserving queueing discipline, the average response time $T(x)$ for jobs of length x, satisfies the following equation:*

$$\int_{0^-}^{\infty} T(x)[1 - B(x)]dx = \frac{\overline{X^2}}{2(1-\rho)}$$

where ρ, $B(x)$ and $\overline{X^2}$ represent the average load of the system, the cumulative distribution of job sizes and the second moment of the job size distribution, respectively.

*This work was supported in part by NSF grants CAREER ANI-0096045 and ANI-0095988.

†A full version of this paper is available as Technical Report BUCS TR-2002-009 at `http://www.cs.bu.edu/techreports`



(a) Bounded Pareto input (b) Exponential input

Figure 1: Performance Comparison of PS, ML-PRIO and ML-DPS

Therefore, if $T(x)$ has only one free variable, we can utilize the Conservation Law to solve for the closed-form of $T(x)$, given the job size distribution and average system load.

We next approximate the multi-class M/G/1/ML-DPS system by multiple loosely coupled M/G/1 processor sharing systems so that we can represent each $T_i(x_i)$ by a linear function in x_i, i.e. $T_i(x_i) = \theta_i x_i$, where θ_i's are the so-called stretch factors. The detailed solution is given in the full version of this paper. The set of equations have degree of freedom of M (θ_i's) and we have M independent linear equations, we can thus solve for $T(x)$.

2.2 Validation by Simulation

Notice that the solution above applies to general distributions which have finite first and second moments. We now study the accuracy of our analysis for different job size distributions. As an example, we show here the cases where job sizes follow the Bounded Pareto distribution $B_{BP}(x, \alpha, k, p)$ and the generalized Exponential distribution $B_{EXP}(x, \mu, k)$ [4]. The Bounded Pareto distributions have finite first and second moments, but they do possess the *Heavy Tailed (HT) property*, as defined in [2]. On the contrary, the generalized Exponential distribution does not have such property since the probability of having large jobs is very small.

We study the case of a two-level system and the cutoff size is set to $b_1 = 50$ (about two times the average size). For the ML-DPS system, we set the weight factor to be $\mathbf{g} = (5, 1)$, i.e., each class-1 job gets 5 times unit of service as each class-2 job. We also let $g_1 = \infty$ to obtain results for the ML-PRIO scheme. Figures 1(a) and 1(b) show the simulation as well as analytical results for cases in which job sizes follow $B_{BP}(x, 1.2, 4, 200000)$ and $B_{EXP}(x, 17.243, 4)$, respectively. We assume $C = 2000$ and $\lambda = 91.32$. Thus, the total load on the system is approximately 0.97. In the figures, PS denotes the nominal processor sharing scheme (where g_i's are all equal to 1).

3. CONCLUSION

• For bounded Pareto input, the response time function at the second-level queue (i.e., job sizes greater than 50) can be well-approximated by a linear function, thus our analysis gives very accurate prediction. On the contrary, our analysis is not accurate for Exponential input. The actual response function further penalizes jobs whose sizes are just above the cutoff threshold. We also notice that our analysis still gives relatively good approximation under ML-DPS when the relative weight of the low priority jobs is not too small.

• When the job size distribution has the HT property, size-aware scheduling significantly reduces the response time of small jobs but only slightly increases the response time of long jobs. The ML-PRIO scheme, which gives absolute priority to short jobs, can reduce small job response time by a factor of 15, while only increase long job response time by a factor of 1.008. The ML-DPS scheme performs between ML-PRIO and the PS scheme.

• In case of exponential distributions which have very light tails, the benefit of giving preferential treatment to short jobs (e.g. in ML-PRIO, a factor of 4.02) is achieved at the expense of significantly sacrificing long jobs' performance (by a factor of 8.29 in ML-PRIO). Moreover, although a very large weight (a factor of 5) is given to short jobs, the overall performance enhancement by employing the DPS scheduling algorithm is limited (a factor of 1.323). This is vastly different from what we observed in the previous scenario where file sizes possess the HT property, in which case the same DPS scheduling algorithm can reduce short job response times by a factor of 3.25.

We now give an intuitive explanation of why a linear response time function is a good approximation for job size distributions with a HT property but not for those without. We notice that the asymptotic behavior of $T(x)$ as $x \to \infty$ is always $T(x) \sim \frac{x}{1-\rho}$ [5]. With our analysis, when the job size distribution has the HT property, the worst case penalty factor for large jobs (generating most of the load of the system) is already very close to $\frac{1}{1-\rho}$, thus the approximation is good.

4. REFERENCES

[1] L. Kleinrock, *Queueing Systems, Volume II: Computer Applications*, ISBN 0-471-49111-1. John Wiley & Sons Inc., 1976.

[2] N. Bansal and M. Harchol-Balter, "Analysis of SRPT Scheduling: Investigating Unfairness," in *Proc. of ACM SIGMETRICS 2001*, Boston, MA., June 2001.

[3] L. Kleinrock and R.R. Muntz, "Processor Sharing Queueing Models of Mixed Scheduling Disciplines for Time Shared Systems," *Journal of the ACM*, vol. 19(3), pp. 464–482, July 1972.

[4] N.L. Johnson and S. Kotz, *Continuous Univariate Distributions-1*, Houghton Mifflin Co., Boston, 1970.

[5] G. Fayolle, I. Mitrani, and R.Iasnogorodski, "Sharing a Processor among Many Job Classes," *Journal of the ACM*, vol. 27(3), pp. 519–532, July 1980.