The Impact of Quanta on the Performance of Multi-level Time Sharing Policy under Heavy-tailed Workloads

Malith Jayasinghe[1] **Zahir Tari**[1] **Panlop Zeephongsekul**[2]

[1] School of Computer Science and Information Technology
RMIT University
Melbourne, Australia
Email: `mjayasin@cs.rmit.edu.au, zahir.tari@rmit.edu.au`

[2] School of Mathematical and Geospatial Sciences
Melbourne, Australia
RMIT University
Email: `panlopz@rmit.edu.au`

Abstract

Recent research indicates that modern computer workloads (e.g. processing time of web requests) follow heavy-tailed distributions. In a heavy-tailed distribution there are a large number of small tasks and a small number of large tasks. The rationale for using a multi-level time sharing policy is that it can minimise both waiting time and slowdown of tasks that require relatively small service requirements. This in turn will improve the overall performance of the system. Using a 2-level system (policy), we investigate the effect of quanta on the overall performance of a multi-level time sharing policy under a range of workloads and task size variabilities. We measure the performance using slowdown and flow time. First, we show that for most workloads and task size variabilities there exists a unique set of quanta ('optimal' set of quanta) that would result in the best performance. Second, we investigate the performance degradation in one metric under the optimal parameters of other metric. Through an extensive numerical analysis, we find that under high system loads and task size variabilities using the optimal set of quanta corresponding to overall expected slowdown can result in the overall expected flow time to deteriorate significantly. Finally we show that a 3-level system with the optimal set of quanta outperforms a 2-level system with the optimal set of quanta for all the scenarios considered.

Keywords: Heavy-tailed property, Overall expected flow time, Overall expected slowdown

1 Introduction

Given that modern computer workloads have service requirements that are best characterised by heavy-tailed distributions (Arlitt & Jin 1999, Arlitt & Williamson 1996, Barford et al. 1999), there is an urgent need to investigate the performance of many traditional scheduling policies under heavy-tailed workloads. Recent research shows that many traditional policies that perform well under distributions such as exponential distributions do not perform well when the service time distribution has the heavy-tailed property (Crovella, Harchol-Balter & Murta 1998).

The rationale for using a multi-level time sharing policy is that it can minimise both the waiting time and slowdown of jobs that require relatively small processing requirements. In a heavy-tailed distribution, as there are a large number of such small jobs, by improving the performance of these small jobs improve the overall performance also. Moreover, research indicates (Righter & Shanthi Kumar 1990, Yashkov 1987) that when the service time distribution of tasks have a decreasing failure rate (the longer a task has run, the longer it is expected to run) multi-level level time sharing policies can result in significant performance improvements over other policies. One key property of a heavy-tailed distribution is that it has a decreasing failure rate.

There is a significant body of work that investigates the performance of various scheduling policies under heavy-tailed workloads (Harchol-Balter 2000, Broberg et al. 2006, Psounis et al. 2005). Most of these assume that the jobs are served in a First Come First Serve (FCFS) manner at the server (Harchol-Balter 2000, Broberg et al. 2006, Psounis et al. 2005). Many modern computer systems, however, do not use FCFS but use pre-emptive policies such as round-robin, multi-level time sharing and variants of these two policies.

The analysis of time sharing policies (pre-emptive policies) are relatively complex compared to that of FCFS because the analysis needs to take into account the partially completed jobs that are scattered on various levels throughout the system. Many existing papers focusing on time sharing policies assume a Poisson arrival pattern and an exponential service time distribution. Given that many modern computer workloads have the heavy-tailed property (Arlitt & Jin 1999, Arlitt & Williamson 1996, Barford et al. 1999), the models (policies) that assume exponential service time distributions and Poisson arrival patterns are of limited capability.

Although there exist many variants of time sharing policies, these can be divided into four broad categories namely: processor sharing, round-robin, multi-level processor sharing and multi-level time sharing. Processor sharing can be considered as the limiting case of round-robin and its analysis typically assumes infinitely small quantum whereas multi-level processor sharing can be considered as the limiting case of multi-level time sharing and it assumes infinitely small quanta and infinite number of levels.

As far as multi-level time sharing and multi-level processor disciplines are concerned much of early research on these were done by Schrage (1967) and Coffman & Kleinrock (1968). In Coffman & Kleinrock (1968), they obtained the expected waiting time,

Copyright ©2009, Australian Computer Society, Inc. This paper appeared at the 32nd Australasian Computer Science Conference (ACSC 2009), Wellington, New Zealand, January 2009. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 91, Bernard Mans, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

based on the length of the service for both M/M/1 multi-level time sharing and M/M/1 multi-level processor sharing disciplines. In this paper, they investigated the impact of the length of quantum on the conditional expected waiting time and showed that the plot of conditional expected waiting time vs quantum size has a shape of a saw tooth. In Schrage (1967), Schrage derived the expected time in the system, based on the length of the service of a multi-level time sharing policy under an arbitrary service time distribution when the arrival process is Poisson.

Recent work on multi-level processor sharing discipline includes Aalto et al. (2007, 2005, 2004). In Aalto et al. (2005), the mean delay of a multi-level processor sharing policy was investigated under a more general class of service distributions called increasing residual lifetime (IMRL). In Aalto et al. (2007), the service differentiation capabilities of time sharing policies are studied.

Systems that use a multi-level time sharing policy include web servers, operating systems and routers. A multi-level time sharing policy with higher number of levels is more suitable for systems that have relatively large memory capacities and low context switch overhead time. Systems with small memory can also benefit from it if certain constraints are met. Such constraints will include the number of levels and average arrival rate into the system. As the number of levels increase, the amount of memory required to store the information about partially completed jobs will increase. Therefore, prior to increasing the number of levels, system designers need to ensure that there is sufficient memory to store all the partially completed states. Such consideration is especially important for embedded systems that have tight memory constraints.

Although there is evidence showing that a multi-level time sharing policy can result in improved performance under heavy-tailed workloads (Yashkov 1987, Nuyens & Wierman 2008), prior work does not investigate the effect of quanta (amount of service allocated to a task on various levels) on the overall performance because much prior work assumes infinitely small quanta.

In this paper, we investigate the impact of quanta on the overall performance of a multi-level time sharing policy for a range of system loads and task size variabilities. We investigate the performance using two metrics: the overall expected slowdown and overall expected flow time. Slowdown is defined as the ratio between waiting time and processing time of a task and it is measures the fairness of a given scheduling policy. Flow time, on the other hand, measures the total time that a task spends in the system and it includes both the waiting time and processing time.

Using a 2-level system, we show that for a given system load and task size variability, there exists a unique set of quanta that would produce the minimal overall expected slowdown. The set of quanta that would produce the minimal overall expected flow time, however, may not be unique for a few workloads and task size variabilities. Moreover, we find that there is a sudden drop in 1^{st} optimal quantum (and an increase in 2^{nd} 'optimal' quantum) that occurs between the system loads of 0.5 and 0.7 when the performance is measured using the overall expected flow time. Such a drop is not observed when the performance is evaluated using the overall expected slowdown.

Second, we investigate the fraction of tasks completed at levels under a range of task size variabilities and system loads. Using a 2-level system, we show that the fractions of tasks completed at levels do not vary much with the variability of tasks under a constant system load. This behaviour is particularly evident when the policy's quanta are computed to optimise the overall expected slowdown. This means that under a given system load, once the optimal set of quanta are computed for one task size variability, the optimal set of quanta of other variabilities can be computed using the cumulative distribution function of the service time distribution.

Third, we investigate performance degradation in one performance measure under the optimal parameters (i.e. optimal set of quanta) of the other performance measure. We find that under high system loads and task size variabilities the use of the optimal set of quanta corresponding to overall expected slowdown can result in a significant deterioration (250%) of the overall expected flow time. However, performance degradation in the overall expected slowdown is less when the optimal set of quanta corresponding to overall expected flow time are used. Finally, we briefly investigate the behaviour of quanta for the policy consisting of 3-levels. We show that a 3-level system with the optimal set of quanta outperforms a 2-level system with the optimal set of quanta for all the system loads and task size variablities.

The rest of this paper is organised as follows. In Section 2 we briefly discuss heavy-tailed distributions and the Poisson process and obtain the key performance metrics. In Section 3 we transform the quantum-based policy into a cutoff point based policy. Section 4 investigates the effect of quanta on the overall expected flow time and in Section 5 we investigate the effect of quanta on the overall expected slowdown. In Section 6 we study the fraction of tasks completed at levels. Section 7 investigates performance degradation in one performance measure under the optimal parameters of other performance measure. In Section 8 we briefly investigate the behaviour of quanta under 3-levels. Section 9 concludes the paper.

2 Background

In this section we will present the background that is needed to understand the rest of this paper. In Section 2.1 we discuss key properties of heavy-tailed distributions. In Section 2.2 we present the quantum-based multi-level time sharing model and present the two performance metrics that we use to evaluate its performance.

2.1 Heavy-tailed workloads and Poisson process

A random variable X is said to be heavy-tailed

$$P(X > x) \sim x^{-\alpha} \quad 0 < \alpha < 2 \tag{1}$$

In Equation 1, α represents the variability of tasks in the service time distribution. The lower the value of α the higher the variability of tasks. As α increases, the tail of the distribution becomes thinner (in area) indicating that the variability of the distribution is decreasing (see fig 1). In the case of file sizes stored on servers, α lies in the range 1.1 to 1.3 (Crovella & Bestavros 1997, Crovella, Taqqu & Bestavros 1998). One key property of a heavy-tailed distribution is that its variance is infinite, because of this, for modelling purposes, heavy-tailed distributions are typically represented by a Bounded Pareto Distribution (Harchol-Balter 2000) that has an upper bound (p) and lower bound (k). Another important property of a heavy-tailed distribution is that it has a decreasing failure rate. This means that the longer a task has run the longer it is expected to run.

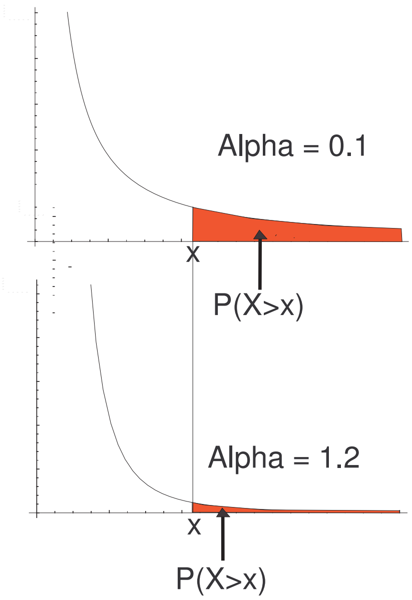

Figure 1: The effect of α on the tail

The probability distribution function of Bounded Pareto Distribution is given by;

$$f(x) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}}\ x^{-\alpha-1},\ \ k < x < p \qquad (2)$$

The moments of Bounded Pareto Distribution are finite for all α values and it is easy to show that they are given by;

$$E[X^j] = \begin{cases} \frac{\alpha k^{\alpha}(k^{j-\alpha}-p^{j-\alpha})}{(\alpha-j)(1-(k/p)^{\alpha})} & \text{if } \alpha \neq j, \\ \frac{k}{(1-(\frac{k}{p}))}(\ln p - \ln k) & \text{if } \alpha = j \end{cases} \qquad (3)$$

The expected value of the Bounded Pareto Distribution is obtained by substituting $j = 1$ into Equation 3. In this paper, we fix the expected value (average) of the Bounded Pareto distribution at 3000 ms. It has been shown that the average size of a static web page is 3000 bytes (Crovella & Bestavros 1997). In the case of static web requests the time it takes to serve a web page is proportional to the size of the page (in bytes).

The second important parameter in a queueing system (scheduling policy) is the arrival process. In this paper, we assume that the task arrive according to Poisson Process. A Poisson process is a stochastic process (random variables indexed by time) in which the probability of more than one arrival at a given instance is equal to 0. When the tasks arrive according to a Poisson process, the number of tasks that arrive in two consecutive periods of time is independent of each other (independent increments). Moreover, when tasks arrive according to a Poisson process, it can be shown that the inter-arrival times follow an exponential distribution with the mean of $\frac{1}{\lambda}$ (λ is rate of the Poisson process).

2.2 Multi-level time sharing model

Figure 2 illustrates the time sharing model that we consider in this paper. Figure 3 is a different representation of the same model. Newly arriving tasks are first placed in the lowest level queue which has the highest priority. At each level, tasks are served in a first come first serve (FCFS) fashion. A task waiting in a particular queue (level) will only be served if there are no tasks waiting in the lower level queues. We denote the maximum amount of service allocated to a task on i^{th} level by q_i (i^{th} quantum). A task on i^{th} level can receive up to q_i service and if its service requirement is greater than q_i it will be preempted from the server and placed in the next lower level queue. If its service requirement is less than q_i, the task departs the system from i^{th} level. We assume that the process switching time (i.e. context switch time) is negligible and could be equated to zero. In the case where there is significant process switching time, the model can be easily modified to cater for process switching time.

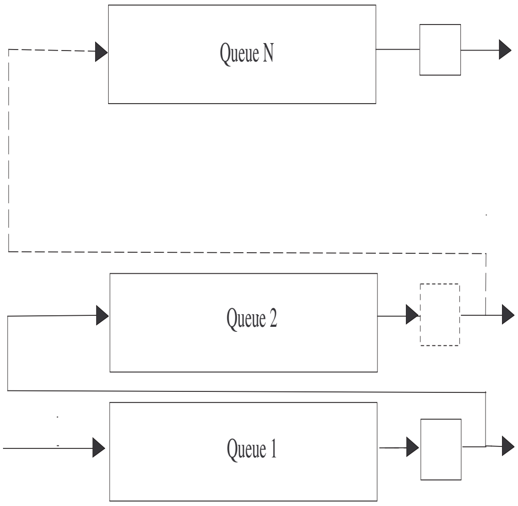

Figure 2: Multi-level time sharing model

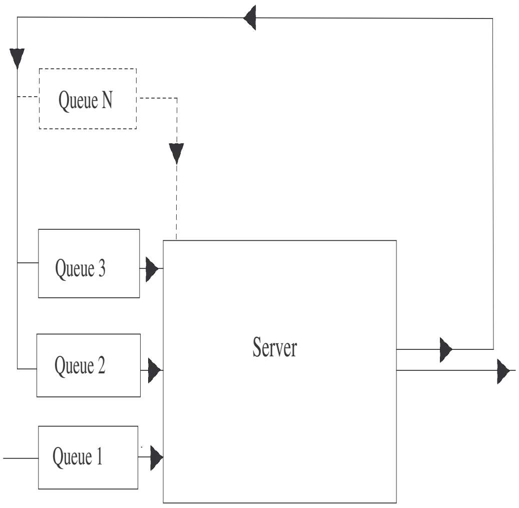

Figure 3: Multi-level time sharing model

As discussed, we represent the maximum amount of service allocated to a task on i^{th} level (i.e. i^{th} quantum) by q_i and the sum these service times up to and including i^{th} service time by Q_i.

$$Q_i = q_1 + q_2 + + + q_i \qquad (4)$$

We defined the overall performance based on the conditional expected waiting time derived in Schrage (1967). We use the following to notation represent a multi-level time sharing policy. (This notation is same as the notation used in Schrage (1967))

1. λ = arrival rate into 1^{st} queue

2. $T_i = i^{th}$ simple processing; the length of processing time which a task on i^{th} queue receives
3. $F(t)$ = Cumulative distribution function (CDF) of service time distribution
4. $F_i(t) = P[T_i \leq t]$ CDF of i^{th} simple processing time
5. $S_i = T_1 + T_2 + ... + T_i$ given that the job returns to the system at least i times
6. $U_i = S_i$ if the job returns to the system at least $i-1$ times
$U_i = T_1 + T_2 + ... + T_k$ if the job returns the system only $k-1$ times, $k < i$
7. N = number of queues (levels)
8. $\Lambda_k = \lambda(1 - \int_0^{S_i} dF(t))$
9. FT_i = time in the system up to and including i^{th} simple processing time
10. W_i = waiting time in the system up to i^{th} level. This time does not include i simple processing times (i.e. $T_1, T2, , , T_i$)

The expected flow time given that the service time of a task is higher than Q_{i-1} and less than Q_i is given by;

$$E[FT_i] = \frac{\lambda E[U_i^2] + \sum_{k=i+1}^{N} \Lambda_k E[T_k^2]}{2(1-\lambda E[U_{i-1}])(1-\lambda E[U_i])} + \frac{Q_{i-1}}{(1-\lambda E[U_{i-1}])} + E[T_i] \tag{5}$$

The above equation is obtained by representing the total delay experienced by a random arrival to the system as the sum of independent delay components. The reader may refer to Schrage (1967) or Jayasinghe et al. (2008) for further details.

The expected waiting time given that the service time of a task is higher than Q_{i-1} and less than Q_i is given by;

$$E[W_i] = E[FT_i] - Q_{i-1} - E[T_i] \tag{6}$$

The expected slowdown given that the service time of a task is higher than Q_{i-1} and less than Q_i is given by;

$$E[SD_i] = E[W_i]\ E\left[\frac{1}{Q_{i-1} + T_i}\right] \tag{7}$$

$E[U_i]$, $E[U_{i-1}]$, $E[U_i^2]$, $E[T_k^2]$, $E[T_k]$, $E[T_k^2]$, $E[T_k^{-1}]$ and Λ_k are obtained for a Bounded Pareto service time distribution. For those readers who are interested in the derivation of these formulae, they may refer to the technical report Jayasinghe et al. (2008).

We multiply each $E[FT_i]$ by the probability that service requirement is within the interval $[Q_{i-1}, Q_i], i = 1, 2, , , , , N(Q_0 = 0)$ and then summing over all i gives the overall expected flow time;

$$E[FT]_{overall} = E[FT_1] \int_0^{Q_1} f(x)dx + E[FT_2] \int_{Q_1}^{Q_2} f(x)dx + ... + E[FT_N] \int_{Q_{N-1}}^{Q_N} f(x)dx \tag{8}$$

Similarly, we obtain the overall expected slowdown;

$$E[SD]_{overall} = E[SD_1] \int_0^{Q_1} f(x)dx + E[SD_2] \int_{Q_1}^{Q_2} f(x)dx + ... + E[SD_N] \int_{Q_{N-1}}^{Q_N} f(x)dx \tag{9}$$

3 Transformation of quanta into cut-offs

We now transform the quantum based multi-level time sharing system discussed in Section 2.2 into a cut-offs (cut-off points) based multi-level time sharing system by partitioning the domain $[0, p]$ of a Pareto's distribution into a series of cutoff points $p_1, p_2, ..., p_N$.

$$\begin{aligned} q_N &= p - p_{N-1} \\ q_1 &= p_1 \\ q_i &= p_i - p_{i-1}; 1 < i < N \end{aligned} \tag{10}$$

We assume that the upper bound, p, of the Bounded Pareto Distribution is 10^7 (Broberg et al. 2006). The high value of p will ensure that the Bounded Pareto distribution will represent a realistic heavy-tailed workload (Broberg et al. 2006). The relationship between the quanta and cut-offs are given by;

$$\begin{aligned} q_N &= 10^7 - p_{N-1} \\ q_1 &= p_1 \\ q_i &= p_i - p_{i-1}; 1 < i < N \end{aligned} \tag{11}$$

We can now represent N quanta in an N-level multi-level time sharing model using N-1 cut-offs. For example, a 3-level policy will now have 2 cut-offs as opposed to 3 quanta. For a 3-level policy, the relationship between cut-offs and quanta are given by;

$$\begin{aligned} q_1 &= p_1 \\ q_2 &= p_2 - p_1 \\ q_3 &= 10^7 - p_2 \end{aligned} \tag{12}$$

4 The impact of p_1 on overall expected flow time for the case of 2 levels

In this section, we consider the effect of p_1 on the overall expected flow time. Both $p1$ and overall expected flow time have the same units. For example, if the unit of $p1$ is milliseconds so is the unit of overall expected flow time. We represent the cut-off (cut-off point), p_i, corresponding to the minimal overall expected slowdown and flow time using $p_{i\_sd\_opt}$ and $p_{i\_ft\_opt}$ respectively. We see from Figure 4, the value of $p_{1\_ft\_opt}$ is higher for low (0.3) and moderate (0.5) system loads. For high system loads, (i.e. 0.7 and 0.9) the value of $p_{1\_ft\_opt}$ is relatively small. For the range of workloads (i.e. 0.3, 0.5, 0.7 and 0.9) and task size variabilities (0.4 - 1.95) considered, $p_{1\_ft\_opt}$ is unique for a given system load and task size variability.

Between the system loads of 0.5 and 0.7, the plot of overall expected flow time vs p_1 consists of two minima whose performance are not significantly different from each other. In such cases, the policy designer may use p_1 corresponding to either minima as they both would result in the similar performance. Furthermore, within this range it is possible for $p_{1\_ft\_opt}$ to be not unique.

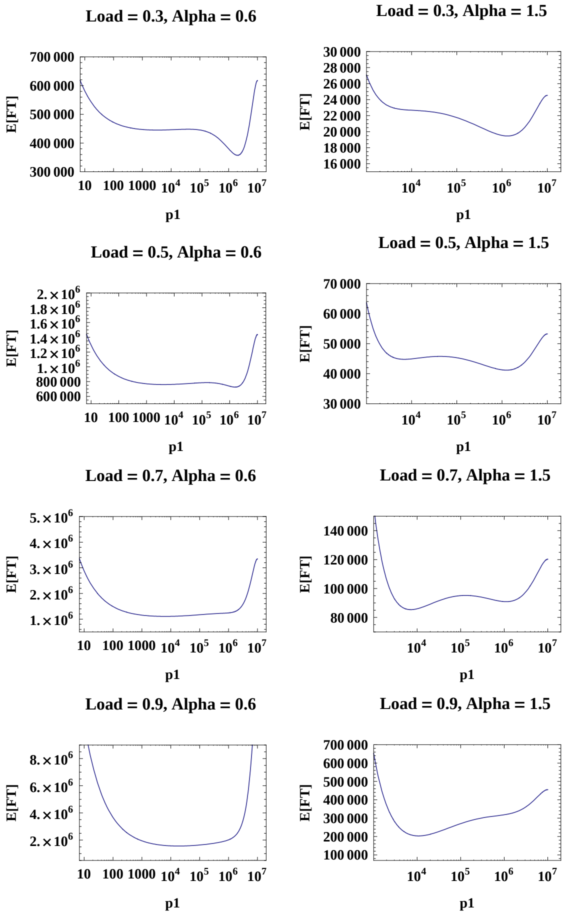

Figure 4: Impact of p_1 on overall expected flow time (N=2)

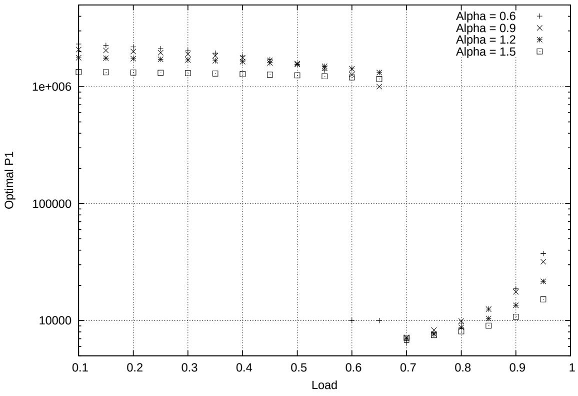

Figure 5: Behaviour of $p1_{opt\_ft}$ with system load (N=2)

In the case where there are two such minima, p_1 corresponding to the minimum on the left is highly sensitive to the overall expected flow time. Therefore, if the system designer cannot estimate p_1 accurately it is recommended that the higher value of p_1 be used (corresponds the minimum on the right). This will minimise the performance degradation due to slightly overestimating or underestimating of $p_{1\_opt\_ft}$.

The plot of p_1 vs system load (Figure 5) indicates that there is a sudden drop in $p_{1\_ft\_opt}$. Notice from Figure 5 that this sudden drop occurs between the system loads of 0.6 and 0.7. This further justifies our earlier observation of high p_1 for low and moderate system loads and low p_1 for high workloads.

Figure 6 illustrates the effect of α on $p_{1\_opt\_ft}$. We see from Figure 6, that under low and moderate workloads as α increases $p_{1\_ft\_opt}$ decreases. In the case of high system workloads, there are no such clear patterns in the variation in $p_{1\_ft\_opt}$ with α.

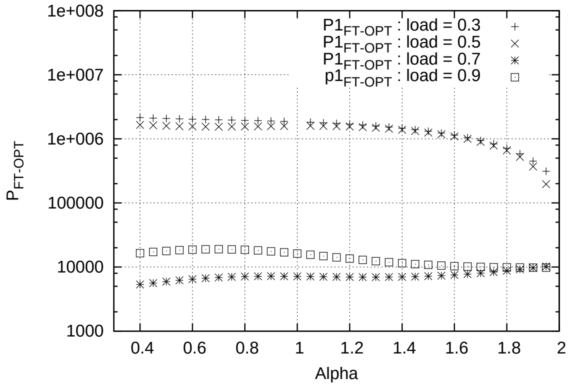

Figure 6: Behaviour of $p_{1\_ft\_opt}$ with α (N=2)

5 The impact of p_1 on the overall expected slowdown

Figure 7 illustrates behaviour of the overall expected slowdown with α. We can clearly see that for a given system load and task size variability the plots of overall expected slowdown vs p_1 always have a unique global minimum. Therefore, $p_{1\_sd\_opt}$ is unique for a given system load and task size variability.

Moreover, we see (see Figure 8) that $p_{1\_sd\_opt}$ is very small compared to the largest task, p (=10^7), in the service time distribution. Under a constant system load an increase in α will result in $p_{1\_sd\_opt}$ increasing (see Figure 8).

6 Fraction of tasks completed at levels

In this section, we investigate the fraction of tasks completed at levels using a 2-level policy. We compute the fraction of tasks completed at levels using the cumulative distribution function of the Bounded Pareto Distribution. The cumulative distribution function of Bounded Pareto Distribution is given by;

$$\begin{aligned} F(x) &= \int_k^x \frac{\alpha k^\alpha}{1-(k/p)^\alpha} \;\; x^{-\alpha-1}, \;\; k < x < p \\ &= -\frac{k^\alpha}{1-(k/p)^\alpha}(x^{-\alpha} - k^{-\alpha}) \end{aligned} \tag{13}$$

6.1 Overall expected flow time

We compute the fraction of tasks completed at 1^{st} and 2^{nd} levels as follows.

$$\begin{aligned} Frac\_L1\_ft &= F(p_{1\_ft\_opt}) \\ Frac\_L2\_ft &= 1 - F(p_{1\_ft\_opt}) \end{aligned} \tag{14}$$

We can see from Figure 9 that more than 95% of tasks are completed at the first level for the range

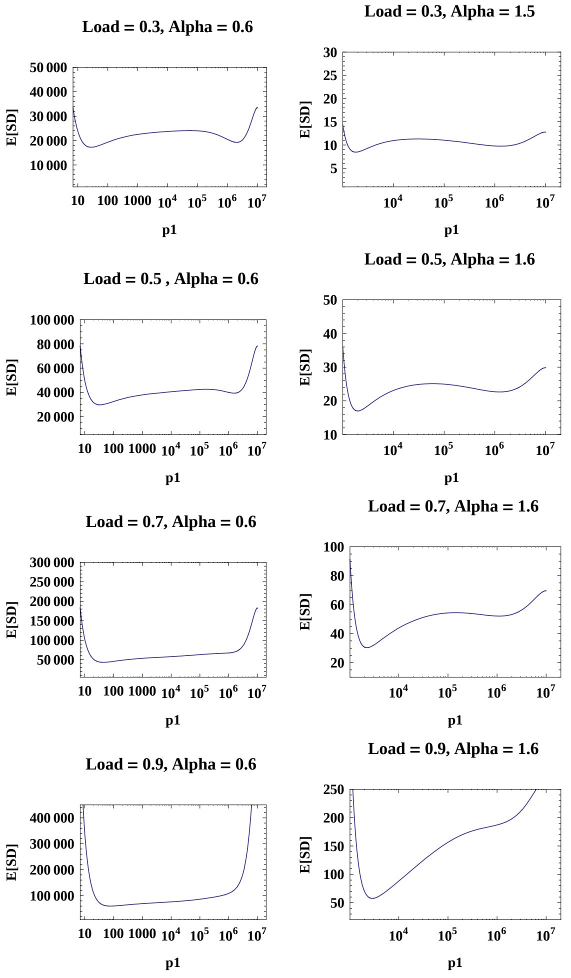

Figure 7: Impact of p_1 on overall expected slowdown (N=2)

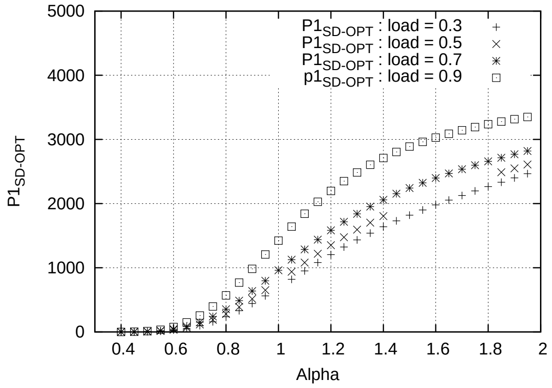

Figure 8: Behaviour of $p_{1\_opt\_sd}$ with α (N=2)

of workloads considered. Under low and moderate system loads, the fraction of tasks completed at level 1 is very high (0.999). Previously, we noticed high $p_{1\_ft\_opt}$ under low and moderate workloads indicating a large fraction of jobs being completed at 1^{st} level. As the system load increases, the fraction of tasks completed at 1^{st} level decreases by a small amount (0.05). As the system load increases, the policy improves the overall expected flow time by increasing the degree of preferential treatment given to small tasks. Earlier, we saw (see Figure 4) low $p_{1\_ft\_opt}$ under high system workloads (0.7, 0.9).

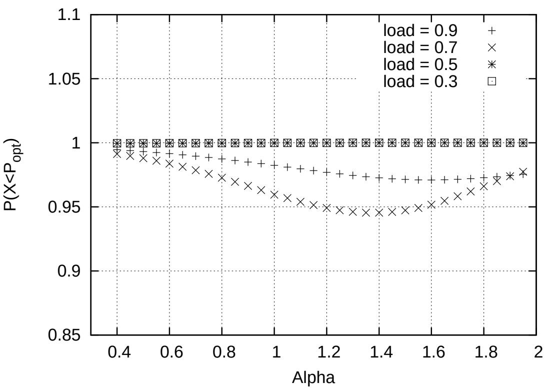

Figure 9: $Frac\_L1\_ft$ (N=2)

6.2 Overall expected slow down

In the case of overall expected slowdown, the fraction of tasks completed at 1^{st} and 2^{nd} levels are given by

$$\begin{aligned} Frac\_L1\_sd &= F(p_{1\_opt\_sd}) \\ Frac\_L2\_sd &= 1 - F(p_{1\_opt\_sd}) \end{aligned} \tag{15}$$

Here we see that $Frac\_L1\_sd$ are not as high as $Frac\_L1\_ft$ for the range of workloads considered. The highest fraction of tasks completed is about 80%. Under low and moderate system loads the fraction of tasks completed at level one is less than 70%. We see that as the system load increases, the fraction of tasks completed at 1^{st} level increases (different from what we saw before in Fig 9). Moreover, we notice that under a constant system load when α lies in the range 0.8 and 1.6 the fraction of task completed at levels stagnates.

This means that under a fixed system load once $p_{1\_opt\_sd}$ is computed for one α value using an optimisation program (routine), $p_{1\_opt\_sd}$ for other α values can be computed simply by substituting $F(p_{1\_opt\_sd})$, α, p and k values into the inverse cumulative distribution function of the Bounded Pareto distribution. When designing systems that utilise adaptive (optimal cut-offs are computed on the fly) multi-level time sharing policies, such a method can be very useful as it will allow optimal cut-off to be computed in less time. Moreover, systems that do not have adequate computational resources to solve complex optimisation problems can also benefit.

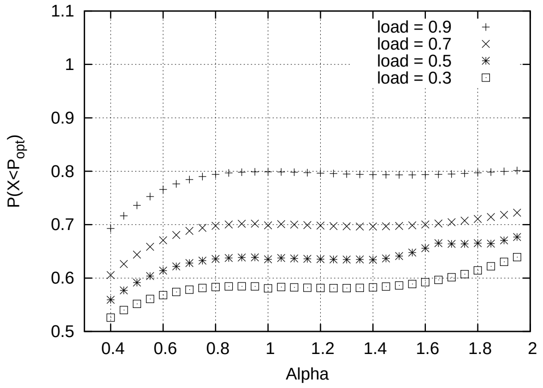

Figure 10: $Frac\_L1\_sd$ (N=2)

7 Performance degradation in one metric under the optimal parameters of other metric

Several scheduling policies have been developed over the years that can distribute heavy-tailed workloads efficiently. Most of these policies have been proposed for distributed web server farms (Harchol-Balter 2000, Harchol-Balter et al. 1999, Tari et al. 2005, Broberg et al. 2006). The performance of these policies are typically evaluated using expected slowdown, flow time or waiting time. In such evaluations, parameters of the scheduling policy (e.g. quanta, cut-offs and system load) are computed so that a given metric is optimised.

One common issue with existing work is that the authors do not investigate the performance degradation in one metric under the optimal parameters of other metric. The purpose of this section is to investigate this problem using a multi-level time sharing system. We investigate the performance degradation in overall expected slowdown under the optimal cut-offs of overall expected flow time and vice versa. We present our results for a range of task variabilities (α values) and system loads. For each system load and α, we compute the value of the metric by substituting the optimal cut-offs of other metric.

We define the percentage performance degradation in the overall expected flow time as follows.

$$E[FT]_{Deg}\% = \frac{E[FT]_{sd\_cutoff} - E[FT]_{optimal}}{E[FT]_{optimal}}\% \tag{16}$$

$E[FT]_{optimal}$ denotes the minimal overall flow time under a given system load and task size variability. Under the same system load and task size variability, $E[FT]_{sd\_cutoff}$ denotes the overall expected flow time when the optimal cut-offs of overall expected slowdown are used.

Similarly, we define the percentage performance degradation in overall expected slowdown as follows.

$$E[SD]_{Deg}\% = \frac{E[SD]_{ft\_cutoff} - E[SD]_{optimal}}{E[SD]_{optimal}}\% \tag{17}$$

Figures 11 and 12 illustrate the performance degradation in $E[FT]$ and $E[SD]$ respectively. Under high system loads and task size variabilities, $E[FT]_{Deg}\%$ is very high. For example, under the system load of 0.9 when α equals 0.4, $E[FT]_{Deg}\%$ is equal to 250%. $E[FT]_{Deg}\%$ decreases consistently with increasing α. This is because as α increases $p_{1\_sd\_opt}$ approaches $p_{1\_ft\_opt}$ (see Figures 5 and 7).

We notice that $E[SD]_{Deg}\%$ lies in the range of 10%- 60% for all system loads and task size variabilities considered. In general, small p_1 values (cut-offs) improve the both the overall expected slowdown and flow time. However, the use of very small p_1 values to optimise the overall expected slowdown can result in the overall expected flow time (or waiting time) deteriorate significantly (250%).

8 The impact of quanta (cut-offs) for more than 2-levels

In this section, we briefly discuss the effect of cut-offs on a multi-level time sharing policy consisting of 3 levels. We define the factor of improvement in performance in a 3-level policy over a 3-level policy as follows. The factor improvement in overall expected flow time is given by;

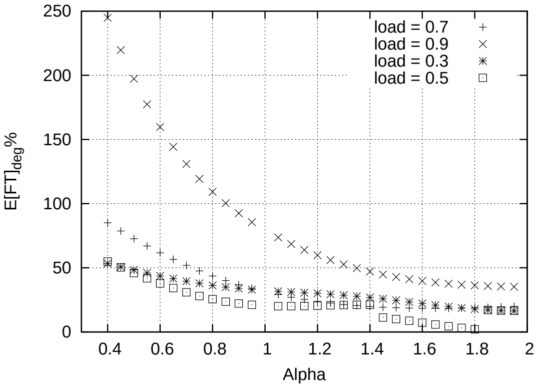

Figure 11: Performance degradation in E[FT]

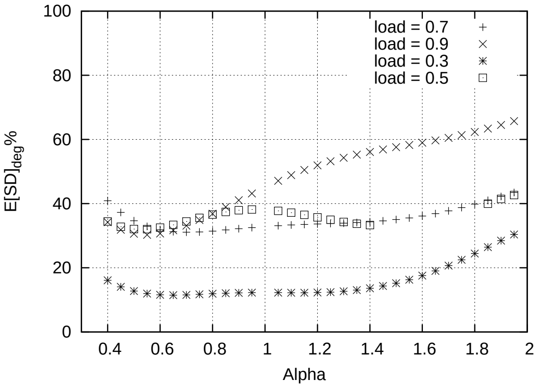

Figure 12: Performance degradation in E[SD]

$$E[FT]_{Imp} = \frac{E[FT]_{optimal\_N=2}}{E[FT]_{optimal\_N=3}} \tag{18}$$

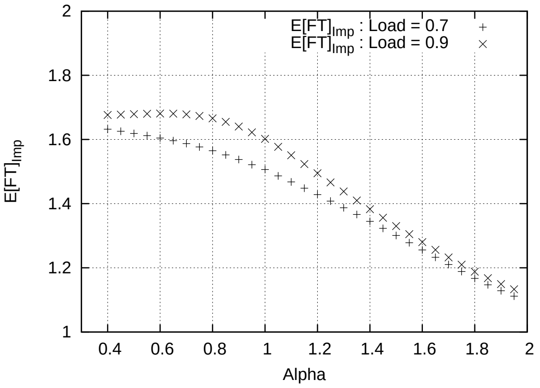

Figure 13: Factor of improvement E[FT]

Similarly the factor of improvement in overall expected slowdown is given by;

$$E[SD]_{Imp} = \frac{E[SD]_{optimal\_N=2}}{E[SD]_{optimal\_N=3}} \tag{19}$$

Figures 13 and 14 plot the behaviour of $E[FT]_{Imp}$ and $E[SD]_{Imp}$. The important point here is that 3-level policy outperforms 2-level policy for all the cases.

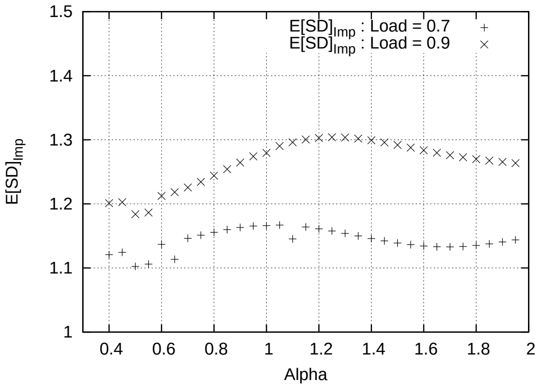

Figure 14: Factor of improvement E[SD]

Figures 13 and 14 illustrate the effect of cut-offs (i.e. p_1 and p_2) on the performance. We see from the figures that for each metric there is minimum which corresponds to optimal cut-offs. As we saw before, optimal p_1 is small compared to the largest task, p ($=10^7$), in the service time distribution. In this paper we do not investigate the uniqueness of quanta under 3-level policy. However, as we saw before we see that there is a sudden drop in optimal p_1 and optimal p_2 of overall expected flow time (see Figure 15). As the number of levels increase, finding the optimal cut-offs becomes more difficult. Graphical representations such as figures 15 and 16 will allow us to identify the ranges of the optimal cut-offs approximately. However, such plots are only possible for the case of 2 and 3 levels.

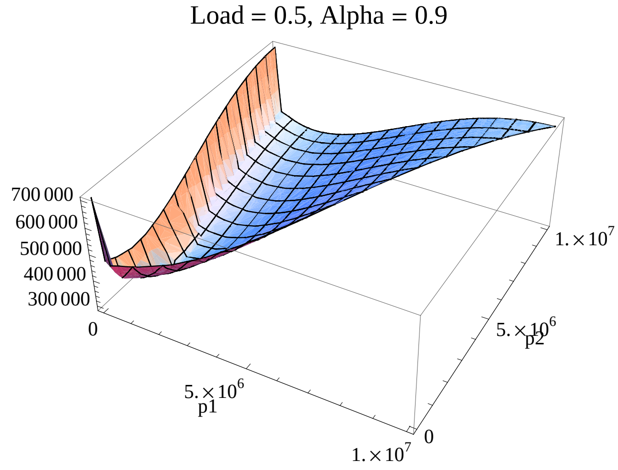

Figure 15: Overall expected Flowtime

9 Conclusion

In this paper, we investigated the effect of quanta on the overall performance of a multi-level time sharing under Heavy-tailed workloads. Using a 2-level system, we showed that for a given system load and task size variability there exists a unique set of quanta that would produce the minimal overall expected slowdown. The set of quanta that would produce the minimal overall expected flow time, however, may not be unique when the system load is inbetween 0.5 and 0.6. We showed that there is a sudden drop in 1^{st} optimal quantum (and an increase in 2^{nd} 'optimal' quantum) that occurs between the system loads of 0.6 and 0.7 when the performance is measured using overall expected flow time. In section 8 we showed that a 3-level policy with the optimal set of quanta outperforms a 2-level policy with the optimal set of quanta. In this paper, we did not investigate the uniqueness of optimal quanta for more than 2 levels.

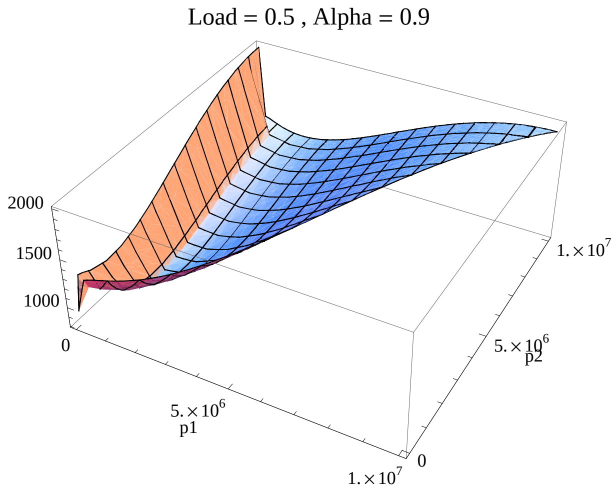

Figure 16: Overall expected slowdown

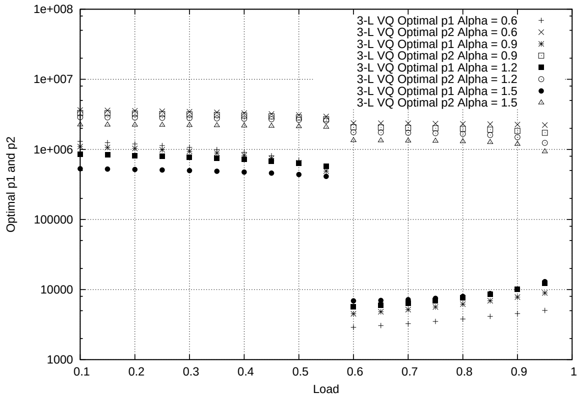

Figure 17: Behaviour of Optimal p_i

When the number of level are equal to 2, p_1 ($=1^{st}$ quantum) corresponding to minimal overall expected slowdown is very small compared to the p_1 corresponding to overall expected flow time. When the policy is performing at its minimal overall expected flow time, the fraction of tasks completed at level 1 is more than 95% for all the scenarios considered. We showed that under high system loads and task size variabilities $E[FT]_{Deg}\%$ is very high. $E[SD]_{Deg}\%$, however, lies in the range of 10%- 60% for all system loads and task size variabilities considered. In general, small p_1 improves both the overall expected slowdown and flow time. However, the use of very small of p_1 in order to optimise the overall expected slowdown can result in the overall expected flow time to deteriorate significantly (by around 250%).

References

Aalto, S., Ayesta, U., Borst, S., Misra, V. & Nez-Queija, R. (2007), 'Beyond processor sharing', *ACM SIGMETRICS Performance Evaluation Review* **34**(4).

Aalto, S., Ayesta, U. & Nyberg-Oksanen, E. (2004), 'Two-level processor-sharing scheduling disciplines: mean delay analysis', *Proceedings of the joint international conference on Measurement and modeling* *of computer systems, New York, USA* **32**(1), 97–105.

Aalto, S., Ayesta, U. & Nyberg-Oksanen, E. (2005), 'M/g/1/mlps compared to m/g/1/ps', *Operations Research Letters* **33**(5), 519–524.

Arlitt, M. & Jin, T. (1999), 'Workload characterization of the 1998 world cup web site', *technical report, Hewlett-Packard Laboratories* .

Arlitt, M. & Williamson, C. L. (1996), 'Web server workload characterization, the search for invariants', *In Proc. 1996 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems* pp. 126–138.

Barford, P., Bestavros, A., Bradley, A. & Crovella, M. E. (1999), 'Changes in web client access patterns: Characteristics and caching implications', *World Wide Web, special issue on characterization and performance evaluation* **2**, 15–28.

Broberg, J., Tari, Z. & Zeephongsekul, P. (2006), 'Task assignment with work-conserving migration', *Parallel Comput.* **32**(11-12), 808–830.

Coffman, E. G. & Kleinrock, L. (1968), 'Feedback queueing models for time-shared systems', *Journal of the ACM (JACM)* **15**(4), 549–576.

Crovella, M. E. & Bestavros, A. (1997), 'Self-similarity in world wide web traffic: evidence and possible causes', *IEEE/ACM Trans. Netw.* **5**(6), 835–846.

Crovella, M. E., Taqqu, M. S. & Bestavros, A. (1998), 'Heavy-tailed probability distributions in the world wide web', pp. 3–25.

Crovella, M., Harchol-Balter, M. & Murta, C. (1998), 'Task assignment in a distributed system: Improving performance by unbalancing load', *Proc. ACM Sigmetrics Conf. Measurement and Modeling of Computer Systems* pp. 268–269.

Harchol-Balter, M. (2000), 'Task assignment with unknown duration', p. 214.

Harchol-Balter, M., Crovella, M. & Murta, C. D. (1999), 'On choosing a task assignment policy for a distributed server system', *Journal of Parallel Distributed Computing* **59**(2), 204–228.

Heacox, H. C. & Purdom, P. W. (1974), 'Analysis of a multi-level time-sharing model', *BIT Numerical Mathematics* **14**(4), 407–412.

Jayasinghe, M., Tari, Z., Zeephongsekul, P. & Broberg, J. (2008), 'On the performance of multi-level time sharing policy under heavy-tailed workloads', *http://goanna.cs.rmit.edu.au/∼mjayasin/publications/technicalreport1.pdf* .

Nuyens, M. & Wierman, A. (2008), 'The foreground-background queue: a survey', *Performance Evaluation* **65**(3-4), 286–307.

Psounis, K., Molinero-Fernández, P., Prabhakar, B. & Papadopoulos, F. (2005), 'Systems with multiple servers under heavy-tailed workloads', *Perform. Eval.* **62**(1-4), 456–474.

Righter, R. & Shanthikumar, J. G. (1990), 'On external service disciplines in single-stage queueing systems', *Journal of Applied Probability* **27**, 409–416.

Schrage, L. E. (1967), 'The queue m/g/1 with feedback to lower priority queues', *Management Science* **13**(7), 466–471.

Tari, Z., Broberg, J., Zomaya, A. Y. & Baldoni, R. (2005), 'A least flow-time first load sharing approach for distributed server farm', *Journal of Parallel Distributed Computing* **65**, 832–842.

Yashkov, S. F. (1987), 'Processor-sharing queues: some progress in analysis', *Queueing Syst. Theory Appl.* **2**(1), 1–17.