On Maximizing Provider Revenue in Market-Based Compute Grids

Vladimir Marbukh and Kevin Mills
National Institute of Standards and Technology
100 Bureau Drive, Stop 8920
Gaithersburg, MD 20899-8920
E-mail: marbukh@nist.gov

Abstract- Market-based compute grids encompass service providers offering limited resources to potential users with varying demands and willingness to pay. Providers face difficult decisions about which jobs to admit and when to schedule admitted jobs. For this reason, researchers investigate various heuristics for admission control and scheduling that aim to yield high revenue for providers. Such research has no framework within which to understand the revenue bounds associated with various workloads. This paper proposes a tractable analytical model for joint optimization of job admission and scheduling strategies aimed at provider revenue maximization. We show how solving this model yields maximum provider revenue given a linear user utility function. Our model can be used to understand the operating limits of heuristics for admission control and scheduling, and can also be used to investigate the implication of varying job mixes.

I INTRODUCTION

Emerging Grid technologies pose a challenging problem of efficient resource allocation in complex, decentralized systems. Since resources are shared by multiple users (i.e., applications) and performance of each user is typically characterized by multiple competing criteria, resource management includes the following two major tasks: (a) making best use of allocated resources for each user by resolving trade-offs among competing user criteria, and (b) sharing resources among different users.

Fig.1 illustrates the general nature of resource management in a market-based compute grid. Each user discovers some service providers within a grid and then selects one provider with which to interact. The user submits a job request indicating the required resources and amount the user is willing to pay. Typically, a user is willing to pay a fixed amount for a job completed by a specified deadline, but a discounted amount for a late job. The service provider considers the request in light of existing and expected workload and then decides whether or not to accept the job. Upon completion of an accepted job, a provider earns revenue according to the user's expressed willingness to pay. Late jobs may require the provider to pay a penalty; thus, revenue for a job could be negative. If the service provider rejects the job, then the user forwards its request to another service provider, and so on.

In this paper, we investigate resource management from the viewpoint of a service provider who aims to maximize revenue, given various user demands with associated potential for rewards and penalties. In related work, researchers who investigate market-based compute grids typically model users willingness to pay (utility) as a reward for completing a job by a deadline and a decay rate, which defines the slope of a linearly decreasing function of the reward over time for late jobs [1]-[7]. As reward decays beyond zero, user utility becomes negative and a provider must pay a corresponding penalty. Many researchers devise heuristics for admission control and scheduling of jobs by service providers and then use simulation to evaluate performance of those heuristics when subjected to a mix of job classes. Each job class is defined by a deadline and associated reward, along with a decay rate for exceeding the deadline (and possibly a bound on the penalty for late jobs).

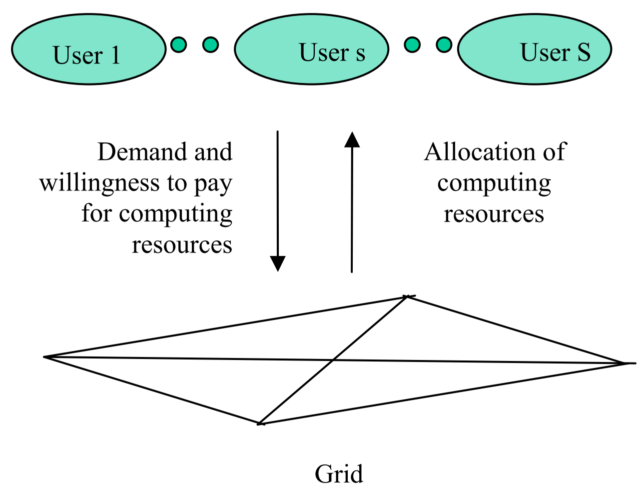

Figure 1. General view of resource management in a Grid

This paper proposes a tractable analytical model for joint optimization of job admission and scheduling strategies aimed at provider revenue maximization. We solve this model under the assumption that potential provider penalties are unbounded, and we analyze key model parameters. In this particular case the scheduling optimization problem can be solved explicitly, yielding priority scheduling with priorities determined by job urgency. We demonstrate three applications of the model. First, given two job classes, we determine optimal admission probabilities and associated revenue as the job mix varies for a fixed aggregate load. Second, given a fixed job mix with four classes, we determine optimal admission probabilities and associated revenue as aggregate load varies. Third, we show how optimal admission probabilities computed from our model allow a simulated, compute cluster to achieve maximum revenue even under bounded penalties. Our analytical solution may be used to determine maximum achievable provider revenue for a given aggregate load and job mix, assuming unbounded provider penalties. Our model can be used to understand the operating limits of heuristics for admission control and scheduling, and can also be used to investigate the implication of varying job mixes.

We begin by describing (in section II) our model of user criteria for job classes. In section III, we present our general provider model, which includes both admission control and scheduling, and describe a provider with linear utility. Section IV solves our model analytically and numerically for the case of a single job class with a linear revenue function. Section V gives numerical results from our model when considering user demands with two and four job classes. Section VI applies our analytical model to a simulated, compute cluster that operates with bounded penalties. We conclude in section VII.

II USER MODEL

We model users as job submitters, where each job includes an expressed willingness to pay that consists of two parts: base value and delay-dependent decay, which can be seen as diminished value for jobs completed late. Thus, we model jobs as being delay sensitive.

We assume that there are S classes of jobs, where all jobs of each class $s = 1,..,S$ have the same delay sensitivity. A job's delay sensitivity is quantified by the non-increasing utility function $u_s(\tau)$, where τ is the queuing delay. Function $u_s(\tau)$ can be interpreted as the willingness to pay. We assume that different jobs $j = 1,..,J_s$ of the same class $s = 1,..,S$ have the different budgets, and model this situation by assuming that job j of the class s has utility function

$$u_{sj}(\tau) = u_{sj} - v_s(\tau), \quad (1)$$

where function $v_s(\tau)$ depends on the queuing delay τ for job j and constant u_{sj} represents the basic value of the job j of class s .

We consider a generic utility function (see Fig. 2) often used [1]-[6] in grid computing:

$$v(\tau) = \begin{cases} v^{+} & if \quad 0 < \tau \le \tau^{\min} \\ a - b\tau & if \quad \tau^{\min} < \tau \le \tau^{\max} \\ -v^{-} & if \quad \tau^{\max} < \tau \end{cases} \quad (2)$$

where

$$a = \frac{v^{+}\tau^{\max} - v^{-}\tau^{\min}}{\tau^{\max} - \tau^{\min}} \quad (3)$$

$$b = \frac{v^{+} + v^{-}}{\tau^{\max} - \tau^{\min}} \quad (4)$$

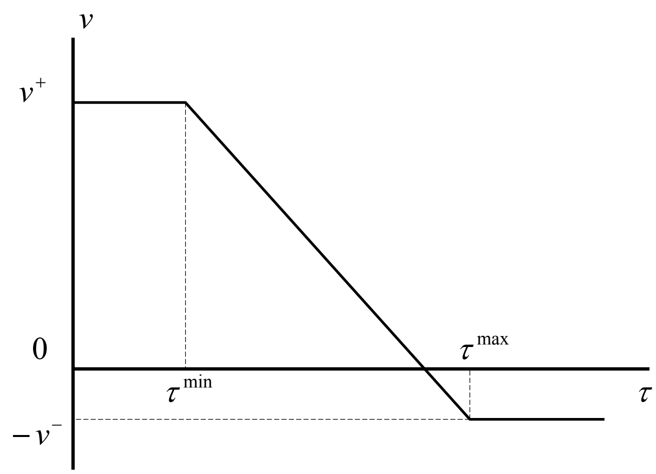

Figure 2. Generic utility function

Piecewise linear utility function (2)-(4) represents willingness to pay as a base value (v^{+}) for completing a job on time (by $\tau^{\min}$) with a decreasing value for late jobs, up to some bound ($-v^{-}$).

We can simplify (2)-(4) to a linear utility function,

$$v_s(\tau) = v_{0s} - v_{1s}\tau, \quad (5)$$

as depicted in Fig. 3, where v_{0s} represents base value for completing a job in class s without queuing delay and v_{1s} represents the rate of decay in job value due to queuing delay. Linear utility function (5) ignores the offset ($\tau^{\min}$) and removes the penalty bound ($-v^{-}$). We will see that these restrictions simplify the analysis, and that the offset can be reintroduced later when using numerical methods.

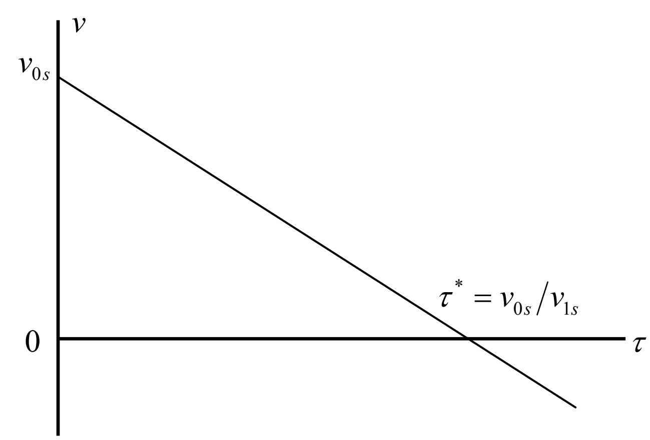

Figure 3. Linear utility function

Given queuing delay time τ, we assume that a provider charges a job of class $s = 1,..,S$ amount

$$p_s(\tau) = p_{0s} - v_s(\tau), \tag{6}$$

where p_{0s} is the base price for the job and $v_s(\tau)$ is a price reduction based on the queuing delay incurred by the job. Note that when $p_s(\tau) < 0$ the user is reimbursed for poor service. A user will submit job $j = 1,..,J_s$ of class $s = 1,..,S$ to a provider only when $p_{0s} \le u_{sj}$. Thus, the service provider can control demand by varying base prices p_{0s}, $s = 1,..,S$.

III Provider Model: Admission&Scheduling

A service provider can maximize revenue by controlling the admission and scheduling of jobs in various classes, which have associated utility functions. The problem of finding optimal admission probabilities and a related schedule for these jobs is nontrivial. In this section we show how solving two inter-related optimization problems (determining optimal admission probabilities and a related, optimal schedule) allows a service provider to maximize revenue.

We assume a service provider uses a pricing scheme (6) where base prices p_{0s} are fixed. We also assume that jobs of class s arrive according to a Poisson process of rate μ_s. We assume an $M/G/1$ service model: all accepted requests are serviced by a single server of capacity C, with service time for requests of class s being a random variable with probability distribution $B_s(t)$ with moments $b_s^{(i)} = \int t^{(i)} dB_s(t)$.

The provider employs admission control and scheduling, as shown in Fig. 4. Admission control admits an arriving job of class s with probability $q_s \in [0,1]$ and rejects this job with probability $1 - q_s$. Rejected jobs leave the system. After admission, jobs of class s enter a queue according to a Poisson process of rate $\lambda_s = q_s \mu_s$, which is serviced according to some scheduling regime. We assume non-preemptive, priority scheduling.

Since a completed job $j = 1,..,J_s$ of class $s = 1,..,S$ brings revenue (6), given queuing time $\tau = \tau_{js}$, the average provider revenue is

$$\overline{R} = \sum_s \lambda_s \{ p_{0s} - E[v_s(\tau_s)] \}, \tag{7}$$

which represents the sum over all job classes of the base price for jobs in a class minus the decay in value determined by the expected queuing delay for jobs in the class, weighted by the proportional arrival rate for jobs in the class. The goal of the provider is to maximize the average total revenue

$$\max_{0 \le \lambda_s \le \mu_s} \max_{scheduling} \overline{R} \tag{8}$$

where the average total revenue is given by (7).

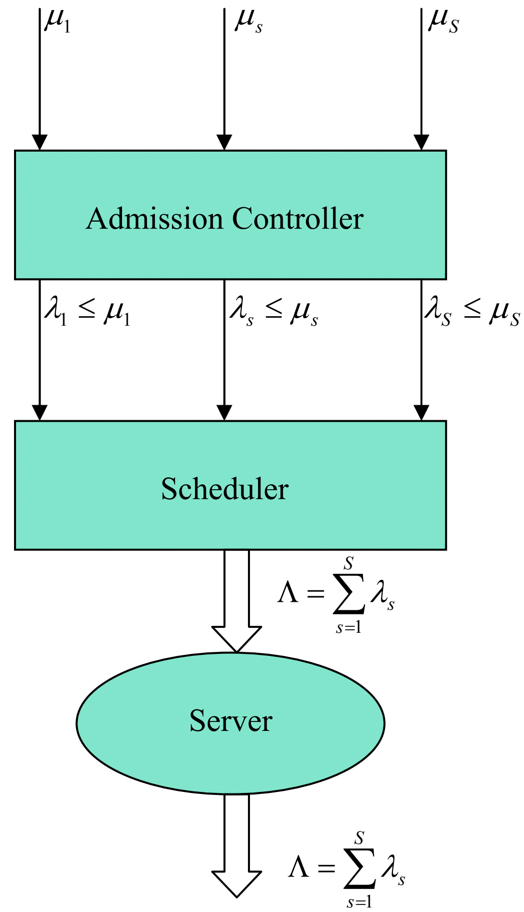

Figure 4. Admission Control and Scheduling

The case of a linear utility function (6) is comparatively simple because (a) average utility is equal to the utility of the average queuing delay, and (b) optimization problem (8) can be solved explicitly yielding the optimal scheduling. Given linear utility function (5) and assuming $v_{0s} = 0$, then $-v_{1s}\tau$ represents the amount of revenue lost when a late job is completed by time τ. From this, we obtain the following expression for the average total revenue (7):

$$\overline{R} = \sum_s \lambda_s (p_{0s} - v_{1s} T_s) \tag{9}$$

where the average waiting time of the class s is $T_s = E[\tau_s]$. Thus, optimization problem (8) takes the following form:

$$\max_{0\le\lambda_s\le\mu_s}\max_{scheduling}\sum_s \lambda_s(p_{0s}-v_{1s}T_s) \quad (10)$$

Optimization problem (10) can be decomposed into two optimization problems as follows. The first optimization problem (11) finds the optimal scheduling discipline and the vector of corresponding average waiting times $T^*(\lambda)=\left(T_1^*(\lambda),..,T_S^*(\lambda)\right)$, given vector of arrival rates $\lambda=(\lambda_1,..,\lambda_S)$:

$$\min_{scheduling}\sum_s \lambda_s v_{1s}T_s \quad (11)$$

The second optimization problem finds the optimal admission probabilities:

$$q_s^{opt}=\lambda_s^{opt}/\mu_s,\ \ s=1,..,S \quad (12)$$

where the vector of optimal arrival rates $\lambda^{opt}=(\lambda_s^{opt})$ is given by the solution to the following optimization problem:

$$\lambda^{opt}=\arg\max_{0\le\lambda_s\le\mu_s}\sum_s \lambda_s[p_{0s}-v_{1s}T_s^*(\lambda)] \quad (13)$$

In the class of scheduling disciplines without preemption, the solution to optimization problem (11) is as follows. Introduce the following notation:

$$f_s \overset{def}{=} v_{1s}/b_s^{(1)},\ \ s=1,..,S \quad (14)$$

and without loss of generality assume that job classes are served in the following order:

$$f_1\le f_2\le..\le f_S \quad (15)$$

Kleinrock [8] shows that, for the class of non-preemptive, work-conserving scheduling disciplines, the solution to optimization problem (11) is given by the Head-Of-the-Line (HOL) discipline with ordering shown by (15), where job class $i=2,..,S$ has priority over job class $j=1,..,S$ if $i>j$. The corresponding optimal average waiting times are

$$T_s^*=\frac{T_0}{(1-\sigma_s)(1-\sigma_{s+1})} \quad (16)$$

where

$$T_0=\frac{1}{2}\sum_s \lambda_s b_s^{(2)} \quad (17)$$

server utilization by a job class $s=2,..,S$ is

$$\rho_s=\lambda_s b_s^{(1)}/C \quad (18)$$

and server utilization by job classes $i=s,..,S$ is

$$\sigma_s=\sum_{i=s}^{S}\rho_i \quad (19)$$

Thus, for the optimal scheduling discipline, the average total revenue (9) becomes

$$\overline{R}=C\sum_s \frac{\rho_s}{b_s^{(1)}}\left[p_{0s}-\frac{T_0 v_{1s}}{(1-\sigma_s)(1-\sigma_{s+1})}\right] \quad (20)$$

and the optimization problem (13) takes the following form:

$$\max_{0\le\rho_s\le\beta_s}\sum_s \frac{\rho_s}{b_s^{(1)}}\left[p_{0s}-\frac{Cv_{1s}}{2}\frac{\sum_i \rho_i b_i^{(2)}}{\left(1-\sum_{i\ge s}\rho_i\right)\left(1-\sum_{i\ge s+1}\rho_i\right)}\right] \quad (21)$$

where $\beta_s=\mu_s b_s^{(1)}/C$.

IV Analysis & Numerical Results for One Job Class

In the case of a single job class, $S=1$, optimization problem (21) takes the following form:

$$\max_{0\le\rho\le\beta}\rho\left(p_0-\widetilde{v}_1\frac{\rho}{1-\rho}\right) \quad (22)$$

where

$$\widetilde{v}_1=\frac{1}{2}Cv_1b^{(2)} \quad (23)$$

Optimization problem (22)-(23) can be solved explicitly.

Introduce notation

$$\rho^*=1-\sqrt{\frac{\widetilde{v}_1}{p_0+\widetilde{v}_1}} \quad (24)$$

Solving optimization problem (22)-(23) yields the following admission probability $q^{opt}=\lambda^{opt}/\mu$:

$$q^{opt}=\min\left(1,\ \ \rho^*/\beta\right) \quad (25)$$

Combining (23)-(25) we obtain the following parameter region where admission control is ineffective:

$$\frac{p_0}{v_1}\ge\frac{Cb^{(2)}}{2}\left[\frac{1}{(1-\beta)^2}-1\right] \quad (26)$$

and the following parameter region where admission control is effective and the optimal admission probability $q^{opt}<1$:

$$\frac{p_0}{v_1}=\frac{Cb^{(2)}}{2}\left[\frac{1}{(1-q^{opt}\beta)^2}-1\right] \quad (27)$$

Regions (26) and (27) are shown in Fig. 5.

We can use numerical methods (e.g., conjugate gradient method) to solve for q_s^{opt} in order to maximize average total revenue in the case of an arbitrary number of job classes. We demonstrate the approach using a single job class. We reintroduce offset $\tau^{\min}$ to ensure that revenue associated with queued jobs does not begin to decay until after delay $\tau^{\min}$. Thus, we alter (20) accordingly:

$$\overline{R}=C\sum_s \frac{\rho_s}{b_s^{(1)}}\left[p_{0s}-\frac{\max(0,T_0-\tau^{\min})v_{1s}}{(1-\sigma_s)(1-\sigma_{s+1})}\right]. \quad (28)$$

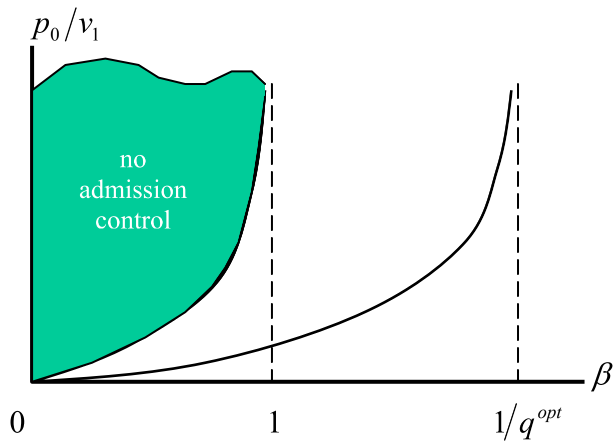

Figure 5. Phase diagram in the case of a single job class.

In the following, we assume normally distributed job lengths with moments: $b_s^{(1)} = 4400$ s and $b_s^{(2)}$ = (4400^2 + 1500^2) s. We further assume that our service provider can run 30 jobs in parallel $(C = 30/4400)$. We also assume a single job class for which computing time costs \$1000/hr (i.e., $p_0 = 1000$). We fix the delay before utility begins to decay as $\tau^{\min} = 3 \times b_s^{(1)}$. We assume a base decay rate d = .001. We can skew the decay rate by a factor k, such that $v_1 = d \times k$. Define $\tau = p_0/v_1$ to be the delay before revenue for a job becomes negative. We use (28) to compute the average total revenue as we vary utilization (ρ) and as we vary k to create a range of decay rates and corresponding values for τ. Fig. 6 plots the results for decreasing τ.

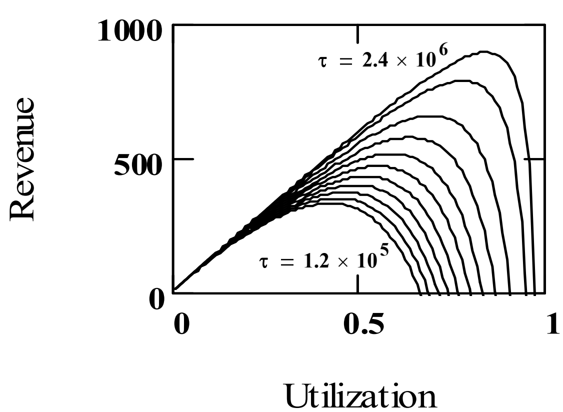

Figure 6. Revenue vs. utilization for varying decay rates absent admission control.

Each curve reveals a similar story. Revenue increases to a maximum and then drops steeply as the server receives too many jobs to provide the required delay. The period of increase represents the region where admission control is not necessary and the period of decrease represents the region where admission control would prove effective. Shorter values of τ (i.e., higher decay rates v_1) yield lower maximum revenues and cause revenue to begin declining at lower utilizations.

Next we solve optimization problem (8), using equations (14) and (15) and then use those results to solve numerically optimization problem (21), yielding q^{opt}, as plotted in Fig. 7 for the same values of τ shown in Fig. 6. Applying these optimal admission probabilities will prevent server overload and allow the provider to maintain maximum revenue under increasing load. Fig. 8 gives the revenue curves when jobs are admitted with probability q^{opt}.

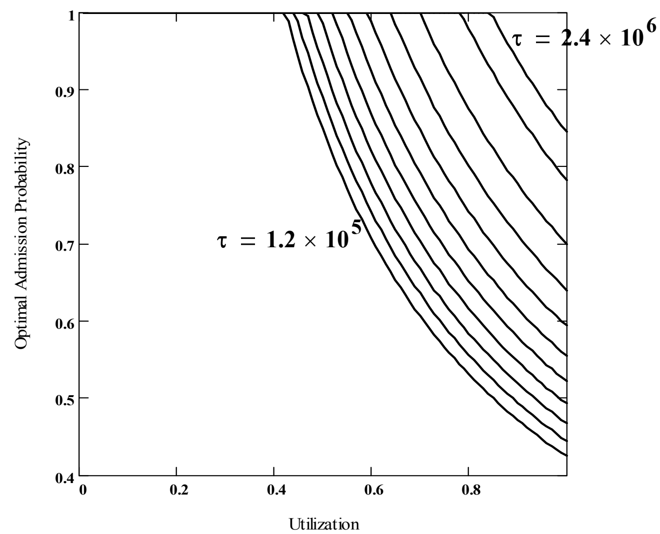

Figure 7. Optimal admission probability vs. utilization for varying decay rates.

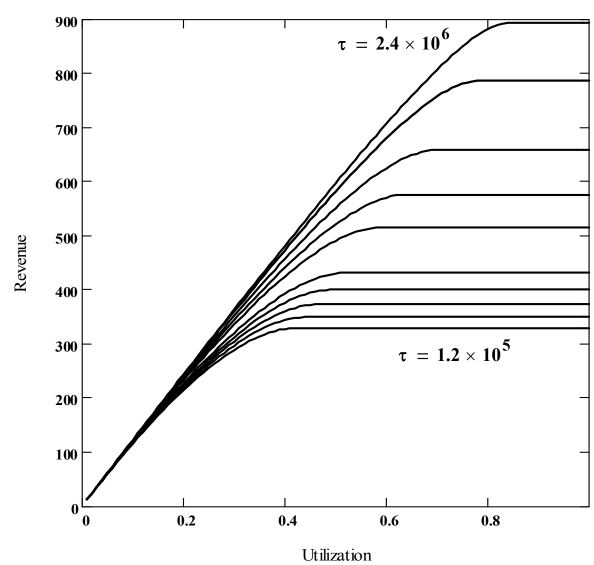

Figure 8. Revenue vs. utilization for varying decay rates when using optimal admission control.

V NUMERICAL RESULTS FOR VARYING JOB MIXES

Next we consider a workload with two job classes, $S = 2$. For one job class (high-value, high-urgency) $\tau = 3\times10^5$ and for the other (low-value, low-urgency) $\tau = 6\times10^5$. Fig. 9 shows the utility functions. Our analysis investigates how q^{opt} changes as we vary the job mix by altering the ratio of high-value jobs to low-value jobs as we hold ρ constant at 80%. We also examine how these changes affect average total revenue. We keep C, $b_s^{(1)}$, $b_s^{(2)}$ and $\tau^{\min}$ unchanged from our analysis of one job class.

We assume that high-value jobs yield five times the revenue of low-value jobs (\$30,000 vs. \$6,000) and decay 10 times as fast (0.1 vs. 0.01). Fig. 10 plots the optimal admission probabilities for each job class, while varying the proportion of high-value jobs. To maximize revenue, the service provider must admit all high-value jobs until the proportion of high-value jobs reaches 75 % (see Fig. 11, where revenue reaches a maximum at this job mix). Further, if there are any high-value jobs, then the service provider cannot admit more than 70 % of the low-value jobs, an admission rate that falls as the proportion of high-value jobs increases until the service provider must reject all low-value jobs once the proportion of high-value jobs reaches 60 %.

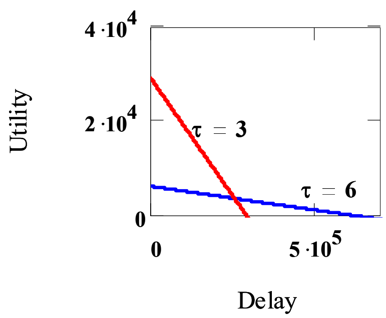

Figure 9. Linear utility functions for two job classes.

Next we consider a more complex workload with four job classes, $S = 4$, as illustrated in Fig. 12. Five percent of jobs (high-value, high-urgency) yield about \$60,000 and decay at a rate of \$4 per second $(\tau = 1.5\times10^4)$. Twenty percent of jobs (high-value, low-urgency) yield \$40,000 and decay at a rate of \$1.33 per second $(\tau = 3\times10^4)$. Fifteen percent of jobs (low-value, high-urgency) yield \$20,000 and decay at \$1.00 a second $(\tau = 2\times10^4)$. The remaining 60 % of jobs (low-value, low-urgency) yield \$10,000 and decay at \$0.25 a second $(\tau = 4\times10^4)$. Even for this complex workload, our model can be used to determine the optimal admission probabilities to achieve maximum revenue for varying aggregate workloads.

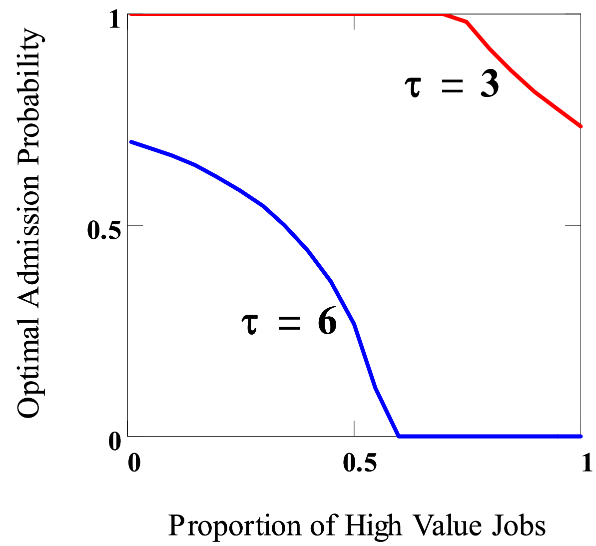

Figure 10. Optimal admission probability for varying proportion of high-value jobs given utilization of 80 % (τ is in units of 10^5).

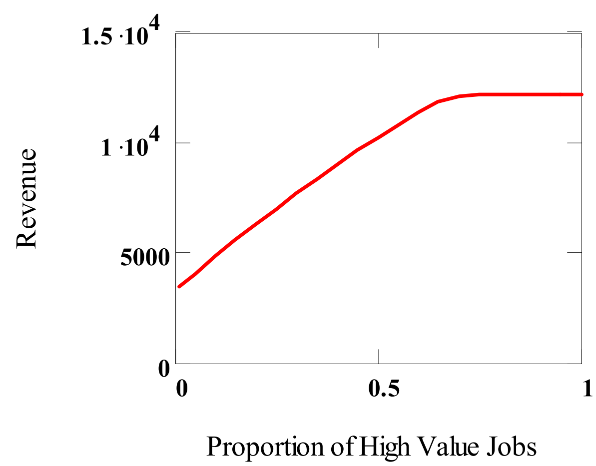

Figure 11. Revenue for varying proportion of high-value jobs, given utilization of 80 % and applying optimal admission probabilities from Figure 10.

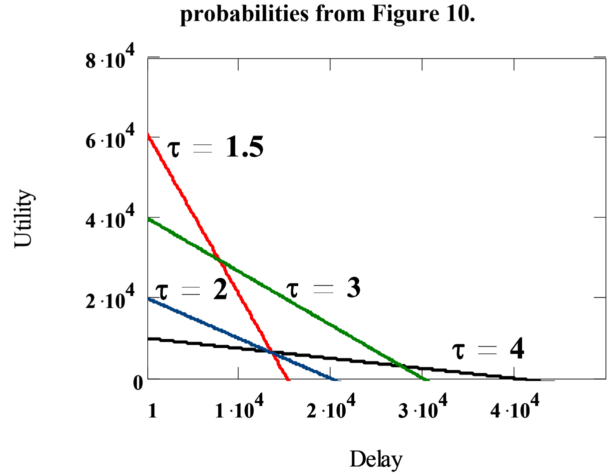

Figure 12. Linear utility functions for four job classes.

Figures 13-19 plot various system characteristics against aggregate load

$$\beta_{\Sigma} = \frac{1}{C}\sum_{s=1}^{S} \mu_s b_s^{(1)} \quad (29)$$

assuming that the load by each class $s = 1,..,S$ changes proportionally to the aggregate load (29), so the load by each class stays constant relative to the aggregate load (29).

Fig. 13 plots the optimal admission probabilities for each job class as aggregate load increases to 140 %. Note that admission probabilities are quite sensitive to load. For example, low-value, low-urgency jobs are shut out over a 6 % load range (between 15 % and 21 % load). Next, low-value, high-urgency jobs are cut off as load moves over the range of 28 % to 40 %. High-value, high-urgency jobs are gradually squeezed out over the range of 60 % to 80 % load. Finally, high-value, low-urgency jobs begin to be rejected as aggregate load passes 90 %. These admission probabilities lead to the maximum revenue curve shown in Fig. 14. The revenue curve exhibits four distinct, linear regions – where the slope changes as each job class becomes subjected to admission control.

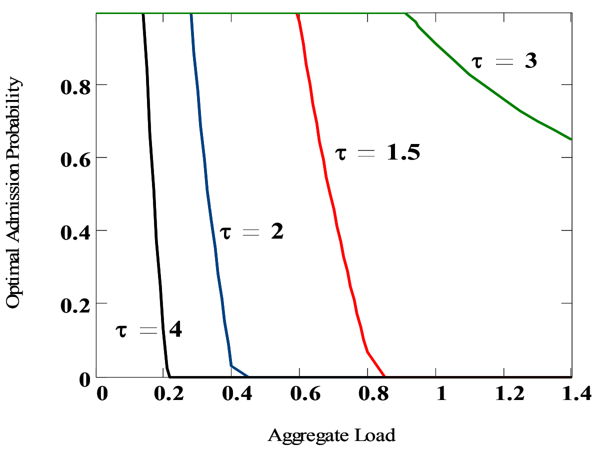

Figure 13. Optimal admission probability for each of four job classes as aggregate load increases (τ is in units of 10^4).

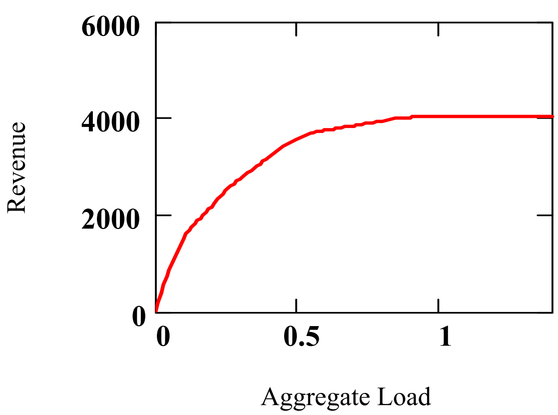

Figure 14. Revenue as aggregate load increases while applying optimal admission probabilities from Fig. 13 to the job mix defined in Fig.12.

VI Simulation Results for Bounded Penalties

Through simulation, we demonstrate that our analytical model for unbounded penalties can be applied to maximize provider revenue even when penalties are bounded. We simulated a 5000-processor cluster facing a stream of jobs with the general characteristics given in Table 1.

Table 1. General Job Characteristics

Job Width		500 processors
Job Length	Average	7200 seconds
	Std. Dev.	2400 seconds
	Minimum	300 seconds
Job Value	Base Value	\$1.00/processor/hour[1]
	High Skew	5 x Base Value
Job Decay Rate	Base Decay	\$0.10/second
	Urgent Skew	10 x Base Decay
$\tau^{\min}$		3 x Job Length
Maximum Penalty		5 % of Job Value

For our job mix, we assumed 20 % of jobs had high value (80 % had base value) and 20 % of jobs were urgent (80 % were not). We combined these assumptions about value and urgency to form four job classes, as shown in Table 2, which also indicates the value and decay rate for an average job in each class. On average, low-value (Lv) jobs are worth \$1000 and high-value (Hv) jobs are worth \$5000; thus, the weighted average job value is \$1800. Low-urgency (Lu) jobs expire (τ) 10,000 seconds after $\tau^{\min}$ and high-urgency (Hu) jobs expire 1,000 seconds after $\tau^{\min}$. An expired job is removed from the queue and the provider is assessed the maximum penalty.

Table 2. Characteristics Distinguishing Job Classes

Class	LvLu	LvHu	HvLu	HvHu
Ratio	64 %	16 %	16 %	4 %
Value	Job Length x Job Width x Base Value	Job Length x Job Width x Base Value	Job Length x Job Width x Base Value x High Skew	Job Length x Job Width x Base Value x High Skew
For Avg. Job	2 x 500 x 1 = \$1000	2 x 500 x 1 = \$1000	2 x 500 x 1 x 5 = \$5000	2 x 500 x 1 x 5 = \$5000
Decay Rate	Base Decay	Base Decay x Urgent Skew	Base Decay x High Skew	Base Decay x High Skew x Urgent Skew
For Avg. Job	\$0.10/s	\$0.10/s x 10 = \$1.00/s	\$0.10/s x 5 = \$0.50/s	\$0.10/s x 5 x 10 = \$5.00/s

Our simulator scheduled jobs in priority order according to (14)-(15). We simulated two alternatives for admission control: (a) admit all jobs and (b) admit jobs according to

[1] Base job value derived from recent commercial offerings.

q_s^{opt}, where s denotes the job class. To compute q_s^{opt} we used numerical methods as outlined in previous sections, but we made one minor adjustment. Specifically, we multiplied estimated queuing delay by a coefficient $(\Phi = 0.56)$ to account for the fact that expired jobs are removed from the queue, which reduces queuing delay under higher utilizations. To determine a value for Φ we compared results from two sets of simulations that admit all jobs. In one set, penalties were bounded and expired jobs were removed from the queue. In the second set, penalties were unbounded and jobs did not expire. Simulated queuing delays diverged with increasing utilization, with the greatest divergence occurring over the range of 0.85 to 1.00, where bounded penalties led to queuing delays on average 56 % of those with unbounded penalties.

We incorporated Φ into (28) to yield a modified revenue function,

$$\bar{R} = C \sum_s \frac{\rho_s}{b_s^{(1)}} \left[p_{0s} - \frac{\max(0, \Phi \cdot T_0 - \tau^{\min}) v_{1s}}{(1-\sigma_s)(1-\sigma_{s+1})} \right], \qquad (30)$$

and then at each load of interest we solved for q_s^{opt}. Figure 15 plots q_s^{opt} for the job classes depicted in Table 2.

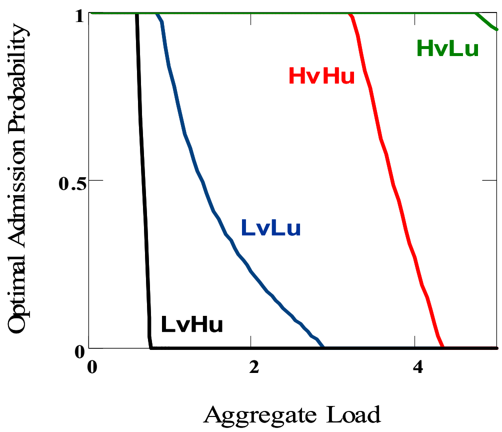

Figure 15. Optimal admission probability vs. aggregate load for simulated job classes described in Table 2.

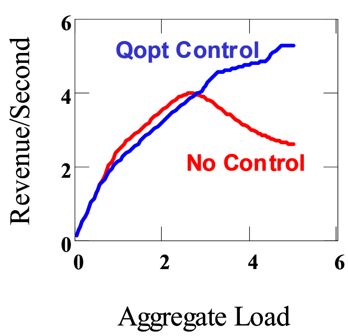

Figure 16. Average revenue per second vs. aggregate load when admitting all jobs (red) and when applying optimal admission probabilities (blue).

Using our two admission-control strategies, we subjected our simulation to aggregate workloads of up to 500 % and measured various performance indicators, such as utilization, revenue per second, revenue per job and queuing delay. At each load we simulated 500 six-month periods. Figures 16-19 graph key performance results for two cases: no admission control and optimal admission control, where both cases assume optimal priority scheduling. Note that due to bounded penalties, even in a case of “no control”, optimal priority scheduling effectively introduces a form of admission control since expired jobs are removed from the queue. This can explain some peculiarities in Figures 16-19, including revenue rate increase even as the system becomes overloaded (Fig. 16) as well as decreases in queuing delay (Fig. 18) and utilization (Fig. 19) with increase in aggregate load.

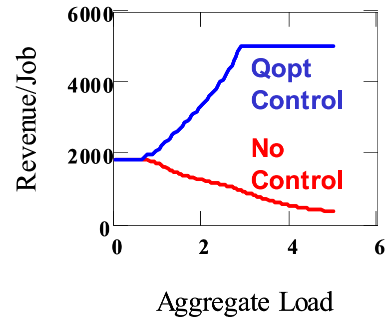

Figure 17. Average revenue per job vs. aggregate load when admitting all jobs (red) and when applying optimal admission probabilities (blue).

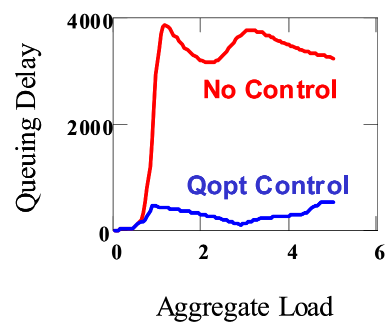

Figure 18. Average queuing delay vs. aggregate load when admitting all jobs (red) and when applying optimal admission probabilities (blue).

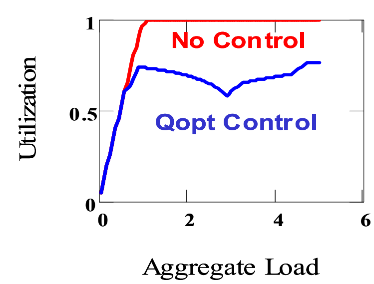

Figure 19. Average utilization vs. aggregate load when admitting all jobs (red) and when applying optimal admission probabilities (blue).

As illustrated in Figure 16, applying optimal admission probability for each of the four job classes enabled the service provider to achieve higher revenue beyond 300 % load and to reach maximum revenue as load neared 500 %, where the arrival of high-value jobs was sufficient to allow rejection of all low-value jobs. In an interval between 65 % and 300 % load, admitting all jobs achieved more revenue, which results from the approximation of Φ used in (29).

When all jobs were admitted, the maximum penalty for expired jobs limited the provider's lost revenue. Further, expiring jobs were removed from the queue, which served as an ex post facto admission control. These factors enabled the provider to increase earned revenue until load reached about 300 %, after which revenue declined because of an increasing rate of expired jobs. These effects can also be seen in Figure 17, where both control strategies yield an average value of $1800 per job until 90 % load. Beyond 90 % load, admitting all jobs led to a steady decline in revenue per job as losses mounted due to the increasing rate of expired jobs. On the other hand, beyond 90 % load, applying optimal admission probabilities had two positive effects on revenue per job: (a) rejected jobs did not incur penalty costs and (b) a higher proportion of high-value jobs were admitted in place of low-value jobs. Beyond 285 %, applying optimal admission probabilities resulted in acceptance of only high-value jobs, leading to average revenue of $5000 per job.

Figure 18 shows that applying optimal admission probabilities enabled the service provider to hold queuing delays low, as compared with the case of admitting all jobs, where high queuing delays caused some jobs to decay in value and other jobs to expire and incur penalty costs. Figure 19 illustrates that limiting admissions to maximize revenue reduced utilization. When all jobs were admitted, cluster utilization reached 100 % (given sufficient load). When job admissions were restricted, cluster utilization averaged about 69 %.

We also simulated a scenario where job widths varied: 50 % of the workload required 250 processors, 30 % of the workload required 750 processors and 20 % of the workload required 1500 processors. Those results (not shown here) provide the same fundamental pattern of revenue and queuing delay. Of course, sometimes a scheduled job would not fit into the cluster and had to be delayed until there was sufficient space (we did not simulate backfilling). This led to lower utilizations: 90 % average when all jobs were admitted and 65 % average when optimal admission probabilities were applied. Traditionally, operators of large compute clusters strive to achieve utilizations above 90 % (higher is considered better). Our results suggest that in market-based compute grids service providers might need to take a different view.

VII CONCLUSIONS

Market-based compute grids encompass service providers offering limited resources to potential users with varying demands and utility (willingness to pay). Researchers typically model utility as a reward for completing a job by a deadline and a decay rate, which defines the slope of a linearly decreasing function of the reward over time if a job is late. As reward decays beyond zero, user utility becomes negative and a provider must pay a corresponding penalty. Under such conditions, providers face difficult decisions about which jobs to admit and when to schedule admitted jobs. For this reason, researchers investigate various heuristics for admission control and scheduling that aim to yield high revenue for providers.

This paper has proposed a tractable analytical model for joint optimization of job admission and scheduling strategies aimed at provider revenue maximization. We solved this model under the assumption that potential provider penalties are unbounded, and we analyzed key model parameters. We demonstrated how the model could be used to compute optimal admission probabilities under a complex mix of jobs. We reported simulation results suggesting that, in market-based compute grids, where poor service has associated costs, providers must restrict resource utilization in order to maximize revenue. Our model could be used to understand the operating limits of proposed heuristics for admission control and scheduling, and could also be used to investigate the implication of varying job mixes and workloads.

Further work remains to incorporate effects from server price adjustments and to consider collections of multiple service providers and clients within a grid economy. We also plan to validate our optimization results against available data [9] on current web services.

VIII. ACKNOWLEGMENTS

We thank C. Dabrowski and A. Nakassis for their careful reading of the paper and for their useful comments.

REFERENCES

[1] B. N. Chun and D. E. Culler. User-centric performance analysis of market-based cluster batch schedulers. *Proceedings of the 2nd IEEE International Symposium on Cluster Coniputing and the Grid* May 2002; pp. 30-38.

[2] D.E. Irwin, L. E. Grit and J. S. Chase. Balancing Risk and Reward in a Market-Based Task Service. *Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing* 2004; pp. 160-169.

[3] F. I. Popovici and J. Wilkes. Profitable services in an uncertain world. *Proceedings of the ACM/IEEE Supercomputing Conference* Nov. 2005; pp. 36-47.

[4] C.S Yeo and R. Buyya. Service level agreement based allocation of cluster resources: Handling penalty to enhance utility. Proceedings of the 7th IEEE International Conference on Cluster Computing 2005; pp. .

[5] D. Vengerov Adaptive Utility-Based Scheduling in Resource-Constrained Systems. *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence*, December 2005; LNCS 3809: pp. 477-488.

[6] A. AuYoung, L. Grit, J. Wiener and J. Wilkes. Service contracts and aggregate utility functions. *Proceedings of the 15th IEEE International Symposium on High Performance Distributed Computing* 2006; pp. 119-131.

[7] A. Verma and S. Ghosal. On Admission Control for Profit Maximization of Networked Service Providers. *Proceedings of the Conference on World Wide Web*, 2003.

[8] L. Kleinrock, Queueing Systems, Volume II:Compute Applications, John Willey, 1976.

[9] A.K. Iyengar, M.S. Squillante, and L. Zhang. Analysis and characterization of large-scale Web Server Access patterns and performance. *Proceedings of the Conference on World Wide Web*, 1999.