Game-Theoretic Approach for Grace-Period Policy in Supercomputers

Fei He
Texas A&M University-Kingsville
Kingsville, TX, USA
fei.he@tamuk.edu

Nageswara S. V. Rao
Oak Ridge National Laboratory
Oak Ridge, TN 37831, USA
raons@ornl.gov

Chris Y. T. Ma
The Hang Seng University of Hong Kong
chrisma@hsu.edu.hk

***Abstract*—Job scheduling at supercomputing facilities is important for achieving high utilization of these valuable resources while ensuring effective execution of jobs submitted by users. The jobs are scheduled according to their specified resource demands such as expected job completion times, and the available resources based on allocations. Jobs that overrun their allocated times are terminated, for example, after a grace-period. It is non-trivial and often very complex for users to accurately estimate the completion times of their jobs, and consequently they face a dilemma: underestimate the job time to have a higher priority and risk job termination due to overrun, or overestimate it to ensure its completion and risk its delayed execution. In this paper, we investigate whether providing grace-period can benefit facility performance by developing a game-theoretic model between a facility provider and multiple users for a simplified scheduling scenario based on job execution times. We present closed-form expressions for the provider's and user's best-response strategies to maximize their respective utility functions. We describe conditions under which offering a grace-period is advantageous to both facility provider and users by deriving the Nash equilibrium of the game.**

***Index Terms*—grace-period, job completion times, under- and over-requested time, game theory, supercomputers**

I. Introduction

Supercomputing centers typically support batch scheduling of computing jobs wherein users submit jobs with specified execution times. These times are used by the facility providers to schedule the jobs by adding them to the batch queues. Typically, jobs with smaller specified times receive a higher priority, and also are more likely to fit within the gaps between larger jobs and hence are likely to be scheduled for an earlier execution as a part of backfilling [7].

Jobs that run past their specified times are terminated according to a specified policy, which often involves waiting for a grace period as practiced at Argonne Leadership Computing Center (ALCF) [1] and National Energy Research Scientific Computing Center (NERSC) [2]. However, it is not always possible to accurately predict the execution times of jobs [3], [11], and it is reported that users' requested job execution times were usually inaccurate [4], [8], [10]. For instance, Mu'alem *et al.* [8] found from three datasets that for jobs which ran to completion, most of them were evenly distributed in using 5% to 95% of their requested job execution times, that is, most of them overestimated the job time. Meanwhile, the portion of users who underestimated the execution times of their jobs is found to vary from 4.4% to 12.6%, depending on the dataset used. Typically, users are faced with two conflicting choices: overstate the execution times to ensure job completion at the risk of lower priority and hence longer turnaround time, or understate them to ensure higher priority at the risk of premature termination.

The performance of supercomputers is typically measured by the efficiency of resource utilization including the use of computational resources and running times of successfully completed jobs. Ideally, the facility is fully utilized and all jobs are successfully completed without terminations. While job completions benefit both users and the provider, their goals are somewhat orthogonal: users want to minimize the job completion times whereas the provider wants to maximize facility utilization by completed jobs. Meanwhile, job terminations negatively affect both, and they can be reduced by allowing jobs to run past the allocated times. Thus, one consideration of job scheduling policy in supercomputers is whether to offer grace period for users' submitted jobs.

This paper investigates whether offering grace period is beneficial to supercomputers' performance by developing a simple game-theoretic model between a facility provider and multiple users. The conditions under which offering grace period is optimal are explored under a simple scenario of scheduling a single computing system based on job execution times. This task entails utilizing users' strategy information to avoid rewarding users that intentionally underestimate the job execution times, while accommodating compliant users with inaccurate estimates. Here, the users' strategy entails requesting execution time by taking into account the grace period, which may or may not be disclosed by the provider.

It is natural to apply game-theoretic models to study the interactions between the users and facility provider on job scheduling, as they both need to fuse the information about the other in determining their strategies. Indeed, game theory has been applied to study various scheduling aspects. Sedighi

This work is funded by the RAMSES project, Office of Advanced Computing Research, U.S. Department of Energy, and performed at Oak Ridge National Laboratory managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725; and partially supported by grant UGC/FDS14/E01/19 from the Research Grants Council of the Hong Kong Special Administrative Region.

et al. [9] modeled the interactions between users in a HPC environment as a zero-sum game while the scheduler acts as the mediator. In the paper, however, they assumed that each player wants to obtain as many resources as possible instead of having a specific number of resources necessary to have the job completes. Feldman and Tamir [6] studied how coalition formation affects the performance of NE strategies in job scheduling systems where n jobs are assigned to m identical machines and incur a cost which is equal to the total load on the machine they are assigned to. The authors showed that in the job scheduling problem, NE schedules provide approximate stability against coalitional deviations, and a subclass of NE schedules produced by the Longest Processing Time (LPT) rule give better stability towards such deviations. Czumaj *et al.* [5] studied the routing problem of serving n data streams by m servers, which is similar to the scheduling problem above except for the cost function considered, i.e., waiting or service time is used instead of the loading at the machine. The authors studied the price of selfish routing in non-cooperative networks, which is defined as the difference in cost between the worst possible Nash equilibrium and the social optimum. In particular, one of the cases investigated is queuing under heterogeneous traffic in which data streams are of different sizes, and the authors showed that the ratio between the cost of the worst possible Nash equilibrium and the social optimum could be unbounded if the servers have to accept all the requests.

In this paper, we formulate a game-theoretic model for a simplified scenario wherein N jobs are scheduled based on the users' specified job execution times, and those exceeding grace period w are terminated. The provider's utility is based on the facility utilization and a linear combination of terms corresponding to job execution times and job terminations, which is optimized by choosing the grace period w for the facility. The utility function of a user is based on a reward term for job completion time and linear terms corresponding to job completions and terminations, which is optimized by choosing the requested job execution time Q_n for job n. We derive closed-form expressions for both optimal provider and user strategies, namely, w and Q_n, and algorithms with complexity $O(N^2)$ and $O(N)$, respectively, to compute them. We also provide the conditions under which grace period is optimal by deriving the Nash equilibrium of the game.

This paper is organized as follows. The game-theoretic problem of scheduling under grace period is formulated in Section II. The best-response strategies for users and provider are derived and shown to be computable with polynomial time complexity in Sections II-A and II-B, respectively. Nash equilibrium conditions are presented in Section III. Conclusions are presented in Section IV.

II. Game-theoretic modeling of grace period policy

The impact of grace period policy on the performance of supercomputers not only is a facility provider's decision but also relies on users' behavior on requested job time. The more inaccurately estimated job execution times will likely decrease supercomputers' resource utilization efficiency more. Consider a simultaneous game with complete information between a facility provider and a user, in which user knows whether the supercomputer offers grace period. From the user's perspective, the requested time will differ in response to whether grace period is offered. Usually users underestimate job time when grace period is offered and overestimate job time without grace period. User's estimated time likely deviates from actual job time. Let

$$\sigma_n = \frac{Q_n - J_n}{J_n}$$

be the ratio of time over-estimated or under-estimated to actual job time, $\sigma_n \in (-1, \sigma_{max}]$. Here $Q_n > 0$ is user n's requested job time, J_n is user n's actual job time, and σ_{max} is the maximum ratio of time estimation deviation to actual job time. $\sigma_n < 0$ when user n underestimates the job time, and $\sigma_n > 0$ when user n overestimates the job time. Let w_n be the ratio of grace period to user n's requested job time and w be the grace period. Specifically, $w_n = \frac{w}{Q_n} = \frac{w}{J_n(1+\sigma_n)}$, and $w \in [0, w_{\max}]$.

When the job from user n is completed successfully without early termination, we have $Q_n - J_n + w \geq 0$. Hence, user n's job is completed when $w_n \geq -\frac{\sigma_n}{\sigma_n+1}$, or equivalently, $\sigma_n \geq -\frac{w_n}{w_n+1}$, and is terminated otherwise.

Figure 1 shows the grace period ratio w_n correlates non-linearly to the requested job time deviation σ_n.

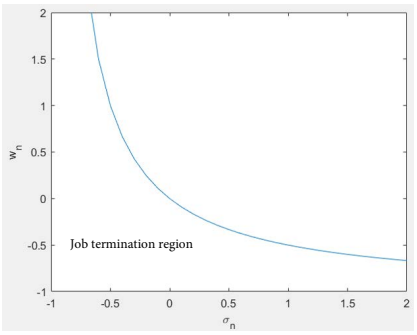

Fig. 1. Job termination determined by correlation between σ_n and w_n

σ_n	-50%	-20%	0	10%	20%	30%	40%
w_n	100%	25%	0	-9.09%	-16.67%	-23.08%	-28.57%

TABLE I
Relationship between σ_n and w_n for successful job completion

Table I shows the relationship between σ_n and w_n when the requested job time is moderated by the grace period to be the actual job time, i.e., the job is completed exactly by the requested time plus the grace period. It shows that when a user under-requests 20% of actual job time, the facility provider should give a grace period of 25% of underestimated time. When the user over-requests 10% of actual job time, the provider needs to lower down the user's requested time by 9.09% to complete the job exactly, which is not done in practice because $w \geq 0$.

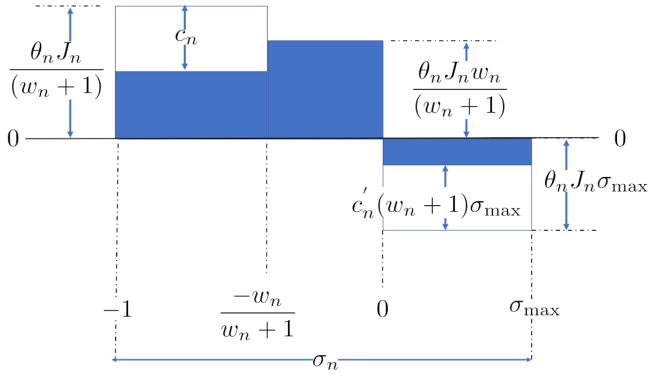

Fig. 2. User's response regions $r_n(.)$ for different σ_n ranges.

User underestimates job time (i.e., $\sigma_n < 0$) without grace period and overestimates ($\sigma_n > 0$) when grace period is offered. Either over-estimated or under-estimated job time provides users benefit and facility provider loss. Assume the loss due to job termination is linearly proportional to the amount of job time deviation. In the following we develop game-theoretic models between a facility provider and users under different grace period policies.

A. User's utility model

A user's utility is evaluated based on whether a job is successfully completed and the benefit from requested job time deviation.

$$U_u(\sigma_n|w,J_n) = \begin{cases} r_n(\sigma_n,w,J_n) - \theta_n J_n \sigma_n + \\ \quad c_n \min(\sigma_n + w_n + w_n\sigma_n, 0) & \text{if } -1 < \sigma_n \le 0 \\ r_n(\sigma_n,w,J_n) + \theta_n J_n \sigma_n - \\ \quad c_n^{'} \max(\sigma_n + w_n + w_n\sigma_n, 0) & \text{if } 0 < \sigma_n \le \sigma_{max} \end{cases}$$

where $r_n(\cdot)$ is user n's reward function for job completion; θ_n is user n's earnest level of job completion time for under-estimating job time, and valuation of successful job completion for over estimation; c_n is the loss due to job termination; $c_n^{'}$ is the loss from low priority in the queue due to overestimation of requested job time. When a job time is under-requested ($\sigma_n < 0$), the job is terminated when $J_n > Q_n + w$, i.e., $J_n > Q_n + w_n J_n(1+\sigma_n)$, then $\sigma_n + w_n + w_n\sigma_n < 0$, i.e., $w_n < -\frac{\sigma_n}{\sigma_n+1}$. When a job time is over-requested ($\sigma_n > 0$), loss from low priority in the queue occurs when $w_n > -\frac{\sigma_n}{\sigma_n+1}$.

User's best-response condition can be obtained as follows.

$$\frac{dr_n(\cdot)}{d\sigma_n} = \begin{cases} \theta_n J_n - c_n(w_n+1) & \text{if } -1 < \sigma_n < -\frac{w_n}{w_n+1} \\ \theta_n J_n & \text{if } -\frac{w_n}{w_n+1} \le \sigma_n \le 0 \\ -\theta_n J_n + c_n^{'}(w_n+1) & \text{if } 0 < \sigma_n \le \sigma_{\max} \end{cases}$$

Therefore, user's best response yields the following reward function.

$$r_n(\cdot) = \begin{cases} \frac{\theta_n J_n}{w_n+1} - c_n & \text{if } -1 < \sigma_n < -\frac{w_n}{w_n+1} \\ \theta_n J_n \frac{w_n}{w_n+1} & \text{if } -\frac{w_n}{w_n+1} \le \sigma_n < 0 \\ [-\theta_n J_n + c_n^{'}(w_n+1)]\sigma_{\max} & \text{if } 0 \le \sigma_n \le \sigma_{\max} \end{cases}$$

This response is specified by closed-form expressions in three separate regions for ratio σ_n of user n's job as shown in Figure 2, which represent three fixed values. When the user overestimates, the best-response is given by a single expression

$$[c_n^{'}(w_n+1) - \theta_n J_n]\sigma_{\max} = c_n^{'}\frac{(w+Q_n)}{Q_n}\sigma_{\max} - \theta_n J_n \sigma_{\max},$$

which can be used to derive an expression for $U_u(\sigma_n|w,J_n)$ in terms of Q_n. Then, this can be ensured to be larger than a given quantity by solving the resultant quadratic equation for Q_n.

When the user underestimates, the response is specified in two different regions:

- Under the condition $-1 < \sigma_n < -\frac{w_n}{w_n+1}$ that results in job termination, we have the best response expression $\frac{\theta_n J_n Q_n}{(w+Q_n)} - C_n$ that can be used to estimate $U_u(\sigma_n|w,J_n)$ in terms of Q_n. The resultant quadratic equation in terms of Q_n can be solved to obtain its best strategic estimate.
- Under the condition $-\frac{w_n}{w_n+1} \le \sigma_n \le 0$ that ensures job completion, the response is given by $\frac{\theta_n J_n w}{(w+Q_n)}$, which can be used to estimate $U_u(\sigma_n|w,J_n)$. Again, this can be ensured to be larger than a given quantity by solving the resultant quadratic equation for Q_n.

Based on $w_n = \frac{w}{J_n(\sigma_n+1)}$, user's best response $\hat{r}_n(\sigma_n|w,J_n)$ can be represented in the following form.

$$r_n(\cdot) = \begin{cases} \frac{\theta_n J_n^2(\sigma_n+1)}{w+J_n(\sigma_n+1)} - c_n, & \text{if } \sigma_n < -\frac{w}{J_n} \\ \frac{\theta_n J_n w}{w+J_n(1+\sigma_n)}, & \text{if } -\frac{w}{J_n} \le \sigma_n < 0 \\ \Big[-\theta_n J_n + c_n^{'}\big(\frac{w}{J_n(\sigma_n+1)}+1\big)\Big]\sigma_{\max}, & \text{if } 0 \le \sigma_n \le \sigma_{\max} \end{cases}$$

Under complete knowledge, including w and J_n, a user can compute the utility in each of the regions and pick Q_n to lie in region with the highest utility; this algorithm has $O(N)$ complexity for all jobs. In practice, however, w may not be disclosed by the provider and an accurate estimation of J_n remains a challenge. In such cases, the utility estimates in the individual regions can be used to estimate bounds on the errors due to the inaccurate estimates of w and J_n.

B. Facility provider's utility model

When a facility provider offers grace period to jobs exceeding requested time, his utility is evaluated as the reward from job completion subtracted by cost of inaccurate requested time and job termination.

$$U_p(w|\sigma,J) = R(w,\sigma,J) - \sum_{n=1}^{N}\Big[c_e^{'}\max\{J_n\sigma_n,0\} + c_e^{''}\max\{-J_n\sigma_n,0\} + c_p\max\{-w - J_n\sigma_n,0\}\Big]$$

where $R(\cdot)$ is the provider's reward and $c_e^{'}, c_e^{''}$ and c_p are unit loss of inaccurately requested job time. Specifically, $c_e^{'}$ is the unit loss due to over-requested job time, $c_e^{''}$ is the unit loss

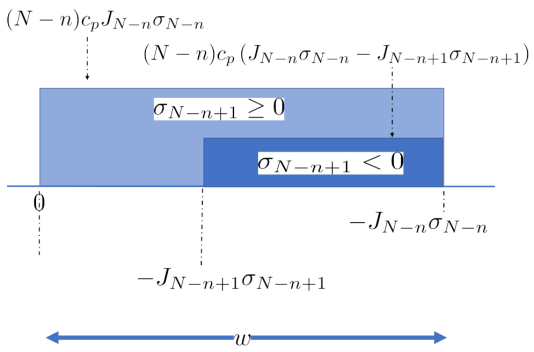

Fig. 3. Provider's response region $R(\cdot)$ for $\sigma_{N-n}<0$ for different w ranges.

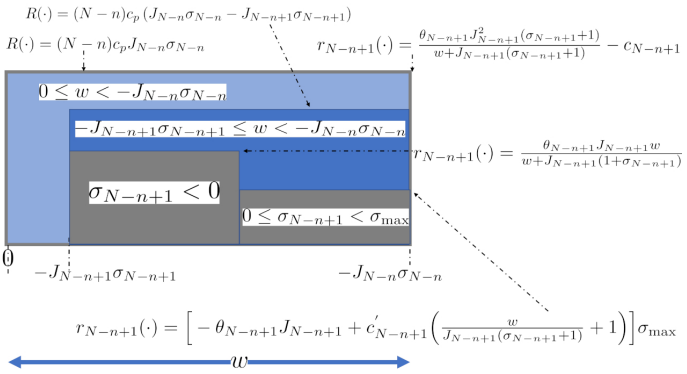

Fig. 4. Provider's and users response regions $R(\cdot)$ for different w ranges at Nash equilibrium.

due to under-requested job time, and c_p is the loss due to job termination.

Provider's best response is derived as follows.

$$\frac{dR(\cdot)}{dw}=$$
$$\begin{cases}-Nc_p & \text{if } 0\leq w<-J_N\sigma_N\\ -(N-1)c_p & \text{if } \max(-J_N\sigma_N,0)\leq w<-J_{N-1}\sigma_{N-1}\\ \quad\cdots & \\ -(N-n)c_p & \text{if } \max(-J_{N-n+1}\sigma_{N-n+1},0)\leq w\\ & \qquad\qquad<-J_{N-n}\sigma_{N-n}\\ \quad\cdots & \\ 0 & \text{if } \max(-J_1\sigma_1,0)\leq w\leq w_{\max}\end{cases}$$

where $J_1\sigma_1<J_2\sigma_2<\cdots<J_N\sigma_N$ is the order statistic of users' requested time deviation from actual ones. Therefore, the provider's best response satisfies the following condition.

$$R(\cdot)=$$
$$\begin{cases}Nc_pJ_N\sigma_N\\ \quad\text{if } 0\leq w<-J_N\sigma_N,\sigma_N<0\\ (N-1)c_p(J_{N-1}\sigma_{N-1}-J_N\sigma_N)\\ \quad\text{if } -J_N\sigma_N\leq w\leq -J_{N-1}\sigma_{N-1},\sigma_N<0\\ (N-1)c_pJ_{N-1}\sigma_{N-1}\\ \quad\text{if } 0\leq w\leq -J_{N-1}\sigma_{N-1},\sigma_N\geq 0\\ \quad\cdots\\ (N-n)c_p(J_{N-n}\sigma_{N-n}-J_{N-n+1}\sigma_{N-n+1})\\ \quad\text{if } -J_{N-n+1}\sigma_{N-n+1}\leq w\leq -J_{N-n}\sigma_{N-n},\\ \qquad\sigma_{N-n+1}<0\\ (N-n)c_pJ_{N-n}\sigma_{N-n}\\ \quad\text{if } 0\leq w\leq -J_{N-n}\sigma_{N-n},\sigma_{N-n+1}\geq 0,\\ \quad\cdots\\ w_{\max}+J_1\sigma_1\\ \quad\text{if } -J_1\sigma_1\leq w\leq w_{\max},\sigma_1<0\\ w_{\max}\\ \quad\text{if } 0\leq w\leq w_{\max},\sigma_1\geq 0\end{cases}$$

This set of equations provide $N+1$ values for $R(\cdot)$ for the corresponding intervals of w; the interval n exists based on condition $\sigma_{N-n}<0$, as illustrated in Fig. 3. This interval has the right hand limit of $-J_{N-n}\sigma_{N-n}$, while its value of $R(\cdot)$ and its left limit depends on σ_{N-n+1}:

(i) under condition $\sigma_{N-n+1}\geq 0$, its value is

$$(N-n)c_pJ_{N-n}\sigma_{N-n}$$

in the interval $[-J_{N-n+1}\sigma_{N-n+1}:-J_{N-n}\sigma_{N-n}]$ and 0 elsewhere; and

(ii) under condition $\sigma_{N-n+1}<0$, its value is

$$(N-n)c_p\left(J_{N-n}\sigma_{N-n}-J_{N-n+1}\sigma_{N-n+1}\right)$$

in the interval $[0:-J_{N-n}\sigma_{N-n}]$ and zero elsewhere.

In each interval, $U_p(w|\sigma,J)$ can be estimated based on the execution times of the jobs, and w can be selected as any value within the interval with the highest utility value. The time complexity of this algorithm is $O(N^2)$ since computation of $U_p(w|\sigma,J)$ in each region has $O(N)$ complexity. Unlike the utility of the users, the underlying quantities are all known to the provider.

III. Nash Equilibrium Conditions

When both users and facility provider play their best responses as shown in Sections II-A and II-B, respectively, the interactions reach Nash equilibrium specified by the following

conditions:

$$
\begin{aligned}
&r_N(\cdot) = \frac{\theta_N J_N^2(\sigma_N+1)}{w + J_N(\sigma_N+1)} - c_N,\\
&R(\cdot) = N c_p J_N \sigma_N,\\
&\qquad \text{if } 0 \le w < -J_N\sigma_N, \sigma_N < 0\\
&r_N(\cdot) = \frac{\theta_N J_N w}{w + J_N(\sigma_N+1)},\\
&R(\cdot) = (N-1)c_p(J_{N-1}\sigma_{N-1} - J_N\sigma_N),\\
&\qquad \text{if } -J_N\sigma_N \le w < -J_{N-1}\sigma_{N-1}, \sigma_N < 0\\
&r_N(\cdot) = \Big[-\theta_N J_N + c_N^{'}\big(\frac{w}{J_N(\sigma_N+1)}+1\big)\Big]\sigma_{\max},\\
&R(\cdot) = (N-1)c_p J_{N-1}\sigma_{N-1},\\
&\qquad \text{if } 0 \le w < -J_{N-1}\sigma_{N-1}, 0 \le \sigma_N \le \sigma_{\max}\\
&\quad \cdots\\
&r_{N-n+1}(\cdot) = \frac{\theta_{N-n+1} J_{N-n+1}^2(\sigma_{N-n+1}+1)}{w + J_{N-n+1}(\sigma_{N-n+1}+1)} - c_{N-n+1},\\
&R(\cdot) = (N-n+1)c_p(J_{N-n+1}\sigma_{N-n+1} - J_{N-n+2}\sigma_{N-n+2}),\\
&\qquad \text{if } -J_{N-n+2}\sigma_{N-n+2} \le w < -J_{N-n+1}\sigma_{N-n+1},\\
&\qquad\quad \sigma_{N-n+2} < 0\\
&r_{N-n+1}(\cdot) = \frac{\theta_{N-n+1} J_{N-n+1}^2(\sigma_{N-n+1}+1)}{w + J_{N-n+1}(\sigma_{N-n+1}+1)} - c_{N-n+1},\\
&R(\cdot) = (N-n+1)c_p J_{N-n+1}\sigma_{N-n+1},\\
&\qquad \text{if } 0 \le w < -J_{N-n+1}\sigma_{N-n+1}, 0 \le \sigma_{N-n+2} \le \sigma_{\max}\\
&r_{N-n+1}(\cdot) = \frac{\theta_{N-n+1} J_{N-n+1} w}{w + J_{N-n+1}(\sigma_{N-n+1}+1)},\\
&R(\cdot) = (N-n)c_p(J_{N-n}\sigma_{N-n} - J_{N-n+1}\sigma_{N-n+1}),\\
&\qquad \text{if } -J_{N-n+1}\sigma_{N-n+1} \le w < -J_{N-n}\sigma_{N-n},\\
&\qquad\quad \sigma_{N-n+1} < 0\\
&r_{N-n+1}(\cdot) = \Big[-\theta_{N-n+1}J_{N-n+1} +\\
&\qquad\quad c_{N-n+1}^{'}\Big(\frac{w}{J_{N-n+1}(\sigma_{N-n+1}+1)}+1\Big)\Big]\sigma_{\max},\\
&R(\cdot) = (N-n)c_p J_{N-n}\sigma_{N-n},\\
&\qquad \text{if } 0 \le w < -J_{N-n}\sigma_{N-n}, 0 \le \sigma_{N-n+1} \le \sigma_{\max}\\
&\quad \cdots\\
&r_1(\cdot) = \frac{\theta_1 J_1^2(\sigma_1+1)}{w + J_1(\sigma_1+1)} - c_1,\\
&R(\cdot) = c_p J_1 \sigma_1,\\
&\qquad \text{if } 0 \le w < -J_1\sigma_1, 0 \le \sigma_2 \le \sigma_{\max}\\
&r_1(\cdot) = \frac{\theta_1 J_1 w}{w + J_1(\sigma_1+1)},\\
&R(\cdot) = w_{\max} + J_1\sigma_1,\\
&\qquad \text{if } -J_1\sigma_1 \le w \le w_{\max}, \sigma_1 < 0\\
&r_1(\cdot) = \Big[-\theta_1 J_1 + c_1^{'}\big(\frac{w}{J_1(\sigma_1+1)}+1\big)\Big]\sigma_{\max},\\
&R(\cdot) = w_{\max},\\
&\qquad \text{if } 0 \le w \le w_{\max}, 0 \le \sigma_1 \le \sigma_{\max}
\end{aligned}
$$

A generic case of provider and user response regions based on these equations are illustrated in Fig. 4.

The NE condition shows that provider's reward depends on the extent of all users' requested time deviation from actual time ($J_n\sigma_n$), and the grace period (w). Users' reward is determined by the valuation of priority in the waiting queue and job completion (θ_n), under- or over-requested time ratio (σ_n), job actual time (J_n) and provided grace period (w). Consider a low-dimension case in which there are two jobs, $N = 2$. Let $J_1 = J_2 = \theta_1 = \theta_2 = c_1 = c_2 = c_1^{'} = c_2^{'} = c_p = 1$, then the NE condition turns out to be as follows.

$$
\begin{aligned}
&(i)\quad r_1(\cdot) = \frac{\sigma_1+1}{w+\sigma_1+1} - 1, r_2(\cdot) = \frac{\sigma_2+1}{w+\sigma_2+1} - 1,\\
&\qquad R(\cdot) = 2\sigma_2, \text{ if } 0 \le w < -\sigma_2, \sigma_2 < 0;\\
&(ii)\quad r_1(\cdot) = \frac{\sigma_1+1}{w+\sigma_1+1} - 1, r_2(\cdot) = \frac{w}{w+\sigma_2+1},\\
&\qquad R(\cdot) = \sigma_1 - \sigma_2, \text{ if } -\sigma_2 \le w < -\sigma_1, \sigma_2 < 0;\\
&(iii)\quad r_1(\cdot) = \frac{w}{w+\sigma_1+1} - 1, r_2(\cdot) = \frac{w\sigma_{\max}}{\sigma_2+1},\\
&\qquad R(\cdot) = \sigma_1, \text{ if } -\sigma_2 \le w < -\sigma_1, -1 < \sigma_1 < 0,\\
&\qquad\qquad 0 \le \sigma_2 \le \sigma_{\max};\\
&(iv)\quad r_1(\cdot) = \frac{w}{w+\sigma_1+1}, r_2(\cdot) = \frac{w}{w+\sigma_2+1},\\
&\qquad R(\cdot) = w_{\max} + \sigma_1, \text{ if } -\sigma_1 \le w \le \sigma_{\max},\\
&\qquad\qquad -1 < \sigma_1 < 0, \sigma_2 < 0;\\
&(v)\quad r_1(\cdot) = \frac{w}{w+\sigma_1+1}, r_2(\cdot) = \frac{w\sigma_{\max}}{\sigma_2+1},\\
&\qquad R(\cdot) = w_{\max} + \sigma_1, \text{ if } -\sigma_1 \le w \le \sigma_{\max},\\
&\qquad\qquad 0 \le \sigma_2 < \sigma_{\max};\\
&(vi)\quad r_1(\cdot) = \frac{w\sigma_{\max}}{\sigma_1+1}, r_2(\cdot) = \frac{w\sigma_{\max}}{\sigma_2+1},\\
&\qquad R(\cdot) = w_{\max}, \text{ if } 0 \le w \le w_{\max}, 0 \le \sigma_1 \le \sigma_{\max}.
\end{aligned}
$$

When grace period w is shorter than both users' under-requested amount of time (case i), provider's reward equals to twice of the shorter under-requested job time; and users' reward is negatively correlated to the length of grace period. When grace period is longer than the user with smaller under-requested time but shorter than that of the other user, and both users under-request job time (case ii), provider's reward equals to the difference of requested deviation time ratios. User's reward with shorter deviated requested time (r_2) is positively correlated to grace period w and user with longer deviated requested time (r_1) is negatively correlated to w. When one user under-requests job time and the other over-requests job time (case iii) both users' rewards are positively correlated to grace period w. Provider's reward equals to the under-requested job time. When grace period is longer than both users' under-requested time (case iv), provider's reward is determined by the summation of shorter deviated request time and the maximum possible grace period. In this case, users' reward are positively correlated to grace period w. When one user under-requests job time and the other over-requests time, while the grace period is longer than the under-requested job

time (case v), both users' rewards are positively correlated to the length of grace period. When both users over-request job time (case vi), provider's reward equals to the maximum possible grace period time. Users' rewards reach maximum level of $\frac{w\sigma_{\max}}{\sigma_n+1}, n = 1, 2$.

Therefore, facility provider is suggested to provide shorter or no grace period, i.e., $0 \leq w < -\sigma_2$ when $\sigma_2 > \max\{-\sigma_1, \frac{\sigma_1+w_{\max}}{2}, \frac{w_{\max}}{2}\}$; relatively small grace period $-\sigma_2 \leq w < -\sigma_1$ when $\sigma_1 < \min\{\frac{\sigma_2}{3}, -w_{\max}, \sigma_1 - w_{\max}\}$; relatively large grace period $-\sigma_1 \leq w \leq \sigma_{\max}$ when $-w_{\max} < \sigma_2 < \frac{1}{2}(\sigma_1 + w_{\max})$ and at least one user over-requests job time $\sigma_1 > 0$; and large grace period $0 \leq w \leq \sigma_{\max}$ when $\sigma_1 - w_{\max} < \sigma_2 < \frac{1}{2}w_{\max}$ and at least one user under request job time.

In general, the grace period policy that can maximize provider's reward depends on the distribution of deviation of users' under- or over-requested job time from actual time (σ_n), the maximum grace period that the facility provider is willing to provide ($w_{\max}$), user's earnest level of job completion and valuation of successful job completion (θ_n), user's valuation of loss from job termination and low priority in the waiting queue ($c_n, c_n^{'}$), the actual job time J_n, and provider's valuation of job termination c_p.

IV. Conclusions

In this paper, we investigated whether providing grace-period for job executions on the supercomputing facility performance can benefit facility provider and users, an issue that is overlooked in analytical approaches in the literature. We performed the study by developing game-theoretic models between the facility provider and n users for a simplified scheduling scenario in which provider determines grace period length and users determine the under- or over-requested job time. We derived the best responses and Nash Equilibrium of the game, and showed that the facility provider and n users could maximize their utilities by implementing different grace period policies. The grace-period policy is affected by the distribution of deviation of users' under- or over-requested job time from actual time, the maximum grace period that the facility provider is willing to provide, user's valuation of waiting time and successful completion of job, and provider's valuation of job termination.

Future work may consider a finer facility model by accounting for the number of processing units such as CPUs and GPUs as a part of user's request. Here, jobs with smaller requests can be back-filled into existing allocations, and it would be of future interest to consider impacts of grace-period on user and provider strategies. It would of future interest to develop methods for users to estimate w when it is not disclosed by the provider by strategically selecting Q_n values of set of jobs to ensure a combination of completions and terminations.

References

[1] Argonne leadership computing facility.

[2] National energy research scientific computing center.

[3] C. Bailey Lee, Y. Schwartzman, J. Hardy, and A. Snavely. Are user runtime estimates inherently inaccurate? In D. G. Feitelson, L. Rudolph, and U. Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, pages 253–263, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[4] S.-H. Chiang and M. K. Vernon. Characteristics of a large shared memory production workload. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 159–187, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[5] A. Czumaj, P. Krysta, and B. Vöcking. Selfish traffic allocation for server farms. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 287–296, New York, NY, USA, 2002. Association for Computing Machinery.

[6] M. Feldman and T. Tamir. Approximate strong equilibrium in job scheduling games. In B. Monien and U.-P. Schroeder, editors, *Algorithmic Game Theory*, pages 58–69, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[7] D. A. Lifka. The anl/ibm sp scheduling system. In D. G. Feitelson and L. Rudolph, editors, *Job Scheduling Strategies for Parallel Processing*, pages 295–303, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[8] A. W. Mu'alem and D. G. Feitelson. Utilization, predictability, workloads, and user runtime estimates in scheduling the ibm sp2 with backfilling. *IEEE Transactions on Parallel and Distributed Systems*, 12(6):529–543, 2001.

[9] A. Sedighi, Y. Deng, and P. Zhang. Fairness of task scheduling in high performance computing environments. *Scalable Computing: Practice and Experience*, 15:273–285, Sept 2014.

[10] W. Tang, N. Desai, D. Buettner, and Z. Lan. Job scheduling with adjusted runtime estimates on production supercomputers. *Journal of Parallel and Distributed Computing*, 73(7):926–938, 2013.

[11] D. Tsafrir, Y. Etsion, and D. G. Feitelson. Modeling user runtime estimates. In D. Feitelson, E. Frachtenberg, L. Rudolph, and U. Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, pages 1–35, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.