Saving Power Cost of HPC Clusters through A Power Aware Job Scheduling Algorithm

B Yao, R Parpart, A Nickolich, HB Runesha

Abstract

The electricity bill of HPC Cluster is a major component of the cost of operating a data center. Here, we propose a novel job schecheduling algorithm that takes into account dynamic electricity price and the variation in the job's Power Usage Rate (PUR) and compared the performace our algorithm to an existing algorithm. The advantage of our algorithm is that (1) we uses historical electricity price to predict price in the next 24 hours dynamically instead of assuming a static deterministic prediction; (2) we uses Switched Cabinet Distribution Unit to measure PUR for different types of jobs accurately. We show that given a list of job with predetermined PUR and estimated runtime, our algorithm schedules high PUR jobs to run when electricity is relatively low, thus reducing the total cost.

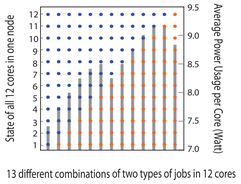
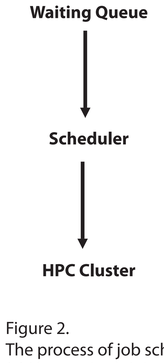


Figure 1.
Measure Power Usage Rate of two types of jobs (namd and hpcg). We choose namd and hpcg jobs because they are among the most common jobs running in Midway HPC cluster. Each column is one possible state of the node. In the first column, all the nodes run namd jobs and the power usage per core is about 7.4 Watt higher than the sleeping state. This number is used as the PUR for namd jobs.
We also noticed that when the node has both jobs running simultaneously, the resulting power usage is not a linear combination of the PUR of namd and hpcg. This means that different cores in one node are correlated and affect each other's power usage. Given this relationship, we can calculate power usage of the node just from the combination.



	FCFS (First Come First Served)	BPP (Binary Price Prediction) in Yang et al, 2013	HPP (Historical Price Prediction) in our algorithm
Schedule Algorithm:			
Job Priority Formula:	$F_{j,t} = A_{j,t}$	$F_{j,t} = A_{j,t} + w_t \cdot R_j$	$F_{j,t} = A_{j,t} + w_t \cdot R_j \cdot P_j$

$A_{j,t}$ Age of job j at time t

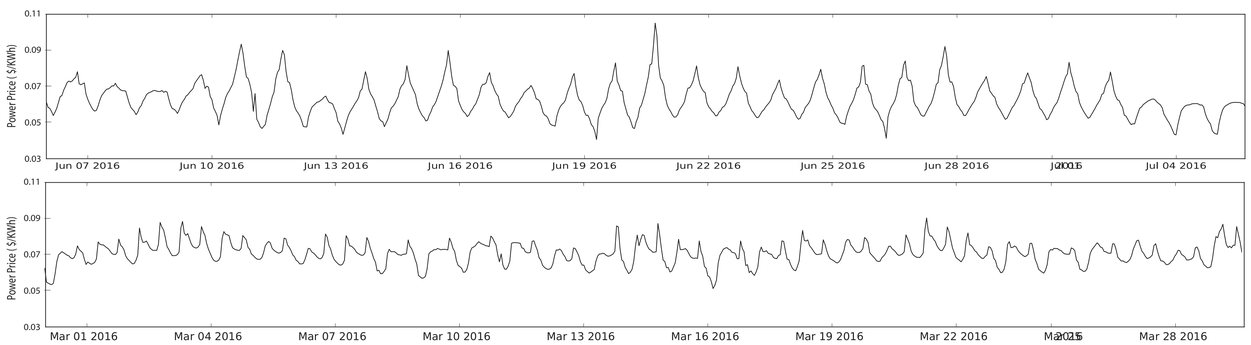
R_j Power Usage Rate(PUR) of job j

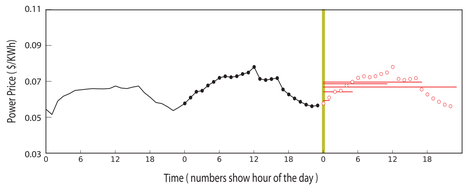
D_j Estimated Runtime/Duration of job j

P_j Averaged electricity price during runtime of job j, if job j starts immediately

w_t Weight of PUR.
Positive if price at time t is lower than average.
Negative if price at time t is higher than average.

Figure 2.
The process of job scheduling in HPC clusters.
(1) Users submit jobs to waiting queue.
(2) The scheduler uses a formula to calculate a priority factor for each job at current time $F_{j,t}$ and dispatches the jobs with highest priority values to available compute nodes.
Different scheduler uses different formula to calculate the priority factor.
(3) The compute nodes execute the job and becomes unavailable untill the current job finishes.





$$P_j = \frac{1}{D_j} \int_t^{t+D_j} p_x \, dx$$

p_x Electricity price at time x

Figure 3.
(top) Historical electricity price shows periodicity. The period is about 24 hours.
(bottom) Price Prediction in our algorithm: Since the price in the next 24 hours is similar to the price in the last 24 hours, we update our price prediction for the next 24 hours every hour by removing the oldes price and adding the price in the past hour that just became available. Yellow vertical line indicates current time. Black line shows knownhistorical price. Black dots show the price in the last 24 hours used to predict price in the next 24 hours indicated by red circles. Using this price prediction we can calculate the average electricity price of jobs with different estimated runtime if the job starts immediately. The length of the horizontal red lines shows runtime of different jobs; their height shows the average electricity price during runtime.

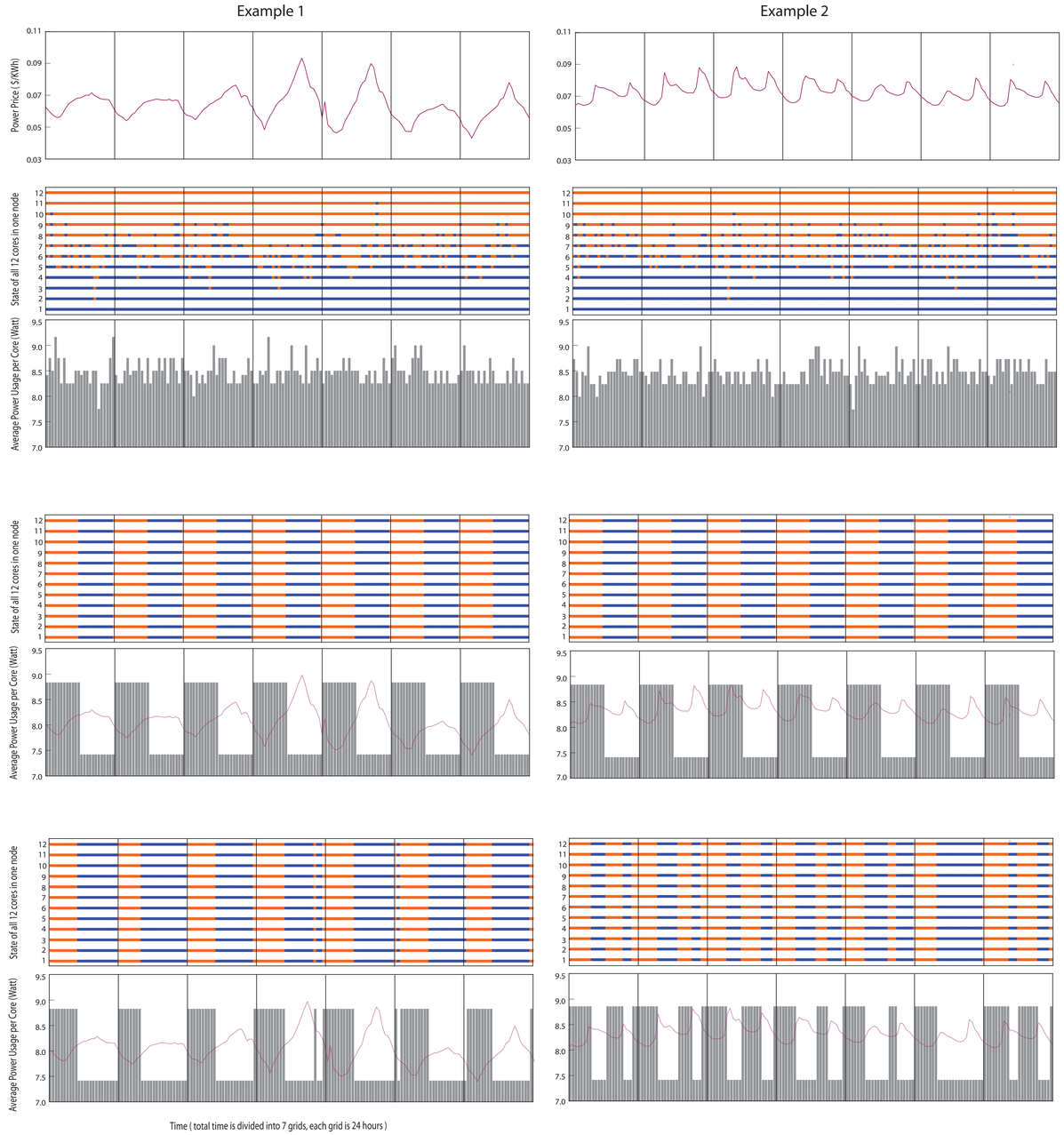


Figure 4.
Two examples of scheduling a list of namd and hpcg jobs. The first panel shows the electricity price. The next three panels show how jobs are scheduled with three different algorithms: FCFS, BPP, HPP. For each algorithm, the figure shows (1) the type of job running in the 12 cores at every time point; (2) power usage per core for the whole node given the combination of namd and hpcg jobs running in the node at every time point.