An energy-aware job scheduling method supporting on-demand job execution

Daiki Nakai
Graduate School of Information Sciences, Tohoku University
Sendai, Miyagi, Japan

Keichi Takahashi
Cyberscience Center, Tohoku University
Sendai, Miyagi, Japan

Yoichi Shimomura
Cyberscience Center, Tohoku University
Sendai, Miyagi, Japan

Hiroyuki Takizawa
Cyberscience Center, Tohoku University
Sendai, Miyagi, Japan

1 INTRODUCTION

Recently, high performance computing (HPC) systems are not only used for traditional batch jobs but also for on-demand jobs with deadlines. However, it is not affordable to always keep a whole HPC system available only for executing on-demand jobs. On the other hand, a publicly shared HPC system may be busy for executing other jobs and thus its computing resources are unavailable when on-demand job execution is requested. In such a case, one or more running jobs must be suspended to make computing resources available to the on-demand job.

An HPC system usually has a power management mechanism, and thus a node could be powered off or turned into a low-power mode based on the operation policy of the system [3]. Hence, there are two potential reasons that a node is unavailable for on-demand job execution upon the job request. One is that the node is running another job, and the other is that the node is in a low-power mode or powered off, referred to as sleeping.

2 APPROACH

This work proposes a job scheduling method for selecting whether to suspend running jobs or to start up sleeping nodes when a sufficient amount of resource is unavailable to complete an on-demand job in time. The goal of this work is to achieve power-efficient job scheduling while completing on-demand job execution by the deadline. In addition, by estimating the time required to suspend and resume a job are proportional to the memory usage of the jobs, the proposed method considers an appropriate combination of jobs to be suspended to limit the increase in time for suspending the jobs.

The node selection procedure for executing an on-demand job is described as follows. If there are a sufficient number of idle nodes, backfilling is used to immediately execute the on-demand job by overtaking other jobs. If the number of idle nodes is less than the number of nodes requested by the on-demand job, more nodes are required in addition to the idle nodes. In this case, if there is no sleeping node, there is no choice but to suspend a sufficient number of running jobs. However, if there exist sleeping nodes, we need to further consider several options to secure a sufficient number of nodes for on-demand job execution. If there is no margin time to start up a sleeping node until the deadline of the on-demand job, a sufficient number of running jobs must be suspended. If there is margin time to start up a sleeping node, those sleeping nodes are started up.

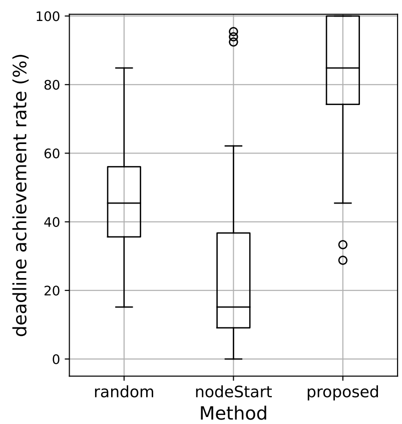

Figure 1: Deadline achievement rate of urgent job

3 EVALUATION

In the following evaluation, the simulation in this work assumes the system configuration of AOBA-A [2] installed at the Cyberscience Center, Tohoku University. We consider two types of on-demand jobs: jobs that are strict deadlines (*e.g.*, urgent jobs for disaster prevention and mitigation [1]) and jobs that can extend their deadline if necessary (*e.g.*, interactive jobs).

As a result of the evaluation, the deadline achievement rate was improved by 39% due to proper selection of running jobs to be suspended for urgent jobs with strict deadlines. In the case of interactive jobs with loose deadlines, node startup was successfully used to improve the execution efficiency of normal jobs by about 18%, and suppress the increase in power consumption by about 0.26% while maintaining the deadline achievement rate of 100%.

REFERENCES

[1] Noriki Uchida, Kazuo Takahata, and Yoshitaka Shibata. 2011. Disaster Information System from Communication Traffic Analysis and Connectivity (Quick Report from Japan Earthquake and Tsunami on March 11th, 2011). *IEEE Access* (2011), 279–285.

[2] Hiroyuki Takizawa, Keichi Takahasi, Yoichi Shimomura, Ryusuke Egawa, Kenji Oizumi, Satoshi Ono, Takeshi Yamashita, and Atsuko Saito. 2023. AOBA: The Most Powerful Vector Supercomputer in the World. In *Sustained Simulation Performance 2022*. Springer Nature.

[3] Tatsuyoshi, Ohmura, Yoichi, Shimomura, Ryusuke, Egawa, and Hiroyuki, Takizawa. 2023. Toward Building a Digital Twin of Job Scheduling and Power Management on an HPC System. In *Job Scheduling Strategies for Parallel Processing*. 47–67.