International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 8 Aug. 2022, pp: 672-676 www.ijaem.net ISSN: 2395-5252

"Improving the performance of job scheduling in heterogeneous cluster using ML"

Mr. Aihtesham N. Kazi, Dr. D.N. Chaudhari
Research Scholar, Dept. Computer Science & Engineering
Dept. Computer Science & Engineering, Jawaharlal Darda Institute of Engineering, Yavatmal, Maharashtra-India

Submitted: 05-08-2022 Revised: 11-08-2022 Accepted: 15-08-2022

ABSTRACT— Task management system is the HPC middleware responsible for distributing computing power to applications. Although these systems generate an increasing amount of data, they are characterized by uncertainties in some parameters, such as the processing time of the task. The question raised in this work is: to what extent is it possible or useful to take into account predictions about the running time of a task and performance improvement in the global scheduling in case of heterogeneous environments. This paper presents a comprehensive study to answer this question, assuming backfill algorithms and scheduling principles. More specifically, we rely on some classical machine learning methods and propose new improved methods well adapted to this type of problem. We then designed a solution through intensive simulations using multiple inputs based on prediction by training the system on production logs of past planning scenarios. Finally, we propose a new scheduling algorithm that outperforms the backfilling scheduling algorithms.

Keywords—Hadoop; Machine learning; HPC; Cluster Scheduling;

I. INTRODUCTION

The efficient use all of the available computing devices is an important issue for heterogeneous computing systems. The ability to choose a CPU or GPU processor for a specific task has a positive impact on the performance of systems. It helps to reduce the total processing time and to achieve the uniform system utilization. We propose a scheduler algorithm that selects the executing device after prior training, based on the size of the input data and predicated execution time of jobs.

Large-scale high performance computing (HPC) platforms are becoming increasingly complex systems as data grows every day.

There exists a broad range and variety of HPC architectures and platforms for handling this big data. Consequently, the job management system (which is the middleware responsible for managing and scheduling jobs) should be tailored to deal with this complexity. By establishing efficient allocation and scheduling strategies that can deal with such complex systems, we accommodate evolutionary strategic and difficult challenges.

More and more data is produced in the systems by platform monitoring (CPU utilization, I/O traffic, power consumption, etc.), task management system (i.e. done) and application-level analytics (various parameters, results and interim results can be used). All of this data is ignored most of the time by real job scheduling systems. Technologies and methods studied in big data (including Machine Learning) could and must be used for job scheduling in new HPC platforms.

II. LITERATURE REVIEW & RELATED WORK

Study has been carried out by the author that stated that by engaging state of art machine learning techniques can speed up the optimization process which could otherwise take many machine to explore the runtime and efficiency of jobs. The complex machine learning methodologies to be applied to solve difficult real world problems using parallel programming models. Parallel machine learning aims to capitalize on the simultaneous execution of sophisticated machine learning algorithms using modern hardware architectures. [3]

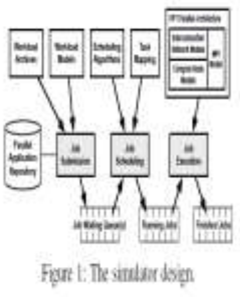
There exists a close connection between machine learning and high performance computing as machine learning algorithms that has been employed to perform auto tuning of multicore based cross platforms for performance optimization

[3] and HPC technologies has been utilized in many big data frameworks to enhance the performance and throughput.

This research tries to identify how machine learning and HPC technologies are employed in corresponding frameworks to provide a comparison of existing auto tuning methodologies. researchers have illustrated that their system was able to train 1 billion parameter networks using only 3 machines and 11 billion parameters using 16 machines in a couple of days to claim the scalability of their system [10]. This study focused on two parallel machine learning algorithms: K means clustering (6 variations) and Multidimensional Scaling (MDS - 2 variations) for the performance analysis.

III. HETEROGENEOUS RESOURCE SCHEDULING

In this work job scheduler simulator, which has been seamlessly incorporated with full-scale HPC system simulator, Prediction Toolkit (PPT) PPT supports rapid assessment and performance prediction of large-scale scientific applications on HPC architectures. This job scheduler simulator can be used to study the performance of various job scheduling and resource provisioning algorithms with sufficiently detailed models for resource contention, job interaction and interference on specific target HPC systems.



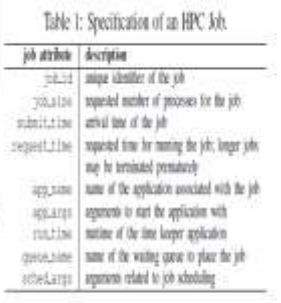


Table 1: Specification of an HPC Job.

job attribute	description
job_id	unique identifier of the job
job_size	requested number of processes for the job
submit_time	arrival time of the job
request_time	requested time for running the job; longer jobs may be terminated prematurely
app_name	name of the application associated with the job
app_args	arguments to start the application with
run_time	runtime of the time keeper application
queue_name	name of the waiting queue to place the job
sched_args	arguments related to job scheduling

Figure 1: The Simulator Design

The heterogeneous computing systems deliver high performance and flexibility while increasing system management complexity. Resource management is particularly affected, because of the variety of computing, memory, and storage resources to handle. This describes Cognitive ScHeduler (CuSH), a resource management and job scheduling framework that leverages deep neural networks and reinforcement learning.

To handle the complexity of heterogeneous resource scheduling, CuSH employs a two-step hierarchical solution. This solution decouples the job selection from the policy selection problem. CuSH has been evaluated using a simulator that is based on experiments executed on an IBM PowerR system. Results show that CuSH outperforms traditional heuristic-based approaches on all the use cases, delivering significantly lower normalized turnaround time.

A. Machine Learning and its types

There are so many types of Machine Learning Techniques and which are used to classify the data sets [1, 2] They are Supervised, Unsupervised Semi-Supervised, Reinforcement, Evolutionary learning, and Deep learning algorithms [3].

a. Supervised learning- It provides a training set of examples with suitable targets, and based on this training set, the algorithms respond correctly to all feasible inputs. Learning from examples is another name for supervised learning [4]. Classification and regression are types of supervised learning.
b. Unsupervised learning- The unsupervised learning technique tries to find similarities between the input data and based on these similarities, the unsupervised learning technique classifies the data. This is also known as density estimation [5]. Unsupervised learning includes clustering, which creates clusters based on similarity.
c. Semi-supervised learning- A semi-supervised learning technique is a class of supervised learning techniques. This learning also used unlabeled data for training purposes (generally a minimal amount of labeled data with a large amount of unlabeled data) [6]. Semi-supervised learning lies between unsupervised learning (unlabeled data) and supervised learning (labeled data).
d. Reinforcement learning- This learning is supported by behaviorist psychology. The algorithm is told when the answer is wrong, but not told how to correct it. He has to research and test different options until he finds the right answers [7]. This is also known as critical learning.
e. Evolutionary Learning- This learning of biological evolution can be thought of as a learning process, such as how biological organisms adapt to achieve advancements in survival rates and chances of having offspring [8]. Using the idea of fitness to check how accurate the solution is, we can run this model on a computer.

f. Deep learning- In this technique, backpropagation jointly learns all model parameters to optimize the output of the task [9]. It uses a deep graph with different layers of processing, made up of many linear and non-linear transformations.

B. Adapted Methodology

The job scheduling module manages job scheduling and task mapping. It keeps track of both the machines and processors occupied by the running jobs and those available for new jobs. When a new job is submitted or when the resources are returned to the system due to the completion of a running job, the job scheduler is invoked. The job scheduling algorithm selects the best candidate among all jobs waiting in the queues and, if enough resource is available, removes the candidate job from the queue and allocates the requested number of machines and processors to run the job. To do that, the selected task mapping algorithm determines the best placement for the job before launching the job's application.

The job trace needed to drive the improved scheduler for predicting the proper job selection that can be created with a workload model, for example, using an exponential distribution for the inter-arrival time of the jobs, a normal distribution for the job size, and yet another for the job's runtime.
The job-level workload logs can be parsed and translated into the trace logs required by machine learning based scheduler.

C. There are multiple ways to use the workload archive:
1. One can directly use the runtime information from the logs for the time keeper application described. In this case, we can use fixed runtime from the logs and we do not model the detailed application runtime behavior.
2. One can use workload models derived from analyzing the logs in the parallel workload archive, for example, by estimating the distribution of job inter-arrival time, job runtime, job size, and so on. In this case, we can use the time keeper application with the estimated runtime as the argument. Again, this method does not include detailed application runtime behavior.In traditional HPC job schedulers, such as IBMc Spectrum LSF2, Slurm3, or FLUX4, job submission includes information regarding the required hardware resources (e.g., CPUs, GPUs, Memory, etc.) and the expected execution time (an upper bound of the actual completion time). The environment simulates an HPC cluster. The simulator returns the execution time of the jobs and takes into account how resource locality affects

D. Collection of DATASET for training

We can collect the execution time of different workloads to create a dataset for a realistic environment simulator. The initial dataset is based on the data gathered from a real HPC system then it can be extended by additional samples with a linear extrapolation of the collected real data. Specifically, the environment simulator calculates the job service time based on the allocation policy (e.g., how we use resources) and the workload types

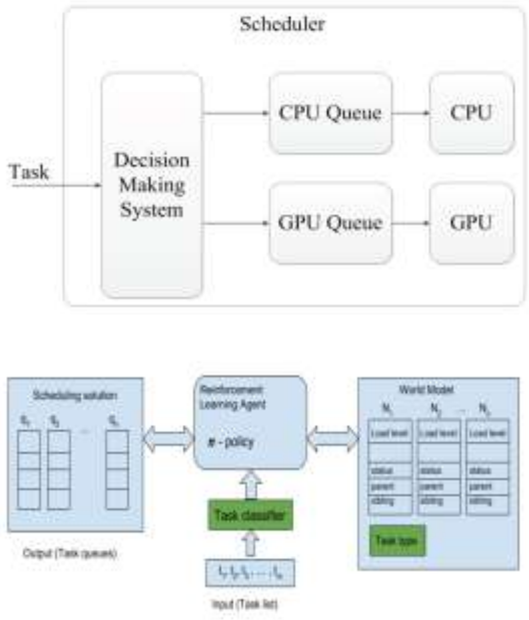


Fig.2 Conceptual Model of scheduling

A new task comes to the scheduler, where static or dynamic scheduling is carried out. Dynamic scheduling of computations is performed by using algorithms of machine learning. As an algorithm for the system responsible for selection of CUs for execution, support vector method (SVM) [10] is used. Since the efficiency of SVM depends on core selection, its parameters, and soft margin value, a core in the form of a radial basis function [11] was selected as the most appropriate for the task of classification between the CPU and GPU. C value is selected during the learning, in order to achieve the highest possible accuracy.

E. Recent Trends in High performance computing and machine learning.

While traditional batch job scheduling still has a place in many businesses, the job is now much more complex than it used to be, thanks to the use of different computing platforms, multiple computing environments, and business processes that run 24/7. Add to that the complex interdependencies that often exist between different processes and the frequent need to run jobs dynamically based on changing circumstances, and it's clear that the requirements of the traditional job scheduler have evolved into a broader workload automation capability to handle today's complex circumstances. [6]

A modern job scheduler must provide a range of features that enable efficient scheduling of distributed jobs and in a manner that is consistent with the business's operational goals.

Planners must be flexible enough to accommodate varying technology, business, and resource requirements. If the task scheduler is able to sequence processes and manage resource contention, applications will run faster and more predictably, and scheduling throughput will increase.

Problem/issue: This means you need to purchase new hardware or set up an additional scheduler or workload system to handle the additional workload. Another important factor is the programming language used to develop the automation engine. For example, engines that are developed in Java often have performance issues under heavy workloads and require additional hardware.

The growth of artificial intelligence (AI) capabilities, data volumes and computing power in recent years means that AI is now a serious consideration for most organisations. Combined with high-performance computing (HPC) capabilities, its potential continues to grow.

AI workloads must then integrate with your existing HPC infrastructure to achieve the best results in the most efficient and cost-effective way. [5]

The key considerations for organizations looking to bring AI into their HPC environment, and steps they can take to ensure the success of their first forays into HPC/AI convergence.

HPC professionals are facing both new challenges and opportunities from the rise of AI. Although research into the use of AI has been around for decades, for a long time the volume of data and computing power needed to derive insights and make strong decisions with it were out of reach. The flood of data that most academic organizations, governments and enterprises generate, along with the significant increase in compute capability and decrease in cost, mean that AI is finally becoming a viable option. We should consider what we mean by 'convergence'. There is no single definition, as AI and HPC can be combined in a number of different ways depending on the organization's needs and use cases.

For example, the following scenarios could all be described as HPC/AI convergence:

a. Introducing and running AI frameworks like Tensor flow on an existing HPC infrastructure.
b. Enhancing existing HPC workloads like simulation and modeling by using an AI engine to analyze output data once a model has been run.

F. Different Interfaces of ML and HPC:

We have identified several important distinctly different links between machine learning (ML) and HPC.

We define two broad categories: HPCforML and MLforHPC

A. HPCforML: Using HPC to execute and enhance ML performance, or using HPC simulations to train ML algorithms (theory guided machine learning), which are then used to understand experimental data or simulations
B. MLforHPC: Using ML to enhance HPC applications and systems. MLautotuning: Using ML to configure (autotune) ML or HPC simulations. Already, autotuning with systems like ATLAS is hugely successful and gives an initial view of MLautotuning. As well as choosing block sizes to improve cache use and vectorization, MLautotuning can also be used for simulation mesh sizes [9] and in big data problems for configuring databases and complex systems like Hadoop and Spark [10], [11].

Cloud computing has recently emerged as a cost effective alternative to dedicated infrastructure for HPC applications. Running an application in cloud avoids the long lead time, high capital expenditure, and large operational costs associated with a dedicated HPC infrastructure.

1. HPC-Aware Clouds

We identify the opportunities and challenges of virtual machine consolidation for HPC in the cloud. In addition, we develop scheduling algorithms that optimize resource allocation while being HPC aware.

2. Cloud-Aware HPC

The proposed dynamic load balancing for efficient execution of tightly-coupled iterative HPC applications in heterogeneous and dynamic cloud environment. The main idea is periodic refinement of task distribution using measured CPU loads, task loads and idle times.

3. HPC -Aware Cloud Scheduler

The second method which adopt to bridge the gap between HPC and cloud is to focus on cloud schedulers and explore opportunities to a) improve HPC performance in cloud and b) reduce HPC cost when running in cloud. Current strategies for placement of VMs to physical machines are mostly HPC-agnostics, that is they do not consider the intrinsic nature of HPC applications.

4. Cloud-Aware HPC Runtime

The final approach follows to adapt HPC runtime to meet the needs of cloud environments. Our hypothesis is that the parallel runtime system should be able to adapt to the dynamic variations in the cloud execution environment, resulting in improved performance. In addition, by providing runtime support for dynamically expanding/shrinking parallel jobs, significant gains in terms of higher resource utilization and cost savings can be achieved by leveraging cloud features such as variable pricing.

REFERENCES

[1] Giacomo Domeniconi, Eun Kyung Lee, Alessandro Morari "CuSH: Cognitive ScHeduler for Heterogeneous HighPerformance Computing System". DRL4KDD, 2019, Alaska – USA.

[2] Zixia Liu, Hong Zhang, Bingbing Rao, Liqiang Wang, A Reinforcement Learning Based Resource Management Approach for Time-critical Workloads in Distributed Computing Environment,

[3] Kadupitiya Kadupitige, Intersection of HPC and Machine Learning, ENGR-E 687 IND STUDY INTEL SYS: FINAL REPORT

[4] White Paper: Make innovation real with AI and high performance computing Dell Emc.

[5] White Paper: Optimizing HPC architecture for AI convergence Intel corporation

[6] Whie Paper: Key Requirements for a Job Scheduling and Workload Automation Solution, Broadcom June 04 2019

[7] Learning Everywhere: Pervasive Machine Learning for Effective High-Performance Computation

[8] W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer Towards Efficient Mapping, Scheduling, and Execution of HPC Applications on Platforms in Cloud Simulation Of Hpc Job Scheduling And Large-Scale Parallel Workloads, Proceedings of the 2017 Winter Simulation Conference

[9] Albert Reuther, Chansup Byun, William Arcand ,Scheduler Technologies in Support of High Performance Data Analysis.

[10] The scheduling of task graphs in high performance computing as service clouds

[11] Kazumasa Sakiyama,, Shinpei Kato, Yutaka Ishikawa, Atsushi Hori, Abraham Monrroy,, Deep Learning on Large-scale Muticore Clusters, 2018 30th International Symposium on Computer Architecture and High Performance Computing

[12] Eric Gaussier, David Glesser "Improving Backfilling by using Machine Learning to Predict Running Times." SC '15, November 15-20, 2015, Austin, TX, USA