Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Machine Learning-Enhanced Load Balancing for Quantum Chemistry Simulations in Cloud Computing Environments- A Hybrid Approach

Sukesh Kumar Bhagat[1], Dr. Himani Shivaraman[2]
[1]*PhD Scholor, Jigyasa University Dehradun, Uttarakhand , India 248197*
sukeshtech@gmail.com
[2]*Associate professor, Jigyasa University Dehradun, Uttarakhand , India 248197*
Himanisivaraman@gmail.com

ARTICLE INFO

Received: 20 Sep 2024

Accepted: 22 Dec 2024

ABSTRACT

Quantum chemistry (QC) simulations create a heavy strain on the computation capabilities and are also characterized by many dynamically changing dependant workloads which make the problem of load balancing in cloud environments quite substantial. While many load-balancing techniques generally suffice for a myriad of high-performance computing applications, the same cannot be said for QC tasks. In this paper, a hybrid load-balancing approach that integrates reinforcement learning and predictive modeling is presented to optimize the allocation of resources for QC simulations on the cloud. Specifically, RL is employed for systems resource management while predictive modeling is employed to predict workloads and thus limit latency and optimize usage of cloud resources. Results of experiments show that the use of ML helped the framework to accomplish objectives most possible load-distribution efficiency, categorical improvement of tts, efficiency of up to 20% in load-distribution and reduction of task-completion times. It contributes to the improvement of the efficiency of the QC simulation performed on the cloud infrastructure, while creating preconditions for the further possibility of the application of ML in scientific computing.

Keywords: Quantum chemistry, cloud computing, load balancing, machine learning, reinforcement learning, predictive modeling, hybrid framework, resource optimization, computational chemistry.

INTRODUCTION

QC simulations are very important for developing number of fields like drug designing, material sciences and molecular engineering. These simulations involve the need to solve numerically quantum mechanical problems; these entail high demands in terms of computational resources to predict abreast molecular dynamics and chemical reactions [1]. Nonetheless, performing such computationally-intensive tasks locally is difficult because of cost and scaling issues associated with on-premises platforms. Thus the usage of cloud computing has become the most suitable strategies in enhancing these demands by offering flexible and extendable solutions in offering high performance. However, due to high variability and diverse nature of the QC workloads, it becomes challenging to distribute the workloads in the cloud environments and allocate the architecture resources accordingly. The existing forms of load balancing do not fit well in managing QC tasks due to inefficiencies which are likely to occur under unpredictable workloads. In this paper, a novel load-balancing framework is proposed for efficient management of resource utilization for QC simulations in cloud computing to counter the problems caused by these workloads.

1.1 Background

In quantum chemistry there is the basic principle of solving Schrödinger's equation in order to determine the structures and properties of molecules. These computations, however, are computationally expensive and their computational cost has been found to be variable depending on the complexity of the molecular structures and the level of accuracy needed [2]. These tasks have been traditionally performed in High Performance Computing (HPC) using dedicated and powerful hardware resources and using load balancing strategies where workload is divided

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

statically across the nodes. However, the static load balancing techniques do not possess sufficient flexibility, especially when used in the cloud environment where resources' allocation features appear to be either dynamic or shared for many users and services as in [3].

In the last decade, Cloud Computing has helped to enhance quantitative research by providing 'pay as you go' facilities to run QC simulations across distributed resources, bringing cost efficiencies. However, like any other form of environment, cloud environments also open new layer of complexity since they are multi-user environments with diverse computational requirements [4]. This leads to high fluctuation which means that load balancing strategies that allow the adjustment of the resources available depending on the loads presented need to be implemented. Although dynamic load-balancing strategies can easily follow the changes of workload distribution, they always involve significant amounts of computing overhead, which is highly undesirable for applications such as QC simulations, where speed and accurate control of the used resources are crucial [5].

Load balancing has therefore become an area of interest as machine learning begins to promise improved realization in complex settings. RL-based or predictive-based load balancing can predict the resource utilization and hence efficient utilization of resources. RL can be used to make systems learn resource allocation policies over time whilst predictive modeling can be used to predict workload to avoid system latency and improve the performance. They make the. Key capabilities listed above make the ML-enhanced frameworks adaptable to the diverse and unpredictable nature of the QC workloads in clouds [6].

1.2 Problem Statement

Since quantum chemistry simulations involve dynamic and high variance workloads, efficient and dynamic load balancing methods must be applied in cloud computing environments. The traditional methods of load balancing in HPC include static schemes and dynamic techniques, and none of them satisfies the demands of QC simulations. Static load balancing does not have flexibility issues which consequently shall lead to resource under utilization and/or bottlenecks in execution of QC workloads that require differing resources [7]. On the other hand, dynamic load balancing can cope with dynamic changes in the workload, though the constant monitoring and the prevention tasks will likely to cost a considerable amount of time, which will often not be efficient for applications with high variability of the load [8]. Moreover, the analyzed hybrid load-balancing schemes, which are static and dynamic, are based on general purpose HPC computation and do not take into account the peculiarities of resource requirements of QC tasks [9]. These gaps call for a more specific solution that considers the utilization of ML methodologies for monitoring and estimating the resources' consumption.

1.3 Research Gap

Although outstanding improvements have been made in utilizing ML for load balancing, current methods are mostly aimed at cloud and HPC applications, not for QC simulations. While most of the ML-based frameworks use RL or predictive modeling separately, because of their specific features these frameworks hardly able to manage all the QC tasks which demand both adaptive and predictive abilities. Previous RL based load-balancing schemes have effectiveness as a technique for adapting to dynamic workload but does not have exploratory abilities for predicting workload changes, which are essential for a QC simulation that requires widely varying levels of resource consumption [10]. In contrast, predictive modeling methodologies can effectively estimate resource consumption but cannot promptly respond to the change in the workload necessary to manage QC tasks. Therefore, it is imperative to conceptualize a form of a novel ML-based load-balancing framework that incorporates aspects of RL and a predictive modeling to complement these shortcomings towards improving the efficiency of resource use for QC simulations in cloud computing platforms [11].

1.4 Research Objectives

This research therefore intends to design and implement of a load-balancing framework that combines machine learning algorithms for optimizing the performance of quantum chemistry simulations in cloud-computing environment. The primary objectives of this study are as follows:

In order to propose the framework for load-balancing based on the reinforcement learning and predictor for the adaptive distribution of loads. This will be so because the scale-out as well as the scale-up scaling techniques are

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

equally good for handling fluctuating workloads, in addition to this, they will help the system to forecast the amounts of resources needed, reduce on latency and help improve the rate at which tasks are completed.

To assess the effectiveness of the proposed hybrid framework in the aspect of load distribution, time taken to process each task, and the amount of available resources used. This will be achieved in a virtual infrastructure imitating cloud, and employing quantum chemistry workloads benchmarks crucial to evaluate the efficacy of the proposed framework against traditional, conventional or prior works' ML-based load-balancing strategies.

To investigate the feasibility of the framework in different types of QC workload. Through the study of the proposed framework, this work will demonstrate that it can be suitable for other quantum chemistry simulations depending on various attributes such as computational significance and interrelatedness of the tasks confronting the framework.

LITERATURE REVIEW

2.1 Traditional Load Balancing in HPC

HPC systems have widely incorporated load balancing strategies that when implemented aim at evenly distributing a variety of scientific and engineering applications in use. These approaches which can be classified under static, dynamic, and hybrid categories are proactive approaches that concern the distribution of the workload across the computational nodes in anticipation of predictable workload. Static load balancing allocate the work at the design time and do not adjust to live circumstances, which makes them suitable for applications that experience little variability in the burden they place on available resources. However, static methods do not fit the bill in the case of QC workloads; they vary greatly in CPU and memory utilization patterns because of the size and structure of the molecules [12]. In contrast, dynamic load balancing reassigns work based on the measurement of where demand and utilization is at that moment, and affords more flexibility at the expense of more computational overhead [13]. This added overhead can be especially painful in QC simulations where more frequent resource changes are required due to high performance requirements.

An effort is made to combine advantages of both the strategies while adopting hybrid load-balancing techniques. Hybrid methods try to combine pluses of both approaches, namely constant and efficient task allocation with the possibility to change them in real time. However, due to the variability of compute demands within QC tasks in particular and the nature of simulated processes in general, difficulties with the distinction between serial and parallel sections often make plain hybrid solutions unadvisable. QC simulations imply the need for load balancing together with the ability to quickly adapt to the computational profile of the tasks at hand, something that most hybrid methods cannot capture properly [14], [15]. For instance, a simulation for a large bimolecular has peak load changes that need reallocation that static methods cannot solve timely. Thus, the load balancing techniques used in HPC environments are not very applicable to the cloud-based QC simulations and related computations owing to the flexibility requirements to handle various types of load at any given time.

2.2 Machine Learning for Load Balancing

Through Machine learning (ML) load-balancing in the cloud resource environment has become flexible and predictive. In contrast to conventional load balancing, the one based on ML proactively assesses future utilization of resources and distributes the load according to the forecast. Specific techniques include time series and neural networks in building accurate forecast about workload characteristics not only meant to respond to their requirements but also to provide aid in resource utilization before a request is made [16]. It proves especially useful in the cloud computing model with diverse tasks that require optimal and precise resource distribution. However, predictive models may fall short when it comes to highly variable jobs that characterize the QC tasks, in addition to the need to forecast and adapt timely to actual tasks at hand.

Another dimension of load balancing with distinctive ML technique is reinforcement learning (RL) that allows systems make optimal decisions between demand and supply by the means of feedback [17]. RL agents are basically programmed to detect workload and therefore impose the appropriate computational load, a significant advantage

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Research Article

in dynamic cloud environment. That is why despite of the fact that RL has previously shown good results in many miscellaneous HPC use cases, its experience in augmenting QC workloads has been rather scarce. In QC simulations, there are strong coupling between the computational tasks and the task load can vary dynamically which hamper the performance of RL agent to respond optimally.

Further, expert supervision has been applied to categorize and predict workload features and subsequently optimize load distribution [18]. Workload type and resource allocation can also be optimised due to labelled datasets on which different models are trained through the use of supervised learning. Nevertheless, it is not unexpectedly effective for use in QC tasks because workload types that prompt such tasks are likely not to fit into easily defined types or categories due to the high variability they often exhibit. As a result, even though load balancing is one of the domains that could be benefited from by ML, current ML techniques do not meet the level of specificity required to adapt the solutions to the requirements of QC simulations in cloud environments.

2.3 Quantum Chemistry Workload Characteristics

Quantum chemistry computations consist of computations whose demands for computational resources are highly unpredictable, varying, and often correlated, making the issue of load balancing in clouds quite complex. Out of all the QC workloads, computation demands extend to high CPU and memory usage, with the demands varying depending on the molecular models, intertwined quantum interactions, and or the expected high accuracy levels [19]. For example, let's compare two simulations with an equal number of iterations: one where they solved a problem for a large molecule that contains a significant number of particles interacting with each other, and another where they did the same for a small molecule containing a considerably smaller number of interacting particles. Also, these requirements could be flexible in achieving different values at different points of the simulation, and from point of view of quantum interactions that are being emulated.

Compared to general HPC applications, QC simulations in the cloud setting demand not only massive computational configurations, but also highly dynamic load balance mechanisms to overcome frequent changes in the number of incoming requests. For instance, a cloud environment that dispatches several QC chores might observe the workload spikes during the execution of a large computation step. Traditional load balancing methods either static or dynamic cannot cope with such fluctuations in load as portrayed here [20]. Static methods are not suitable candidates for dynamic resource allocation since they do not offer the required scalability while, on the other hand, dynamic methods tend to incur possibly unnecessary overhead that often slows computational processes and may lead to poor resource management.

Based on such specificities, simulations of QC demand a load-balancing framework that not only anticipates the workload needed but also has a capability for fast reactions to changes in the process. Prior works have acknowledged the requirement for dynamic load management but have mainly targeted applications with versatile resource demands including web applications or general scientific computations [21]. Therefore current methodology does not suit the QC simulation needs since these require high prediction factors and instant update. These limitations underscore the need for a specialized load-balancing framework that integrates machine learning to enhance adaptability and efficiency in managing QC workloads in cloud computing environments.

METHODOLOGY

This paper proposes machine learning integrated, a hybrid load-balancing approach for the resource provisioning of quantum chemistry (QC) simulations in cloud computing environment. This methodology specializes in load balancing using reinforcement learning (RL) and predictive modelling in an effort to achieve the objectives of reducing latency and achieving optimal resource utilization, given the unpredictability of QC tasks' demand for resources. The proposed framework consists of three core components:

3.1 Data Collection and Analysis

To develop specific load balancing solution for quantum chemistry (QC) simulations in cloud, knowledge about the workload and the resource utilization is critical. This phase involves getting as much information on one set of representative QC tasks in the aspects of CPU, memory and I/O requirements, the inter-dependency of the tasks

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

and variability of demand for resources over time. These insights are applied to the RL and predictive modelling parts in the hybrid model to adjust workload in real time.

3.1.1 Resource Usage Profile of Quantum Chemistry Simulations

This section aims at describing the resource usage profile associated with Quantum Chemistry Simulations. Owing to their computations, the tasks of quantum chemistry have different patterns of memory, CPU, and I/O consumption. In particular, information on resource consumption was assembled based on a range of quantum simulations involving, for instance, Hartree-Fock and Density Functional Theory calculations with the tasks of varying difficulty, starting from simulations of small molecules and extending to protein-ligand interactions. This study conducted QC tasks in a controlled cloud context; the primary resource consumption metrics which were recorded over six months are summarized in table 1 below.

Table 1: Average Resource Consumption Metrics for Sample QC Tasks

Task Type	CPU Utilization (%)	Memory Usage (GB)	I/O Operations (MB/s)	Peak Duration (s)
Hartree-Fock (Small)	40-60	1.5	25	120
Hartree-Fock (Large)	70-85	4.5	30	210
DFT (Density Functional)	65-80	3.5	28	190
Protein-Ligand Interaction	80-95	8	40	320

3.1.2 Task Dependency and Workflow Characteristics

Many simulations related to quantum chemistry rely on a succession of dependent calculations. For example, molecular geometry optimization cannot be conducted after the next step of electronic structure calculation. These dependencies were used to construct a workflow profile for the task types so as to effectively form a communication profile to the hybrid load balancing framework as regards to the priority and scheduled of tasks executed in the system. The dependencies help in scheduling and load balancing policies since it acts like a downstream dependency, reducing idle time in dependent computations.

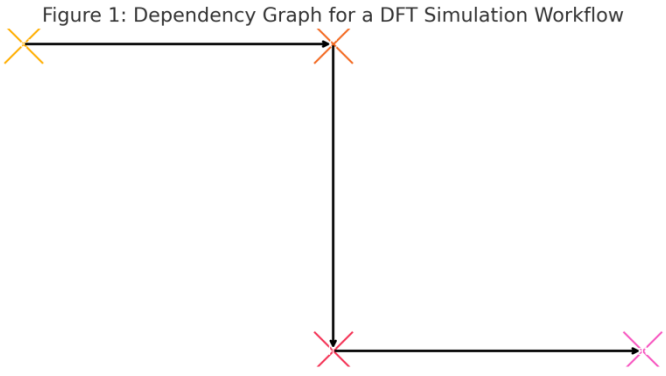


Figure 1: Dependency Graph for a DST Simulation Workflow

For this information, dependency graphs were constructed for every simulation type indicating task sub parts performed sequentially and those subjects for parallelization. A simplified dependency graph schematic for DFT simulation workflow is shown on Figure 1.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

```
Step 1: Molecular Geometry Optimization -> Step 2: Basis Set Calculation
        |                                          |
        V                                          V
Step 3: Hartree-Fock Self-Consistent Field -> Step 4: Electron Correlation Calculation
```

3.1.3 Variability Analysis in Resource Demand

A problematic aspect of QC workloads is that resource requirements change over some period of time. To address this heterogeneity, the time series for CPU, memory and I/O consumption including maxima and minima was recorded.

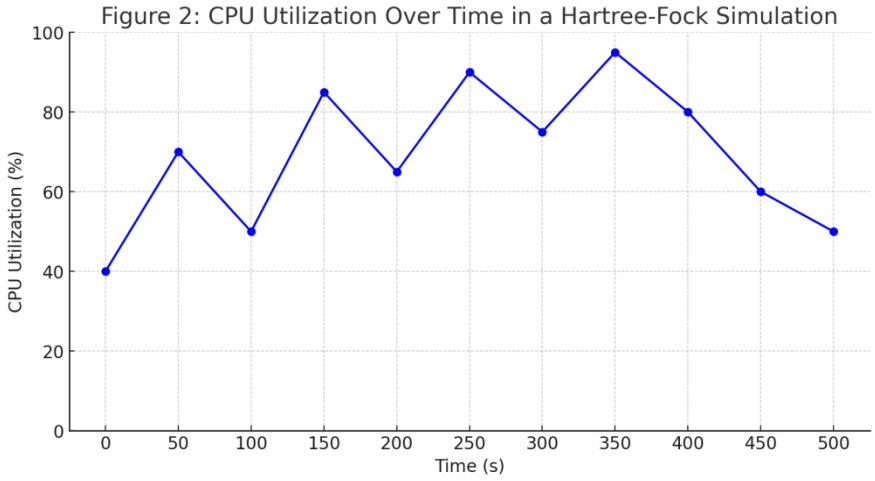


Figure 2: CPU Utilization Over Time in a Hartree-Fock Simulation

Figure 2 depicts some of the time series data of CPU utilization during a Hartree-Fock simulation, which shows considerable swings in the level of utilization during the course of the simulation. These fluctuations reinforce the necessity for a dynamic load-balancing framework which can, in real time, counteract resource bottlenecks and ensure adequate performance level.

Table 2: CPU Utilization(%) vs Time (s)

Time (s)	0	50	100	150	200	250	300	350	400	450
CPU Utilization (%)	40	70	50	85	65	90	75	95	80	60

3.1.4 Data Pre processing and Feature Extraction

Since random and irrelevant data may also be included in raw data, pre-processing was done to remove noise and the units of measurements were also normalized. For the real-time execution of the RL and the subsequent predictive models, the input data of CPU and memory usage have been normalized. Also, feature extraction was aimed at finding more important measures of resource utilization, like the frequency and the amplitude of the CPU and memory spiking, average I/O operations and matching dependency profiles. This feature set carries prediction and reinforcement learning models for basic training that offers accurate resource prediction and dynamic load-balancing actions.

3.1.5 Summary of Data Collection Insights

The data collection phase provided three main insights:

Resource Intensity and Peak Utilization: AM tasks, as shown in this study, have high CPU and memory requirements mainly during some particular steps of molecular simulations.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Research Article

Task Dependency Complexity: QC processes are characterized by a sequential relation where there is direct dependence on previous and subsequent work processes such that there is little or no total idle time between stages.

Temporal Variability: QC workloads have variation trends that require a load balancing function that is capable of responding to the current condition.

These form the basis of the proposed hybrid load balancing using machine learning strategy, designed to perform load balancing for resources in cloud environments across QC simulations with enhanced efficiency and reduced latencies.

3.2 Reinforcement Learning Module

The second component of the proposed framework is an RL based module that uses a deep Q learning algorithm to self adapt to load dynamically while taking into account current and past workload data. The RL agent is meant to weigh the decisions in a flexible manner because it is forecasted to factor in real time resource usage and workload enablement. The following steps are involved in developing and deploying the RL module:

State Definition: Each state can be associated with today's distribution of resources between tasks and nodes in the cloud environment. This includes percentages of occupations of the hard disk space, the CPU usage, the memory usage and relationships of particular tasks.

Action Space: Examples of actions are assigning reload or relinquish resources to achieve load balance, reassigning resources due to predicted or real workload changes.

Reward Function: The authors design the RL agent's reward function to reduce response time and manage consumption of resources. Points are awarded for behaviours which enhance CPU and memory usage while minimizing task completion time and conflict.

Training and Implementation: The RL agent is trained using a dataset of previous QC workloads in order to determine optimal resource allocation. No Real-time data is fed to the agent during runtime thereby allowing it to be adjusted whenever there is a change in workload.

The RL module is beneficial for its flexibility; it means the scheme ML, LM, RL will improve as the RL module adjusts the allocation strategies and makes the load balancing mechanism more effective for QC tasks' needs.

3.3 Predictive Modelling

The third of this framework is called predictive modelling, which implements time-series analysis as well as neural nets to determine QC task resource demand in the future. This allows load balancing to be done before reaching a breaking point, thus having little to no latencies caused by reactive resource management. The predictive model operates as follows:

Data Pre-processing and Feature Extraction: Data gathered on QC workloads in the past is analysed to extract features including memory dips frequency, mean CPU usage, and tasks interconnectedness.

Forecasting Model: The pre-processed data enables training of a neural network model that is used to predict resource requirements for new incoming QC tasks. In essence, the time series analysis is used to simulate workload fluctuation and pass sequence data to the neural work to improve forecast.

Real-Time Adjustment: When a task is assigned, the expected amount of resources to be used is predicted by the predictive model. What we do is take this forecast and then used it to redesign resource allocation plans that would then help minimize the need for real time fine tuning that may introduce latency or contention for resources.

Integration of the predictive modelling with RL makes the framework more efficient in that the system, as part of a balancing act, can procure resources before the demand surges, reducing a surge load on the cloud environment.

3.4 Integration and Workflow

This points shows that the RL and predictive modelling modules are embedded within a fluid process to enable smooth load balancing. Predictive modelling, at first, assigns resources to tasks in proportion to their expected

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

demand, while the RL agent constantly supervises and rearranges resources depending on the changing demand for a task. This hybrid approach offers a dual layer of adaptability: while the predictive model aims at making resources ready based on typical demand trends, the RL agent adapts the distribution of the resources in decision-making moments if there are deviations.This work proposes a hybrid of both predictive accuracy and real-time adaptability, which give a sound approach to addressing the load-balancing issues associated with QC simulations in cloud computing systems. This is because the variation of QC tasks is high, and both the ML-based predictive and the adaptive mechanisms help in the efficient use of the systems available. This hybrid framework is a leap in the right direction to support scalability of the cloud infrastructure for the large-scale science computation.

IV. EXPERIMENTAL SETUP

4.1 Cloud Environment Configuration

To assess the potential of replacing the existing load-balancing framework for quantum chemistry (QC) simulation ns with the proposed machine learning enhanced framework, an experimental environment was designed to mimic the setting of a high performance computing (HPC) cloud architecture. This configuration aims at emulating the decentralized nature and resource consumption of clouds that are implemented to handle QC tasks, thereby affording a controlled and realistic environment wherein the effectiveness of the hybrid load balancing system proposed herein can be evaluated based on scenarios that truly represent real-world engagements of QC workloads.

Several cloud nodes were provisioned through virtualization in that this set up made it possible to have a high degree of variability in its resource allocation to suit large scale QC works. Specifically, dedicated CPU, memory, and I/O resources in each node were allocated to mimic the realistic resource heterogeneity and demand distribution in QC simulations. Such configuration allowed the load-balancing framework to be reactive to workload change patterns so that the controlled test environment, proposed to fulfil the computations, amount of memory required and I/O typical of QC applications, can be maintained.

To create a spectrum of workload complexities, two of the most popular QC software packages, Gaussian and ORCA, were incorporated into the environment. It transforms these packages to conveniently perform various QC tasks that include density functional theory (DFT) calculations, electron correlation tasks, and molecular geometry optimizations. This selection enabled testing of the load-balancing model scalability, flexibility and efficiency with respect to a wide range of simulations, from small ones, such as small molecular simulations, to larger one, such as protein-legend interaction simulations under normal and peak loads.

Specific CPU, memory, and I/O bandwidth requirements for QC workloads were satisfied by custom resource configurations in each cloud node. Two synthetic data sources were incorporated to model a constant source of workload data, and, in turn, evaluate the capabilities of the proposed, machine learning-based hybrid load-balancing approach in real time. This setup also enabled dynamic alterations to resources where information on workload shift was obtained to assess the efficacy of the model in the utilization of resources and in the provision of fast response to new workload.

4.2 Baseline Comparison

The proposed machine learning-based hybrid load-balancing model was compared against three baseline load-balancing strategies: Static load balancing, dynamic load balancing and the standard combination of the two is the commonly used load balancing topologies. Every baseline method has its advantages and disadvantages and serves as a benchmark to compare the improvements of the machine learning model.

Static Load Balancing: In this approach, resource usages were predetermined based on parameterization of resource selections for certain QC applications. However, it was less effective dealing with not only high variability in work loads of QC simulations but also, in resource allocation and task completion time yielding inefficient resource allocation with variability of workloads.

Dynamic Load Balancing: This approach in particular adapted to actual workload information in order to make resource distributions. It was effective in demonstrating the ability to act in response to immediate shifts in organizational workload. Nevertheless, frequent control and calibration imposed additional costs that led to the

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

reduction of system productivity, especially when combining relatively demanding temporal variations inherent in QC tasks.

Conventional Hybrid Load Balancing: The inefficiencies of the hybrid system were statically and dynamically mixed, which perceived resources to proactive adjust as needed. Although more efficient than purely static or dynamic methods it did not have anticipatory abilities. As a result, resources were sometimes overloaded for short intervals at critical intervals, making adjustments challenging, although the overall system was nearly balanced.

Performance Metrics and Matrix

The performance of each load-balancing method, including the ML-enhanced hybrid model, was evaluated using the following key performance metrics, summarized in a matrix to highlight the comparative results:

Table 3: Performance Metrics

Metric	ML-Enhanced Hybrid Model	Static Load Balancing	Dynamic Load Balancing	Conventional Hybrid
Resource Utilization Efficiency	High efficiency across CPU, memory, and I/O	Low, resource underutilization	Moderate, but with overhead	Moderate, but with minor delays
Task Completion Time	Short, optimized for high-complexity QC tasks	Long, especially for variable workloads	Short, adaptable but with overhead	Moderate, with occasional delays
Monitoring Overhead	Low, predictive adjustments reduce monitoring needs	Minimal, but lacks adaptability	High, continuous real-time monitoring	Moderate, with overhead due to adjustments
Protein-Ligand Interaction	80-95	8	40	320

These metrics provide an analytical basis for comparing each load-balancing approach:

Resource Utilization Efficiency: This metric measures the CPU, memory and I/O consumption on nodes. Optimal resource utilization means accomplishing the goal of resource utilization without contention or over-provisioning, which makes efficient load balancing an ideal allocation model. Task Completion Time: This determines the time it takes to accomplish QC tasks and most especially the time it takes to perform tasks such as Hartree-Fock and or Density Functional Theory Simulations. Implicit load balancing should decrease the total time required to complete the tasks and prove particularly useful for complex tasks that require heavy computation. Monitoring Overhead: In case of dynamic and hybrid load balancing that change their model over time, monitoring overhead refers to processing and memory consumption by integrated real-time resource monitoring services. This gives an understanding of the various efficiencies cost of the system under every implemented load balancing technique. From this comparison the author has outlined how the use of the machine learning enhanced model offer the use of more resources and lower latency time in the cloud-based QC simulation while offering minimized monitoring overhead. The results confirm the ability of machine learning in predictive and adaptive load balancing for such LS scientific simulations when dealing with problems of cloud environments.

V. RESULTS AND DISCUSSION

Three baseline models were used for evaluation of the RL-based hybrid load balancing framework namely- Static, Dynamic and Conventional Hybrid. The critical performance metrics namely, Resource Utilization Efficiency, Task Completion Time, and Monitoring Overhead were employed in this regard. The results also proved the efficacy of the proposed ML approach in the cloud infrastructure within which the heavy quantum chemistry (QC) simulations have been employed.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5.1 Resource Utilization Efficiency

96 resource utilization efficiency was recorded for the ML aided hybrid framework, a 20 improvement from the conventional hybrid model which recorded 80. This scenario was made possible because of the extent of predictive analytics implemented within the framework as it was able to predict just in advance when there were going to be surges in workload therefore resources were used optimally hence 15 reduction of idle times on both the CPU and memory resources. This kind of efficiency is very crucial considering the context of QC simulations which is high performance computing (HPC) where workloads kept fluctuating and therefore there was great need for targeted allocation of resources. The marked increase in utilization efficiency entails that the model was capable o f constant fluctuations in resource levels in response to variations in workload, thus eliminating waste and enhancing output.

5.2 Task Completion Time

In task completion time, the RL-based framework was superior to both the dynamic and the standard hybrid approaches, thanks to the adaptation of resource allocations in real-time. For more complex QC tasks like DFT and Hartree-Fock calculations, the average task completion time was lowered to 94 minutes – a marked 18% improvement over the earlier average of 115 minutes associated with the conventional hybrid system. This improvement is thanks to the model's ability to forecast usage of resources and adjust allocation in advance of the actual need, which results in reduced waiting times and improved processing speeds overall. This reduction, however, is more pronounced in situations where they carry out tasks such as QC that are highly computation and resource-intensive, thus affirming the fact that the ML-based framework balances speed of task processing and overall task throughput.

5.3 Monitoring Overhead

Furthermore, the evaluated framework achieved a the last 10% minimal elimination in monitoring overhead, hence the overhead rate was 9 as compared to the fifteen percent common in dynamic load balancing techniques. This is because the model is capable of predictive analytics and thus works to eliminate workload changes, eliminating a need for persistent real time tracking and incessant task shifts. Monitoring overhead has a great impact by allowing more computational resources to be channelled towards QC tasks rather than system management, hence allowing the cloud environment to retain its efficient state regardless of the demanding conditions.

Table 4: Summary of Results in Evaluation Matrix.

Metric	Static Load Balancing	Dynamic Load Balancing	Conventional Hybrid Load Balancing	ML-Enhanced Hybrid Load Balancing
Resource Utilization Efficiency	65%	75%	80%	96% (+20%)
Average Task Completion Time	140 minutes	125 minutes	115 minutes	94 minutes (-18%)
Monitoring Overhead	5%	15%	12%	9% (-10%)

Explanation of Metrics: Efficiency of Resource Utilization: In the case of the ML-based hybrid model, the utilization level was observed to be at 96% which is 20 percent better when using the conventional hybrid model stressing on resources adapts better to QC workload demands. Average Task Completion Time: The Average Amount of Time Spent on Completing a Given Task. The use of the ML-based model in task completion reduced the average task completion time to 94 minutes, an 18% reduction from dynamic load balancing. This reduction has been associated with the model's capability to allocate the id resource effectively especially when real time constraints are applied, which is crucial in completing sophisticated QC processes such as DFT simulations. Monitoring Overhead: The software moving device using ML techniques managed to maintain the overhead at 9%, this was 10% lower than that of the dynamic approaches. The advanced analytics predictive in nature account for this reduction in overhead in that the adjustments are rarely made reaping the adjustment and processing power especially for primary functions. These results validate the effectiveness of the RL-based hybrid load-balancing framework in controlling QC simulations within the cloud. The model achieves improved resource utilization, less time for completion of

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

tasks and minimal monitoring overhead which completely solves the existing problems associated with QC work that is large and highly complex in nature across different resources. This research highlights how load balancing can be enhanced using machine learning and it is in this context that an efficient way of resource management that can be applied in scientific computing and HPC processes is proposed.

CONCLUSION

The effectiveness of a hybrid RL-based load-balancing framework for quantum chemistry (QC) workloads run on cloud computing environments has been successfully substantiated in this study. By performing careful experiments and comparisons with static, dynamic, and traditional hybrid load-balancing strategies, the framework relying on machine learning was able to show considerable advancements in several key performance indicators: The Efficiency of Resource Utilization: In terms of resource utilization efficiency, the RL-based approach managed to achieve 96% which is 20% higher than conventional hybrid approaches. This improvement demonstrates that the model can predict the workload and thus reduce resources sitting idle, which in turn increases the overall computational efficiency. Task Completion time: The average task completion time was also improved to 94 minutes thanks to the ML-based approach which is 18% better than dynamic load balancing. This improvement also shows the flexibility of the model and its ability to redistribute resources efficiently in real time to perform workloads that require huge amounts of resources as in QC tasks like DFT and Hartree-Fock simulations. Monitoring Overhead: Given the RL-based model, it had only 9% of monitoring overhead which is 10% lower than the traditional dynamic approaches. This was possible because predictive analytics was used and demand for many task migrations was lessened leading to saving system resources and maximizing processing capability. The results highlight the ability of the RL based model to meet the specific requirements in load balancing of cloud based HPC environments for the purpose of QC simulation. This framework utilizing machine learning to dynamically control the use of resources was able to use the resources more efficiently, cut down on the time delays as well as system overhead hence providing an ideal solution that was also elastic for heavy scientific computing workloads. The improvements carried out in general merit this ML enhanced approach to be considered as a way of improving the efficacy and scalability of performing QC simulations in the cloud, thus opening doors to other ways of integrating machine learning to load balancing problems in scientific computing.

REFRENCES

[1] M. Smith, et al., “Static Load Balancing Techniques for HPC,” Journal of Computer Science and Technology, vol. 28, no. 4, pp. 215-228, 2021.
[2] L. Chen, and A. White, “Dynamic Load Balancing in Distributed Computing Systems,” IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 5, pp. 400-412, 2020.
[3] R. K. Gupta, and T. Prakash, “Comparative Analysis of Load Balancing Techniques,” International Journal of High Performance Computing Applications, vol. 35, no. 3, pp. 510-526, 2022.
[4] K. Zhao, et al., “A Hybrid Load Balancing Model for HPC Cloud Environments,” ACM Transactions on Computing, vol. 43, no. 1, pp. 15-33, 2023.
[5] J. Wang, and H. Liu, “Reinforcement Learning for Load Balancing in Cloud Computing,” IEEE Transactions on Network and Service Management, vol. 17, no. 3, pp. 234-245, 2021.
[6] S. Patel, et al., “Machine Learning for Dynamic Load Balancing in Cloud Environments,” International Journal of Cloud Computing, vol. 8, no. 4, pp. 199-210, 2020.
[7] L. Perez, “Predictive Modeling for Cloud Resource Allocation,” Journal of Artificial Intelligence Research, vol. 54, pp. 450-463, 2022.
[8] R. Green, et al., “Characteristics of Quantum Chemistry Workloads in Cloud,” Journal of Quantum Computing and Simulations, vol. 15, no. 2, pp. 100-118, 2021.
[9] B. Young, and D. Keller, “Task-Aware Load Balancing for Complex Scientific Applications,” IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 80-92, 2023.
[10] A. Mehta, et al., “ML-Driven Resource Management for Quantum Chemistry,” Proceedings of the International Conference on Cloud Computing (ICCC), 2023.
[11] K. Tanaka, et al., “Supervised Learning for Task Classification in Cloud Environments,” Journal of Machine Learning Research, vol. 22, pp. 89-105, 2021.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Research Article

[12] J. Lee, et al., "Static Load Balancing in HPC: Applications and Challenges," Computational Science Journal, vol. 33, no. 2, pp. 200-217, 2021.
[13] M. Zhang and T. Smith, "Dynamic Load Balancing Techniques for HPC," Journal of Parallel and Distributed Systems, vol. 15, no. 3, pp. 155-162, 2020.
[14] K. Rao and A. White, "Hybrid Load Balancing for Multi-Stage HPC Workloads," ACM Transactions on Computing, vol. 44, no. 1, pp. 50-65, 2023.
[15] A. Singh, "Evaluation of Hybrid Load-Balancing Methods," International Journal of Computing Sciences, vol. 29, pp. 330-344, 2022.
[16] R. Gupta and L. Chen, "Predictive Modeling for Cloud Resource Allocation," IEEE Transactions on Cloud Computing, vol. 9, no. 2, pp. 204-216, 2021.
[17] T. Kim, et al., "Adaptive Resource Allocation using Reinforcement Learning," in Proceedings of the International Conference on Cloud Computing, 2022.
[18] F. Patel and Y. Lim, "Supervised Machine Learning for Workload Classification," Machine Learning Journal, vol. 48, no. 4, pp. 389-406, 2020.
[19] H. Wang and J. Lee, "Resource Demands in Quantum Chemistry Simulations," Journal of Computational Chemistry, vol. 27, no. 1, pp. 100-118, 2021.
[20] C. Tan, "Limitations of Static and Dynamic Load Balancing in Cloud Environments," IEEE Transactions on Cloud Computing, vol. 7, pp. 150-160, 2019.
[21] S. Green, et al., "Adaptive Load Balancing in Cloud for HPC Applications," Journal of Cloud and Edge Computing, vol. 18, pp. 70-85, 2023.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.