EMERGING TECH CONFERENCE – Edge Intelligence

Volume 01, 2022, Page 271 – 274

Proceedings of Emerging Tech Conference:
Edge Intelligence 2022

Estimating runtime, cost for Weather Research and Forecast tasks on AWS

Deepika Udupa*, Gregoris Malekos*, Kostas Flokos

dudupa@upcominds.com, gmalekos@upcominds.com

Abstract

Cloud scheduling is an NP-hard problem. In this paper, we propose an end-to-end system that executes a WRF (Weather Research and Forecast) task by providing recommendations of resource configurations. The optimal executions scenarios containing 3 options: fastest, cheapest and a trade-off is generated by estimating the runtime using a data-driven ensemble regression method.

1. Introduction

To create a reliable cloud scheduling platform for high performance computational jobs, a generalizable, accurate and fast method to predict the runtime of a given task execution on a given cloud configuration is needed [1]. This is particularly challenging to achieve because of the large numbers of parameters that influence the runtime, associated both with the task and with the cloud. Additionally, the problem is inherently stochastic [2] as two identical cloud configurations may have different hardware behind them. Finally, a method to break up a parallelizable workload between many virtual machines needs to be implemented. In this paper, we propose a system that uses a data-driven ensemble method to recommend cloud resource configurations for a user-specified deadline and cost. The goal of the system is to promote effective resource utilization for high performance computing tasks on AWS by identifying the right resource and level of parallelization of the task.

2. Related Work

In recent years, many machine-learning approaches have been suggested to solve the cloud scheduling problem. T.P. Pham et al. [3] proposed a two-stage algorithm, that used pre-runtime parameters to predict execution time. Other recent research, including the works of F. Nadeem et al. [4] and M Tanash et al. [5] point to ensemble methods being the most suited for this problem.

3. Methodology

In this section, the system is decomposed into 4 ‘tiers’ or ‘layers’: the platform, execution manager, orchestration manager, and the infrastructure layer. These layers work together to utilize the elasticity of AWS and recommend execution scenarios for the user to choose from. More detail is provided in the following subsections.

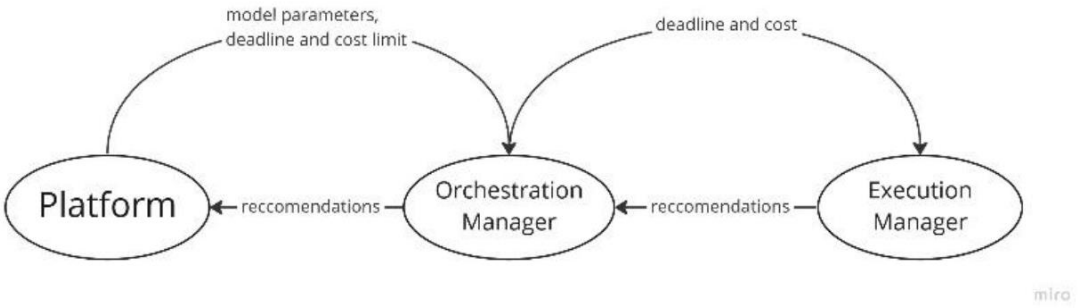

Figure 1: ENOS recommendation generation workflow

3.1 Platform

Through the platform the user can initiate the WRF task execution by defining the model parameters, uploading the dataset, and specifying the preferred deadline and cost limit for the execution. The user can choose one of the two ways to submit the execution: using the smart recommendation system or an expert mode with customizable resources. On successful completion, the platform provides an option for the user to download the results of the WRF task.

3.2 Orchestration Manager

The orchestration manager (OM) is the recommendation system that generates optimal hardware suggestions for each task. This is done using a data driven method which predicts the runtime and cost of executing a WRF task on a given AWS configuration. A smart multi-objective selection algorithm identifies the optimal options.

The OM has two main responsibilities: Estimation of runtime and cost of the WRF task for different AWS configurations and trimming the predictions using a smart selection mechanism for cheapest, fastest, and trade-off solutions.

3.2.1 Prediction

Each valid execution configuration gets added in an array. An ensemble regression algorithm is used on this array to estimate the runtime of each configuration. The dataset for training and the algorithm are built in-house. The features used for this prediction step include static and dynamic parameters of the problem and resources. The cost can then be calculated based on the runtime predicted. The features used have been kept as generic as possible to allow for a future application of the same algorithm to other problems and cloud platforms. Each submitted execution is retrofitted to the dataset and used to periodically retrain the regressor to achieve a continuous improvement on prediction accuracy.

3.2.2 Selection

The solution array of execution scenarios with the runtime and cost predictions is trimmed using a NSGA-II [6] based selection mechanism. The user defined deadline and cost limit are used as constraints to arrive at 3 optimal solutions. The one with the lowest execution time, the one with the lowest cost, as well as a trade-off. The solutions are sent to the platform via the execution manager.

3.3 Execution Manager

This component is a broker between the platform, orchestration manager and the available infrastructure. It is responsible for user management and workflow management. The execution manager allocates and deploys resources dynamically using the infrastructure layer, monitors the status of executions, and collects execution metrics.

3.4 Infrastructure Layer

The infrastructure layer uses Kubernetes to deploy and scale executions to the desired level of parallelization. Container images for WRF are pre-built and are configured at runtime by loading the necessary data (dataset, config files, parameters) using a distributed filesystem.

4. Results

Although the percentage error of the runtime prediction can vary a lot for most cases it is under 20%. In most cases, points with errors over this mark correspond to smaller workflows for which an error of a few minutes can translate into a significant percentage error. Additionally, some very lengthy workflows produced large errors as the amount of training data for this type of workflow is limited.

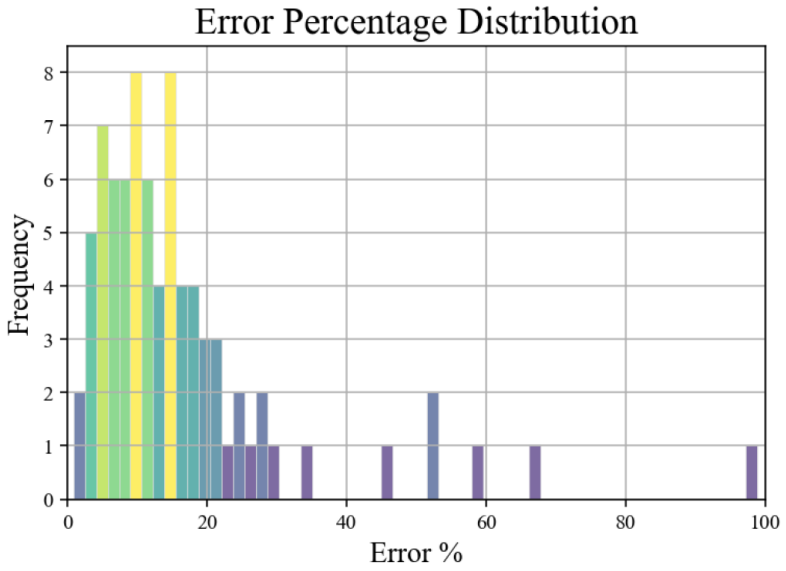

Figure 2: Percentage error plot

5. Conclusion

In this paper, we introduced an end-end system that assists the users to execute their WRF task on AWS by providing configuration recommendations. The results presented can be further improved by training on a larger dataset and collecting more diverse datapoints. Future work will include the creation of a large-scale dataset for runtime prediction of scientific workflows. We also aim to generalize our prediction capabilities to include many different problems on many different clouds. Further research is required to identify measures that can be used to correlate the data gathered for one type of scientific workflow to another.

References

[1] Awan, Mehwish, and Munam Ali Shah. "A survey on task scheduling algorithms in cloud computing environment." International Journal ofComputer and Information Technology 4.2 (2015): 441-448.

[2] Lu Yin, Junlong Zhou, Jin Sun, A stochastic algorithm for scheduling bag-of-tasks applications on hybrid clouds under task duration variations, Journal of Systems and Software, 10.1016/j.jss.2021.111123, 184,(111123), (2022).

[3] T.Pham et al., Predicting Workflow Task Execution Time in the Cloud Using A Two-Stage Machine Learning Approach, IEEETRANSACTIONS ON CLOUD COMPUTING, VOL. 8, NO. 1, JANUARY-MARCH 2020

[4] M.Tanash et al., Ensemble Prediction of Job Resources to Improve SystemPerformance for Slurm-Based HPC Systems, PEARC '21: Practice andExperience in Advanced Research Computing July 2021

[5] F.Nadeem et al. Using Machine Learning Ensemble Methods to Predict Execution Time of e-Science Workflows in Heterogeneous DistributedSystems. Digital Object Identifier 10.1109/ACCESS.2019.2899985

[6] K.Deb et al. A fast and elitist multiobjective genetic algorithm: NSGA-II,IEEE Transactions on Evolutionary Computation (Volume: 6, Issue: 2, April 2002)