Ad-hoc runtime prediction of scientific workflow tasks - performance comparison of state-of-the-art approaches
-
Bachelor Thesis Exposé

Wilhelm Simon

October 2022

1 Abstract

Scientific workflows are popular in various sciences today. They serve as a way to execute collections of interdependent tasks. Scientific workflows are often deployed in clouds and can be computationally intensive. Many users, therefore, opt for deployment in clouds of cloud service providers. Users pay for these clouds based on resource usage. Various approaches exist that address workflow scheduling in clouds, aiming to reduce the cost of executing scientific workflows. However, precise information about tasks is required to apply these scheduling approaches. One important information about a task is its runtime. The scheduler requires this information before the task is started.

In this thesis, we will evaluate online machine learning approaches' performance in the prediction of the runtime of scientific workflow tasks. For evaluation, we compare the predicted runtimes with actual runtimes. Implementations will be done in real systems. The goal of this research is to understand how well state-of-the-art online machine learning approaches for scientific workflow task runtime prediction perform in these real systems.

2 Research Question

Scientific workflows are often deployed in clouds provided by online cloud service providers. Users pay for these services based on resource usage. Resource usage can be reduced through intelligent scheduling strategies. Accurate prediction of the runtime of scientific workflow tasks enables more informed scheduling. Precise task runtime predictions are possible using machine learning [1]. Multiple different machine learning methods have been researched, including online machine learning [2]. Additionally, research claims online machine learning as well-suited for making predictions on scientific workflows [3, 4].

In this thesis, we will evaluate the prediction performance of multiple online machine learning algorithms regarding scientific workflow task runtime prediction. This thesis considers online learning as an approach that does not use historical data from previous workflow executions. Instead, only data from the current workflow execution is used for training, and the model is updated as soon as a task finishes. This is possible since workflows execute similar tasks dozens to hundred times. We will compare different algorithms and analyze their accuracy, variance, rate of convergence and resource consumption in real systems. This leads us to the following research question:

How does ad-hoc runtime prediction of cloud-deployed scientific workflows perform?

3 Tooling

Nextflow is a DSL and Scientific Workflow Management System for writing and deploying scientific workflows in distributed environments [5]. A variety of Nextflow workflows are available through nf-core, a community-driven collection of analysis pipelines built for Nextflow. In this thesis, we will implement different task runtime prediction approaches and compare them using nf-core workflows.

Kubernetes is an open-source, container-based orchestration tool used for various tasks in software deployment, such as the execution of scientific workflows, software scaling, and software management. It is frequently used by cloud service providers to manage their workloads. The task runtime prediction approaches will be evaluated in Kubernetes clusters.

Docker is a software containerization solution maintained by Docker, Inc. and provides OS-level virtualization of software packages. Docker is one of the containerization systems used by Kubernetes to deploy software, and will therefore be used in this paper to deploy Nextflow workflows.

4 Related Work

Scientific workflows pose various problems to users, like accurate resource provisioning, or efficient scheduling of tasks during workflow execution. Machine learning methods provide one way to work towards practical solutions for these problems [3, 4]. In this section, we will look at research relevant to our thesis. We group the presented research by machine learning methods used and computing environments targeted. First, we will look at research exploring attribute prediction of scientific workflows using regression and classification methods in different distributed environments. Second, we will examine research about online machine learning strategies, such as reinforcement learning, for scientific workflow runtime prediction in clouds.

McKenna et al. use regression methods to establish an approach to predict runtimes of tasks in high-performance computing (HPC) clusters [6]. To predict these runtimes, machine learning implementations of decision trees, random forests, and K-nearest neighbors algorithms are evaluated. The decision tree implementation was found to be the best-performing.

Rosa et al. explore the efficient execution of scientific workflows in a federated cloud [1]. The research used bioinformatics workflows and prediction of tasks' runtimes was performed. A combination of heuristics and model-based machine learning regression algorithms was used. The models were trained on historical data and, when lacking such data, simulations of workflow runs that were performed.

Further research is done by Gulino et al. [7]. Here, model-based machine learning is combined with statistical methods to make a general performance prediction of cloud-deployed scientific workflows. To make this general performance prediction, multiple workflow characteristics were predicted. This included the prediction of individual workflow tasks to guess the workflows overall runtime. The approach does not require historical data or simulated runs of workflows to make the performance predictions. However, the models used are pre-trained, and the research evaluates results exclusively in Apache Spark environments.

Next, we will look at reinforcement-learning and online learning methods. First, Yang et al. [8] implement a model-based reinforcement-learning approach that includes runtime prediction for scientific workflows.

In [9], Nascimento et al. use Q-learning, a model-free reinforcement learning method, to increase the efficiency of scientific workflow scheduling in clouds. However, no direct runtime prediction is performed. Instead, the machine learning algorithm optimizes workflow execution by deciding which action to perform next, based on internal scores given to possible next actions.

Hilman et al. research online machine learning. The research implements an online incremental approach to predict the runtime of tasks of scientific workflows [2]. Recurring neural networks (RNNs) and an online learning K-nearest neighbors implementation, called IBk, are used. Both approaches are used to predict the runtime of multiple different workflows and their performance is evaluated. The evaluation is done by comparing the prediction performance of the RNN, IBk, and a batch offline-learning baseline based on work by Da Silva et al. [10]. The online learning approach outperforms the baseline and the state-of-the-art prediction approach implemented by Hilman et al. However, the approach relies on feature extraction which requires time-series performance monitoring data of the computing environment. Additionally, the online learning IBk model was pre-trained.

To the best of our knowledge, no paper compares online machine learning runtime prediction of scientific workflow tasks implemented in real systems and with real workflows. This thesis seeks to fill this gap.

5 Research Approach

Nextflow is a commonly used framework for scientific workflow deployment and Kubernetes is a common tool for orchestrating distributed computing environments, used by many cloud service providers today. Therefore, the implementation of state-of-the-art approaches for runtime prediction of Nextflow workflows deployed in a Kubernetes cluster without the use of historical data is a relevant extension of existing research.

To implement an online machine learning approach for scientific workflow task runtime prediction in Nextflow, we will extend Nextflow and implement a custom scheduler. The custom Nextflow instance will expose needed data points. The custom scheduler will be responsible for predicting the runtime of Nextflow workflow tasks at submission time, and for collecting runtime data for analysis purposes. Runtime predictions will be done with state-of-the-art online machine learning approaches for runtime prediction of scientific workflow tasks, established by existing research. We will implement several online machine learning algorithms leveraging input data to predict runtimes and then update the algorithms using the actual task runtime. This input data consists of the size and number of files used per task, the contents of the tasks' Nextflow scripts, and the executing node.

Additionally, monitoring data of the computing environment, like memory and CPU usage, are available. We will test the algorithms on Nextflow workflows deployed in a local kind cluster. Subsequently, we test each algorithm with two different scientific workflows taken from nf-core. There will be no use of time series data for making task runtime predictions. Additionally, we will attempt to make the predictions without training the online machine learning models a priori.

After testing, we will run the workflows in Kubernetes clusters and collect the necessary data to then evaluate the runtime predictions. Necessary data is at least the predicted runtime by each different implementation combined with the actual runtime of the workflow tasks, which is known at time of task completion. We will then evaluate the results by comparing each online machine learning algorithm for each workflow. The comparison will be done over prediction performance, resource consumption, variance, accuracy, and rate of convergence. This allows us to determine which approach achieves the best results and eventually make some comparisons to existing research on this topic.

References

[1] Michel J. F. Rosa, Aletéia P. F. Araüjo, and Felipe L. S. Mendes. Cost and time prediction for efficient execution of bioinformatics workflows in federated cloud. 2018.

[2] Muhammad Hafizhuddin Hilman, Maria Alejandra Rodriguez, and Rajkumar Buyya. Task runtime prediction in scientific workflows using an online incremental learning approach. 2018.

[3] Ewa Deelman, Anirban Mandal, Ming Jiang, and Rizos Sakellariou. The role of machine learning in scientific workflows. *The International Journal of High Performance Computing Applications*, 33(6), 2019.

[4] Azita Nouri, Philip E. Davis, Pradeep Subedi, and Manish Parashar. Exploring the role of machine learning in scientific workflows: Opportunities and challenges. 2021.

[5] Paolo Di Tommaso, Maria Chatzou, Evan W. Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. Nextflow enables reproducible computational workflows. 2017.

[6] Ryan McKenna, Stephen Herbein, Adam Moody, Todd Gamblin, and Michela Taufer. Machine learning predictions of runtime and io traffic on high-end clusters. 2016.

[7] Andrea Gulino, Arif Canakoglu, Stefano Ceri, and Danilo Ardagna. Performance prediction for data-driven workflows on apache spark. 2020.

[8] Zhe Yang, Phuong Nguyen, Haiming Jin, and Klara Nahrstedt. Miras: Model-based reinforcement learning for microservice resource allocation over scientific workflows. 2019.

[9] André Nascimento, Vítor Silva, Aline Paes, and Daniel de Oliveira. An incremental reinforcement learning scheduling strategy for data-intensive scientific workflows in the cloud. 2021.

[10] Rafael Ferreira da Silva, Gideon Juve, Mats Rynge, Ewa Deelman, and Miron Livny. Online task resource consumption prediction for scientific workflows. 2015.