Resource Demand Profiling of Monolithic Workflows

Ivo Rohwer
ivo.rohwer@uni-wuerzburg.de
Julius-Maximilians-Universität Würzburg
Würzburg, Germany

Maximilian Schwinger
maximilian.schwinger@dlr.de
German Aerospace Center (DLR)
Oberpfaffenhofen, Germany

Nikolas Herbst
nikolas.herbst@uni-wuerzburg.de
Julius-Maximilians-Universität Würzburg
Würzburg, Germany

Peter Friedl
peter.friedl@dlr.de
German Aerospace Center (DLR)
Oberpfaffenhofen, Germany

Michael Stephan
michael.stephan@lrz.de
Leibniz Rechenzentrum
Garching, Germany

Samuel Kounev
samuel.kounev@uni-wuerzburg.de
Julius-Maximilians-Universität Würzburg
Würzburg, Germany

ABSTRACT

We propose a novel approach for resource demand profiling of resource-intensive monolithic workflows that consist of different phases. Workflow profiling aims to estimate the resource demands of workflows. Such estimates are important for workflow scheduling in data centers and enable the efficient use of available resources. Our approach considers the workflows as black boxes, in other words, our approach can fully rely on recorded system-level metrics, which is the standard scenario from the perspective of data center operators. Our approach first performs an offline analysis of a dataset of resource consumption values of different runs of a considered workflow. For this analysis, we apply the time series segmentation algorithm PELT and the clustering algorithm DBSCAN. This analysis extracts individual phases and the respective resource demands. We then use the results of this analysis to train a Hidden Markov Model in a supervised manner for online phase detection. Furthermore, we provide a method to update the resource demand profiles at run-time of the workflows based on this phase detection. We test our approach on Earth Observation workflows that process satellite data. The results imply that our approach already works in some common scenarios. On the other hand, for cases where the behavior of individual phases is changed too much by contention, we identify room and next steps for improvements.

CCS CONCEPTS

• **Software and its engineering → Process management**; • **General and reference → Cross-computing tools and techniques**.

KEYWORDS

Phase Detection, HMM, Workflow Profiling

ACM Reference Format:
Ivo Rohwer, Maximilian Schwinger, Nikolas Herbst, Peter Friedl, Michael Stephan, and Samuel Kounev. 2024. Resource Demand Profiling of Monolithic Workflows. In *Companion of the 15th ACM/SPEC International Conference on Performance Engineering (ICPE Companion '24), May 7–11, 2024, London, United Kingdom.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3629527.3651425

1 INTRODUCTION

As the possibilities for collecting data increase year on year, the efficient processing of these data volumes is becoming an ever-growing challenge. A frequently used approach for processing large amounts of data are scientific workflows. Data centers are trying to make the best possible use of their available resources without causing a decline in the quality of service. To achieve this, the prediction of resource demands plays an important role, as these estimates are the basis for workflow scheduling decisions. The simplest but often used resource demand estimates focus on determining a worst-case value for each workflow and do not take into account the occurrence of different phases with different resource demands within a workflow. On the other hand, some approaches use the measured resource utilization history to predict the resource utilization for a future time period of a specified fixed length. Even more precise predictions are possible using approaches that predict the resource demand of individual workflows taking into account different phases. These approaches use unsupervised clustering methods, such as k-means or Hidden Markov Models (HMMs). In this paper, we present a new approach that enables supervised training of a HMM by combining offline and online algorithms. In this way, as few as just recorded systems-level metrics of about 30 executions are required as training data. Furthermore, we provide a method that uses the online phase detection of the HMM to update online the offline-created resource demand profiles.

The use of a method that predicts the resource demand of a single workflow makes sense for monolithic workflows that have such a high resource consumption that it is worth predicting each individual workflow. We select as application domain Earth Observation (EO) workflows. EO workflows are scientific workflows that process satellite data and, for example, are used to determine the condition of forests [11], record the movement of Antarctic ice [2], or monitor daily changes in global freshwater bodies [7]. Because of the mentioned characteristics of EO workflows, we use them to

Resource Demand Profiles

We understand a resource demand profile to be an estimate of the resource demands of a specific workflow over time. The profile indicates for the workflow under consideration how long it is expected to run at maximum and indicates for each point in time during this period how high the maximum resource demand is expected to be.

evaluate our approach. We consider the workflows as black boxes, but we assume that we have a data set of resource consumption values available for each individual workflow.

The remainder of this workshop paper is organized as follows: In Section 2, we summarize related work. In Section 3, we describe our approach and discuss the algorithms we chose for the individual steps of our approach. In addition, in Section 4, we show the performed experiments and present the results. Finally, in Section 5, we give a conclusion and discuss future work.

2 RELATED WORK

In the following, we present briefly related work regarding our approach. **Workflow prediction for sets of applications** There are many workflow prediction approaches that use classic time series prediction methods for resource demand prediction, like for example, [10], [14] or [13]. This is not an ideal solution in the scenario we consider, as most time series prediction approaches are not designed to predict changepoints of time series, but rather, like ARIMA or ARMA for example, assume that the time series retains its statistical properties, which is precisely not the case after a phase transition that can be seen as a change point of the resource consumption time series. Different papers have also been published in this area using HMMs, such as the work by Kahn et al. [6]. They do take phases into account, but only to the extent that they determine the current phase and predict the resource demands of the next phase: The approach deals only with intervals of fixed length (for example 15-minute intervals in their paper). In addition, their approach assigns a phase only one of five possible CPU utilization levels. In contrast, we model sequences of phases of arbitrary lengths and resource demands. Furthermore, we adjust flexibly our resource demand profiles based on the observed phase lengths of already completed phases. Another example of the use of HMMs for workflow prediction is the work of Adel et al. [1]. In their approach, a HMM is trained for each resource, which is then used to predict the utilization of the individual resource, whereby they distinguish between four different load state classes. The goal of their approach was to use the predictions for autoscaling. The approach was tested successfully in a simulation.

Resource demand prediction of individual applications Gupta et al. [5] proposed a relatively simple approach to phase-based resource demand estimation based on the k-means algorithm. This is used to cluster the time series into phases. Subsequently, a phase transition table can be created based on these results. During operation, the k-means algorithm is also used to identify the current phase and then the upcoming phases are predicted using the transition table. Furthermore, in this approach, a table is created to track the CPU utilization of phase combinations of different workflows in order to make scheduling decisions based on this information.

Prats et al. [8] proposed an approach for phase-based prediction of workflows using HMM to profile individual workloads. However, unlike us, they train their HMM unsupervised, i.e., they also use it to extract the phases from the time series of resource consumption values. In contrast to this, we use deterministic algorithms to extract the phases from these time series. With the labels created in this way, we train the HMM in a supervised manner and use it for phase detection and prediction.

This makes it possible to predict the phase progression of a workflow from start to finish. This is not possible when using the unsupervised approaches, as it is not possible to prevent two separate segments in a run from being assigned to the same phase type. However, this means that no start-to-finish predictions can be derived from the phase transition tables resulting from these segmentations, as circles are possible in the phase sequences.

3 APPROACH

As described in the introduction, the goal of our work is to create a model for the resource demand estimation of monolithic workflows that we regard as black boxes. We assume that a dataset of recorded monitoring data from different example runs for a considered workflow type is available.

The first step is to divide the example runs from the data-set into meaningful phases. This is done in our approach by the PELT algorithm. Once the example runs are segmented into meaningful phases, we have to determine which of these segments from different example runs belong to the same phase. It is not possible to derive this information from the order of the phase segments, since, for example, some phases can only occur optionally or can be changed in their properties by contention. For this reason, we use the DBSCAN clustering algorithm for this task. Each of the resulting clusters represents one phase type. All identified phase types can now be characterized in terms of their run durations and resource demands. With this information, it is possible to estimate the possible start and end dates of the individual phases. Based on these estimates, a resource demand profile can be created for a workflow, which gives at each point in time a maximum resource demand value of all phases that could possibly be active at that point in time.

Our approach aims to update the estimates of the start times of the individual phases at the runtime of the individual workflow instances. This makes it possible to significantly increase the accuracy of the profiles: For example, if a phase of a workflow has a high resource demand, the appropriate amount of resources must be reserved for this phase for the entire time in which it may occur. This time is usually significantly longer than the actual maximum runtime of this phase, as we do not know exactly when this phase actually will start. This means that in this case, the actual resource demands of a workflow are significantly overestimated by the initial resource demand profile. However, online phase recognition makes it possible to narrow down the estimates of the start times of the remaining phases. With this information, the resource demand profile can then be updated and the overestimation of resource demands can be significantly reduced. For the task of online phase detection, we train a HMM in a supervised manner, based on this phase segmentation and labeling. Our code and data artifacts are available for reproducibility and reusability via a CodeOcean Capsule at [9].

3.1 Resource Demand Profile Creation

Our approach creates a resource demand profile at the start of a workflow as well as at each recognized phase transition. The first step of our resource demand profiling approach is to determine all possible phase sequences that may follow the current phase. This is done by evaluating the transition probability table of the HMM.

Each of these possible phase sequences is then considered individually. The minimum possible start time and the maximum possible end time are calculated for each phase in each sequence. We assume the minimum duration of a phase to be the mean duration minus two standard deviations and the maximum length to be the mean duration plus two standard deviations. The earliest and the latest point in time at which a phase can possibly start in a considered sequence is given by the sum of either the minimum or the maximum lengths of all previous phases of this phase. Now, it is possible to calculate for each phase sequence for any time step all phases that can possibly occur at this time step. For each time step, we can now calculate the maximum of all mean resource consumption values plus two standard deviations of all phases that may occur in this time step.

If these maximum values are calculated for all time steps of all possible phase sequences that can follow from the current phase, the final resource demand profile can be calculated as follows: For each time step, we iterate through all values at this time step of the different sequences and take the maximum value for this time step. In this way, we can create an upper-bound estimation for the future demand of a specific resource for the considered workflow.

4 PRELIMINARY EVALUATION

We evaluate our approach on the terrabyte platform [3, 4] hosted at the Leibniz Rechenzentrum. As described in the introduction, we profile EO workflows to test our approach. In the following, we present the results of our experiments with the Multi-SAR workflow [12] regarding the memory consumption profile of this workflow. The Multi-SAR workflow processes data from radar satellites. We consider a scenario where a set of Multi-SAR workflow instances is to be executed concerning a certain memory limit. We compare the results of ASAP (as soon as possible) scheduling of these workflow instances in combination with different resource demand profile types including our approach. We measure the total execution times for the entire workflow set, i.e. the time span between the start of the first workflow instance and the end of the last workflow instance. The following three profile types are compared by us:

(1) **Constant** resource demand profiles, which contain a value based on the maximum resource consumption of a workflow
(2) **Static** resource demand profiles, which are non-constant and take into account the different resource demands of individual phases but are not updated at the runtime of the workflows
(3) **Dynamic** resource demand profiles, which are created by our approach and are updated at the runtime of individual workflow instances

The static profiles correspond to the initial profiles generated by our model but are not updated at the runtime of the workflows.

In Table 1, the exact number of time steps required to complete 20 instances of the Multi-SAR workflow that were scheduled by ASAP in combination with the three compared resource demand profile types. As we can see, the time required in the case of constants and static profiles does not differ significantly in the scenario under consideration. In contrast, our approach seems to significantly speed up the execution of the workflows: In this experiment, the workflows were completed more than 23% faster without exceeding the resource limit.

limit	*number of workflow instances*	*profiling approach*	*duration in time steps*	*improvement vs constant*
200 GB	20	constant	650	-
200 GB	20	static	641	1.38%
200 GB	20	dynamic	499	23.23%

Table 1: The table shows the results of our experiments on the terrabyte platform in terms of the time required for 20 workflow instances. The improvement is shown in comparison with the constant procedure.

The results of our experiments show that our approach works for the memory consumption values of the Multi-SAR workflow. However, additional experiments have shown that our approach runs into problems if the behavior of individual phases is changed too much by contention. This is especially the case with time series of CPU data from multi-threaded workflows, as these time series exhibit a high variance.

5 CONCLUSION AND FUTURE WORK

In this paper, we present an approach for phase-based workflow profiling and online phase-detection for monolithic workflows. We use offline time series segmentation and density-based clustering to identify individual phases in a resource consumption value time series dataset. Subsequently, we train a HMM for online phase detection of the corresponding workflow. Furthermore, we present a method to utilize this phase detection to adjust the generated resource demand profiles at the runtime of the workflows. We evaluate our approach using Earth Observation workflows. Our experiments regarding memory consumption show that our approach works well in scenarios where the behavior of individual phases is not changed too much by contention.

However, if contention changes the behavior of individual phases too much, our approach has problems creating a consistent phase model. Here it seems to be a problem to call the PELT algorithm with the same hyper-parameters for highly different time series. Therefore, we aim to optimize our approach in this respect. In addition, further experiments are also needed, especially experiments that combine our approach with more specialized scheduling algorithms, in order to evaluate how well our approach would work in practice.

ACKNOWLEDGMENTS

This research is funded by the Bavarian Research Institute for Digital Transformation (bidt), an institute of the Bavarian Academy of Sciences and Humanities as part of the project ROOT: Real-time earth Observation of fOrest dynamics and biodiversiTy (KON-22-024).

REFERENCES

[1] Ahmed Adel and Amr El Mougy. 2022. Cloud Computing Predictive Resource Management Framework Using Hidden Markov Model. In *2022 5th Conference on Cloud and Internet of Things (CIoT)*. 205–212. https://doi.org/10.1109/CIoT53061.2022.9766809

[2] Celia A. Baumhoer, Andreas J. Dietz, C. Kneisel, and C. Kuenzer. 2019. Automated Extraction of Antarctic Glacier and Ice Shelf Fronts from Sentinel-1 Imagery Using Deep Learning. *Remote Sensing* 11, 21 (2019). https://doi.org/10.3390/rs11212529

[3] Jonas Eberle, Maximilian Schwinger, and Julian Zeidler. 2023. Challenges in the development of the EO Exploitation Platform terrabyte. In *Proceedings of the 2023 Conference on Big Data from Space (BiDS'23) – From foresight to impact* (Vienna, Austria). Publications Office of the European Union, 97–100. https://doi.org/10.2760/46796

[4] German Aerospace Center (DLR). [n. d.]. terrabyte. https://www.dlr.de/eoc/terrabyte. Accessed: 2023-12-04.

[5] Piyush Gupta, Shashidhar G Koolagudi, Rahul Khanna, Mrittika Ganguli, and Ananth Narayan Sankaranarayanan. 2015. Analytic technique for optimal workload scheduling in data-center using phase detection. In *5th International Conference on Energy Aware Computing Systems & Applications*. IEEE, 1–4.

[6] Arijit Khan, Xifeng Yan, Shu Tao, and Nikos Anerousis. 2012. Workload characterization and prediction in the cloud: A multiple time series approach. In *2012 IEEE Network Operations and Management Symposium*. 1287–1294. https://doi.org/10.1109/NOMS.2012.6212065

[7] Igor Klein, Ursula Gessner, Andreas J. Dietz, and Claudia Kuenzer. 2017. Global WaterPack – A 250m resolution dataset revealing the daily dynamics of global inland water bodies. *Remote Sensing of Environment* 198 (2017), 345–362. https://doi.org/10.1016/j.rse.2017.06.045

[8] David Buchaca Prats, Josep Lluís Berral, and David Carrera. 2017. Automatic generation of workload profiles using unsupervised learning pipelines. *IEEE Transactions on Network and Service Management* 15, 1 (2017), 142–155.

[9] Ivo Rohwer. 2024. Resource Demand Profiling of Monolithic Workflows. https://www.codeocean.com/. https://doi.org/10.24433/CO.0301206.v1

[10] Nilabja Roy, Abhishek Dubey, and Aniruddha Gokhale. 2011. Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting. In *2011 IEEE 4th International Conference on Cloud Computing*. 500–507. https://doi.org/10.1109/CLOUD.2011.42

[11] Frank Thonfeld, Ursula Gessner, Stefanie Holzwarth, Jennifer Kriese, Emmanuel Canova, Juliane Huth, and Claudia Kuenzer. 2022. A First Assessment of Canopy Cover Loss in Germany's Forests after the 2018–2020 Drought Years. *Remote Sensing* 14 (01 2022), 562. https://doi.org/10.3390/rs14030562

[12] Anna Wendleder, Daniel Abele, Martin Huber, Birgit Wessel, Dennis Kaiser, John Truckenbrodt, Peter Friedl, Sandro Groth, Florian Fichtner, and Jonas Eberle. 2023. Sentinel-1 Normalized Radar Backscatter processing on the High-Performance Data Platform terrabyte. In *IGARSS 2023*. https://elib.dlr.de/196780/

[13] Da Yu Xu, Shan Lin Yang, and Ren Ping Liu. 2013. A mixture of HMM, GA, and Elman network for load prediction in cloud-oriented data centers. *Journal of Zhejiang University: Science C* 14, 11 (Nov. 2013), 845–858. https://doi.org/10.1631/jzus.C1300109

[14] Qi Zhang, Mohamed Faten Zhani, Shuo Zhang, Quanyan Zhu, Raouf Boutaba, and Joseph L. Hellerstein. 2012. Dynamic Energy-Aware Capacity Provisioning for Cloud Computing Environments. In *Proceedings of the 9th International Conference on Autonomic Computing* (San Jose, California, USA) *(ICAC '12)*. Association for Computing Machinery, New York, NY, USA, 145–154. https://doi.org/10.1145/2371536.2371562