k-Dispatch: Enabling Cost-Optimized Biomedical Workflow Offloading

Marta Jaros
Faculty of Information Technology,
Brno University of Technology
Brno, Czech Republic
jarosmarta@fit.vutbr.cz

Jiri Jaros
Faculty of Information Technology,
Brno University of Technology
Brno, Czech Republic
jarosjir@fit.vutbr.cz

ABSTRACT

Automated execution of computational workflows has become a critical issue in achieving high productivity in various research and development fields. Over the last few years, workflows have emerged as a significant abstraction of numerous real-world processes and phenomena, including digital twins, personalized medicine, and simulation-based science in general. k-Dispatch is a novel tool designed for the efficient offloading of biomedical workflows to remote high-performance computing clusters or cloud. In addition to data transfers, reporting, error handling and remote computations monitoring, k-Dispatch leverages a set of optimizations to dynamically determine suitable execution parameters for individual tasks within workflows, aiming to meet predefined constraints and optimization criteria. k-Dispatch has been successfully deployed within k-Plan, an advanced modelling tool for planning transcranial ultrasound stimulation (TUS) procedures.

CCS CONCEPTS

• **Computing methodologies** → **Planning with abstraction and generalization**.

KEYWORDS

Workflow scheduling, Multi-criteria optimization, HPC, Cloud

ACM Reference Format:
Marta Jaros and Jiri Jaros. 2024. k-Dispatch: Enabling Cost-Optimized Biomedical Workflow Offloading. In *The 33rd International Symposium on High-Performance Parallel and Distributed Computing (HPDC '24), June 3–7, 2024, Pisa, Italy*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3625549.3658828

1 INTRODUCTION

Biomedical ultrasound plays a crucial role in cancer diagnosis and treatment, offering non-invasive solutions with fewer complications compared to traditional methods like biopsy or surgery. It encompasses various applications such as tissue ablation, drug delivery, and neurostimulation for conditions like epilepsy and Alzheimer's disease [7]. To tailor therapeutic ultrasound procedures to individual patients, evaluating complex physical models beforehand is crucial. The k-Wave toolbox [5], an open-source platform widely adopted in the scientific community, facilitates time-domain simulation of acoustic waves in tissues across different computing architectures, including distributed high-performance computing (HPC) clusters.

In managing complex workflows and their distribution across computational resources, workflow management systems (WMSs) like Pegasus and NextFlow have emerged [1]. These systems aim to automate and streamline processes, provide remote computation monitoring and logging, and offer user-friendly interfaces for interaction with computing facilities. This paper specifically addresses the efficient execution of complex ultrasound workflows on large-scale distributed computing clusters, focusing on optimizing resource allocation for each task within workflows.

The k-Wave toolbox implements its physical models as distributed and moldable programs, where users specify execution parameters based on program scaling, input data size, and cluster utilization. However, achieving perfect performance scaling is challenging, and computational costs may increase with resource utilization. Additionally, scaling behavior depends heavily on processor architecture and memory subsystems. Moreover, data collected on one system may not be easily transferable to others.

To address these challenges, this paper introduces a novel approach that utilizes historically collected performance data to estimate suitable execution parameters for each task within workflows at a global level. This approach aims to optimize resource allocation, preventing system failures and long waiting times in computational queues, thus enhancing the efficiency of batch schedulers on remote computing facilities.

2 OPEN PROBLEMS

Workflow scheduling is a complex and challenging problem. The execution of many tasks on distributed and heterogeneous computing systems needs to be precisely coordinated as it affects results delivery time, workflow makespan (total execution time including queuing times) and cost. A list of selected open problems and challenges follows:

- **Heterogeneous computing resources.** This includes challenges such as load balancing, data transfer optimization, and resource allocation. Computing platforms differ in computing power and capacity, data storage, and network connectivity.
- **Real-time decision making.** When planning the executions of workflows, many factors need to be considered, e.g., task input data and related parameters, type and amount of requested resources, current availability and utilization of

HPDC '24, June 3–7, 2024, Pisa, Italy

ACM ISBN 979-8-4007-0413-0/24/06
https://doi.org/10.1145/3625549.3658828

computing facility, time and cost constraints, and so on. The decision must be made promptly before the situation at the facility has changed.

- **Quality of Service (QoS).** Workflow execution and scheduling are supposed to provide some guarantees about the QoS including throughput, reliability and fault tolerance. Proper monitoring of executed workflows and remote computing facilities is essential for QoS. Although the process of monitoring itself may be straightforward, the challenge is to properly recognize and handle uncommon and suspicious situations.
- **Security.** Workflow executions have to ensure the security and privacy of workflow data and computations, especially when involving sensitive data, such as personal health information or financial transactions.
- **Multi-objective optimization.** To balance multiple objectives, such as execution time, computational cost, energy consumption, and resource utilization, workflow scheduling algorithms face several challenges. It is usually necessary to build a performance database and employ different optimization algorithms and heuristics to operate over this database, and finally perform, e.g., Pareto optimization or trade-off analysis.

Considerations for users' challenges are important. Depending on their expertise, users may face various difficulties throughout the workflow execution process, from constructing the workflow to managing data transfers, execution, and monitoring. Constructing the workflow involves defining inputs, outputs, and dependencies, along with cluster-specific execution parameters. Submission is the most crucial point where a low-level knowledge of executed programs may be required and must be performed thoroughly.

3 SOFTWARE ARCHITECTURE

k-Dispatch [4] provides HPC-as-a-service, ensuring automated and failure-free job execution with advanced planning. Unlike other WMSs, it targets users without prior computational science expertise. Offering biomedical workflows, k-Dispatch handles execution offloading, planning, and monitoring, alongside supporting mechanisms like accounting, reporting, file transfers, and fault tolerance. Deployed within k-Plan for transcranial ultrasound stimulation (TUS) planning, it utilizes remote Czech and UK computational resources.

The architecture of k-Dispatch, illustrated in Fig. 1, consists of three main modules: the Web server, Dispatch database, and Dispatch core. User applications, such as medical GUIs or web apps, communicate with the Web server via HTTPS and REST API. The Dispatch database contains information about users, submitted workflows, computational resources, available executable binaries and permissions, including performance data crucial for workflow planning, HPC selection, and accounting. The Dispatch core is pivotal, managing workflow planning, execution, and monitoring, and communicating with HPC (PBS and Slurm-based) and cloud facilities (Amazon Web Services, Google Cloud) via SSH and RSYNC protocols.

Developed under clinical standards, k-Dispatch adheres to strict rules, undergoes thorough testing, and is versioned in a GitLab repository. Only certified binaries are allowed in workflows, and all data is anonymized and temporarily stored as needed.

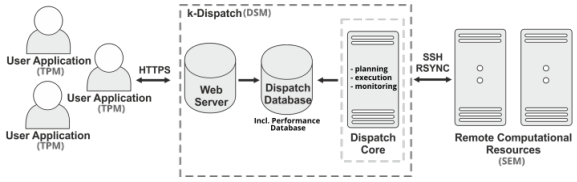

Figure 1: Simplified architecture of k-Dispatch (DSM, Dispatch Server Module) showing its three essential modules - Web Server, Dispatch Database including Performance Database, and Dispatch Core, and their connection to user applications (TPM, Treatment Planning Module) and computational resources (SEM, Simulation Execution Module).

Although being proposed for a set of predefined biomedical workflows, k-Dispatch is not limited to them. k-Dispatch is based on modular design and implements a so-called plug&execute approach. It allows easy extensions by adding (1) new computational workflows, (2) task execution planning strategies and algorithms, and (3) computing resources. Since k-Dispatch provides a simple and well-documented interface in the form of Python virtual classes, new functionality can be added by employing inheritance and polymorphism.

4 OPTIMIZATION OF EXECUTION PARAMETERS

Before conducting experiments, we collected extensive performance data covering various simulation sizes and execution parameters, including conventional production domain sizes (ranging from 512^3 to 1024^3 with low prime factors) and unconventional domain sizes with high prime factors and suboptimal workload distribution across ranks.

Optimization of execution parameters stands on a dynamically updated performance database and four performance modules: (1) *Optimizer* selects suitable execution parameters based on data from the performance database using techniques such as genetic algorithms or simulated annealing. (2) *Estimator* is invoked if the collected performance data is incomplete or missing for a given input. (3) *Evaluator* regularly assesses candidate workflows based on predefined constraints and optimization criteria using a cluster simulator. Finally, (4) *Collector* updates the performance database after each successful completion of the workflow calculation.

For multi-objective optimization, we designed three fitness functions: (a) *Local (task-level) optimization*, minimizing the execution time per task and considering only one optimization criterion. The overall execution time (makespan) may not be the fastest one due to longer waiting times for free resources. (b)) *Global (workflow-level) Optimization using on-Demand resource Allocations (GODA)* balancing workflow makespan and cost with a trade-off coefficient. This approach finds time or cost-constrained solutions as well as differently balanced solutions and is suitable for HPC centres. Finally,

(c) *Global (workflow-level) Optimization using Static resource Allocations (GOSA)*, minimizing latency without a trade-off coefficient, suitable for cloud environments. [2]

Experimental results show that workflows with 64 tasks can be optimized within 15 s with a 95% success rate using local optimization, and within a minute using global optimization approaches.

The fitness function is encapsulated by the GA. The fitness function is, however, universal enabling the GA to be easily replaced by different optimization methods. The encapsulating method represents a black box inside which the execution parameters are selected based on the task input and collected performance data in history. To obtain consistent results, all experiments were performed under a static allocation of 64 computational nodes on IT4Innovations' clusters[1]. The results were compared against the ALEA cluster simulator and a custom one-pass cluster simulator called Tetrisator [3].

In Estimator, we used linear (LI) and quadratic interpolation (QI) methods to estimate the missing values. LI achieved promising results within an already seen single domain, achieving errors as low as 4% on the training dataset but performance diminished considerably on validation and testing datasets with errors escalating to nearly 25%. QI managed to achieve a more consistent error rate of 10.5% across unseen domain sizes. Both LI and QI were selected for k-Wave codes as they have $O(n \log n)$ time complexity.

Therefore, we designed two other approaches using symbolic regression (SR) and artificial neural networks (ANN) to estimate the execution time of a single k-Wave task.

We utilized HeuristicLab [6], a framework for heuristic and evolutionary algorithms, to train SR models. Custom operators tailored to our workflows, including modulo operations and rounding functions, were developed to extend HeuristicLab's capabilities. To validate the models, we employed a validation testing set to evaluate performance on previously unseen data. Due to the time-intensive nature of the training procedure, we utilized the Barbora HPC cluster, where training a single parameter configuration typically took between 10 and 20 hours.

The ANN utilizes the Tensorflow library within the Google Colab environment. Data entering the ANN is first normalized using the Normalization layer to adjust the mean to 0 and the standard deviation to 1. The network architecture consists of a first layer with 12 neurons, followed by six repetitions of a fully connected layer with the ReLU activation function and an active Dropout layer. The final layer is a fully connected layer with one neuron, representing the predicted output. This architecture was finalized after extensive experimentation with hyperparameters and network architectures.

The experiments revealed a complementary relationship between both models. The SR model demonstrated impressive accuracy, especially in scenarios optimized for the k-Wave toolbox, achieving an average error of 5.64%. In broader situations where domain sizes deviated from this policy, the trained ANN displayed acceptable predictive capability, with an error margin of 8.25%.

5 CONCLUSIONS

This research notably improves the prediction of execution time for distributed ultrasound simulations, emphasizing the correlation between simulation size and resource allocation. Our genetic algorithm-based optimization approach utilizes four performance modules with the performance database. Central to multi-objective optimization is the fitness function, for which we designed three variations addressing diverse optimization criteria and computing resources. Furthermore, we explored three methods to estimate missing data in the performance database, assessing their efficacy.

In summary, our findings lay a strong foundation for future developments in execution time prediction, with potential applications beyond ultrasound simulations. Notably, the approach is adaptable. While interpolation methods are straightforward and can be accurate, their error rate fluctuates. The Neural Network is preferred for blind predictions, whereas the Symbolic Regression model excels when specific factors like domain size factorization, node occupancy, and load balance influence execution time.

Moving forward, integrating these modules into k-Dispatch will enable fully automated optimization for ultrasound workflows, greatly enhancing efficiency and resource management.

6 ACKNOWLEDGMENTS

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254). This project has received funding from the European Unions Horizon Europe research and innovation programme under grant agreement No 101071008. This work was supported by Brno University of Technology under project number FIT-S-23-8141.

REFERENCES

[1] Ewa Deelman, Karan Vahi, Mats Rynge, Rajiv Mayani, Rafael Ferreira da Silva, George Papadimitriou, and Miron Livny. 2019. The Evolution of the Pegasus Workflow Management Software. *Computing in Science & Engineering* 21, 4 (2019), 22–36. https://doi.org/10.1109/MCSE.2019.2919690

[2] Marta Jaros and Jiri Jaros. 2023. Optimization of Execution Parameters of Moldable Workflows under Incomplete Performance Data. In *Job Scheduling Strategies for Parallel Processing. JSSPP 2022 (Lecture Notes in Computer Science, Vol. 13592)*. Springer Nature Switzerland AG, 152–171. https://doi.org/10.1007/978-3-031-22698-4_8

[3] Marta Jaros, Dalibor Klusacek, and Jiri Jaros. 2020. Optimizing Biomedical Ultrasound Workflow Scheduling Using Cluster Simulations. In *Job Scheduling Strategies for Parallel Processing. JSSSP 2020* (New Orleans, US) *(Lecture Notes in Computer Science, Vol. 12326)*. Springer Nature Switzerland AG, 68–84. https://doi.org/10.1007/978-3-030-63171-0_4

[4] Marta Jaros, E. Bradley Treeby, Panayiotis Georgiou, and Jiri Jaros. 2020. k-Dispatch: A Workflow Management System for the Automated Execution of Biomedical Ultrasound Simulations on Remote Computing Resources. In *Proceedings of the Platform for Advanced Scientific Computing Conference, PASC 2020* (New York, US). Association for Computing Machinery, 1–10. https://doi.org/10.1145/3394277.3401854

[5] Bradley E Treeby and Ben T Cox. 2010. k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave-fields. *Journal of Biomedical Optics* 15, 2 (2010), 21314.

[6] Stefan Wagner, Gabriel Kronberger, Andreas Beham, Michael Kommenda, Andreas Scheibenpflug, Erik Pitzer, Stefan Vonolfen, Monika Kofler, Stephan Winkler, Viktoria Dorfer, and Michael Affenzeller. 2014. *Advanced Methods and Applications in Computational Intelligence*. Topics in Intelligent Engineering and Informatics, Vol. 6. Springer, Chapter Architecture and Design of the HeuristicLab Optimization Environment, 197–261.

[7] Yu-Feng Zhou, Ali Syed Arbab, and Ronald Xiaorong Xu. 2011. High intensity focused ultrasound in clinical tumor ablation. *World journal of clinical oncology* 2, 1 (2011), 8–27. https://doi.org/10.5306/wjco.v2.i1.8

Received 21 March 2024; revised 19 April 2024; accepted 12 April 2024

[1] https://www.it4i.cz/en/infrastructure/barbora