

SOME ASPECTS OF THE WORKFLOW SCHEDULING IN THE COMPUTING CONTINUUM SYSTEMS

V. Kashansky^{1,a}, R. Prodan¹, G. Radchenko²

¹ *Institute of Information Technology, University of Klagenfurt, Austria*

² *Silicon Austria*

E-mail: ^avladislav.kashanskii@aau.at

Contemporary computing systems are commonly characterized in terms of data-intensive workflows, that are managed by utilizing large number of heterogeneous computing and storage elements interconnected through complex communication topologies. As the scale of the system grows and workloads become more heterogeneous in both inner structure and the arrival patterns, scheduling problem becomes exponentially harder, requiring problem-specific heuristics. Despite several decades of the active research on it, one issue that still requires effort is to enable efficient workflows scheduling in such complex environments, while preserving robustness of the results. Moreover, recent research trend coined under term "computing continuum" prescribes convergence of the multi-scale computational systems with complex spatio-temporal dynamics and diverse sets of the management policies. This paper contributes with the set of recommendations and brief analysis for the existing scheduling algorithms.

Keywords: scheduling, algorithms, brief review, workflows

Vladislav Kashansky, Radu Prodan, Gleb Radchenko

Copyright © 2021 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Introduction

Recent advancements [1-2] in the field of parallel and distributed computing led to the definition of the computing continuum [3] as the environment incorporating highly heterogeneous systems with dynamic spatio-temporal organizational structures [4], varying in-nature workloads (Fig 1-a), complex control hierarchies [5], governing computational clusters with multiple scales of the processing latencies, and diverse sets of the management policies [6-7]. The emergence of these systems is the natural response to the ever-growing variability of computational demands. However, architecting of the algorithms in such environments, e.g. task schedulers, I/O schedulers and resource scalers, is affected by the high degree of uncertainty in relation to the future operational conditions and suffers from tractability issues.

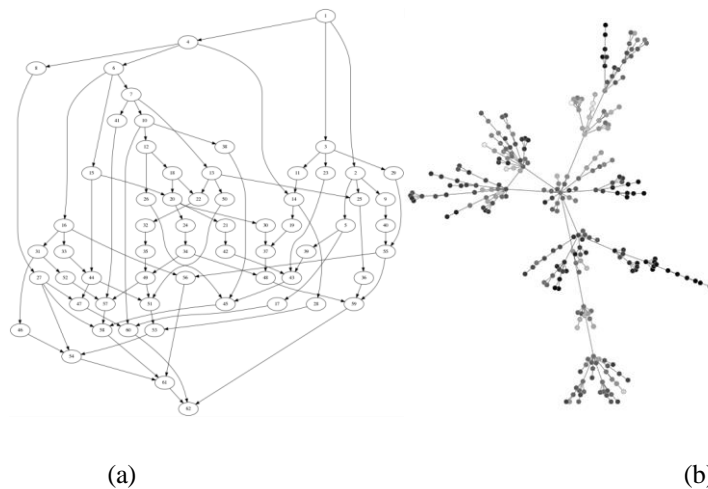


Figure 1. Examples of networks studied in one of the last works [8]. (a) - directed acyclic graph of 60 tasks, rendered und library PSLIB. (b) is a network of 256 agents of the computational continuum with a scale-free topology [9].

As the scale of the system grows and the workloads become more heterogeneous in the inner structure and the arrival patterns, scheduling problem becomes exponentially harder, requiring problem-specific heuristics. This paper contributes with the set of recommendations and brief analysis for the existing scheduling algorithms. Due to the lack of space we address reader interested in theoretical aspects and definition of the computing continuum to the works [3,8].

2. Scheduling Methods

Due to the NP-hardness [11], the large varieties of algorithms were proposed. We summarized some of those in the Table 1. To begin with, one possible approach involves problem reformulation in order to have a more flexible and less computationally demanding control structures [7]. List-scheduling methods (Tab.1 #1, 2) do not provide full knowledge on the workflow execution, by effectively skipping scheduling phase, and replace it with the simplest possible ranking and matching policies [10]. Such methods typically run very fast in polynomial time, however precision is the major problem in this case. It is important to highlight, that those approaches are de-facto standard in the case of the large-scale computational systems / workflows and maintain significant robustness with low resource consumption. Further, non-stationary aspects (e.g. price, performance, reliability prediction etc.) of the computational network are normally out of scope in this approach and can be incorporated via various averaging techniques. More specifically, in case we have set of *identified* ordinary differential equations or hybrid stochastic/difference equations, we can propagate dynamics further in time, but use only averaged values in the ranking and matching phases. Interruptions are not considered here, since schedule is not generated at all.

Table 1. Comparative summary on the various workflow scheduling algorithms

<i>Method</i>	<i>Complexity</i>	<i>Precision</i>	<i>Dynamic Aspects</i>	<i>System Type</i>	<i>Int.</i>	<i>Stoch. Aspects</i>	<i>Based On</i>	<i>Complete Schedule</i>	<i>Method Type</i>
1. HEFT & Variations	Low	Low	Averaging	Not Important	No	Averaging	Recursive Ranking [10,7]	No	Heuristic
2. CPOP & Variations	Low	Low	Averaging	Not Important	No	Averaging	Critical Path [10,7]	No	Heuristic
3. 2P-SGS	Medium	Medium	Averaging	Discrete	No	Monte Carlo	Direct Graph Propagation [11]	Yes	Heuristic
4. GQAP ¹	High	Medium	Mean Field	Not Important	No	Averaging	Quadratic MIP Precedence Relaxation [12]	No	Heuristic
5. GAP	High	Medium	Mean Field	Not Important	No	Averaging	Recursive Ranking / GAP ² Relaxation [13]	No	Heuristic
6. 2P-DYNA	High	Very High	ODE Formulation	Continuous / Mixed	Yes	Monte Carlo	Pontryagin's Principle [14] (FDT0, FOTD)	Yes	Exact
7. MINLP Family	Very High	Very High	Any	Mixed	Yes	Monte Carlo	Direct / MILP Relaxation Solution [15]	Yes	Exact

Generally speaking, we would call these approaches *highly entropic*, since here we are dealing with coarse-grained processes. However there is still a very intriguing research question on how to integrate simple HEFT-like ranking policies [10] and FCFS insertion-based policies with dynamics propagation at different time-scales.

Another two methods, namely GQAP [12] and GAP [13] are based on the quadratic and linear precedence relaxation in MIP³ formulation. Precedence constraints, prescribed by the directed acyclic graph (DAG), are not reflected in the equations. These approaches can be effectively combined with previous two and offer global picture, when performing matching of the tasks to the partitions of machines. Again, applicability of these techniques significantly depends on the structure of the optimization objectives, as quadratic and linear relaxations can inadequately model minimum makespan problems with the workflow structure. It contrasts, for example, with cost optimization, which can be independent of the task running times. Dynamic aspects and interruptions are not considered, because schedule is not generated. One of the drawbacks here is that unbalanced centralized optimization approach with conventional optimization packages CPLEX or SCIP becomes impractical for large-scale systems, due to the large number of variables, delays and information exchange volumes. It leads to huge computational and communication costs. An open question to investigate is how to define static/dynamic relationships between these controllers and groups of controlled agents. Therefore, several hybrid schemes have been proposed to solve large-scale problems, allowing to divide a complex problem into several less complex subtasks. Distributed hierarchical system management strategies are expected to allow sub-optimal performance of control systems in large-scale environments, using the principles of locality while balancing the performance of the whole system, thus enabling scalable infrastructures.

1 GQAP – Generalized Quadratic Assignment Problem
 2 GAP – Generalized Assignment Problem
 3 MIP – Mixed-Integer Programming

2P-SGS is a polynomial heuristic method in MRCPSP notation [11] which takes an intermediate place as it allows to compute complete schedule in two phases. First phase prescribes matching of the tasks to machines and second phase performs activity sequencing subject to the precedence and resource constraints. Algorithm works for discrete execution times and normally suitable for medium scales, as complexity of the SGS phase is $O(nk)$, where n is the number of tasks and k is the number of resource constraints. More information on that approach can be found in [11].

We call 2P-DYNA family of the exact algorithms based on the ODE formulation and optimal control theory principles [14-16]. Interruptions can be considered here, however discretization intervals must allow that. Non-stationary aspects (e.g. price, performance, reliability prediction etc.) of the computational network can be modeled very well in this approach and can be incorporated also with various averaging techniques. Both FDTO⁴ and FOTD⁵ formulations are possible, but each will lead to the different family of methods, including solution of the TPBVP⁶ and MIP [15, 16]. The general possible approach to optimization at the planning horizon is to implement it by solving the MILP or MINLP problem (Tab. 1. #7) with appropriate direct, heuristic and metaheuristics methods. Book [17] contains excellent review of the various mathematical models, including exact and heuristic methods. Both methods #6 and #7 offer very high precision, but also pose extreme difficulty to solve. Normally, attempt to find exact optimum within these techniques can be used in frames of Monte Carlo supercomputing simulations of the complex multilayered computing continuum networks.

4. Conclusion

In this paper we attempted to summarize the set of recommendations and brief analysis for the existing scheduling algorithms. Ideally this paper will help beginners in the field to get on track and select corresponding research direction. It can also serve as the reference point.

5. Acknowledgement

This work has been supported by ADAPT Project funded by the Austrian Research Promotion Agency (FFG) under grant agreement No. 881703 and ASPIDE Project funded by the European Union's Horizon 2020 Programme (H2020) under grant agreement No. 801091.

4 FDTO – First Discretize Then Optimize

5 FOTD – First Optimize Then Discretize

6 TPBVP – Two Point Boundary Value Problem

References

- [1] Reed, D.A. and Dongarra, J., 2015. Exascale computing and big data. *Communications of the ACM*, 58(7), pp.56-68.
- [2] Asch, M., et al., 2018. Big data and extreme-scale computing: Pathways to convergence-toward a shaping strategy for a future software and data ecosystem for scientific inquiry. *The International Journal of High Performance Computing Applications*, 32(4), pp.435-479.
- [3] Beckman, P., Dongarra, J., Ferrier, N., Fox, G., Moore, T., Reed, D. and Beck, M., 2020. Harnessing the computing continuum for programming our world. *Fog Computing: Theory and Practice*, pp.215-230.
- [4] Kashansky, V., et al. 2021, September. The ADAPT Project: Adaptive and Autonomous Data Performance Connectivity and Decentralized Transport Network. In *Proceedings of the Conference on Information Technology for Social Good* (pp. 115-120).
- [5] Copil, G., Moldovan, D., Truong, H.L. and Dustdar, S., 2013, December. Multi-level elasticity control of cloud services. In *International Conference on Service-Oriented Computing* (pp. 429-436). Springer, Berlin, Heidelberg.
- [6] Reuther, A., et al. 2018. Scalable system scheduling for HPC and big data. *Journal of Parallel and Distributed Computing*, 111, pp.76-92.
- [7] Ilyushkin, A., Ali-Eldin, A., Herbst, N., Papadopoulos, A.V., Ghit, B., Epema, D. and Iosup, A., 2017, April. An experimental performance evaluation of autoscaling policies for complex workflows. In *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering* (pp. 75-86).
- [8] Kashansky, V., Radchenko, G. and Prodan, R., 2021, June. Monte Carlo Approach to the Computational Capacities Analysis of the Computing Continuum. In *International Conference on Computational Science* (pp. 779-793). Springer, Cham.
- [9] Albert, R. and Barabási, A.L., 2002. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1), p.47.
- [10] Topcuoglu, H., Hariri, S. and Wu, M.Y., 2002. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE transactions on parallel and distributed systems*, 13(3), pp.260-274.
- [11] Kolisch, R. and Hartmann, S., 1999. Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis. In *Project scheduling* (pp. 147-178). Springer, Boston, MA.
- [12] Lee, C.G. and Ma, Z., 2004. The generalized quadratic assignment problem. *Research Rep., Dept., Mechanical Industrial Eng., Univ. Toronto, Canada*, p.M5S.
- [13] Cattrysse, D.G. and Van Wassenhove, L.N., 1992. A survey of algorithms for the generalized assignment problem. *European journal of operational research*, 60(3), pp.260-272.
- [14] Dolgui, A., Ivanov, D., Sethi, S.P. and Sokolov, B., 2019. Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications. *International Journal of Production Research*, 57(2), pp.411-432.
- [15] Moiseev, N. N. 1974. *Element of the Optimal Systems Theory*. Moscow: Nauka (in Russian).
- [16] Zimin, I.N. and Ivanilov, Y.P., 1971. Solution of network planning problems by reducing them to optimal control problems. *USSR Computational Mathematics and Mathematical Physics*, 11(3), pp.113-124.
- [17] Brucker P, Knust S (2012) *Complex scheduling*, 2nd edn. Springer, Berlin. DOI: <https://doi.org/10.1007/978-3-642-23929-8>