Decentralized optimization

Geir Horn

29 April 2023

1 Motivation

Cloud applications are distributed applications consisting of a set of communicating components that can be software modules, or microservices provided by a set of containers, or stateless functions and third-party software services invoked as needed. By definition, computing is about transforming input data to output data, and the components receives data from sensors or other components, and forwards the results of the computations to downstream components. Application management is interesting for persistent applications that are supposed to run continuously as long as there is more data to process. This means that during the application's lifetime it will experience varying demands and a changing execution context defined by events external to the application and beyond its control.

The goal of the automatic application management is to react to these changes and make rational choices regarding its deployment to maximize the application owner's utility [1]. Application management in a pure Cloud computing setting is simplified by the promise of infinite elasticity of the Cloud resources needed by the application. It is always possible to give the application the resources needed for it to maximise its utility. However, this is not possible for applications using computational resources in the network and closer to the sources of the data being acquired and processed by the application. These resources are not abundant, and geographical location may play a role making it hard or impossible to move the application components to more available or better performing resources.

Finally, the Cloud computing paradigm is predominantly a business model prescribing that you can rent virtual resources only for the periods of time when they are needed. The owner of the physical infrastructure hosting the virtual resources can simultaneously let other users rent virtual resources on the same physical infrastructure. The virtual servers can be co-located on the same physical server and reallocated if virtual machines from a user negatively affect the computations made by other users. The principle of geographical locality and limited resources prevent such approaches to be applied for application components deployed on rented servers in the network or at the edge of the network. Hence, the activities of one application can interfere with the operations of other applications and have a severe negative impact on the other applications' performance. Remedy actions taken by the application management of the affected applications can again affect other applications. Thus, there is no way to guarantee that application management is effective in this setting, and there is no gurantee that independent application controllers will converge on a new stable deployment equilibrium of all applications simultaneously running on the shared infrastructure.

2 Current status

Space does not allow a thorough review of different ideas for automatic application management, and the focus will be on integrated results from the European Union funded projects as accessible open source solutions. The project *Multi-cloud Execution ware for Large scale Optimised Data Intensive Computing*[1] (MELODIC) developed a platform where application's components may be hosted by multiple Cloud providers [2]. It works by maximizing the application *utility function* as it is an established way to capture the application owner's utility in autonomic computing [3]. The utility function, $U(\boldsymbol{c} \,|\, \boldsymbol{\theta}(t_k), \boldsymbol{\psi}(\boldsymbol{c} \,|\, \boldsymbol{\theta})) \in [0,1]$, maps a proposed application configuration, $\boldsymbol{c}$, to a utility value in the unit interval given the measurements of the application execution context, $\boldsymbol{\theta}(t_k)$, at the current time, t_k, and the application's performance indicators $\boldsymbol{\psi}(\boldsymbol{c} \,|\, \boldsymbol{\theta}(t_k))$. The utility function may trade off different dimensions and conflicting goals like minimizing the deployment cost while maximizing the application performance. The application is deployed or reconfigured in the configuration that maximises the utility function.

[1] https://melodic.cloud/

The MELODIC platform was extended in the project *Modelling and Orchestrating heterogeneous Resources and Polymorphic applications for Holistic Execution and adaptation of Models In the Cloud*[2] (MORPHEMIC) with polymorphic adaptation where application components may support various hardware accelerators, and with proactive adaptation forecasting the future application execution context, $\hat{\boldsymbol{\theta}}(t_{k+h})$, and regression models for the performance indicators to calculate the estimated utility value at the future time t_{k+h} [4]. Proactive adaptation compensates for the time needed to acquire and configure new resources for the application.

These ideas are currently taken forward in the project to develop *A meta operating system for brokering hyper-distributed applications on Cloud computing continuums*[3] (NebulOuS) extending the automatic control to applications being deployed also onto servers in the network or at the edge. NebulOuS will therefore face the challenge of multiple applications from different tenants competing for the same restricted resources. Essentially, it will have a vector of utility values for the n applications controlled, $[U_1(\boldsymbol{c}_1|\boldsymbol{\theta}, \boldsymbol{\psi}), \ldots, U_n(\boldsymbol{c}_n|\boldsymbol{\theta}, \boldsymbol{\psi})]^T$, where all values are for the same execution context $\boldsymbol{\theta}(t_k)$ and the arguments are omitted on the performance indicators for brevity. It may not be possible to increase the utility value for one application without deteriorating the utility value of the others, *i.e.*, the utility value vector is a point on the Pareto front of the multi-objective optimization problem. These problems are typically addressed with evolutionary algorithms [5]. However, such algorithms must run in a central location, and they have exponential time complexity with growing system size, n, and it is therefore a limit on how many applications can be co-optimised within the time available for optimization. Furthermore, the multi-objective optimization implies prioritizing some applications over others, and there is no global criterium to guide this prioritization.

3 Research challenges

Novel approaches must therefore find scalable solutions to combine the flexible and pro-active application management offered by individual application utility maximization with multi-tenant use of resource constraint servers. The suggested way forward is to abandon the centralised planning and rather let each application optimise its own utility by local decisions made by the application components. This resembles the way car drivers individually makes their route planning maximising the utility of fast arrivals, while constantly taking into consideration the current traffic situation.

Distributed optimisation where multiple agents collaborate on solving a joint optimisation problem are known to converge for linear and convex problems [6]. The Distributed Constraint Optimization Problem (DCOP) [7] is an alternative view where the agents control variables and local, multivariate cost functions to jointly minimize a global objective function, which is the sum of the local cost functions. There are also many algorithms for the similar case where the cost functions are univariate, and the agent communication is restricted by the communication network graph [8]. Furthermore, there is huge corpus of research results for achieving collective behaviour in multi-agent systems [9].

None of these approaches satisfies the need for decentralized algorithms for optimization. There are few approaches in this direction [10]. Furthermore, the multi-agent solutions available are developed for continuous decision variables whereas the location and capacity assignment problems in computing are typically discrete. Only a handful of directions have been proposed for decentralized combinatorial optimization. Pournaras *et al.* recently proposed an algorithm for Iterative Economic Planning and Optimized Selections (I-EPOS) [11]. The problem at hand can also be seen as a grouping of components from multiple tenants on a set of servers, *i.e.*, decentralized coalition structure generation [12]. The problem may also be addressed by a decentralized stochastic game of reinforcement learning automata [13]. A challenge in all iterative learning based solutions is the relatively slow convergence, and one should investigate if it is possible to use actor-critic implementations [14], or decentralize the Distributed Policy Optimzation (DPO) algorithm [15]. More research is definitely needed to enable fully decentralized application management.

Acknowledgments

This work has received funding from the European Union grant agreement 101070516 NebulOuS: *A meta operating system for brokering hyper-distributed applications on Cloud computing continuums*. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Horizon Europe. Neither the European Union nor the granting authority can be held responsible for them.

[2] https://www.morphemic.cloud/

[3] https://www.nebulouscloud.eu/

References

[1] Itzhak Gilboa, *Rational Choice*. Cambridge, MA, USA: MIT Press, 2010, XVIII, 158, ISBN: 978-0-262-01400-7.

[2] Geir Horn and Paweł Skrzypek, "Melodic: Utility based cross cloud deployment optimisation," in *Proceedings of the 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Conference Location: Krakow, Poland: IEEE Computer Society, May 16–18, 2018, pp. 360–367. DOI: 10.1109/WAINA.2018.00112.

[3] Jeffrey O. Kephart and Rajarshi Das, "Achieving self-management via utility functions," *IEEE Internet Computing*, vol. 11, no. 1, pp. 40–48, Jan. 2007, ISSN: 1089-7801. DOI: 10.1109/MIC.2007.2.

[4] Marta Różańska and Geir Horn, "Proactive autonomic cloud application management," in *Proceedings of the 15th IEEE/ACM International Conference on Utility and Cloud Computing (UCC2022)*, Conference Location: Vancouver, Washington, USA: IEEE/ACM, Dec. 6–9, 2022, pp. 102–111, ISBN: 978-1-66546-087-3. DOI: 10.1109/UCC56403.2022.00021.

[5] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Second edition, ser. Genetic and Evolutionary Computation Series. Boston, MA: Springer US, 2007, XXI, 800, ISBN: 978-0-387-33254-3. DOI: 10.1007/978-0-387-36797-2.

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jul. 2011, ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000016.

[7] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh, "Distributed constraint optimization problems and applications: A survey," *Journal of Artificial Intelligence Research*, vol. 61, pp. 623–698, Mar. 2018, ISSN: 1076-9757. DOI: 10.1613/jair.5565.

[8] Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, Jan. 2019, ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2019.05.006.

[9] Federico Rossi, Saptarshi Bandyopadhyay, Michael T. Wolf, and Marco Pavone, "Multi-agent algorithms for collective behavior: A structural and application-focused atlas," Mar. 2021. arXiv: 2103.11067 [cs].

[10] Huiwei Wang, Huaqing Li, and Bo Zhou, *Distributed Optimization, Game and Learning Algorithms: Theory and Applications in Smart Grid Systems*, First edition. Singapore: Springer, 2021, XVII, 217, ISBN: 978-981-334-528-7. DOI: 10.1007/978-981-33-4528-7.

[11] Evangelos Pournaras, Peter Pilgerstorfer, and Thomas Asikis, "Decentralized collective learning for self-managed sharing economies," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 13, no. 2, 10:1–10:33, Nov. 2018, ISSN: 1556-4665. DOI: 10.1145/3277668.

[12] Jörg Bremer and Sebastian Lehnhoff, "Decentralized coalition formation with agent-based combinatorial heuristics," *Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 6, no. 3, pp. 29–44, Sep. 2017, ISSN: 2255-2863. DOI: 10.14201/ADCAIJ2017632944.

[13] Omkar Tilak, Ryan Martin, and Snehasis Mukhopadhyay, "Decentralized indirect methods for learning automata games," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 5, pp. 1213–1223, Oct. 2011, ISSN: 1083-4419. DOI: 10.1109/TSMCB.2011.2118749.

[14] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," Feb. 27, 2018. arXiv: 1802.08757 [cs, math, stat].

[15] Kefan Su and Zongqing Lu. "Decentralized policy optimization." arXiv: 2211.03032 [cs]. (Nov. 2022), (visited on 03/15/2023), preprint.