Applying Genetic Algorithms to Resource Allocation in Cloud Computing: A Review

Andrew Pozzuoli

COSC 5P07

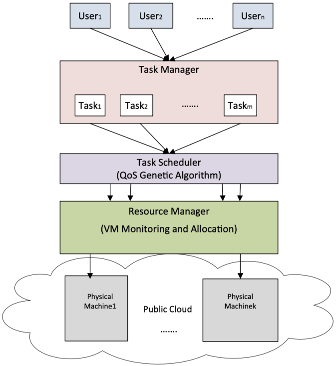
March 17, 2021

Overview

Cloud Computing

- Computing, database, and storage resources of a data center provided to clients over the internet via pay-as-you-use policy [1].
- Eliminates the need for installing software on the end user's machine [2].
- There is increasing demand for cloud computing due to the benefits of offloading resources [3].

Figure: Cloud Resource Scheduling Architecture (Devarasetty et al., 2019) [1].



The Problem of Resource Allocation

- There are finite resources that need to be allocated to a number of user machines based on client needs.
- Resource allocation strategies need to balance a high quality of service with low operational costs [3], [4].
- Bin packing problem.
- NP-complete [5].
- A typical cloud consists of hundreds to thousands of physical and virtual servers [6].
- Effective resource allocation improves performance for both the cloud provider and user [4].

The Problem of Resource Allocation

The following issues need to be avoided in order to have optimal resource allocation [4]:

- Resource contention when multiple applications try to access the same resource.
- Resource scarcity when demand increases but resources are finite.
- Resource fragmentation when small resource entities are isolated and application cannot have more resources allocated to them leading to wasted resources.
- Over/under-provisioning of resources.

Proposed Solutions

A number of static scheduling algorithms have been proposed to address this issue [2], [3]:

- First fit
- Round Robin
- Preemption scheduling
- CPU utilization prediction
- Adaptive heuristics

Due to the NP-complete nature of the problem, no one algorithm is capable of easily finding the optimal solution better than any other.

Genetic Algorithm

Genetic algorithms (GAs) evolve a solution over a number of generations using principles from Charles Darwin's theory of evolution and natural selection [2].

- Initial population of random solutions.
- Each solution is ranked according to a fitness function.
- The best solutions are more likely to pass their "genes" along to the next generation.
- Solutions go through crossover where child solutions are generated by mixing parent solutions along with some random mutation.
- The process repeats over many generations until a solution approaches a good-enough optimal solution.

Genetic algorithms are powerful for approaching NP-complete problems since it is able to combine the exploitation of good solutions to approach better solutions as well as use mutation to explore more of the search space and to not get stuck in a local best solution.

Solutions: Elana et al., 2011

In the paper, *Efficient manager for virtualized resource provisioning in Cloud Systems* (Elena et al., 2011), a GA was implemented that optimizes for performance and scalability [7].

- The algorithm considers time, cost, and physical resources
- Takes into account VM capabilities and well defined policies.

The paper found that the scheduling policies used by GA improved the utilization rate when allocating resources to VMs.

Solutions: Arianyan et al., 2012

Efficient Resource Allocation in Cloud Data Centers Through Genetic Algorithm (Arianyan et al., 2012), built off of previous work by taking advantage of genetic algorithm for resource allocation [2].

- GA is able to consider more parameters in its decision than static resource allocation models.
- The importance of parameters was also taken into consideration (weights were assigned to the parameters and used in the fitness function).
- Considering parameter importance had different impacts on the solution. For example, placing more importance on number of VMs led to lower operational costs.

The proposed GA enhanced resource utilization and lowered operational costs.

Solutions: Bakalla et al., 2017

A QoS-Aware and Energy Efficient Genetic Resource Allocation Algorithm for Cloud Data Centers (Bakalla et al., 2017) proposed a GA for resource allocation that took into consideration both energy consumption and quality of service and simulated on real workloads [3].

- Cloud users have Service-Level Agreement contracts with the cloud providers. If not enough resources are provided, there is a Service-Level Agreement Violation (SLAV).
- This paper used GA to maintain high resource utilization while minimizing energy cost and SLAVs.
- 65.82% reduction in energy consumption over NonPowerAware algorithm and similar energy consumption performance to Dynamic Voltage Power Frequency (DVPF) algorithm which is designed for energy efficiency.

Solutions: Devarasetty & Reddy, 2019

Genetic Algorithm for quality of service based resource allocation in cloud computing (Devarasetty & Reddy, 2019) proposed a GA that considered deployment cost and response time as its major parameters [1].

- Compared results to QCost algorithm, a non-GA resource allocation algorithm designed to reduce deployment cost while maintaining a high quality of service.
- The GA outperformed QCost as an efficient balancer of deployment cost and quality of service.

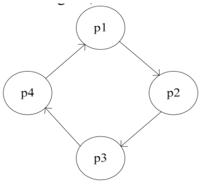
Variations: Zheng et al., 2011

An Approach for Cloud Resource Scheduling Based on Parallel Genetic Algorithm (Zheng et al., 2011), proposed a parallel genetic algorithm (PGA) [6].

- GA is demanding to memory and CPU leading to slow runtime especially as the number of VMs increases.
- PGA separates the population into sub-populations (demes) to evolve in parallel with a chance for individuals to migrate to a different deme.
- PGA improves on the speed of finding a good resource allocation over GA.

- The paper also found that this GA improved the utilization rate over Round Robin and Greedy algorithms.

Figure: Sub-population topology (Zheng et al., 2019) [6].



Variations: Shirali et al., 2019

An extended Multi-Population Genetic Algorithm for Resource Allocation in Service Hosting Platforms enhanced the GA approach to resource allocation and applied it to Service Hosting Platforms [5].

- Same problems as cloud resource allocation (allocating resources to user requests with increasing demand as time goes on).
- Evolve multiple sub-populations separately to use more of the search space thus it is able to explore multiple optima rather than just one.
- If the number of sub-populations is greater than the number of optima then some populations are wasteful.
- The paper used a self-adaptive system for finding the best amount of sub-populations, creating new populations if all others converge on a solution.

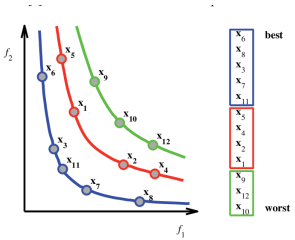
Simulation showed that this method outperformed standard GA since the algorithm is able to track many different optima simultaneously.

Variations: Miriam et al., 2020

Non-dominated Sorting Genetic Algorithm (NSGA-III) for effective resource allocation in cloud proposed a modified non-dominated sorting GA for resource allocation [4].

- Multi-objective functions with conflicting parameters do not have one solution that optimizes everything.
- NSGA sorts solutions into pools that are ranked.
- Modified without crossover to preserve order of VMs in solutions.

Figure: Non-dominated sorting (Kadlec et al., 2013) [8].



The proposed framework effectively allocated resources in any cloud model and outperformed state-of-the-art algorithms.

Conclusion

GA seems to be an effective algorithm for resource allocation in the cloud.

- GA is a non-greedy algorithm for multi-objective optimization.
- Able to consider many more parameters than conventional non-genetic algorithms and search more of the search space.
- Generally, the papers showed that GA enhances resource utilization, reduces deployment cost and power consumption, and maintains a high quality of service.
- Outperforms standard non-GA algorithms for resource allocation.

- Not one example of GA used in real world as a cloud resource allocation method.
- Long runtime since the GA needs to run for many generations to find a good resource allocation. Could run into issues in a dynamic environment.
- Non-deterministic. Difficult to predict allocation and keep it consistent.
- Requires more niche/specialized knowledge compared to standard algorithms.

Due to these problems, despite the benefits GA brings, it does not seem like a feasible industry alternative to standard algorithms yet.

Future Work

Hard to ignore the enhancements that GA brings to the multi-objective problem of resource allocation in the cloud.

To become more useful to the industry, the following should be explored:

- Reducing runtime of GA.
  - Less expensive parallel GA.
  - Smaller population.
  - Reduce number of objectives.
- Evolve an algorithm for resource allocation instead of evolving a solution every time.
  - Genetic Programming.
  - Evolve an algorithm that can be applied to any configuration of users and resources.
- Deploy GA to a real world cloud data center to find problems that would not arise in simulations.
  - As computers become more powerful and less expensive, the runtime problems of GA may vanish.
  - Useful to find unexpected issues before that time comes.

Bibliography I

[1] P. Devarasetty and S. Reddy, "Genetic algorithm for quality of service based resource allocation in cloud computing," en, *Evolutionary Intelligence*, Apr. 2019, ISSN: 1864-5909, 1864-5917. DOI: 10.1007/s12065-019-00233-6. [Online]. Available: http://link.springer.com/10.1007/s12065-019-00233-6 (visited on 02/26/2021).

[2] E. Arianyan, D. Maleki, A. Yari, and I. Arianyan, "Efficient resource allocation in cloud data centers through genetic algorithm," en, in *6th International Symposium on Telecommunications (IST)*, Tehran, Iran: IEEE, Nov. 2012, pp. 566–570, ISBN: 978-1-4673-2073-3 978-1-4673-2072-6 978-1-4673-2071-9. DOI: 10.1109/ISTEL.2012.6483053. [Online]. Available: http://ieeexplore.ieee.org/document/6483053/ (visited on 02/26/2021).

Bibliography II

[3] M. Bakalla, H. Al-Jami, H. Kurdi, and S. Alsalamah, "A QoS-aware and energy-efficient genetic resource allocation algorithm for cloud data centers," en, in *2017 9th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Munich: IEEE, Nov. 2017, pp. 244–249, ISBN: 978-1-5386-3435-6. DOI: 10.1109/ICUMT.2017.8255166. [Online]. Available: http://ieeexplore.ieee.org/document/8255166/ (visited on 02/26/2021).

[4] A. J. Miriam, R. Saminathan, and S. Chakaravarthi, "Non-dominated Sorting Genetic Algorithm (NSGA-III) for effective resource allocation in cloud," en, *Evolutionary Intelligence*, Jun. 2020, ISSN: 1864-5909, 1864-5917. DOI: 10.1007/s12065-020-00436-2. [Online]. Available: http://link.springer.com/10.1007/s12065-020-00436-2 (visited on 02/26/2021).

[5] A. Shirali and M. R. Meybodi, "An extended multi-population genetic algorithm for resource allocation in service hosting platforms," en, in *2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN)*, Qazvin: IEEE, Apr. 2018, pp. 37–45, ISBN: 978-1-5386-5706-5. DOI: 10.1109/RIOS.2018.8406629. [Online]. Available: https://ieeexplore.ieee.org/document/8406629/ (visited on 02/26/2021).

Bibliography IV

[6] Zhongni Zheng, Rui Wang, Hai Zhong, and Xuejie Zhang, "An approach for cloud resource scheduling based on Parallel Genetic Algorithm," en, in *2011 3rd International Conference on Computer Research and Development*, Shanghai, China: IEEE, Mar. 2011, pp. 444–447, ISBN: 978-1-61284-839-6. DOI: 10.1109/ICCRD.2011.5764170. [Online]. Available: http://ieeexplore.ieee.org/document/5764170/ (visited on 02/26/2021).

[7] E. Apostol, I. Băluță, A. Gorgoi, and V. Cristea, "Efficient manager for virtualized resource provisioning in cloud systems," in *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, 2011, pp. 511–517. DOI: 10.1109/ICCP.2011.6047924.

Bibliography V

[8] P. Kadlec, Z. Raida, and J. Dřínovský, "Multi-objective self-organizing migrating algorithm: Sensitivity on controlling parameters," *Radioengineering*, vol. 22, pp. 296–308, Apr. 2013.