Many-Objective Performance Enhancement in Computing Clusters

A s m Rizvi[1], Tarik Reza Toha[2], Siddhartha Shankar Das[3], Sriram Chellappan[4], and A. B. M. Alim Al Islam [5]
[1] Department of Computer Science, University of Southern California, USA
[1, 2, 3, 5]Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh
[4] Department of Computer Science and Engineering , University of South Florida, USA
Email: asmrizvi@usc.edu, 1205082.trt@ugrad.cse.buet.ac.bd, siddhartha047@cse.buet.ac.bd, sriramc@usf.edu, and alim_razi@cse.buet.ac.bd

***Abstract*—In a heterogeneous computing cluster, cluster objectives are conflicting to each other. Selecting a right combination of machines is necessary to enhance cluster performance, and to optimize all the cluster objectives. In this paper, we perform empirical performance analyses of a real cluster with our year-long collected data, formulate a new many-objective optimization problem for clusters, and integrate a greedy approach with the existing NSGA-III algorithm to solve this problem. From our experimental results, we find our approach performs better than existing optimization approaches.**

I. Introduction

Computing clusters are useful for doing extensive research, computation, and parallel computing. There are different performance metrics for computing clusters in the literature. Energy consumption, computation time, cost, and utilization are the common cluster objectives. We consider CPU usage and memory usage for resource utilization, and at the same time, we consider computation energy and cooling energy for energy consumption.

These cluster objectives are sometimes conflicting to each other. We find, from our year-long collected data, that increasing the number of machines can reduce computation time, total energy consumption. At the same time, it can decrease resource utilization, and increase maintenance cost as well.

There exist different techniques to solve multiple conflicting objectives [2]. However, solving conflicting cluster objectives is not focused much in the literature. In this paper, we propose a new approach to solve the conflicting objectives of a computing cluster. We perform a rigorous empirical performance analysis to formulate an optimization problem for clusters. Our solution approach combines both greedy method and NSGA-III algorithm to solve the optimization problem. We finally give an operational set of machines to the cluster administrator which will optimize the conflicting cluster objectives. We compare our approach and other existing approaches to show our approach performs better.

II. Related Work

In cluster computing, several other studies consider performance optimization. The study in [9] presents a stochastic approach for the performance optimization, however, for more than three objectives [1] it does not consider any many-objective optimization approach. Integrating cooling energy consumption is also absent from this study which is an important part of total energy consumption.

This research work has been funded by the ICT Division, Government of the People's Republic of Bangladesh.

Multi-objective optimization techniques are considered in virtual machine based schemes [4], and also in resource provisioning in cloud platform [6]. However, due to significant architectural gap between clusters and clouds we cannot consider these approaches in computing clusters.

There are some studies on multi-objective performance optimization in computing clusters and clouds based on Particle Swarm Optimization (PSO) [5] and Ant Colony Optimization (ACO) approach [3]. These approaches are not used for many-objective optimization problems yet. Moreover, to the best of our knowledge, existing studies do not integrate empirical performance characterization of clusters. Such empirical characterization is useful to reveal impacts of operational factors over cluster performance.

III. Cluster Objectives and Problem Formulation

We define the four cluster objectives - minimizing computation time, minimizing total energy consumption, minimizing cost, and maximizing utilization. We define effectiveness of a machine, Machine Value, MV, which can have value from 0 to 100. We define MV of a machine i by $MV_i = w_1 \times P_1 + w_2 \times P_2 + w_3 \times P_3 + ... + w_N \times P_N$, where $w_1, w_2, w_3, ..., w_N$ are the weights of N properties.

Minimizing computation time: Computation time decreases with the increase in machine value (using high-performing machines). Thus, we define our objective as a part of a minimization problem:

$$min \quad \sum_{i=1}^{N_M} (100 - s_i(MV)_i) \tag{1}$$

Here, N_M is the total number of machines, s_i is a decision variable which is 1 if we select i^{th} machine operational and 0 if we do not select i^{th} machine operational. Computation time also depends on work load - increased when workload gets increased. If the workload for machine i is W_i, then we can write:

$$\min \quad \frac{\sum_{i=1}^{N_M} (100 - s_i(MV)_i)}{\sum_{i=1}^{N_M} s_i \times 100} \times WT_1 + \frac{W_i}{W_{max}(i)} \times WT_2 \tag{2}$$

Here, $W_{max(i)}$ is the maximum allowable workload for machine i, and W_i is the machine workload. WT_1 and WT_2 are the weights for machine value and workload. We take same valued weights in both parts of Eq. 2.

Minimizing total energy consumption: Energy consumption exhibits a decreasing trend with an increase in the number of machines. Besides, if we need to decrease T_{diff} temperature by our cooling devices, then the total energy consumption will increase with the increase of T_{diff} value. We can write the following objective function:

$$\min \quad \frac{\sum_{i=1}^{N_M} s_i}{N_M} \times WT_3 + \frac{T_{diff}}{T_{MaxDiff}} \times WT_4 \tag{3}$$
$$\text{subject to} \quad T_L \leq T_D \leq T_M$$

978-1-5090-6468-7/17/$31.00 ©2017 IEEE

Here, WT_3 and WT_4 indicate weights, and we assume same values for the weights in Eq. 3. $T_{MaxDiff}$ is the maximum possible temperature difference.

Minimizing cost: Cost will be increased with an increase in the machine value:

$$\min \quad \sum_{i=1}^{N_M} s_i(MV)_i \tag{4}$$

Maximizing utilization: Utilization increases with a decrease in the number of machines:

$$\min \quad \frac{\sum_{i=1}^{N_M} s_i}{N_M} \times 100 \tag{5}$$

Note that, if we have $\sum_{i=1}^{N_M} s_i$ operational machines and W_i workload, then there will be $\frac{W_i}{\sum_{i=1}^{N_M} s_i}$ workload per machine. If HDD_i is the hard disk size of machine i , the we have a constraint as $\forall i \frac{W_i}{\sum_{i=1}^{N_M} s_i} \leq HDD_i$. We can write the optimization problem as:

$$min \begin{cases} \frac{\sum_{i=1}^{N_M} (100 - s_i(MV)_i)}{\sum_{i=1}^{N_M} s_i \times 100} \times WT_1 + \frac{W_i}{W_{max(i)}} \times WT_2 \\ \frac{\sum_{i=1}^{N_M} s_i}{N_M} \times WT_3 + \frac{T_{diff}}{T_{MaxDiff}} \times WT_4 \\ \sum_{i=1}^{N_M} s_i(MV)_i \\ \frac{\sum_{i=1}^{N_M} s_i}{N_M} \times 100 \end{cases}$$

$$\textit{subject to} \begin{cases} \forall i \frac{W_i}{\sum_{i=1}^{N_M} s_i} \leq HDD_i \\ T_L \leq T_D \leq T_M \\ \sum_{i=1}^{N_M} s_i(MV)_i > 0 \end{cases}$$

Next, we present a solution approach for the above optimization problem.

IV. Proposed Solution Approach

We have a many-objective optimization problem for cluster and we use NSGA-III [2] as our base algorithm to solve this problem. We use our empirical analysis to design selection, crossover, and mutation functions of NSGA-III algorithm. If there are N_M machines, then we consider N_M number of decision variables. We also use the temperature of cooling devices as a decision variable. Our algorithm finally gives the number of machines that we need to select, a selected set of machines, and a temperature that needs to be maintained by the cooling devices. Now, we are describing the modifications over the functions of NSGA-III.

Population Selection From our empirical analysis, we find that selecting a small number of machines results in a high computation time and energy consumption. We eliminate the population that has fewer operating machines from going into the next generation. We take this threshold value, T_h as $\frac{N_M}{6}$.

Crossover Following a greedy approach, we give bias for high-performing machines when we make crossover. In this way, we can ensure higher chances of their selection in the next generation. We describe the three steps crossover below:

Half uniform crossover: We implement half-uniform crossover [8] as our basic crossover approach. We have binary decision variable, hence, we only make crossover when the particular chromosome value is different from each other.

Crossover based on clustering: We separate N_M machines into two clusters with Cluster 3.0 tool using Euclidean distance in k-means clustering. With some probabilistic condition, we try to take machines from our top-performing cluster and do not try to take machines from the other cluster into our next generation.

Workload	Improvement over PSO (%)			Improvement over ACO (%)		
	Time	Cooling energy	Comp. energy	Time	Cooling energy	Comp. energy
67.7 GB	21	13	10	43	10	5
50.4 GB	36	11	10	17	8	8
28.3 GB	43	13	10	15	5	0

TABLE I: Improvement over PSO and ACO for various workloads in *SimGrid* with 30 machines

Solution Filtering We get a pareto-front with multiple solutions from our algorithm and we take only those solutions which have 25% to 75% values for all objectives. We select one solution for the administrator based on the administrator's defined weight function:

$$F_{total} = W_{obj1} \times V_{obj1} + W_{obj2} \times V_{obj2} + \cdots + W_{objN} \times V_{objN} \tag{6}$$

In Eq. 6, every objective value, V_{obj} should be within 25% to 75% of the corresponding objective value and weights of these objectives will be defined by the cluster administrator. Then we take the solution having the lowest merged objective value as we are solving a minimization problem.

V. Evaluation Results

We evaluate performances of our algorithm and other existing approaches in a simulation platform named *SimGrid*. *SimGrid* is able to calculate both computation energy and cooling energy [7]. We experiment in clusters having 30 machines. Table I shows corresponding improvements achieved by our approach for 30 machines.

VI. Conclusion

This paper aims to solve conflicting objectives in clusters to provide administrators a set of machines. In this paper, we provide a solution for cluster administrators exploiting a synergy between a greedy approach and NSGA-III algorithm. Our approach can be used in real clusters, which will fulfill the cluster objectives.

References

[1] S. Chand and M. Wagner. Evolutionary many-objective optimization: a quick-start guide. *Surveys in Operations Research and Management Science*, 20(2):35–42, 2015.

[2] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation*, 18(4):577–601, 2014.

[3] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *Journal of Computer and System Sciences*, 79(8):1230–1242, 2013.

[4] R. Li, Q. Zheng, X. Li, and J. Wu. A novel multi-objective optimization scheme for rebalancing virtual machine placement. In 9^{th} *IEEE CLOUD*, pages 710–717, 2016.

[5] C. Lijun and L. Xiyin. Modeling server load balance in cloud clusters based on multi-objective particle swarm optimization. *IJGDC*, 8(3):87–96, 2015.

[6] S.S. Manvi and G.K. Shyam. Resource management for infrastructure as a service (iaas) in cloud computing: A survey. *Journal of Network and Computer Applications*, 41:424–440, 2014.

[7] A. Rizvi, T.R. Toha, M.M.R. Lunar, M.A. Adnan, and ABM A. Al Islam. Cooling energy integration in simgrid. In *NSysS*, pages 132–137. IEEE, 2017.

[8] W. M. Spears and V. Anand. A study of crossover operators in genetic programming. In *International Symposium on Methodologies for Intelligent Systems*, pages 409–418. Springer, 1991.

[9] K.M. Tarplee, A.A. Maciejewski, and H.J. Siegel. Robust performance-based resource provisioning using a steady-state model for multi-objective stochastic programming. *IEEE Transactions on Cloud Computing*, 2016.