ISBN 978-981-11-0008-6
Proceedings of 2016 6th International Workshop on Computer Science and Engineering (WCSE 2016)
Tokyo, Japan, 17-19 June, 2016, pp. 453-458
doi: 10.18178/wcse.2016.06.076

Automatic Discovery of IaaS Cloud Workload Types

Piotr Orzechowski[1], Jerzy Proficz[1+], Henryk Krawczyk[2] and Julian Szymański[2]

[1] Academic Computer Center, Gdańsk University of Technology, Poland

[2] Faculty of Electronics, Telecommunication and Informatics, Gdańsk University of Technology, Poland

Abstract. The paper presents an approach to automatic discovery of workloads types. We perform functional characteristics of the workloads executed in our cloud environment, that have been used to create model of the computations. To categorize the resources utilization we used K-means algorithm, that allow us automatically select six types of computations. We perform analysis of the discovered types against to typical computational benchmarks, finding the strong correlation between functional classes and the resource utilization.

Keywords: IaaS, workload discovery, k-means, cloud load prediction,

1. Introduction

One of the recent trends in private infrastructure management is migration from own data centers to clouds. It is caused by more and more expensive infrastructure maintenance and purchase. Clouds provide different models of computation but one of them, Infrastructure of a Service (*IaaS*)[1] is the most popular choice for companies with custom requirements on installed software, quality and security. The model *IaaS* allows users to manage rented infrastructure on their own. Clients specify hardware requirements, define network connections, security policy, and as a result get prepared IT infrastructure. Power management, physical security, cooling etc., which administrators have to keep in mind in own data centers, in IaaS model they are delegated to cloud providers. Moreover scalability of *IaaS* model is better, and what's more important it is simpler then in traditional private data center, but requires to reserve more resources.

On the other hand cloud providers pursue to minimize costs of cloud infrastructure maintenance. Actually, providers put emphasis on improvement resources allocation. There are many methods of scheduling virtual machines depending on the machine size, geographic locations or requirements on specific resources like GPU accelerators. Our research focuses on finding mapping between functional categorization workload types and resource requirements of workloads. These categorization could help providers to accurate schedule workloads based on workload functional type chosen by cloud user.

Workload resource requirements and grouping of workloads by resource usage is popular subject in literature [2] [3] [4]. There are many techniques of allocation and monitoring but most of proposed methods focuses on appropriate scheduling of new jobs only based on other workloads resource utilization.

One of the most popular approach in grouping workloads focuses on resource usage. Proposed classes are: CPU-Intensive, Memory-Intensive, I/O-Intensive, Idle and Network-Intensive [5][6]. In [5] authors present FBWC method (Feedback-Based Workload Classification) which classify workloads to mentioned groups based on 22 metrics. In another example [6], authors use classifier based on Primary Component Analysis (PCA) to optimize resource allocation. They saved about 20% of resources during test with algorithm which use distinguished groups. Both examples has been performed using benchmarks to emulate real world workloads.

[+] Corresponding author. Tel.: +48 58 348 6343; fax: + 48 58 347 10 06.
E-mail address: jerp@task.gda.pl.

There are also complete frameworks which predict resource utilization of workloads. Distributed iBalloon framework [7] could forecast resource utilization only for 2-layers web application. Another framework VCONF [8] can automatic configuration of VM's. This one is able to predict resource usage for e-commerce, online transaction processing and webserver applications. Both frameworks use metrics like CPU and memory utilization, I/O operations, swap usage. So only base metrics for hardware are used.

Another example is analysis of workload characteristic described in [9] based on measurements in Google Cloud where workload classification with resource usage criteria has been proposed. The analysis input metrics are execution time, CPU and memory utilization.

Some examples of functional classification also could be found in literature. Kundu et al. [10] propose grouping of workloads as follows: application server, database, file server, etc. Similar to mentioned researches following metrics have been monitored: usage of CPU and memory, latency of I/O with relation to the storage access. SVM and neural network have been tested and both got good accuracy which has been increased by adding regression model.

Other proposition of functional classification of workloads has been presented in [11]. Authors divide workloads to following classes: idle, OLTP, file server, science, application server. These classification has been used to choose method to migrate virtual machine in efficient way.

Mentioned works shows either automatic and functional classifications of the workloads, but none of them is matching both concepts. That is the direction which could improve the classification results. To extend presented examples we proposed to use another metrics which extend standard set of CPU, memory, I/O and network utilization (for example cache references).

2. Automatic Workload Discovery

In our research we propose categorization of workloads based on monitoring resources utilization and then with unsupervised machine learning techniques we perform their analysis.

First we create common functional categorization which covers most popular workloads used in cloud environment. Based on related works and research on most used applications hosted by cloud providers we propose following functional groups:

1. Science,
2. Big Data,
3. OLTP,
4. Caching,
5. Streaming,
6. Web serving.

Selection of categorization algorithm that can group our workloads by resource utilization has been preceded of research in state of the art. As one of the simplest algorithm with proofed accuracy in this concept [12] is k-means clustering method. For each proposed functional class, a number of representative benchmarks were chosen to emulate real-life workloads.

In the next step we deploy and configure benchmarks in or laboratory that emulates the cloud. The experiments were performed in an HPC cluster environment: Galera Plus supercomputer located at the Academic Computer Center of Gdansk University of Technology in Poland. Infrastructure consists of six compute nodes, each equipped with 2 Intel Xeon L5640 processors (2.27 GHz, 12 MB cache) and 16 GB RAM memory interconnected by Gigabit Ethernet.

Software layer based on KVM hypervisor installed on Linux Ubuntu Server v12.04 as host with v14.04 deployed as guest operating systems. One client virtual machine was executed at the same time on each host node. All resources from host machine were assigned to guest machine where the workloads have been executed. Allocated hardware included 24 virtual CPUs and 12 GB of physical memory. All measurements have been done on the host machine using own tool based on the PAPI library. The monitoring software trace 115 metrics of resource utilization related to different hardware and OS parameters. The full list of the observable metrics can be found in [13]. All of the gathered data has been stored in NoSQL Database (MongoDB).

3. Categorization of the workloads with k-means

Following plan and rules has been applied to perform the experiments. Each measured workload has been deployed on a separate virtual machine. Another machine has been create if it was required by benchmark, e.g. client machine for web serving benchmarks. Another parameters have been defined depending on specific requirements of workload class. Example class parameters are: e.g. number of concurrent users for web serving, types of queries for OLTP, length and bitrate of the streams for streaming. Each test with set of defined parameters is called configuration.

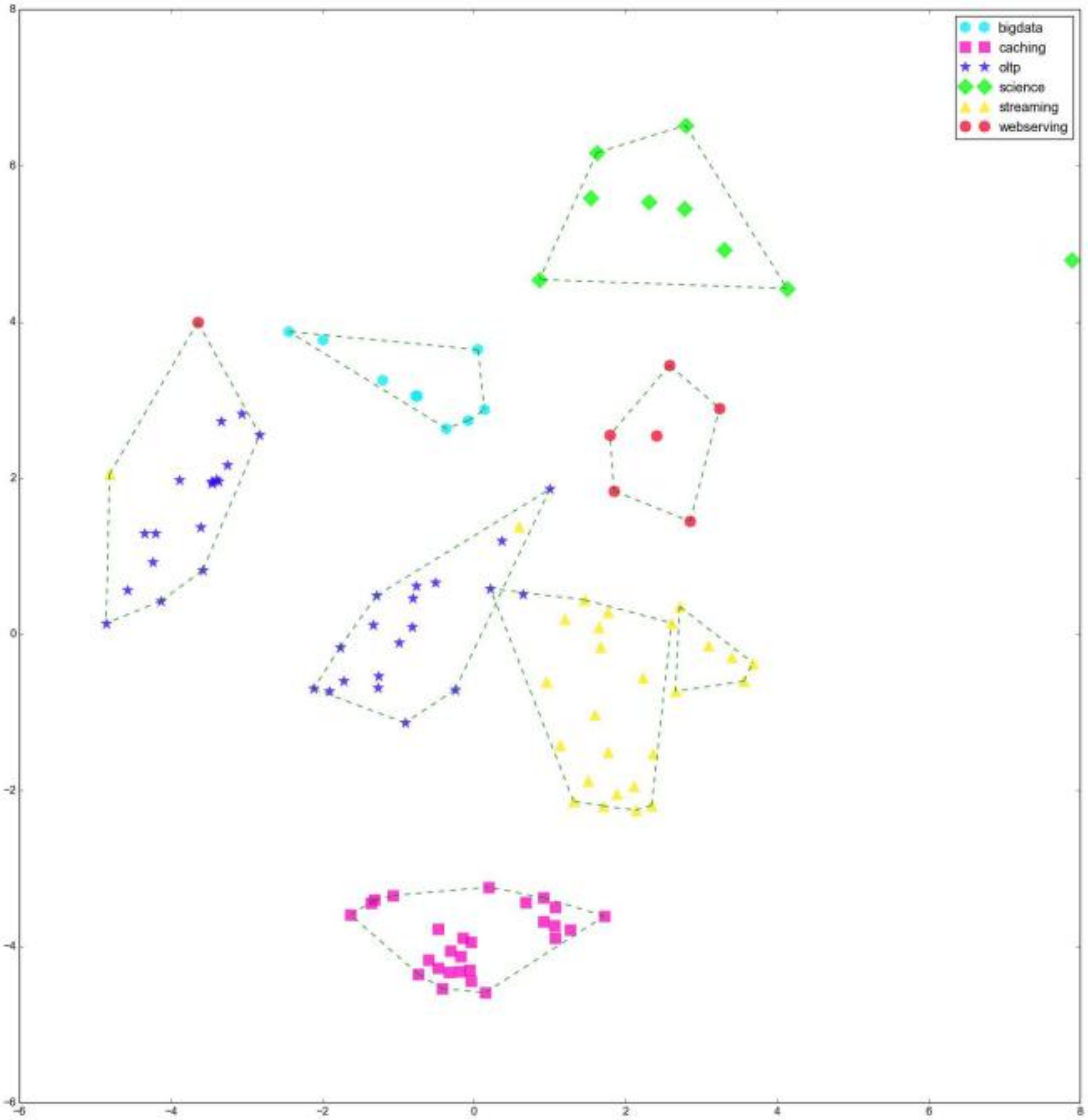

Fig. 1 MDS projection of the discovered groups of the workloads

For every configuration dedicated tests were prepared and performed. Results from all experiments have been collected and further analyzed. Each data sample contains 115 hardware metrics. From all experiments over 136.000 samples were gathered. For each of the tested configurations, statistics were calculated, which included the average, minimum, maximum and standard deviation of every metric. For visualization, we used Multidimensional Scaling (MDS) [14], which allowed us to map the 115-dimensional feature space into a 2D chart, keeping the minimal distortion of the distances between the points. Fig. 1 presents visualization of the data in 2D space, reduced with MDS (colors) and results of automatic workloads categorization with K-means (areas selected with dashed lines).

On the figure it is worth to note that proposed functional classes are concentrated in separate groups. It is caused by that in 115-dimensional space there are also groups which are correlated with proposed functional categorization. K-means algorithms [15] has been applied to validate this thesis. Moving average has been used to automatically determine the number of the groups in the input data set. Found number of clusters was applied as input to K-means. The analysis results (mapped into 2D) are shown as groups surrounded by dashed line in Fig. 1.

Conducted experiments and results of analysis prove there is correlation between assumed functional classed and resource utilization. Over 94% of consistency has been reached during evaluation in 10-fold cross-validation. But there are still workloads which different from others in the same category. Misclassification is shown in Tab. 1. An assumption is that for further research in this area some supervised learning should be prepared. Especially when more workloads will be added.

Table. 1 Detailed information about misclassification between particular workload types.

Categories	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Big Data	22.22%	77.78%	0	0	0	0
Caching	3.57%	0	0	0	0	96.43%
OLTP	45.95%	0	0	0	54.05%	0
Science	0	0	0	100%	0	0
Streaming	7.14%	0	89.29%	0	3.57%	0
Web serving	14.29%	85.71%	0	0	0	0

Results presented above show that proposed functional classification is consistent with classification based on cloud workloads utilization. However we assume that some cases of the workloads could have been omitted and that additional method customization may be necessary. A good example is OLTP category which split to half what can suggests that it should be two separated classes or subclasses for these cases. To improve the results, additional workload-based resource utilization measurements or end user application supervised categorization [16] can be performed.

4. Conclusions and future works

In the paper we propose a method for automatic discovery of workload types based on functional categorization. We construct the parameter space, using 105 resource utilization measurements, in which we performed unsupervised analysis with K-means algorithm. Categorization which has been achieved match with groups, that is a consequence of functional features of used benchmarks.

In the future work we plan to test other, more complex algorithms of clustering and modify monitored metrics set. Another perspective is to extend number of the benchmarks. There is possibility that some or all of these changes can influence on accuracy of classification. Moreover, potentially some of the monitored metrics can be correlated, thus deeper analysis based on feature selection methods [17] has to be performed. In the most of the cases groups of workloads identified by clustering are consistent with functional assignment. It can indicate that the proposed approach can be used for identification of workloads implemented in end-user applications.

5. Acknowledgements

This work was carried out as a part of the research project "Recommendation Component for Intelligent Computing Clouds", co-financed by the European Regional Development Fund, Innovative Economy Operational Programme 2007-2013, Priority Axis 1: "Research and development of state-of-the-art technologies". The experiments were performed using high-performance computing infrastructure provided by the Academic Computer Centre in Gdansk (CI TASK).

6. References

[1] P. Mell, T. Grance. The nist definition of cloud computing. In: NIST Special Publication 800-145. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, 2011.

[2] S. Kundu, R. Rangaswami, A. Gulati, M. Zhao, and K. Dutta, “Modeling virtualized applications using machine learning techniques,” in ACM SIGPLAN Notices, vol. 47, no. 7. ACM, 2012, pp. 3–14.

[3] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, “Performance and energy modeling for live migration of virtual machines,” Cluster computing, vol. 16, no. 2, pp. 249–264, 2013.

[4] J. Rao, X. Bu, C.-Z. Xu, L. Wang, and G. Yin, “Vconf: a reinforcement learning approach to virtual machines auto-configuration,” in Proceedings of the 6th international conference on Autonomic computing. ACM, 2009, pp. 137–146.

[5] X. Zhao, J. Yin, Z. Chen, and S. He, “Workload classification model for specializing virtual machine operating system,” in Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on, June 2013,pp. 343–350.

[6] J. Zhang and R. Figueiredo, “Application classification through monitoring and learning of resource consumption patterns,” in Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, April 2006.

[7] J. Rao, X. Bu, K. Wang, and C.-Z. Xu, “Self-adaptive provisioning of virtualized resources in cloud computing,” in Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, ser. SIGMETRICS ’11. New York, NY, USA: ACM, 2011, pp. 129–130. [Online]. Available: http://doi.acm.org/10.1145/1993744.1993790

[8] J. Rao, X. Bu, C.-Z. Xu, L. Wang, and G. Yin, “Vconf: A reinforcement learning approach to virtual machines autoconfiguration,” in Proceedings of the 6th International Conference on Autonomic Computing, ser. ICAC ’09. New York, NY, USA: ACM, 2009, pp. 137–146. [Online]. Available: http://doi.acm.org/10.1145/1555228.1555263

[9] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das, “Towards characterizing cloud backend workloads: insights from google compute clusters,” SIGMETRICS Perform. Eval. Rev., vol. 37, no. 4, pp. 34–41, Mar. 2010.

[10] S. Kundu, R. Rangaswami, A. Gulati, M. Zhao, and K. Dutta, “Modeling virtualized applications using machine learning techniques,” SIGPLAN Not., vol. 47, no. 7, pp. 3–14, Mar. 2012.

[11] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, “Performance and energy modeling for live migration of virtual machines,” in Proceedings of the 20th International Symposium on High Performance Distributed Computing, ser. HPDC ’11. New York, NY, USA: ACM, 2011, pp. 171–182. [Online]. Available: http://doi.acm.org/10.1145/1996130.1996154

[12] T. Hastie, R. Tibshirani, and J. Friedman, Unsupervised learning. Springer, 2009.

[13] P. Czarnul, “Model of a computational node in a cloud evaluation of hardware metrics,” in TASK internal report. Gdansk University of Technology, 2015, pp. 1–26.

[14] J. B. Kruskal and M. Wish, Multidimensional scaling. Sage, 1978, vol. 11.

[15] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” Applied statistics, pp. 100–108, 1979.

[16] S. B. Kotsiantis, “Supervised machine learning: A review of classification techniques,” in Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies. IOS Press, 2007, pp. 3–24.

[17] J. Rzeniewicz and J. Szymanski, “Selecting features with SVM,” in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 18th Iberoamerican Congress, CIARP 2013, 2013, pp. 319–325.