DOI: 10.5604/20830157.1176574

MASSIVE SIMULATIONS USING MAPREDUCE MODEL

Artur Krupa, Bartosz Sawicki
Politechnika Warszawska, Instytut Elektrotechniki Teoretycznej i Systemów Informacyjno-Pomiarowych

Abstract. *In the last few years cloud computing is growing as a dominant solution for large scale numerical problems. It is based on MapReduce programming model, which provides high scalability and flexibility, but also optimizes costs of computing infrastructure. This paper studies feasibility of MapReduce model for scientific problems consisting of many independent simulations. Experiment based on variability analysis for simple electromagnetic problem with over 10,000 scenarios proves that platform has nearly linear scalability with over 80% of theoretical maximum performance.*

Keywords: mapreduce, cloud computing, platform performance, hadoop

MODEL MAPREDUCE W WIELOKROTNYCH OBLICZENIACH NUMERYCZNYCH

Streszczenie. *W ostatnich latach chmury obliczeniowe stały się dominującym rozwiązaniem używanym do wielkoskalowych obliczeń numerycznych. Najczęściej są one oparte o programistyczny model MapReduce, który zapewnia wysoką skalowalność, elastyczność, oraz optymalizację kosztów infrastruktury. Artykuł w analityczny sposób przedstawia wykorzystanie MapReduce w rozwiązywaniu problemów naukowych złożonych z wielu niezależnych symulacji. Przeprowadzony eksperyment, złożony z ponad 10 000 przypadków, oparty o analizę zmienności pola elektromagnetycznego pokazuje niemal liniową skalowalność platformy i jej ponad 80% wydajności w stosunku do teoretycznego maksimum.*

Słowa kluczowe: mapreduce, chmura obliczeniowa, wydajność platformy, hadoop

Introduction

Distributed processing has a long history in a computer science. Since early beginning it has taken advantage of multiple processors available at the same time. Recently popular term "cloud computing" is not scientifically precise, but its general understanding is based on flexibility of architecture, calculations on-demand, and lower costs related with the model.

Despite objections related with privacy, cloud computing is one of the most growing elements of todays technology [3]. It provides both, flexibility and performance [6]. Top IT enterprises (eg. Amazon, Microsoft, Google) are developing their datacenters with hundreds of thousands of computing nodes to process huge datasets. Academia users also intensively investigate how new architecture could be applied into scientific problems [1].

It can not be denied that cloud computing has commercial roots. But it is worth to mention that similar systems could be created in educational or community based environments. Volunteer computing is based on thousands of private machines working together to solve important, large scale problems [5].

In this paper we will verify usability and overall performance of cloud computing scheme in real engineering problem. Variability analysis for simple electromagnetic model is used as a exemplary problem for MapReduce massive simulations platform. Obtained results will be compared with classical single processing algorithm.

1. MapReduce model

1.1. MapReduce processing

Cloud computing is deeply connected with MapReduce distributed processing model. Idea behind MapReduce is to automatically split processing into containers of datasets, which could be distributed over many independent computing nodes. Each of the nodes is a computational unit, it could be high power rack server, standard computer or even single virtual machine.

During MapReduce operation (as shown on Fig. 1), firstly, master node is splitting InputData into datasets to distribute them among all working nodes. This stage is also about making a copies of all data to provide redundancy.

Second step is to run application transforming splitted InputData parts into intermediate key-value format data for further processing. This process is called Mapping. Mapper task makes another split of data and working on a smaller datasets in own allocated memory. Output results of Mapper stage are sets of non-sorted data. Because Mapping is a processing of all data and each Mapper is working independently, we do not know when and where a specific InputData dataset is processed. Next phase can start, when all Mappers finish processing of assigned part of data. When all data are processed, Mappers are terminated to release memory.

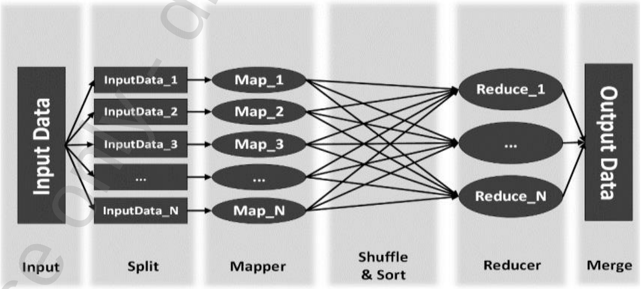

Fig. 1. MapReduce is a primary model for cloud computing

Third step of MapReduce algorithm is called "shuffle and sort". During this phase, Reducers (also launched on working nodes) start to work and get data from partitioned outputs from Mappers. This looks like searching for a values (small results) with indexes and making a list in a numerical order.

When all outputs in datasets from Mappers are indexed and sorted into size-accepted blocks – they are ready to final merge. Last step is also called "reduce". During this stage, Reducers are using a list of order and saving all data in one single file into final output. Such scheme is universal, and allows to efficiently solve wide range of problems.

There are number of numerical algorithms where simulator is run repetitively for many different parameters. Stochastic optimization, evolutionary algorithms, Monte Carlo methods, variability analysis are only the most popular examples [2]. As described above MapReduce model consists of two main stages: mapper and reducer. First of them is directly connected with input data, what makes him suitable for processing of many similar simulations.

Straight forward approach to use MapReduce for variability analysis is to replace mapper by the problem simulator and input data by the problem parameters [4]. Then MapReduce framework will spread calculations over all available computing nodes. Proper use of reducer stage is the key for optimal implementation. Its role is to aggregate mapper results.

Depending of type of the algorithm, reducer could be used to choose optimal solution, find the best individual in population, calculate histogram of solutions, or to other tasks.

1.2. Hadoop basics

MapReduce model has many implementations, such as MongoDB, Riak, DiscoProject. But there is one which is definitely the most popular, Hadoop project [8] developed by Apache foundation in open-source model.

Hadoop Project is an advanced, modular environment based on Java Virtual Machine (JVM) which provides many benefits: multiplatform support, multi-language scripts solutions allowance, code (tasks) protection during execution. Hadoop platform offers a optimization of hardware utilization and redundancy management features which are important, because cloud clusters are usually based on low-costs, faulty machines.

2. Efficiency of the platform

2.1. Numerical experiment

To test platform efficiency we constructed variability analysis of simple electromagnetic problem. As it is shown on Fig. 2, it is based on square shaped conductor in which circular hole is created. Obviously location of the hole will influence resistance of the conductor, but it is non-linear and not trivial relation. We used MapReduce model to conduct analysis of the problem.

Firstly we created a file with 100×100 scenarios to calculate a resistance for each center of the hole (x,y). Points were generated using uniform method which provides an even distribution of parameters inside given range.

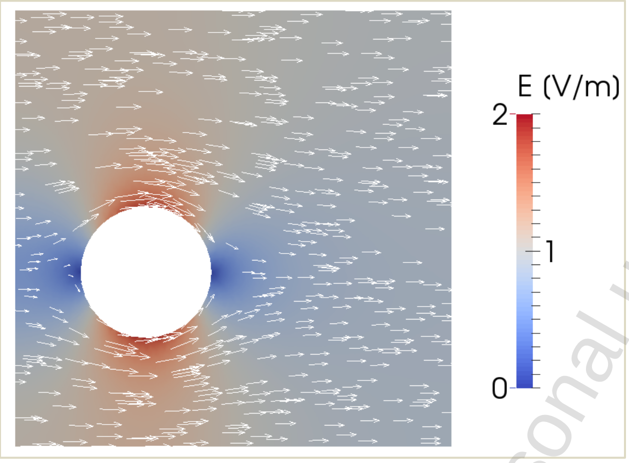

Fig. 2. Current field in experiment solved using FEM

To fit into Hadoop framework mapper and reducer has to be provided. Our mapper code is a Finite Element Method (FEM) simulator based on FEniCS library. Current field distribution is described by the simple Laplace's equation for electric scalar potential (φ):

$$\nabla \cdot \sigma \nabla \varphi = 0$$

with conductivity σ, and voltage U as a boundary condition.

Having solution for scalar potential, resistance R could be found as:

$$\mathrm{R} = \frac{U^2}{\int \sigma(\nabla \cdot \varphi)^2 dv}$$

In case of our test problem, reducer is not intended to provide statistical function, but it is simple input-output rewriter. For efficiency investigations final result is computation time needed to process all scenarios.

Single simulation (solution of FEM problem) in our experiment takes about 4 seconds on generic, modern computer. To complete all of the analysis, total number of 101×101=10201 scenarios has to be solved. What means that simple, serial processing will take 40804 seconds (11 hours 20 minutes and 4 seconds). Reducing this time is the main reason for application of Hadoop platform.

2.2. Hadoop platform configuration

Test platform is a cluster of four physical servers. Each of them has two quad-core Intel Xeon E5620 CPUs running at 1.6 GHz, and 256GB SSD drive. Total memory available for Hadoop was 24GB. Nodes were connected to gigabit Ethernet and running GNU/Linux 64-bit version. The newest release of Hadoop 2.6.0 had been installed.

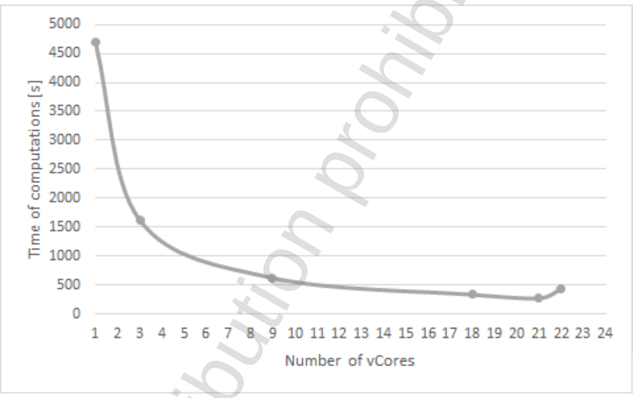

Fig. 3. Optimization of number of vCores based on computation time

One of the servers is devoted for administration of the cluster, so total number of available computational cores is 24 CPU. Hadoop could be configured to use any number of vCores. Results for different settings are presented on Fig. 3. One can see that optimal results are for 22 cores. The general rule is that because of platform stability and ApplicationMaster process, two cores has to be dedicated for internal Hadoop management.

2.3. Hadoop performance

Next performance experiment is designed to determine scalability of the platform. Series of simulations for different numbers of scenarios were run and total time of computations were measured. Results are presented on Fig. 4a) for very small cases (<25) and on Fig. 4b) for larger problems. For comparison reasons results from single computer (PC) were also plotted.

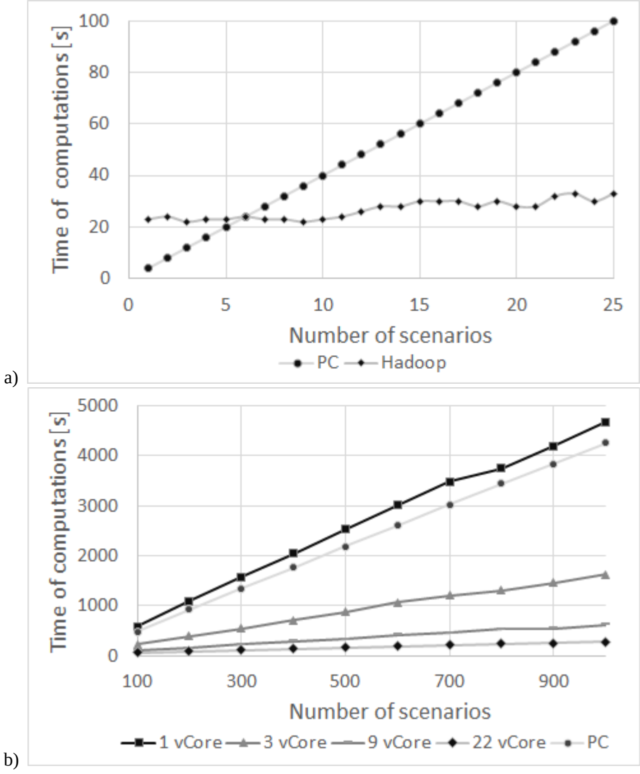

Fig. 4. Computation time of PC and Hadoop for small (a) and large (b) number of scenarios

As seen (Fig. 4), Hadoop startup time is about 20 seconds, so for less than 6 scenarios (4 seconds each) we see that single computer is faster, and there is no need to use cloud-based computing environment. For larger problems cloud platform shows its advantage. Scaling plot is very close to linear function, where steepness of the line is controlled by the number of processing cores attached to Hadoop.

The most spectacular speed-ups are observed for large numbers of scenarios. For instance, for 1066 scenarios single PC simulation time is about 4300 seconds, while using Hadoop with 22 cores configuration only 275 seconds. What gives impressive 15× performance gain. On the other hand, it is still below theoretical maximum speed-up which is 22.

Hadoop is complicated computing engine providing flexibility, fault control and other features for managing very large problems. It is expected that it will introduce some overhead, but one can ask what is the level and scalability of the overhead. To address this issue we run set of larger problems, reaching up to 10,000 of scenarios.

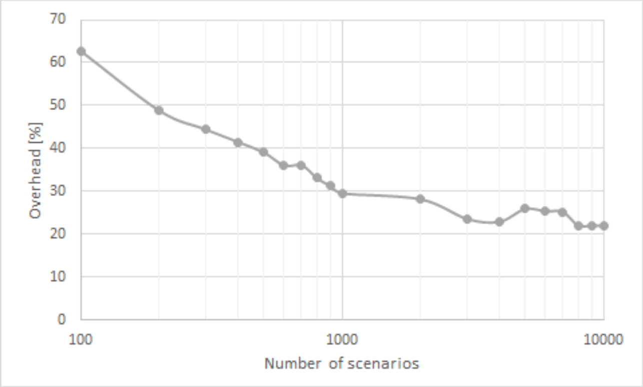

Fig. 5. Hadoop platform overhead as a function of problem size

Fig. 5. shows overhead understood as indirect time costs of using Hadoop. As one can see – as number of scenarios grows overhead is going down. This is very desirable effect, which allows us to use nearly 80% of theoretical platform performance for 10,000 scenarios. At the same time, we can make prediction that even for 1 million of scenarios it will be above 10%. This level of overhead can not be neglected, but it also could be explained as a necessary cost for cloud computing flexibility.

3. Conclusions

Presented experiments have shown that cloud computing is attractive solution for chosen scientific analyses. MapReduce scheme could be easily used to efficiently manage massive numbers of independent simulations.

Apache Hadoop platform connected with electromagnetic solver implemented with FEM presented nearly perfect scalability. On the other hand, we observed overhead at the level of 20%. It can be justified by unique features of Hadoop, but we believe that more efforts at configuration level should reduce this value. Performance and overhead issues are very problem specific. In our variability analysis external application is intensively called what is not typical for MapReduce, so results should not be projected onto different types of problems.

Cloud computing was inspired by economic aspects of calculations, so the financial costs of presented solutions should be also discussed. Cloud services prices are decreasing every day, but even now they are below 0.05 USD per hour of server (see Table 1). What means that total cost of our 10,000 scenarios experiment would be about 5 USD. This amount is practically negligible comparing to a classical clusters investments. So we can conclude that, the migration scientific problems into the cloud technologies, could save time as well as money.

Table 1. Different cloud computing machines (prices in USD/hour) [7, 9, 10]

Cloud Provider	Machine Type	vCPU	RAM	Storage	Price
Microsoft Azure	A1	1 core	1.75 GB	70 GB	$0.06
Microsoft Azure	D1	1 core	3.5 GB	50 GB	$0.09
Amazon EC2	t2.medium	2 cores	4 GB	-	$0.052
Amazon EC2	m3.large	2 cores	7.5 GB	32 GB	$0.133
Google Cloud	n1-standard-1	1 core	3.75 GB	-	$0.05
Google Cloud	n1-highcpu-4	4 cores	3.60 GB	-	$0.152

References

[1] Barker A., Varghese B., Ward J. S., Sommerville I.: Academic Cloud Computing Research: Five Pitfalls and Five Opportunities, in 6th USENIX Workshop on Hot Topics in Cloud Computing, 2014.
[2] Cunha A. Jr., Nasser R., Sampaio R., Lopes H., Breitman K., Uncertainty quantification through the Monte Carlo method in a cloud computing setting, vol. 185, 2014, 1355–1363.
[3] D'Angelo G., Marzolla M.: New trends in parallel and distributed simulation: From many-cores to Cloud Computing, Simul. Model. Pract. Theory, 2014, 126.
[4] Kim B. S., Lee S. J., Kim T. G.: MapReduce Based Experimental Frame for Parallel and Distributed Simulation Using Hadoop Platform, in Proceedings 28th European Conference on Modelling and Simulation, 2012.
[5] Kondo D., Javadi B., Malecot P., Cappello F., Anderson D. P., Berkeley U. C., Cost-Benefit Analysis of Cloud Computing versus Desktop Grids, 2009.
[6] Sakellari G., Loukas G., A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing, Simul. Model. Pract. Theory, vol. 39, 2013, pp. 92103.
[7] Amazon EC2 Pricing, http://aws.amazon.com/ec2/pricing/, access date: [13.08.2015]
[8] Apache Hadoop project, http://hadoop.apache.org, access date: [21.01.2015]
[9] Google Cloud Platform Pricing, https://cloud.google.com/compute/pricing/, access date: [13.08.2015]
[10] Microsoft Azure Pricing, https://azure.microsoft.com/en-gb/pricing/calculator/, access date: [13.08.2015]

M.Sc. Artur Krupa
e-mail: artur.krupa@ee.pw.edu.pl

Artur Krupa received the M.Sc. degree in Electrical Engineering from the Warsaw University of Technology, Poland, in 2014. In the same year started doctoral studies at Department of Theory of Electrical Engineering and Computer Engineering in Computer Science specialization.
His scientific interests are information and communication technologies, security, virtualization and cloud computing.

Ph.D. Bartosz Sawicki
e-mail: bartosz.sawicki@ee.pw.edu.pl.

Bartosz Sawicki received the Ph.D. degree in electrical engineering from the Warsaw University of Technology, Warsaw, Poland, in 2003.
From 2009 to 2010, he was a Postdoctoral Fellow at University of Calgary, Calgary, AB, Canada. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Warsaw University of Technology.
His scientific interests include open-source software applied for simulations of bioelectromagnetic problems using distributed platforms.

otrzymano/received: 27.07.2015 *przyjęto do druku/accepted: 27.08.2015*