STUDIA INFORMATICA 2012
Volume 33 Number 4 (109)

Łukasz J. JANIK
Silesian University of Technology, Institute of Informatics
Stuart E. BARNES
Cranfiled University, School of Engineering

COMPARISION BETWEEN COMPUTER CLOUDS AND LOCAL CLUSTERS IN CFD SOFTWARE APPLICATION

Summary. Work is a comparison of HPC clusters and computer cloud considering problem size and communication pattern. The paper addresses aspects like performance, costs, scalability, elasticity, reliability, and resource utilization. Comparison has been made indicating application of CFD (Computational Fluid Dynamics) simulations.

Keywords: computing, computer cloud, performance evaluation, HPC, cluster

PORÓWNANIE CHMUR KOMPUTEROWYCH I KLASTRÓW LOKALNYCH W ZASTOSOWANIACH CFD

Streszczenie. Praca jest porównaniem wydajności klastrów wysokowydajnościowych i chmur komputerowych, biorąc pod uwagę rozmiar problemu oraz sposób komunikacji. Publikacja porusza aspekty, takie jak wydajność, koszty, skalowalność, elastyczność, niezawodność i efektywność wykorzystania zasobów. Porównanie zostało wykonane pod kątem zapotrzebowania na symulacje CFD (Computational Fluid Dynamics).

Słowa kluczowe: obliczenia, chmury komputerowe, ocena wydajności, obliczenia wysokiej wydajności, klastry komputerowe

1. Introduction

1.1. Background

Computer power and the availability of computing resource availability increases every month enabling the application of computers to new areas, such as the rapidly growing area of bioinformatics, weather forecasting, CFD and the movie industry. Demand for accessing high performance computing power arises in research, commercial and private use. Nowadays, improvements in the rate of processors clock speed has decreased [1], thus single processor machines do not provide sufficient performance for many computational problems. As a result, commercial companies and research facilities build up data centres where large collections of computers work together as a team to solve a single problem. There is a need for cheap and efficient systems. In the past, only super computers and local clusters were able to deliver sufficient performance to solve many of the computational problems. Now there is a completely new solution - computer clouds, which have revolutionized high performance computing since 2006 when the first commercial cloud was launched [2]. Cloud computing was initially designed for web-service based applications, with a relatively low requirement in respect of communication speed between compute nodes. As a result, the question of using an application computing cloud to solving numerical problems is now an object of many discussions in science and engineering. Computer clouds and local clusters use different architecture. More over available cloud usually offer hardware configuration for general computing, while cluster is customized for specific needs. Therefore in this paper we will try to identify the strengths and weaknesses of both systems and identify the specific areas of computing where one system is better than other.

1.2. Contribution of the present paper

In this work we focus mainly on IaaS (Infrastructure as a Service) cloud. We build up a cloud based on the IaaS concept, commonly used for HPC applications, where the developer is presented with virtualized computing resources and has full privileges to install and configure a software stack in a virtual environment.

We performed a comprehensive comparison in terms of costs, performance, scalability, resource utilization, adaptability and reliability between HPC cluster (High Performance Cluster) and computer clouds. In order to evaluate performance we built a local private cloud based on the open source cloud environment Eucalyptus and ran benchmarks on various configurations with intra- and inter- node communication. We also considered other private IaaS (Infrastructure as a Service) cloud environments, such as Nimbus and OpenNebula. The latter goes further than other cloud packages; it provides a cloud service similar to other cloud packages, but also an abstraction layer over the various cloud environments. A coherent interface allows the deployment of virtual machines (VM) over multiple cloud services compatible with EC2 CLI. This approach represents a plausible future model of cloud computing, where the user can deploy VM onto various cloud environments using a single client. Unfortunately, at the time of this study OpenNebula was not yet a mature server, featuring plenty of defects in operation and therefore was not considered ready for commercial or academic use.

There are plenty of definitions of computer clouds. For the purpose of this paper we use definition of cloud proposed in "Cloud Computing and Grid Computing 360-Degree Compared" where authors introduced following definition: *A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet* [1]. Our cloud environment

We understand IaaS cloud as a private or public computer cloud with following features:

- Abstract infrastructure - due to abstraction layer of VM and cloud, software environment and physical location of cloud nodes are not relevant for user (Fig. 1.)
- Pay as You Go scheme – no initial costs, paying only for used resources [2] (refers only to public clouds)
- Theoretically unlimited resources [3] – amount of assigned resources as demanded, continuous ability to satisfy new demands [2]
- Scalability – amount of allocated resources can be increased instantly at any time
- Shared infrastructure – infrastructure is shared across multiple users. It enhances load balancing; however without good isolation it could be a security threat
- Portability – VMs could be stopped at any time and moved to another cloud; this ability guarantees constant access to computing power even in case of a breakdown of one vendor's infrastructure
- Undetermined interconnections – most vendors do not guarantee types of interconnections between nodes, usually nodes within a cloud are connected by Gigabit Ethernet or equivalent [4]. The situation with interconnections has changed recently with the introduction by Amazon of Quadruple Extra Large Instance virtual machine types, which guarantee 10 Gigabit Ethernet interconnection.

Our work focuses on particular characteristics of HPC (High Performance Computing) applications, and their execution on cloud and local cluster computing resources. Our performance evaluation focuses mainly on problem with tightly-coupled communication pattern, where density of communication is high in compare to density of calculations. I.e. communication is frequent, and calculations are stopped until updated data from another node is received. It is opposite to loosely-coupled problems, where communication frequency is much lower, thus bigger parts of computations can be performed without updated data from another node. Performance in resolving such problems is much less vulnerable on communication latencies, than for tightly-coupled problems. We used knowledge of computing and communication patterns required by the target application to prepare suitable test environments to extract maximum performance from particular system architecture. In addition, we performed a cost analysis relying on prices of the largest public cloud provider, Amazon AWS.

In our initial analysis we enumerated the following advantages and disadvantages of architectures for comparison:

Table 1

Advantages and disadvantages of Computer Cloud and HPC cluster

Commercial Cloud (IaaS)	HPC cluster
+ no start-up costs	– high start-up costs
+ pay-per-use	– constant high maintains costs
+ flexible application stack	– limited application stack
+ on-demand access	– jobs have to wait in queues
+ portability	– no portability
+ quite resistant on hardware failure	– variable on hardware failure
– resource shared among users from around a world (potential security threat)	+ only authorized users have access
+ load balancing is no user concern	– difficult to keep hardware working all the time
+ life cycle is not user concern	–life cycle limited (after release of next generation of architecture Flop per Watt from cluster is more expensive than from new architecture)
–– undetermined interconnections	+++ interconnections can be adjusted and customized for user needs
*unlimited number of nodes to extend with uncertain scalability radio	*limited number of nodes to extent with high scalability ratio

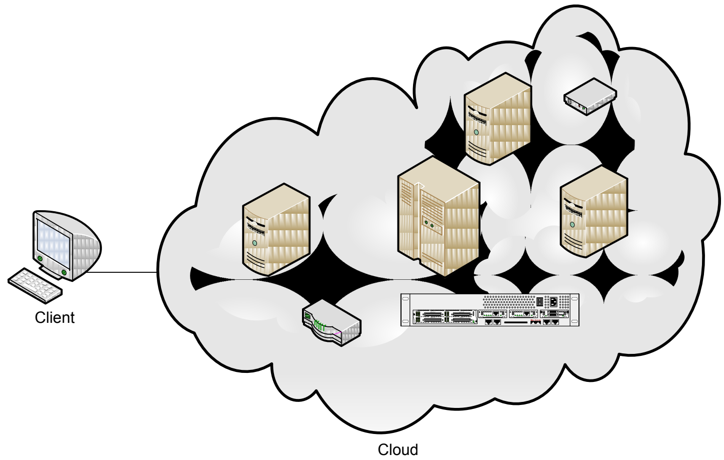

Fig. 1. Complexity of cloud internal architecture is encapsulated for user
Rys. 1. Skomplikowana architektura wewnętrzna chmury jest niewidoczna dla użytkownika

2. Related works

There are several related works concerning the comparison of local and cloud infrastructures for HPC applications. In [5] authors examined performance of Amazon EC2 services with LINPACK for various cluster sizes. They proposed GFLOP/$ ratio and showed how it changes depending to the cluster size. In [4] performance comparisons were made of various EC2 instances, local cluster (2 nodes) and single local machine in MPI (Message Passing Interface) environment. Performance has been evaluated with IMB benchmark. Authors concluded good suitability of cloud environments for loosely-coupled problems, where data is exchanged infrequently in large blocks and low performance of cloud environments for applications with frequent data exchange pattern. An analysis of public cloud bottlenecks, costs, advantages and disadvantages is also included in [2]. Performance study of applications with various data dependency profiles in cloud environment is described in [6, 7, 8].

Authors of [9] gave a comprehensive study of advantages and disadvantages of virtualization. They built a virtualised cluster with over 100 VMs on three sets of physical resources. The cluster was used to examine the performance of two applications, one with a tightly-coupled communication pattern and another with a loosely-coupled communication pattern. They deduced that in a VM environment for applications with tightly-coupled communication pattern, cost of communication is critical for large clusters. However, applications with loosely-coupled communications enjoy considerable speedups.

Comparison between Amazon EC2 and UEC were performed by Jonathan S Ward, and results are presented in [10]. UEC had better memory and cache bandwidth, but worse performance of storage. Author recommended UEC as computational environment for problems that do not require fast parallel storage, while EC2 was recommended for problems with extensive usage of parallel storage.

Power consumption of clusters is analysed and described in [11, 12]. The power consumption of a single node (dual Northwood Xeon, 1GB of RAM, single HDD and GbE card) is analysed in [11]. The authors compared the total power consumption and consumption of individual components in two states: load, and idle with power saving enabled. They deduced that switching hardware to power saving when idle was an effective method to decrease power consumption with low impact on performance. The rising problem of cooling server rooms in modern data centres, caused by impact of Moore's Law, was described in [12]. The authors recommend considering operational cost as part of total cost of computing and showed ways of enhancing cooling efficiency.

3. Methodology

3.1. Environment

A local HPC cluster called Astral represents the test cluster used for the baseline comparisons. Astral is composed of 214 nodes and an HP High Scalable Share File Storage System (Lustre) connected with fast Infiniband network. Each node contains two dual core Intel Xeon 51xx "Woodcrest" or two dual core AMD Opteron "Santa Rosa" processors and 8GB of memory, with a memory/core ratio of 2GB. The entire system possesses 856 cores in total. Individual nodes in the cluster use 64bit Linux with the 2.6.x kernel. LSF-HPC 6.2 SLURM is responsible for resource allocation and load balancing.

The test Cloud environment contains two nodes (Table 2: Node1, Node2), Node3 (used as NFS server), and Node4 (used as Cloud Controller). All hardware elements of the cloud are connected via Gigabit Ethernet. VMs are run on cloud nodes, which CPUs based on Nehalem architecture. The Node4 has a much slower CPU and its NIC supports only fast Ethernet, but it did not have a significant impact on performance of VM instances, which resided on the nodes with Nehalem architecture. Both the physical cloud nodes and the Astral compute nodes represent similar absolute performance in single processes HPL (High Performance Linpack) test (Table 2). The amount of physical RAM memory varied between the different nodes, therefore the performance tests were selected to run on a basic node with 4GB of RAM to eliminate any distortion caused by swapping to page files from influencing the results. Swap file statistics were gathered and checked after each test, to ensure that swapping had not occurred. In order to further secure results from bias, the local network environment was separated from external network by router to ensure that only traffic associated with the tests was present. Although the compute nodes and the Node4 belonged to the same logical network, VM instances created a virtual network which was logically isolated from both – computational nodes and Node4. A Node3 was configured as part of the same subnet as the VM instances and was therefore directly accessible from the VM instances. In the barenode tests, the Node3 was configured for the same subnet as the compute nodes. The Node3 provided a shared file system used for storage of configuration files. It also provided shared directories required to achieve good performance in the STAR-CD and Fluent test applications.

Nodes were running Linux. The specific distributions chosen to run on the computational nodes varied according to the test being executed, however both the Node4 and the computational nodes always ran Eucalyptus 1.6.2. The Node4 worked always on 32-bit Ubuntu Enterprise Cloud (UEC) 10.04. The Node4 (used as Cloud Controller), as name indicates, is responsible for management of cloud and virtual networks, image deployment and storage of VM images. It also provides a web service with a CLI (Command Line Interface), which serves as interface between user and cloud. The computational nodes work on OpenSUSE 11.1 with Xen 3.3 for test with Xen, on UEC 10.04 for KVM (Kernel-based Virtual Machine) 0.12.3 for test with KVM or CentOS 5.4 for barenode testing. The VM instances ran on CentOS 5.3. Nodes used in experiments were components of cloud with architecture featured on Fig. 2.

Table 2

Hardware environment used for performance evaluation

Feature	CPU	CPUs	Cores	Cache (Mb)	Freq (GHz)	Interc onnect	Mem (GB)	Mem /core	Abs pref
Astral	Xeon 51xx	2	4	4	3	Infini band	8	2	1,01
Node1	i7 Core 920	1	4	8	2,67	GbE	4	1	1,04
Node2	Xeon E5462	2	8	12	2,8	GbE	16	2	1,04
Node3	Core Quad	1	4	n/b	2,66	GbE	1	n/b	n/b
Node4	Pentium IV	1	4	n/b	2,8	100BA SE-T	0.5	n/b	n/b

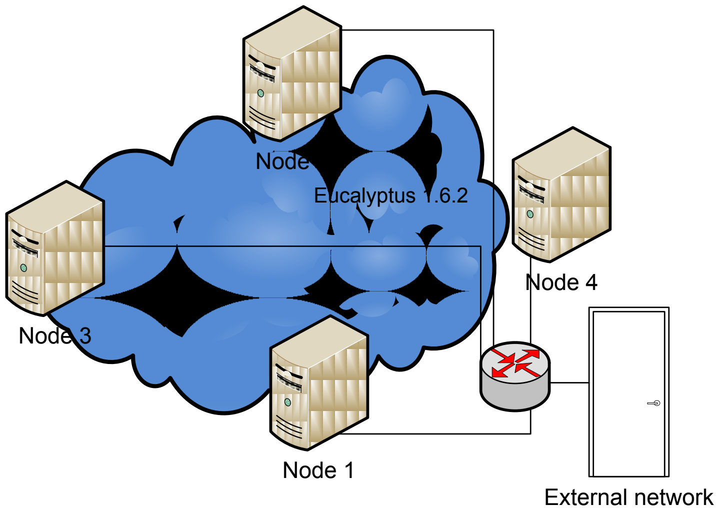

Fig. 2. Architecture of cloud used for tests
Rys. 2. Architektura chmury użytej do testów

CPUs of computational nodes support Hardware Assisted Virtualization (HAV). HAV is widely advertised as technology which can significantly increase performance of virtualized resource. In order to verified above statement, we run all test using hardware assisted virtualization (KVM hypervisor) and for comparison without HAV, but using paravirtualization (Xen hypervisor). Unfortunately, our hardware does not support I/O virtualization which probably has much higher impact on performance than CPU VT in application with tightly-coupled communication pattern.

3.2. Test configurations

Most of tests were run to examine performance and scalability for different sized VMs' instances according to Table 3. To achieve this aim we gradually increased the available resources (number of CPU cores and size of memory). The majority of tests were performed on following resources size: 1, 4, 8, 12, 8inter. Tests 1, 2, and 4 were run on Node2, while 8inter and 12 on Node2 and Node1. Due to the fact that physical resources are usually homogenous and in our lab are not, we added the "8inter" test, which refers to a more realistic scenario in which nodes possess equal number of cores. This differs from the 12 core test, which use 8 cores of Node2 and 4 cores of Node1. In the "8inter" test, 8 cores were used: 4 on Node2 and 4 on Node1. Comparison between 8inter and 8x test will allow us to examine the influence on performance of two possible bottlenecks – memory and network. We also performed barenode tests. Those tests have been performed on Node1, Node2 or both, but without virtualization. Such tests allow estimating impact of virtualization on performance.

Each test was repeated three times, average of three trials was taken as final result.

Table 3

Instances types used in tests

Instance type	Core	Mem(MB)	Mem/core
c1.small	1	850	850
c1.medium	2	1700	850
c1.large	4	3400	850
c1.xlarge	8	6800	850
barenode*	8	16000	2000
*2nd barenode comprise of 4 cores and 4GB memory			

$$overhaeds\_x = \frac{x - base}{base} * 100\% \tag{1}$$

Formula (1) was used to calculate overheads. x is obtained performance, base is referential performance, while overhead_x is percentage gain or loss of performance of x VMs' instance in compare to referential instance. Overheads can be a positive or negative percentage number, which express percentage difference in calculation time of referred system to calculation time of base system. Using this formula negative number corresponds to scenario when referred system has better performance than base, while when number is positive referred system has worst performance than base.

$$\eta_n = \frac{t_1}{t_n * n} * 100\% \tag{2}$$

Formula (2) is used to calculate efficiency. η_n is efficiency of VM' instance using n concurrent threads; t_n is total computing time in benchmark using single thread; t_n is total computing time in benchmark using n threads; n is a number of threads

3.3. Test cases and scenarios

CFD software is very sensitive to network latencies and takes advantage of high bandwidth, low latency interconnects, such as Infiniband present in the HPC test cluster. Test cases were prepared to examine speedup; overheads of virtualization, in particular the overheads of the virtual network; and, performance of different instance types. Tests were also performed on HPC cluster to compare performances between cluster and cloud, and to examine if cloud offers sufficient performance advantages for CFD simulations. Most tests were run from a shared directory, which contained the CFD configuration files. This allowed the environment to be kept constant when more nodes were involved in the tests.

Some tests were performed in local directory of the VM instance virtual hard disk. This approach allows the avoidance of overheads caused by data transfer to and from Node 3, thus the performance should have been improved. The purpose of this test was to examine the usability of cloud environments for small and medium cases, which require no more cores than is available in single node (12 for modern machines). The advantage of the cloud environment in these cases is the lack of time wasted by jobs waiting in queues.

Tests were performed for two industrially important cases. The first was a benchmark problem for multiple turbulence flow models [13]. The case (called "backward step") simulates turbulent flow through a three dimensional pipe with a downcast (Fig. 3), behind which the flow becomes turbulent. It is simulation of a steady system and was performed for 0.5M, 1M and 2M grid cells. The second test (called "fluid removal") is pipe shutdown in a two dimensional scenario, where a less viscous fluid is used to remove residual fluid of a higher viscosity, which had been transported before (Fig. 4). The high viscosity fluid flow is laminar, while the low viscosity fluid flow is turbulent. The key difficulty in the calculation occurs because every cell in the domain could experience laminar and turbulent flow. Simulations of this unsteady system were performed for 280k grid cells. Both cases were simulated using the RANS model.

In order to get results that were independent from the application, and thus more reliable, we used two commercial CFD software applications: STAR-CD 4.03 and Fluent 12.1SP1. "Fluid removal" was simulated with both applications, while "Backward step" on STAR-CD only. For tests on cloud we used the version of MPI provided with the application: LAM/MPI 7.0 for STAR-CD and Intel MPI 3.2 for Fluent. On Astral MPI was supported by HP-MPI for both Fluent and STAR-CD.

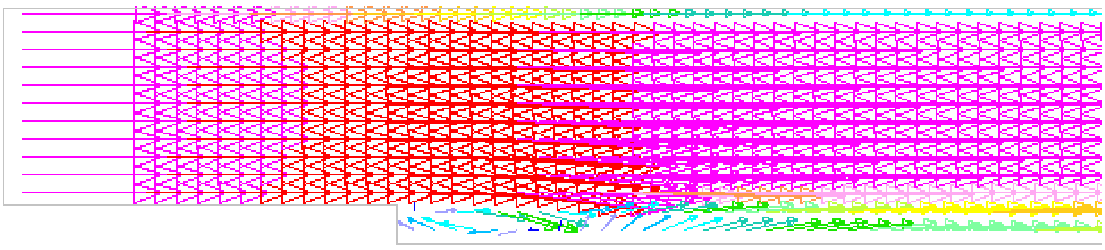

Fig. 3. Backward step case, velocity vector coloured by magnitude
Rys. 3. Przypadek krok w tył, wektory prędkości, dla których kolor odpowiada modułowi

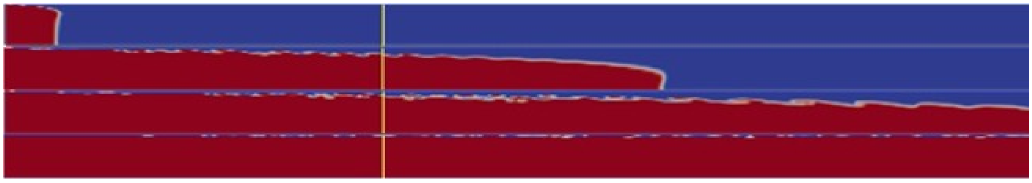

Fig. 4. Fluid removal case, phase of fluid removal
Rys.4. Przypadek usuwanie płynu, fazy usuwania

Table 4

Experiment scenarios

Scenario	Number of threads on Node1/Node 2	Number of threads on Astral Nodes (3 nodes involved)
1x	1/0	1/0/0
2x	2/0	2/0/0
4x	4/0	4/0/0
8x	8/0	4/4/0
12x	8/4	4/4/4
8x inter	4/4	-----

Experiments were performed for six scenarios (*1x, 2x, 4x, 8x, 12x, 8x inter*; Table 4) of five cases (StarCD grid 0.5M, StarCD grid 1M, StarCD grid 2M, StarCD 250K unsteady, Fluent 250k unsteady). *8x inter* refers only to cloud, however on cluster *8x* is its equivalent.

4. Results and discussion

4.1. Performance comparison

We show the results from the various experiments in this section. The number of concurrent processing threads is marked on the charts with a number, however in order to increase readability in description we use add letter “x”. E.g. case with eight concurrent threads in chart is marker with “8” while in description with “*8x*”. The number of concurrent threads is always equal to number of CPUs cores involved. We also tried to add more concurrent threads than cores involved, in result computing time usually was longer than in case when number of concurrent threads was equal to number of cores available.

Negative overheads means that the compared system has better performance than base one. On charts below particular series correspond to VM instance type. Series without postfix “_k” indicate Xen instances, while with “_k” postfix KVM instances. In presented charts in order to enhance readability we left only best KVM results in order to prevent axis from scaling too much. To save space we also remove from charts *1x* and *2x* scenarios, when they were not informative.

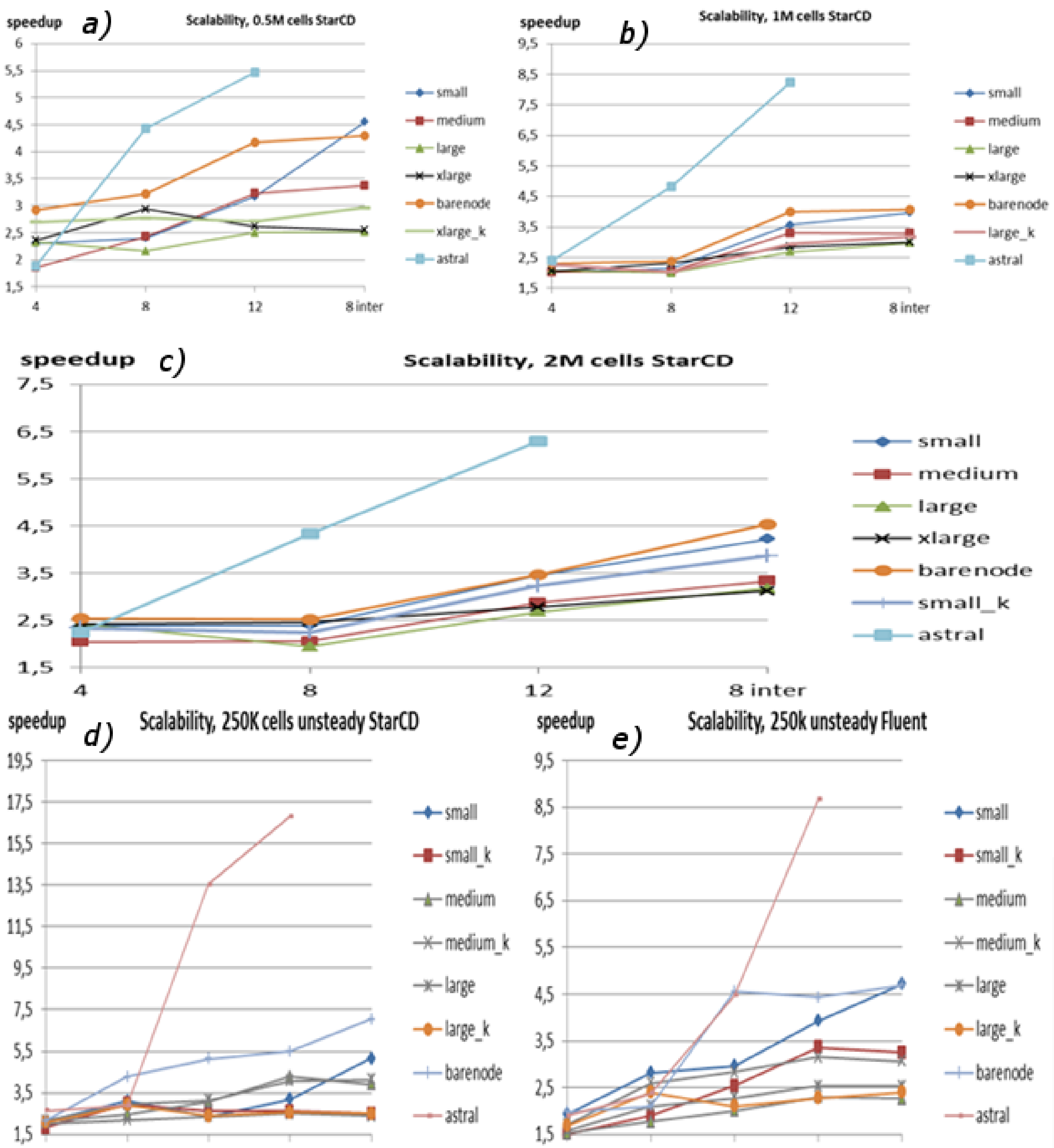

Fig. 5. Charts feature scalability depending on number of concurrent processing threads. a), b), c) refers to "Backward step" case. d), e) refers to "Remove fluid" case

Rys. 5. Wykresy pokazują skalowalność w zależności od ilości równoległych wątków przetwarzających. a), b), c) odnosi się do przypadku „Krok w tył". d), e) natomiast do przypadku „Usuwanie płynu"

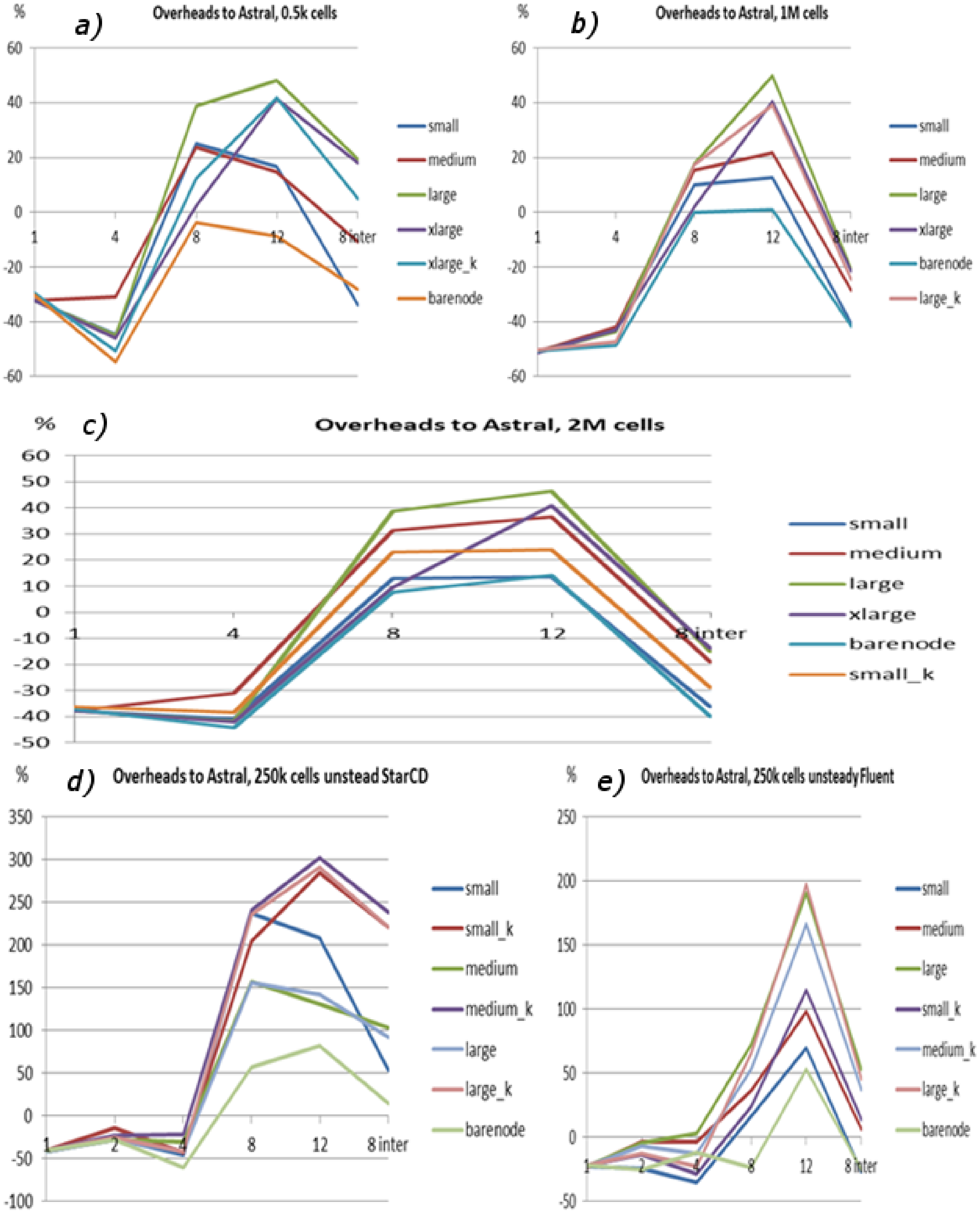

Fig. 6. Charts feature overheads to Astral depending on number of concurrent processing threads. a), b), c) refers to "Backward step". d), e) refers to "Remove fluid" case

Rys. 6. Wykresy pokazują różnice w wydajności między Astralem a poszczególnymi VM'ami w zależności od ilości równoległych wątków przetwarzających. a), b), c) odnosi się do przypadku „Krok w tył". d), e) natomiast do przypadku „Usuwanie płynu"

Fig. 5. and Fig. 6. feature results obtained during experiments. HPC cluster (Fig. 5.) cluster speedup increases almost linearly. For barenode and VMs' speedup increase clearly

when additional physical node is added (the *12x* and *8x* scenario), whereas just a bit when threads are added to the same physical node (*4x* and *8x* resides within same physical node). For large models (please see Fig. 5. "*c)"* with 2M cells and "*a)"* with 0.5 M cells) with more cells it is easier to distribute computing over multiple nodes (speedup presented on Fig. 5. on part "*c)*" is higher than on part "*a)"*), than for small one.

Due to a more efficient architecture of the underlying hardware in the cloud nodes, small VMs' instances and barenode have better performance than the cluster in all scenarios except the *8x* and *12x* scenario with 2M grid size (Fig. 6.). In these scenario, the performance of the cloud environment drops because of the high network card (NIC) load, which has to handle transfer of 8 processes on the cloud, in comparison to the network card load in the cluster (where resources are homogenous), which has to handle only 4 processes simultaneously.

All VM instances and barenode configurations have clearly better performance for 1 and 4 processes. Advantage drops in scenarios with inter node communication (*12x* and *8x* inter), and then only the small Xen instance and the barenode configuration are able to maintain better performance than the cluster. All KVM instances and rest of Xen instances have the worst performance in scenario with inter node communication (Fig. 6.).

Small VMs' instances have usually better performance than large, and this difference increases when inter communication is involved. Without inter communication, the advantage is barely visible (Fig. 6.).

Tests in *8x*, 1*2x* and *8x intra* scenarios show that CFD software is sensitive to the number of processes sharing one NIC. All instances get better results in *8x intra* scenario, than in *8x* and *12x*. In *12x* scenario four additional cores improve performance (in comparison to *8x*), however the advantage of four extra cores does not balance losses caused by overheads induced by sharing NIC by 8 processes, thus performance in *8x* intra scenario is better (Fig. 6.).

The Cluster has high scalability compared with the cloud environment, and maintains efficiency between 50 and 70%. Cloud does not show any increase in performance when number of cores increases from 4 to 8, thus efficiency and scalability for *8x* and *12x* is low. Xen small instance and the native system show good efficiencies when intercommunication is involved, which is a good result when the much slower interconnections are factored in. It proves that virtual cluster connected by Gigabit Ethernet (GbE) is as suitable as previous-generation HPC cluster for small and medium CFD problems, when number of involved nodes is low or medium (as showed in [9] critical cluster size of tightly-coupled problems is around 32, for larger cluster speedup growth is much slower).

The performance of the Cloud for homogenous scenarios was usually higher than performance of previous-generation HPC cluster. The Cloud obtained higher performance (homogenous resources scenario, steady case) at average of 33% for 0.5M grid, 42% for 1M and 36% for 2M (Fig. 6.). Fluctuations (between various VMs' instances excluding native OS) of median performance advantage (median function separate for each instance on all three results of homogenous scenarios) over cluster were 9% for 1M and 2M grid and 16% for 0.5M grid. In unsteady simulations (using homogenous resources), the Xen small instances were faster than the cluster at an average of 30% in Fluent. In the StarCD test they were 33% faster in the scenario without intercommunication and 14% worse than the cluster in the scenario with intercommunication. The large instance and KVM instances had similar performance advantages without intercommunication (in both StarCD and Fluent) and over 40% overheads (small KVM only 14%) with intercommunication in Fluent, over 100% in StarCD. In the scenario with heterogeneous resources with one physical node sharing a single NIC between 8 processes, Cloud performance was much worse. In steady simulation, Xen small instance had average overheads of 10% depending on grid size; for the remaining instances overheads were in the range from 20% to 48%. In the unsteady simulation, overheads were over 130% (for most of instances over 200%) in StarCD. In Fluent around 20% for small instance (Xen and KVM) in 8x scenario (80% in 12x scenario); for remaining instances around 50% in 8x scenario and over 160% in 12x scenario.

Summarising, various test scenarios and various CFD applications gave different performance relationships between cloud and cluster. The fastest VM's instance, which is the Xen small proved at average 40% higher performance than cluster (except unsteady analysis with StarCD), when total number of processes per physical node were no higher than 4. For more than 4 processes per NIC, performance drops significantly. It is difficult to estimate cluster performance for more than 4 processes sharing a single NIC (the cluster available for tests has only 4 cores per node), however a comparison of performance between best results of virtualized OS and non-virtualized OS on the same physical hardware indicates that virtualization is not a reason of low performance when more than 4 processes share single NIC. On the other hand, the xlarge VM instance which is able to fully fill the physical hardware (number of assigned cores equals total number of available cores) showed lower performance (than small) for scenario with 4 processes per NIC and intercommunication, but still surpassed previous-generation cluster in steady simulation (Fig. 6.).

4.2. Cost comparison

We compared the computing costs to Amazon EC2 instances (AWS EU). Middle size cluster like Astral consumes 80kWh and another 45kWh is consumed by its cooling system what gives 125 kWh at total. This energy is consumed every day, every hour. Market average energy price for 1 kWh is £0.13. This gives a yearly cost of £142k just for energy bills. When one adds staff salary and technical support, it is not unreasonable to raise this figure to approximately £400k. The cost of a single twin-node, which consists of two servers, each with dual Xeon quad, 12GB of memory and SDR Infiniband card is £5500k. Above was calculated basing on prices from [14]. Simple calculation for 840 servers (420 Twin nodes) gives cost of £2.31M. It does not include cost of network switches, Keyboard Mouse Devices, parallel storage, cooling and power system, and installation. Considering discount for buying so many nodes and reaming costs, £3M is good approximation. As mentioned earlier, the life cycle is around 5 years, after this period similar amount of money needs to be invested for new cluster. The total cost for 5 years is £5M.

Table 5 features the number of equivalent EC2 hours per day (and per week respectively) when 600 instances of the specified VM instance type can be run over a 5 year period for the same amount of money required to operate the cluster proposed above in the same 5 year period. The 600 instance/day column relates to the scenario in which instances are used only when needed (so better elasticity), while 300+300 instance/day is an scenario when 300 instances are rented constantly over the 5 years period, and because of discount are cheaper, while the remaining 300 are rented only when needed. Cost of transferring 80TB monthly (2,6TB daily) to EC2 is included.

Table 5

Number of EC2 hours which could be rented for amount equal to total cost of middle size HPC cluster

Type(AWS EU)	600 instance/day	600 instance/week	300+300 instance/day	300+300 instance/week
m1.small	45	315	84	587
c1.medium	22	158	40	283
m1.large	11	79	18	125
m2.xlarge	7	53	10	74
m1.xlarge	6	39	7	46
c1.xlarge	6	39	7	46
m2.2xlarge	3	22	2	12
cc1.4xlarge(US only)	3	19	0	0
m2.4xlarge	2	11	-1	-10

Combination of Table 5 and EC2 prices available on [15] show that for HPC purposes c1.xlarge is instance which offers best performance/cost ratio. Presented data shows that use of cloud for computing is far more expensive than use HPC cluster. Even in very optimistic scenario where the prices of instances will be decreasing when next-generation of CPU is released and fluctuations of demands on HPC will be high (unfavourable for constant resources) computing in cloud is more expensive.

5. Conclusion

The results from these experiments exhibited better performance in real-world problems of small instances over large, however performance drops significantly when physical node is shared with other VM instances running applications with similar communication patterns. Due to this, small instances should be chosen only when it is confident that they will be deployed exclusively on computational node, in order to secure good performance.

Cloud features like elasticity, abstraction layer over hardware and portability guarantee significant advantages over traditional clusters. However, performance for large scale tightly-coupled problems typical in CFD simulations is low on cloud. Clusters are and will continue to be commonly used for tightly-coupled HPC problems, unless Clouds can offer a similar performance at a more competitive price. Perhaps, it will change when hardware I/O virtualization starts to be commonly used. Unfortunately, as mentioned earlier, due to lack of I/O virtualization in our hardware we were not able to check impact on performance of I/O hardware virtualization. We consider I/O HAV as important factor of cloud performance, thus we address problem of performance evaluation using environment with I/O HAV as future work. Since high computational intensity HPC applications running on clusters will remain with us for some time to come, it is essential to use power saving measures in cluster nodes and server rooms to limit the negative environmental impact of these systems.

Acknowledgements „This work was supported by the European Union from the European Social Fund (UDA-POKL.04.01.01-00-106/09-00 Ł. Janik)".

BIBLIOGRAPHY

1. Foster I., Zhao Y., Raicu I., Lu S.: Cloud Computing and Grid Computing 360-Degree Compared. 2008 Grid Computing Environments Workshop, Nov. 2008, p. 1÷10.
2. Armbrust M. et al.: Above the Clouds : A Berkeley View of Cloud Computing Cloud Computing : An Old Idea Whose Time Has (Finally) Come. Computing, 2009, p. 07÷013.
3. Liu X., Yuan D., Zhang G., Chen J., Yang Y.: Handbook of Cloud Computing. 2010.
4. Hill Z., Humphrey M.: A quantitative analysis of high performance computing with Amazon's EC2 infrastructure: The death of the local cluster? 2009 10th IEEE/ACM International Conference on Grid Computing, Oct. 2009, p. 26÷33.
5. Napper J., Universiteit V.: Can Cloud Computing Reach The TOP500 ? Categories and Subject Descriptors. October, p. 17÷20.

6. Ekanayake J., Qiu X., Gunarathne T., Beason S., Fox G.: High Performance Parallel Computing with Cloud and Cloud Technologies. Technology, p. 1÷39.
7. Walker E.: Benchmarking Amazon EC2 for high-performance scientific computing. 2008, Vol. 33, p. 18÷23.
8. Juve G. et al.: Scientific workflow applications on Amazon EC2. 2009 5th IEEE International Conference on E-Science Workshops, Dec. 2009, p. 59÷66.
9. Martinez J. C., Wang L., Zhao M., Sadjadi S. M.: Experimental study of large-scale computing on virtualized resources. Proceedings of the 3rd international workshop on Virtualization technologies in distributed computing - VTDC ’09, 2009, p. 35.
10. Ec A., Cloud U. E., Ward J. S.: A Performance Comparison of Clouds. 2009,.
11. Minartz T., Kunkel J. M., Ludwig T.: Simulation of power consumption of energy efficient cluster hardware. Computer Science - Research and Development, Jul. 2010,.
12. Sullivan R. F.: The impact of Moore’s Law on the total cost of computing and how inefficiencies in the data center increase these costs. 2007. [Online]. Available: http://findarticles.com/p/articles/mi_m5PRC/is_1_113/ai_n25007422/?t. [Accessed: 02--Jan-2012].
13. Pope S. B.: Turbulent Flows. Cambridge University Press, p. 382÷383.
14. http://www.rackservers.com/. [Online]. Available: http://www.rackservers.com/. [Accessed: 03-Aug-2011].
15. EC2 pricing. [Online]. Available: http://aws.amazon.com/ec2/pricing/. [Accessed: 04--Aug-2011].

Wpłynęło do Redakcji 30 stycznia 2012 r.

Omówienie

Praca skupia się na porównaniu chmur obliczeniowych i klastrów wysokowydajnościowych w celu oceny ich przydatności do obliczeń naukowych i inżynierskich. Autorzy w ramach projektu skonfigurowali chmurę lokalną, składającą się z czterech węzłów (kontroler chmury, serwer NFS, dwa węzły obliczeniowe – tabela 2) i zarządzaną przez środowisko chmurowe Eucalyptus (rys. 2). W celu oceny wydajności na klastrze wysokowydajnościowym (HPC) został użyty klaster bazujący na połączeniu Infiniband i Xeonach z serii 51xx. Wydajność jednostkowa została oceniona za pomocą HPL (High Performance Linpack) i jest porównywalna dla obu środowisk. Przed rozpoczęciem porównań i oceny wydajności autorzy wypunktowali zalety i wady obu architektur w postaci tabeli 1. Celem pracy jest ocena przydatności środowisk pod kątem obliczeń CFD, dlatego też wydajność jest oceniana na podstawie czasu obliczenia przygotowanych wcześniej modeli. Symulacje są wykonywane za pomocą dwóch pakietów do obliczeń CFD – StarCD oraz Fluent. Symulacje CFD zostały przygotowane w taki sposób, aby przetestować wydajność dla najczęściej spotykanych modeli matematycznych w symulacjach CFD. Koszty korzystania z publicznej chmury obliczeniowej zostały oszacowane na podstawie cennika Amazona EC2 dla UE.

Eksperymenty wykazały wyższą skalowalność i wydajność klastra HPC. Skalowalność w stosunku do chmury wzrasta zdecydowanie dla dużych modeli, w których symulacje jest zaangażowanych wiele rdzeni. Przewaga HPC jest jeszcze wyraźniejsza, gdy pojawia się częsta wymiana danych przez sieć. Przyczyną tego są opóźniania protokołu Gigabit Ethernet, które są znacznie wyższe niż dla Infiniband. Symulację kosztów wykonano dla okresu pięciu lat, który jest średnim czasem eksploatacji klastra. Symulacja wykazała, iż dla najczęściej spotykanych wzorców wykorzystania klastrów, używanie klastra w pięcioletnim okresie eksploatacji jest tańsze niż używanie chmury – tabela 5. Używanie chmur z kolei jest tańsze w przypadku bardzo nieregularnego zapotrzebowania na moc obliczeniową.

Na koniec autorzy ponownie wyliczyli zalety chmur, jak elastyczność, abstrakcja nad warstwą sprzętową, przenośność. Wykonane eksperymenty wykazały, iż obecna oferta chmur publicznych ze względu na wysokie koszty i małą wydajność nie nadaje się do obliczeń wymagających częstej komunikacji między wątkami, jak ma to miejsce w przypadku symulacji CFD.

Addresses

Łukasz J. JANIK: Silesian University of Technology, Institute of Informatics, Akademicka 16, 44-100 Gliwice, Poland, Lukasz.Janik@polsl.pl.

Stuart E. BARNES: Cranfield University, School of Engineering, Applied Mathematics and Computing Group, Cranfield, United Kingdom , s.e.barnes@cranfield.ac.uk.