Moving Multimedia Simulations into the Cloud: a Cost-Effective Solution

Moving Multimedia Simulations into the Cloud: a Cost-Effective Solution

Enrico Masala
Control and Computer Engineering Department
Politecnico di Torino
Torino, Italy
Email: masala@polito.it

***Abstract*—Researchers often demand bursts of computing power to quickly obtain the results of certain simulation activities. Multimedia communication simulations usually belong to such category. They may require several days on a generic PC to test a comprehensive set of conditions depending on the complexity of the scenario. This paper proposes to use a cloud computing framework to accelerate these simulations and, consequently, research activities, while at the same time reducing the overall costs. A practical simulation example is shown, representative of a typical simulation of H.264/AVC video communications over a wireless channel. This work shows that, by means of a commercial cloud computing provider, the gains of the proposed technique compared to more traditional solutions using dedicated computers can be significant in terms of speed and cost reduction.**

***Keywords*-Multimedia communications; transmission simulations; cloud computing; Amazon EC2**

I. Introduction

Simulations are a key part of the research activity in digital communication systems since they allow to validate the behavior of the proposed techniques that would be too complex to study analytically. Indeed, to achieve significant confidence levels, the performance of the systems under test must be evaluated in many different conditions, such as using various bitrates, noise levels, rate control algorithms, etc. Moreover, the same experiments are repeated tens of times to achieve statistical significance, and need to be rerun quite often as new fixes and improvements are developed.

Running such simulations may require up to a few days when they are particularly complex. Results are then used to improve the developed techniques, thus the amount of time spent in simulations is a significant share of the whole duration of the research activities. Since the simulation is part of a develop-simulate-improve cycle, any possibility of increasing the simulation speed immediately reflects in a reduction of the total development time of the techniques.

Many of the simulations considered in this work are easily parallelizable, since different runs of the same simulation engine with different input parameters are independent and can be run on different computers. However, computers may not always be available in the organization, may not have enough power to run them efficiently due to lack of, e.g., fast CPUs or memory, since the available computers might be old. Buying new computers is not always an option since their usage ratio might be low and management costs must also be considered, therefore the overall price may increase.

Luckily, nowadays resources can be rent from a cloud computing provider, therefore simulations can easily be run in parallel while paying only for the resources effectively used. While many works addressed the issue of employing cloud computing for the class of so called high performance computing (HPC) problems, there is a significant lack of works in the scientific literature that focus on small scale research simulations, such as the ones targeted by this work, in terms of efficiency and costs with reference to a practical case. This work provides some quantitative results to help identifying the main tradeoffs involved in choosing between cloud computing and dedicated computers, highlighting advantages and disadvantages of both.

Some issues about using cloud computing for scientific workloads need to be investigated, since it is still not clear if a cloud architecture designed to support mainly web and small database workloads can cope well with scientific computing workloads which have different requirements [1]. However, running computationally heavy scientific applications has been reported by many to achieve significant savings in terms of costs with respect to owned resources, due to the cost model employed in cloud computing: pay only for the used resources, administration and maintenance costs split among all the cloud users [2].

Although the performance of general purpose cloud architectures, such as the one provided by Amazon [3], are usually up to an order of magnitude lower than those of conventional HPC clusters [4], the savings they can provide make them a good candidate for scientific workloads that require resources in an instant and temporary way [1]. Most of the research works in this field often employ benchmarks which try to predict the performance of complex scientific applications. Others, such as [5], focus on performance bounds for the scenario of a large number of nodes and

high parallel tasks competing for CPU and communication resources.

The types of simulation addressed in this work, instead, involve a limited number of nodes and very few interactions between them. In one sentence, they could be defined as too large to run on just a few PCs in a reasonable time, and too small to sustain the investments in dedicated computers necessary to achieve a significant speedup. This paper tries to quantify, with an actual simulation example, the economic advantages and drawbacks of moving into the cloud the typical simulations carried out in a small multimedia communications research lab.

Some works focused on the execution performance of typical scientific workloads using a cloud computing solution comparing it to traditional approaches based, e.g., on workstations, clusters, or HPC shared resources, as in [6], which particularly focus on predicting the performance with good accuracy. However, the work do not present comparisons in terms of economic costs. The issue of porting a scientific tasks typically run on a cluster to the cloud architecture is considered in [7], which first develop a method to recreate the cluster in the Amazon infrastructure through EC2 instances, then focus in particular on how the storage subsystem influences the performance. Economic costs are investigated in details for the cloud solution, however no comparisons with the cluster architecture are shown.

To the best of our knowledge, no works focused on the potential gains, especially in economic terms and improved development time, that can be achieved on the relatively small size simulations addressed here. Although this may seem a quite specific scenario, it is representative of a large number of situations since many multimedia communications researchers need to perform simulations of the size considered here. Note also that the constant increase of multimedia quality (e.g., in terms of video resolution and bandwidth) makes the computational requirements of the simulations constantly growing, hence it is definitely interesting to find cost effective solutions to run them.

This work presents both an architecture to run such types of simulation in the cloud and an investigation of the tradeoffs involved in moving the simulations into the cloud, with reference to an actual simulation example of H.264/AVC video transmissions on a packet lossy network. Prices set by a commercial cloud provider are also considered to assess, in practical terms, the costs of such a solution.

This paper is organized as follows. Section II describes the typical simulations characteristics in multimedia communication research and discuss their suitability for the cloud. Section III describes an architecture to efficiently run such types of simulations in the cloud. In Section IV, the offer of the Amazon cloud computing platform, employed in this work, is discussed in terms of cost effectiveness. Section V provides details about the simulation example used in this work, followed by Section VI which contains a comparison of the performance that can be achieved using either the cloud or dedicated computers. Conclusions are drawn in Section VII.

II. Multimedia Simulation Characteristics

Typically, the results of simulations of multimedia communication systems are averaged over a number of different random channel realizations to achieve statistically significant results. Therefore, such runs, which differ only for the initial random seed, are easily parallelizable. Moreover, several values of the key parameters of the models used in the simulations must be tested in order to get a comprehensive view of the performance of the system under test as a function of various signal characteristics and channel or network conditions. Finally, different test signals, i.e., video sequences, are generally used to ensure that good performance is maintained across a wide range of input signals. Other typical simulation activities include heavy precomputation tasks on pre-processed and encoded multimedia. In this case, information is computed and later used to optimize the communication with low-complexity algorithms.

In general, it can be concluded that multimedia communication simulations require very little effort to speed up them by means of parallel execution. These parallel tasks can run concurrently both within the CPU, using all the available cores of a multi-core architectures, and at the CPU level using more than one processor. This is important since, as modern computers, also cloud platforms provide (virtualized) CPUs which include more than one core which clearly need to be fully exploited in order to optimize the cost performance tradeoff.

Finally, note that multimedia simulations are mainly CPU-bounded, and the heaviest requirement not concerning CPU is imposed on the I/O subsystem to read and write uncompressed video sequences. Such requirement is often mitigated by the operating system disk cache which transforms the I/O throughput requirement into memory requirements which can be managed much more easily, e.g., using a suitable amount of RAM.

III. Architecture for Cloud Simulation

This work focuses on the Amazon AWS offer as of January 2012. One of the basic elements is the so called “Elastic Compute Cloud” service, often abbreviated as “Amazon EC2”, which comprises a number of virtual machine types (“instances”) with a different amount of CPU, RAM and I/O resources. Both an API for various programming languages and a web interface are available in order to control the deployment of instances in the cloud platform. In both cases, creating new instances in the cloud and monitoring them require a number of operations. Basic administration procedures can be directly carried out by the user by means of the web based interface. However, when more complex

Table I
AVAILABLE EC2 INSTANCES, WITH FEATURES AND COSTS, IN THE EU REGION.

Instance name	Cores/instance	ECU/core	Total ECU	RAM (GB)	$/h	$/(ECU·h)
micro	up to 2	1	up to 2	0.613	0.025	-
std small	1	1	1	1.7	0.095	0.095
std large	2	2	4	7.5	0.38	0.095
std xlarge	2	4	8	15	0.76	0.095
hi-cpu medium	2.5	2	5	1.7	0.19	0.038
hi-cpu xlarge	2.5	8	20	7	0.76	0.038
hi-mem xlarge	3.25	2	6.5	17.1	0.57	0.088
hi-mem dxlarge	3.25	4	13	34.2	1.14	0.088
hi-mem qxlarge	3.25	8	26	68.4	2.28	0.088

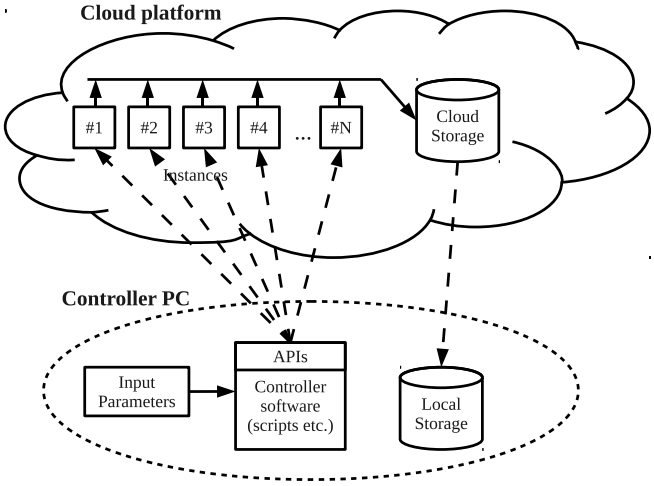

Figure 1. General architecture to run multimedia communication simulations in the cloud.

configurations are needed, e.g., setting up several instances at the same time to quickly run a simulation, it is strongly preferable to have a client program which quickly handles the operations by means of the API, launching the requested part of simulation on the right instance and eliminating the probability of human errors.

Therefore, a simple software has been written specifically for this aim in order to create and terminate instances, to run the requested simulations, to monitor the execution status and to collect the results. This architecture is shown in Figure 1. For the purpose of saving the results, data are temporarily stored in the S3 storage system provided by Amazon, then they are later downloaded to the user's PC. The software acts as the controller of the simulation, it runs on a controller PC, typically the researcher's own computer, and it can periodically check the status of the simulation or collect the results at the end by means of specific options. Moreover, note that before running the simulations, a virtual machine image, named AMI by Amazon, must be prepared with all the tools needed to perform the simulation and a script that can run a given set of simulations on the basis of the content of a given input file. The AMI is then instantiated multiple times to run different parts of the simulation set. During this phase the video sequences can also be included into the AMI, since sequences used in multimedia experiments are generally four or five, which are enough to represent a number of typical characteristics of multimedia signals. The Amazon fee is very low, i.e., 0.15 $ per GB stored for one month. Once the simulation is run, sequences will be held in the Amazon Elastic Block Storage (EBS) which is automatically connected with the virtual machine when it is instantiated.

The controller software can be configured to use various instances, in terms of number and types. However, deciding which are the most suitable ones, in terms of cost performance tradeoff, for the given simulation is not trivial and it will be the subject of the rest of this work.

IV. AMAZON EC2 PERFORMANCE

The Amazon measure of computing power relies on the so called EC2 compute unit (ECU), defined as the equivalent to the CPU computing power of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. The characteristics of the Amazon EC2 offer as of Jan. 2012 [8] are listed in Table I. Apart from the *micro* instance type, all the other instances can be grouped into three different families, namely *std*, *hi-cpu* and *hi-mem*. Within each family, the cost per hour per nominal ECU provided is the same. Therefore, from the economic point of view, within the same family, doubling the computing power requirement correspond to doubling the costs regardless of the number of instances actually employed.

The remainder of this section aims at investigating the actual computing performance and checking the correspondence with the declared CPU performance in terms of ECUs. First, the computing performance of the different instances has been tested by using a simple CPU-intensive program, i.e., decoding an H.264/AVC encoded video, which is a task very similar to the one that will be performed in our sample simulations. Each experiment is repeated 10 times. The resulting uncompressed video file is written into RAM, so that the storage system performance does not influence the measurements.

Table II
CPU TIME AND RELATIVE PERFORMANCE OF VARIOUS EC2 INSTANCE TYPES, USING ONLY ONE CORE. VALUES REFER TO THE H.264/AVC DECODING TASK. THE *std small* IS ASSUMED TO PROVIDE 1 ECU AS DECLARED.

Name	Nominal ECU/core	Time (s)	Measured ECU (1 core)
std small	1	130.1	1.00 (assumed)
std large	2	85.4	1.52
std xlarge	2	67.2	1.94
hi-cpu medium	2.5	57.6	2.26
hi-cpu xlarge	2.5	61.9	2.10
micro	up to 2	356.1	0.37

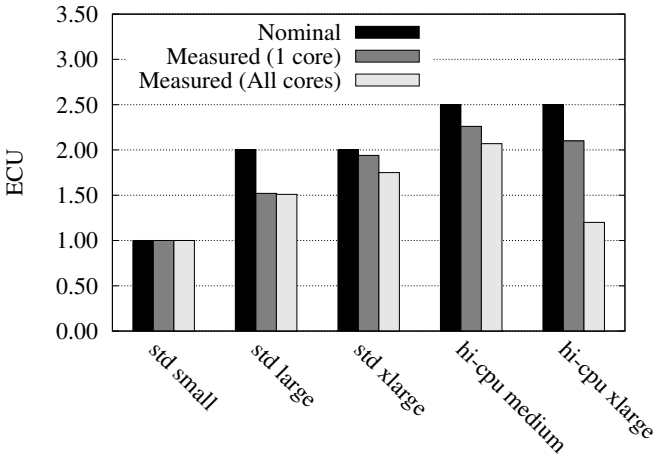

Figure 2. Computing power (normalized by the number of used cores) of different instances measured by means of the H.264/AVC decoding task (dark grey bar refer to the use of one core only, light grey bar refers to running as many tasks in parallel as the number of cores available in the tested instance).

Table II shows the results. Assuming that the *std small* instance provides exactly 1 ECU as declared, the effective ECU of the other instances (when only one core is used, i.e., no task parallelization is employed) is inversely proportional to the time required to complete the operation. Since the nominal value for *std small* is exactly one, the values of the last column in Table II can be directly compared with the nominal ECU/core of each instance.

Note that the nominal value of the *micro* instance is not an absolute value but an upper bound. Amazon, in fact, declares that the instance is not suitable for a continuous load, since it is designed for instances which are mostly idle but sometimes have to deal with short bursts of loads, as in the case of, e.g., web servers which are rarely accessed. Trying to load the *micro* instance with continuous computing operations for more than few seconds results in a strong slow-down for several tens of seconds. Therefore, the value shown in the table is the average of several cycles in which the instance runs at either full or strongly reduced speed. Since this instance is not aimed at heavy computing load, it will not be considered in the rest of this work.

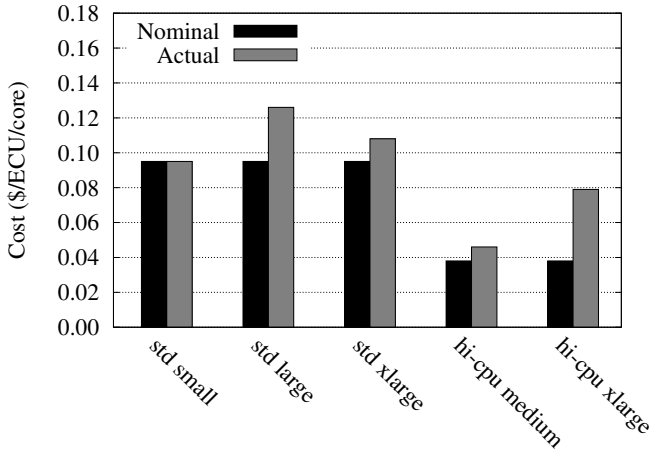

Figure 3. Cost of each instance, normalized by the provided ECUs and number of cores. "Actual" refers to the use of the measured ECUs instead of the nominal ones.

Many instances also provide multi-core CPUs, therefore more than one process can run in parallel on the same instance. The previous H.264/AVC decoding experiment has been repeated by running more than one process at a time on the same instance, up to the number of cores available in that instance.

Figure 2 shows the actual computing power as measured with the H.264/AVC decoding tasks. The light grey bar refers to running as many tasks in parallel as the number of cores available in the tested instance. Values show that the *std xlarge* and *hi-cpu medium* instances offer performance quite close to the ones declared by Amazon. The *std large* instance provides almost the same computing performance per core when using one or both cores. On the contrary, the *hi-cpu xlarge* instance particularly suffers from increasing the number of parallel tasks up to the number of available cores. In the latter case, the performance is nearly halved.

Thus, it is possible to compute a cost value, for each instance, for each unit of *actual* computing power, as shown in Figure 3. From this point of view, the most convenient instance is the *hi-cpu medium* one. Therefore, for the same cost, it is better to use four *hi-cpu medium* instances rather than one *hi-cpu xlarge* instance, which is however advertised by Amazon as having four times the computing performance of the former — and proportionally priced.

Note that the H.264/AVC decoding task does not pretend to be an exhaustive benchmark of the computing performance of the instance but it is employed here in order to give at least a rough indication of how suitable each type of instance is for the purpose of running the typical software needed to evaluate the quality needed by multimedia communication simulations. However, it must be noted that this test only measures the CPU performance while video communication simulations also include some,

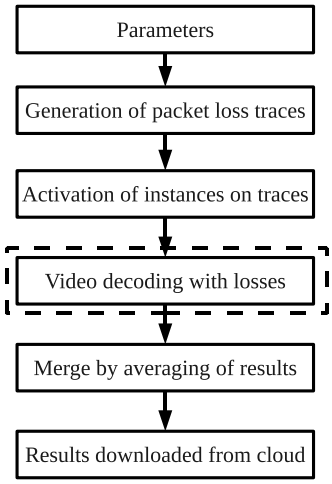

Figure 4. Steps of the video communication simulation. The most complex task is highlighted by a dashed box.

although moderate, I/O activity.

V. Simulation Setup

A typical set of video communication simulations has been considered in order to investigate the cost-performance tradeoffs involved in moving such tasks into the cloud. The simulation scenario consists in a compressed H.264/AVC [9] video stream which is transmitted in an IEEE 802.11 wireless network [10] where the packet loss probability depends on the characteristics of the physical channel and packet size. The experiments are repeated for several values of bandwidth, packet sizes, channel SNR, and using different standard test video sequences. Moreover, for each condition (formed by the combination of the previous parameters) the random channel has been simulated many times (50 in our tests) to compute statistically significant results.

Figure 4 shows the steps of the video communication simulation. The most complex task of these simulations is to decode the corrupted video bitstream so that a quality measure can be computed with respect to the original uncompressed error-free video sequence.

Here we consider a typical set of simulations which includes 5 different values for three different parameter, e.g., SNR, packet size and bitrate, 50 channel realizations and 4 video sequences of 300 frames each at CIF resolution (352×288 pixels). The time required to run such a simulation *without parallelization* on a computer with an Intel i5 M560 processor at 2.67 GHz and 4 GB RAM is 144,708 s, i.e., about 40 hours, nearly two days. About the same time is needed by a more expensive computer based on an Intel i7 2630QM processor at 2.00 GHz with 8 GB RAM, which needs 144,030 s (again without parallelization.)

However, note that for larger resolutions, that are increasingly used also in wireless networks, the duration can be even higher. Indeed, video decoding time is proportional to the number of pixels in the frame.

Clearly, if the simulation set needs to be run several times, as it is the case every time improvements of the optimization algorithm need to be tested, the speed of the development process could be significantly affected, hence speeding up the simulations while limiting costs is definitely interesting.

VI. Results

Every set of simulation experiments requires a certain amount of computations to be performed. To determine this value, we executed a small set of the simulation experiments involving ten loss traces both on the *std small* instance and on the Intel i5 computer. Times are, respectively, 187.60 s and 57.88 s, hence assuming that the *std small* provides 1 ECU for this particular type of task, the i5 processor provides a computing power equal to 3.24 ECU. Thus, the workload required by the whole set of simulation experiments, which is composed by 25,000 loss traces, can be expressed as 130.28 ECU·h, where ECU times hours is a measure of computing "energy" similarly to the case of measuring the amount of energy provided by the electric power grid using KWh.

If a given owned computer is used to run these experiments, the simulation speed is limited by the maximum computing power of all the available cores in the processor, as it is the instantaneous amount of electrical power that can be drawn by the grid given, e.g., the cable infrastructure to which the user is connected.

A cloud computing infrastructure, instead, allows to freely trade off computing power for execution time, while keeping the cost constant. As a consequence, simulation experiments can be speeded up as needed. This is indeed the case of the Amazon infrastructure when considering instances belonging to the same family. In fact, Table I shows that the cost of per unit of computing "energy" (ECU·h) is constant within the family. Clearly this scenario implies that simulations can be run as a number of parallel processes high enough to keep all the computers fully loaded. This is indeed quite reasonable since the analysis of the requirements of this type of simulations described in Section II shows that the tasks are highly parallelizable.

In order to provide some quantitative results with the sample simulation set previously described, we assume that the cost of the computer based on the *i5* processor mentioned before, which contains two cores that can perform computations independently, is about 1,000 $, while the system based on the *i7* processor has been recently bought for about 2,600 $. Note however that the analysis provided in this work is still reasonable even if those figures vary but the order of magnitude remains the same. It is indeed very difficult to assign economic values to hardware since its cost depends on many factors and quickly changes depending on the technological advances.

Table III

THEORETIC TRADEOFF BETWEEN DURATION AND COST TO RUN THE SIMULATION DESCRIBED IN THIS PAPER. COLUMN "S" REFERS TO THE NUMBER OF SECONDS FOR EACH INSTANCE, "H" TO THE NUMBER OF HOURS FOR EACH INSTANCE (ROUNDED UP TO THE NEAREST INTEGER), $ TO THE TOTAL COSTS OF RUNNING ALL INSTANCES (THEIR NUMBER IS SHOWN IN THE FIRST COLUMN) FOR THE GIVEN NUMBER OF HOURS "H".

Instance type	*std small*			*std large*			*std xlarge*			*hi-cpu medium*			*hi-cpu xlarge*		
# instances	s	h	$	s	h	$	s	h	$	s	h	$	s	h	$
1	469005	131	12.45	117251	33	12.54	58626	17	12.92	93801	27	5.13	23450	7	5.32
2	234503	66	12.54	58626	17	12.92	29313	9	13.68	46901	14	5.32	11725	4	6.08
3	156335	44	12.54	39084	11	12.54	19542	6	13.68	31267	9	5.13	7817	3	6.84
4	117251	33	12.54	29313	9	13.68	14656	5	15.20	23450	7	5.32	5863	2	6.08
5	93801	27	12.83	23450	7	13.30	11725	4	15.20	18760	6	5.70	4690	2	7.60
6	78168	22	12.54	19542	6	13.68	9771	3	13.68	15634	5	5.70	3908	2	9.12
7	67001	19	12.64	16750	5	13.30	8375	3	15.96	13400	4	5.32	3350	**1**	**5.32**
8	58626	17	12.92	14656	5	15.20	7328	3	18.24	11725	4	6.08	2931	**1**	**6.08**
9	52112	15	12.83	13028	4	13.68	6514	2	13.68	10422	3	5.13	2606	**1**	**6.84**
10	46901	14	13.30	11725	4	15.20	5863	2	15.20	9380	3	5.70	2345	**1**	**7.60**
11	42637	12	12.54	10659	3	12.54	5330	2	16.72	8527	3	6.27	2132	**1**	**8.36**
12	39084	11	12.54	9771	3	13.68	4885	2	18.24	7817	3	6.84	1954	**1**	**9.12**
13	36077	11	13.59	9019	3	14.82	4510	2	19.76	7215	3	7.41	1804	**1**	**9.88**
14	33500	10	13.30	8375	3	15.96	4188	2	21.28	6700	2	5.32	1675	**1**	**10.64**
15	31267	9	12.83	7817	3	17.10	3908	2	22.80	6253	2	5.70	1563	**1**	**11.40**
16	29313	9	13.68	7328	3	18.24	3664	2	24.32	5863	2	6.08	1466	**1**	**12.16**

Given the definition of the ECU (the amount of computation that can be performed by the *std small* instance) the computing power of the considered i5-based computer system is about 5.85 ECU when taking advantage of the two cores and the hyper-threading feature of each one of them, running 4 task in parallel. In case only two tasks ran in parallel, the total computing power of the considered i5-based computer system would be 5.56 ECU. For the i7-based computer system, the maximum computing power is 13.99 ECU, achieved by loading all the four cores with two tasks each. Thus the whole set of simulation experiments would require 80,111 s, i.e., more than 22 hours, on the i5 computer, while 33,522 s (more than 9 hours) on the i7 computer. These duration values are fixed and cannot vary without changing the type of the computer system, which would obviously affect its cost.

On the contrary, if tasks are performed into the cloud, their cost is fixed, determined by the instance type used to run the simulations, while the time needed to obtain the results varies depending on how much the tasks can be parallelized, i.e., the number of instances employed to run the simulation. In the previous example, using the cheapest *hi-cpu* family of instances, the cost of running the simulation, according to the nominal ECU values provided by Amazon, would be 4.95 $. Again, note that this cost is independent of the amount of instances used to run the simulations.

For completeness, it should be noted that the time overhead needed to activate and terminate instances is not included. In our experiments this time has always been negligible, less than one minute. However, it has been noted that in some unfortunate cases the time needed to acquire some instances can be up to two minutes [1].

Moreover, Amazon puts an economic incentive to avoid that the user activates and terminates instances too often. This is achieved by computing the cost of each instance in integer hours, i.e., fractional hours are rounded up to the nearest integer. Therefore, it is not convenient to keep instances active but idle, since the actual cost per hour that must be considered in our simulation costs increases. This also highlights the need of a software for automatic management of the instances in the cloud, so that all of them can start at the same time, with all the data required for the simulation automatically assigned to them, and terminated as soon as their tasks end to avoid having to pay for idle instances.

A. Theoretical Performance

The duration and cost of a simulation given the number of used instances can be computed as follows. The total simulation time is derived by the simulation workload expressed in ECU·h by dividing it by the total ECU power provided of each instance and by the number of instances activated in parallel. Such value is reported, in seconds, in Table III as the columns marked with "s". To compute the corresponding cost the simulation time, which is also the time each instance is active, needs to be rounded up to the nearest integer hour. Then that value is multiplied by the unitary cost of the instance and by the number of instances to obtain the total cost. Those values are reported in Table III as the "h" and "$" columns, respectively.

Note that the cost of the simulation is approximately constant since, as already stated, the same workload can be performed by more or less instances, running for less or more time. Without the Amazon policy about partial hours, it would be exactly the same, but such a policy creates small variations if the number of instances changes. Such variations are especially noticeable when the total duration of the simulation, i.e., the time each instance is active, approximates one hour. If the number of instances

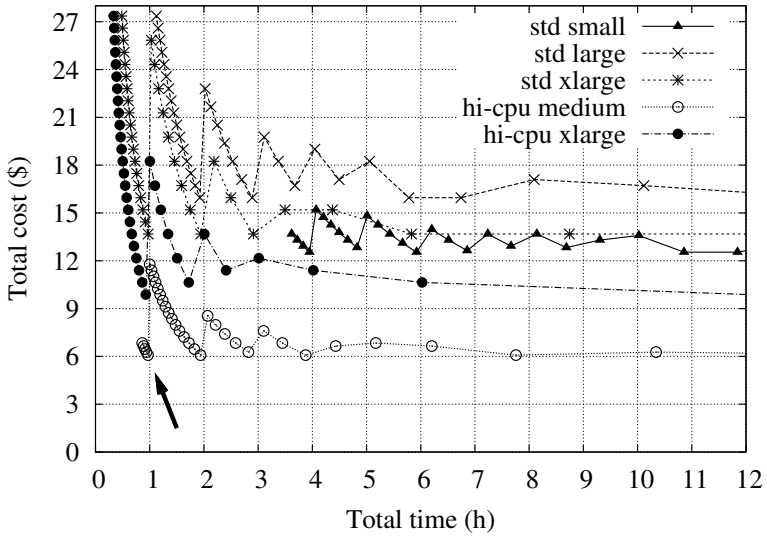

Figure 5. Performance cost tradeoff for various instances to run the whole simulation described in this work. The arrow indicates the point corresponding to the lowest cost, which implies using *hi-cpu medium* instances.

is high, in fact, instances may be underutilized because they are not exploited for all the time they are paid for, and since their number is high the cost of such instances proportionally increases the simulation cost. See, e.g., the price increase for the *std xlarge* instance type when the duration, rounded up, is two hours. Moreover, if the duration of the simulation is less than one hour, this is particularly evident. For instance, the simulation cost for the *hi-cpu xlarge* type of instances, highlighted in bold in Table III, shows a cost which monotonically increases, being, in this condition, inversely proportional to the duration of the simulation.

Note that the previous analysis does not take into account the cost of disk I/O transactions of the instances in the Amazon cloud platform, which are charged separately from instance activation costs. However, in all our experiments, these costs have always been negligible compared to the cost of the instances. Moreover, Amazon charges only for actual disk transactions, therefore the operating system disk cache contributes to greatly mitigate this issue.

B. Actual Performance

The previous results are based on the nominal ECU values provided by Amazon for each instance. Experiments with the Amazon instances are needed in order to assess actual running times of the simulation tasks on each type of instance. These experiments also allow to identify which is the best instance type, in terms of the cost performance tradeoff, to run our simulation. Given a new type of simulation, this test should be the preliminary step so that the cloud services are then exploited with the maximum efficiency.

Figure 5 shows the actual costs and total duration that could be achieved using the Amazon cloud infrastructure to run the simulations described in this work. For each instance, we run many tasks in parallel as the number of available cores in that instance. Every point in the graph is representative of a different tradeoff between cost and time, and corresponds to a certain number of instances. The data clearly shows that the best performance is achieved by using the *hi-cpu medium* instance type (the lowest point starting from the left, highlighted by the arrow in the graph). Since the task is computationally heavy with few disk accesses, instances belonging to the *hi-cpu* family are better suited for this simulation, as expected. Moreover, the better performance of the *hi-cpu medium* instances compared to the *hi-cpu xlarge* may be explained by the lower number of virtual cores in each instance. This condition seems to provide better performance in the EC2 infrastructure.

Figure 5 also shows that the actual behavior in practical cases can be quite different from the expected one derived from the nominal computing power values of the instances. This is particularly true when a high number of cores is involved. Hence it is clear that in order to exploit the cloud platform in the most cost-effective way a few preliminary experiments must be run in order to determine the instance type which is better suited for the particular simulations to run.

Note also that Figure 5 do not include the single point corresponding to run the simulation on a dedicated computer, since time and price would be 80,111 s and 1,000 $ for the i5 computer, and 33,522 s and 2,600 $ for the i7 computer. At the most convenient simulation price in the cloud (the point

highlighted by the arrow), i.e., 6.08 $ which correspond to 3,490 seconds, the number of simulations that could be run before spending the same amount of money invested in the computer is about 164 or 427 depending on the considered computer (i5 or i7), not counting the fact that the cloud provides results, at the same cost, much quicker than the computer. Indeed, the i5 computer needs about 22 hours whereas the i7 needs about 9 hours, compared to less than one hour of the cloud system.

Finally, note that in the computer price we did not consider a number of costs such as management and administration, potential failures of the hardware, and electricity. Other costs might be even more difficult to quantify, such as the fact that the developers of the algorithms to be simulated cannot work for many hours between each simulation run, because they must wait for the results, or one run of simulations may end at night when presumably developers do not work thus additional time is wasted, etc. In the case of the cloud computing solution, all the mentioned costs are zero — they are already included in the Amazon fee — and, working at the best tradeoff between cost and performance, every set of simulations terminates in about 1 hour, which makes the cloud particularly interesting for the time saving that allows a faster development cycle of the algorithms to be simulated, avoiding the need to buy several computers to achieve the same time performance.

VII. Conclusions

This paper addressed the issue of employing a cloud computing approach to run some common types of simulations typically performed by researchers in the multimedia communication field. The traditional approach to run this type of simulations on a dedicated computer has been compared with the case of moving the tasks to the cloud, where computing power can be rented as needed and correspondingly paid. An actual test case including a set of typical video transmission simulations over a packet lossy network has been considered to show, in a practical way, the savings that can be achieved in terms of time and costs, with reference to the offer of a major cloud computing provider such as Amazon. The results show that, given the current prices and policies of the considered cloud computing platform, the cloud approach is convenient, not only in economical terms, but also for the time saving that would allow a faster development cycle of the algorithms to be simulated, without the need to buy several computers to achieve the same time performance.

Acknowledgment

The author would like to thank Daniele Angeli for providing part of the software used to perform some of the simulations described in this work.

References

[1] A. Iosup, S. Ostermann, M. Nezih Yigitbasi, R. Prodan, T. Fahringer, and D.H.J. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 931–945, Jun. 2011.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," University of California, Berkeley, Technical Report UCB/EECS-2009-28, Feb. 2009.

[3] "Amazon Web Services," *URL: https://aws.amazon.com, last accessed: Jan 23, 2012*.

[4] E. Walker, "Benchmarking Amazon EC2 for high performance scientific computing," *;login: The USENIX Magazine*, vol. 33, no. 5, Oct. 2008.

[5] F.A.B. da Silva and H. Senger, "Scalability limits of Bag-of-Tasks applications running on hierarchical platforms," *J. Parallel Distrib. Comput.*, vol. 71, no. 6, pp. 788–801, Jun. 2011.

[6] Y. Simmhan and L. Ramakrishnan, "Comparison of resource platform selection approaches for scientific workflows," in *19th ACM Intl. Symp. on High Performance Distributed Computing (HPDC)*, Chicago, IL, Jun. 2010, pp. 445–450.

[7] K.R. Jackson, L. Ramakrishnan, K.J. Runge, and R.C. Thomas, "Seeking supernovae in the clouds: a performance study," in *19th ACM Intl. Symp. on High Performance Distributed Computing (HPDC)*, Chicago, IL, Jun. 2010, pp. 421–429.

[8] Amazon Web Services, "Amazon EC2 instance types," *URL: https://aws.amazon.com/ec2/instance-types, last accessed: Jan 23, 2012*.

[9] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, "Advanced video coding for generic audiovisual services," *ITU-T (2003)*, May 2003.

[10] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. ISO/IEC 8802-11, ANSI/IEEE Std 802.11, 1999.