A Cloud System Implementation for the Analysis of Civil Engineering Structures

J.M. Alonso[1], A. Alonso[2], P. de la Fuente[1], F. Gómez[2], V. Hernández[1], P. Lozano[1], A. Pérez[2]
[1]Institute of Instrumentation for Molecular Imaging, Universitat Politècnica de València, Valencia, Spain
[2] Department of Mechanics of the Continuous and Structure Theory, Universitat Politècnica de València, Valencia, Spain

Abstract - *This paper describes the development of a high throughput and reliable Cloud Service to perform on-demand structural analysis over the Windows Azure-based Cloud infrastructure provided by the EC VENUS-C project. All the simulations in the Cloud are governed by Architrave®, an advanced software environment for the design and analysis of buildings and civil engineering structures. The migration to the Cloud has been implemented by means of the Generic Worker component, a web-role implementation for Windows Azure that manages the execution of the remote tasks. The CDMI standard has been used for uploading and downloading data. A GUI Client has also been implemented, in charge of defining and managing the remote simulations, transferring the data and informing the user about the status of the simulations. A real large building has been chosen as a case study to show the advantages of the cost-effective Cloud system with respect to the sequential approach.*

Keywords: Cloud Computing, Cloud Service, structural analysis, VENUS-C Project

1 Introduction

Structural analysis of buildings, or civil engineering structures, is the process to determine the response of a structure to different prescribed applied loads. This response is usually measured by establishing the stresses, tensions and displacements at any point of the structural elements.

In a linear dynamic analysis [1], where the external loads (earthquake, wind load, etc.) change along the time, the second order differential equations in time that governs the motion of structural problems must be solved. Direct time integration algorithms are techniques usually applied for solving this computationally demanding equation of motion, using a time step-by-step numerical integration procedure that provides the response of the structure along the time. The accuracy of the results depends on the time increment employed.

The realistic 3D structural dynamic analysis of large scale structures can demand an important computational power, give place a huge volume of data and become one of the most time consuming phases in the design cycle of a building or a civil engineering structure. For this reason, this analysis has been traditionally solved by introducing a variety of simplifications (unsuitable for complex structures) in order to reduce the problem size and the volume of the data, and obtain the results in reasonable simulation times.

Architects and structural engineers need thus powerful software applications able to simulate efficiently the accurate response of the structure. High Performance Computing (HPC) techniques provide powerful numerical and programming tools to develop applications able to simulate, efficiently and in a realistic way, large dimension structures, in very reasonable response times. However, commercial applications usually offer traditional approaches, computing sequential structural analysis on the user's local machine. As a result, the size and the complexity of the structure to be analysed, the type of structural analysis employed and the total number of the different structural solutions or even earthquakes evaluated are limited by the performance of the computational resources used by the users. With the purpose of overcoming these limitations, Architrave® [2], an advanced software environment for the design, 3D linear static and dynamic analysis and visualisation of buildings and civil engineering structures, was developed. Architrave is composed of these three different and independent components, although interacting among them:

- The Design Component: An interactive AutoCAD-based application where the user can draw the model and define the structural properties and the external loads.
- The Analysis Component: An interactive GUI application, where the user can modify the structural properties and the external loads, analyse the structure and visualise the results.
- The Structural Simulator: A batch MPI-based parallel application used by the Analysis Component to simulate the response of the structure by means of the Finite Element Method.

Notwithstanding, studios for architecture and engineering rarely own parallel platforms to execute an HPC-based application. Thus, the users install and run Architrave in their personal computers, and the time spent on the calculations by means of the Structural Simulator component depends on the performance of their machines.

Fortunately, Cloud Computing technology has emerged as a solution for the computational requirements of organizations, enabling the usage of non-owned remote resources which delivers enough power and storage space to satisfy the computational and disk requirements of the resource-starved dynamic structural simulations of high-rise buildings. Moreover, Cloud technology allows sharing not only computing power, but also another kind of resources such as storage space, data and even software packages.

This document is composed of the following sections. First of all, the VENUS-C project and the Generic Worker component are described in Section 2. The porting of a structural analysis application to the Cloud is explained in Section 3. This section describes the design adopted for the Generic Worker execution model and the software architecture. The different VENUS-C components used are explained and the details of the deployment in the platform tested are finally exposed. Section 4 presents the representative case study selected in order to validate the solution implemented in this work. Finally, the Section 5 contains the conclusions.

2 The VENUS-C project and the Generic Worker component

VENUS-C (Virtual multidisciplinary EnviroNments USing Cloud infrastructures) [3] is a European Research Infrastructure Project that aims at providing an easy-to-use computing platform, based on the virtualisation and service orientation. In simple terms, it allows researchers to use the Cloud Computing model in science.

VENUS-C is associated with the concept of PaaS (Platform as a Service), which means that an application is inserted into an existing Virtual Machine (VM) image with an appropriate runtime environment, and then executed. The developer need not to maintain, manage or update the VM's operating system or runtime environment. Instead, he focuses on the functionality of his application. Also, this application can be easily scalable, secure and ensure high availability.

This is what provides other platforms like Windows Azure [4] or Google App Engine. However, the novelty of this initiative is, among other ones, the easiness of porting an existing application to the Cloud and the interoperability between different infrastructures without having to adapt the application. In this regard, VENUS-C exposes an OGF BES/JSDL [5][6] compliant web service interface and client libraries for Java and .NET.

VENUS-C is a platform composed by several components which provide different services. The two essential services are data and job management. The data management service includes transferring input data to the Cloud and retrieval of output results. It supports the Cloud Data Management Interface (CDMI) specification [7], developed by the SNIA, and considers also blobs from Windows Azure Storage for transferring data.

The job management service allows the scientist to allocate compute resources in the Cloud, submit tasks, and to manage a job's lifecycle, i.e. to monitor the execution status and terminate a job if necessary. Other services provided are Elasticity, Monitoring, Accounting, Billing, and traffic redundancy elimination.

There are two components available that support the job management service (Figure 1): the Generic Worker (GW) [8][9] from Microsoft Research, and the PMES-COMPSs [10] from the Barcelona Supercomputing Center (BSC). Each of these components has a different programming model. In the implementation for this work, the first of them has been used, which is intended to be deployed on the Windows Azure platform, running over VMs with Windows Server 2008 and .NET framework.

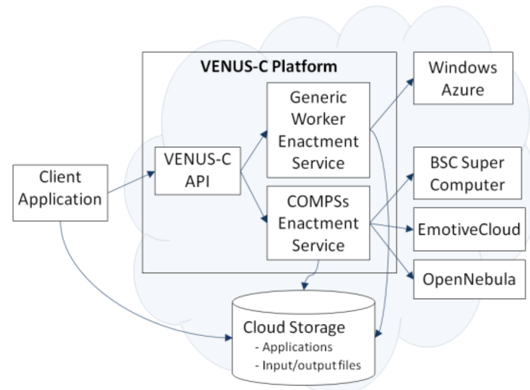

Fig. 1. The basic architecture of VENUS-C.

Actually, the GW is a Windows Azure web role attending for requests of registering a job, getting the status of a task, terminating a job, etc. (supporting, in this way, the VENUS-C API). In addition, this same web role has another process running that checks for new jobs registered to be processed. Obviously, one or many instances of this web role can be deployed working together in coordination.

The GW provides the common glue code that the developers have to write in order to port their applications to Windows Azure. The difference is that this service executes *generic* jobs, i.e. every job has a description that specifies the application to run, the input/output files and the parameters to be passed to the executable binary.

All interactions with the GW can be authenticated by the username and password mechanism and controlled by authorization policies, based on certain user roles previously defined and a table that specifies which roles has each of the users. Also, the communications can be protected through WS-Security with security tokens.

The GW exposes an additional interface to the administrator, through which the number of running instances can be scaled up/down to ensure the computational resource demands of the clients. Thus, the scaling decisions can be taken automatically according to different criteria.

3 Design and implementation

This section describes the design adopted to offer the capability to carry out remote structural analysis in the Cloud. The implementation was structured in different components and layers considering usability and easiness to adaptation to the features provided by the VENUS-C software.

The architecture of the Cloud Service implemented, based on the GW component, is shown in the Figure 2.

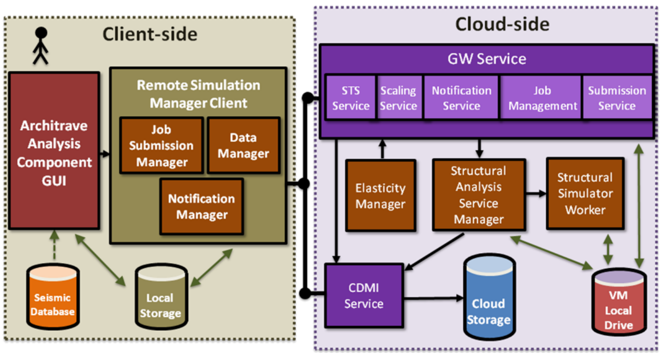

Fig. 2. Software architecture of the GW-based Cloud system.

Firstly, it consists on a Windows client that executes the Architrave Analysis Component to modify the structural properties, apply the external actions, define the simulation and visualise the results. In addition, a GUI tool, called as the Remote Simulation Manager Client, has been implemented to submit and manage the simulations in the Cloud and receive the results. Data communications between the local client and the Cloud service are carried out by means of the standard CDMI service [11]. On the Cloud-side, user authentication is implemented by means of the Security Token Service of the GW, and a Notification Service, also belonging to the GW, is used to inform the user about the status changes of the simulations. Remote jobs are managed and submitted thanks to the Job Management and Submission Service components of the GW. Structural simulations in the Cloud are executed by means of the Architrave Structural Simulator component. A Structural Analysis Service Manager module, in charge of launching the Architrave Structural Simulator Worker with the appropriate parameters, depending on the type of analysis, and managing the needed simulation input and output data has been implemented. Finally, an Elasticity Manager component has been developed to provide the system with the elasticity capability, using the Scaling Service of the GW. A more detail explanation is given in the following subsections.

3.1 The Remote Simulation Manager Client

On the client-side, the local or remote simulations are defined exactly in the same way, using the Architrave Analysis Component, the GUI application where the structural properties can be modified and response of the structures is visualized, after the simulation. This application now incorporates the Remote Simulation Manager Client, a tool developed also for .NET with a comprehensive GUI to configure and manage the simulations in the Cloud, to know in real-time the status of the remote executions and to download the results. This new application is launched by Architrave Analysis Component as an independent executable binary, which can even continue running (for downloading results, for example) when Architrave Analysis Component application is closed. In this tool, some modules can be distinguished:

- The Job Submission Manager: When the Remote Simulation Manager Client is launched, for the first time, this module checks the status of all the simulations in the Cloud (submitting, pending, running, finished, failed, downloading, downloaded) and informs the user. Then, this module will be responsible for submitting each new simulation to the Cloud.
- The Data Manager: This component uploads the input data and downloads the corresponding results and meta-data using a CDMI service or directly the Windows Azure Storage service.
- The Notification Manager: It obtains the changes in the simulation status and updates the associated information.

Every time the Remote Simulation Manager Client is launched, the user must be authenticated in the Cloud by means of the username and the password. If the user has permission to use the Cloud Service, i.e. he has been registered in the users table, then a list of his remote simulations will be shown, and he will be able to submit new simulation jobs, download results, cancel executions, and so on. For each simulation, geometric information related to the structure analysed, the type of analysis, the kind of data to be retrieved, the way of downloaded the results, etc. is exposed, including the current status, which is updated every time with a configurable frequency by the Notification Manager.

When the user submits a simulation to be executed in the Cloud, the following steps take place: First of all, the Architrave Analysis Component writes a binary file containing the building geometry, the external loads applied (such as an earthquake) and the simulation parameters, and sends a message containing the file path to the Remote Simulation Manager Client. Then, the Remote Simulation Manager Client reads the input binary file and the remote simulation is registered. Next, the Data Manager uploads the input binary file using the CDMI service to the CS. Finally, the Job Submission Manager submits the job to the Cloud Service in the way of a .jsdl document and the Notification Manager consults periodically the significant status changes in the Cloud Service. When the results are ready, the Data Manager will download them also by means of the CDMI service.

The user can configure the amount of results to be generated in the Cloud and to be received in the local machine, and the way to be downloaded. Thus, the Data Manager can be demanded to retrieve the simulation results automatically (when they are available) or manually (when the user requests them by clicking on the corresponding button). Also, in a dynamic analysis, the user can save on the CS the simulation results for all the time steps; the simulation results of just the 3 most unfavourable time steps according to the X, Y and module base shears or, finally, a video with the graphical response of the structure along the time. Besides, the user can configure the amount of data to be downloaded, obviously depending on the results previously computed and stored in the Cloud. In any case, a list with the X, Y and module base shears for all time steps will be always downloaded in order to allow the user to retrieve subsequently the amount of unfavourable and most significant simulation time step results that he desires.

Moreover, the Remote Simulation Manager Client can indicate the Data Manager to remove the input and output simulation data automatically (when the data results have been successfully received) or manually (when the user desires), to cancel the execution of a structural analysis or cancel the retrieval of the results.

3.2 The Structural Analysis Cloud Service

On the Cloud-side, the Structural Analysis Cloud Service is composed of the following components:

- The GW Service, i.e. a web-role implementation for Windows Azure that manages the execution of the remote tasks. This Service is composed of the Security Token Service (to perform the user authentication and allow that just authorized users can use it), the Scaling Service (to increase or decrease dynamically the number of worker instances), the Notification Service (to inform the user about the status of the jobs), the Submission Service (to send the simulations to the worker instances) and the Job Management (to provide information related to the status of the tasks).
- The Structural Analysis Service Manager, which generates the input files needed by the Structural Simulator and launches it and encapsulates the results.
- The Structural Simulator Worker, i.e. the HPC Architrave module that runs the simulations in the Cloud.
- The CDMI Service, which uploads and downloads the data using the standard CDMI protocol.
- The Elasticity Manager, which sets automatically the number of worker instances according to the workload.

For each simulation request, the associated input binary file is moved from the CS to the local drive of the VM by the GW Service, and the Structural Analysis Service Manager is launched. This executable file reads the input archive and generates the input file needed by the Structural Simulator Worker. Next, the Structural Simulator Worker is executed with the appropriate parameters, depending on the type of analysis, defined by the user. As a result, the structure is analysed and the output files are periodically generated and saved on the local drive. For each simulation time step, in case of a dynamic analysis with response along the time, all the multiple output files generated are encapsulated in just a single output file and sent to the CS during the execution by the Structural Analysis Service Manager.

A notification scheme is employed, where the Notification Service of the GW puts the status changes of the simulations into an Azure queue, which is consulted periodically by the Notification Manager component of the Remote Simulation Manager Client. In this way, the Remote Simulation Manager Client informs properly the user and downloads the output files to the local machine when the execution has finished, or when each simulation time step is ready to be retrieved, following a data-driven model. As a consequence, the remote simulation execution and the result retrieval phase are overlapped in time and, thereby, the whole time involved in the simulation is dramatically reduced.

The Structural Analysis Cloud Service contains an Elasticity Manager component able to adjust the number of available worker resources through the Scaling Service of the GW, depending on the workload in the system. Specifically, this component monitors continually the number of workers in execution, and a new instance worker is launched automatically when the processing workload is increasing. For that, when the number of instances waiting for jobs is lower than two, a new worker is deployed. On the other hand, if all the worker instances are idle, most of them will be terminated and the number of workers will reduced to the minimum.

4 Execution results and performance analysis

This section explains the representative case study applied to validate the Cloud system. The following subsections describe the structural and computational interest of the case study and the expected improvement benefits that provides the Cloud Computing solution.

4.1 Introduction to the case study

For the case study, we have selected a structure (Figure 3) corresponding to a reinterpretation, according to the current usages and regulations, of the original structure of the Nordic Bank, located in Helsinki. This is a work of the Finnish architect Alvar Aalto in 1962. It consists on a portal frame structure, solved by means of slabs over interior concrete columns and steel columns at the facade. The core of the vertical communication is materialized as reinforced concrete walls serving as vertical structure and lateral bracing. The spans are moderated, except in some zones at lowest levels where the foundation lab is reinforced with hanging concrete beams. The facade steel columns are disposed to half spans, with beams of diversion at the level of first floor. The foundation consists of a slab, due to the presence of the phreatic stratum. The model has been designed with bars for columns and beams, and 2D medium sized finite elements for slabs and walls. For the slabs, a Delaunay mesh has been employed, while the walls are solved with a simple mesh.

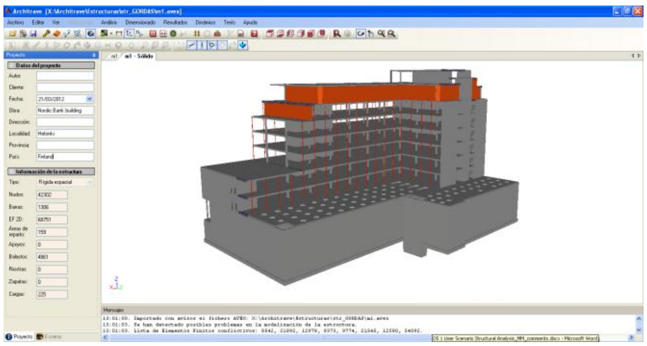

Fig. 3. Nordic Bank building.

The case study presented consisted on testing and simulating 10 different structural solutions that came from the same structural design, composed of 253812 degrees of freedom, with 1306 columns and beams, and 68751 2D finite elements. Each structural solution was composed of a variation of the column dimensions, and the slab and wall thickness. Moreover, each of them was dynamically tested under the influence of 5 representative earthquakes of 12 seconds of duration, with a simulation time increment of 0.01 seconds. These results were stored every 0.05 seconds. In this way, 1200 times steps were simulated and 240 of them were saved on disk. It should be noted that the aim of this case study was to select the best one of the 10 structural solutions that accomplished the structural and safety regulations and presented the most economic final cost.

Therefore, for the case study execution, a set of 50 independent 3D dynamic simulations were launched and computed, firstly locally in a conventional PC (Intel core i5, CPU@ 3.20 GHz and 4 Gbytes of RAM), and then in the Cloud, following two different configurations. On the configuration A, all the simulation time steps remotely saved were retrieved by the client; and on the configuration B, only the 3 most unfavourable time steps were saved and retrieved by the user. Furthermore, the two configurations were tested over different deployments composed of 1, 10, 25 and 50 medium-sized Web role Azure instances (CPU@ 1.60 GHz and 3.5 Gbytes of RAM).

4.2 Performance analysis

The validation tests demonstrate the behaviour of the VENUS-C platform according to the case study described in the previous section. For that, a quantitative evaluation has been carried out, measuring the efficiency of the platform by means of the response time and the speed-up, obtained by the Cloud approach with respect to the local approach and with respect to employ just one Azure instance in the Cloud.

Before executing the whole case study, the simulation of one of these structural solutions was computed firstly on premises and then in the Cloud. The Table 1 shows the results when computing the structural solution in the local machine of the user, together with the response time, the size of the input data and the amount of data downloaded by the local client, for the two described configurations.

Table 1: Results corresponding to the simulation of one of the structural solutions in the Cloud.

Type of execution	Response time (minutes)	Input data (Mbytes)	Output data (Gbytes)
Local execution - Saving the results of the 240 time steps	120.03	5.43	2.48
Configuration A	279.48	5.43	2.48
Configuration B	277.46	5.43	0.12

It should be clear that, for the configuration A, all the needed data movements (from the VM local disk to the CS and from the CS to the user machine) were overlapped with the execution. However, for the configuration B, the data transference just could start once the simulation had finished and the most adverse results had been computed.

The results of the Table 1 show a big difference between the local response time (120.03 minutes) and the remote response time (279.48 minutes for the configuration A, and 277.46 minutes for the configuration B).

For the itemization of the overhead times, the stages that give place to an overhead faced with the local execution have been analysed. Table 2 shows the time involved in each of the stages that compose the simulation of a single structural solution, where only the most unfavourable results are moved to the CS and downloaded later (configuration B). As it can be seen, the time involved in computation implies a delay of 155.28 minutes, with respect to the local execution.

Table 2: Execution times of each of the different stages involved in the simulation in the Cloud (configuration B).

	Data and job submission	Application download and job initialization	Structural Simulator Worker execution and result encapsulation	Result upload to the CS	Result download to the client machine
Local execution	-	-	120m 02s	-	-
Remote execution	6s	39s	275m 19s	50s	34s
Overhead time	6s	39s	155m 17s	50s	34s

The values of the Table 2 reflect that the most of the overhead time resides on the execution of the Structural

Simulator Worker, responsible of analysing the structure, and the Structural Analysis Service Manager, in charge of encapsulating the results generated. As the computational complexity is similar on the local and the Cloud simulation execution, it can be assumed that the main overhead resides on the difference on hardware characteristics, especially on the CPU features and the speed of accessing the disk for the read and write operations.

The overhead times at the configuration A shows a similar behaviour than in the case of the configuration B, as reflected in the Table 3. However, this is due to the fact that, as indicated above, the Structural Simulator Worker execution time is overlapped with the output data upload to the CS and with the result download to the client machine. In this way, the column 4 includes the time spent on the simulation execution, where the results of most of the time steps had been transferred simultaneously to the CS and received by the client. On the other hand, the column 5 shows the time required for the data upload to the CS and the result download to the client for the last time steps, i.e. once the Structural Simulator Worker execution has finished. Therefore, the main overhead appears during the simulation of the structure, with a delay of 154.88 minutes with respect to the local execution.

Table 3: Execution times of each of the different stages involved in the simulation in the Cloud (configuration A).

	Data and job submission	Application download and job initialization	Structural Simulator Worker execution and result encapsulation	Result upload to the CS and download to the client machine
Local execution	-	-	120m 02s	-
Remote execution	6s	39s	274m 55s	3m 49s
Overhead time	6s	39s	154m 53s	3m 49s

Once analysed the execution times of a single structural solution, the whole case study execution was performed, whose fifty different simulations were launched at the same time, and the status of the simulations were consulted periodically by means the Remote Simulation Manager Client. The evaluation compares the response time of the local execution, considering that each structural solution is analysed after another in the machine of the user, with the remote response time, for the two different previous configurations, over a set of different Cloud deployments composed of 1, 10, 25 and 50 medium-sized Azure instances. The response time was measured as the difference of time between the first job submission and the final result data downloaded corresponding to the last simulation.

The execution of the 50 simulations that compose the case study spent 100.03 CPU hours in a traditional approach, generating output results in the order of 124.04 Gbytes. In the Cloud deployment, the execution of the whole case use configuration A required 271.5 Mbytes as input data and 124.04 Gbytes of output data were produced. For the configuration B, 271.5 Mbytes were required as input data and 5.75 Gbytes of output data were generated.

Figure 4 shows the execution times of the whole case use, for both configurations and over a Cloud deployment composed of different number of Azure instances.

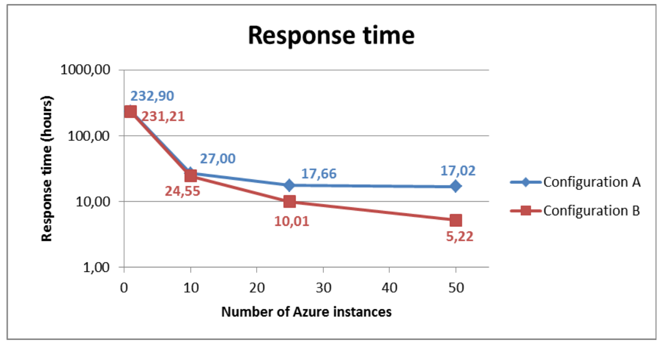

Fig. 4. Response time in hours for the whole case use.

As it can be noticed, the response time decreases gradually when the number of Azure instances is increased at the configuration B. In the case of the configuration A, the results show initially a similar trend. However, when the number of Azure instances that executes the case study is greater than 25, the results acquire a value of around 17 hours. This is due to the bottleneck of the network in the retrieval of this large volume of output data by the client, since the time required for downloading all the results generated is much greater than the time required for executing the simulations. Therefore, although the execution and the result retrieval stages are properly overlapped, the time required for the complete case study is mainly determined by the time spent on the reception of the 124.04 Gbytes of results generated.

In any case, it should be taken into account the dramatically reduction of the total time required for the execution. Whereas more than 4 days were needed in the traditional approach, just 17.02 and 5.22 hours were spent, respectively for the configurations A and B, when computing remotely all the simulations in the Cloud.

Figure 5 shows the behaviour of the Cloud system, when the number of Azure instances is increased, in terms of speed-up, with respect to the sequential approach (each structural solution analysed after another in the client local machine) and with regard to the results of the execution over just a single Azure instance. In this figure, it can be appreciated how the results of the speed-up, compared with the sequential approach, are much lower than results of the execution in an Azure instance, at the two configurations. This difference in the results is mainly due to the difference in the hardware characteristics between the local testing machine and the Azure instances.

Whereas the speed-up at the configuration B with regard to an Azure instance obtains values near to the ideal case (44.29 for 50 machines), the speed-up with respect to the sequential approach is far from these values (19.16 when using 50 instances). In the case of the configuration A, it can be appreciated how the value of the speed-up does not increase appreciably when the number of Azure instances is greater than 25, due to the reasons exposed above related to the bottleneck of the network in the retrieval of the results.

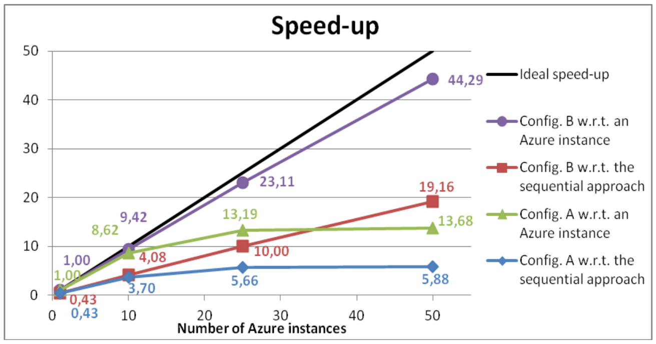

Fig. 5. Speed-up for the whole case use.

The behaviour, in terms of efficiency, of the Cloud system is reflected in the Figure 6, with respect to the sequential approach and with regard to an Azure instance.

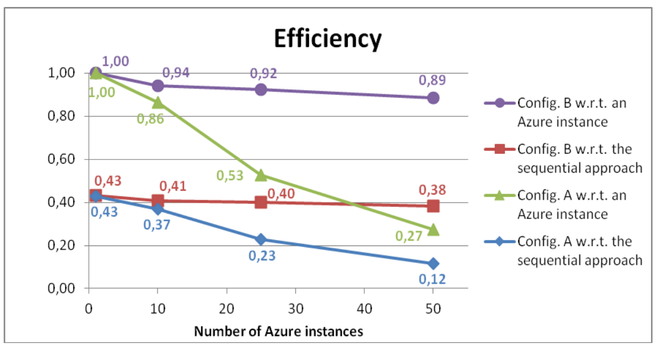

Fig. 6. Efficiency for the whole case use.

As expected, excellent efficiencies were obtained for the configuration B, but worse efficiencies were computed for the configuration A. Clearly, it can be noticed how the efficiency decrement is very low at the configuration B when the number of Azure instances is increased, whereas the tendency of the decrement at the configuration A is much more pronounced.

5 Conclusions

In this work, a Structural Analysis Cloud Service has been implemented, based on the GW component developed of the VENUS-C project, and deployed for Windows Azure. The software architecture of Cloud system is described, together with its design and the different elements that composed it.

All the remote simulations are managed by Architrave, an advanced software for structural analysis. The Architrave Analysis Component and the Architrave Structural Simulator have been properly adapted to work in the Cloud. In order to launch and manage the executions, the Remote Simulation Manager Client application has been implemented.

Thanks to the high throughput and reliable Structural Analysis Cloud Service implemented, researchers will have available a huge number of computational resources to be on-demand employed and lots of cost-effective simulations will be launched simultaneously. Thus, more experiments will be analysed per time unit, increasing the number of structures simulated and speeding up the research process.

The structural community will be able to solve larger scale problems, increase the complexity of the structure to be analysed, and carry out a larger number of realistic dynamic simulations. In this way, the reliability and safety of the results obtained will be improved and new structural problems will be tackled. Since the time spent on the design of buildings and civil engineering structures will be reduced, the engineering companies and the architectural studios will increase their productivity and volume of business.

Finally, there will be no need of acquiring software licenses in property and expensive hardware for solving large-scale structural problems (just pay per use), and the users will not be worried about new software updates.

6 References

[1] R.W Clough and J. Penzien. “Dynamics of structures”. Second Edition. Computers and Structures, Inc., 2004.

[2] Architrave website: http://www.architrave.es

[3] VENUS-C project website: http://www.venus-c.eu

[4] D. Betts, et al. “Moving Applications to the Cloud on the Windows Azure Platform”. Microsoft, 2012.

[5] I. Foster et. al. “OGSA Basic Execution Service Version 1.0”. Grid Forum Document GFD-RP. 108. 8/8/2007.

[6] A. Savva. “Job Submission Description Language (JSDL) Specification”. Version 1.0, GFD-R.056, 2005.

[7] SNIA CDMI: http://www.snia.org/tech_activities/ standards/curr_standards/cdmi

[8] Y. Simmhan, C. van Ingen, G. Subramanian, and J. Li. “Bridging the Gap between the Cloud and an eScience Application Platform”. Microsoft Research. U.S., 2010.

[9] Christian Geuer-Pollmann. “The VENUS-C Generic Worker”. 1st annual SICS Cloud Day, 2011.

[10] D. Lezzi, et al. “COMPSs in the VENUS-C Platform: Enabling e-Science Applications on the Cloud”. Proceedings of the IBERGRID 2011 Conference, Santander, 2011.

[11] I. Livenson, E. Laure. “Towards Transparent Integration of Heterogeneous Cloud Storage Platforms”. Fourth International Workshop on Data Intensive Distributed Computing (DIDC 2011), San José, 2011.