

ATINER CONFERENCE PAPER SERIES No: COM2013-0487

Athens Institute for Education and Research

ATINER



ATINER's Conference Paper Series

COM2013-0487

**Seismic Analysis using Cloud
Computing (Windows Azure)**

Mihail Mateev

Chief Assistant Prof.essor Eng. UACEG

Faculty of Structural Engineering,

Computer-Aided Engineering

Bulgaria

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece
Tel: + 30 210 3634210 Fax: + 30 210 3634209
Email: info@atiner.gr URL: www.atiner.gr
URL Conference Papers Series: www.atiner.gr/papers.htm

Printed in Athens, Greece by the Athens Institute for Education and Research.
All rights reserved. Reproduction is allowed for non-commercial purposes if the
source is fully acknowledged.

ISSN 2241-2891

2/08/2013

An Introduction to ATINER's Conference Paper Series

ATINER started to publish this conference papers series in 2012. It includes only the papers submitted for publication after they were presented at one of the conferences organized by our Institute every year. The papers published in the series have not been refereed and are published as they were submitted by the author. The series serves two purposes. First, we want to disseminate the information as fast as possible. Second, by doing so, the authors can receive comments useful to revise their papers before they are considered for publication in one of ATINER's books, following our standard procedures of a blind review.

Dr. Gregory T. Papanikos
President
Athens Institute for Education and Research

This paper should be cited as follows:

Mateev, M. (2013) "Seismic Analysis using Cloud Computing (Windows Azure)" Athens: ATINER'S Conference Paper Series, No: COM2013-0487.

Seismic Analysis using Cloud Computing (Windows Azure)

Mihail Mateev
Chief Assistant Prof.essor Eng. UACEG
Faculty of Structural Engineering,
Computer-Aided Engineering
Bulgaria

Abstract

Many practical tasks such as seismic analysis of structures require more computational resources, memory and data storage. These resources have used to solve certain tasks and then not be necessary. Cloud computing may become a major tool for researchers to access processors and storage. This paper refers to the use of Windows Azure, since most software used for seismic analysis work on Microsoft platforms. One of the popular cloud platforms - Windows Azure provides different opportunities for seismic analysis. This can has accomplished in several ways:

1. Using of existing software for seismic analysis in Windows Azure by VM Role (Migration of the existing design and analysis software for structural engineers in the Windows Azure)
2. Creating scalable PaaS applications for Windows Azure. It is possible to use existing modules or applications without UI.
3. Using the data in Windows Azure (Azure Blob Storage) data exchange between different desktop applications and applications running in Windows Azure.
4. Creating HPC applications using Parallel Processing specifically for Windows Azure.
5. Using a remote automation of Windows Applications by Azure Service Bus

This paper will demonstrate solutions based on the first three ways to use Windows Azure, mentioned above.

Comparison has made between the different approaches to analyze the advantages and disadvantages of each. The comparison has made in financial and technical aspects.

Keywords:

Corresponding Author:

INTRODUCTION

The aim of many professionals dealing with seismology is to be able to solve complex problems for seismic analysis related to the use of distributed and parallel computing resources. These resources may be systems for grid computing and / or cloud computing. A great example of such a decision is Network for Earthquake Engineering Simulation (NEES).

The NEEScomm (NEES Community and Communications) Team includes partners from the University of Washington, University of Texas at Austin, University of Kansas, San Jose State University, the University of Michigan, and Fermi National Accelerator Laboratory. NEES network features 14 geographically distributed shared-use laboratories that support several types of experimental work: geotechnical centrifuge research, shake table tests, large-scale structural testing, tsunami wave basin experiments, and field site research. The equipment sites (labs) and a central data repository have connected to the global earthquake engineering community via the NEEShub.

The NEES cyber infrastructure is the Information Technology that enables the Network of Earthquake Engineers (NEES) to work as an integrated co-laboratory (Figure 5). It is a collection of tools, servers, and networks to support NEES. The centerpiece of the NEES cyber infrastructure is the NEEShub website located at <http://nees.org>. The website servers and data servers for the project warehouse exist at Purdue University. Each NEES project has a complex directory structure (Figure 6).

OpenSees (Open System for Earthquake Engineering Simulation) is a software framework for developing applications to simulate the performance of structural and geotechnical systems subjected to earthquakes. The latest version of OpenSees is also available for execution on the NEEShub. This software is appropriate for large-scale structural and geotechnical simulations, structural and geotechnical mechanics and modeling, advanced computing and simulation. NEES includes different solutions in its network – Grid, clouds, stand-alone server.

Solutions like NEES are very expensive (it costs around 120 million USD). The objective of this research is to propose a relative cheap approach to build systems, solving complex problems for seismic analysis using distributed and parallel computing resources.

One of the possible solution is to use a public cloud, but which cloud model to use SaaS, PaaS or IaaS (Figure 1) IaaS and PaaS are used to develop cloud based solutions (we will be focused on these models). SaaS offers mainly end user solutions.

- **IaaS** gives the users low-level resource management mechanisms (VM provisioning, distributed storage, etc.) and it is up to them to develop their application stack (operating system, HTTP server, MVC framework, database, etc.). While this provides extreme flexibility for the user, it also means that they must manage scaling and fault tolerance on their own.

- Using **SaaS** users simply consume software that is hosted on third-party services. Both the application and the data is stored on a provider's server. This convenient for most users, but also possibly unacceptable to others who desire more control over their applications and data.

Many experts are asking how to bring their solutions for seismic analysis in the public cloud. The options are different:

- To migrate existing systems with minimal changes build solutions based on IaaS (Infrastructure as a Structure).
- To build new solutions based on PaaS (Platform as a Structure) using existing modules or applications. Solutions will be easily scalable. In this case, there will be additional work to integrate existing solutions.
- To build completely new solutions based on PaaS (Platform as a Service) - in this case can be used more effectively the benefit from the scalability of the systems in cloud computing, but this approach requires more additional development.
- To offer High-Performance Computing (HPC) Solutions. Commonly known as high-performance computing (HPC), this approach relies on executing code on many machines at the same time. To allow this, Windows Azure provides the HPC Scheduler (Figure 11).

This study focused on the first two approaches: Most of the existing software has designed as a standalone desktop application. It is too expensive to redesign it in the short-term period. Estimation of the expected efforts could be done with so-called “complexity evaluation” [10]. Usually changes in the existing code are very expensive. That is the reason to look for solutions where can integrate existing modules and applications.

RESEARCH

The objective of this study is to examine and compare the possibilities of using seismic analysis software in Windows Azure public cloud.

Migrating Seismic Analysis Applications to Windows Azure (PAAS)

Migrating WEB applications to Windows Azure (PaaS)

If you have web applications based on Microsoft .NET related technology: follow the steps above to publish these applications to Windows Azure

- Create a new Windows Azure Storage Account and a Hosted Service
- Migrating Data to Windows Azure Drives
- Add a cloud service project in Solution

- Add your app in the cloud service and edit configuration settings.
- Publish the Windows Azure Cloud Service:

Migrating Data to Windows Azure (PaaS)

Windows Azure presents option when you file your sources for spatial data is to use Windows Azure Blob Storage. It is possible to use blob storage to store different file formats – including Tcl files (OpenSees projects).

- Windows Azure Storage Abstractions

Accessing the Windows Azure Storage is done via a storage account. A storage account can have many blob containers (Figure 4).

- Windows Azure Blobs

A container is a user-defined set of blobs that has only properties, which include a list of the blobs it contains. Containers don't store data directly.

The word blob stands for binary large objects. Blobs store binary data, files, and other content as an arbitrary array of bytes within a container Blob URL format is: `http://<Account>.blob.core.windows.net/<Container>/<BlobName>`

There are two types of Blobs:

Block Blob

A Block Blob is a collection of blocks, where each block has a block ID (of 64 bytes). A Block Blob can be up to 200GB. You can upload a maximum of 64MB of data in a Put Blob operation to create a blob. If it is larger than 64MB it must be sent in multiple chunks of no more than 4MB, with Put Block operations.

Page Blob

A Page Blob can be up to 1TB in size and has organized into 512-byte pages within the blob. Any point in the blob can be accessed for read or write operations by using the offset from the start of the blob (Figure 3). Blob storage is accessible via abstractions account; container and blob name. More details about how to use data from blob storage you could find in [3] and [4]. This approach does not work when your Azure PaaS applications contains modules that work with the physical file system. It is possible to use the file system of the VM, used from your Web Role, but this local file system is not persistent.

Virtual Hard Drives (VHDs)

You can use drives, disks, and images in Windows Azure as VHDs stored as page blobs within your storage account (Figure 4.1). You can upload your VHD into your storage account to use it for either PaaS or IaaS. Since drives, disks, and images are all stored in storage, your data will be persisted even when your virtual machine has to be moved to another physical machine.

Azure Drives (PaaS)

Azure Drives are used by the PaaS roles (Worker Role, Web Role, and VM Role) to mount a VHD and assign a drive letter. Drives are implemented with a

kernel mode driver that runs within VM, so the disk IO to and from the drive in the VM will cause network IO to and from the VM to the page blob in Windows Azure Storage.

Using Azure Drives and Web Roles / Worker Roles it is possible to include in Azure solutions based on PaaS modules and whole applications than have no graphical UI and could be launched with commands.

Azure cloud service Implementation

There are two major kinds of the Azure cloud service: Web role and Worker role. Web role is a part of any application which we host on IIS, should be created as web role while hosting it on cloud. Example: WCF, ASP.NET, Silverlight, etc. Web role accepts the requests directly from client and responds back.

Steps to implement Web Role that runs legacy console app (exe file) (Figure 7)

- Including EXE as a part of the project
- Getting EXE path and parameters
- Download project file from Azure blob storage to local storage or Azure drive
- Run EXE in web role (WCF service)

Migrating Seismic Analysis Applications to Windows Azure (IAAS)

Seismic analysis applications, which have graphical UI, can work in Azure VM and maintained via Remote desktop or automation. Applications without graphical UI and those of apps, which have graphical UI, and supports automation can be started also via other applications Web applications and Windows services on VM could call local applications.

You could easily scale your applications using “out of the box” load balancer (Figure 12). Applications for parallel MPI computation with MPICH2 could be set without any changes. In each VM could be installed application and MPICH2. It is important to add a TCP entry point for the specified port for each VM.

Experimental environment

Prerequisites:

- Azure account (VMs with Windows Server 2012 are used)
- VS 2012 and Azure SDK
- OpenSees analysis software.

The (OpenSees) Open System for Earthquake Engineering Simulation is: A software framework for developing sequential, parallel and grid-enabled finite element applications in earthquake engineering. • It was developed primarily in the object-oriented programming language C++. OpenSees supports Parallel Applications in Windows. There are two OpenSees Interpreters for parallel applications:

- OpenSeesSP
- OpenSeesMP

OpenSeesSP is designed to analyze large models on parallel machines. • OpenSeesSP is created with the two goals: minimal changes to input scripts and minimizing the required knowledge of parallel processing

OpenSeesMP is created for performing parameter studies AND for analyzing large models on parallel machines.

To execute the application (OpenSees) in parallel on your VM, build your application and go to the folder where your application executable is. The parallel MPI computation is launched using `mpiexec -n <number of processes> <application executable>`.

Prototypes:

In this research are used three prototypes:

- Web Role (including Silverlight Application that uses Open Sees via WCF Service (PAAS))
- Azure VMs that uses Web application and OpenSees for scalable applications. (Azure supports load balancer out of the box)
- Azure VMs that uses MPIEXEC and OpenSees for parallel tasks (IaaS)

Creating our sample applications for the first and second approaches would involve the following steps:

- Including EXE as a part of the project
- Getting EXE path and parameters
- Uploading project files from local computer to Azure blob storage
- Download project file from Azure blob storage to local storage or Azure drive
- Run EXE in web role (WCF service)

The third approach requires only settings – not development.

Results:

Because of the research are different real workable solutions in the cloud (Windows Azure public cloud):

- PaaS cloud service using OpenSees with cloud drive.
- IaaS VM where is deployed web service (WCF) that use OpenSees (same code is used like in PaaS solution)
- IaaS VM solution with Parallel OpenSees (OpenSeesMP). Solutions with different VMs are compared

Figure 7 demonstrates the architecture of the web application used in the first two approaches. Figure 8 and Figure 10 show how cloud drive is used. There is a comparison between different number of cores / VMs for the third solution (IaaS VM solution with Parallel OpenSees)

The structures used to test the prototypes are in Figures 15 and 17.

Analysis that are used are available in Figures 16 and 18

Table 1 and Table 2 demonstrates time to solve one time history analysis with used model (1 story frame – Figure 17) using different solutions – 1 VM with 1 to 4 cores and 1 to 4 VMs with 1 cores. The charts with these results are available in figures 19 and 20. Results demonstrate that solution is scalable. Comparison between on VM with many cores and many VMs with single core are closer – single VM is a little bit faster because of communication reasons and when are used up to eight cores this solution is a little faster.

The cost is the same for VM and cloud service resources (Figures 13 and 14). Migration of the systems using IaaS (MVs) requires much less development and additional work that PaaS and this solution is more cost effective and easy to implement for now. Comparison between one VM with many cores and many VMs with one core shows closer results. Single VM is a little bit faster but it can support up to eight cores. If you need more compute, you need to use several VMs

CONCLUSION

With Windows Azure, it is possible to migrate applications for seismic analysis in several ways:

- To migrate existing systems with minimal changes build solutions based on IaaS
- To build IaaS solutions where to use existing modules via WEB application
- To build new solutions based on PaaS using existing modules or applications.
- To offer High-Performance Computing (HPC)

The current article covers the first three approaches:

The first approach using Azure IaaS is easier and faster for implementation. Blob storage offer more flexibility and scalability when you have a huge amount of data. Cloud drive is a solution that could let legacy applications and module to work with azure storage without changes of the existing applications

REFERENCES

- V. Subramanian, L. Wang (2010) 'Rapid Processing of Synthetic Seismograms Using Windows Azure Cloud'. Paper presented at the 2nd International Conference on Cloud Computing 2010, Nov 30 - Dec 3, 2010, Indianapolis, USA

- Z. Hill, J. Li, M. Mao, A. Ruiz-Alvarez, and M. Humphrey (2010). Early observations on the performance of windows azure. In the 1st Workshop on Scientific Cloud Computing (ScienceCloud 2010). ACM, 2010.
- Mihail Mateev (2011) Using spatial data in the Windows Azure Blob Storage. Available at <http://bit.ly/hgYnAN> [17 April 2011]
- Mihail Mateev (2012) Using Spatial Data in Windows Azure (SQL azure and azure blob storage), ATINER's Conference Paper Series COM2012-0019. Available at <http://www.atiner.gr/papers/COM2012-0019.pdf>
- Yi Jiao , Yinghui Wang, Shaohua Zhang , Yin Li, Baoming Yang, Lei Yuan (2013), A cloud approach to unified lifecycle data management in architecture, engineering, construction and facilities management: Integrating BIMs and SNS, *Advanced Engineering Informatics* 27 (2013) 173–188
- Ismail Ari, Nitel Muhtaroglu (2012), Design and implementation of a cloud computing service for finite element analysis, *Advances in Engineering Software* (November 7, 2012) Available at <http://www.sciencedirect.com/science/article/pii/S096599781200141X>
- Frank McKenna, Gregory L. Fenves (2007), Using the OpenSeesInterpreter on Parallel Computers (2007-11-04), University of California, Berkeley. Available at http://nees.org/resources/3970/download/Using_the_OpenSees_Interpreter_on_Parallel_Computers.pdf
- Gregory Rodgers (2011), Batch Environment on NEEShub, OpenSeesDays (August 23, 2011). Available at <http://opensees.berkeley.edu/OpenSees/workshops/OpenSeesDays2011/GregNEEShub.pdf>
- David Chappell (March 2009), Introducing Windows Azure, DavidChappell & Associates (2009). Available at http://www.davidchappell.com/writing/white_papers/introducing_windows_azure_v1-chappell.pdf
- Mihail Mateev (2012) Opportunities for Automating Seismic Analysis Using the Windows Azure Cloud, *International Jubilee Conference UACEG2012: Science & Practice* (2012)
- Van T. K. Tran, Kevin Lee, Alan Fekete, Anna Liu and Jacky Keung (2011), 2011 Int'l Symp. Empirical Software Engineering and Measurement, pp. 265–274
- Vitor Ciaramella (2013), *Advanced Cloud Services Development (PaaS)* , Azure Summit Brazil (2013)

Figure 1. *Cloud Computing Service Levels. Cloud computing into three distinct classes of service.*

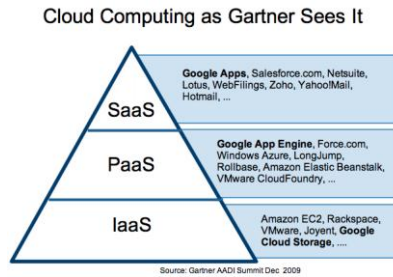


Figure 2. *Cloud services according to the purpose*

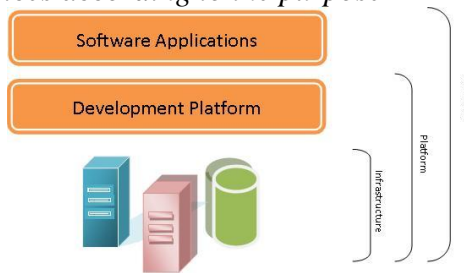


Figure 3. *Multiple-instances Windows Azure application*

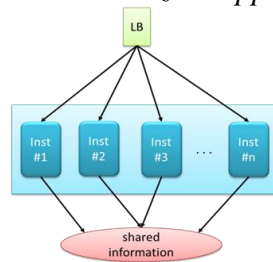


Figure 4. *Windows Azure Blob Storage Architecture*

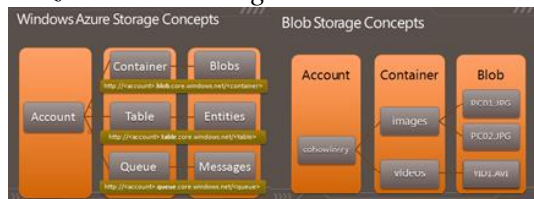


Figure 4.1 *Windows Azure Drives*

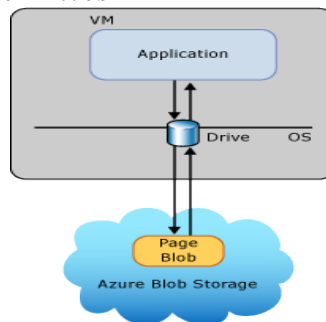


Figure 5. *NEES Cyberinfrastructure. NEES Cyber infrastructure by Gregory P Rodgers, Rajesh Thyagarajan. Available at <http://nees.org/topics/NEESCyberinfrastructure>*

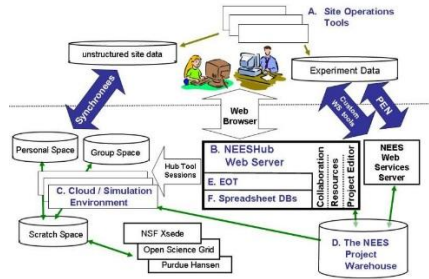


Figure 6. *NEES Project Directory Structure. The image maintained by Stanislav Pejisa, Purdue University. Available at <http://nees.org/topics/NEESProjectDirectoryStructure>*

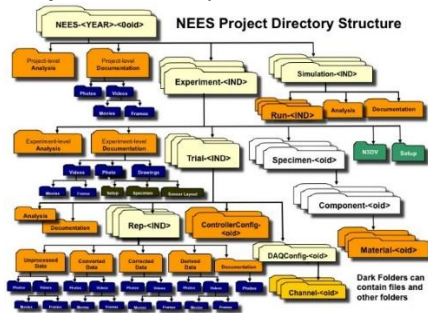


Figure 7. *Ruining OpenSees in Azure (Web Role & WCF Service). Block Diagram*

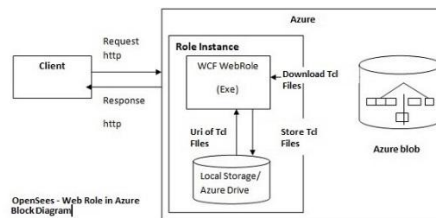


Figure 8. *Drives used by the PaaS roles. PaaS roles (Worker Role, Web Role, and VM Role) to mount a VHD and assign a drive letter*

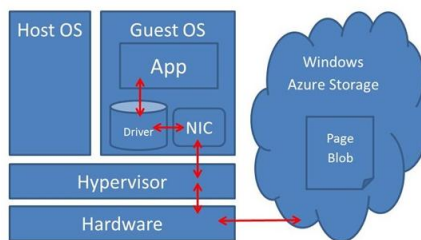


Figure 9. Scalable Azure Web Application - David Chappell [9]

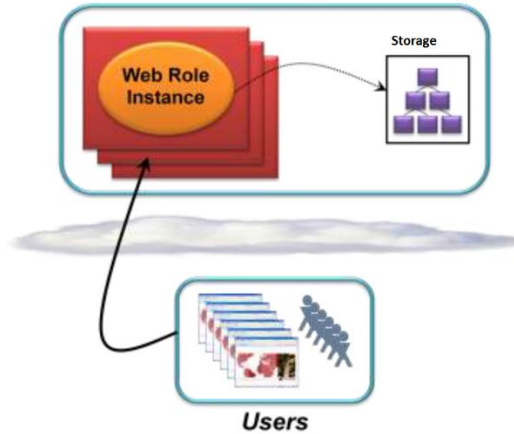


Figure 10. Mount the VHD as local drive in each instance. Only one instance can mount it with Read/Write access

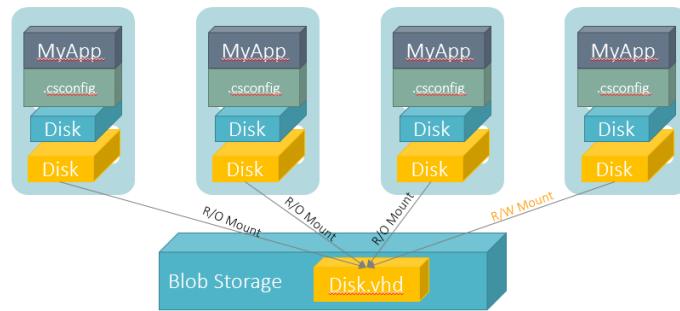


Figure 11. Windows Azure HPC Application

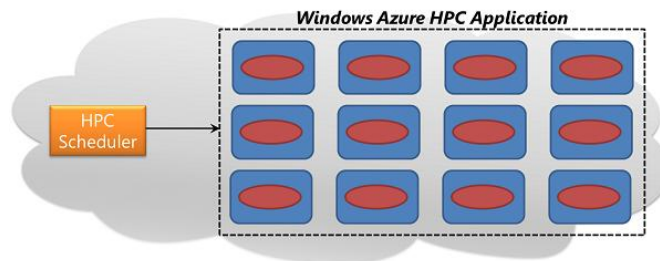


Figure 12. VM instances hosted on Windows Azure will be load-balanced automatically

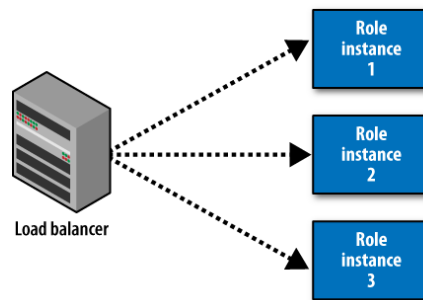


Figure 13. Azure VM pricing details

COMPUTE INSTANCE NAME	VIRTUAL CORES	RAM	PRICE PER HOUR (GA PRICE STARTING JUN 1ST)	PRICE PER HOUR (PROMOTIONAL PRICE UNTIL MAY 31ST)
Extra Small (A0)	Shared	768 MB	\$0.02 (~\$15/month)	\$0.0133 (~\$10/month)
Small (A1)	1	1.75 GB	\$0.09 (~\$67/month)	\$0.08 (~\$60/month)
Medium (A2)	2	3.5 GB	\$0.18 (~\$134/month)	\$0.16 (~\$119/month)
Large (A3)	4	7 GB	\$0.36 (~\$268/month)	\$0.32 (~\$238/month)
Extra Large (A4)	8	14 GB	\$0.72 (~\$536/month)	\$0.64 (~\$476/month)

Figure 14. Azure Cloud Services pricing details

NAME	VIRTUAL CORES	RAM	PRICE PER HOUR
Extra Small (A0)	Shared	768 MB	\$0.02 (~\$15/month)
Small (A1)	1	1.75 GB	\$0.08 (~\$60/month)
Medium (A2)	2	3.5 GB	\$0.16 (~\$119/month)
Large (A3)	4	7 GB	\$0.32 (~\$238/month)
Extra Large (A4)	8	14 GB	\$0.64 (~\$476/month)

Figure 15. Restrained beam under thermal expansion. Simple project to demonstrate FEM Analysis

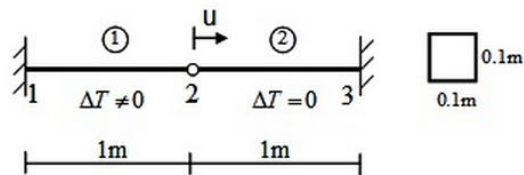


Figure 16. Restrained beam displacement results

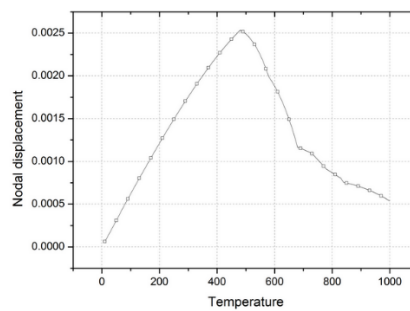


Figure 17. Seismic analysis of the reinforced concrete frame. A reinforced concrete portal frame - uniform earthquake excitation

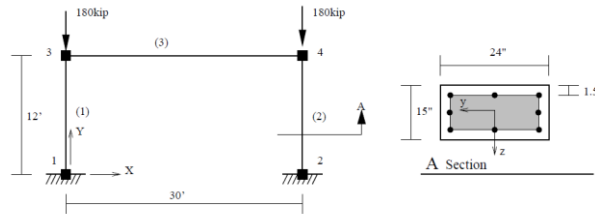


Figure 18. Lateral Displacement at node 3 (Seismic analysis)

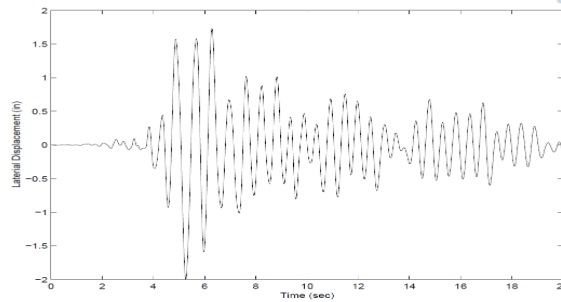


Figure 19. Results: Execution time using up to 4 extra small VMs

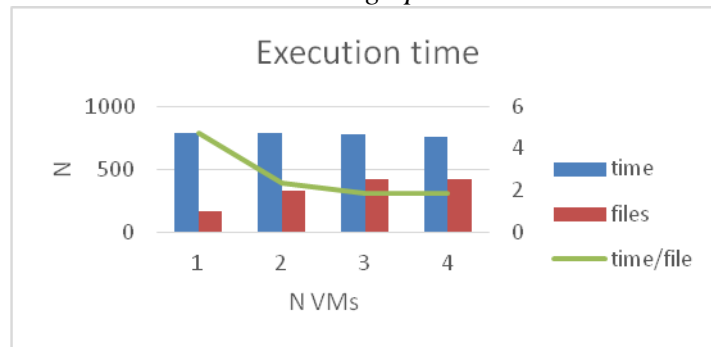


Figure 20. Results: Execution time using one large VM

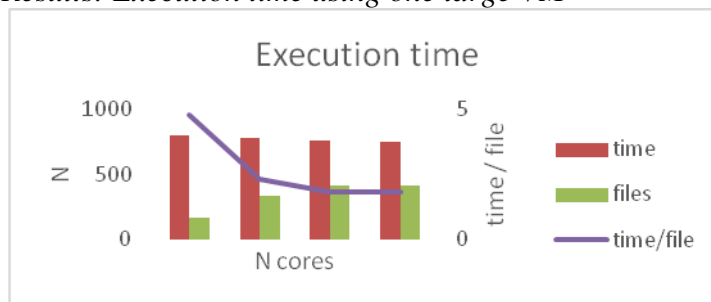


Table 1. Execution time (up to 4 extra small VMs)

N VMs	1	2	3	4
Time (sec)	792	785	775	761
Files	167	333	416	416
time/file (sec)	4.742515	2.357357	1.862981	1.829327

Table 2. *Execution time (1 large VMs)*

N Cores	1	2	3	4
time	792	772	755	744
files	167	333	416	416
time/file	4.742515	2.318318	1.814904	1.788462