Introducing a new cloud computation paradigm for rock engineering problems based on XaaS Model and the proposed A4 Framework

Ali Bonakdar
Individual private geomechanical advisor, Tehran, Iran

ABSTRACT: Nowadays, High Performance Computing (HPC) is being used as one of the most interesting topics in both aspects of scientific and practical, relies on recent special and exquisite developments in software architectures and frameworks and also hardware improvements in distributed networks over the Internet. As a geo-engineering problem to be solved, there are some limitations against today single-node commercial applications, containing runtime, analysis costs, scale, and accessibility. In this research, a new paradigm has been proposed base on a new developed framework (A4) and XaaS model. Using the new paradigm, significant modifications will be applied in computation runtime, decrease in computation costs, high software accessibility, in addition to scalability and diversity of problem solving.
An online web-based 3D visualization platform (containing pre-processing and post-processing of numerical modeling) has been implemented to remove the limitations of the available single-node conventional applications to enable them running on a simple Internet browser with an affordable cost.

Keywords: Cloud Computation, High Performance Computing (HPC), Distributed Computation, Parallel Numerical Analysis, Computation Farm, Web 3D Visualization.

1 INTRODUCTION

Numerical analysis of the engineering problems, in most cases, contains either a domain with large geometry dimensions (with coarse mesh grids) or a small domain (with ultra-fine mesh grids) to obtain a fast or precise results, respectively.

This issue can be interpreted using a parallel computation over the Internet network (which is referred to as computation on the air) and will be one of the requirements for such large-scale (or refine-gridded) geo-mechanical engineering problems over the next few years. Parallel computation provides the engineers with the possibility to distribute the calculations load over many other available running systems on the Internet (called cloud computation farm in this paper).

Within the Cloud Computation domain, there are various terms for specific pieces of the cloud. Some of these terms include the different variations of X in "XaaS" - meaning "something" as a service; the X is a variable that can be replaced by specific entities to represent several abilities for scientists. This term (i.e. XaaS) simply describes the distribution of different ICT components within a Cloud Computation service model, and denotes everything as a service.

Moreover, in numerous cases, researchers and engineers may need only to run a simple model or run a complicated one, just for an specific project (or case study), so they will not prefer to buy a full-priced pack-age of a numerical computer application like commercial ones (such as Itasca codes, Rocscience packages, Geoslope software, PLAXIS, and more). In these cases, a SaaS (Software as a Service) architecture would be the most helpful and useful tool for Anyone in Anytime, in Anywhere with Any device (A4 framework), for handling the problem in a rapid and affordable approach.

In this research, a set of web-based packages have been designed, programmed, and implemented, that will be applied in a sequential steps, to complete an online process for covering a fast, robust, and distributed modeling and simulation in the proposed A4 framework.

2 THE BACKGROUND AND LITERATURE REVIEW

Many efforts have been made to implement online tools in the past few years by different researchers, to make desktop applications accessible over the web. Ashutosh Sharma et. al., developed a web interface, targeting the learning community, by providing an efficient web interface for MATLAB by coalescing the two major concepts of cloud computing and open source, by making use of the concept of cloud compiler. In their work, they implemented a website as an interface between the user and the MATLAB compiler (which was hosted on a remote server). The website was responsible to pass the user's command to the server and return back the results in some 2D or 3D graphics.

The Weblinux tool, the other case in this session, provides a Linux OS running entirely in the browser and solves the installation problem for the learners, especially early beginners.

Many commercial companies have spent time and cost to develop platforms to fulfill such capabilities to make regular applications accessible over the web. The most common and well-known Infrastructures as a Service (IaaS) for HPC purposes could be fit in the following list: Google Compute Engine, Azure Virtual Machines, Rescale, G2 Deals, Amazon EC2, IBM Cloud, OpenStack, Linode, Digital Ocean.

3 DESIGN AND ANALYSIS

Casspel & Balltazar, two designed Desktop & Web applications, in collaboration with each other, provide an environment to transform existing single-node geomechanical applications into a distributed system quickly and easily (with SaaS, DaaS, and XaaS models), without any significant concern relating to re-architecting entire of the available applications. Since adapting to a web platform requires some modifications to establish internal communication between web nodes, the target application (i.e. the application supposed to solve the geomechanical problem like FLAC, called FLAC as a Service or FaaS) should be changed from a single-node desktop application to a distributed web service that can serve per request of the users.

With the proposed paradigm, any geomechanical problem could be solved not only by the available commercial codes (such as FLAC, UDEC, PFC), but also any user-defined functions (from simple numerical functions to complicated artificial intelligence (AI) ones or image processing and computer vision (CV) libraries) can be used remotely from the air (i.e. the Internet) from any active device. Also, a computation farm could be raised just only with some active home desktop computers, without any intermediate inter-face.

The principles behind the proposed scenario, is Microservices Architecture on a web platform (using RESTful API) with many computation nodes with DaaS (Desktop as a Service) architecture, totally called a computation farm, to perform a real-time and distributed simulation runs, as per user requests, following cost minimization and rapid computation for a wide range of geomechanical practical problems. Communicating between Numerical Application Servers (such as Itasca FLAC engine) and CASSPEL Server will be performed through RESTful API and Web Services.

Representational State Transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the Internet (Chandrasekaran 2015). A RESTful API is an application programing interface (API) that uses HTTP requests to handle communicating between the Server (CASSPEL Engine) and the Client (end-user or the company who provides numerical calculation for solving the problem).

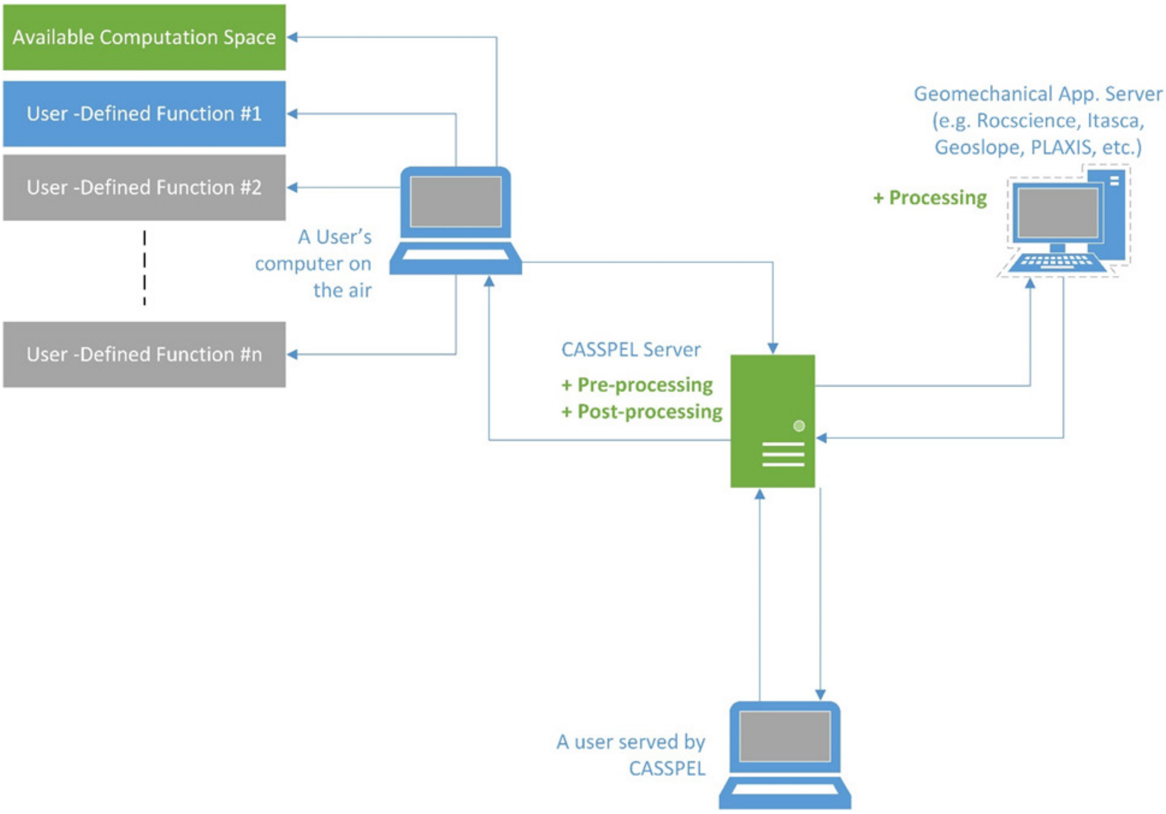

Figure 1. A general overview of communicating nodes in the proposed framework.

RESTful APIs will used for third parties (i.e. the companies like Rocscience, PLAXIS, etc.) for communicating with the CASSPEL Engine for generating desired responsive and interactive output to be illustrated in the end-user computer. RESTful API also will be sued for programmers who intend to develop new numerical applications, that will use an outsource solver engine like FLAC Web Engine.

The XaaS model, the idea has been exploited from, describes here simply that every small code (defined by any researcher located anywhere in the world) can be called from other researchers to be used in a numerical analysis run (powered by any company located anywhere in the world). By this scenario, no any researcher needs to collect everything before a numerical run conducted (such as purchasing a new full pack-age or learning a new numerical software), in addition to merging various applications would be realized. Also, a wide variety of free of use functions developed by other researchers (for constitutive models using for rock and soil and support elements, for generating rough or fine meshes, for automatic material allocation to rock blocks or joint blocks, user-defined voroni geometry, and more) will be available to the researchers.

Moreover, a new 3D Engine has been created for post-processing of a back-end simulation results that will be used to render simulation results for the end-user in an interactive environment (with full and comprehensive adaptability for any device, such as desktop computers or tablet and smart phone devices, in front-end). In addition, a comprehensive built-in RESTful API is considered for developers to get services from the core engine of the proposed system, regarding academic development purposes or commercial uses.

The embedded 3D-Engine library used in the system, contains both low-level computation mathematics (for matrices, rotation, transformation, reflection, rendering, etc.) and high-level functions for illustrating support elements (rock bolt, pretension cable, shotcrete and full lining (beam), pile, etc.), groundwater, geomaterial, boundary conditions, annotations and more. Also multiple graphical functions are considered for general purposes such as gradient fill, perspective toolbox, viewing and zooming, and printing.

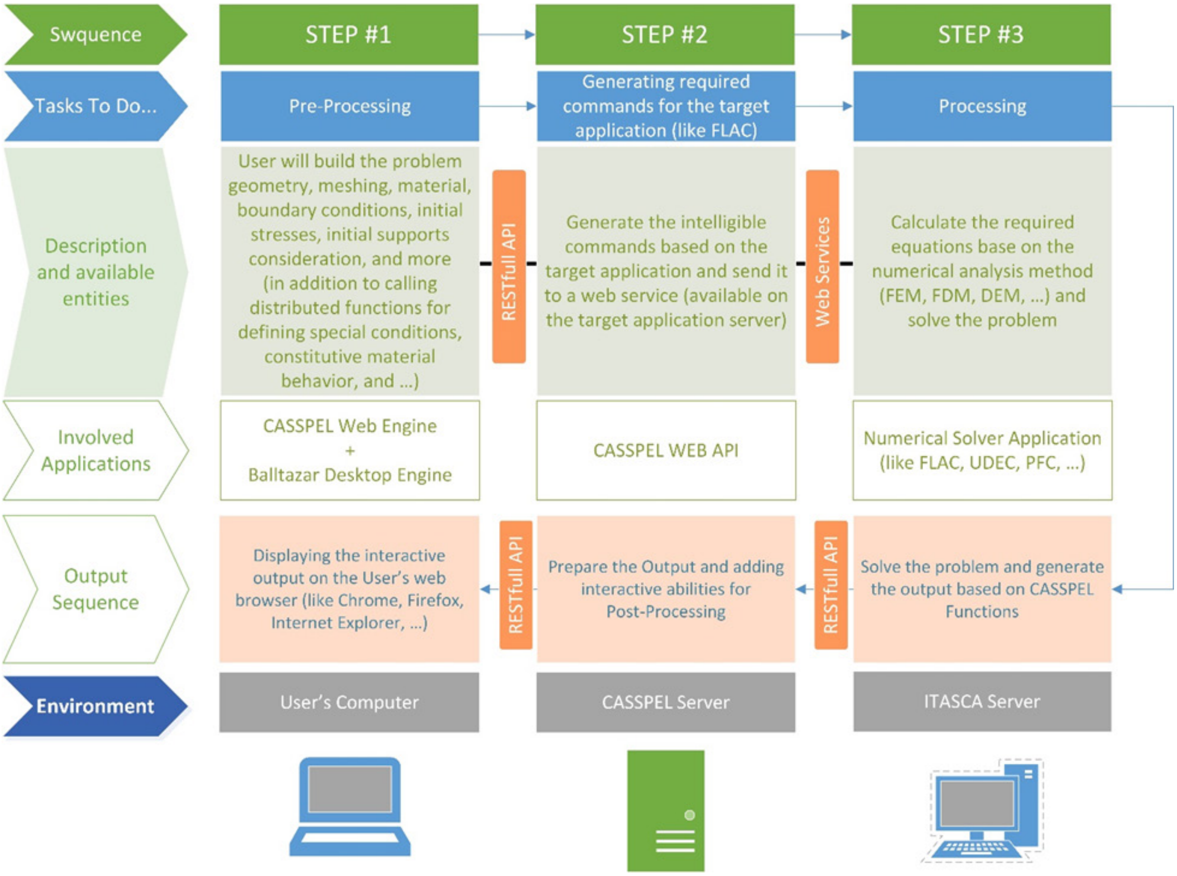

Figure 2. Sequential diagram of Data-Flow with a full description of each node (from start point of designing the geomechanical problem geometry to receiving the simulated result on the end-user computer) for a hypothetical model that supposed to be simulated by Itasca codes (like FLAC).

As illustrated in Figure 2, a simple paradigm will handle the problem solving (in the back-end of processing part) and will generate the results output for viewing in the end-user's device in the front-end of post-processing. This procedure could be repeated for every modifications occurred in the model and requesting new simulation run. The user will just pay the simulation cost, as much as the FLAC resource are consumed. In fact, simulation runtime and other consumed server resources will determine how much should be paid per request.

4 RESULTS AND DISCUSSION

Cloud computing is one of the most popular technologies that has become an integral part of the computing world nowadays. The usage and popularity of cloud computing is increasing every day and is expected to be increased further. Many frequent Internet users are heavily dependent on cloud-based applications for their day-to-day activities in both professional and personal life. Cloud computing has emerged as a technology to realize the utility model of computing while using Internet for accessing applications.

One of the most significant aspects of implementing an online engineering system, is serving as request and payment. The past 5 years has seen the rapid development of SaaS in online general services (for public usage in addition to specialized fields), called cloud computing platform.

As discussed in this paper, merging all available geomechanical single-node applications into an integrated environment, to take advantages of the ability of all existing tools, is one of the goals following up by implementing the proposed system. Everything as a Service, i.e. XaaS, makes it possible to collect all available features needed for running a large-scale numerical simulation model, which will led to a rapid and cost-optimized numerical analysis.

5 CONCLUSION

Although such an online simulation using some available platforms, like Rescale, has been already realized with virtualization or any other methods, but these cases only provide the users with running a real program that is installed on a remote server without accessing to anymore distributed functionalities, that were proposed in this research work.

The proposed framework will make it possible for instance for FLAC to be FaaS (FLAC as a Service) or Geoslope to be GaaS, providing any numerical model to be solved with the computational engine developed by the own companies, for Anyone in Anytime, in Anywhere with Any device (A4), with a low-cost and quick generating of the model. Although it is still in development, it's expected that CASSPEL could be a standard pioneer to encapsulate back-end solver for researchers and front-end interface for commercial companies who are involved in producing numerical engineering applications, on the cloud, the Internet.

It's not only a utility, but a distributed framework which can be extended fully by developers and researchers all over the world, without any limitations. Thanks to the RESTful API incorporated in the proposed framework, development of the third-parties entities (including procedures or functions, libraries, and packages) could be implemented by any popular programming languages such as Python, C#, Java, PHP, and even JavaScript (in local host or on the server in Node.js runtime environment).

REFERENCES

http://www.orecad.com/casspel/usermanual

http://www.orecad.com/balltazar/usermanual

Chandrasekaran, K. 2015. Essentials of Cloud Computing. *CRC Press*, Florida, USA.

ITASCA Consulting Group. 2017. FLAC3D User's Manual. Minneapolis, USA.

Kurgalin, S. & Borzunov, S. 2020. A Practical Approach to High-Performance Computing. *Springer*, pp. 68-95.

Sharma, P. & Sharma, M. & Elhoseny, M. 2020. Applications of Cloud Computing. *CRC Press*, pp. 81-116.

Sharma A. & Singh P. & Khatri A. 2015. Web Interface for MATLAB. *International Journal of Advances in Engineering Sciences Vol.5, Issue 1*, pp. 12-16.

Sharrock R. & Angrave L. & Hamonic E. 2018. Weblinux: a scalable in-browser and client-side Linux and IDE. *Complex systems simulation and modelling Conference*, France.

Glander T. 2019. ReWeb3D - Enabling Desktop 3D Applications to Run in the Web. *Proceedings of the 18th International Conference on 3D Web Technology*, San Sebastian, Spain.