NUCLEAR THEORY, Vol. 32 (2013)
eds. A.I. Georgieva, N. Minkov, Heron Press, Sofia

The Cloud Computational Environment – a Blueprint for Applications in Nuclear Structure Physics

S. Mishev

Joint Institute for Nuclear Research, Dubna, 141980 Russia

Abstract. The utility of the cloud computational model for studies in the field of the nuclear structure theory is addressed. In particular, a class of theoretical many-body approaches which could benefit from this technology is delineated. An architecture suitable for dealing with high performance computations for nuclear structure theories in a cloud is outlined.

Alongside that, a nuclear theory aggregation software platform for presenting reports on calculations from various models is discussed

1 Motivation

The massive adoption of the cloud model [1] in the IT industry has reflected the way new high performance (HP) computing facilities designed for solving problems in different fields of science are built. Although the idea of cloud computing emerged more than half a century ago [2], it was the shift to the pay-per-use computing services (as opposed to organization owned assets) in the recent decade that unleashed its numerous realizations. Performing computations on remote resources have been a commonplace for scientists since the invention of computers - supercomputers and computer clusters being the most widely used options. From the end user point of view, the two characteristics of the cloud model which distinguish it from the conventional high performance facilities are its “elasticity” and self-service [1]. The provided option to the end users to manage the topology and the power of the processing units as well as storage makes the developers in the cloud environment largely independent of the infrastructure provider thus lowering the barrier to entering into the high performance computing sector. This extra degree of freedom coupled with the virtually unlimited resources of the cloud constitutes the appeal of this platform. Additionally the costs of adding new resources increase linearly with demand (“scalability”) and moreover these resources can be added or removed in real time (“elasticity”).

In order to achieve scalability, a software running in a cloud infrastructure must take advantage of its platform virtualization as well as the object and block storage capabilities. The potential for utilizing the cloud model for scientific research, have been envisaged lately by CERN's IT department [3] where a cloud infrastructure is build on the basis of the Openstack cloud computing platform [4]. Openstack itself is a descendant of the NASA's Nebula [5] platform for managing the virtual infrastructure of data centers.

Despite being one of the trending technologies in computing, in the field of many-body physics, the opportunities provided by the cloud computational model seem to be largely neglected to date. One of the its rare uses is realized in modeling of biomolecules using the molecular dynamics techniques [6].

The purpose of this note is to sketch some use cases for cloud computations which could possibly be beneficial for the nuclear structure theorists who build software.

2 Architectural Remarks

The process of building a high performance cluster in the cloud involves the utilization of high performance virtual computers (referred to as instances) using enough block storage and connected in a fast network. The computational tasks are distributed over the instances using a job scheduler. Specially designed toolkits for automating the process of building and managing the cluster, as for example StarCluster [7] for managing instances from the Amazons Elastic Compute Cloud (EC2) [8], allow to deploy numerical libraries to the instances thus saving much effort to researchers. Another approach to building an ensemble of working nodes in a cloud is to use its inherent capabilities of cloning instances. To this end one first creates a single node with all the required libraries installed and then use it as a template (referred to as "snapshot") for launching multiple identical copy of it.

Let us try to outline an architecture suitable for dealing with the task of calculating the wave functions and energies of certain states belonging to a set of (potentially all) nuclei solving the equations derived from a nuclear model as well as subsequently performing operations on the obtained states. An essential ingredient in assembling a proper HP system for a certain nuclear model is the choice of instances' type. For example, since the ab initio shell model calculations require the diagonalization of huge matrices, memory optimized instances must be considered, whereas for the models based on the nuclear mean field or the density functional theory compute optimized instances must be prefer [1]. The judgement on the instances' performance should also be based on the consideration that heavy open-shell nuclei prove to be consuming much more computational resources than the calculations for the light ones near the magic numbers.

The results of the calculations are usually stored in voluminous files on the associated block storage. In order to extract the physical content, the accumulated data, which can be accessed after the termination of the instances, has to

[1] We used the instances categorization terminology used in EC2 (`http://aws.amazon.com/ec2/instance-types/instance-details/`)

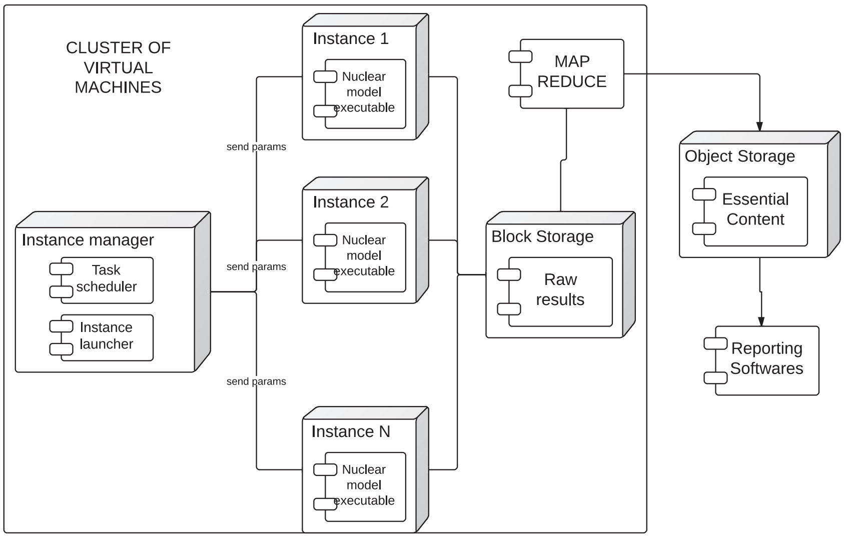

Figure 1. A draft architecture representing basic components of a HP system, capable of serving the demands of nuclear theorists.

undergo further processing, the results of which would serve as input for report generation. A vastly popular model for handling such problems in the cloud is the MapReduce approach, with Apache Hadoop [9] being one of its most widely used implementation to data processing. Another native feature of the cloud environment relevant to storing the results from the postprocessing phase is the its object storage. The advantages of using this kind of storage rather than block storage is that the former provides higher scalability and distributed access which would be beneficial in cases where this data is analyzed by different research groups. A visual representation of the processes described in the last two paragraphs is given in Figure 1.

The final purpose of this note is give a brief description of a platform, named `NuclearModels.net` [10], which aims at aggregating the reports based on the calculations for nuclei from the whole chart of isotopes. It addresses the need of nuclear theorists to share more detailed reports on their calculations than the ones presented in journal articles as well as compare them with other researchers' results. Moreover, given that detailed reports from calculations based on different nuclear models are collected in one place, this platform might facilitate the experimentalists in interpreting their data from different angles. The reports themselves are located at remote locations and are accessed by the system over the HTTP protocol. An URL[1] for a report, along with other attributes, can be assigned to a given nucleus using the system's administrative panel. A

[1] Uniform Resource Locator

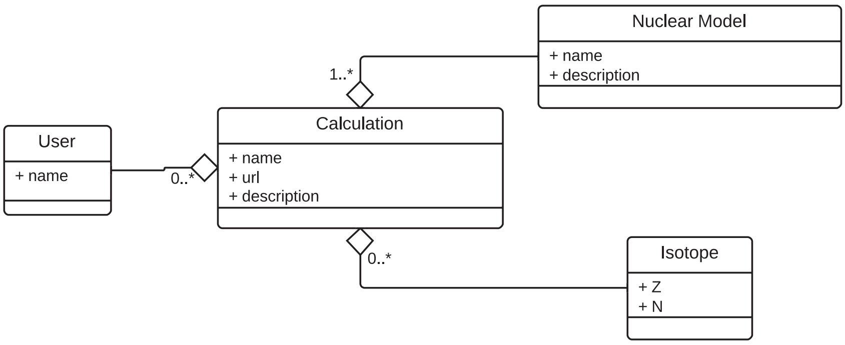

Figure 2. A simplified UML class diagram depicting the relationships between the main object in `NuclearModels.net`. See text for details.

very basic entity relationship diagram for the platform's main objects is given in Figure 2. The option for remote software components to access the reports from the calculations is implemented via a RESTful [11] API[2], which is to be described elsewhere.

3 Conclusion

The cloud model opens new vistas to researchers from all fields of science. In this note a short introduction to this computational model with a focus on nuclear structure is presented. The draft architecture in Figure 1 is only one avenue to follow and it is given to open a discussion on this subject in the community.

Software products helping scientists perform calculations in the cloud seem to be at an early stage of their development. A step in this direction, along with the aforementioned StarCluster toolkit, is a project, currently in development, for automating the process described in Section 2 using the Openstack API.

Acknowledgements

The author is thankful to “Altscale Cloud Services” [12] for kindly providing access to resources of their Openstack cloud infrastructure.

This work was partly supported by the Bulgarian National Science Fund, under contract DID-02/16.

References

[1] Peter Mell and Timothy Grance, *The NIST Definition of Cloud Computing, NIST Special Publication 800-145* (September 2011, National Institute of Standards and Technology, U.S. Department of Commerce.

[2] Application Programming Interface

[2] John McCarthy, speaking at the MIT Centennial in 1961, “Architects of the Information Society, Thirty-Five Years of the Laboratory for Computer Science at MIT”, Edited by Hal Abelson.

[3] P. Andrade *et al.*, *J. Phys.: Conf. Ser.* **396** (2012) 042002.

[4] The Openstack Cloud Operating System (`http://www.openstack.org/`)

[5] Nebula Cloud Computing Platform (`http://nebula.nasa.gov/`)

[6] NAMD code (`http://www.ks.uiuc.edu/Research/namd/`)

[7] StarCluster (`http://star.mit.edu/cluster/docs/latest/index.html`)

[8] Amazon Elastic Compute Cloud (`http://aws.amazon.com/ec2/`)

[9] Apache Hadoop `http://hadoop.apache.org/`

[10] `http://www.nuclearmodels.net`

[11] R.T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, PhD dissertation (University of California, Irvine, 2000).

[12] `http://www.altscale.com`