An HPC Application Deployment Model on Azure Cloud for SMEs

Fan Ding[1,2], Dieter an Mey[2], Sandra Wienke[2], Ruisheng Zhang[3] and Lian Li[1]
[1]*School of Mathematics and Statistics, Lanzhou University, Lanzhou, China*
[2]*Center for Computing and Communication, RWTH Aachen University, Aachen, Germany*
[3]*School of Information Science & Engineering, Lanzhou University, Lanzhou, China*

Keywords: Azure Cloud, MPI, HPC, Azure HPC Scheduler, SMEs.

Abstract: With the advance of high-performance computing (HPC), more and more scientific applications which cannot be satisfied by on-premises compute power need large scale of computing resources, especially for small and medium-sized enterprises (SMEs). Emerging cloud computing offerings promise to provide us with enormous on-demand computing power. Many cloud platforms have been developed to provide users with various kinds of computer and storage resources. The user only needs to pay for the required resources and does not need to struggle with the underlying configuration of the operation system. But it is not always convenient for a user to migrate on-premises applications to these cloud platforms, which is especially true for an HPC application. In this paper, we proposed an HPC application deployment model based on the Windows Azure cloud platform, and developed an MPI application case on Azure.

1 INTRODUCTION

High-performance computing (HPC) dedicates big processing power to compute-intensive complex applications for scientific research, engineering and academic subjects. From the mainframe era to clusters and then grid, more and more available computing resources can be employed by HPC applications on-premises. Nowadays we have come to the times of cloud computing. There are some differences to provide users with the resources between on-premises and cloud. On-premises require users to invest much cost in purchasing equipment and software for building their infrastructures at their initial development. This may be a challenge for some SMEs (small and medium-sized businesses) since they have not enough capability to invest. With cloud computing, users can obtain on-demand resources from cloud virtual servers not only for the basic infrastructure but also for the extra resources which used to process complex work cannot be finished by the enterprise/researcher competence of its own. Moreover, the massive data centers in the cloud can meet the requirement of data-intensive applications. By means of these advantages of the cloud paradigm, it will be an inevitable trend to migrate the HPC applications into the cloud.

Nowadays, there are many cloud service platforms provided by different cloud vendors. The major cloud platforms include Amazon's Elastic Compute Cloud (EC2) (see Amazon), IBM SmartCloud (see IBM), Google Apps (see Google Apps) and Microsoft's Azure cloud (see Windows Azure). Users can choose the different kind of cloud platform according to which level of cloud service they need. Generally, they provide three types of cloud services, SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service), also called service models. These platforms with their support for HPC have been summarized in Table 1.

From the point of view of a user, who is not familiar with complex computer configuration, it is difficult to migrate existing on-premises applications into the cloud because of the differences and the complex configuration of the user interface in these cloud platforms. Some methods or middleware which enable users to use cloud platform easily are required by scientists and other potential users of cloud resources. Much work has been done to study how to take an existing application into the cloud. The work presented in paper (Costaa and Cruzb,2012) is similar to ours, but the authors focused on moving a web application to Azure. CloudSNAP also was a web deployment platform to

Ding F., an Mey D., Wienke S., Zhang R. and Li L..
An HPC Application Deployment Model on Azure Cloud for SMEs.
DOI: 10.5220/0004412202530259
In *Proceedings of the 3rd International Conference on Cloud Computing and Services Science* (CLOSER-2013), pages 253-259
ISBN: 978-989-8565-52-5

deploy Java EE web applications into a Cloud infrastructure (Mondéjar, Pedro, Carles, Lluis, 2013). In paper (Marozzo, Lordan, Rafanell, Lezzi and Badia, 2012), the authors developed a framework to execute an e-Science application on Azure platform through expending COMPSs programming framework. But all of these aforementioned efforts did not consider parallel computing for the compute-intensive applications in HPC.

Table 1: Major current cloud platforms.

Service vendor	Cloud Platform	Description	Support for HPC
Amazon	EC2 (IaaS)	Provides scalable web services, enables users to change capacity quickly.	Cluster Compute and Cluster GPU Instances
IBM	IBM SmartCloud (IaaS)	Allows users to operate and manage virtual machines and data storage according to their need	IBM HPC clouds
Microsoft	Windows Azure (PaaS)	Composed of Windows Azure, SQL Azure and Azure AppFabric	Azure HPC Scheduler
Google	Google Apps (SaaS)	Provisions web-based applications (Gmail, Google Talk and Google Calendar) and file storage (Google Docs)	Google Compute Engine (see hpccloud)

In our paper, we take the advantage of Azure to develop a cloud deployment model through expending the Azure HPC scheduler (see HPC scheduler). Windows Azure is an open cloud platform developed by Microsoft on which users can build, deploy and manage applications across a global network of the Microsoft datacenter. Our work is based on the project 'Technical Cloud Computing for SMEs in Manufacturing'. In this project, the application "ZaKo3D" (Brecher, Gorgels, Kauffmann, Röthlingshöfer, Flodin and Henser, 2010) developed by WZL (the Institute for Machine Tools and Production Engineering, RWTH Aachen University) (see RWTH WZL), which aims to do FE-based 3D Tooth Contact Analysis, is a high-performance technical computing software tool based on simulation of the tooth contact. It reads several geometry data of the flank and a FE-Model of a gear section as the software's variation and then performs a set complex of variation computations. As a result of the variation computations, one gets contact distance, loads and deflections on the tooth. Such a variation computation includes e.g. thousands of variants to be processed which lead to computing times of around months on a single desktop PC. Currently, this challenge is addressed by employing on-premises

HPC resources are available at RWTH Aachen University, e.g. if one variant takes 1 hour computing time, the entire variant time for 5000 variations would take 5000 hours computing time. This exceeds by far the capabilities even of multi-socket multi-core workstations, but can well be performed in parallel by an HPC cluster. But the small and medium-sized businesses in general can neither access those, nor do they maintain similar capabilities themselves. As described earlier, the availability of HPC resources in the cloud with a pay-on-demand model may significantly change this picture.

In this paper, we also developed a case which deploys the "ZaKo3D" application on Azure according to our deployment model. This use case convinces SMEs to adopt cloud computing to address this computational challenge. Moreover, this application is a serial version, as outlined before, to execute a large scale of variation calculation for a long time. It is a challenge for us to design an optimal method to improve the computing efficiency. We have figured out a framework to deploy the application on Azure which parallelizes the variation computation and run the application on Azure by using our deployment model.

The rest of the paper is structured as follows. In Section 2, we introduce the Azure cloud and its module Azure HPC Scheduler. Section 3 describes our problem statement. Followed by our deployment model and framework for parallel execution of the Zako3D application on Azure presented in Section 4. In Section 5, we compare the runtime between on-premises and cloud for HPC by deploying our application on the RWTH Cluster and Azure. This comparison can be used as a reference point for potential users to consider whether employ the on-demand resources. Finally, Section 6 concludes the paper with a summary and future work.

2 WINDOWS AZURE AND HPC SCHEDULER

In this section we introduce the Azure cloud platform and related technologies that we have used in the development of the deployment model architecture. The Windows Azure platform was announced by our collaborative partner Microsoft in 2010. In our work, they provided us the Azure cloud accounts for initial development and testing. This platform includes Windows Azure, SQL Azure and AppFabric. Our work focuses on Windows Azure, which is a Platform as a Service offering and provides us the compute resources and scalable storage services. We employ Windows Azure HPC Scheduler to deploy the serial ZaKo3D by means of an MPI-based framework developed by us, to the Azure compute resources. For data management, the Azure storage blob service facilitates the transfer and storage of massive data in the execution of our MPI application on the Azure cloud.

2.1 The Three Roles of Windows Azure

Windows Azure provides the user with three types of roles to develop a cloud application: Web Roles, Work Roles and VM Roles. Web Roles aim to display websites and present web applications, supported by IIS. Work Roles are used to execute tasks which require the compute resources. Work roles can communicate with other roles by means of Message queues as a choice in various techniques. The VM Role differs from the other two roles in that it acts as an IaaS to provide services instead of PaaS. The VM Role allows us to run a customized instance of Windows Server 2008 R2 in Windows Azure. It facilities migrate with some application, which is difficult to bring to cloud, into the cloud.

2.2 HPC Scheduler for High-performance Computing in Azure Cloud

Microsoft developed the HPC Scheduler to support running HPC applications in the Azure cloud. Compute resources are virtualized as instances on Windows Azure. When an HPC application requires an Azure instance to execute, it means the work will be divided into lots of small work items, all running in parallel on many virtual machines simultaneously. The HPC Scheduler allows scheduling this kind of applications built to use the Message Passing Interface (MPI) and distributes their works across some instances. The deployment build with Windows Azure SDK includes a job scheduling module and a web portal to submit job and resource management.

The role types and the service topology can be defined when creating a service model in configuring cloud hosting service. HPC Scheduler supports Windows Azure roles through offering plug-ins. There are three types of nodes which provide different function and runtime support.

- Head node: Windows Azure work role with HpcHeadNode plug-in, provides job scheduling and resource management functionality.
- Compute node: Windows Azure work role with HpcComputeNode plug-in, provides runtime support for MPI and SOA.
- Front node: Windows Azure web role with HpcFrontEnd plug-in, provides web portal (based on REST) as the job submission interface for HPC Scheduler.

Visual studio has been specified as the development environment for this component.

2.3 Azure Storage Blob Service

Azure storage service provides data storage and transfer to applications in Windows Azure and supports multiple types of data: binary, text data, message and structured data. It includes three types of storage service:

- Blobs (Binary Large Objects), the simplest way for storing binary and text data (up to 1TB for each file) that can be accessed from anywhere in the world via HTTP or HTTPS.
- Tables, for storing non-relational data using structured storage method.
- Queues, for storing messages that may be accessed by a client, and communicate messages between two roles asynchronously.

In our deployment model, we employ the Blob storage service to manage our application's data because ZaKo3D uses text data for input files and output files. As described in Figure 1, there are three layers in the concept of Blob storage. In order to store data into the Windows Azure with the blob storage service, a blob storage account has to be created which can contain multiple containers. A container looks like a folder in which we place our blob items.

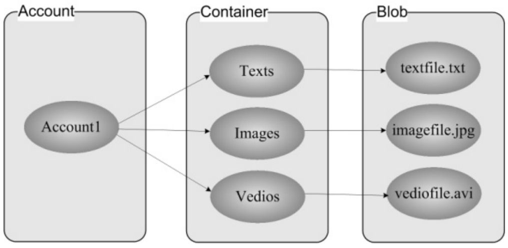

Figure 1: Three layers concept of Blobs storage

3 PROBLEM STATEMENT

As described in the introduction, ZaKo3D is a software package, part of the WZL Gear toolbox, and has been used to process the FE-based 3D Tooth Contact Analysis. It reads several geometric data of the flank and a FE-Model of a gear section as the software's variation. The results of the execution are the contact distance, loads and deflections on the tooth.

```
...
*2003 2
*2010  *2011   *2012  *2013
4      -0.18    0.18     10
3      -0.23    0.23     20
2      -0.37     0.37    10
1      -0.55     0.55     2
*2080   2
*FINDATEN
...
```

Figure 2: ZaKo3D variation file.

The variant computing will process a large numbers of variants. For example, there are 8 deviations at pinion and gear in the input data of one analysis, and each deviation with 4 values, so the number of variants the ZaKo3D needs to process is, 48= 65536 variants. Calculating such large amounts of variants one by one on a single PC, it would take way too long. Obviously, this traditional way does not work well. Our method is to split these variants and then compute them on different work units in the cloud in parallel. Figure 2 represents the variants description in the parameter input file. We consider ZaKo3D as an HPC application and develop an automatic parallel framework to distribute the parameter file over a fixed number of cloud nodes and execute the application in parallel.

4 HPC APPLICATION DEPLOYMENT MODEL AND PARALLEL FRAMEWORK

In order to deploy our HPC application on the Azure cloud platform, we developed an HPC application deployment model as described in Figure 3. Furthermore, a parallel framework based on MPI was developed to ensure effective and efficient execution of the application. The function of the framework includes parsing variation file and distributing the variations on Azure nodes in parallel.

The deployment model combines on-premises resources and cloud resources. The scientist deploys the application binary through an on-premises cluster server or any windows desktop based on HPC Scheduler. All computing tasks will be processed by the Azure cloud computing resources.

4.1 Move Application to Cloud

Firstly, the head node and compute nodes on Azure have to be configured with Azure HPC Scheduler. The number of compute nodes as worker role in Azure is allowed to be set from the deployment interface according to the user's requirement. Secondly, after configuration for the cloud hosting service, there are two methods to move an application onto the configured Azure instance: from a local server and from a head node on the Azure portal. We focus on the way of using the head node. Here we have three steps: 1. Move the HPC application pack on the head node. 2. Upload the app package to Azure blob storage. 3. Synchronize all compute nodes with the app package. These operations result in each compute instance maintaining a copy of the HPC application.

4.2 Parallel Framework and HPC Job Scheduler

As described in Figure 3, the deployment module takes the application binary to Azure with a parallel framework based on MPI as the job scheduler. This framework aims to distribute and run in parallel the application binary onto Azure nodes. We take the ZaKo3D for example of HPC application. ZaKo3D is a serial application. It will take a long time for executing a number of parameters due to a large numbers of variations computing involved. Look back to the problem statement, the variation file

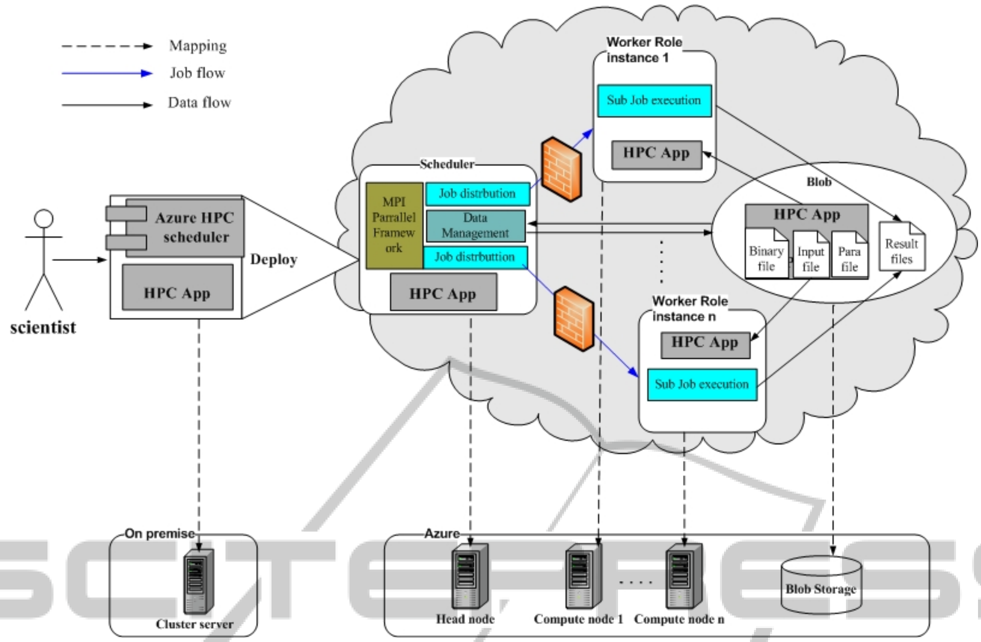

Figure 3: HPC application deployment architecture.

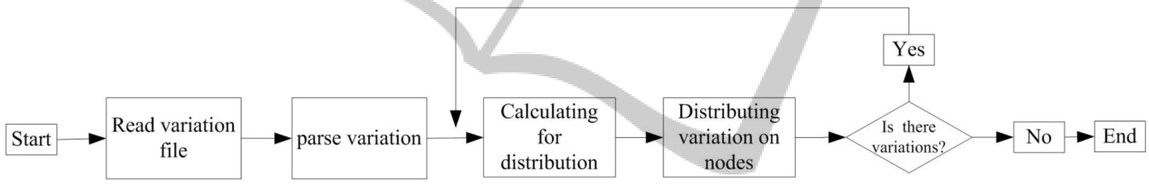

Figure 4: Variation distribution workflow.

needs to be divided and distributed to Azure nodes. When an HPC job in the cloud was scheduled, firstly, the framework will perform a distribution workflow for parsing variation file and distributing variations can be seen in Figure 4, in order to ensure each node gets same number of variations, the variation distribution workflow adopts loop method to distribute the variations to Azure nodes based MPI. After completing the variation distribution workflow, each MPI process will get a part of variations which is an average of total variation, and then merge these obtained variations into a sub parameter file on different Azure nodes owned by the MPI process. This process is depicted by the job flow in Figure 3.

After job submission with our parallel framework, the MPI job will execute on Azure which needs to configure windows firewall setting to allow these MPI sub jobs to communicate across Azure node. The application binary with the sub parameter file is deployed on each compute node called Azure worker instance, and then executed on these allocated nodes in parallel. Windows Azure deals with load balancing for us, so we do not need to handle this on our own.

Three methods can be used to submit an HPC job by means of this model.

1. Azure portal: Azure provides us with a job portal in the HPC scheduler. Through this portal, we can manage all jobs, submit a new job or delete a job, or view the status of a job.
2. Command prompt on Azure node: job submission API is supported by Azure job submission interface similar as in Windows HPC Server 2008 R2 through a remote connection on the command prompt of an Azure node.
3. HpcJobManager on Azure node: an interface for submitting and monitoring jobs, similar to the HPC Job Manager on cluster.

4.3 Data Management

Data management, described by data flow in Figure 3, is supported by a sub module which dedicates to manage the application data on Blob storage, and gather output results from each compute node. The application data, which includes input files, library files and the application executable file, is synchronized on each compute node. As a result, all compute instances get a copy of the application. For gathering the work result, the results generated on each compute node are merged and then copied to the Blob Container by this module. Afterwards, results can be viewed and downloaded from the Azure web portal.

5 PERFORMANCE ANALYSIS

We have conducted a set of experiments on both Azure platform and the RWTH Cluster to compare the difference between cloud resource and on-premises. Moreover, we hope to evaluate whether SMEs can profit from cloud's advantage in HPC.

We deployed ZaKo3D application with the developed parallel framework using our developed deployment model with the same number of variations (120) on the two different platforms. We distribute the work on respectively 1, 2, 3, 4, 5, 6, 8, 10, 12, 15 compute nodes of Azure and cluster (the number of instances to distribute variations must be the divisor of the number of variations in our case), this means that each experiment submits a job on different numbers of processors. It should be pointed out that there are some differences regarding the deployment on these two platforms: After the deployment of the application, all Azure nodes have a copy of the application data automatically, whereas using the cluster, we need to create a copy of the application data manually for each compute node.

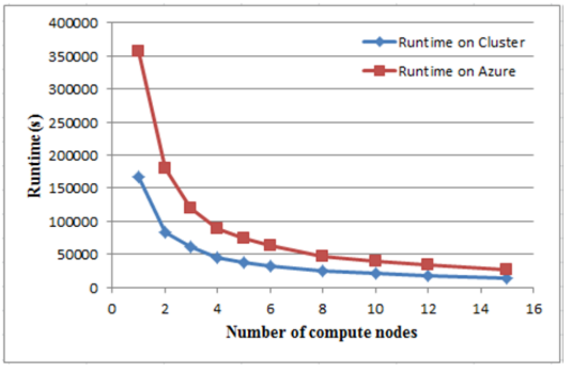

Figure 5: Runtime of ZaKo3D on Cluster and Azure nodes.

We gathered our results on up to 15 nodes of the Windows part of the RWTH Compute Cluster, each node containing 2 Intel Xeon E5450 8-core CPU running at 3.00 GHz with 16 GB memory. The Windows Azure platform can supply us a hosted service with max 20 small virtual instances as compute nodes with Quad-Core AMD Opteron Processors at 2.10 GHz, 1.75 GB of memory.

Figure 5 presents the runtimes on different number of nodes of Cluster and Azure. Due to the different node configuration, we can see that the Azure's curve is always above the cluster one. Azure's performance cannot catch up with the cluster. But as the number of compute nodes increases, the two curves will probably get closer as depicted with 15 nodes in Figure 5. Furthermore, from Figure 6, we can see that scaling is well for a small number of nodes. However, due to the design of the application with a portion of 1.6% sequential code, we are restricted by Amdahl's law and could get a maximum speedup of 58. We assume that applications with a higher portion of parallel code may scale well on Azure nodes for a high number of nodes. This indicates that cloud have good scalability and cloud's power can support HPC application's execution under the circumstances that user's inadequate on-premises resources cannot satisfy the requirements of large scale of compute-intensive applications.

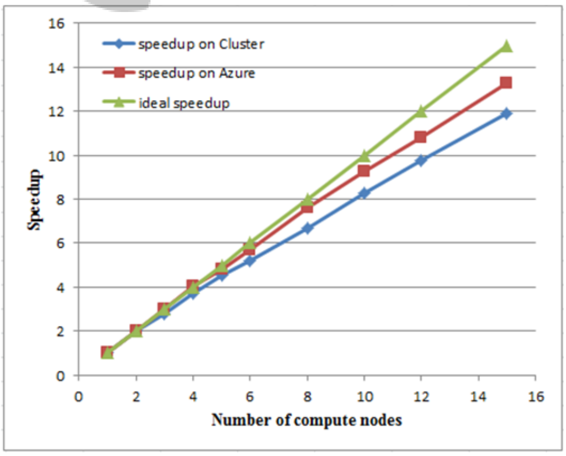

Figure 6: Scalability of ZaKo3D execution on different number of Cluster and Azure nodes.

6 CONCLUSIONS

In this paper, we have presented a cloud deployment model for an HPC application. Moreover, a parallel framework for the HPC application ZaKo3D has been developed which enables the application to run on a number of cloud nodes, thus easing the migrating process of HPC application from existed on-premises resources into the cloud. The advantage of running HPC applications in the cloud environment is that using on-demand cloud resources can reduce the cost of maintenance and save on purchase of the software and equipment.

This work can give a reference to SMEs (small and medium-sized enterprises) to develop their HPC applications for cloud environments. We have to point out that although cloud can leverage the enterprise's HPC application development, current cloud power can only be used to supply to the status when an organization does not have enough on-premises resources to support its development, due to the capability of current cloud cannot catch up with on-premises HPC resources.

Considering our future research, for the parallel framework, we will make efforts to decrease the overhead in the parallel scheduler. Furthermore, based on our performance analysis of the cluster and the cloud, in the next step we will investigate the price of cluster and cloud, through comparing differences between these two platforms, figure out an available rule for users to make the best decision to choose HPC platforms in rational combination of the price and performance within their capability.

REFERENCES

Amazon. Amazon's Elastic Compute Cloud. http://aws.amazon.com/ec2/.

Brecher, C., Gorgels, C., Kauffmann, P., Röthlingshöfer, T., Flodin, A., Henser, J., 2010. ZaKo3D-Simulation Possibilities for PM Gears. In *World Congress on Powder Metal, Florenz.*

Costaa, P., Cruzb, A., 2012. Migration to Windows Azure-Analysis and Comparison. In *CENTERIS 2012 - Conference on ENTERprise Information Systems / HCIST 2012 – International Conference on Health and Social Care Information Systems and Technologies*. Procedia Technology 5 (2012) 93 – 102.

Google Apps. http://www.google.com/apps/intl/en/business/cloud.html.

Hpccloud. Google Enters IaaS Cloud Race. http://www.hpcinthecloud.com/hpccloud/2012-07-03/google_enters_iaas_cloud_race.html.

HPC scheduler. Windows Azure HPC scheduler, http://msdn.microsoft.com/en-us/library/hh560247(v=vs.85).aspxWindows Azure. http://www.microsoft.com/windowsazure/.

IBM. IBM SmartCloud. http://www.ibm.com/cloud-computing/us/en/.

Mondéjar, R., Pedro, ., Carles, P., Lluis, P., 2013. CloudSNAP: A transparent infrastructure for decentralized web deployment using distributed interception. Future Generation Computer Systems 29 (2013) 370–380.

Marozzo, F., Lordan, F., Rafanell, R., Lezzi, D., Talia D., and Badia, R., 2012. Enabling Cloud Interoperability with COMPSs. In *Euro-Par 2012*, LNCS 7484(2012) 16–27.

RWTH WZL. http://www.wzl.rwth-aachen.de/en/index.htm.