VCL Instances Using Cloud Web API 2 Application

[1]Ritika Chugh, [2]Nishu Bansal
Indo Global College of Engg, Abhipur, Mohali

***Abstract* - In this paper, is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier. Web API2 interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.**

***Keywords* – WEB API 2, Service Layer, Balance Loading, Azure Configuration**

I. INTRODUCTION

In the cloud computing field, the development of cloud services by different companies has impacted the scientific community. Several companies, such as Amazon, Google, IBM and Microsoft, are providing scalable cloud computing services that can potentially replace private cluster and grid systems [1]. Among these, the Amazon cloud has been extensively evaluated in the scientific community [2][3]. The reason for this popularity is that it uses the Infrastructure as a Service (IaaS) model [4]. In this model, developers have more flexibility to adapt the programming and execution environment to the needs of their applications. Deployment on IaaS is made using virtual machines, where the developer can choose the operating system and support libraries, among others.

Cyber infrastructure makes applications dramatically easier to develop and deploy, thus expanding the feasible scope of applications possible within budget and organizational constraints, and shifting the scientist's and engineer's effort away from information technology development and concentrating it on scientific and engineering research. Cyber infrastructure also increases efficiency, quality, and reliability by capturing commonalities among application needs, and facilitates the efficient sharing of equipment and services. Today, almost any business or major activity uses or relies in some form on IT and IT services.

SOA is not a new concept, although it again has been receiving considerable attention in recent years . Examples of some of the first network-based service-oriented architectures are remote procedure calls (RPC), DCOM and Object Request Brokers (ORBs) based on the CORBA specifications

On the other hand, the Windows Azure platform is based on the Platform as a Service (PaaS) model, and evaluating its performance using standard HPC applications and benchmarks presents some challenges. In the PaaS model, the developer does not have control over the cloud infrastructure and the operating system, and has to use the tools and libraries supplied by the service provider. More specifically, the PaaS model leads to two challenges for the execution of HPC applications on Azure. First, since the source code of HPC applications is usually aimed at operating systems compatible with Unix, it needs to be converted to the Windows programming model supported by Azure. This is difficult in many cases, especially if support libraries (such as parallelization APIs or I/O libraries) are needed or if the build system or compiler are not compatible with Windows. Second, there are additional modifications necessary to adapt the application to the requirements of the Azure PaaS, such as the transformation of the source code into a library executable on Azure. For these reasons, there are few studies evaluating the performance of HPC applications on Windows Azure. Previous research in HPC on Azure focuses on writing new applications specifically for Azure [5] [6], which leads to duplication of work and lack of comparability between different solutions. The goal of this paper is to port a set of well- known HPC benchmarks, the NAS Parallel Benchmarks (NPB) [7], to the Azure PaaS and evaluate their performance. Our work has three main contributions. First, we analyze the complexity of porting existing HPC applications to the Azure platform, detailing the challenges and their resolutions. Second, we introduce a metric which compares the price and performance of different cloud computing solutions, which leads to the notion of efficiency of a cloud computing solution. Finally, we evaluate the performance and efficiency of Azure by comparing it to a real machine and to an Amazon IaaS cloud.

II. LITERATURE REVIEW

Related work in this area can be divided into three categories: research on running Instance applications on Windows Azure, performance evaluation of clouds, and migration of applications into the cloud. In this section, we will give a brief overview over this research and compare it to our work. Lu et al. [6] and Li et al. [5] evaluate the possibility of writing scientific applications to run on the Azure platform. They conclude that it is a viable solution for scientific applications that involve large data sets. In contrast to these works, we do not write new applications in Azure, but rather port existing Instance applications to the Azure platform and evaluate the possibility and the difficulties of doing so.

The article of Walker [8] evaluates the performance degradation of Amazon EC2 instances. He compares the performance of the NAS Parallel Benchmarks with their performance on a real machine and analyzes the network efficiency by executing the mpptest [9] MPI benchmark. Both computing and network performance proved to be challenges for the cloud solution. In this paper, we perform similar experiments with the Azure cloud and compare the performance and efficiency to the Amazon cloud and a real machine.

Deelman et al. [2] evaluate the cost of Amazon EC2 by porting a real-life astronomy application to the cloud and execute it using different resource provisioning plans. They conclude that the cloud is a cost-effective option since the scientific application provider does not need to buy an entire cluster for a few runs of the application. Many clusters are underused as the hardware quickly becomes obsolete. The cloud solves this problem as it is a responsibility of the cloud provider to keep upgrading the hardware and provide an upto- date service to the users. In our research, we compare Azure and EC2 and evaluate the relation between cost and performance of both clouds.

The performance and performance variability of the EC2 cloud is evaluated by Jackson et al. [10], comparing the cloud system to real clusters by running several scientific applications. Their results show that performance degradation and variability of the EC2 are very high, citing the speed of the interconnections as a major problem.

Ekanayake et al. [11] evaluate the performance of a new MapReduce implementation, CLG- MapReduce, and compare it to the existing implementations Hadoop and Dryad on cloud platforms. Furthermore, they compare the performance to a more traditional MPI implementation. Their results show that for various applications, CLG-MapReduce has a similar performance to MPI. However, almost all evaluated applications, such as the EP benchmark from NAS, have little or no communication between the threads. As communication is an important part of HPC applications, we evaluate a wider range of applications, with different amounts of communication and varying communication patterns.

III. PROBLEM FORMULATION

In this section, we explain the porting process for an Instance application to the Azure PaaS platform. First, we give a short overview of the platform. Afterwards, we explain in detail the challenges and solutions of porting the application to the Win32 programming model and to the Azure platform. The Web Role is the instance that provides the web interface to the user, allowing interactivity with the OS in order to program and execute code.

In our research work we faces the following problems in already built app :-

- Platform dependent
- Only work on web not on desktop
- Do not support for hybrid cloud
- No use of rest API

So by keeping these things in mind we started our research work so our present work is to make a window store app which generally called as(metro app) that app includes a feature for:- To create a native service app for SQL that having Saas as a window azure which includes:-

- Restore and replication with Saas service.
- Dial up dial down.
- CDN Cloud with API2 Services with Instance over USA and INDIA
- Access from Mobile and Web Interface
- Microsoft remote desktop protocol. The methodology we are using in our work are:-Azure RemoteApp helps employees stay productive anywhere, and on variety of devices - Windows, Mac OS X, iOS, or Android. Your company's applications run on Windows Server in the Azure cloud, where they're easier to scale and update. Users can access their applications remotely from their Internet-connected laptop, tablet, or phone.

Window protocol directory services

The Lightweight Directory Access Protocol (LDAP) is a directory service protocol that runs on a layer above the TCP/IP stack. It provides a mechanism used to connect to, search, and modify Internet directories.

ADSL directory

Azure Active Directory is a service that provides identity and access capabilities for on- premises and cloud applications. Developers can use features of Azure AD to: Implement single sign-on and single sign-out for line of business (LoB) applications and software as service (SaaS) providers.

Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed datacenters. It provides both PaaS and IaaS services and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems.

IV. CONCLUSION

Our research work is based on Azure in which a metro app is to be implemented having feature of platform independence, supporting web as well as desktop application and work on hybrid cloud using WEB API2. In this paper, we evaluated the Windows Azure platform as a platform to run general Instance applications. We showed how to port applications written for the UNIX programming model to Azure and compared their performance to a real machine and an Amazon EC2 instance with similar characteristics. We measured the execution time and examined the efficiency of the cloud solutions in terms of performance and cost with a new metric.

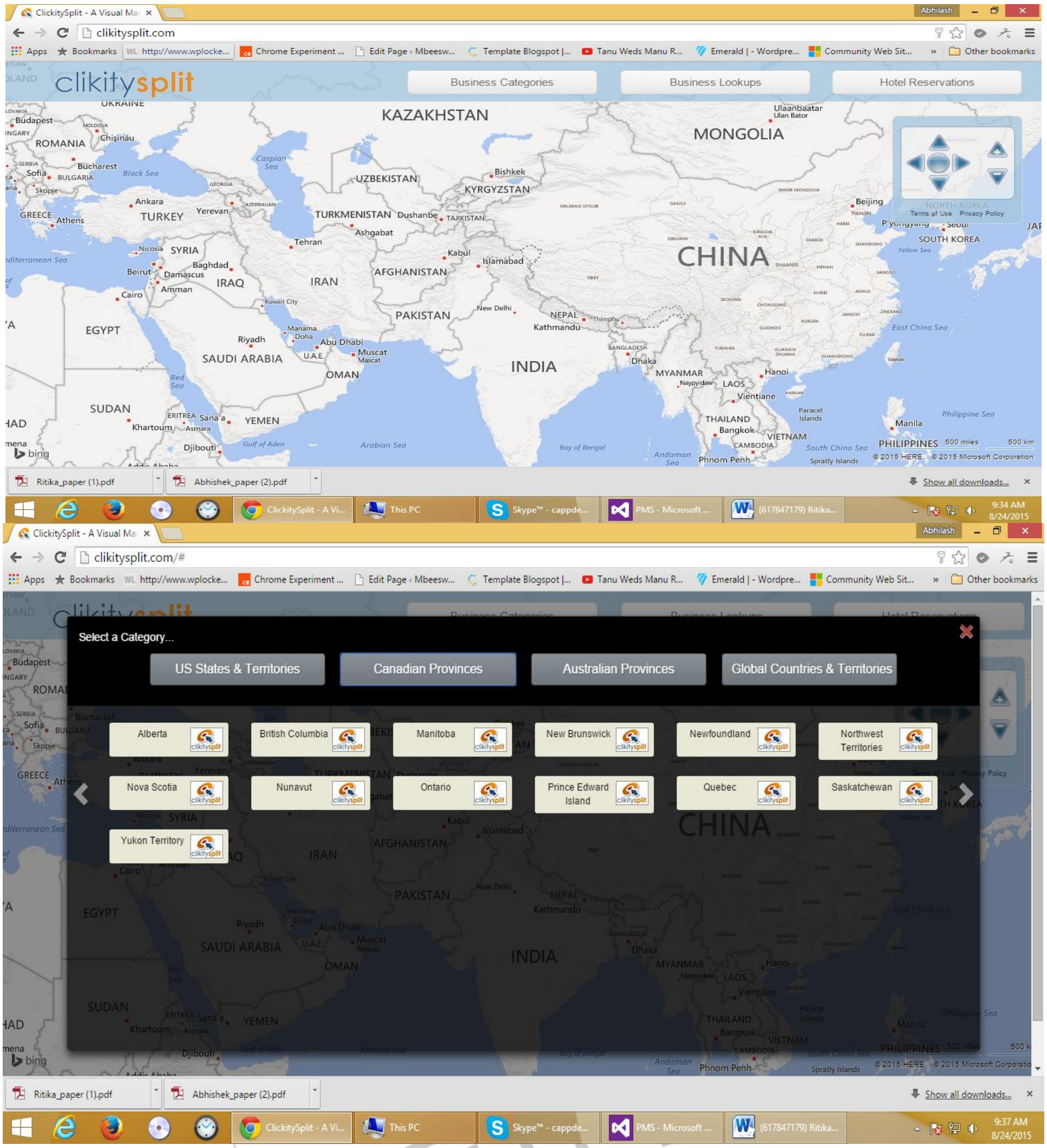

Fig 1:Wep API 2 Universal Database in Azure

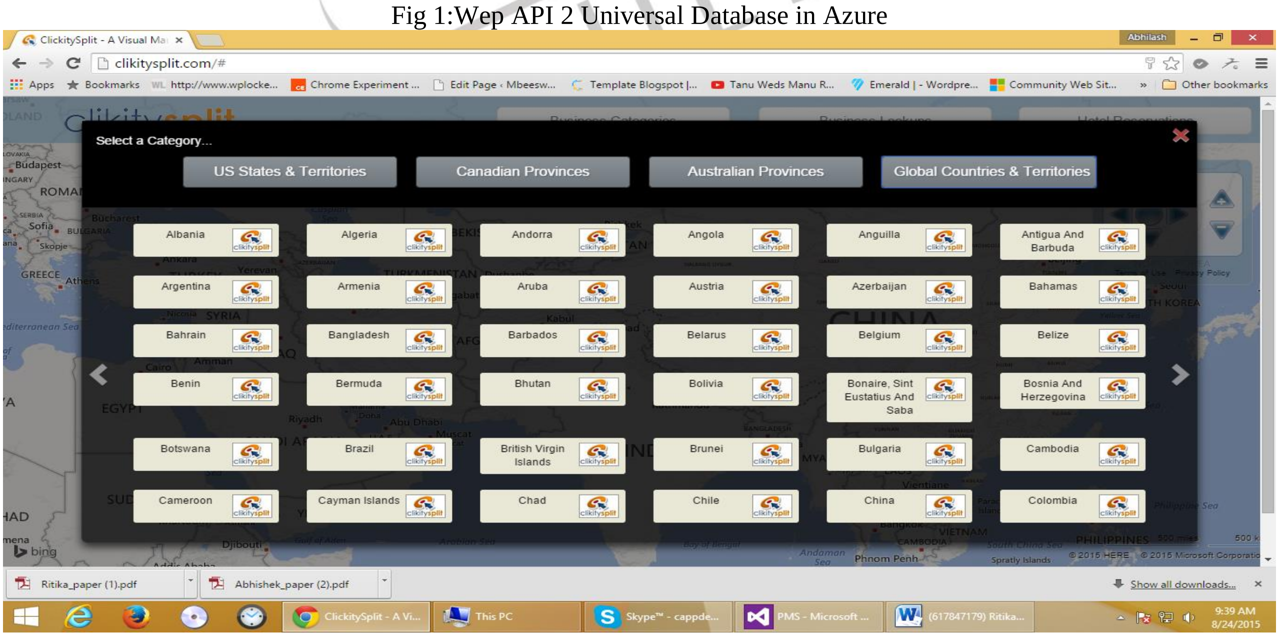

Fig 2: Other Countries Database Access

V. ACKNOWLEDGEMENT

Working on this thesis **of VCL INSTANCES USING CLOUD WEB API 2** provided a unique experience and analysis, I feel great pleasure and privilege in working over this research. I am deeply indebted to **“Indo Global Colleges”** for the invaluable guidance, support and motivation for the many other aids without which it would have been impossible to complete this project. I have no words to express my deep sense of gratitude for (Nishu Bansal) for her enlightening guidance, directive encouragement, suggestions and constructive criticism for always listening to our problems and helping us out with their full cooperation.

VI. REFERENCES

[1] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. Anderson, “Cost-benefit analysis of cloud computing versus desktop grids,” in Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on, may 2009, pp. 1 –12.

[2] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, “The cost of doing science on the cloud: The montage example,” in High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for, nov. 2008, pp. 1 –12.

[3] A. Iosup, S. Ostermann, M. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, “Performance analysis of cloud computing services for many-tasks scientific computing,” Parallel and Distributed Systems, IEEE Transactions on, vol. 22, no. 6, pp. 931 –945, june 2011.

[4] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” Tech. Rep., 2011. [Online]. Available:http://www.mendeley.com/research/ the-nist-definition-about-cloud-computing/

[5] J. Li, M. Humphrey, D. Agarwal, K. Jackson, C. van Ingen, and Y. Ryu, “escience in the cloud: A modis satellite data reprojection and reduction pipeline in the windows azure platform,” in Parallel Distributed Processing (IPDPS), 2010 IEEE International Symposium on, april 2010, pp. 1 –10.

[6] W. Lu, J. Jackson, and R. Barga, “Azureblast: a case study of developing science applications on the cloud,” In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, ser. HPDC ’10. New York, NY, USA: ACM, 2010, pp. 413–420. [Online]. Available:http://doi.acm.org/10.1145/1851476.1851537

[7] H. Jin, M. Frumkin, and J. Yan, “The OpenMP implementation of NAS parallel benchmarks and its performance,” NASAAmes Research Center, Technical Report NAS-99-011, 1999.

[8] E. Walker, “Benchmarking amazon ec2 for high-performance scientific computing,” Usenix Login, vol. 33, no. 5, pp. 18–23, 2008.

[9] W. Gropp, “Reproducible measurements of MPI performance characteristics,” Recent Advances in Parallel Virtual Machine and, 1999. [Online]. Available: http://www.springerlink.com/ index/PE2J40W56L9GM1Y9.pdf

[10] K. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. Wasserman, and N. Wright, “Performance nalysis of high performance computing applications on the amazon web services cloud,” in Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, 30 2010-dec. 3 2010, pp. 159 –168.

[11] J. Ekanayake and G. Fox, “High performance parallel computing with clouds and cloud technologies,”Cloud Computing, pp. 20–38, 2010.

[12] A. Li, X. Yang, S. Kandula, and M. Zhang, “Cloudcmp: comparing public cloud providers,” in Proceedings of the 10thannual conference on Internet measurement. ACM, 2010, pp. 1–14.