RESEARCH ARTICLE OPEN ACCESS

Geochronos File Sharing Application Using Cloud

Sathish.Y[#1], Surabhi. S[*2], Sanila.S[#3]
B.E(Computer Science and Engineering) Angel College of Engineering and Technology, Tirupur Anna University, Chennai

Abstract
Accessing, running and sharing applications and data at present face many challenges. Cloud Computing and Social Networking technologies have the potential to simplify or eliminate many of these challenges. Social Networking technologies provide a means for easily sharing applications and data. Now a day's people want to be connected 24x7 to the world around them. Networking and Communication have come together to make the world a small place to live in. People want to be in constant touch with their subordinates where ever they are and avail emergency services whenever needed. In this paper we present an on-line/on-demand interactive application service (Software as a Service). The service is built on a cloud computing basement that provisions virtualized application servers based on user demand. An open source social networking platform is leveraged to establish a portal front-end that enables applications and results to be easily shared between users. In the proposed system users can access the documents uploaded into the cloud by others and provide any data they have in hand to other users through the same cloud. This also allows the users to have an interactive session through the chat screens present in the cloud. The paper also highlights some major security issues existing in current cloud computing environment.
Keywords- Cloud Computing, Social Networking, Virtualized application, Interactive session.

I. INTRODUCTION

Researchers face many computational and data related challenges. These challenges can detract and take away valuable time that could be better spent focusing on their research. One such challenge is the management of applications that researchers use for conducting simulations, processing and analyzing data, and visualizing results. Such applications could be commercial, open source, or developed in-house. Different applications may be designed for different operating systems and have different software dependencies. New versions of applications may also be released frequently. Many of these applications are interactive in nature, have a graphical user interface and visualization capability or provide a way to receive feedback from the user. These applications typically haven't been suited for batch based high performance computing environments where on-demand access to resources has not been practical due to queue wait times and limited preemption capabilities. Downloading, installing, configuring and keeping applications up to date on heterogeneous computing systems can be a daunting and difficult task. Providing access to these applications from anywhere (i.e., home, office, lab, field, etc.) and at anytime is also challenging. In addition, many researchers may develop their own software, without realizing that others might have already done so. The ability to discover, share and access applications, data and results in a more transparent and on-demand manner could be of great benefit to researchers. Recent advances in cyber infrastructure technologies are making it possible to develop solutions to address the above concerns. Cloud computing solutions, such as the Amazon Elastic Compute Cloud (EC2) [2], enable transparent and on-demand access to dynamic and scalable applications/ services over the Internet. Virtual machine technologies,used by many cloud computing solutions, enable different applications developed for heterogeneous systems to be hosted on the same physical hardware with their own custom operating system and software stack. Software and applications can be maintained and updated by the developers or a technical team, so researchers can focus on using the applications rather than managing them. Social networking platforms, such as Facebook [10] which has grown to over 250 million users, have become very popular. They allow individuals to easily share/publish data and media and keep in contact with one another. Social networks can enable researchers to more easily share applications and results and collaborate one with another. In this paper we present a service called the Interactive Application Service (IAS) that combines cloud computing and social networking technologies to provide researchers with the ability to more easily discover, share and access applications and results from anywhere at any time. Users are able to access applications, including legacy applications without modification of any code, over the Internet just as if they were running it on their local system.

II. RELATEDWORK

A. IAS Architecture

The Interactive Application Service (IAS) combines the use of a cloud based infrastructure with a social networking front-end. IAS users are members of a social networking portal on which they can communicate with other members as well as exchange files, write blogs, take part in forum discussions, and comment on topics and issues. In addition to these interactions, IAS allows portal users to run applications in the cloud. This functionality is achieved by the integration of IAS with the Web portal as well as with a cloud computing environment in which the applications are run. An overview of the IAS architecture is shown in Figure 1 The service architecture can be conceptually divided into three layers: cloud infrastructure, IAS service, and social networking front-end. The infrastructure is composed of network connected compute nodes, virtualization software and a set of services that provide dynamic and scalable computing environment. The IAS service consists of the software components that enable the execution and visualization of interactive applications on the cloud. The social networking front-end is comprised of a Web server, a social networking engine, and a database management system.

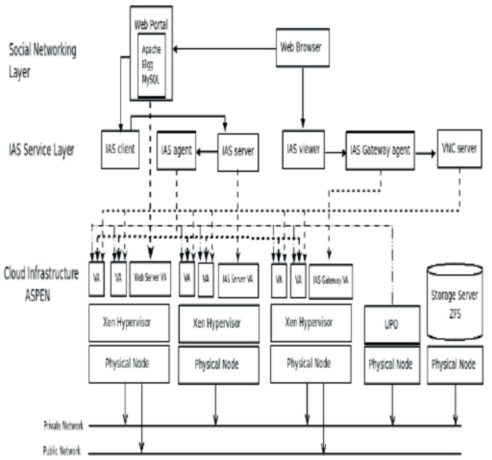

Figure 1. IAS architecture.

B. Cloud Infrastructure

The cloud infrastructure on which the IAS has being deployed is the Automated Service Provisioning Environment (ASPEN) The goal of ASPEN is to enable the dynamic allocation of computational resources in a flexible and controlled manner. In order to satisfy dynamic demands for computational resources a virtualization layer which provides dynamic resource allocation is installed on top of each physical node. A utility provider orchestrator (UPO) is responsible for the management of physical resources and for the allocation of virtual appliances (VAs) to these resources.A VA, in the context of this work, is defined as a virtual machine which is streamlined for fast boot-up and configured to run a specific application.

All VAs run on top of the Xen hypervisor [3]. Xen is a high-performance, virtual machine monitor which runs on top of the physical hardware and enables the execution of multiple operating systems on a single computing appliance.This is achieved by logically partitioning the hardware resources and executing virtual machines (VMs) in a concurrent manner. Xen supports the dynamic instantiation of VMs which is advantageous in environments in which the demand for computation dynamically changes over time.

III. GEOCHRONOS PLATFORM

The Interactive Application Service has been integrated as part of GeoChronos [11], a portal aimed at enabling members of the earth observation science community to share data and scientific applications, and to collaborate more effectively.

The project is funded as part of the CANARIE Network Enabled Platforms program [4]. Earth observation scientists collect data from a variety of sensors at different spatial and temporal resolutions, including ground based, airborne and satellite sensors, that help them in analyzing the Earth's ecosystem. The GeoChronos portal facilitates the collection, storage, sharing, processing and analysis of this data. The IAS, is a service currently available to scientists on the portal that will enable them to visualize and analyze data using common tools used by the community on-line. Other services that are currently being developed include a batch processing service for handling automated work flows and longer running tasks, and libraries for storing and browsing spectra and other data.

The architecture for the GeoChronos portal and Interactive Application Service follows that laid out in Section 3.The portal is built on top of the Elgg social networking platform. The Elgg base provides users with a variety of social networking functionality including blogs, tags, media/ document sharing, wikis, friends/contacts, groups, discussions, message boards, calendars, status, activity feeds and more. Much of this functionality is available from a users customizable dashboard as can be seen in Figure 3.

IV. IMPLEMENTATION AND EVALUATION

To enable portability between various devices, we built our prototype implementation in java. Our implementation was tested on one of the state of the art 3G devices.

Since the main bottleneck of this protocol is the device-todevice authentication that is based on a ZKP protocol, wefocus our evaluation on this aspect. It is the most challenging part both in terms of computation and bandwidth. The applied ZKP protocol is based on graph isomorphism that we described in our earlier work [34], [36]. It requires two parties: a client and a server. In our protocol, devices act as both clients and servers for the device-to-device authentication. From the perspective of computations and data generations,the client application, which is in our case a browser executed at a user's device, is responsible for the following operations:

Generating a private key, Pk, from the user password

Generating a public key, Puk, from the private key

Generating random data of size (Pk)2 for the ZKP authentication

Responding Server's challenge vector Checking server timeout Delaying responses to provide a correct implementation for concurrent environments

The server, which is in our case a home network gateway, is responsible for the following operations:

Looking-up the user's public key Storing user's random data Generating a random vector of size (Pk) to match the user's random data Verifying the user's response Checking its clients timeouts

In comparison with classical authentication approaches, such as HTTP Basic and Digest, there is a requirement of additional computations for both parties. Table I depicts how the ratio of the computations that are conducted by the server and the client depends on the number of authentication challenges and the graph size. We observe that once we increase one of these parameters, the server needs proportionally more time to respond the client's request.

However, as we see in Figure 5, the amount of data sent by the server, which is equal to data received by the client, is much smaller than the data sent by client.

TABLE I
DATA MEASUREMENTS

Graph size	Challenges	Sent [KB]	Received [KB]	Total [KB]
32	10	3.7	1.3	5.0
32	20	7.2	2.3	9.5
32	30	10.7	3.2	14.0
64	10	12.3	3.0	15.4
64	20	24.5	5.0	29.5
64	30	36.7	6.9	43.5
128	10	45.0	8.0	53.0
128	20	89.8	11.9	101.7
128	30	134.6	15.7	150.3
256	10	171.7	24.2	195.9
256	20	343.2	31.8	375.1
256	30	514.7	39.5	554.3

The number of bytes exchanged for various authentication parameters.

The number of bytes exchanged for various authentication parameters. Figure 5 depicts the impact of the graph size and the number of challenges on the authentication times. It shows that together with the increase of the number of challenges, the time required for computations grows linearly. However, the increase of the size of the graph causes quadratic time increase for the same number of challenges (see Fig. 6).

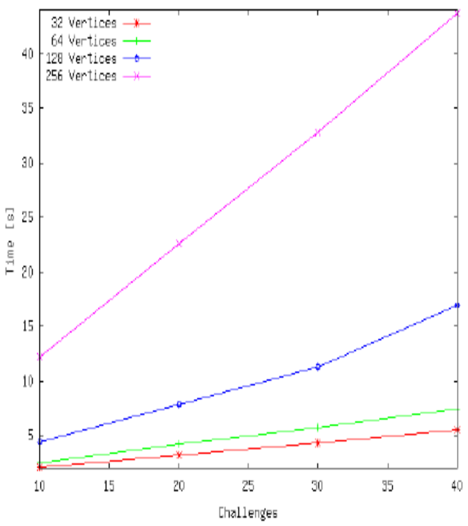

Figure 5: The authentication time for various parameters

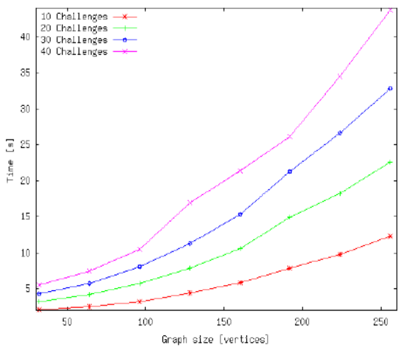

Figure 6: Relation between graph size and time in function of the number of challenges

We observe that the amount of data exchanged between the client and the server grows linearly for a fixed graph size and the increasing number of challenges. Increasing the graph size has, however, polynomial impact on the number of data exchanged. This dependency is best described by O(n2). The quadratic dependency relates to the fact that the graphs are kept in matrices whose sizes are square function of their number of vertices. This property is important when customers of a given system are required to pay for bandwidth usage. In such a situation, the selection of this parameter would influence not only the system security properties but also the price of the service.

V. CONCLUSIONS AND FUTUREWORK

In this paper we introduced the Interactive Application Service (IAS), a service providing users on-demand access to applications interactively over the Internet. The service is built on a cloud computing infrastructure called ASPEN,that dynamically adapts to meet the needs of users and uses efficient image management techniques to rapidly provision resources. With some modifications, the service could also be adapted to work with other cloud platforms. Collaboration and sharing of applications and data are achieved through a social networking front end built on the Elgg open source social networking platform. The IAS has been successfully deployed in the GeoChronos portal, a portal enabling members of the earth observation science community to share applications and data with each other.

Naturally the greatest risk to any such system is at the human level. It has been assumed that high levels of trust exist when services are shared. But there are risks that a friend or neighbor may decide to abuse the trust shown to them and effectively hi-jack a user's services and personalized content. In this paper we have not directly considered the potential for such service hi-jacking and associated preventative measures. From the perspective of a service provider there are straightforward session management mechanisms for ensuring that only a single real-time access to a cloud service is available to a registered user and these could be readily adapted to address such issues. However this lies outside the scope of the current work and we prefer to leave a detailed consideration of such issues for now. Again we recall that the purpose of this paper is to examine how we can move beyond the traditional model of device authentication for home networks, and begin to implement a more user-centric approach in line with current trends in mobile network services. It is shown that ZKP techniques combined with cloud computing services can offer a suitable and practical approach to this problem.

VI. ACKNOWLEDGMENT

Thanks to Mr.S.Balaji,Asst.Prof,Department of Computer Science and Engineering,Angel College of Engineering and Technology for his contribution to complete this project and motivating us to do better in future.

REFERENCES

[1] Academia.edu Website.*http://www.academia.edu/*. Accessed July, 2009.
[2] Amazon. Elastic compute cloud website.*http://aws.amazon.com/ec2*. Accessed July, 2009.
[3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho,R. Neugebauer, I. Pratt, and A. *Warfield. Xen and the art of virtualization.* In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pages 164–177, 2003.
[4] CANARIE Network Enabled Platforms (NEP) Program. *http://www.canarie.ca/funding/nep/funded projects.html*. Accessed July, 2009.

[5] A. Charbonneau and V. Terskikh. *Spectrogrid: Providing simple secure remote access to scientific instruments*. In *HPCS*, pages 76–82. IEEE Computer Society, 2008.
[6] Cluster Resources, Inc. Moab Cluster Suite Website. *http://www.clusterresources.com/ pages/products/moabcluster-suite.php*. Accessed July, 2009.
[7] Cluster Resources, Inc. Torque Resource Manager Website.*http://www.clusterresour ces.com/pages/products/torqueresource-ma nager.php*. Accessed April, 2008.
[8] R. Curry, C. Kiddle, N. Markatchev, R. Simmonds, T. Tan,M. Arlitt, and B. Walker. *ASPEN: An automated service provisioning environment for data centres.* In *Proceedings of the 15th HP Software University Association (HP-SUA) Workshop*, pages 189–200, 2008.
[9] Elgg Website. *http://elgg.org/*. Accessed July, 2009.
[10] Facebook Website. *http://www.facebook. com /*. Accessed July, 2009.
[11] GeoChronos Portal. *http://geochronos.org/*. Accessed July,2009.
[12] Globus Alliance Website. *http://www.globus .org/*. Accessed July, 2009.
[13] Google App Engine Website. *http://code.google.com/appengine/*. Accessed July, 2009.
[14] Java web start. Accessed July, 2009. h*ttp://java.sun.com/javase/technologies/desk top/javawebstart/index.jsp*.
[15] P. J¨onsson and L. Eklundh. *TIMESAT - a program for analysing time-series of satellite sensor data. Computers andGeosciences*, 30:833–845, 2004.
[16] K. Keahey, R. Figueiredo, J. Fortes, T. Freeman, and M. Tsugawa. *Science clouds: Early experiences in cloud computing for scientific applications.* In *Proceedings of the First Wrokshop on Cloud Computing and Its Applications*, 2008.
[17] KVM Website. *http://www.linux-kvm.org/*. Accessed July,2009.
[18] LinkedInWebsite. *http://www.linkedin.com/*. Accessed July,2009.
[19] Microsoft. Windows Azure Website.*http://www.microsoft.com/azure/*. Accessed July, 2009.
[20] MODIS reprojection tool page. *https://lpdaac.usgs.gov/lpdaac/tools/modis reprojection tool*.Accessed July, 2009.
[21] Spec Net toolkit software page. *http://spectralnetwork.net/specnet toolkit. htm*. Accessed July, 2009.