Computational Cloud framework implementation to support scalable and cost efficient services for e-Learning systems

Shahbaz Zafar
Research Scholar
Department of Computer Science
PACIFIC, University-Udaipur (Rajasthan)

Abstract
Cloud Framework presented by us is a Cloud implementation to support scalable and cost efficient services for e-Learning systems from basic to higher education in a country like India. The main focus of this framework is on improving resource utilization through smart load balancing and providing scalable IT services for e-Learning organizations. Another design proposed by us is a Cloud application provides the infrastructure for implementing an e-Learning system. It highlights the benefits of Cloud computing as a way to efficiently provide computational and storage resources on demand (SaaS). Our proposed e-Learning Computational Cloud framework is motivated from Apache Struts, Sun Microsystem (Oracle) Java Server Faces and Spring Source Spring frameworks. This framework provides comparative flexibility and makes it easier to build a development framework. Finally, session maintenance and transaction control does not depend on connection protocol such as HTTP, SOAP, cookies etc. Session and transaction maintenance is defined and configured in the database. The usage of EJB ensures that coordination between the container objects in distributed network will be reliable.

Keywords: e-Learning systems, Computational Cloud framework, Service Oriented Architecture

1 Introduction

The cloud computing environment provides the necessary foundation for the integration of platform and technology. It integrates teaching and research resources distributed over various locations by utilizing existing conditions as much as possible to meet the demands of the teaching and research activities. The cloud computing environment with respect to e-Learning offers new ideas and solutions in achieving interoperability among heterogeneous resources and systems. The cloud service model of e-Learning mean that the Internet can be used as huge workspace, repository, platform, and infrastructure. Learners can access to the Internet from anywhere at any time, using widely spread mobile devices but the existing cloud computing technologies are only passively responsive to users’ needs. This situation necessitates proactive cloud services rather than passive services. Since learners typically carry mobile devices of some kind at their hands, the volume of information and services processed through the devices continues to increase. Clouds can provide device independence from...
any particular hardware vendor and offer implementation of resource and cost sharing from among a large pool of users. Within this resource sharing concept, additional specific implementations help to enhance these general gains in technical performance, with potential follow-on economic savings. For example, technical efficiency and scalability is enhanced with centralization of infrastructure, location independence (as well as device independence), and efficiency in utilization through management of user demand load to the cloud system through implementation of software that controls simultaneous multi-user or project access. Beyond these general technical enhancements, the idea of individual cloud architecture designs, specific implementations, and usage profiles have the potential for additional technical and economic impacts that can lead to better performance, throughput, and reduced costs. Areas at each specific site where such economies may be improved include:

- Network bandwidth and network load to the system
- Reliability and “up-time” of the system
- Site specific operational profile, including concurrent resource usage profile
- Services mix (IaaS, PaaS or SaaS - which ones and in what proportion)
- Efficient on demand allocation and aggregation, and de-allocation and deaggregation, of CPU, storage, and network resources
- Type of virtualization used (bare-metal to virtual load ratio)
- Scalability and rate of adaptability of the cloud to meet changing user demands
- Sustainability of the system under varying workloads and infrastructure pressures
- Serviceability and maintainability of the architecture along with the overall cloud computing system and user interfaces and application programming interfaces (API) Etc.

All of these factors can certainly apply to both commercial as well as educational/non-profit institutions. However, the weight or emphasis of each of these individual factors will be influenced by the design goals for each cloud computing installation and the intended use of the cloud computing system for a particular user base. This weighted emphasis on different aspects of cloud computing are driven by a combination of user requirements, and in the case of many commercial operations, a business plan that hopes to capture a niche of the business computational market by selling these services and capabilities at a profit.

2 User Requirements

The first questions any IT architect must address when designing and configuring a system that provides IT services and capabilities is to understand who the users are and what they require from the system to support their work, i.e., the architect needs to construct and understand the operational profile of the system [4, 5]. There are several categories of users in a higher educational institution environment. There are developers of the base-line computing system infrastructure and of the base-line services (e.g., bare-metal images), there are software authors, and there are developers of enhanced images and services and service integrators (e.g., lecturers and teaching assistants). However, by far, the largest fraction of users in a higher educational institution are students and faculty (the end-users) who for most part are not interested in the underlying IT and prefer “one-button” access to services they need. It is this group that is also the driving engine and whose usage patterns can validate and justify the economic aspects of any cloud computing implementation. Cloud computing systems serving users within a university environment must at least provide the following capabilities for the faculty, researchers, staff and students:

Shahbaz Zafar: - Computational Cloud framework implementation to support scalable and cost efficient services for e-Learning systems
Deliver services and support to a wide range of users from the novice to the most sophisticated expert researcher - from users who can barely find the terminal, to those who are expert supercomputer users.

Deliver a wide-range of course materials and academic support tools to instructors, teachers, professors, and other educators and university staff as part of the academic mission of the institution.

Deliver research level computational systems and services in support of the research mission of the university. Fulfilling such a set of user requirements across components of a distributed system of hardware and software, that is often coupled with a given level of network support, is sometimes categorized under the term of a “service-oriented architecture”.

These types of IT systems provide the user with a given functionality, capacity, and quality of delivery connected through a mix of some set of both tightly and loosely coupled components. The distributed components have qualities that are on-demand or batch, reusable, sustainable, scalable, customizable, secure, and reliable. In addition to all of these technical characteristics, the IT architect of such a cloud computing system must demonstrate that such a system is cost-effective to operate and maintain.

3 E-Learning Task Controller Cloud

The eLTC has a Task Controller Service Bus as a common platform for communication with the Task Controller's components e.g. the Object Serializer/Deserializer performs marshalling/unmarshalling of the Task Service Model's data transfer object. The Service Verification Engine validates the requested service as per the rules defined in the Data Model and thereby it guards against further execution process in case of validation failure. The Service Event Controller is a Task Manager for the Web services event notification between service consumer (View) and service provider (Model). The Transaction Controller manages the Web services transactions across distributed systems including Local Database, Message Queues etc. for two-phase atomic commit and concurrency control. The Authentication/Authorization Engine confirms service request corresponding to the given actor or user as per the authentication and/or authorization policies defined in the Local Database (i.e. Data Model). The Business Delegate identifies actual endpoint services (as per the Request Binder mapping) from the Task Service Pool which is a Model and delegates the filtered service request to the Model for task processing. The Data Model is also used by the eLC and eLTC for service related data processing like service definition, transactional data etc. Java Database Connectivity (JDBC) or Object-relational Mapping (ORM) is used to connect to the Data Model. The Task Service Pool or Model encapsulates endpoint service objects which implement the actual task definition so that any e-Learning Task (like Quiz, Examination, Contest, Lecture etc.) can be defined in the form of Java Bean as a service into the Task Service Container. Therefore, the Task Controller Endpoint EJB acts as a proxy (stateless session EJB) to the underlying Task Service definitions in the Model. This elicits the Web services Orchestration in the one-to-many relationships [5]. The Model has a common Plug-in Service Bus as a communication channel to the task-related add-on engines such as Formula, Grading, Human Language, Alert etc. These plug-in engines can also be accessed as independent services through eLTC. In J2EE framework, Web services are deployed mainly in two modes: the Servlet-based Web services endpoint model that is placed in a Web-tier and the Stateless Session Bean-based Web services endpoint model that is placed in an EJB-tier. Under the J2EE context, Web services are...
defined as a set of endpoints operating on messages which means that the endpoints receive request messages and then send back response messages. These endpoints operate within a container, which provides a Web services runtime environment in the same way the EJB container provides a runtime environment for EJB beans.

4 Modules of eLC System model

Our e-Learning system based on Cloud computing infrastructure is composed of three layers:

*Infrastructure layer*, Content layer, and Application layer. It has four working modules: monitoring module, policy module, arbitration module, and provision module. Infrastructure layer is the resource pool of an e-learning system. Hardware and software virtualization technologies are used to ensure the stability and reliability of the infrastructure. Supplying computation and storage capacities for higher layers, it is the energy source of an e-learning ecosystem.

*Content layer* mainly consists of e-learning contents, such as Web file systems, database systems, Web Services, and so on. Except for content storage and maintenance, this layer exposes the standard interfaces and APIs of contents for higher layers.

*Application layer* consists of e-learning services, systems, tools, and so on. It also provides functions and interaction interfaces for users or other programs.

*Monitoring module* is keeping track of the executions of requests, the real-time configuration information and resource utilization levels of species, including the health of CPU, memory, I/O, and so on. The data of monitoring module is the source to adjust the balance of an e-learning system.

*Policy module* establishes and maintains the teaching and learning strategies, the run-time and resource scheduling strategies. According to the data from monitoring module and the strategies of its own, policy module establishes specific solutions, and then triggers provision module. Policy module also decides which species to get higher priorities on resource scheduling according to some e-learning policies in order to safeguard the running of critical businesses. Policy module is the core of an e-learning system.

*In the arbitration module*, some policies are made by experts manually; requests from users are completed; and some disputes among species within the e-learning system are solved. Arbitration module amends, adjusts, and improves the resource allocation and management. It also establishes usage modes for different kinds of users based on the learning styles, learning preferences, and cognitive levels. Arbitration module is an efficient complement to the policy module, while the privilege of its policy is higher than the one in the policy module. Provision module starts the execution of resource allocation solutions set by the policy module and arbitration module, and deploys resources referred to users or species automatically in a short time. If the request comes from a user, some related information such as IP, user name and password will be supplied.

5 Prototype Design

Shahbaz Zafar - Computational Cloud framework implementation to support scalable and cost efficient services for e-Learning systems
We used standard Java technologies and tested them on Google's scalable web application infrastructure. The Java environment provides a Java 6 JVM, a Java Servlets interface, and support for standard interfaces to the App Engine scalable data store and services, such as JDO, JPA, Java Mail, and J Cache. App Engine runs Java applications using the Java 6 virtual machine (JVM). The App Engine SDK supports Java 5 and later, and the Java 6 JVM can use classes compiled with any version of the Java compiler up to Java 6. App Engine uses the Java Servlet standard for web applications. App's servlet classes, Java Server Pages (JSPs), static files and data files, along with the deployment descriptor (the web.xml file) and other configuration files, in a standard WAR directory structure. The JVM runs in a secured "sandbox" environment to isolate your application for service and security. App Engine provides scalable services that apps can use to store persistent data, access resources over the network, and perform other tasks like manipulating image data. Apps can use the App Engine data store for reliable, scalable persistent storage of data. The data store supports two standard Java interfaces: Java Data Objects (JDO) 2.3 and Java Persistence API (JPA) 1.0. The App Engine Memcache provides fast, transient distributed storage for caching the results of datastore queries and calculations. The Java interface implements JCache (JSR 107). The service can handle CPU-intensive image processing tasks, leaving more resources available for the application server to handle web requests. The App Engine Java SDK includes tools for testing application, uploading your application files, and downloading log data. The SDK also includes a component for Apache Ant to simplify tasks common to App Engine projects. The development server runs your application on your local computer for development and testing. The development server can also generate configuration for datastore indexes based on the queries the app performs during testing.

6 System Design

This section is very important for us as a application developer; here strong OOP concepts have been implemented to design such a prototype for our eLC model. For our model we created a very simple class design to show service development in Google cloud. We design our system with a very simple design to test how to run a simple application for e-learning in cloud and manage the service. In proposed system we have used PaaS (platform as service model) and design it that with Oracle Fusion Middleware that provides us a comprehensive foundation for e-learning private clouds. With mature, reliable, high-performance clustering mechanisms as well as pervasive, unified security and management, Fusion Middleware delivers the most effective dynamic resourcing in the industry. Modularity, sharability, and composability enabled by Fusion Middleware SOA, BPM, and user interaction technologies complement the dynamic resourcing to support a powerful self-service platform of reusable components. These platform components can easily managed by a central IT function and easily composed into applications by departments within the enterprise. The user will able to connect with central server using any client device having browser software only.

7 Model Application implementation

The figure below shows the basic architecture of a platform-as-a-service cloud design for e-Learning model it will offer a central IT function to set up within a university.
Once the basic platform is up and running, the application owners within the enterprise’s departments can set up their respective applications. Again, depending on the nature of the domain and enterprise, this may involve fairly simple application composition using platform components, or it may involve a substantial amount of custom application development. Once an application has been deployed on the platform, the third step is simply the use of the application. From the users’ perspective, the application is no different from any other network/Web-based application they would use within the enterprise—there is nothing special about the fact that it’s running on a cloud platform as far as they are concerned. Finally, there is of course ongoing administration of the platform as well as the applications, which is carried out by the central IT function. Depending on the nature of the applications, the application owners may also carry out some amount of administration, such as adding and removing users or other high-level functions specific to the application. Central IT is concerned with lower-level issues such as whether the application is resourced appropriately, is meeting its service-level agreements (SLAs), etc. One of the goals in setting up shared infrastructure in general and private cloud in particular is to exploit as many economies of scale and opportunities for efficiency as possible. Among these is the opportunity to automate dynamic resource allocation and optimization, enabling the elastic capacity that characterizes cloud. This also enables the continuously high responsiveness demanded by users irrespective of load and minimizes manual intervention.

8 Conclusion

By creating eLC, we propose a Software Development Platform for future e-Learning systems based on cloud technology. It enables clients (mainly students and trainers) to specify and design their task hosted as a service through distributed Cloud which is interfaced with Web services technology. Since

Shahbaz Zafar: Computational Cloud framework implementation to support scalable and cost efficient services for e-Learning systems
the eLC cloud is loosely coupled with that of View Cloud, the View pattern can be outsourced to provide a universal structure for a high variety of end users. The presented model can significantly improve teacher-student collaborations through adaptation of devices to a variety of end-users. In the traditional e-learning ecosystem, physical machines are usually simply and exclusively stacked, and most resources are deployed and assigned for some specific tasks. Moreover, the utilization of those resources becomes urgent problem. In our proposed solution we argued for a new e-learning system based on Cloud computing infrastructure. We proved that Cloud computing based e-learning system is reliable, flexible, cost-efficient, self-regulated, and QoS-guaranteed. It has some mechanisms to guarantee the teaching and learning activities, the quality and the running of the system. Finally, our research results answers the main research question of the study: —What can be the main issues in use cloud computing based e-learning systems and how can we made a cloud based e-learning system and we propose our new model named eLC.

9 References


[10] Nilekani takes charge, TNN, Jul 24, 2009, 05.06am IST, http://timesofindia.indiatimes.com/articleshow/4812763.cms, as on 12.30 pm, 05/07/07


