

CcMS: A Cloud Computing Based Content Management System

Diptoneel Kayal

*School of Computer Science & Engineering,
West Bengal University of Technology, India.*

Abstract

The amount of digital content is increasing at a rapid rate. These contents are being accessed using a wide range heterogeneous set of devices. Managing all these data is becoming difficult and time consuming without an efficient and robust content management system. In this paper a cloud computing based content management system, CcMS is proposed which is capable of handling all the issues regarding the management of large amount of data effectively, while provides an easy to use interface to the user with different platform. As the proposed system is based on cloud computing, it has the capability of abstracting the physical architecture and implementation procedure from the user and provides better data security mechanism than that of the traditional content management system. CcMS uses representational state transfer (REST) architecture which provides better support, specially for distributed system and increases the capability and flexibility of the proposed system.

Keyword: REST, Cloud computing, XML, CMS, HTTP.

1. Introduction

Today digital data is an integral part of our day to day life and plays a major role in making our life more comfortable and secure. But the amount of data is so large that it is increasing becoming difficult to manage. Incorrect method of handling may results in data loss. Some of the data are very sensitive in terms of sensitivity and criticality. The challenge becomes higher when the users having a heterogeneous set of devices ranging from lightweight mobile devices, having very limited resources to

sophisticated multiprocessor servers. Using a robust content management system these challenges could be handled efficiently. A content management system allows organizations to collaboratively create, edit, manage and publish any type of digital information such as text, images, video, sound, documents, etc. This kind of systems is characterized by the need for efficient storage and retrieval of high volume of content under strict rules controlling the sharing of information among users and organizations.

In this paper we have identified various challenges regarding the management of contents and proposed a content management system, namely 'Cloud based content Management System' (CcMS) which is capable of handling a heterogeneous set of user devices in a distributed environment. The proposed system is designed based on the cloud computing paradigm. Cloud computing is one of the most rapid growing area of Information Technology. Cloud computing can be considered as a virtual pool of various computing resources such as storage device, CPU etc to meet user's requirements as and when needed on pay-per-use basis. Cloud computing is based on five attributes: shared resources, massive scalability, elasticity, pay as you go and self-provisioning of resources. Unlike previous computing models, which is assumed dedicated resources, cloud computing is based on a business model in which resources are shared at network level, host level and application level. Cloud computing allows the users to increase or decrease their computing resources as and when needed. Interest in the cloud computing is growing because cloud solutions helps business organizations to decrease the cost of computing resources significantly. There are three deployment models are available for cloud computing. They are (1) public cloud, (2) private cloud and (3) hybrid cloud. A public cloud is hosted, operated and managed by a third-party vendor from one or more data centers. Normally in a private cloud model the day-to-day operations including the security management are handled by the implementing organization itself or by some third party contractual SLAs. A hybrid cloud environment consisting of multiple internal and/or external providers is a possible deployment for the organizations. In a hybrid cloud environment organizations might run non-core applications in a public cloud, while maintaining core applications and sensitive data using a private cloud. In section II we have provided all technical details of CcMS alongwith the characteristics a content management system should possess, in section III implementation details of CcMS has been provided and section IV deals with the conclusion and references.

2. Technical Details of CcMS

Managing digital content efficiently usually requires special solutions which depend on the way that the content is used. Moreover, the nature of the content as well as the environment of use are related to the way that the content needs to be managed. Because of the complex and multidimensional nature of managing content, no general guidelines exist. According to [1] content can be any kind of audiovisual, visual, sound, or textual information. In this paper we have used terminology defined by Society of Motion Picture and Television Engineers (SMPTE) and the European

Broadcasting Union (EBU). According to this terminology content consists of the metadata and the essence. Metadata is the descriptive information about the essence and its representations. It can be related to content, location or media/material [1]. Essence is the core part of the content [1]. In order to develop a robust content management system the following essential characteristics a content management system should possess.

1. From the end user and content interaction perspective, the content needs to be stored to a location that is safe and where it can be accessed regardless of time or place [2].
2. Searching for the content and navigating the results should be easy and intuitive [2].
3. A distributed content management system needs to monitor where content is referenced [3].
4. As much metadata as possible needs to be extracted and produced from the content and attached to it after the creation of the content. [4]
5. The system needs to support describing the content with heterogeneous and arbitrary metadata [5].
6. In a heterogeneous environment, content management system needs to offer a uniform interface for all the resources [3].
7. A distributed content management system in a heterogeneous network needs to offer an efficient way to access the content with user specific settings and configuration [3]

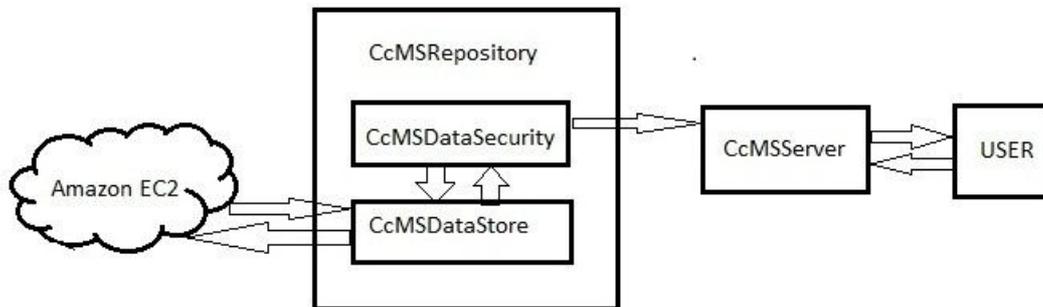


Figure 1: Architecture of CcMS.

For developing of CcMS, representational state transfer (REST) is used which an architectural style for building network based software as described in [6]. REST is utilized in the context of web applications. In this context REST becomes a resource oriented architecture where resources are exposed by servers and consumed by the clients using HTTP methods. A resource is accessed via its URL and its state is transferred using its representation. A key characteristic of a RESTful interface is the

clear division of application state between the client and the server. Three major properties of REST are the use of layered client-server style, the uniform interface and the possibility to negotiate a suitable representation for a resource. The first property allows the introduction of intermediaries such as caches and gateways. The second permits a wide range of clients to utilize the system. Finally the negotiable representations allow humans and programmed clients to simultaneously use the system.

XML, XLSST and Extensible Messaging and Presence Protocol (XMPP) have also been used in CcMS along with REST. XML and XLSST are used for data storage, browsing, data manipulation and presentation purpose, while XMPP is a protocol designed for near-real-time and request response services streaming XML messages. XMPP and its extensions provide support for establishing presence, authentication, one-to-one and multi-party messages and notifications. These services have been used to build systems for instant messaging, systems control, and lightweight middleware [7].

CcMS facilitates users' access to the essence and the metadata of their content items that are distributed across many devices. The users can register their desktop, laptop and mobile devices to CcMS and access their content from a single entry point, the CcMS server. The server provides a web interface and Atom Feeds [12] for viewing the content. It also keeps track of different versions of the content. New versions are automatically created when content is updated, so content is never lost in CcMS. In CcMS a CcMSServer is used as an agent in between the users and CcMSRepository. CcMSRepository is used for storing and maintenance of data. It also provides proper security to the data. CcMSRepository is divided in two parts, (1) CcMSDataStore and (2) CcMSDataSecurity. CcMSDataStore component is responsible for storing and backup of data. This component uses Amazon EC2 cloud computing services for storage and backup of data. For automatic data backup CcMSDataStore component has an interface which is used to communicate directly with Amazon EC2. The backup operation is done at an instance defined by the user. This paper uses REST interface operation for the communication purpose using HTTP POST and PUT method. CcMSDataSecurity component is used to handle the security mechanisms and user authentication of the data. For the simplicity of the system only username-password based security mechanism is used, although implementation of bio-metric authentication is also possible.

CcMSServer is one of the key components of the proposed system. It acts as an intermediary in between the users and CcMSRepository. CcMSServer performs another important job to serve a number of users at a time. When users want to use CcMS, he/she first have to connect CcMSServer, which will authenticate him/her with the help of CcMSDataSecurity component. Once the user is authenticated and identified CcMSServer provides the user a web interface, by using which the user can use the system.

3. Implementation of CcMS

CcMS is based on private cloud computing architecture. As per the implementation is concerned all the major components of CcMS, i.e. CcMSServer, CcMSDataStore and CcMSDataSecurity, all are implemented and maintained by the user. For only backup process Amazon EC2 is used. It virtualizes the resources for each and every user as per their requirements. For the implementation of CcMS, Ubuntu 12.04 LTS for Cloud, Apache 2 web server and Virtualbox 4.1.22 (for Linux) is used. VirtualBox for Linux (Ubuntu 12.04 LTS version) is used to achieve virtualization of resources. To deploy cloud structure we have used Xen as it is included with Ubuntu Server 12.04 LTS.

4. Conclusion

In this paper implementation and design of a low cost, cloud based content management system is proposed. This technique is suitable for any organizations who wish to implement a private cloud computing based content management system. We have used only open source software to keep the cost low as much as we can without compromising the quality of the implementation. All the tools used to develop the proposed system can be downloaded from their website. The system has good flexibility and reliability. In future it can further be expanded to incorporate automatic process for resource allocation and more secure user authentication method.

References

- [1] A. Mauthe and P. Thomas, Professional content management systems: handling digital media assets. West Sussex, England: John Wiley & Sons, 2004.
- [2] A. Spedalieri, A. Sinfreu, G. Sisto, P. Cesar, I. Vaishnavi, and D. Melpignano, "Multimedia content management support in next generation service platforms," in Proceedings of the 3rd international conference on Mobile multimedia communications, ser. MobiMedia '07. ICST, Brussels, Belgium: ICST, 2007, pp. 14:1–14:7.
- [3] C. D. Cranor, R. Ethington, A. Sehgal, D. Shur, C. Sreenan, and J. E. van der Merwe, "Design and implementation of a distributed content management system," in Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video, ser. NOSSDAV '03. New York, NY, USA: ACM, 2003, pp. 4–11.
- [4] K. Curtis, W. P. Foster, and F. Stentiford, "Metadata – the key to content management services," in Proceedings of the Meta-Data'99, ser. The Third IEEE Meta-Data Conference, Bethesda, Maryland, USA, 1999.

- [5] P. Bolettieri, F. Falchi, C. Gennaro, and F. Rabitti, "Automatic metadata extraction and indexing for reusing e-learning multimedia objects," in Workshop on multimedia information retrieval on The many faces of multimedia semantics, ser. MS '07. New York, NY, USA: ACM, 2007, pp. 21–28.
- [6] R. T. Fielding, "REST: architectural styles and the design of network-based software architectures," Doctoral dissertation, University of California, Irvine, 2000.
- [7] P. Saint-Andre, K. Smith, and R. TronCon, XMPP: The Definitive Guide: Building Real-Time Applications with Jabber Technologies, 1st ed. Sebastopol, CA, US: O'Reilly Media, 2009.
- [8] W. Dawoud. I. Takouna, C. Meinel, "Infrastructures as a service security: Challenges and solutions" 7th International Conference on Informatics and Systems, 2010, pp. 1-8.
- [9] Zhang L., Zhou Q., "CCOA: Cloud Computing open Architecture", IEEE international Conference on Web Services, 2009, pp 607-616.