A CLOUD-BASED FRAMEWORK FOR ONLINE MANAGEMENT OF MASSIVE BIMS USING HADOOP AND WEBGL

*Hung-Ming Chen, Chuan-Chien Hou, and Tsung-Hsi Lin
Department of Construction Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
*(*Corresponding author: hungming@mail.ntust.edu.tw)*

A CLOUD-BASED FRAMEWORK FOR ONLINE MANAGEMENT OF MASSIVE BIMS USING HADOOP AND WEBGL

ABSTRACT

This paper presents a Cloud-based system framework for providing a web-based building information model (BIM) service. Cloud computing technologies were used to develop a BIM data center, which can handle the big data of massive BIMs using multiple servers in a distributed manner. In addition, the system allows multiple users to access the BIM data center online to submit and review BIMs concurrently. In this study, the advanced technologies/tools of web development and Cloud computing were applied and incorporated to develop a prototype system, named CloudBIM. On the client side, this study applied WebGL and HTML5 in the web-based interface development to achieve the 3D display of BIMs and the dynamic interaction with viewers through browsers. Users can access the services of CloudBIM by using PCs, laptops and tablets anytime anywhere to view the 3D model via browsers, which support WebGL. On the server side, this study used Apache Hadoop, an open-sourced distributed file system based on Google's technologies, to establish a BIM data center. The Hadoop Distributed File System (HDFS) can utilize multiple servers to provide mass storage spaces in a distributed manner. Based on the use of Apache HBase, which provides Bigtable-like structured storage for HDFS, this study proposes database schema for storing the information from massive BIMs in two bigtables, named Object table and Property table respectively. In addition, Hadoop MapReduce, a Hadoop component for the parallel processing of large data sets, is utilized for model data acquisition by parsing the IFC files of uploaded models in parallel.

KEYWORDS

Building Information Model, Cloud Computing, WebGL, Hadoop, IFC

INTRODUCTION

Most current commercial BIM software is a stand-alone system, where the majority of the computation is executed on a single computer. As the requirements of storage space and computing resources continues to rise, it becomes harder for a stand-alone personal PC to handle the operation, thus, the efficiency of management, sharing and synchronization of BIMs is affected. In addition, there are many vendors of BIM software, however, not only can BIMs constructed using software developed by different vendors not be shared conveniently, but even BIM software developed by the same vendor can experience a loss of information when using different versions. As a result, as the number of BIMs increases, the task of managing many different versions of BIMs becomes more difficult. Although some BIM software vendors have developed Viewers for viewing different versions of BIMs, managing BIMs in many projects in an integrated manner remains a challenge. In view of the fact that a stand-alone system is overburdened by running the BIM software, the difficulty in managing the BIMs of multiple projects and an unfriendly information-sharing environment for multiple users who are cooperating in the same project, a BIM system developed using new technologies is presented in this study. Cloud computing technology offers a solution to the limitations of system resources. This technology is employed in this study to develop a web-based BIM service. The proposed web-based BIM service does not rely on high performance hardware facilities, and can be utilized by multiple users concurrently.

In recent years, Cloud computing technology has gradually advanced, and has been applied in various fields (Calyam et al., 2011; Spillane et al., 2011; Srirama et al., 2012). The definition of the system framework of cloud computing, according to the National Institute of Standards and Technology (NIST), contains five essential characteristics, four deployment models, and three service models (Mell & Grance, 2011).

The five essential characteristics are:

1. On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
2. Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms.
3. Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model.
4. Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand.
5. Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service.

The four deployment models are:

1. Public cloud: The cloud infrastructure is provisioned for open use by the general public.
2. Private cloud: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers.
3. Community cloud: The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns.
4. Hybrid cloud: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public).

The three service models are:

1. Software as a Service (SaaS): The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure.
2. Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.
3. Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.

When stand-alone software or a system is combined with Cloud technology, a network that provides a shared service for multiple users is formed. The service provider of this network has the capacity to store big data and conduct big data analysis. Therefore, such a combination enables online sharing of BIMs, and the increased storage space for big data and computing capacity for big data analysis facilitates storage and management of massive BIMs. BIM services in cloud computing models must comply with the five essential characteristics; one of the four deployment models and one of the three service models must be selected in order to build the web-based BIM system using cloud computing technology. The system developed in this study to provide a web-based BIM service features the five essential characteristics, a private cloud model, and SaaS.

The proposed Cloud-based system for providing a web-based BIM service is named CloudBIM. Among the service models defined in cloud computing, it employs SaaS. At the design planning stage, engineering consultants or architects and associates complete the BIM using a stand-alone commercial BIM software. They can then upload the BIM to CloudBIM and publish it for online sharing. The building information in the BIM that has been uploaded to CloudBIM is analyzed. Those data that contain 3D components of the BIM and their attribute data will be distributedly stored in the big data table. In such a system, subcontractors to engineering consultants or architects and associates, construction firms and development companies can easily view information of the BIM via browsers using PCs, laptops, or tablets connected to the internet. This system can also be applied to public construction bidding. The BIM designed by the bidder can be submitted to CloudBIM and reviewed by the authorities.

The proposed CloudBIM can provide real-time web-based services to multiple users concurrently, enable storage of big data, and perform analysis and computing. It solves the problems faced in the use of a stand-alone BIM system, where users cannot use the system simultaneously and substantial computing resources are required. There are two major requirements for realizing such a web-based multi-user BIM service. First, with regard to the model presentation, the system should display all online models in a 3D visualized way and allow user to access the properties of model components using any online devices. Second, with regard to the system performance, the massive BIMs accumulated in the "BIM library" on the server side should have the ability to be accessed concurrently for reviewing by multiple users. In this study, the following technologies are integrated to develop the CloudBIM system that fulfills these two requirements. (1)Apache Hadoop: a distributed file system with Cloud framework. The Hadoop Distributed File System (HDFS) can provide storage and backup of big data. Hadoop MapReduce can perform big data analysis. Such a distributed system features high availability and file load balancing. (2)Apache HBase: a structured database in NoSQL, that is attached to HDFS. (3)WebGL: a 3D display technology under HTML5 framework to achieve dynamic interaction with internet users via browsers without any plug-in. Existing BIM servers include Autodesk Revit BIM Server, Graphisoft ArchiCAD BIM Server and Bentley ProjectWise Integration Server. These BIM servers run on a single computer to provide BIM file storage service. To view a BIM on these servers, the BIM file must be downloaded in whole to the client side and then opened using specific stand-alone BIM software. Unlike these existing BIM servers, the proposed CloudBIM retrieves the information from massive BIMs and distributedly stores them in a structured NoSQL database using multiple servers. In addition, an online BIM can be viewed on the client side via standard browser, and only the requested information are transmitted while viewing.

SYSTEM FRAMEWORK

The framework of the proposed CloudBIM system is shown in Figure 1. Users on the client side can view the 3D model of a BIM via browsers using HTML5 and WebGL technologies. The information is provided by Cloud platform and HBase database on the server side. Users can also upload BIMs in IFC files onto the system, and once processed by MapReduce, they will be stored in HBase for other users on the client side to view and use.

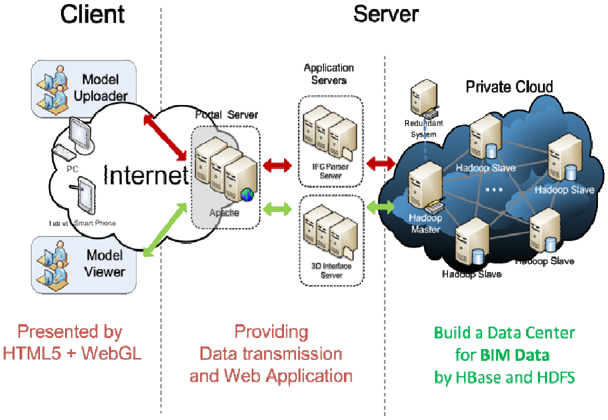


Figure 1 – System framework of the CloudBIM

In the proposed system framework, users can be divided into two types: model uploaders and model viewers. The two types of user are described in detail below:

1. Model uploaders: users connect to the system login server using internet access devices. IFC files of uploaded BIMs are parsed and transferred to the private cloud. These IFC files are processed in the private cloud so that uploaded BIMs in IFC file formats are converted to 3D component formats that can be read by a WebGL module on the client side. In addition, attributes all BIM components in the IFC files are extracted and stored in the private cloud.

2. Model viewers: after login into the server, these users select the BIM that they want to view. Component data of the BIM is obtained from the private cloud and presented to the model viewers, through the 3D Interface server driven by WebGL. Viewers can interact with the displayed 3D model, using the mouse to adjust the view and change viewing angles, using the keyboard to move the viewpoint, and clicking the mouse to select components and see their attributes.

The proposed system framework allows users to share the BIM with others. Similar to YouTube, where users upload edited videos through simple steps and share them with other internet users; in the web-based CloudBIM system, users upload a BIM onto the cloud server following simple procedures, and the information is stored in cloud database, HBase, where it can be viewed by many other users.

SYSTEM OPERATING MECHANISMS

The technologies integrated to build a CloudBIM system are demonstrated in Figure 2. The structure has three levels: (1) Web UI layer: provides users on the client side with a web-based BIM system service. Its main function is to display the BIM in 3D. (2) Data access and control layer: facilitates communication between the Web UI layer on the client side and the cloud service layer by employing software suites and programs. The query of BIM component attributes and IFC-based BIM uploading are all carried out on this layer. (3) Cloud service layer: provides storage of big data, performs big data analysis and computing, and achieves data locality and backup. The data storage space can be expanded horizontally and provides service using multiple computers. Regarding system operation, the CloudBIM system are mainly based on the following modules: BIM uploading and analysis, Apache Thrift data connection, BIM 3D viewing, and database schema for BIM data storage. The operating mechanisms of these modules are introduced in detail below.

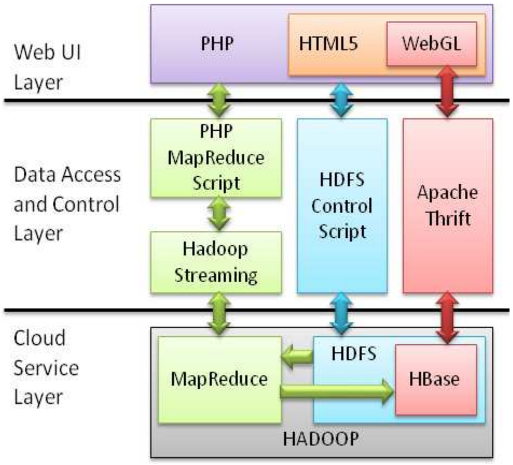


Figure 2 – Technology integration framework of CloudBIM

BIM uploading and analysis

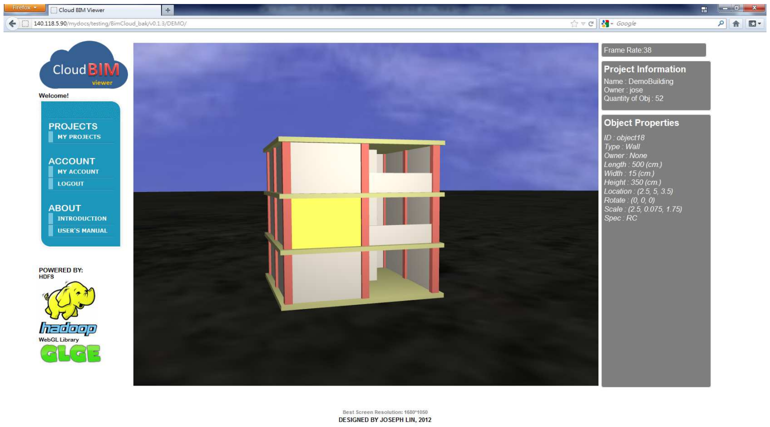
This module receives BIMs in IFC files uploaded by users and determines whether they are in IFC format. It then passes the IFC files to the IFC pre-processing module. IFC files are in hierarchical data format, and therefore cannot be separated. The pre-processing module converts the IFC files to a format so that they can be partitioned while data for each single component still remains intact. By using partitioned files, problems relating to referencing other files are eliminated. This allows each node to obtain its required data and perform the analysis on BIM components. MapReduce is employed to conduct IFC computing and analysis. It is based on the distributed computing principle proposed by Google Inc. Files are partitioned, distributed, and uploaded to a distributed file system. Files are distributed to several server nodes where they are computed in parallel; this is called Map computing. Computed results from each node are then transferred to a certain specified node for further computing; this is called Reduce computing. Both Map and Reduce computing can be defined according to various algorithms as specified by developers. After the IFC files are converted to formats that can be processed by MapReduce, JobTracker assigns each TaskTracker to parse the files into a format that the BIM 3D viewing module on the client side can read. The results are passed back to Reducer. Reducer puts the received results in order, numbers every set of data, and saves them in HBase cloud database. Through a structured index, requested data can be located quickly in HBase. The data stored in HBase can be retrieved and displayed directly in the WebGL 3D model display frame on the user's browser.

Apache Thrift data connection

Apache Thrift serves as a connecting tool for the HBase database on the server side and hypertext preprocessor (PHP) on the client side. Thrift is a communication software suite that attaches to HBase, PHP, and Apache HTTP Server. PHP can access everything in HBase using Thrift. Three programs in the CloudBIM need to use Thrift: reducer.php, writes processed component information of uploaded BIMs into the data table in HBase via PHP; display.php, as WebGL starts, all components of the BIM are extracted from HBase via PHP; ajaxlink.php, retrieves component attributes from the property table when a component is selected by a user.

BIM 3D viewing

The client side of the system utilizes WebGL technology to display BIM in 3D on browser, as shown in Figure 3. The significance of 3D visualization lies in the viewer interaction. WebGL offers many application programming interfaces (APIs) that enable 3D modeling objects for viewer interaction. Model data need to be imported all at once at the initiation of WebGL. In this study, BIMs are saved in HBase. PHP and Thrift are employed to import components of a BIM in HBase into WebGL. JavaScript is used to engage keyboard and mouse events. When a component in BIM are selected, its attributes in HBase are retrieved and sent back to WebGL via AJAX in real time, thus achieving the required interaction between users and the display. WebGL technology uses the "canvas" element in HTML5. This interface element enables a 3D component to be displayed directly on the browser; no browser plug-ins are required and Model viewers can see the BIM online via any browser that supports WebGL.



Frame Rate:38

Project Information
Name : DemoBuilding
Owner : jose
Quantity of Obj : 52

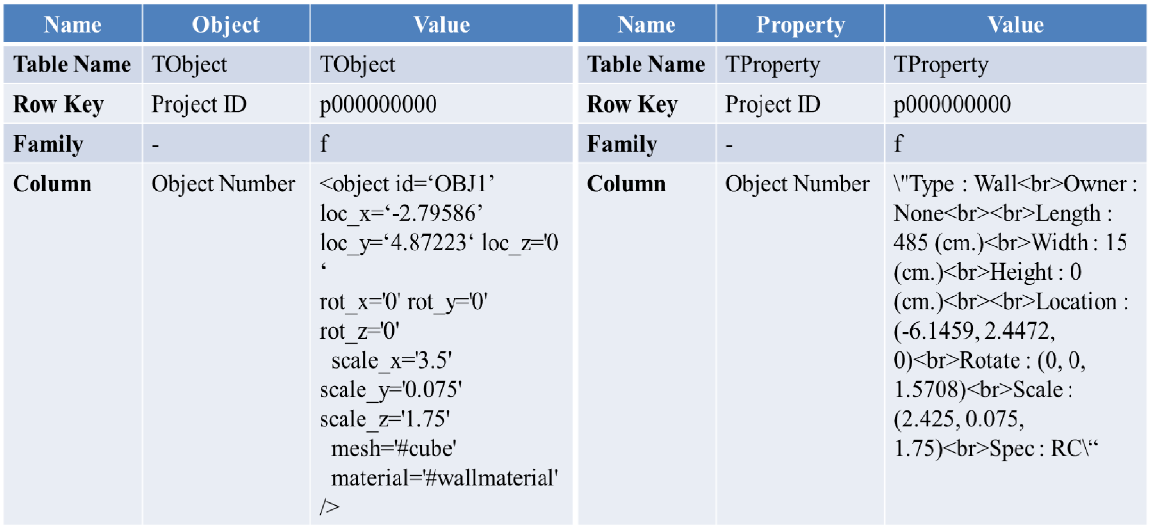
Object Properties
ID : object18
Type : Wall
Owner : None
Length : 500 (cm.)
Width : 15 (cm.)
Height : 350 (cm.)
Location : (2.5, 5, 3.5)
Rotate : (0, 0, 0)
Scale : (2.5, 0.075, 1.75)
Spec : RC

Best Screen Resolution: 1680*1050
DESIGNED BY JOSEPH LIN, 2012

Figure 3 – BIM 3D online viewing in the WebGL interface

Database schema for BIM data storage

Apache HBase serves as a database that stores BIM 3D components and attribute data. Two main data tables are constructed. One is the object table that contains components of the 3D model, and the other is the property table that contains component attributes. As can be seen in Figure 4, in the object table, each line of data is for one component; this table includes the component ID, its position, rotation angle, scaling ratio, basic object and the material of the component. Each line in the property table contains attribute details of a component. The design concepts for both tables are identical. They are divided into two data tables for serving different modules. Data for one component is placed in a column. When the data is needed, the whole column is read. This improves the efficiency of data retrieval.



Name	Object	Value	Name	Property	Value
Table Name	TObject	TObject	**Table Name**	TProperty	TProperty
Row Key	Project ID	p000000000	**Row Key**	Project ID	p000000000
Family	-	f	**Family**	-	f
Column	Object Number	<object id=‘OBJ1’ loc_x=‘-2.79586’ loc_y=‘4.87223‘ loc_z='0 ‘ rot_x='0' rot_y='0' rot_z='0' scale_x='3.5' scale_y='0.075' scale_z='1.75' mesh='#cube' material='#wallmaterial' />	**Column**	Object Number	\"Type : Wall Owner : None Length : 485 (cm.) Width : 15 (cm.) Height : 0 (cm.) Location : (-6.1459, 2.4472, 0) Rotate : (0, 0, 1.5708) Scale : (2.425, 0.075, 1.75) Spec : RC\"

Figure 4 – Schema of Object Table and Property Table

SYSTEM VALIDATION

Validation of the developed prototype system is conducted on eight Intel Core i5 computers, each with 16 gigabyte (GB) memory and 500 GB hard drive. Linux Ubuntu 12.04 is used as the operating system. Two computers among the eight serve as login servers. Both of them are equipped with a BIM uploading module and WebGL 3D viewing module. All the computers form a small private cloud. One of them acts as Hadoop Namenode and JobTracker, and another is the HBase HMaster. The remaining computers act as DataNode and TaskTracker for cloud storage and computing.

Multiple users can view the BIMs online concurrently using the proposed system. One or more BIMs can also be viewed at the same time. The 3D viewing module uses WebGL technology. Its performance in presenting the data primarily depends on users' display devices. The data retrieved from the server side are processed BIM component data. When users start the WebGL display interface, the system will retrieve all components from the BIM for model display. Attribute data are obtained separately on request. In addition to viewing by multiple users, the system also supports uploading by multiple users concurrently. The upper limit on the memory is set as 400 megabytes. Currently, the maximum number of BIMs in the private cloud database is more than 500. Each BIM on average contains more than 50 component models and corresponding attributes. Even with this much data, the system still runs efficiently. The cloud-based system enables users to use the service and view BIM components via different types of internet access devices. The deployment and capacity of CloudBIM in this validation test are summarized in Table 1.

Table 1 – The attributes of the validation test on CloudBIM

Item	Deployment / Capacity
Server Hardware	8 Intel Core i5 computers, each with 16 GB memory and 500 GB hard drive
Operating System	Linux Ubuntu 12.04
The number of Login Server	2
The number of Hadoop NameNode and JobTracker	1
The number of HBase HMaster	1
The number of Hadoop DataNode	6
The limit setting of memory	400 MB
The maximum number of BIMs in the private cloud database	More than 500 BIMs (Each BIM on average contains more than 50 component models and corresponding attributes)

CONCLUSIONS

A web-based BIM system that utilizes Cloud technologies is presented in this study. The storage and viewing of BIMs becomes a Cloud service with SaaS as its service model. The proposed system framework can provide online services that allow developers of BIMs to upload BIMs onto the server and share them with others involved in the project. It can be seen as a BIM reviewing platform that facilitates collaboration. Advanced technologies are utilized to develop a prototype system. WebGL technology is employed to display BIMs in 3D on the browser. Database schema for storing BIMs in the cloud database of the web-based BIM system is developed. MapReduce technology is applied to develop an IFC Parser which processes BIM data and saves them in the cloud database. The cloud database and WebGL interface are integrated using Thrift. Validation of the prototype system shows that 3D BIMs can be viewed via various types of internet access devices, cloud database can store massive BIMs, and the system can provide access to BIMs for multiple users concurrently.

ACKNOWLEDGMENTS

This research was supported by the National Science Council of Taiwan under Project Number NSC 101-2221-E-011-068.

REFERENCES

Calyam, P., Patali, R., Berryman, A., Lai, A.M., Ramnath, R. (2011). Utility-directed resource allocation in virtual desktop clouds. *Computer Networks*, 55(18), 4112-4130. doi: 10.1016/j.comnet.2011.07.019

Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing* (NIST Special Publication 800-145). Gaithersburg, MD: National Institute of Standards and Technology.

Spillane, R.P., Shetty, P.J., Zadok, E., Dixit, S., & Archak, S. (2011). An efficient multi-tier Tablet Server Storage architecture. *Proceedings of the 2nd ACM Symposium on Cloud Computing, SOCC 2011*. doi: 10.1145/2038916.2038917

Srirama, S.N., Jakovits, P., Vainikko, E. (2012). Adapting scientific computing problems to clouds using MapReduce. *Future Generation Computer Systems*, 28(1), 184-192. doi: 10.1016/j.future.2011.05.025