



Implementation of 3-Tier Architecture Using C-MART

M.Poovizhi¹, S.Kalpana²

B.Tech, ME (CSE) Part Time Pursuing Department of CSE, Maharaja Engineering College, Avinashi, Tamil Nadu, India¹

Asst. Professor, Department of CSE, Maharaja Engineering College, Avinashi, Tamil Nadu, India²

ABSTRACT: Server environment such as Cloud computing provides resource provisioning; it allows applications to evaluate the performance. Benchmarks are not designed for calculating performance of cloud computing environments. This results in misleading results and produces violations of quality-of-service (QoS) when systems tested using these benchmarks are deployed in production environments. So we present C-MART, new web application benchmark for the Cloud. It is designed using the cloud computing paradigm of elastic scalability at every application tier and utilizes modern web-based technologies such as HTML5, AJAX, jQuery, and SQLite. The design principles of C-MART consist of a client emulator, deployment server, and scaling Application Programming Interface. The API (Application Programming Interface) used to provide interface between the client and the systems. C-MART can horizontally scale at every tier and run with different architectures, applications and therefore C-MART is a platform independent architecture also. The deployment server automatically deploys and configures the test environment in orders of magnitude less time than current benchmarks. The deployment server stores the information of the web hosting websites also. The deployment server is present in the server environment. The scaling API allows users to define and provision their own customized datacenter. The client emulator generates the web workload for the application by emulating complex and varied client behaviors, including decisions based on page content and prior history. We show that C-MART implementing in 3-tier architecture can detect problems in management systems that previous benchmarks fail to identify, such as an increase from 4.4 to 50 percent error in predicting server CPU utilization and resource under provisioning in 22 percent of QoS measurements. Even though C-MART is multi-tier and it is implemented in 2-tier in the recent years. Therefore in this project we are going to implement C-MART under 3-tier architecture. Middleware's such as RMI, CORBA involves in the 3-tier architecture with video on demand.

I INTRODUCTION

Cloud computing is a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. In cloud computing, the word cloud (also phrased as "the cloud") is used as a metaphor for "the Internet," so the phrase cloud computing means "a type of Internet-based computing," where different services such as servers, storage and applications are delivered to an organization's computers and devices through the Internet. Applications are essential for performance testing and validating systems before deployment to production environments. They emulate a typical production system and provide measurements of a secondary system under test (SUT) such as the processing power of a host, the maximum number of serviceable clients, or the efficiency of a resource allocation algorithm.

They allow various configurations, settings, parameters and so on and to be compared so that the best values for a given scenario can be identified. Benchmark web applications typically consists of an example website, such as an online store or a social networking website and a workload generator that sends traffic to the website in a manner emulating web browsing client. Existing benchmarks were not designed for cloud computing environments and therefore produce misleading results when benchmarking cloud management systems. This can result in poor performance, resource under provisioning and quality-of-service (QoS) violations. C-MART, a web application benchmark designed to evaluate systems running in cloud computing environments

1.1 C-MART ARCHITECTURE

C-MART is designed as an online auction and shopping website. It is a multitier application written using the model-view-controller design principle. It can be implemented by servlets. It can be implemented using Java servlets for the application logic due to Java's prevalence in production applications. C-MART includes a custom built workload generator that emulates clients accessing the website. A deployment and scaling API is provided to simplify the management of C-MART. C-MART was created with four main design principles to ensure its suitability for use as a cloud computing benchmark application which are scalability, modern technologies, flexibility, and client realism. These principles ensure that the designs and technologies used mimic the behavior of real world applications.

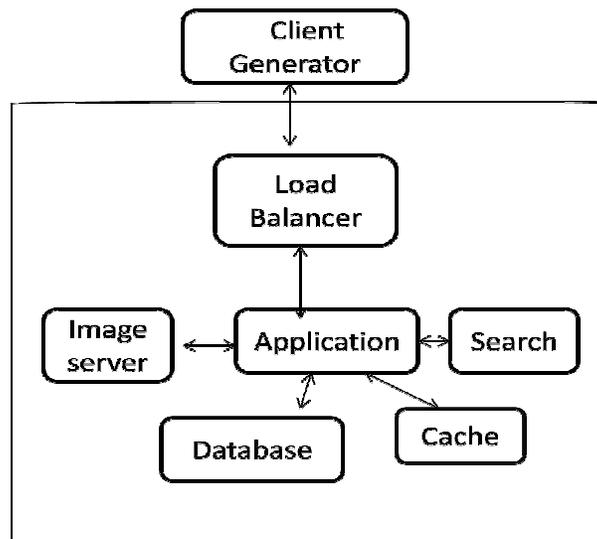


Fig.1. C-MART ARCHITECTURE

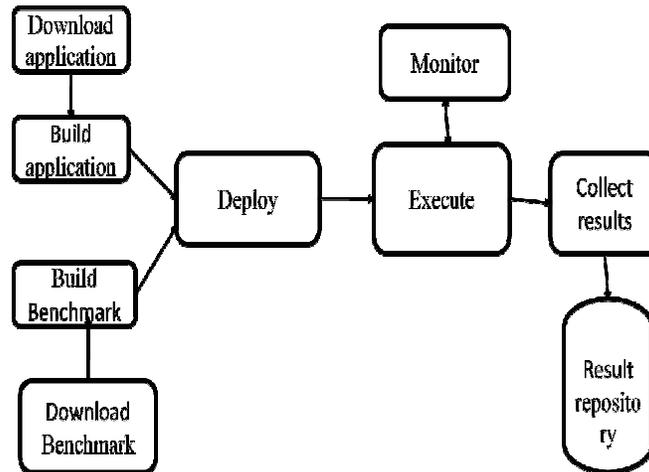


Fig.2. STEPS FOR RUNNING BENCHMARK

II RELATED WORKS

The purpose of a cloud benchmark is to help developers determine the right architecture, services, and settings for their applications by providing a testing platform that delivers relevant and comparable measurements. This section summarizes existing benchmarks and highlights some of the deficiencies that make them unsuitable for use as benchmarks for cloud management systems

2.1 RUBiS[4]

RUBiS is an auction site prototype modeled after eBay.com that is used to evaluate application design patterns and application server's performance scalability. Our auction site benchmark implements the core functionality of an auction site is selling, browsing and bidding. This benchmark does not implement complementary services like instant messaging or newsgroups. This includes three kinds of user sessions and they are visitor, buyer, and seller. For a visitor session, users need not register but are only allowed to browse. Buyer and seller sessions require registration. In addition to the functionality provided during visitor sessions, during a buyer session users can bid on items and consult a summary of their current bids, rating and comments left by other users. Seller sessions require a fee before a user is allowed to put up an item for sale. An auction starts immediately and lasts typically for no more than a week.

2.2 TPC-W[7]

TPC Benchmark is a transactional web benchmark. The workload is performed in a controlled internet commerce environment that simulates the activities of a business oriented transactional web server. The simultaneous execution of multiple transaction types that span a breadth of complexity, On-line transaction execution modes. Databases consisting of many tables with a wide variety of sizes, attributes, and relationships. The performance metric reported by TPC-W is the number of web interactions processed per second. Multiple web interactions are used to simulate the activity of a retail store, and each interaction is subject to a response



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

time constraint. The store size is chosen from among a set of given scale factors, which is the number of items in inventory and varies from 1,000 items to 10,000,000 items. The performance metric for this benchmark is expressed in Web Interactions Per Second (WIPS) at a tested scale factor expressed by WIPS @scale factor where scale factor is the number of items in the item table.

2.3 Cloud stone Olio

Cloudstone's Olio is an open-source social-event calendar web application with web 2.0 features. It utilizes some modern technologies such as AJAX and Memcache. However, the application only has a limited number of functions, uses a single think time distribution, a static probability matrix for page transitions, and does not capture or emulate SQLite. The database tier uses MySQL or PostgreSQL, and is scalable through read-only replicas, rather than the NoSQL approach that is increasingly popular in cloud environments.

III C-MART

C-MART is designed as an online auction and shopping website. It is a multitier application written using the model-view-controller design principle. We use java servlets for the application logic due to java's prevalence in production applications. C-MART includes a custom built workload generator that emulates clients accessing the website. A deployment and scaling API is provided to simplify the management of C-MART. C-MART was created with four design principles and that they are scalability, modern technologies, flexibility and client realism. Client realism is used to identify whether the information's are coming from the good source.

3.1 TIER SCALABILITY

Fig. 1 shows the architecture of C-MART when utilizing all of its possible tiers. Each tier of the application is scalable due to the design of the application and the underlying software. The client can send requests to different front-end load balancers using a system similar to round-robin DNS load balancing. The application tier is scalable as it is stateless, does not use user sessions, and does not acquire locks to shared resources on other tiers, all user state is stored at the data tier. This also allows the application tier to be quickly scaled down if the workload is reduced, as the tier retains no client data. The search, image, and cache tiers scale due to the underlying software, Solr, MongoDB, and Memcache, respectively. Lastly, the database tier scales as C-MART has a NoSQL implementation using Cassandra, again utilizing the scaling ability of the underlying software. Connection caching and limits can be configured via a web API

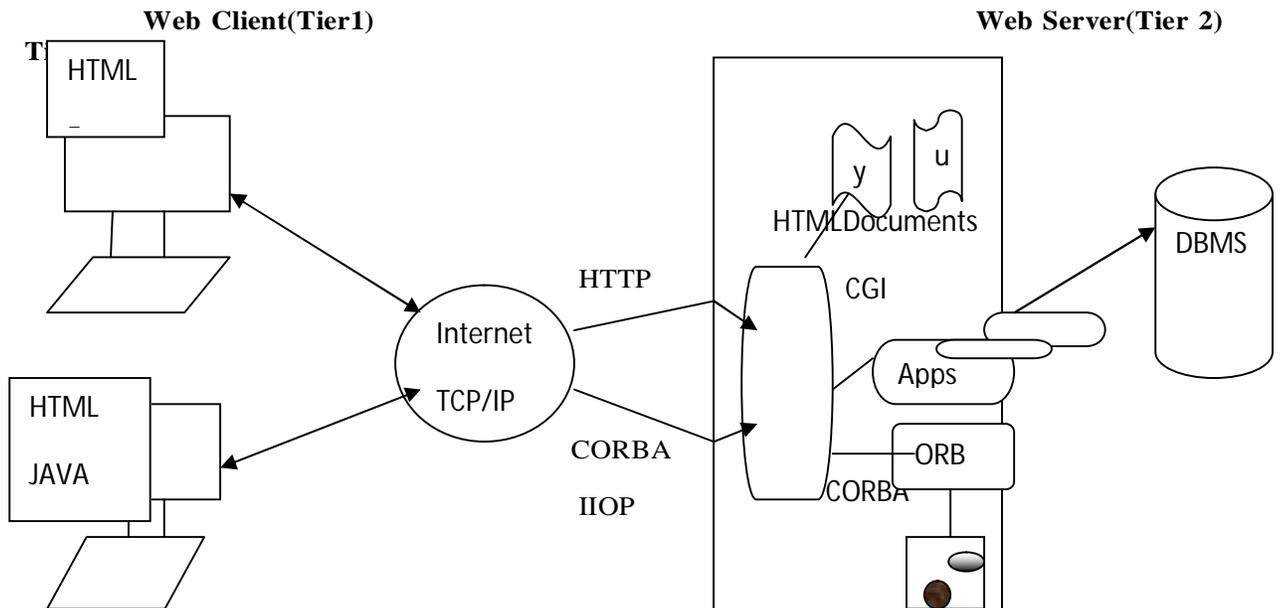


Fig.3. CORBA BENCHMARK

3.2 Modern Technologies

Modern web applications utilize technologies such as HTML5, AJAX, CSS, rich multimedia, and SQLite. These technologies can have a significant impact on resource utilization levels and request access patterns, which ultimately affect the clients' browsing experience. For example, SQLite is used to locally cache dynamic data.

3.3 C-MART BENCHMARK IMPLEMENTATION

This section briefly describes two middleware implementations of the benchmark, one using Microsoft's COM technology while the other using Iona's CORBA technology. Both implementations are based on the three-tier client/server model and they also use the same Java transaction generator. The transaction generator, the three-tier client/server model, and the two implementations are described in this section..

3.4 Three-tiered Client and Server Model

The conventional three-tiered client/server model, as given in [Jennings 1997], is used in both implementations. The subsystems that make up a three-tiered solution are data services, business services, and user services. Figure. 3 shows the traditional three-tiered model.

3.5 C-MART Benchmark Implementation Using Microsoft's COM Technology

This section briefly describes an example implementation of the C-MART benchmark using Microsoft's Component Object Model (COM). The client accesses the Web server through the Internet. The client



application is implemented using Active Server Page technology that adopts server-side scripting to dynamically create HTML responses. The clients' application is written using the ASP script language. IIS serves as the web server that accepts ASP requests from the ASP page sent from the client's browser. The business services in this subsystem are responsible for providing infrastructure and middleware. The business logic of the C-MART benchmark is implemented as COM components. MTS groups the COM components under transactional control. As middleware, the MTS is used to coordinate and monitor the transactional state of the COM components running under its control as these components interact with various transactional systems, such as remote SQL Server or Oracle databases. MTS is tightly integrated with IIS and the ASP engine. When business logic code in one COM component needs to access a remote supplier's database, the database connectivity technology.

3.6 C-MART Benchmark Using CORBA Technology

This section briefly describes an example implementation of the C-MART benchmark using a CORBA technology. In the 3-tier client/server architecture, the client is a Java applet downloaded by a Web browser. The middleware is the OrbixOTS server and business components. These business components are shopping cart component, customer registration component, buy component, order status component, category browsing component, product by category browsing component, product browsing component, and stock level component. The third tier consists of RDBMSs, illustrates the typical workflow of the C-MART benchmark. It demonstrates how the Web-based client interacts with its server. The client makes the initial requested for a page. Web browser receives and downloads an HTML page that includes references to embedded Java applets. Web browser requests the Java applet from the HTTP server. The HTTP server retrieves the applet and downloads it to the browser in the form of bytecodes. Web browser loads applet into memory. Applet invokes CORBA server objects. The Java applet includes the IDL-generated client stub classes, which let it invoke objects on the ORB server. Server objects access the database according to the defined business logic. Server objects return the results back to the Web client, and the results are displayed on the client's browser.

3.7 TIER CONFIGURATION

Not all application tiers are required when running C-MART. It can function as a two-tier application and database server configuration, up to the six-tier architecture shown in Fig.1. In addition, some tiers have multiple options for the underlying technology as C-MART has been implemented using multiple architectures. For example, the database tier has both a SQL and NoSQL implementation which can be chosen using a single flag. This allows C-MART to emulate both a traditional application using a relational data store and a modern application using a cloud storage engine. Any additional storage engine could be used by implementing a provided abstract class. Populators are provided to repopulate the databases with configurable amount of data before each experimental run.

3.8 C-MART using COM Technology

This section briefly describes an example implementation of the C-MART benchmark using Microsoft's Component Object Model (COM) / Microsoft Transaction Server 2.0 (MTS) / Internet Information Server 4.0 (IIS) / ActiveX Server Page 2.0 (ASP) platform. The client accesses the Web server through the Internet. The client application is implemented using Active Server Page technology that adopts server-side scripting to dynamically create HTML responses. The clients' application is written using the ASP script language. IIS serves as the web server that accepts ASP requests from the ASP page sent from the client's browser. The



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

business services in this subsystem are responsible for providing infrastructure and middleware. The business logic of the C-MART benchmark is implemented as COM components. MTS groups the COM components under transactional control.

3.9 PERFORMANCE METRICS

The performance metrics, response time and transaction throughput, are measured by running the C-MART benchmark application while varying a number of parameters such as the "think time", the "key time", the input transaction workload and the number of clients. During the measurement, the system log for the benchmark system is maintained. The mix of transactions in the C-MART benchmark system represents a complete business cycle. It consists of multiple business transactions namely the Shopping Cart Transaction, the Buy Transaction, the Registration Transaction, the Order Status Transaction, the Welcome Page Browsing Transaction, the Category Page Browsing Transaction, the Products by Category Page Browsing/Searching Transaction, the Product Page Browsing/Searching Transaction, and the Stock Level Transaction. The transaction throughput is the total number of completed business transactions divided by the elapsed time of the measurement interval. That is, the throughput is a number of business transactions processed per minute. If a transaction is aborted, say due to a deadlock, it is restarted by the transaction manager. The throughput of the C-MART benchmark system depends on both the system response and human interaction such as clients keying and thinking time and number of users, and percentage load by transaction type.

IV EXPERIMENTAL RESULTS

To identify where current benchmarks may produce inaccurate results for existing management schemes, we run C-MART against current benchmarks for a number of common datacenter scenarios. By comparing the results, we identify where the SUT fails to correctly react in a production environment if they are tested using current benchmarks rather than C-MART. We begin by outlining a number of common management techniques and algorithms and how they are applied to cloud applications. We use a custom datacenter with a flat local area network using commodity hardware. The physical hosts' OS is Red Hat 6.32 and runs the KVM hypervisor. Each server has a dual-core 3.2-GHz CPU with 4 GB of RAM. The hosts are connected via a Gigabit switch. Each VM has one virtual CPU and is allocated 1 GB of memory.

V CONCLUSION

Existing benchmark applications do not represent modern websites and are inappropriate for benchmarking cloud systems. Then presenting C-MART, a new benchmark application designed to emulate the behavior of modern cloud computing applications. C-MART can dynamically scale to support a large number of clients and has a flexible application design allowing it to emulate multiple different application architectures. It uses modern web technologies such as HTML5, CSS, AJAX, and SQLite and includes a workload generator that emulates clients accessing the website. It creates unique clients that change their behavior according to page content, history, and QoS. These factors make the client behavior more realistic and increase the variability of the server utilization and response time. C-MART's deployment server allows automatic, simple, and fast datacenter configuration and resource provisioning. The results show that existing benchmarks are overly optimistic in testing cloud management systems for production environments as they are unable to identify many deficiencies in the SUT. We show that existing workload prediction models could have 1,040 percent greater error than what was previously validated. We also demonstrate how existing management schemes could



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

under provision applications resources in 22 percent of time intervals. We supply Kernel-based Virtual Machine (KVM) images for every tier and install scripts for use on platforms such as Amazon EC2. We also provide operation instructions and detailed architecture descriptions.

VI. FUTURE ENHANCEMENT

The C-MART project is enhanced to n-tier architecture in future. It is also implemented in different middleware's such as RMI, RPC, CORBA and DCOM etc with each and every tier applications with animations technology. This can also be implemented in the mobile devices too. These can also enhance the C-MART tool performance and so that it can detect further errors from that distributed websites. Here application programming interface is used in advance. We supply Kernel-based Virtual Machine (KVM) images for every tier and install scripts for use on platforms such as Amazon EC2. We also provide operation instructions and detailed architecture descriptions. Further enhancement will improve the website performance during web hosting process. This C-MART can be implemented in the different tiers. The 3-tier application contains client tier, application server and the database tier. The Client tier is used to process the client applications. The application server is used to do the server oriented operations and the database tier is used for the database operations.

REFERENCES

- [1] Andrew Turner, Andrew Fox, John Payne, and Hyong Kim.S,'C-Mart: Benchmarking the Cloud',2013.
- [2] Armbrust.M et al., 'Above the Clouds: A Berkeley View of Cloud Computing,' 2009.
- [3] Binnig.C, Kossmann.D, Kraska.T, and Loesing.S, 'How Is the Weather Tomorrow? Towards a Benchmark for the Cloud,' Proc. Second Int'l Workshop Testing Database Systems (DBTest '09), 2009.
- [4] 'RUBiS,' <http://rubis.ow2.org/>, 2013.
- [5] Cooper B.F, Silberstein.A, Tam.E, Ramakrishnan.R, and Sears.R,'Benchmarking Cloud Serving Systems with YCSB,' Proc. First ACM Symp. Cloud Computing (SoCC '10), pp.143-154, 2010.
- [6] Padala.P et al., 'Automated Control of Multiple Virtualized Resources,' Proc. Fourth ACM European Conf. Computer Systems (EuroSys '09), 2009.
- [7] TPC BenchmarkW(Web Commerce) Specification,' San Jose, CA,USA, 2002.