CMPSCI 677 Operating Systems **Spring 2013**

Lecture 20: April 10

Lecturer: Prashant Shenoy *Scribe: Siddharth Chidambaram*

20.1 Cloud Computing

- Private Cloud : Enterprises use clouds for their internal usage . For example the IT department of the company or a University like UMASS might build a cloud just for its user base.
- Public Cloud : Anybody can be an end user . The cloud providers like amazon rent their services to any end user.
- Hybrid Cloud : In some cases when the enterprises use the private clouds and have resource needs that cannot be handled by the private cloud they dynamically pull resources from a public cloud. **Cloud bursting** is the process of moving to the public cloud when the workload goes beyond the private clouds Capacity.

20.1.1 Programming Models

The programming models include traditional Client server programming models or use of Batch processing or Map reduce models. The map reduce models are generally used for data intensive applications . It is a data parellel model where multiple servers are used to process the data . For example facebook applies mapreduce to find the the most popular pages and decide the rates of ads based on popularity .

20.1.2 Challenges in Cloud

One major concern with Cloud is Privacy/Security. Multiple users share a server and the provider has to be make sure that the users are isolated and the performance of an user is not affected by another user. With Extreme Scalability , management becomes a problem . Methods to automatically manage resources have to be incorporated . Appropriate Programming models have to be used to make optimum use of the large number of available servers

20.2 A case study on World Wide Web

20.2.1 WWW Principles

WWW is a standard client server System . The client is a web browser and accesses content from the web server . The web server can access a database to get some data requested by the client.

Server is not a single logical entity . It has a multi tier architecture . It has a front end web tier , application tier and a database tier . Front end is a HTTP server (Apache/IIS). The second tier is where the application

code is written . Applications can be written using Java (Servelets/J2EE), Python,PHP,Ruby on Rails. The HTTP requests from the front end are processed by the application tier and it makes use of the database tier to fetch any data required. On processing the requesting it builds an HTML document and sends it back to the front end.

To handle a large number of requests each of these tiers are replicated . In replication , there is a front end node . The browsers connect to the front end node. There can be any number of replicas . The front end acts as a dispatcher and redirects the request to a replica.

In case of stateful services the use of multiple replicas can lead to problems . The switch therefore has to be stateful and send the requests from a given user to the same replica again. Hence the load balancing is on a session basis .

A detailed Schematic of the switch has been given in the slide. The first time a request comes in from a user the dispatcher decides which replica handles the users request . Once the mapping is done the distributor is informed about the user mapping and a state is established .

The dispatcher is load balancer written by the application developer and hence includes application logic whereas the switch is a standard request forwarding switch and if it is not programmable then the dispatcher and the switch exist as separate entities . If the switch is programmable then can coexist .

The load balancing can be done at a DNS level . If the IP address of the replica is available then the loading balancing can be done using a round robin DNS . If the application is geographically replicated then an intelligent DNS can be used however within a Data center a dispatcher is preferred . Content Distribution Network makes use of intelligent DNS for load balancing .

20.2.2 HTML and Web Documents

The web browsers display HTML web pages . It is the standard for web documents. The table given in the slides includes different kinds of objects like images,audio,video that can be included in a document. the documents are encoded with these objects . A web browser first fetches a web page and parses it to find the page contains any objects like images or multimedia. If it contains objects it has to fetch them . HTML5 has encoding for audio/video and no plug-ins are required .

HTTP(Hypertext Transfer Protocol) is used to fetch the web documents. It runs over TCP and the HTTP requests and responses are used for communication between the browser and the web servers .

There are two HTTP versions

- **HTTP 1.0 :** For every object a new connection is started. Initally a TCP connection is established and the HTTP connection runs over it .This protocol fetchs the object from the server gives it to the browser and tear down the connection. It is called *Non persistent HTTP*
- **HTTP 1.1:** This is called *Persistent Connection* Instead of many connections , a single TCP connection is established and multiple HTTP requests are sent over it.

In case of HTTP 1.0 there is no need for cleanup however the overhead involved in establishing a TCP connection makes it less efficient.

20.2.3 HTTP methods

There are 5 HTTP methods . The Get method is the most commonly used methods and is used to return the document from the client. The Post method is used in the case of forms . The put and the delete methods are not commonly used. Put stores a document and the delete removes the document. Arbitrary users are not generally allowed to remove content unless a scenario where the file transfers are involved . The head method returns the header of the document

20.2.4 Web Services Fundamentals

When two machines communicate with each other without an intervention of an user they make use of web service protocols. Each of the machine exposes an API over HTTP . One machine communicates with another using a RPC call.
Some of the examples include the search box in the browser . In this case the search query is sent as a web service request . A dropbox client on the users machine communicates with the dropbox server using web services.

Two Commonly used protocols are

- **Simple Object Access Protocol (SOAP):** It is essentially RPC over HTTP . The slide gives an example of an XML based SOAP message . Here the name of the function called is alert control and a simple alert message is sent from one machine to another.
- **RESTful services:** Since SOAP is a heavy weight protocol , RESTful services are more commonly used . These are lightweight protocols and are used ony for point to point XML communications . They have the Get, post, put and the Delete methods . They are much closer the web and are much simpler when compared to RPC-style SOAP

The slide shows an example for RESTful Service request and response and the corresponding SOAP messages .The SOAP message is more complex when compared to the RESTful service. In RESTful services the arguements are a part of the URL whereas in SOAP they are a part of the message making RESTful simple and effective.

20.2.5 Comparison between SOAP and RESTful services

- SOAP is transport agnostic whereas the RESTful services work only over HTTP.
- SOAP supports n party communication while RESTful services work only in case of point to point communication
- RESTful services have no standard and the application designer can design it unlike SOAP which has standards and set of prebuilt tools to work on

20.3 Web Proxy Caching

The client-server system is modified to become a client-proxy-server system Proxies cache the data from the server and the client can access the data from the local proxies instead of requesting the server.
Cache hit: If the document requested by the client is present in the proxy then the document is returned

to the client
Cache miss: If the requested document is not present in the proxy , the proxy requests the server for the document and on returns the obtained document to the client

20.3.1 Consistency Issues

The major challenge faced by proxies is to provide consistent data to the client . The web pages at the proxy have to be updated over time to stay consistent with the servers. However, web pages have differing update frequencies making cache consistency complex .

Two messages involved :

- Invalidate message: Invalidate messages can be sent from the server to the caches to mark the pages inconstent .
- Update message: The update message , updates the web page at the cache . There is a overload involved in sending the new page to the cache and is preferred only if the page is popular.

20.3.2 Push based approach

In the push based approach the server tracks all the proxies that have requested the objects . If the web page is modified then each of the proxies are notified. The notification can be an invalidate or an update message. The approach used is to send updates for frequent objects and send invalidates to the less frequent used objects . This techniques provides tight consistency and is very effective when the proxies are not intelligent . However , there is a lot of overhead involved in keeping track of all the proxies that have requested the objects and in case of a server crash all the information is lost .

20.3.3 Pull based approach

In the pull based approach the proxies are responsible for maintaining consistency . They periodically poll the server to see if the object has changed. The frequency of polling however has to be determined . The time-to-live(TTL) assigned by the server gives a hint however there is no guarantee that the object will not be changed in the interim
One technique is to use Intelligent polling. In this technique the refresh rates are computed based on the past observations . Starting from a conservative refresh interval the rates are modified based on the change between two successive polls. This is an adaptive approach and requires no proir knowledge of the object characteristics.