A New Distributed Application Server System Based on Cloud Computing

Hong Sun*, Shi-ping Chen, Li-ping Xu, Ying-ying Chen
University of Shanghai for Science and Technology, Shanghai, 200093, China, telp +86-13916902800
*corresponding author, e-mail:sunhong_sh@sohu.com

Abstract
At this stage the application server systems, such as e-commerce platform, instant messaging system, enterprise information system and so on, will be led to lose connections, the data latency phenomena because of too much concurrent requests, application server architecture, system architecture, etc. In serious cases, the server is running blocked. The new type of application server system contains four parts: a client program, transfer servers, application servers and databases. Application server is the core of the system. Its performance determines the system's performance. At the same time the application servers and transfer servers can be designed as the web service open to be used, and they can be achieved as distributed architecture by a number of hardware servers, which can effectively deal with high concurrent client application requests.

Keywords: cloud computing, application server, distributed system, web service

1. Introduce
At this stage on Cloud Computing, the systems supported by application server are all influenced on the system construction, server's performance, configuration and so on. Usually in the situation of many visitors to the system, the application system will meet many types of errors.

For example, connections lose, transaction rollback and files break down. Then servers will be slower to resolve any request, even turn up the timeout status obviously in transferring the picture, sound and video datas. Regardless of the reasons in the applications' own errors, these situations are caused by the system architecture design, performance and distributed construction. These will not be obvious limitations in the small application system, but serious as growth as the system scale. For example, the accessories increase from five million to twenty five million in the MySpace development process. Then the MySpace is forced to turn the system performance by modifying the server cluster construction for suitting the actual situation. And the primary application server, web server and data warehouse all become the bottlenecks[1].

In addition, we have built a school automatic books system with the function of instant message for verifying our concept. When the visitors get to two thousand, the system becomes slower obviously. Although the server has not crashed, the timeout and connections lose come out. And we need to restore the datas manually for resolving the defections after the high cycle period.

And the situation is more instable when transferring files and audios. This will affect the user friendly degree. We adjust the system cluster construction to resolve the performance problems, then we find these operations useful.

As to the high accessory application system, the reasonable system and distributed application server construction will eliminate the bottlenecks. Now we design a new type of system construction and application server configuration to suit the need for highly concurrent accessories.

2. Cloud Computing Summary
Cloud Computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for
new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. They need not be concerned about overprovisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or underprovisioning for one that becomes wildly popular, thus missing potential customers and revenue. Moreover, companies with large batch-oriented tasks can get results as quickly as their programs can scale, since using 1000 servers for one hour costs no more than using one server for 1000 hours. This elasticity of resources, without paying a premium for large scale, is unprecedented in the history of IT.

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The datacenter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing. We use the term Private Cloud to refer to internal datacenters of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds. People can be users or providers of SaaS, or users or providers of Utility Computing. We focus on SaaS Providers (Cloud Users) and Cloud Providers, which have received less attention than SaaS Users.

From a hardware point of view, three aspects are new in Cloud Computing.
(1) The illusion of infinite computing resources available on demand, thereby eliminating the need for Cloud Computing users to plan far ahead for provisioning.
(2) The elimination of an up-front commitment by Cloud users, thereby allowing companies to start small and increase hardware resources only when there is an increase in their needs.
(3) The ability to pay for use of computing resources on a short-term basis as needed (e.g., processors by the hour and storage by the day) and release them as needed, thereby rewarding conservation by letting machines and storage go when they are no longer useful.

We argue that the construction and operation of extremely large-scale, commodity-computer datacenters at lowcost locations was the key necessary enabler of Cloud Computing, for they uncovered the factors of 5 to 7 decrease in cost of electricity, network bandwidth, operations, software, and hardware available at these very large economies of scale. These factors, combined with statistical multiplexing to increase utilization compared a private cloud, meant that cloud computing could offer services below the costs of a medium-sized datacenter and yet still make a good profit.

Any application needs a model of computation, a model of storage, and a model of communication. The statistical multiplexing necessary to achieve elasticity and the illusion of infinite capacity requires each of these resources to be virtualized to hide the implementation of how they are multiplexed and shared. Our view is that different utility computing offerings will be distinguished based on the level of abstraction presented to the programmer and the level of management of the resources.

Amazon EC2 is at one end of the spectrum. An EC2 instance looks much like physical hardware, and users can control nearly the entire software stack, from the kernel upwards. This low level makes it inherently difficult for Amazon to offer automatic scalability and failover, because the semantics associated with replication and other state management issues are highly application-dependent. At the other extreme of the spectrum are application domainspecific platforms such as Google AppEngine. AppEngine is targeted exclusively at traditional web applications, enforcing an application structure of clean separation between a stateless computation tier and a stateful storage tier.

AppEngine’s impressive automatic scaling and high-availability mechanisms, and the proprietary MegaStore data storage available to AppEngine applications, all rely on these constraints. Applications for Microsoft’s Azure are written using the .NET libraries, and compiled to the Common Language Runtime, a language-independent managed environment. Thus, Azure is intermediate between application frameworks like AppEngine and hardware virtual machines like EC2.

When is Utility Computing preferable to running a Private Cloud? A first case is when demand for a service varies with time. Provisioning a data center for the peak load it must sustain a few days per month leads to underutilization at other times, for example. Instead, Cloud Computing lets an organization pay by the hour for computing resources, potentially
leading to cost savings even if the hourly rate to rent a machine from a cloud provider is higher than the rate to own one. A second case is when demand is unknown in advance. For example, a web startup will need to support a spike in demand when it becomes popular, followed potentially by a reduction once some of the visitors turn away. Finally, organizations that perform batch analytics can use the “cost associativity” of cloud computing to finish computations faster: using 1000 EC2 machines for 1 hour costs the same as using 1 machine for 1000 hours. For the first case of a web business with varying demand over time and revenue proportional to user hours, we have captured the tradeoff in the equation below.

The left-hand side multiplies the net revenue per user-hour by the number of user-hours, giving the expected profit from using Cloud Computing. The right-hand side performs the same calculation for a fixed-capacity datacenter by factoring in the average utilization, including nonpeak workloads, of the datacenter. Whichever side is greater represents the opportunity for higher profit.

Table 1 below previews our ranked list of critical obstacles to growth of Cloud Computing in Section 7. The first three concern adoption, the next five affect growth, and the last two are policy and business obstacles. Each obstacle is paired with an opportunity, ranging from product development to research projects, which can overcome that obstacle.

We predict Cloud Computing will grow, so developers should take it into account. All levels should aim at horizontal scalability of virtual machines over the efficiency on a single VM. In addition

1. Applications Software needs to both scale down rapidly as well as scale up, which is a new requirement. Such software also needs a pay-for-use licensing model to match needs of Cloud Computing.

2. Infrastructure Software needs to be aware that it is no longer running on bare metal but on VMs. Moreover, it needs to have billing built in from the beginning.

3. Hardware Systems should be designed at the scale of a container (at least a dozen racks), which will be is the minimum purchase size. Cost of operation will match performance and cost of purchase in importance, rewarding energy proportionality such as by putting idle portions of the memory, disk, and network into lowpower mode. Processors should work well with VMs, flash memory should be added to the memory hierarchy, and LAN switches and WAN routers must improve in bandwidth and cost [2].

3. Research Method based on Cloud Computing Distributed Server Construction for Applications

There are two parts for our new application system base-on Cloud Computing. First, the application system base-on Cloud Computing must be improved. Second, the system base-on Cloud Computing performance and construction also need to be modified.

In this graph, the whole system base-on Cloud Computing construction has four parts [3]:

![Figure 1. System base-on Cloud Computing architecture](image)
(1) Client program base-on Cloud Computing: the client program can be in the model of B/S as the web client, or C/S as the application client. The web browser client interacts with the application server by the web server, and the application client can work by the socket communication technology. The model in the test application, the B/S model is chosen. And we can achieve the load balance by the web server, system architecture by which we can dispatch the clients’ requests reasonably [1,4].

(2) Transfer server under Cloud Computing: This is used to release the burden for the core application server and edge resolution. Usually, many visitors will share the common transfer server. For example, a instant message application server ability for customers is about fifty thousands after testing, and this can be decreased to thirty thousands by the help of transfer server. Meanwhile, these transfer servers can help to decrease the number of ports opening to all the clients. This means that the transfer server is useful for performance and security. The transfer server’s main duty is to monitor the connection request from the customers and dispatch the requests to the core application servers. In addition, these dispatch operations can be consoled by developers. Except this, the connections between transfer server and application server depends on the method of buffer pools. Because the buffer pools can be designed by programmers, so this will be helpful for avoiding the data and connection lose.

(3) Application Server under Cloud Computing: The application server’s abilities are to load and resolve all the applications. Then create the data source to execute the operations on data warehouse. This is the server which we talk in common. Once the application server receives the request by the transfer server, it will call the relevant applications to resolve through the web container. And the application can call the data source program by the web container, too. At last, the application server will response the request to the destination transfer server. Due to the core functions are in the application server, they need the high performance hardware. In the test environment, the system construction is designed as the distributed cluster for the whole performance, and the server cluster with three nodes work together for getting the system performance, in further, this can raise the stability of the whole system. And the web container, EJB container and data source buffer pool are all needed to designed adhere to the actual visits. Meanwhile, the applications functions will be designed as interfaces for raising the expansibilities of the application server. And the transfer server can be re-configured. Further, the client and transfer servers can be designed by the application server interfaces, and this also reflect the SOA thinking of loose coupling. And all the servers will share the data warehouse, message queues and so on [5,6].

(4) Data Warehouse under Cloud Computing: This main function is to support storing the datas for applications. It can be called as the repository for the distributed servers. It can be the products of database, or web service, configuration file and message queue. Then this part can also be built as a cluster with two or more nodes. In the system desigment, application server is the core. Its duty is to load all the programs except the user interface. The whole system performance is decided by this. And the transfer servers which are used in the distributed system is a innovation in common system. The transfer servers can release the burden of the application servers, and the system performance can be raised by designing the reasonable the transfer application construction [7].

The advantages about this construction contain the following aspects: first, the clients can work with the user interface only. The designer can work with it flexibly through the users’ demand. And the main functions will not be cared by the clients. Meanwhile, the transfer servers can achieve the load balance and raise the system performance. Second, the transfer servers can make the functions of SLL, cookies and so on. Not only can they reduce the core server burden, but also can they take advantages of the hard ware. The application server only need to open little ports for transfer servers, so the stability and security can be raised through this. At last, the application server only receive the base XML format data by the transfer server help. And the application server will understand the less burden. Meanwhile, the function will be extended easily under the format of XML. Except this, the application server functions are all designed as interfaces, and this is beneficial obviously. And the construction is designed as the cluster. In the application server construction, the application server managed by the Manage Node. Its responsibility is to resolve all the sub-node servers. This function achieve by the M-Bean, and its theory is similar to the java virtual machine M-Bean. For example, once the
requests from the transfer server come to the application server, the Manage Node will dispatch this to one of the application nodes. And other nodes can keep synchronize with the first node. The application server nodes will execute the application distributed. Every application server has his own M-Bean, and this is similar to the Manage Bean’s M-Bean. They all can interact with other M-Beans. Now let’s introduce the application system designed in this paper. And this will show the relations among every parts of the system and judge the system by the performance. The application system is used for automatic school books with the function of instant message. The performance will decrease obviously when one hundred concurrent visits. The part of instant message has the problem seriously. So the system choose to adapt the cluster of nine machine, separately one client server, three transfer servers, four application servers and a data warehouse. The instant message function adhere to the protocol called XMPP, and the console operation commands also extend from this protocol.

Every application server system base-on Cloud Computing will adhere to his own communication protocol and data format. At this stage, the XML format becomes the popular one. Such as the SOAP protocol in web service. In our application, the XML data format is used and match the XMPP. The XML format data can be designed to three types based on the function demand. The console commands are all sorted out as the state type. These commands record the relevant status commands. For example, the operation of borrowing and returning the book, users start a chart conference and so on. The message commands have the duty to transfer the messages among system parts. And these commands must be stored when the connections between two customer is closed. The query commands are all used to get information from the application server, for example query the relation between the client and server [8].

Now let’s show the command used to send a request to server for borrowing the book:

```xml
<presence type="active" from="custom01" to="mid-server-01">
  <x xmlns="http://custom-presence/control">
    <control type="subscribe" num="1" name="***" serialnum="AV***"></control>
  </x>
</presence>
```

In this command, the client send a online console request to server for subscribing the book. And it has recorded the base information about the book. And the transfer server will resolve this command later.

4. Results and Analysis

This new type of design base-on Cloud Computing adopts the structure with one console node and three application server nodes. And every nodes will transfer messages and work together. This will show in the Figure 2.

The manage node is responsible for adjusting and transferring the messages. It executes the operations on other cluster nodes and the data warehouse. It will receive the request from transfer servers first, then dispatch them to all the application server nodes.

Every application server node has his own java virtual machine to operate. This means that the applications base-on Cloud Computing will be deployed on every nodes. And each of them has the manage bean and class loader. Except the class loader adhere to the java virtual machine, the application system has been developed three new types of class loaders for the cluster environment. They are War Class Loader, App Server Class Loader and Manage Server Class Loader. First, War Class Loader’s duty is to load the classes which are all built by the applications. Second, the App Server Class Loader loads the EJB files, RAR files and the classes in the tool jar archive. At last, the Manage Server Class Loader will load the classed needed to be loaded once when the server starts. Now the concurrent mechanism will work after the class loader pattern is well. The manage node and application server node all have their own manage bean for interaction with others. This design is similar with the java virtual machine. And the actual technology under the node can be chosen adhere to the demand...The manage bean becomes the interface for outside. This process is easy to understand among these interfaces [9].
The transfer server act as both the server and client. The client can interact with the application server with the mechanism of concurrence and polling. The client can interact with the transfer server with the finite-state machine for the long period request. And the clients can adopt the multi-threads to work with the transfer servers. The load balance can be resolved by
the web server in the transfer server. The transfer server will reset the data getting from the clients to XML format, then send them to the application server cluster.

By the load test, the average response time will reduce from twenty seconds to five seconds in the concurrency up to one hundred. In Figure 3, we construct the load test with one hundred concurrency request lasting for three minutes. Total one thousand three hundred twelve responses come back, and the longest time is fifty seconds. The average time is fourteen seconds [10], and the response time is only one hundred millisecond under the stand-alone request.

![Figure 4. The new test result](image)

In the Figure 4, the average response time is only seven hundreds seventy five millisecond. And the longest time is eight seconds. So the system performance has raised obviously after the improvement. Meanwhile, the function of the system can be extended easily and keep high stability.

5. Conclusion

We have designed a new type of distributed application system base-on Cloud Computing. The mechanism contains three parts of client, transfer server and application server. The system base-on Cloud Computing has the stability and extension. Meanwhile, the middleware can be used for other application. This shows the popular thinking of SOA architecture. And it is the base of the cloudy computing. For example, this application server can support the services for other applications.
Acknowledgement
The National natural Science Foundation of China(61170277). Supported by innovation Program of Shanghai Municipal Education Commission (12zz137)

References