ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **9** (2020), no.7, 5085–5093
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.7.74 Spec. Iss. on AMABDA-2020

# EVALUATION OF PERFORMANCE FOR BRIDGING THE GAP BETWEEN VOLUNTEER AND CLOUD

ANJANA SANGWAN[1] AND RASHID HUSSAIN

ABSTRACT. Unlike cloud computing, volunteer computing is a kind of distributed computing where idle computing resources are harnessed so as to accomplish distributed independent tasks. Volunteer computing propose an advantage of shared volunteer resources that can be used by various scientific and research projects in order to carry out the needs and requirements of the project. For the betterment of research and development, volunteer computing projects play an important role in cloud resource management .The main goal is to utilize the contributed resources volunteered by people so as to generate scientific outcomes.

## 1. INTRODUCTION

Cloud computing is cost effective because in order to implement web services we donâĂŹt require a high computing power. When users have to deal with web based applications, not much storage capacity is required. In cloud computing as there are a less number of programs placed in the memory, it will automatically increase the performance of the system. When the user deals with web-based applications, cloud computing provides automatic updation of software and the latest version of the software is provided. In cloud computing, there is no such issue of compatibility when users have to share some documents or applications.

[1]*corresponding author*

2010 *Mathematics Subject Classification.* 65Y20.
*Key words and phrases.* Cloud, Computing, Scalability, Resources, Management, Volunteer, Network latency.

Cloud computing offers web services for public use, anyone can access these services or resources from anywhere via internet. Users can freely share documents and applications; this will result in better collaborative environment.

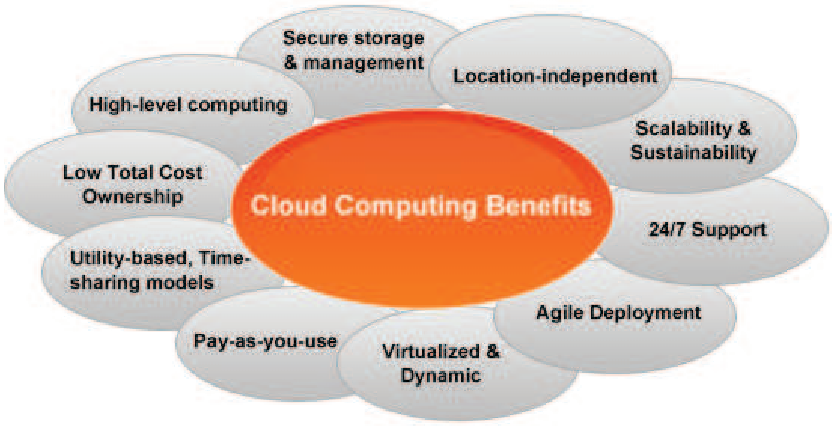


FIGURE 1. Cloud Computing Benfits

Cloud computing offers a number of advantages but there are some issues also: The first one is the availability issue, second is the security issue, sometimes organizations and enterprises feel that their important data stored in someone else server over the internet is at risk or unprotected. And next is the reliability issue.

Valuable references on the topic are given in [1-10].

## 2. VOLUNTEER COMPUTING

Volunteer computing is based on exploiting the computing resources volunteered by individuals for a particular project. (eg. SETI@home involves more than 200,000 users and FOLDING@home is a collection of 450,000 CPU, PCs etc.). Volunteer computing has been successful in collecting a massive number of resources from people across the world. In order to carry out computational needs of an organization for a particular project, SETI@home project offers a collection of computing resources donated by individuals across the globe. Volunteer computing is based on a central server which is responsible for managing the resources and for allocating the jobs to the accessible computers. Science and research projects requires extremely high computing power and for this purpose a large number of a large number of dedicated volunteers and their computing resources have turned out to be extremely useful. In volunteer computing multiple computers are responsible to carry out a complex problem which is further divided into various jobs. To compete with the problems of scientific and research projects, high computing power is a not sufficient, but the mutual intelligence power of volunteers is also required. Large scale volunteer computing systems are based on client server architecture which is defined in the form of layers, where each layer has its own particular function to perform. In this architecture the concept of agent and broker is there, which is responsible for the communication between a projectâĂŹs client and server. It is also known as middleware layer with which the appropriate data that has been processed can be sent back to the scientists. In order to perform scientific computing in a distributed way, volunteer computing make use of computational power proposed by general public. Volunteer computing is a technique that let its internet users to donate or offer their computing resources in easier, faster and a cost-effective manner. And the users and the volunteers who offer their computing resources are considered to be as unidentified.

## 3. VOLUNTEER CLOUD COMPUTING (VCC) ARCHITECTURE

Abdullah, Robert and Gary proposed VCC architecture [3] which makes it easier for volunteer cloud to be an applicable cloud solution. This VCC architecture is composed of three layers:

- Service layer
- Virtual layer
- Physical layer

Each layer is responsible to carry out its own specific task. The middleware layer and physical layer are further divided into sub- components. Maintaining the Integrity of the Specifications the service layer is responsible for delivering the services to customers through an interface formed on SOA approach. The way it deliver its services to customers is same as that of commercial clouds.

3.1. **Virtual layer.** In this layer resources are offered to the service layer. This layer is further divided into two parts: task management and QoS management. Task management deals with tasks obtained from the service layer. The task scheduler is responsible for scheduling the tasks to appropriate resources that are provided in the physical layer. In order to lessen the time taken to execute a task, load balancing make sure that if the load is dispersed properly. In volunteer cloud computing self-automation is used to offer rapid elasticity so that users can access services according to their requirements.

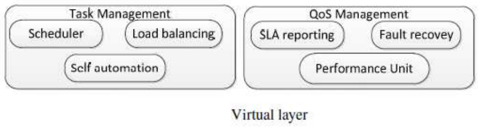


FIGURE 2. *
Virtual layer

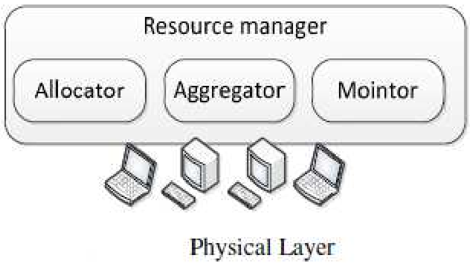


FIGURE 3. *
Physical layer

FIGURE 4. Virtual and Physical Layers

3.2. **Physical Layer.** Physical layer is further divided into three parts: allocator, aggregator and monitor. The combination of these three is known as resource manager. Aggregator component of resource manager is used to combine volunteer resources contributed by various individuals. Depending upon various criteria resources are classified using an aggregation mechanism so as to achieve quality of service. The allocator component is responsible for allocating the tasks obtained from task management to suitable available resources. The monitor component monitors the reliability of these allocated resources from time to time.

3.3. **Second Scenario.** So in order to accomplish this scenario, a Use Case Diagram is described first which depicts all the high level functionality of our entities and next a Sequence Diagram it is also termed as interaction diagram is presented and it shows what will be the sequence of the information that will flow between the objects. Here Enterprise Architect tool is used for designing these UML diagrams. And also display how the entities will be going to interact. Finally the implementation is defined. Suppose ab.domainA.net is the initiating XMPP server or we can refer to it as server1 and domainB.net is the receiving

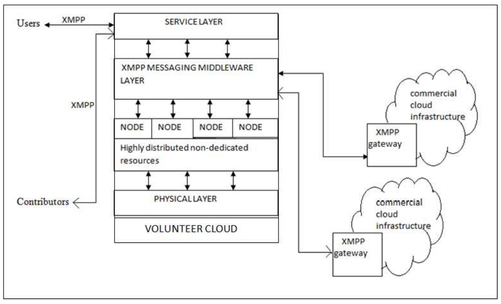


FIGURE 5. XMPP based messaging oriented middleware

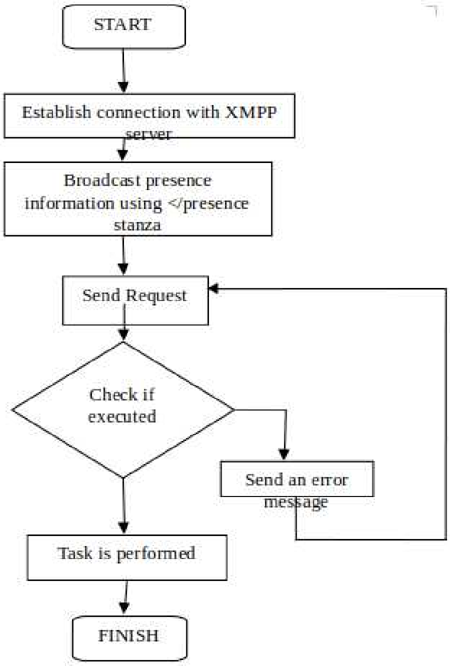


FIGURE 6. Flow chart for internal communication using XMPP

XMPP server or we can call it as server2. Initially server1 has to perform DNS lookup with the authentic server and after clarifying a Service Record of _xmpp-server._tcp.domainB.net and then a request is sent from server1 to server2 in order to establish a TCP connection.

Now server1 set up an XML stream with server2 in order to carry out TLS negotiation. Features of stream along with a stream header are sent as a response from server2 to server1.

```
S2: <stream:stream
      from='domainB.net'
      id='hTiXkW+ih9k2SqdGkk/AZi0OJ/Q='
      to='ab.domainA.net'
      version='1.0'
      xmlns='jabber:server'
      xmlns:stream='http://etherx.jabber.org/streams'>
```

```
S2: <stream:features>
      <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'>
        <required/>
      </starttls>
    </stream:features>
```

In order to carry out TLS negotiation, a STARTTLS command is transmitted from server1 to server2. Now both the servers exchange certificates so as to complete TLS negotiation. After the successful completion of TLS negotiation, a new stream is set up by server1 for SASL authentication.

```
S1: <stream:stream
      from='ab.domainA.net'
      to= 'domainB.net'
      version='1.0'
      xmlns='jabber:server'
      xmlns:stream='http://etherx.jabber.org/streams'>
```

Features of stream (SASL External Mechanism) along with a stream header are sent as a response from server2 to server1.

```
S2: <stream:stream
      from='domainB.net'
      id='RChdjlgj/TIBcbT9Keu31zDihH4='
      to='ab.domainA.net'
      version='1.0'
      xmlns='jabber:server'
      xmlns:stream='http://etherx.jabber.org/streams'>

S2: <stream:features>
      <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
        <mechanism>EXTERNAL</mechanism>
        <required/>
      </mechanisms>
    </stream:features>
```

An external mechanism is selected by server1, this selected external mechanism is sent as a response to server2 along with an encoded identity. If the information given in the certificate matches with the encoded identity sent by server1, then server2 returns SUCCESS in response.

After the authentication process both the servers (clientâĂŹs server and volunteer cloud server) are free to communicate with each other via exchanging XML stanzas. The communicating servers now can send any number of XML stanzas with each other. Suppose $abc@domainA.net$ of $domainA.net$ wants to establish an interaction with $xyz@domainB.net$ of domainB.net for consuming resources that are present on the volunteer cloudâĂŹs server which we denote here as $domainB.net$. Now $domainB.net$ check if the requested resources are available, if available then it grants the resources to the requesting party. It means that these resources are logically present in $domainA.net$ but are physically located on $domainB.net$.

TABLE 1. Feature Comparison

| Feature | XMPP | HTTP+REST | HTTP+SOAP |
|---|---|---|---|
| **Security** | TLS | SSL | SSL |
| **Reliability** | YES | NO | NO |
| **Authentication** | SASL | NO | NO |
| **P2P Support** | YES | YES | YES |
| **Easy to Integrate with Web** | YES | YES | YES |
| **Easy to Integrate with Operators** | EASY | N/A | N/A |
| **Identity Management** | YES | NO | NO |
| **Bi-Directional Communication** | YES | NO | NO |
| **Overhead** | HIGH | HIGH | HIGHEST |

## 4. RESULT AND DISCUSSION

The significant findings include:

Now we will compare XMPP pubsub with other HTTP based protocols based on some metrics:

- Latency
- Bandwidth
- Network overhead

### 4.1. Latency Comparison between polling and XMPP applications.

Minimum 3 network hops are required with HTTP based protocols whereas only a single hop is required with XMPP pubsub. So XMPP pubsub is 66% faster than the other one.

From the above figure it has been shown that the latency in the case of polling will be more as compared to XMPP latency. This is because message travels from server to the browser in about 50 milliseconds in case of polling and also the new request would be sent to the server as soon as the response will be completed. It means that another 50 ms would be taken by the new request and this will further increase the memory consumption of the server.

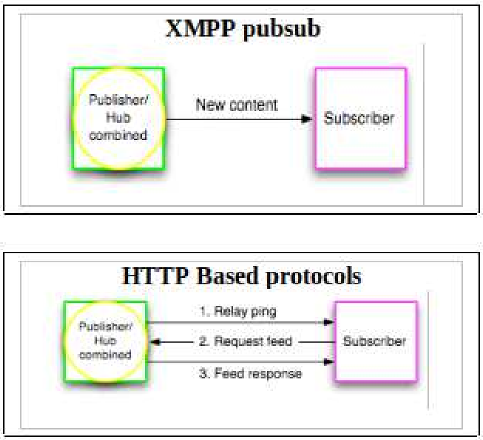


FIGURE 7. Latency Comparison

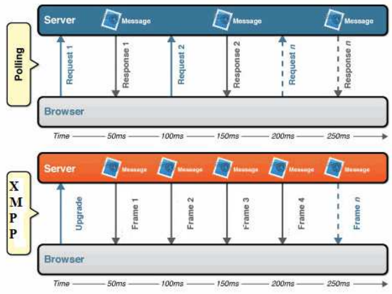


FIGURE 8. Comparison of Latency of HTTP and SAMPP Server

## CONCLUSION

The present study has brought out volunteer cloud computing to another level by providing an effective and more secure communication so that clients can freely consume and contribute their resources. The present study works for the betterment of the computing paradigms as by using XMPP, effective communication will be carried out which will help the organizations to cut down IT cost, and also help in reducing overall power consumption and also deduct the cost related to data migration. XMPP offer a communication channel with which organizations can even donate their unused computing resources. Scientists and researchers utilize the computing power provided by volunteer cloud for generating scientific outcomes by establishing this communication channel.

## REFERENCES

[1] I. FOSTER: *Service-oriented science*, Science **308**(5723) (2005), 814-817. DOI: 10.1126/science.1110411

[2] L.J.L. ZHANG: *EIC Editorial: Introduction to the Body of Knowledge Areas of Services Computing*, IEEE Transactions on Services Computing, **1**(2) (2008), 62âĂŞ74.

[3] I. FOSTER, S. TUECKE: *Describing the Elephant: The Different Faces of IT as Service*, Queue **3**(6) (2005), 26âĂŞ29.

[4] L. WANG, J. TAO, M. KUNZE, A.C. CASTELLANOS, D. KRAMER, W. KARL: *Scientific Cloud Computing: Early Definition and Experience*, 10th IEEE International Conference on High Performance Computing and Communications, Dalian, 2008, 825-830. doi: 10.1109/HPCC.2008.38

[5] L. KLEINROCK: *A vision for the internet*, ST Journal of Research, **2**(1) (2005), 4âĂŞ5.

[6] T. OÂĂŹREILLY: *What is WEB 2.0 (September 2005)*, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[7] RESERVOIR CONSORTIUM: *Reservoir Project (2009)*, http://www-03.ibm.com/press/us/en/pressrelease/23448.wss/

[8] UNIVERSITY OF CHICAGO-UNIVERSITY OF FLORIDA-PURDUE UNIVERSITY-MASARYK UNIVERSITY: *Nimbus-Stratus-Wispy-Kupa Projects (January 2009)*.

[9] DISTRIBUTED SYSTEMS ARCHITECTURE RESEARCH GROUP: *OpenNEbula Project*, Universidad Complutense de Madrid (2009), http://www.opennebula.org/

[10] S. PRABU, V. BALAMURUGAN, K. VENGATESAN: *Design of cognitive image filters for suppression of noise level in medical images*, Measurement **141** (2019), 296-301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINNERING, SURESH GYAN VIHAR UNIVERSITY, JAIPUR, INDIA.

*E-mail address*: sangwan.anjana@gmail.com

*E-mail address*: rashid.hussain@mygyanvihar.com