REST: An Approach to Solve the Server Problem during the Retrieval of Result

Manasa Bhat, PG Student, Dept. of CSE[1], AIET, Moodbidri, India

Chanchal Antony, Assistant Professor, Dept of CSE[2], AIET, Moodbidri, India

Abstract: Popularization of web of things generation and the deployment of embedded gadgets, the kinds and quantities of clever objects are continuously developing, and web of things systems and structures are increasingly more famous in daily lives. The SOAP-based web service is too complex, hardly ever exposed to the linkages and HTTP features are almost never used. In order to overcome those disadvantages, REST architecture is proposed and suggestion of the web of things systems based on it. REST architecture improves a plenty in the flexibility and simplification of the system. Web service is one of the software applications, XML-based languages are used to access the web service. The implementation of REST over cloud is demonstrated by taking the example of the retrieving the results from the VTU website. This reduces the load on the server and helps in fast access, reliable and better maintenance.

Keywords: REST, XML, VTU, SOAP

I.   INTRODUCTION

Cloud computing is a computation and business mode giving users with expandable resources. Cloud computing provides every kind of resources and knowledge on the Internet. The elasticity is the key necessity in these applications of cloud computing. The distribution of cache information ought to be adjusted according to the dynamic changes of network and application in key-value caching system and also the performance of the system is exaggerated by the load on the cache and bandwidth of the system. Cloud computing provides every kind.

REST is a structure fashion and not a protocol, which is usually based at the popular protocols including http, URI, xml and html. All the resources are diagnosed by means of URI. Representational state for transfer abstracts every aid dispensed within the gadget into a completely unique, global identity, so that every aid can be identified. This applies to each web applications and system-to machine communications. Thus uniform resource identifier is the simple premise of REST.

All the sources are related together. Representational state for transfer is absolutely designed for distributed hypermedia structures, and the center concept described here is hypermedia, in other words, the concept of links. For one factor, packages can retrieve extra statistics following the hyperlinks, making all the sources inside the web interconnected. For other, the server gives to the client set of links, so that the state transfer takes place from one state to another by client. Consequently, beneath any possible situations, resources are identified by using links.

Manipulate resources with the same old techniques. Representational state for transfer includes operations like obtain, delete and modify. Those operations precisely correspond to POST, GET, DELETE and PUT in http protocol. Applying these standard techniques make applications a part of the internet and all of the components know-how http protocol can have interaction with programs. So on the way to make the client applications collaborate with resources, resources must properly enforce the default software protocol (http), because of this making use of the usual GET, PUT, DELETE and POST methods.

Multiple representations of resources are possible in Representational state for transfer. Multiple representations of resources for distinctive needs, like HTML and XML representations are provided. This has an important advantage in practice, which can not only be used by the applications, but also be used in any standard web browser. In other words, application information can be accessed for all the web users.

Stateless communication is also one of the principles. Right here "stateless" refers that the connection protocol uses no status characteristics. Representational state for transfer stateless connection requires the statistics send by stateless connection protocol have to contain state information related to the application. For the server scalability these state information must be placed in resource state or stored on the consumer, to attain the scalability of the server. In addition, stateless boundaries keep the changes in server invisible to the client.

A rising paradigm Cloud computing is gaining lot of attention by the day. Result retrieval is that the activity of getting the result from a group of results that keep within the storage system like cloud. Visvesvaraya Technological University (VTU) conducts semester examination for graduates and post graduates technical courses. Students always become so curious to know their results once they complete their examination and they start wandering here and there to get any notification related to the results. Students are also curious and bit nervous about the result. The result of the student of the university is retrieved by using the USN which is assigned to the each student by the university.

When result is out all students try to check their results. Because of huge number of students accessing same server, server becomes very busy and server goes down. It is difficult for students to view their results because server is unavailable. To avoid this REST approach is used which provides solution to this problem.

II.   EXISTING SYSTEM

When the results are announced, servers often go down and due to this, students face a lot of difficulty to retrieve results. Huge number of students appears in the examination. So those students are waiting for the result. Once the result is out, all the students who appear in the examination check the result from VTU website as shown in Figure1. But VTU website is very busy because of huge number of request from the each student.

i.   DRAWBACKS

When result is out all students try to check their results. Because of huge number of students accessing same server at the same time, server becomes very busy and server goes down. It is difficult for students to view their results because server is unavailable.
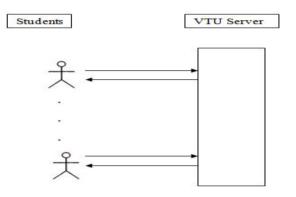

Figure 1: Accessing result from VTU server

III.   PROPOSED SYSTEM

Once the result is announced, servers often unavailable and because of this, for students it is difficult to retrieve results. Many students appear in the examination. All students try to check their result when the result is out but server is very busy because of huge number of request from the each student.

To avoid this An Approach Using REST to Solve VTU Server Problem during the Retrieval of Result is proposed which includes main cloud and campus cloud. REST make full use of Web features, and also has the advantage of simplicity. Main cloud contains results of all college students. If we consider one college and the campus cloud related to that college when the first student from that college sends the request to the campus cloud, the request is forwarded to the main cloud and all the results of that college is send back to the campus cloud. So now students of that college can access the result in faster manner as shown in the Figure 2.
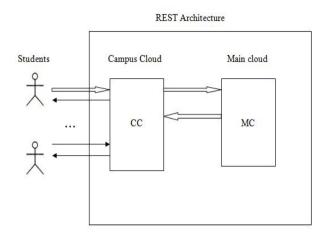

Figure 2: Accessing result from the main cloud along with REST for one college

i.   MOTIVATION

University exams are itself a big deal for students and if there are some added diminishing aspects to it then it just piles on to some extra pressure. Once the VTU result is out all the students of different branch try to check their results by using their unique USN. The result retrieval process begins when student enters a USN which is provided by the university. Many people are trying to access the same server at the same time because of this server goes down and results are unavailable. The problem faced by the student to retrieve the result from server is one of the motivations.

When result is out from VTU, students are eager to check their results. Students are always curious to see their result but the result is not available because of very busy server.

ii.   PROBLEM STATEMENT

Proposed model takes into account the problems of overloading the server by sending many request for retrieval of VTU result of student bearing unique USN. Server goes down if many requests arrive from many students at the same time which is also handled by the proposed model.
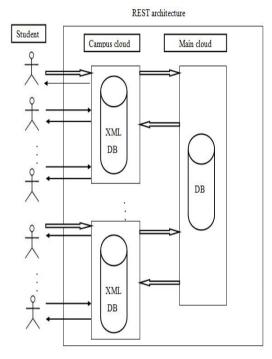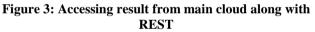
iii.   OBJECTIVES
- Faster result access
- User friendly
- Reliable
- To improve the overall performance
- To reduce the overhead on the server

iv.   METHODOLOGY

Huge number of students accessing same server at the same time, server becomes very busy and server goes down. It is difficult for students to view their results because server is unavailable. To avoid this An Approach Using REST Architecture to Solve VTU Server Problem during The Retrieval of Result is proposed which includes main cloud and campus cloud as shown in Figure 3. Flow

chart of accessing the result is shown in figure 4.These are the modules used in the system.

1) Login to Campus Cloud using USN

2) Requesting Main cloud for student result

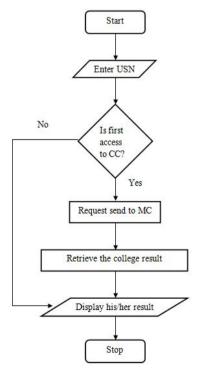3) Retrieving data from Main Cloud to Campus Cloud

4) Result check

Figure 3: Accessing result from main cloud along with REST

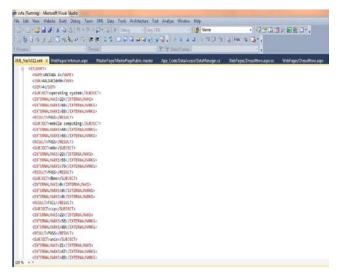**Figure 4: Flow chart of accessing result**

Figure 5: After the first access to the campus cloud all student result of that particular college is stored in XML file

v. APPLICATIONS

- Faster VTU result access
- Government official sites
- Data driven site

IV. CONCLUSION

New mode of thought for service abstraction is REST. REST ensures to truly understand the original look of HTTP and fully utilize current Web features. A problem of overloading the server is taken into consideration by the proposed model by sending the bulk request for retrieval of result of student bearing unique USN. The benefits of using proposed method could be, faster result access, user friendly, reliable, improvement of overall performance and reduce the overhead on the server. Figure 5 shows after the first access to the campus cloud all student result of that particular college is stored in XML file. Later onwards results for that college is retrieved by using that XML file.

REFERENCES

[1] Irwan Alnarus Kautsar1, Shin-Ichiro Kubota2, Yasuo Musashi1, Kenichi Sugitani1, "Redefining Data Provider: The REST Approach To Solve Indonesia Lecturer Administrative Problems", 2014 International Conference of Teaching, Assessment and Learning (TALE)978-1-4799-7672-0/14,2014 IEEE

[2] Bruno Costa, Paulo F. Pires, Flávia C. Delicato, Paulo Merson, "*Evaluating a Representational State Transfer (REST) Architecture* What is the impact of REST in my architecture?", 2014 IEEE/IFIP Conference on Software

[3] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, "Web services composition: A

decade's overview," Information Sciences, vol. 280, pp. 218–238, May. 2014.

[4] Z. Guo, M. Su, Y. Xu, Z. Duan, L. Wang, S. Hui, and H. J. Chao, "Improving the performance of load balancing in software-defined networks through load variance-based synchronization," Computer Networks, vol. 68, pp. 95–109, Aug. 2014.

[5] I.A. Kautsar, S. Kubota, Y. Musashi, and K. Sugitani, "A Supportive Tool for Lecturers to Upload LMS Learning Contents Automatically." 66th Joint Conference of Electrical, Electronics and Information Engineers in Kyushu. Kumamoto University, 2013.

[6] Fernandes, Joel L., et al. Performance evaluation of RESTful web services and AMQP protocol. In: Ubiquitous and Future Networks (ICUFN), 2013 S. 810-815.

[7] Naab M., "All Architecture Evaluation is not the Same – Lessons Learned from more than 50 Architecture Evaluations in Industry," in Architecture Technology User Network (SATURN), 2013

[8] Marinescu, D.C., Cloud Computing: Theory and Practice. 2013: Newnes.

[9] Bass L., Clements P., Kazman R., Software Architecture in Practice (SEI Series in Software Engineering). 3rd ed. Addison-Wesley Professional: 2012.

[10] C. Davis, "What if the Web Were Not RESTful?," in Proceedings of the Third International Workshop on RESTful Design, New York, NY, USA, 2012, pp. 3–10.

[11] Hobfeld, T., et al., Challenges of QoE management for cloud applications. Communications Magazine, IEEE, 2012. 50(4): p. 28-36.

[12] K. Gilly, C. Juiz, and R. Puigjaner, "An Up-to-date Survey in Web Load Balancing," World Wide Web, vol. 14, no. 2, pp. 105–131, Mar. 2011.

[13] L. O. Moreira, F. R. C. Sousa, and J. C. Machado, "A distributed concurrency control mechanism for XML data," Journal of Computer and System Sciences, vol. 77, no. 6, pp. 1009–1022, Nov. 2011.

[14] Hamad, Hatem; SAAD, Motaz; ABED, Ramzi. Performance Evaluation of RESTful Web Services for Mobile Devices. Int. Arab J. e-Technol., 2010, 1. Jg., Nr. 3, S. 72-78.

[15] Xinyang Feng, Jianjing Shen, Ying Fan "REST : An Alternative to RPC for Web Services Architecture" 978-1-4244-5160-9/09/©2009 IEEE

[16] *Pîrnāu Mironela, "The Importance of Web Services Using the RPC and REST Architecture",* International Conference on Computer Technology and Development 978-0-7695-3892-1/09 © 2009 IEEE DOI 10.1109/ICCTD.2009.15

[17] ArchitectureKhaldoon Al-Zoubi Gabriel Wainer "Using REST Web-Services Architecture for Distributed Simulation" 2009 ACM/IEEE/SCS 23rd Workshop on Principles of Advanced and Distributed Simulation

[18] Bianco P., Kotermanski R, Merson P., "Evaluating a Service-Oriented Architecture". Software Engineering Institute (SEI), Technical Report. CMU/SEI-2007-TR-015, September, 2007.

[19] Fielding, R.T., Architectural styles and the design of network-based software architectures. 2000, University of California, Irvine.

[20] Larson, K.D., The role of service level agreements in IT service delivery. Information Management & Computer Security, 1998. 6(3): p. 128-132.