International Journal of Advanced Science and Research
ISSN: 2455-4227
Impact Factor: RJIF 5.12
www.allsciencejournal.com
Volume 3; Special Issue 1; March 2018; Page No. 49-51

A Review on web service QoS requirements

[1] AG Choure, [2] JP Tonde, [3] BT Jadhav

[1] Research Scholar, Department of Computer Science, Shivaji University, Kolhapur, Maharashtra, India

[2] Assistant Professor, Department of Computer Science, MIT ACSC, Alandi, Pune, Maharashtra, India

[3] Professor, Department of Electronics and Computer Science, YC Institute of Science, Satara, Maharashtra, India

Abstract

With the wide use of web services, Quality of service (QoS) becomes an important factor in success of service provider. Popularity of web service depends on usability and utility of web service, both determine by QoS requirements. QoS plays an important role in searching the most appropriate web services for the users. In this paper we look at the various web service QoS requirements, which affect the performance of web services.

Keywords: quality of service (QoS), time-to-repair (TTR), scale invariant feature transform (SIFT)

1. Introduction

A web service is service offered by an electronic device to another electronic device, communicating with each other via the world wide web.W3C define web service as “A web service is a software system designed to support interoperable machine-to-machine interaction over a network.”

There are many standards such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description Discovery and Integration (UDDI) supporting web services.

QoS is a collection of non-functional parameters i.e.; availability, accessibility, Integrity, performance, reliability, regularity, security and scalability. Without knowledge of QoS factors, Service Consumer can select low quality or costly web service.

2. Web service Model

In a Web service model, a service provider offers Web services which provide business operations which can be deployed over the Internet, in the hope that they will be invoked by partners or customers. A Web service requester describes requirements in order to locate service providers. Publishing, binding, and discovering Web services are three major tasks in the model. Discovery is the process of finding Web services provider locations which satisfy specific requirements. Web services are useless if they cannot be discovered. So, discovery is the most important task in the Web service model.

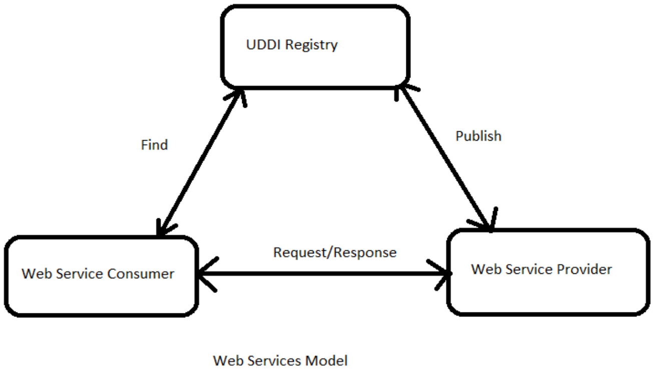

Fig 1

The Web service model in Fig. shows the interaction between a service requester, service providers, and a service discovery system.

a. The service providers offer Web services which provide functions or business operations. They are created by companies or organizations. In order to be invoked, the Web services must be described. This will facilitate discovery and composition. WSDL or service profile of semantic Web service is used to carry out this function.
b. The Web service requester describes requirements in order to locate service providers. Service requesters usually contain a description of the Web service, though it is not a Web service which can run on the Internet. The

requirements are usually described by WSDL, service template or service profile.

c. The Web service discovery or service registry is a broker that provides registry and search functions. The service providers advertise their service information in the discovery system. This information will be stored in the registry and will be searched when there is a request from service requester. UDDI is used as a registry standard for Web service.

Service Consumer not only needs to know what a web service is, but there should be knowledge of how web services can do. Current UDDI registries provide only syntactic information. But problem arises when consumers are keenly interested in knowing semantic information i.e.; response time, availability, content, throughput etc. and in order to provide solution of this problem another question arises that how service provider publishes Quality QoS information of similar web services in UDDI. After publishing QoS information in UDDI, how to update and save information in UDDI if need arises [2].

3. Web service QoS requirements

The major requirements for supporting QoS in Web services are as follows:

Availability

Availability is the quality aspect of whether the Web service is present or ready for immediate use. Availability represents the probability that a service is available. Larger values represent that the service is always ready to use while smaller values indicate unpredictability of whether the service will be available at a particular time. Also associated with availability is time-to-repair (TTR). *TTR* represents the time it takes to repair a service that has failed. Ideally smaller values of TTR are desirable [1]. Availability requirement is the probability that the Web service is up and in a readily usable state. High availability assures that there is the least amount of system failures or server failures even during the peak times when there is heavy traffic to and from the server and that the given service is available relentlessly at all times. Let us say the “down time” is when a system is not available. As the system is either available or unavailable, the remaining time after subtracting the down time can be termed as the “up time” that means the system is available [3].

Accessibility

Accessibility is the quality aspect of a service that represents the degree it is capable of serving a Web service request. It may be expressed as a probability measure denoting the success rate or chance of a successful service instantiation at a point in time. There could be situations when a Web service is available but not accessible. High accessibility of Web services can be achieved by building highly scalable systems [1]. A Web service’s accessibility is the measure of the probability that the client’s request to a Web service will be served. Accessibility is typically a measure of the success rate of a service instantiation at a given time. A Web service might not be accessible even though it is still available, because a system may be up and running but might not be able to process a request possibly because of the load it is experiencing. Accessibility in turn depends on how scalable the Web service system design is, because a highly scalable system constantly serves the request irrespective of the volume of the Web service requests. Accessibility is the ratio of the number of successful responses received from the server to the number of requests messages sent by the clients. It can be characterized as the degree of a system at which it is capable of responding to the user invocations of the service. Irrespective of type of acknowledgements received, (either negative/positive or correct/incorrect), accessibility can be calculated as a ratio of number of successful acknowledgements received to the total number of requests sent [3].

$$\text{Accessibility} = \frac{\text{Number of acknowledgements received}}{\text{Total number of requests sent}}$$

Integrity

Integrity is the quality aspect of how the Web service maintains the correctness of the interaction in respect to the source. Proper execution of Web service transactions will provide the correctness of interaction. A *transaction* refers to a sequence of activities to be treated as a single unit of work. All the activities have to be completed to make the transaction successful. When a transaction does not complete, all the changes made are rolled back [1]. Integrity requirement assures that any modifications to a Web service are performed in an authorized manner. Data integrity assures that the data is not corrupted during the transfer, and if it corrupted, it assures that there are enough mechanisms in the design that can detect such modifications. Data integrity is an important element to consider, because ignoring it may damage large software modules and create errors that are impossible to trace back. Data integrity is the measure of a Web service’s accurate transactional and data delivery abilities. The data messages that are received are verified to see if they have not been modified in transit. This can be done with techniques like checksum calculation or digital signatures. There are a number of tools in the market like SIFT, that can collect and monitor the data being sent and received between the communicating parties. These tools can be used to monitor the number of faulty transactions that are unidentified and the data messages that are received but with the checksum or hash that cannot be tallied. Data integrity is a Boolean value, meaning that data either has integrity or does not. There is no middle ground or a range that can specify how much integrity the data holds. Data integrity can therefore be calculated as the ratio of successful transactions to the total number of transactions [3].

Performance

Performance is the quality aspect of Web service, which is measured in terms of throughput and latency. Higher throughput and lower latency values represent good performance of a Web service. *Throughput* represents the number of Web service requests served at a given time period. *Latency* is the round-trip time between sending a request and receiving the response [1]. Obviously, the response time and the throughput depend on the workload that the Web server is experiencing at that time. Both latency and throughput can be measured by keeping track of the timestamps at the request time and response times. The latency of any request processing is the difference between the timestamps corresponding to the request time and the response time, and the throughput is given by the total number of requests divided by the elapsed time between the request time and the response time [3].

Reliability

Reliability is the quality aspect of a Web service that represents the degree of being capable of maintaining the service and service quality. The number of failures per month or year represents a measure of reliability of a Web service. In another sense, reliability refers to the assured and ordered delivery for messages being sent and received by service requestors and service providers [1]. In fact, Web Service Reliability (or WS-Reliability) is a latest specification for open, reliable Web service messaging. WS-Reliability ensures guaranteed delivery of messages, elimination and/or detection of duplicate messages and the right ordering of messages. The WS-Reliability can be embedded into SOAP as an additional extension rather than to a transport level protocol. This specification provides reliability in addition to interoperability, thus allowing communication in a platform and vendor-independent manner. The WS-Reliability specification defines a set of SOAP headers and instructions in SOAP envelopes that manage the message acknowledgements, message ordering etc. WS-Reliability makes asynchronous messaging a healthy choice because of the extra features that it provides like message acknowledgements and message ordering which in turn allow the communicating parties to be more independent of one another. Reliability determines the percentage of the times an event is completed with success. This numeral will provide an opportunity for the service consumers to expect the probability of a failure that can occur during a transaction. The count on failures is based on the number of dropped deliveries, duplicate deliveries, faulty message deliveries, and out-of-order deliveries. An event may either succeed or fail, and there is no middle ground in that issue. Therefore, total number of events will be the number of failures added to the number of successful events [3].

Regulatory

Regulatory is the quality aspect of the Web service in conformance with the rules, the law, compliance with standards, and the established service level agreement. Web services use a lot of standards such as SOAP, UDDI, and WSDL. Strict adherence to correct versions of standards (for example, SOAP version 1.2) by service providers is necessary for proper invocation of Web services by service requestors.

Security

Security is the quality aspect of the Web service of providing confidentiality and non-repudiation by authenticating the parties involved, encrypting messages, and providing access control. Security has added importance because Web service invocation occurs over the public Internet. The service provider can have different approaches and levels of providing security depending on the service requestor.

Scalability

Scalability requirement defines how expandable a Web service can be. Web services are being introduced to new interfaces and techniques every day and this makes keeping a Web service up-to-date a very important necessity. If the situation demands for more computing capabilities and servicing more requests, the system should be capable of supporting additional systems and newer technologies. The Web service should be able to handle heavy load while making sure that the performance in terms of response time experienced by their clients is not objectionable. The Performance Non-Scalability Likelihood (PNL) metric is a relatively newer technique to predict whether the system is going to be able to withstand the higher loads of traffic without affecting the performance levels. This metric is used to calculate the intensity of the loads at which the system cannot perform without degrading the response time and throughput. The calculation of PNL involves generating potential workloads and studying the behavior of the system which will be similar to how the system would react given such varying workloads. If the system crashes, the engineers will know that it is not scalable enough to accommodate potential future workloads and they could eventually organize an upgrade to the server capability. There can be two states in terms of their behavior. The behavior state can either be acceptable (0) or unacceptable (1). We can also denote the states with relative values in between 0 and 1 (in the interval [0, 1]) instead of constant values to point out the degree at which the system fails to behave acceptably.

4. Conclusions

Quality of services is an important requirement of web services for business-to-business transactions. The various QoS requirments such as availability, accessibility, integrity, performance, reliability, regulatory, security and scalability, need to be addressed in the implementation of Web service applications.

5. References

1. Anbazhagan Mani, ArunNagrajan-IBM Developerworks-Understanding quality of service for web services.
2. Amandeepkaur, RachanaSoni, Determining QoS of Web Services-A review, International Journal of Advanced Research in Computer Science and Software Engineering, 2016; 6(8). ISSN: 2277 128X
3. Rajendran T, Balasubramanie P. Analysis on the Study of QoS-Aware Web Services Discovery, Journal of Computing, 2009; 1(1). ISSN: 2151-9617
4. Bian WU, Xincai WU. A QoS-aware Method for Web Services Discovery, Journal of Geographic Information System, 2010; 2:40-44.
5. http://ieeeexplore.org/Xplore/home.jsp
6. http://nlist.inflibnet.ac.in
7. www.jntuworld.com
8. http://india.acm.org
9. http://citeseerx.ist.psu.edu/index