Improving Web Services Performance Using Priority Allocation Method

Dongjoon Kim, Sangkyu Lee, Sangyong Han and Ajith Abraham
Department of Computer Science & Engineering, Chung Ang Univ. Seoul, Korea
djkim@tysystems.com, sklee@archi.cse.cau.ac.kr, hansy@cau.ac.kr, ajith.abraham@ieee.org

Abstract

Recently many enterprises are adopting web services as the standard between heterogeneous software in the XML message based distributed environment to carry out businesses from B2C to B2B. For effective application of web services, differentiated service quality must be guaranteed. However, majority of the current web services do not differentiate quality of messages and the current web servers do not reflect the quality factors of the service level agreement settled between the service provider and user.

Our research analyzes the appropriate quality factor for the quality level where differentiated service is provided and suggests a method for assigning priorities to web service message processing processes based on these quality factors. The suggested method assigns the priority dynamically in order to satisfy the service level agreement as much as possible.

1. Introduction

Recently many enterprises carry out B2B business by adopting web services, which have settled as a standard in the XML message based distributed environment. In order to apply web services effectively, the service provider must be able to provide web services differentiated according to the various service levels. Furthermore, a Service Level Agreement (SLA) between the user and provider is necessary for specifying the various service levels [5]. The SLA is purported for defining the responsibility relation between the user and provider and to guarantee the quality of the service provided. Hence, the web service provider must be capable of guaranteeing the web service quality agreed by the SLA.

However, current web service technology standardization organizations have not yet established a standard for languages used in describing information on the web services quality evaluation factors and the evaluations themselves. At present only several vendors and universities are individually carrying out specifications and research [2, 3, 4, 6, 7].

Meanwhile, in IETF (Internet Engineering Task Format), DiffServ (Differentiated Service) is suggested for guaranteeing differentiated service quality at the network level [10]. But with DiffServ, the quality of end-to-end transfer via the Internet and the differentiated packet transfer function is implemented only in each separate section. Recently, to improve such disadvantages, some research initiatives on differentiating web services from web servers [8, 10, 11] are underway focused on finding the way for scheduling processes according to the user's request so that the service can be provided not just at the network level but also the application level. At present, most of the web servers apply the FIFO and static priority scheduling methods. But such methods are not capable of dynamically assigning priorities to fit each particular situation and as a result a starvation of low priority processes occurs or the performance evaluation of quality information that had been provided in the past is not reflected.

In order to provide differentiated web services through the web server, our research analyzes various web services quality factors to define the appropriate web services quality factors applied in differentiated web services. Then this quality information is used to assign priorities dynamically to the processes that process messages according to the particular situation in order to suggest a scheduling method that satisfies the service level agreement as much as possible.

2. Related Work

2.1 Service Level Agreement

An official agreement between the service provider and user is required to guarantee the defined level of the web service performance based on service quality factors. Such a service level agreement may be very comprehensive and at the same time very specific. The service level agreement may include the procedures to be followed by the provider and user in the case when either party fails to follow the agreement [2, 5].

IBM developed WSLA (Web Services Level Agreement)[3] for producing and monitoring service level agreements and standards for a web service environment. WSLA is a document of agreement defining the responsibilities of the web service provider and requester when using web services. It was designed considering the nature of the service level agreement environment. WSLA is composed of several elements as illustrated in Figure 1 [3].

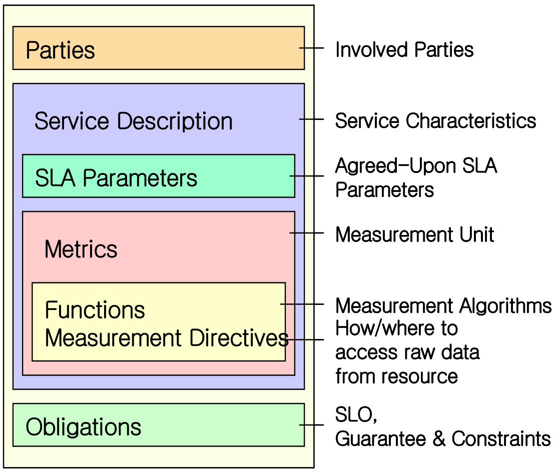


Figure 1. Structure of WSLA

2.2 Differentiated Service from Web Server

On the Internet, all packets are processed identically instead of differentiating packet transfer according to the type of service. Due to this aspect, it is difficult to guarantee quality through differentiated service. In order to solve this problem, DiffServ is suggested at IETF [9]. DiffServ classifies the packets to be transferred by the Internet into 8 to 64 service types according to the distinguishing method designated by the user, designates the processing function each packet exchanger must execute for each service type to allow differentiated Internet services.

But DiffServ does not guarantee the quality of end-to-end transfer via the Internet and the packet transferring function is implemented only for each separate section of the communication network. Due to the fact that the web server delay time is longer than the network delay time when the load is concentrated on the web server, many studies on the web server supporting differentiated services are underway [8, 10, 11]. Among them, one study suggests the WebQoS structure. Upon receiving an HTTP request, this structure classifies services based on the classification policy and differentiates the services according to the class they belong to [8]. The structure is depicted in Figure 2.

Different from the current web server processing of HTTP requests using the FIFO method, WebQoS has a connection manager added to discriminate classes by the priorities as shown in Figure 2. The HTTP requests classified and stored in each class are executed according to the scheduling policy to support differentiated service quality.

This paper suggests a scheduling policy improved using the WebQoS structure.

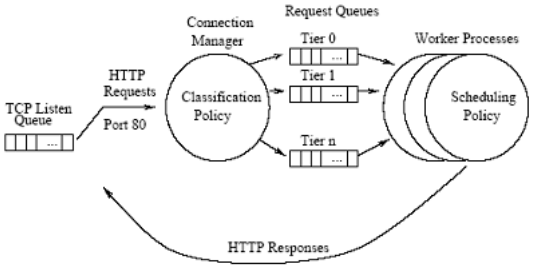


Figure 2. Structure of Web Server Supporting Differentiated Web Services [2]

3 Method for Assigning Priority using Web Services Quality Information

3.1 Used Web Services Quality Factors

According to the research report published by the National Computerization Agency in Korea [2], web services quality factors are classified into aspects of performance and safety, middleware, possibility of management and interoperability etc. However, in this research, the performance and safety quality factor is used because it is used the most in service level agreements and is easy to monitor. The formulas for calculating quality factors concerning performance and safety are given below.

- **Quality Factors Concerning Performance**
  - *Response Time* = Total time taken for message processing / Number of messages processed
  - *Throughput* = Total number of messages processed / Total processing time

- **Quality Factors Concerning Safety**
  - *Accessibility* = Message processing rate for certain message

- *Reliability* = Number of messages processed / Number of messages requested

As with quality factors concerning safety, even though the web services can be used, the services should be considered inaccessible if the performance is bad [1].

The processing rate in the accessibility formula is the same as the formula used for reliability (number of messages processed/number of messages requested). However, even when the server's throughput exceeds the maximum throughput, the accessibility must still be considered in order to guarantee the processing rate for advanced users.

In order to guarantee the web services quality factors defined above, a service level agreement must be settled between the user and provider. But since no standard exists for service level agreements, IBM's WSLA 1.0 [3] is applied in this research.

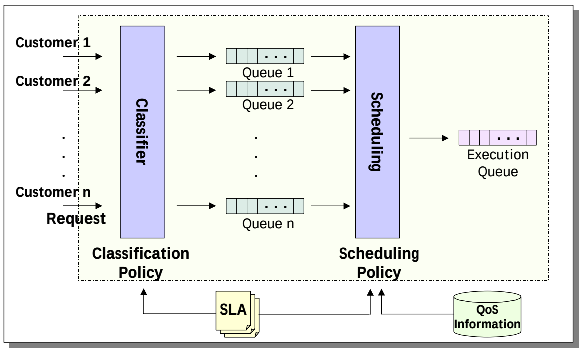


Figure 3. Model for Differentiated Web Server

3.2 Method for Assigning Priorities

The web server model supporting differentiated web services is illustrated in Figure 3. For this model, the service level agreement must be settled between the web service user and provider, and this agreement must contain the web service quality information measurements mentioned in Section 3.1. The web service message requested by the web service user is classified according to the classification policy and the execution queue is allowed to process messages up to the maximum number guaranteeing best processing performance. When requests exceed the limit, the remaining requests are put on standby at each buffer queue. The messages to be sent from the buffer queue to the execution queue are determined according to the priority allocation policy.

Basically, two modules are required for constructing a differentiated web services supporting web server. The module for classifying each message and the module for assigning priorities to the classified messages. The message classification module classifies messages according to the service level agreement and discriminates the messages by the user IP address or the URL form. Meanwhile, the priority assignment module can be applied by using the accumulated web service quality information.

When the execution queue in the priority assigning module is saturated, messages standing by at each classified queue are assigned priorities and must be sent to the execution queue starting with the message with the highest priority. For this study, the priority here was determined by comparing the monitored value of the web services quality factor extracted in Section 3.1 and the quality factor value settled in the service level agreement in order to satisfy the service level agreement at a high level. Basically, the FIFO method is applied in the classified queues so that the first message that is received can be provided the first within the queue. When there are messages standing by in the classified queues, objects for comparison with the first arriving messages of each queue are selected and quality information values are calculated for these messages. The message with the smallest value is assigned the highest priority.

The response time is a factor the user can directly feel during quality factor evaluation, and it is often one of the most important factors taken into account in service level agreements. In order to apply this response time, the value can be calculated using (1).

$$V_{RT} = RT_{SLA} - QT - RT_{mean} \qquad (1)$$

In (1), RT_{SLA} is the value of the average response time agreed on by the user and provider in the service level agreement, QT is the time the message stood by in the queue and RT_{mean} is the measured average response time. Basically, the message with the lowest V_{RT} value, in other words, having the least time left till the time limit, is assigned with the highest priority. In this aspect, this method is similar to the EDF method where the message with the closest time limit is assigned to the highest priority.

This paper suggests other quality factors as well as the response time. The satisfaction rate of each quality factor is calculated in order to improve the overall satisfaction rate including the response time satisfaction rate and the satisfaction rates of other factors. The satisfaction rate (V_T) for amount of message processed is calculated by using (2).

$$V_t = T / T_{SLA} \qquad (2)$$

In (2), T_{SLA} is the throughput agreed on in the service level agreement and T indicates the measured throughput. Furthermore, the accessibility and reliability satisfaction rates (V_{Ac} and V_R respectively) can be calculated as.

$$V_{Ac} = A_C / A_{cSLA} \qquad (3)$$

$$V_R = R / R_{SLA} \qquad (4)$$

In (3) and (4), A_{cSLA} and R_{SLA} are the accessibility and reliability values agreed between the user and provider in the service level agreement and A_C and R indicate the accumulated accessibility and reliability respectively.

The priorities are assigned using the values calculated by formulas (1) to (4). The values to be applied are calculated as below.

$$V_P = V_{RT} \times [(V_T + V_R) / 2] \qquad (5)$$

Equation (5) is used for the first messages arriving and standing by in each standby queue to calculate and compare the V_P value and thereby assign the highest priority to the message with the smallest value. This is based on the response time. Compared to the throughput and reliability values with the service level agreement values, the service level agreement satisfaction rate rises as time passes.

However, it is not easy to apply (5) directly. When the V_P values are negatives, the priority can be reversed if the priority is settled based on the minimum value. As so, the V_P values must be corrected as shown below. Here, the *TimeOut* value is used for making the V_{RT} values into positives. The sum of V_{RT} and *TimeOut* adopts the value of ln to be compared with other V_T and V_R values.

$$V_P = \ln(V_{RT} + TimeOut) \times [(V_T + V_R) / 2] \qquad (6)$$

Using (6), the service level agreements of all requesters can be satisfied within the server's processing range. But if the throughput exceeds the fixed limit, in other words, if the number of requests is larger than the server is capable of processing, someone must give up the service. Here, the service requesters who have contracted the accessibility value in the service level agreement are regarded as advanced users and the number of requests per second when the server performs the best is set as the threshold value. Then, (7) is applied to the messages of users who have contracted the accessibility value and (8) to messages of ordinary users. When the number of requests exceeds the threshold value, the reliability value is applied to guarantee the advanced user the maximum performance and the value of throughput is applied to guarantee the ordinary user the minimum performance.

$$V_P = \ln(V_{RT} + TimeOut) \times V_{Ac} \qquad (7)$$

$$V_P = \ln(V_{RT} + TimeOut) \times V_T \qquad (8)$$

In this study, (6), (7) and (8) are suggested for assigning priorities. But if factors other than the response time are omitted, other equations excluding the particular factors can be applied instead.

4. Experiment and Evaluation

In this Section, we present some experiment results to demonstrate the suggested priority assigning method for differentiated web services and the performance is compared with the EDF method.

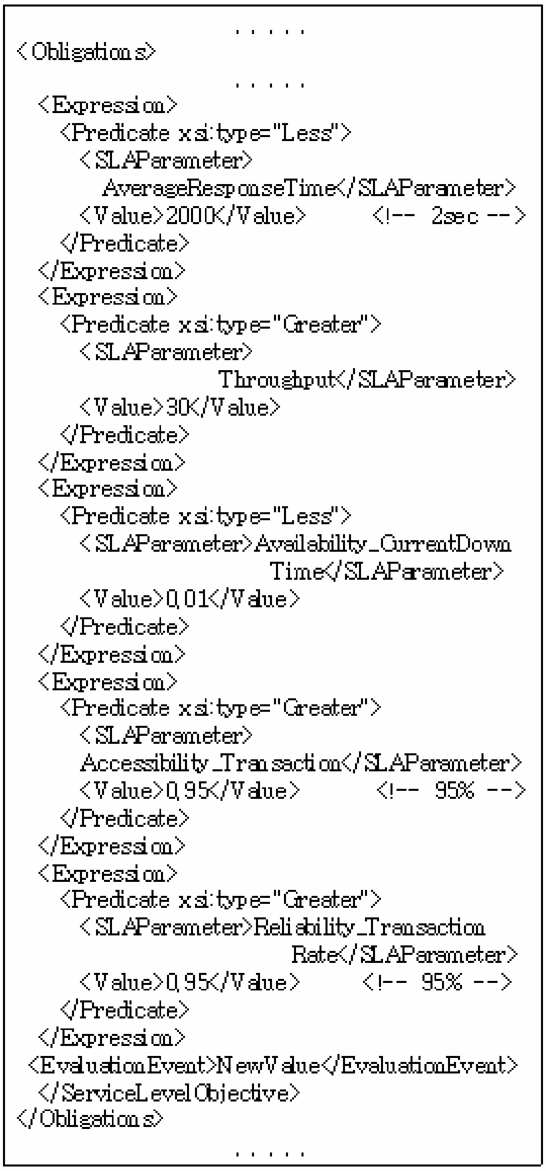
4.1 Experiment Scenario

The network situation is excluded and an application level simulation is executed in the LAN environment. The processing time for each single web service is set at approximately 0.06 seconds, and the reliability, response time and throughput were measured.

Here, the web service messages were classified into three types. Part of WSLA used in the experiment is depicted in Figure 4. As observed in Figure 4, For messages placed in Class 1, it is supposed that the service level agreement defines the response time as two seconds or less, throughput as 30 or more and reliability as 95% or higher. In this experiment, the service quality is guaranteed in the Class 1 > Class 2 > Class 3 order. For messages sorted into Class 3, it is supposed that the service level agreement defines he response time as 3.5 seconds, throughput as 20 and reliability as 80%. The same type of WSLA suggested in Figure 4 can be created.

4.2 Results and Analysis

As shown in Figures 5, 6 and 7, the maximum performance value capable of satisfying all of the service level agreements in the test environment is when the number of request per second is 120. Figure 5 shows the response time deviation between the EDF method and the suggested method. Both methods satisfy the service level agreement when the number of request per second is 120. As evident, with reference to reliability and throughput, the EDF method does not satisfy the service level agreement sufficiently when the number of requests per second is 120. This means that the number of requests per second must stay under 120 to have the EDF method satisfy all of the service level agreements. On the other hand, the suggested method is capable of satisfying all of the service level agreements even when the number of requests per second is 120.



```
                    . . . . .
<Obligations>
                    . . . . .
  <Expression>
    <Predicate xsi:type="Less">
      <SLAParameter>
        AverageResponseTime</SLAParameter>
      <Value>2000</Value>      <!-- 2sec -->
    </Predicate>
  </Expression>
  <Expression>
    <Predicate xsi:type="Greater">
      <SLAParameter>
                   Throughput</SLAParameter>
      <Value>30</Value>
    </Predicate>
  </Expression>
  <Expression>
    <Predicate xsi:type="Less">
      <SLAParameter>Availability_CurrentDown
                        Time</SLAParameter>
      <Value>0.01</Value>
    </Predicate>
  </Expression>
  <Expression>
    <Predicate xsi:type="Greater">
      <SLAParameter>
      Accessibility_Transaction</SLAParameter>
      <Value>0.95</Value>       <!-- 95% -->
    </Predicate>
  </Expression>
  <Expression>
    <Predicate xsi:type="Greater">
      <SLAParameter>Reliability_Transaction
                          Rate</SLAParameter>
      <Value>0.95</Value>      <!-- 95% -->
    </Predicate>
  </Expression>
 <EvaluationEvent>NewValue</EvaluationEvent>
  </ServiceLevelObjective>
</Obligations>
                    . . . . .
```

Figure 4. WSLA of Class 1

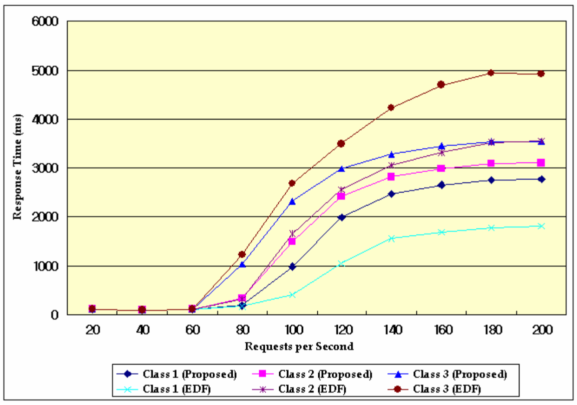


Figure 5. Response time of EDF method and proposed method

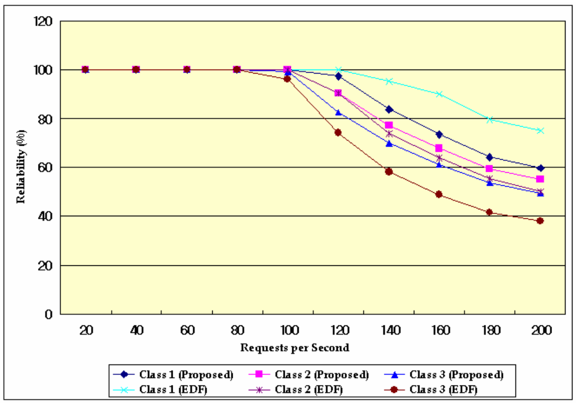


Figure 6. Reliability of EDF method and proposed method

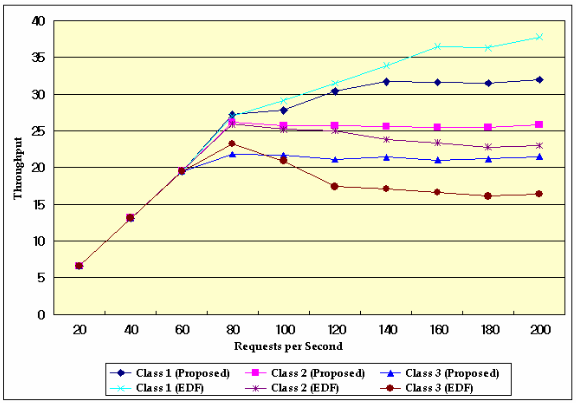


Figure 7. Throughput of EDF method and proposed method

The number of requests per second exceeding 120 can be regarded as exceeding the server's maximum throughput. Hence, all of the service level agreements cannot be satisfied and the requests from ordinary users can only be ignored in order to answer the advanced users' requests. Figure 8 suggests the reliability depending on whether accessibility is considered or not. As observed here, the advanced user can be guaranteed the best performance by using accessibility.

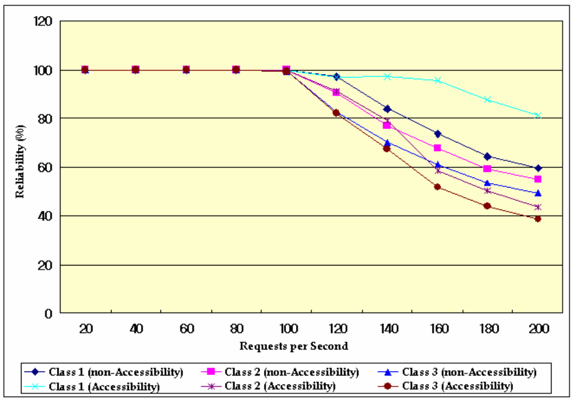


Figure 8. Reliability considering accessibility

5. Conclusions and Future Work

Recently many enterprises are adopting web services, which have settled as the standard between heterogeneous software in the XML message based distributed environment, to carry out businesses from B2C to B2B. For successive business with web services, providers should be capable of offering differentiated services.

However, differentiated web services on the network level do not guarantee end-to-end transfer quality whereas the network environment is improving. Other than that, since web services send messages using XML, XML parsing is inevitable and the majority of current web services use web servers for communication.

For such reasons, this paper suggests a priority assignment method for providing differentiated web services on the web server, not at the network level but at the application level. For implementing this method, the web services quality factors most used in web services were analyzed to extract the factors required for assigning priorities. Using these quality factors and IBM's WSLA, the method for assigning priorities to messages in the web server is produced.

Since this study did not take network speed into account, a method that is compatible with differentiated services at the network level such as DiffServ should be implemented.

6. References

[1] Park, S.K., "Web Service Availability Measurement Model based on Service Degradation Factor," Ph.D. Thesis, Chonnam National University in Korea, 2003.2.

[2] "A Study on Technical Trends and Deployment Strategies of Web Service Quality Management," Research Report, National Computerization Agency in Korea, 2003.12.

[3] A. Keller and H. Ludwig, "The WSLA Framework: Specifying and Monitoring of Service Level Agreements for Web Services," IBM, 2003.6.

[4] Anbazhagan Mani and Arun Naqarajan, "Understanding quality of service for Web services," IBM, 2002.1.

[5] Asit Dan, Heiko Ludwig, and Giovanni Pacifici, "Web services differentiation with service level agreements," White Paper, IBM, 2003.5.

[6] Daniel A. Menasce, "QoS Issues in Web Services," IEEE Internet Computing, 2002.12.

[7] Liangahao Zeng, Boualem Benatallah, Malon Dumas, Jayant Kalagnanam, and Quan Z. Sheng, "Quality Driven Web Services Composition," www2003, 2003.5.

[8] R. Bhatti and R. Friedrich., "Web Server Support for Tiered Services," IEEE Network, 13(5):64-71, 1999.9.

[9] S. Blake, D. Black, M. Carlson, E. Davies, Wang Z. and W. Weiss.m "An architecture for differentiated services," IETF RFC 2475, 1998.12.

[10] Xiangping Chen and Prasant Mohapatra, "Performance Evaluation of Service Differentiating Internet Servers," IEEE Trans. on Computers, 51(11):1,368-1,375, 2002.11.

[11] Xiaobo Zhou, Yu Cai, Ganesh K. Godavari, and C. Edward Chow, "An Adaptive Process Allocation Strategy for Proportional Responsiveness Differentiation on Web Servers," IEEE International Conference on Web Services (ICWS 2004), pp. 142-149, 2004.7.