

Skyline Based Quality Service Selection Through Aggregated Response Matrix Formulator

Balika J. Chelliah* and K. Vivekanandan**

ABSTRACT

In the Technological World, there are different types of application run over the internet, especially when it comes to a Business oriented application; there are many QOS –sensitive application over the internet. The aim is to provide the user-response with high QOS rating. Understanding the QOS requires exact measurement and reporting of throughput over time, along with a periodic review of the application requirements vs. the actual service provided. Different types of jobs are gathered and satisfied by a proper workflow management system wherein the target of the QOS is to choose the required web service to serve the customer needs. The scenarios in the existing approach seems to have set of functionalities to serve the request using service-oriented architecture that in turn fails to choose the right quality of service from the functionality. QOS Modeling is considered to be the important factor to identify the appropriate functionality in order to serve the request. This paper predefines a workflow and the multi handling technique to choose the appropriate service to serve the request. A combinatorial matrix has been generated to rate the QOS for its performance. In turn, a matrix with User Vs Response time quality matrix is formulated through the time manipulations on quality service response. The response time is further breakup to Process in Queue and Process Time (Time for completing the request). This provides a clear picture on where the process is stroked up. In advance, the process in queue is further disseminated into Queuing process delay (Waiting time in Queue), Setup process Delay(Initializing the method for setup towards execution). Fine grained response time information of a service will provides an exact QOS nature in SOA world.

Keywords: ARMF(Aggregated Quantified Response Time Matrix Formulation) Quality of Service, Service oriented application, web services, User-response matrix, Matrix Formulation, QOS Modeling.

1. INTRODUCTION

Organizations operating in modern markets, such as e-commerce activities and distributed Web services interactions require QoS management. Control of quality on the product fulfills the customer expectation.

The above Fig. 1 depicts about the QOS- sensitive application over the internet with functional flow of various activities such as business and Quality component with required specifications. Business Process Initiation component namely Business Entity comprises of set of information about the related web service. The information holds a conceptive specification of relative web services supported by the web application collectively in a well detailed manner. An Entity Service is also known to be Entity-Centric Business service or Business Entity. It bases its functional boundary and context on one or more related business entities. Another Business component, Business Entity Information of Business Service depicts the descriptive information about the particular web service supported by the web application. The Business entity Information receives the specification from Publisher assertion that holds the information of the related web service parties.

* Assistant professor, Department of Computer Science & Engineering, SRM University, Email: balika888@gmail.com

** Department of Computer Science & Engineering, Pondicherry Engineering College, Pondicherry.

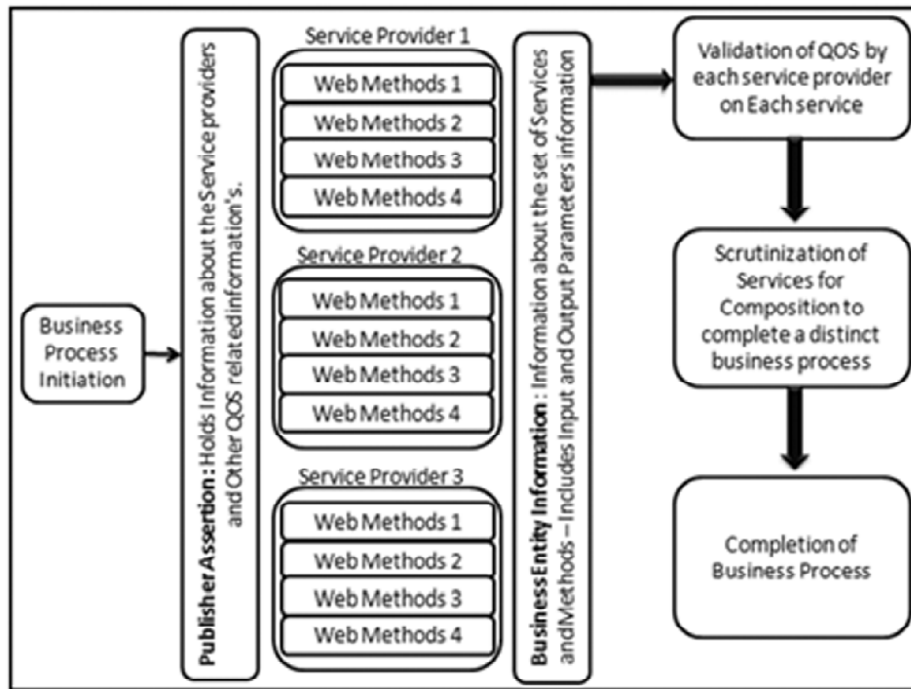


Figure 1: QOS Model for Business Processes

```

<!--element name="AddNewUserResponse">
  <complexType>
    <sequence>
      <!--element minOccurs="0" maxOccurs="1" name="AddNewUserResult">
        <complexType>
          <sequence>
            <!--element minOccurs="0" maxOccurs="unbounded" namespace="http://www.w3.org/2001/XMLSchema" processContents="lax"/>
            <!--element minOccurs="1" namespace="urn:schemas-microsoft-com:xml-diffgram-v1" processContents="lax"/>
            <sequence>
              <complexType>
                <sequence>
                  <complexType>
                    <sequence>
                      <!--element minOccurs="0" maxOccurs="1" name="stfName" type="string"/>
                      <!--element minOccurs="0" maxOccurs="1" name="stlName" type="string"/>
                      <!--element minOccurs="0" maxOccurs="1" name="UName" type="string"/>
                      <!--element minOccurs="0" maxOccurs="1" name="Pwd" type="string"/>
                      <!--element minOccurs="0" maxOccurs="1" name="DOB" type="string"/>
                      <!--element minOccurs="0" maxOccurs="1" name="Mobile" type="string"/>
                    </sequence>
                  </complexType>
                </sequence>
              </complexType>
            </sequence>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
-->
<!--element name="Propo_AddNewUser">
  <complexType>
    <sequence>
      <!--element minOccurs="0" maxOccurs="1" name="stfName" type="string"/>
      <!--element minOccurs="0" maxOccurs="1" name="stlName" type="string"/>
      <!--element minOccurs="0" maxOccurs="1" name="UName" type="string"/>
      <!--element minOccurs="0" maxOccurs="1" name="Pwd" type="string"/>
      <!--element minOccurs="0" maxOccurs="1" name="DOB" type="string"/>
      <!--element minOccurs="0" maxOccurs="1" name="Mobile" type="string"/>
    </sequence>
  </complexType>
</element>
-->

```

Figure 2: Service Model and Binding Data Associated

The Quality component holds Quality information those possess the various Quality factors such as Performance, Reliability, Scalability, Capacity, Robustness, Exception Handling, Accuracy, Integrity, Accessibility, Availability, Interoperability, and Security of the services in a detailed descriptive form. QoS envelops a wide range of techniques in order to match the requirements of service requestors with the service provider’s on the basis of the available network resources. By QoS, we refer to non-functional properties of Web services such as performance, reliability, availability, and security. The Business information collectively gathered from the publisher is comparatively verified with the Quality information that the service can provide for a request. Binding Template is an UDDI Data structure that organizes information for particular instances of service types. A Service Model is an abstraction of service type technical specification that organizes the service type information and provides an accessibility privilege in the registry database. It is also a unique identifier with alphanumeric character. In a Business Environment, if the requirement is to make their specification-compliant services available to the registry, a reference to the Service Model Key is included for the service type in their binding Template data(Fig 2).

In the above performance chart (Fig 5), we are able to identify the execution time of Service1 is lesser and typically the option of preference will be Service1.

Matrix for User Access Vs Execution Time (Authentication Service)

<i>Services Number</i> <i>of Users</i>	<i>Authentication</i> <i>Service 1</i> <i>Execution</i> <i>Time</i>	<i>Authentication</i> <i>Service 2</i> <i>Execution</i> <i>Time</i>
	Iteration 1 for 100 Users	3862
Iteration 2 for 100 Users	3799	6147
Iteration 3 for 100 Users	3722	6872
Iteration 4 for 100 Users	2907	5879
Iteration 5 for 100 Users	2919	6204

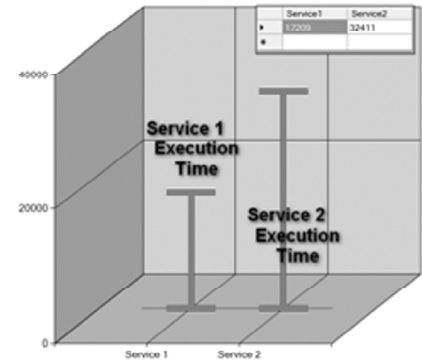


Figure 3: User Access Vs Execution Time (Service 1)

Matrix for User Access Vs Execution Time (MAC Service)

<i>Services Number</i> <i>of Users</i>	<i>MAC</i> <i>Service 1</i> <i>Execution</i> <i>Time</i>	<i>MAC</i> <i>Service 2</i> <i>Execution</i> <i>Time</i>
	Iteration 1 for 100 Users	8817
Iteration 2 for 100 Users	7218	7938
Iteration 3 for 100 Users	8598	7799
Iteration 4 for 100 Users	7368	8350
Iteration 5 for 100 Users	8025	8248

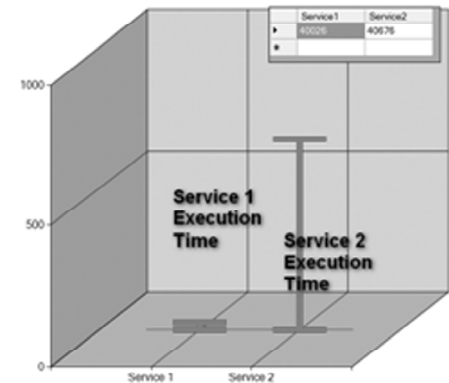


Figure 4: User Access Vs Execution Time (Service 2)

Matrix for User Access Vs Service Availability Time (Authentication Service)

<i>Services Number</i> <i>of Users</i>	<i>Authentication</i> <i>Service</i> <i>1</i> <i>Availability</i>	<i>Authentication</i> <i>Service</i> <i>2</i> <i>Availability</i>
	Iteration 1 for 100 Users	55
Iteration 2 for 100 Users	32	97
Iteration 3 for 100 Users	23	113
Iteration 4 for 100 Users	27	109
Iteration 5 for 100 Users	28	117

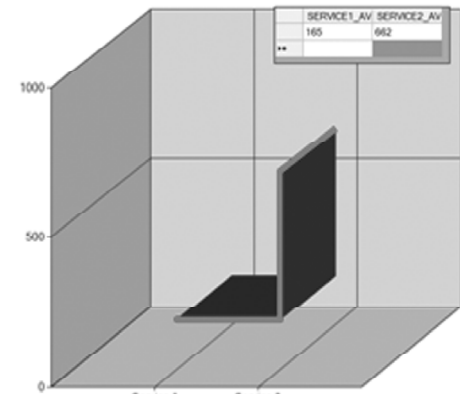


Figure 5: User Access Vs Service Availability Time (Service 1)

In the above performance chart (Fig 6), we are able to identify the execution time of Service1 is lesser and typically the option of preference will be Service 1.

In the above performance chart (Fig 7), we are able to identify the Availability time of Service1 is lesser and typically the option of preference will be Service 2.

QoS based design visualizes the business process effectively. It is designed using various QoS metrics that concentrate on Response Time, Efficient Caching, Aborted page, size of data, Client proxies accessibility. QoS-based selection and execution selects an appropriate workflow to satisfy the request of the customer requirement. The main concentration of the QoS- based Selection and Execution is to serve the client requirement that is an indefinable assignment for the web service operator.

**Matrix for User Access Vs Service Availability Time
(MAC Service)**

<i>Services Number of Users</i>	<i>MAC Service 1 Availability</i>	<i>MAC Service 2 Availability</i>
	Iteration 1 for 100 Users	183
Iteration 2 for 100 Users	82	47
Iteration 3 for 100 Users	90	46
Iteration 4 for 100 Users	82	54
Iteration 5 for 100 Users	91	54

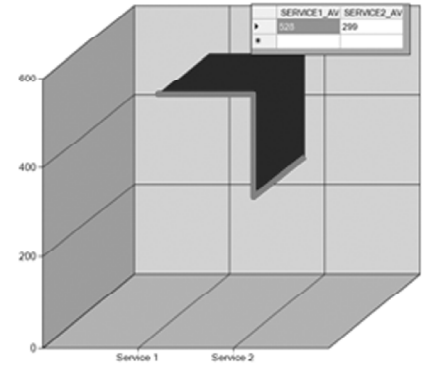


Figure 6: User Access Vs Service Availability Time (Service 2)

In the below performance chart, we are able to identify the Availability time of Service1 is higher and typically the option of preference will be Service1. In peculiar cases like specific method is good when compared to the other methods. A overall performance of the whole service is validated and the decision will be taken accordingly. This depends on the priority and occurrence of service methods for a business process.

QoS monitoring is performed by checking the actual service performance against the QoS stated in Service Level Agreement (SLA).It is a kind of checking against the worthiness of paying and utilizing the web service determining whether the user's expectations are met. It also triggers a set of adaption strategies when undesired metrics are met.

2. LITERATURE SURVEY

Chen, Paik, and Hun proposed a new semantics based framework (1) in which the services are functionally linked across the web to form a global social service specific principles and in addition the external service are linked via the web. Linked data principles were designed to construct global social service network to interconnect isolated service island and provides a link-as-you-go concept of service to service. Paper provides the service selection on multiple factors such as based on business functionality parameters passed to the service and bit of QOS preferences and sociability preferences. Focus on QOS selection is not emphasized and main focus is the selection based on semantic information passed and needed. Ahmed, Wu and Zheng suggest an optimal path for service composition by QOS manipulation using Hidden Markov models (HMM)(2). The principle behind the suggested model is to build a directed graph among the hidden states and predict the response time and an optimal web service can be selected based on the hidden states behavior. Author suggestion on other QOS factors such as availability, reliability may provide more focus on the web service selection which is missed out in the research. Garcia Llinas and Nagi solved the multi constraint heuristic model (Natured high restriction levels and strong QOS needs) with the concept of "potential possibility"(3). Utilization factor is manipulated for the service and the possible path for the flow is decided to reduce the service selection time and the overall response time. In turn, this becomes the possible path concept which becomes a prediction and it may incur severe impact in case of predicting the failure services in business critical composition models. Hwang, Hsu, and Lee (4) specified the probability mass function on the composite service is high whereas the execution time is reasonable. This can be achieved by considering the QOS parameters such as Reliability metrics, Fidelity Metrics and Response Time. The analysis was done on variety of business models such as sequence, parallel, Exclusive choice and Loop on models. Initial data assignments for this metrics and incremental adjustment of the data for each model are achieved through simulated annealing technique. Focus on exact delay time with respect to response time can be incorporated which provides the base for our research. Silic, Delac, and Srbljic (5) emphasize three important clustering base parameters which includes user specific, service specific and environmental specific models. The data collected from these models were clustered using the well known K-means-clustering algorithms and the derived structure is linear regression predictor.

The input parameters such as user location, service location, service load and service class were given into the hypothesis function for linear regression. The aggregated resultant was considered for finalizing the web services. Fine grained information's such as server response, queue delay and process delay can be included to cover up the full functional QoS detection mechanisms. Chen, Huang, Lin and Hu provide a sequential steps using distributed partial selection algorithm (6) in which the constraints are validated to identify the Pareto set for service compositions. Afterwards, a further scrutinization of validation, pareto set generation and composition will have further tune-up. The process is carried out until we get the best composition sequential model for business critical SOA. The overall system is understandable and provides an optimized solution but it will have a major delay due to micro level further tunings.

3. PROPOSED SYSTEM

Web Services is the most likely connection technology of service-oriented architectures. Service-oriented architecture provides base level architecture to achieve rapid service composition and functional processes. It is set to be a subsequent request and subsequent response processing connection between the service consumer and service provider. It is the basic idea in the service-oriented architecture that the communication involves with simple data passing or two or more service coordinating sequential to pass a data according to the request. Responding a request without considering the QoS factor may not satisfy the customer though their requirement has been fulfilled. In order to achieve QoS requirements in different system, it is possible to select an appropriate set of concrete services and compose them to achieve the QoS goals. QoS Modeling identifies the factors that implicate the web service identification for the appropriate functional requirement that service serves. QoS serves as a major factor for identifying judicious candidate web services and web service compositions with identical service functionality. The complete SOA architecture works on the underlying business intelligence SOAP protocol (Fig 3).

4. INTERFACE VIEW POINTS OF SOA ARCHITECTURE

There are three viewpoints to predict the QoS.

1. *Behavioral interface* -> Identifying the Behavior nature and availability of the of the Individual service
2. *Choreography* -> Accessing nature and Properties of the web service.
3. *Orchestration* -> Architecture, Location and Error prone nature of the web service.

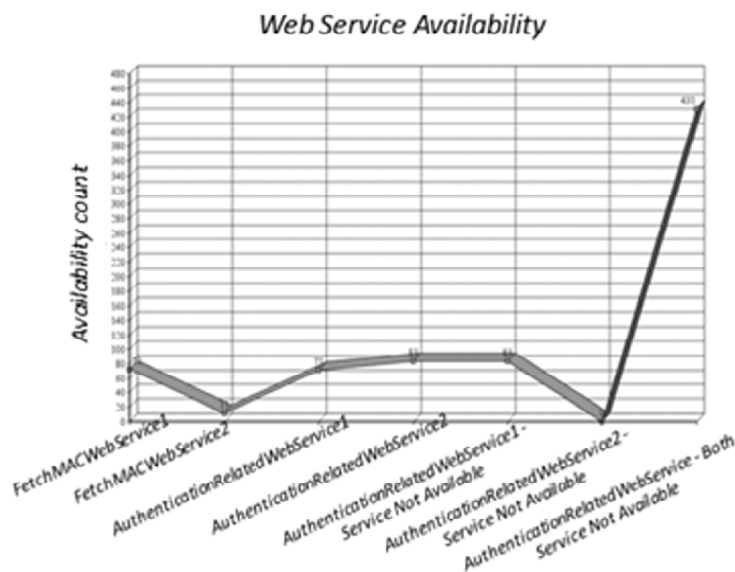


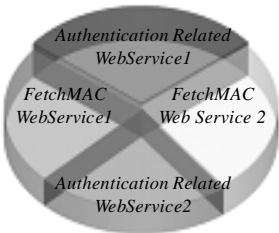

Figure 7: Service Availability Chart

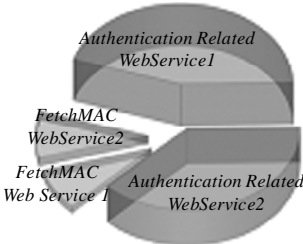

Web Service

<i>Web Service Calls</i>	<i>Availability Count</i>	<i>Availability Result</i>
FetchMAC Web Service 1	72	Web Service 1 Availability
FetchMAC Web Service 2	11	Web Service 2 Availability
Authentication Related Web Service 1	71	Web Service 1 Availability
Authentication Related Web Service 2	83	Web Service 2 Availability
Authentication Related Web Service1 - Service Not Available	83	Web Service 1 Non-Availability
Authentication Related Web Service 2 - Service Not Available	0	Web Service 2 Non-Availability
Authentication Related Web Service - Both Service Not Available	431	Web Service 1& 2 Non-Availability

These are set to be the functional artifact of the web service used in the QoS Modeling Phase. In the current scenario of Web service research, various QoS factors are analyzed handling with variety of technique(Fig 8). In this paper, a new approach of Fine grained information of delayed time is utilized instead of response time manipulation.

The response time manipulation will provide the request/response nature of the service in general. The Fine grained information of delayed time (Processes in queue) and Process time (Time for completing the request) of the projected approach (eqn. 2) will provide more visualization on the Quality of the Services in a system.

<i>Web Service Call</i>	<i>Set Up Delay</i>	<i>Formulation Implemented</i>	<i>Graphical Format</i>	<i>Process Flow</i>
Fetch MAC Web Service 1	491	$\sum_0^{cnt} \frac{SPD_{(t)}}{R_{cnt}}$		
Fetch MAC Web Service 2	609			
Authentication Related Web Service 1	545			
Authentication Related Web Service 2	483			

<i>Web Service Call</i>	<i>Queue Process Delay</i>	<i>Formulation Implemented</i>	<i>Graphical Format</i>	<i>Process Flow</i>
Fetch MAC Web Service 1	31	$\sum_0^{cnt} \frac{QPD_{(t)}}{R_{cnt}}$		
Fetch MAC Web Service 2	36			
Authentication Related Web Service 1	170			
Authentication Related Web Service 2	144			

$$DT_{(t)} = QPD_{(t)} + SPD_{(t)} > \text{eqn. 2}$$

DT_(t) => Delayed Time

QPD_(t) => Queue Process Delay

SPD_(t) => Setup Process Delay

Web Service Call	Delayed Time	Formulation Implemented	Graphical Format	Process Flow
FetchMACWebService1	522	$\sum_0^{cnt} DT_{(t)} =$		
FetchMACWebService2	645	$\sum_0^{cnt} \frac{SPD_{(t)}}{R_{cnt}}$		
Authentication Related WebService1	715	$+ \sum_0^{cnt} \frac{QPD_{(t)}}{R_{cnt}}$		
Authentication Related WebService2	627			

Web Service Call	Process Time	Formulation Implemented	Graphical Format	Process Flow
Fetch MAC Web Service 1	107	$\sum_0^{cnt} \frac{PT_{(t)}}{R_{cnt}}$		
Fetch MAC Web Service 2	139			
Authentication Related Web Service 1	30			
Authentication Related Web Service 2	41			

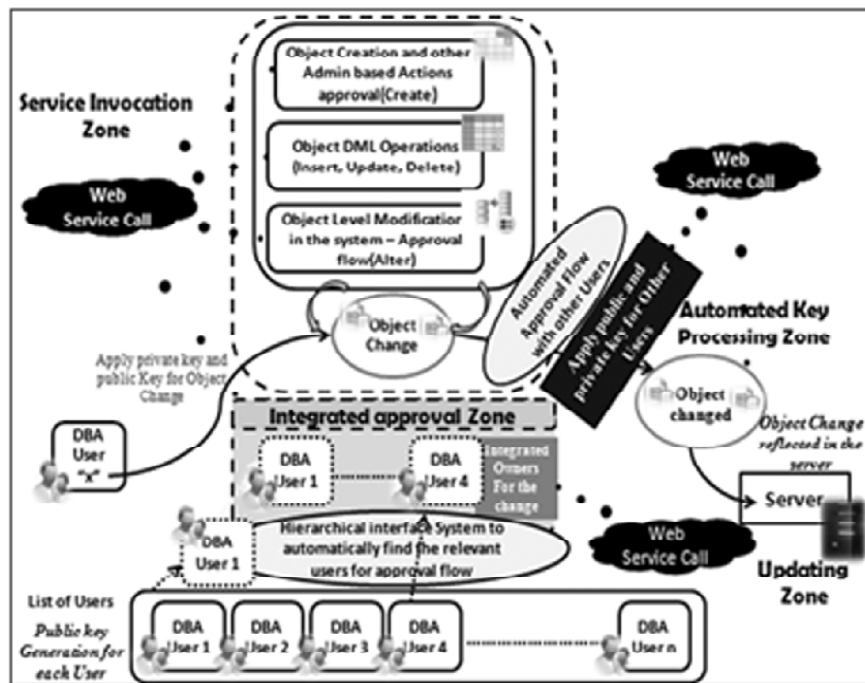


Figure 9: Proposed Architecture for Multi Hand Administration

The delayed process time is further decayed into Queuing process delay (Waiting time in Queue), Setup process Delay (Initializing the method for setup towards execution) – This particular delay is associated with Synchronization process delay too.

The architectural representation illustrate about the web service call that seeks privilege on the approval of owner. Architectural representation can be splitted into Service Invocation Zone, Integrated Approval Zone, Automated Key Processing Zone and Updating Zone. In the invocation zone, Web service call from the requestor is initiated. The type of web service call is analyzed over this zone and the need for the transfer from one web service to another is predicted. If the requested data is under the non privileged accessibility, in order to attain privilege from the owner of the source.

The fig 9 depict about the architectural diagram of the multi handling web service call. call.

The integrated approval zone is the data accessibility approval phase from the owner of the source requested. Authentication for the approval can be arrived the Keys navigated between the users and this can be generated using the above Algorithm (1) specified. This play an important role since the owner of the source has an eye on the data shared to the requestor. Since there may be a set of users under the system, integrated approval zone utilizes a hierarchical interface system to find the relevant users automatically for approval. A key based object modification is carried out where the private key of the web service invocation zone combines with public key of the owner of the source after approval from integrated approval zone. The object modification is reflected in the server after the secure approval by the key based object modification approach.

ALGORITHM 1 – Key Generation Functionality

INPUT: Key generation Method with two string Parameters *cboTable* and *cbocol*

OUTPUT: Algorithm retruns a key in the format of string

1. Generate a byteArraycboTableparameter.
Prerequisite – Data is converted into bytes streams before assigning.
2. Create an object *tripleDES*for the*TripleDESCryptoServiceProvider*class.
3. Assign the value *cbocol* to the parameter *modified*.
4. If the length of *cbocol*is less than*UpperPermissibleLimit*
5. Loop through the *modified* variable and assign *null or empty value* into the variable.
6. End
7. Else
8. Fetch the *permissiblevalue* of the string length and assign to the variable *modified*.
9. End
10. Set the Key for the *tripleDE* Subject by converting the *modified* variableintoUTF8 Encoded byteStream.
11. Set the mode and Padding option for the *tripleDES*object as *Electronic Code Book*and *Public Key Cryptographic Standard* kind of Ciphering.
12. Create the CryptoTransformInterface *cTransform*by creating the encryptor for the Defined*tripleDES*object and assign it.
13. The created cryptotransform*cTransform*can be TranformBlockwith the parameter as the InputArraybyteStream. *resultArray*is the Output bytestreamcreated from the Outcome of*CTransform*
14. Cleanup the *tripleDES*Object by running the relevant destructor class.
15. Initialize the variable *myEncryptedText*with the typecasted valueof Base64String of*resultArray*

16. Initialize the `stringBuilderrandomText` and Initialize another variable `code` of type String.
17. For Looping for `MaxPermissibleLength` of the Code-Depends on Business Requirements
18. Amend the `stringBuilderrandomText` with the `myEncryptedText` character values randomly.
19. End
20. Assign the value of `randomText` to the variable `code` and return it

4.1. Experimental Analysis

Varieties of experiments were made on the various functionality web services zone. And the performance in different perspective was tracked down. Accumulation of services execution time for the request were manipulated and below is the result set of the performance of the web services. As a whole, greater the execution time of the services lesser the performance of the services.

$$T_{(t)} = DT_{(t)} + PT_{(t)} > \text{eqn. 1}$$

$T_{(t)}$ => Response Time

$DT_{(t)}$ => Delayed Time

$PT_{(t)}$ => Process Time

Web Service Call	Set Up Delay	Formulation Implemented	Graphical Format	Process Flow
Fetch MAC Web Service 1	629			
Fetch MAC Web Service 2	784			
Authentication Related Web Service1	745			
Authentication Related Web Service 2	668			

On the whole, validation on the performance of the services based on the user’s access perspective is validated and the skyline processing to identify the computing process on the services offered were performed. Finally the service selection is done on the dominated services (shown below). Below table indicates the better quality services (Chart below).

User ID	Service Processed Queue Process Delay Perspective	Service Processed Set Up Delay Perspective	Service Processed Method Response Time Perspective	Pearson Correlation Co-efficient - Through Skyline Computing Value	Service Offered	Service Offered
57	30	80	105	210	Service 2	
151	30	80	100	210	Service 1	
183	25	114	105	264	Service 2	
187	23	234	105	352	Service 2	

CONCLUSION

Web service QoS analyzing is the predominated aim of the work. The paper illustrates about the work model scenario with multi handle technique for the QoS-sensitive business application. The appropriate service selection by the web service through request processing functionality and the response has been monitored and formulated with to utilize in the matrix. The combinatorial matrix analyses the QOS service

and rate the QOS that satisfies the customer need. Thus the algorithm formulated and concept incorporated proves to be optimal and solves the issues faced by the existing scenarios. Averaging the overall service methods to finalized the best service can be considered for future work of this paper.

REFERENCES

- [1] Wuhui Chen, Incheon Paik, Member, IEEE, and Patrick C.K. Hung, Member, IEEE, Constructing a Global Social Service Network for Better Quality of Web Service Discovery, IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 8, NO. 2, MARCH-APRIL 2015.
- [2] Waseem Ahmed, Yongwei Wu, Member, IEEE, and Weimin Zheng, Member, IEEE, Response Time Based Optimal Web Service Selection, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 26, NO. 2, FEBRUARY 2015.
- [3] Guisselle A. Garcya Llinas and Rakesh Nagi, Network and QoS-Based Selection of Complementary Services, IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 8, NO. 1, JANUARY/FEBRUARY 2015.
- [4] San-Yih Hwang, Chien-Ching Hsu, and Chien-Hsiang Lee, Member, IEEE, "Service Selection for Web Services with Probabilistic QoS", IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 8, NO. 3, MAY/JUNE 2015.
- [5] Marin Silic, Student Member, IEEE, Goran Delac, Student Member, IEEE, and SinisaSrblijic, Senior Member, IEEE, Prediction of Atomic Web Services Reliability for QoS-Aware Recommendation, IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 8, NO. 3, MAY/JUNE 2015.