10

PwCOV in cluster-based web server: an assessment of service-oriented computing for COVID-19 disease processing system

Abhijit Bora, Tulshi Bezboruah
DEPARTMENT OF ELECTRONICS & COMMUNICATION TECHNOLOGY, GAUHATI UNIVERSITY, GUWAHATI, ASSAM, INDIA

1. Introduction

The symptoms and side effects of Coronavirus disease 2019 (COVID-19) have made its critical images in societies worldwide [1]. For international health concern, the World Health Organization has declared it as pandemic and an emergency health management issue. The pharmaceutical and hospital entities worldwide have an important role for treatment of the COVID-19 infected patients. In the year 2020, such entities have experienced different cases based on patient's health conditions and their respective immunity system [2]. Each evaluation process of the treatment is generating different data sets worldwide that comprise of important attributes, primarily required for pandemic disease management and developing clinical process. The characteristic of such data set includes clinical trials along with safety and efficacy of chloroquine or related formulation. Cortegiani et al. [3] highlighted that a data set model of COVID-19 clinical cases primarily comprises of patient population, intervention group, comparison group, primary outcome, study design, country, symptoms, and contacts with suspected or confirmed cases. However, based on isolated data sets in contrast to the clinical process of worldwide entities and results, accessing such data set through a collaborative measure has become an important concern from the perspective of clinical experts, patients, researchers, software practitioners as well as general society members. In such

Data Science for COVID-19. https://doi.org/10.1016/B978-0-12-824536-1.00038-1

cases, the deployment of service-oriented computing (SOC) can enhance the collective methodology of data processing for fighting against COVID-19.

The SOC system executes the entities of operational business logic (BL) in distributed system. The BL entities are referred as services over web. The entities provide its functionality through communication over network. It supports the enhancement of performance, scalability, and availability through the interaction among these services [4]. The SOC follows the principles of service-oriented architecture (SOA). It supports service reusages. It allows a service to be communicated by other service over network. The SOA can be implemented through variety of technologies and programming languages.The service entity can play the role of parent node, consumer node, and broker node. The interaction among such node can be carried out in dynamic mode [5]. A consumer of such service can discover a node at run time or can replace it if not functioning or can add new functionality to create new service. The SOA utilizes the power of extensible markup language (XML) to establish communication among service entities. Recently, the software as a service (SaaS) for deployment of SOA has become an attractive paradigm. It enhances many to many machine interactions over different topologies. As such, the SOA can offer the power of adaptability over different network protocols. In recent decades, the medical industries are observing the demand of the deployment of SOA-based system for health care and medical data processing [6]. The integration of point to point system has become expensive to maintain efficient service delivery. The utmost benefit of SOA-based system is that it provides a common framework for establishing the communication and integrates different features to an existing service comparatively in lesser time. So, the rapid adaptability can be achieved. The system that follows the principles of SOA for deployment of service over network is termed as SOC. Primarily, it utilizes the features of simple object access protocol (SOAP) or representational state transfer (REST), web service description language (WSDL), universal description and discovery integration for establishing service execution, service integration, and service discovery [7−10].

Many prominent research works are carried out to deploy SOC-based system for processing distributed business [11−13]. However, with rapid growth of medical consumers of such service entity, the efficiency of delivering medical service over the web has become crucial among the service providers. As a service provider in the medical domain of COVID-19, the primary role is to deliver proper medical reports even in high stress of service usage. In some cases, the service entity may fail in delivering services. However, for handling such rapid growth of consumers over web, the cluster-based load balancing (CBLB) web server can play a vital role. With the development of organizational infrastructure and service modules, the assessment of SOC based system for COVID-19 data set processing by using CBLB web server against massive growth of consumers has become a crucial challenge. Hence, for the collective support for pandemic disease management, the quality aspects of medical service for massive load of users have become a demanding subject among medical service providers and health care units of COVID-19 disease management system. This research work introduces a novel quality assessment framework for medical service delivery through the paradigm of SOC-based system by using CBLB web servers.

2. Materials and method

The SOC-based system provides necessary framework for deploying SaaS entity and support BL execution over network. It supports the tightly or loosely coupled SaaS entity for BL execution. Modular-oriented SaaS can be deployed through programmable interface [14]. The internal data structure of SaaS is omitted from the consumer entity. The consumer utilizes the WSDL of SaaS entity to establish and execute the BL. The SOC-based system supports the execution of SaaS entity by using the homogeneous or heterogeneous platform. Different ways of communication that can be performed in SOC-based system are discussed elsewhere [15]. Many studies revealed the limitation state of a single web server for massive request execution [8–10,16]. However, deployment of CBLB web server can enhance the service capacity in such a scenario [17,18]. The CBLB web server can be deployed by using a single node or multiple machines. Among different machine, one can create multiple working nodes. Such working node can execute similar BL functionality for end users. As and when a particular working node fails or busy, the load balancer will serve end users with another existing working node. Thus, the CBLB web server enhances the performance of SOC-based system. Such system supports high reliability, availability, and scalability for rapid growth of users. However, among the medical domain and service providers, the identification of processing limit beyond its service execution, correlation of system metrics over SaaS execution and reliability estimation through a novel methodology has obtained a high value.

2.1 Overview of the system metrics

The importance of observing response time, throughput, central processing unit (CPU) utilization, and reliability as system metrics is discussed in different studies [10–13]. The response time of service execution is defined as the round trip delay from the submission of hyper texts transfer protocol (HTTP) request, processing and delivering it to the requesters [19]. The estimation of data bytes processed as per HTTP request in server side is termed as throughput [19]. The CPU utilization of a system is defined as the estimation of processing unit utilization for processing massive request [20]. However, the reliability of the service is defined as the probability that the service will deliver proper response without failure records [21].

2.2 Related work

In the year 2001, Bauer et al. [22] had presented a case study over deployment of medical web service (WS) for automatic analysis of computer tomography angiography data set. They had concluded that clinical routine process can be assisted with WS for better detection and visualization of malformation in human body. However, the study lacks of evident about the system execution in peak usage of server. In the year 2005, Aanzbock and Dustdar [23] had discussed a novel model for WS-oriented medical data processing for the patients by integrating healthcare enterprise. The study had also highlighted the standardization effort and protocol of WS stack in the domain of medical service [23]. In the year 2006, Castro et al. had discussed about the deployment of WS for management of medical images. The composition and identification of WS over web technology can lead to successful deployment of behavioral modeling of business process for medical data processing [24]. The authors had also discussed some limitation of such processing. However, the study had lack of discussion about medical service processing for high stress of web consumers. In the year 2008, Vaida et al. had discussed a novel solution for medical image processing through SOA. The study had highlighted about how the reusability, shared, and composed technology in medical domain can influence better service delivery [4]. In the year 2012, Wu et al. [25] had discussed a novel multiagent-based WS-oriented framework that can be optimized for the quality of medical data processing in e-health care system. The framework was deployed with experimental data sample records of breast cancer patients. The applicability and uniqueness of the framework was verified through a comparative study with the works of other researchers. However, they had not discussed about the system metrics and correlation for massive load of users. In the year 2014, Zieba had proposed a novel decision support system under the paradigm of SOA for identifying the diagnosis problems while using missing or imbalanced data set. They had deployed cost sensitive support vector machine for solving such issues. The author had evaluated the quality and applicability aspects of such system through experimental work [26]. However, the study lacks of evaluation for massive growth of consumers of such services. In the year 2014, Medhi and Bezboruah [11] had addressed a novel assessment methodology for deploying SOC-based system. The study had highlighted the HTTP transaction processing results against different usages of the system. The experimental outcomes had established the applicability of the proposed system. However, the reliability aspects of such deployment were omitted in their study. In the year 2014, Rahmani et al. [13] addressed issues related to reliability and performance aspects for deploying WS-oriented system containing different middle layer configuration technique. They had proposed a novel methodology to evaluate such system. The study established the hypothesis that middle layer configuration can have impact over the execution of such service for data processing. However, the study lacks of any highlight over execution of such service in CBLB web server. In the year 2015, Andrikos et al. [27] had presented a novel methodology for deployment of client server–based system under the paradigm of collaborative WS for medical data processing. The necessary features of multidisciplinary team management among medical practitioners can be delivered through the proposed platform. They had discussed the possibilities of service bottleneck and scalability issues in such platform. However, a lack of study over performance aspects of such system is observed. In the year 2016, Mata et al. [28] had highlighted the deployment methodology of XML technology in web-based system for huge medical data exchanging and processing in medical cases of different diagnosis and treatment. They had addressed different pros and cons of such deployment in medical diagnosis. However, the study lacks the experimental work about their hypothesis. In the year 2016, Tang et al. had discussed a novel quality of service prediction model for network aware WS. The authors had discussed the applicability of such model through an experimental work over dataset collected from real-world domain [29]. The comparative results had established the viability and efficiency of their results. In the year 2016, Medhi et al. [12] had proposed a novel testing methodology for evaluating the performance and security aspects of service-oriented data processing system. They had proposed an architecture for deploying security in service-oriented system. In the year 2017, Liu et al. [30] had addressed a novel decision-based system for medical big data that can facilitate the WS-oriented medical service. The proposed system can promote the decision making for health care issues. However, the author had not discussed about the execution of such service for high load of consumers. In the year 2017, Maheswaria and Karpagam [31] had discussed a WS selection methodology for fulfilling the customer need over web. They had highlighted that wrong selection of WS for composition with other service can lead to server run time complication. In the year 2017, Medhi et al. [32] had proposed a novel prototype testing procedure that can be deployed for evaluating service-oriented system for high load of users. They had predicted that the windows communication foundation technology with Microsoft Visual.Net framework can be deployed for efficiently evaluating the reliability aspects of data processing for different stress level of users. However, the study had lack of highlight over the performance aspects of such execution. In the year 2018, Mougiakakou et al. [33] had developed a monitoring platform for management and treatment of patients of type 1 diabetes mellitus by using the paradigm of information and communication technology (ICT). The functional issues for deployment of such prototype were discussed. However, the reliability and performance aspects of such ICT-based system for medical data processing were not highlighted in this study. In the year 2018, Yamada et al. [34] had carried out a theoretical analysis to study the scalability of aggregating WS for processing data over communication gateway. The laboratory experiment was performed to establish the hypothesis that service segregation among subsystems can deliver better response. However, the study lacks of the comparative scalability assessment to establish the worthiness of WS aggregation.

This research work is novel, as it highlights the key aspects of deployment of SOC-based system in CBLB web server for processing scattered medical data sets. The prototype PwCOV mimics the behavioral aspects of SOC-based system's metrics for different stresses of usage. It describes the key metrics and their correlation for PwCOV execution in CBLB web server.

3. Focus of the study

The study focuses the assessment of SOC-based system for medical data processing by using CBLB web server. The key metrics of system execution and the impact of their correlation over performance of such deployment is another merit of the study. The emphasize is given over such study because the medical practitioner demands the performance metrics of such system before deployment in their domain. Fig. 10.1 below shows the block diagram of the proposed PwCOV system for processing COVID-19 data sets. Here, the end user will access the WS of the COVID-19 data set available at location A. Subsequently, the services available at location B, C, and D along with their respective COVID-19 data set will be invoked and processed for generating clinical assessment reports. As such, the compiled reports based on different data sets will be processed and delivered to the respective end user. For evaluation of such system, the architecture of SOC-based system with multi WS is taken where the role of WS, such as, child, mediator, and parent are segregated among WSs for fetching and processing medical data from database server. Fig. 10.2 shows the experimental arrangement and the architecture of

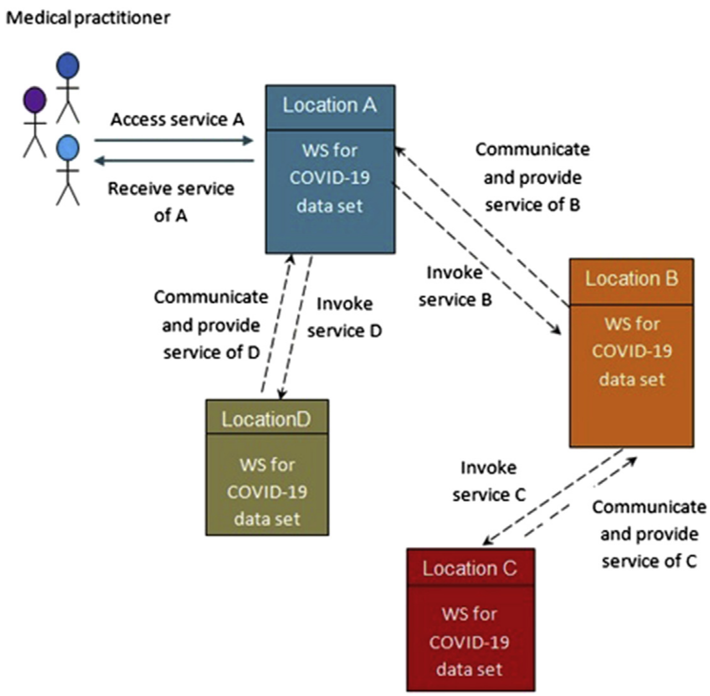

FIGURE 10.1 Block diagram of PwCOV model for processing COVID-19 dataset.

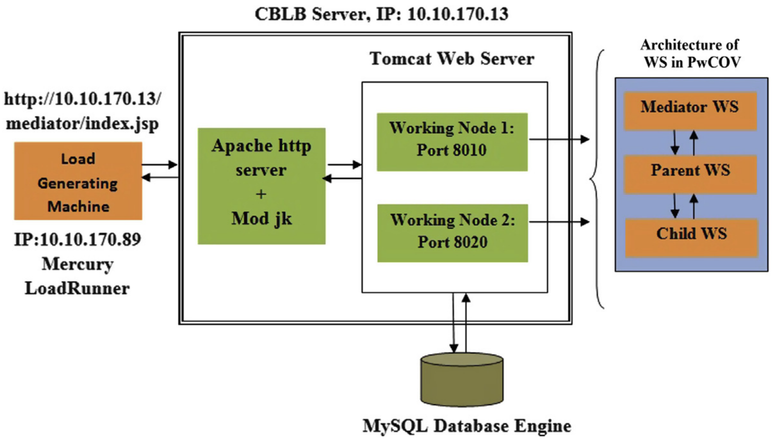

FIGURE 10.2 The experimental arrangement and architecture of SOC-based system with cluster-based load balancing web server.

PwCOV in CBLB web server. The investigation is carried out by configuring the Application Server Apache Tomcat, Web Server Apache HTTP, and Tomcat connector mod_jk. The SOC-based system with different roles of WS is developed by using JAVA and spring framework model view controller (MVC) version 2.5 and a MySQL data base server. The clinical data is prepared for different diseases and captured in the database [35]. A load testing tool Mercury Load Runner version 8.1 is deployed for generating different stress level of users [36]. Such users are system generated virtual users (VUs) that mimic the behavior of real end user.

A flowchart for assessment of PwCOV is developed and is shown in Fig. 10.3. A model test case is designed in Mercury LoadRunner, where all VUs are assigned to execute that test case. The test cases contain the instructions that all VUs follow for execution of the SOC-based system. The test case that followed in the study is discussed elsewhere [9]. The application server is clustered into two working nodes executing in the same architecture of PwCOV. The load over SOC-based system is managed by configuring the web server Apache HTTP along with tomcat connector mod_jk. The statistical analysis is carried out to study the performance metrics, such as: (a) response time, (b) throughput, and (c) CPU utilization for different stress level of VUs. The hardware and software configuration of the experimental arrangement is given in Table 10.1.

4. Testing of PwCOV

Rigorous system testing plays an important candidature for assessing the quality of a SOC-based system [11,16,20]. To assess the quality of SOC-based system in CBLB web

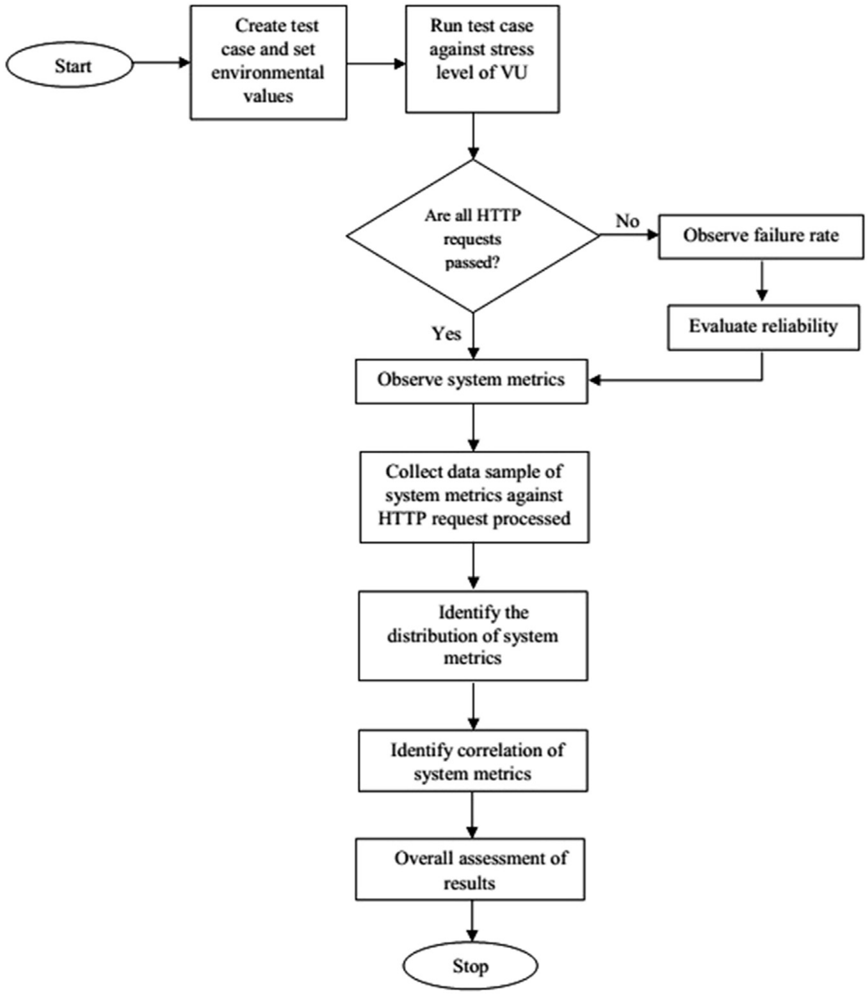

FIGURE 10.3 Flowchart for assessment of the PwCOV.

server, we have evaluated the PwCOV for stress level of 50, 100, 800, 1500, 1700, 1800, and 2000 VUs. A test bed is developed by using load testing tool Mercury Load Runner.

The test bed contains test case for accessing the PwCOV. Execution of test case causes the necessary BL operation to be carried out in CBLB web server. As the results in inter WS communication in SOC-based system is generated, all VUs are set to enter the execution period of the server gradually. During this period, the system metrics are recorded for response time, throughput, CPU utilization, and HTTP failure record by Mercury Load Runner. The assessment of response time, throughput, CPU utilization, and failure rate for PwCOV against different stress level of VUs is plotted and shown in

Table 10.1 Software and hardware specification for the PwCOV assessment.

Software/hardware specification	Load generating system	Service-oriented computing—based system with CBLB web server
Processor	Intel Pentium dual CPUE2200 @ 2.20 GHz	Intel Xeon central processing unit (CPU) E5620 @ 2.40 GHz
Operating system (OS)	Windows XP OS	64-bit windows server 2008 R2 standard OS
RAM	1 GB	8 GB
Web server	Apache HTTP server version 2.2.4 web server	Not applicable (NA)
Application server	NA	Apache tomcat version 7
Database software	NA	MySQL version 5.0
Hard disk	150 GB	600 GB
Application software tools	Mercury LoadRunner load testing tool version 8.1	• Java runtime environment of version 7, • Java development kit of version 7, • Mod JK tomcat connector, • Metro web service stack, • Spring model view controller 2.5 framework, • NetBeans integrated development environment of version 7.0, • Google Chrome web browser

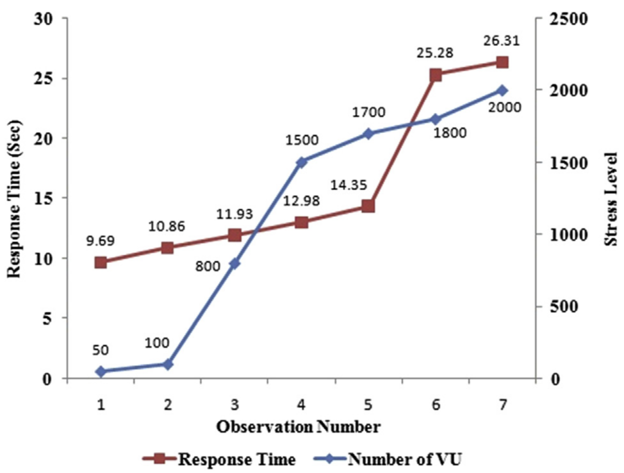

FIGURE 10.4 Recorded response time against different stress level of users.

Figs. 10.4–10.7, respectively. From the **Figures** it is observed that as the stress level is increased, the stress level of recorded system metrics is also increasing gradually.

From Fig. 10.7, it is observed that up to the stress level of 1700 virtual user, the failure rate is 0. Beyond that, it increases gradually.

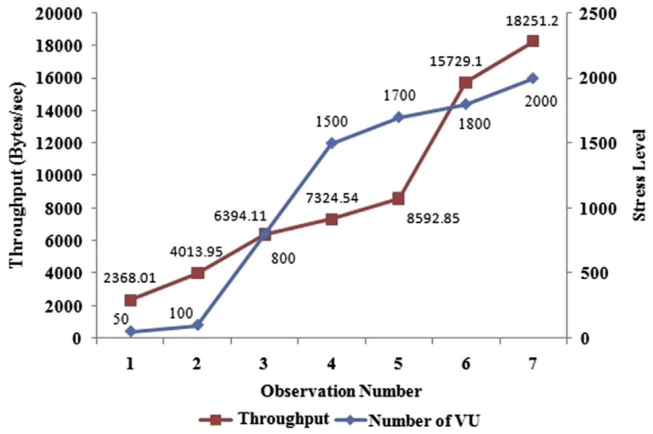

FIGURE 10.5 Recorded throughput against different stress level of users.

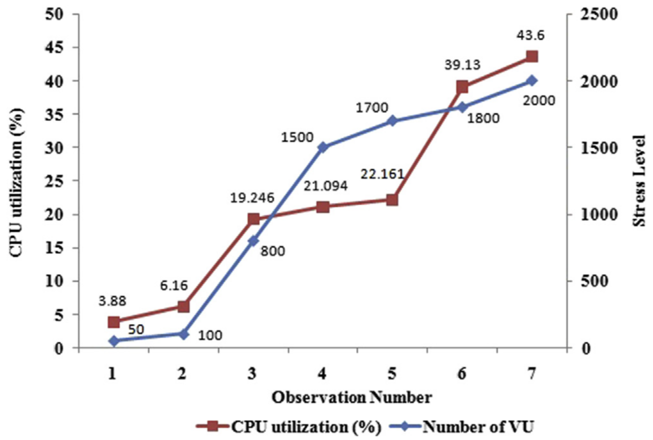

FIGURE 10.6 Recorded central processing unit utilization against different stress level of users.

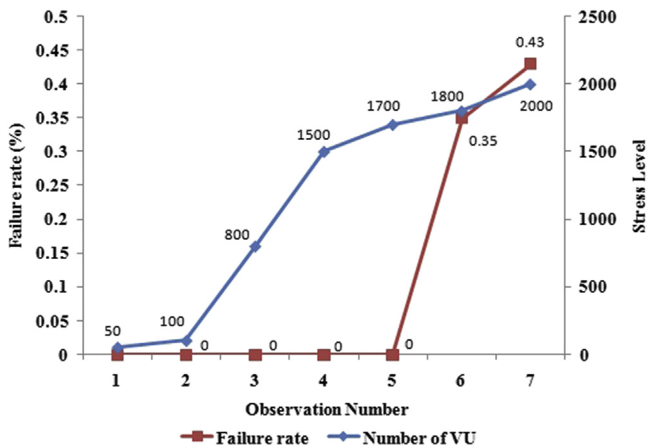

FIGURE 10.7 Observed failure rate against different stress level of users.

4.1 Distribution of data points for 50 virtual user

Normality for recorded parameters is evaluated graphically through Normal Probability Plot (NPP) [37]. To study the distribution, a data sample of repetitive test for 50 virtual users is taken.

About 30 counts are taken for the stress level of 50 VU. The NPP of response time, throughput, and CPU utilization is shown in Figs. 10.8–10.10,respectively. It is

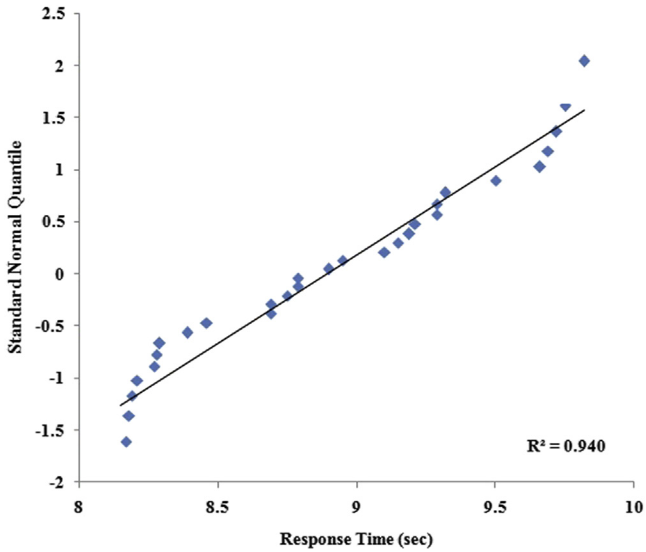

FIGURE 10.8 Observation of Normal Probability Plot for response time.

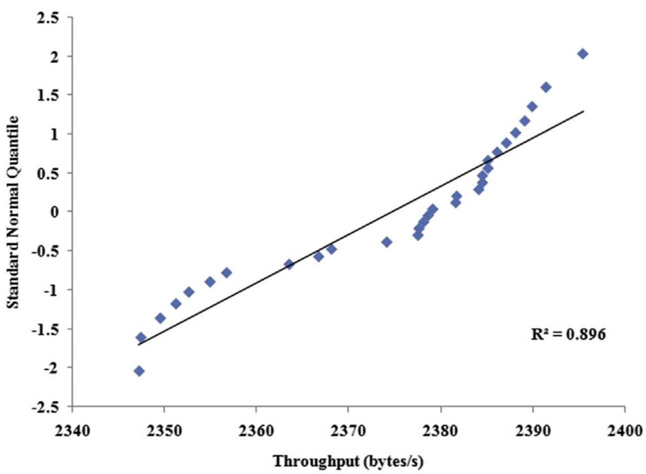

FIGURE 10.9 Observation of Normal Probability Plot for throughput.

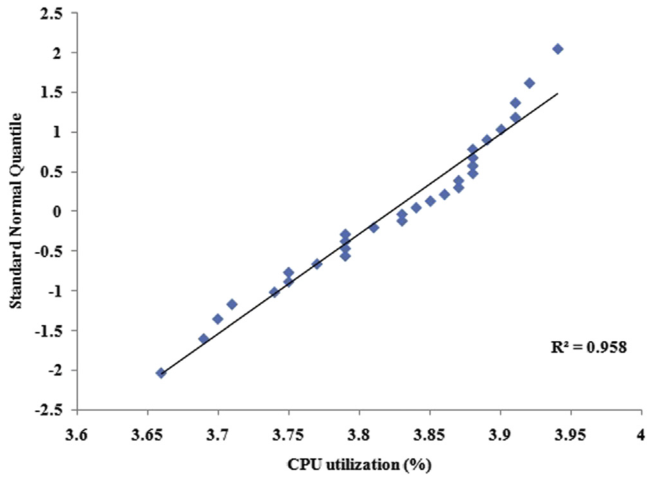

FIGURE 10.10 Observation of Normal Probability Plot for central processing unit utilization.

observed that the recorded system metrics are following straight line. As such, it can be concluded that the data points are linear, normal, and follows the mean of system metrics.

4.2 Correlation of system metrics

The correlations of system metrics are observed through scatter plot. The scatter plot for response time and throughput is shown in Fig. 10.11. The scatter plot for response time and CPU utilization is shown in Fig. 10.12. In both the cases, the R^2 values are near to 1. As such, it can be concluded that the throughput and CPU utilization have individual impact over response time of the system. To study that whether the throughput and CPU utilization have combined impact over response time, the multiple linear regression

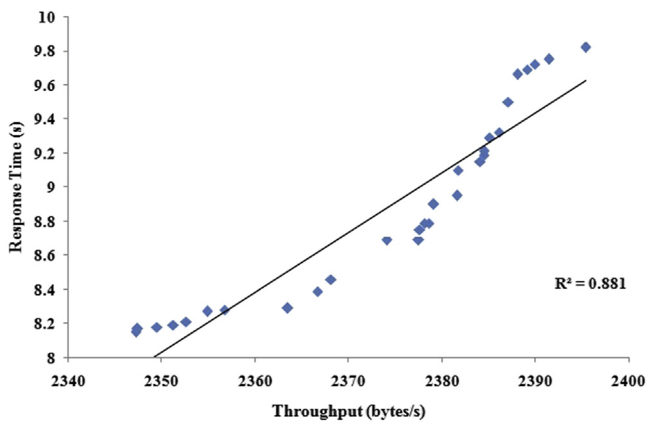

FIGURE 10.11 Correlation of response time and throughput.

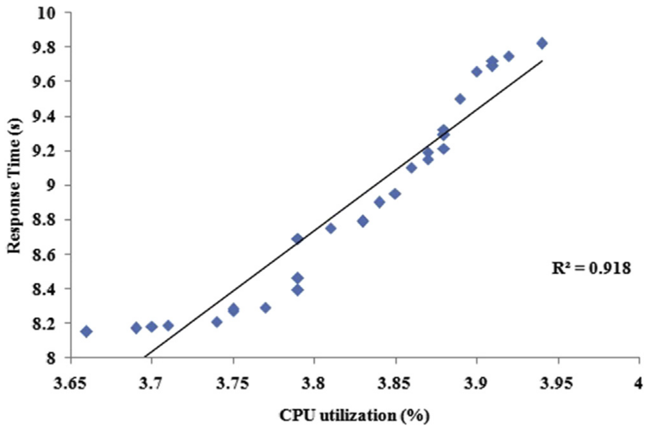

FIGURE 10.12 Correlation of response time and central processing unit utilization.

Table 10.2 Multiple linear regression analysis of recorded system metrics.

F ratio	F table critical value F(2, 27)	Is F ratio > F table critical value?	Adjusted R^2
152.223	3.354	Yes, accept H_o	91.25%

analysis (MLRA) at 95% confidence level is carried out with a hypothesis (H_o) that the correlation among response time, throughput, and CPU utilization exist. The responses of MLRA are shown in Table 10.2. From Table 10.2, it can be concluded that the hypothesis is valid.

5. Reliability of PwCOV

The assessment of the PwCOV for reliability estimation is done through evaluation of fault count. From Table 10.1, it is observed that at 1800 VU stress level, the system throws HTTP error rate. As such, a 30 repeated test at 1800 VU stress level is carried out to study the reliability aspects of PwCOV. The reliability assessment of PwCOV is evaluated by using Eq. (10.1) as defined in Refs. [13,32,38].

$$R = e^{-ft} \tag{10.1}$$

Here, R represents the reliability of PwCOV. Eq. (10.1) utilizes failure rate "f" as observed during exposure period of time "t." The R estimates the reliability of PwCOV execution during high stress of usage by using CBLB web server. The R evaluates to a value that lies within the range of 0–1. A value nearer to one represents strong reliability of the service delivery. In this study, the "f" for the stress level of 1800 VU is evaluated to be 0.35%. As such, for "t" equals to one day, R is evaluated to be 0.7. Hence, we can conclude that strong reliability can be achieved up to 70% of its execution period of the system against the stress level of 1800 VU usage at one time.

6. Overall assessment of PwCOV

The performance of the PwCOV is increasing with the increase in stress level of VUs. Better scalability can be achieved up to a stress level of 1700 VU. The SOC-based system is not showing any failure rate up to this stress level. However, beyond this, failure rate is generating and increasing gradually. The reason for occurrence of failure may be because of utilization limit of feasible system resource that can be assigned for executing massive request in server side. It is observed that the correlation among response time, throughput, and CPU utilization exists. With increase in stress level, the CPU utilization and throughput is increasing. The response time is also increasing with increase in stress level. The throughput and CPU utilization have individual as well as combined impact over the response time. A sudden hype of response time, throughput, and CPU utilization from stress level of 1700 to 1800 VU is observed. The sudden hype of system metrics may be because of JAVA runtime garbage collection error, data base engine error, or lately releasing system resource that are primarily required for executing massive load. It is observed that the applicability and viability of medical data processing through the paradigm of SOC-based system can be achieved up to 1700 VU. For the usage stress of 1800 VU, the reliability can be achieved up to 70% of its execution period. A comparative assessment of the proposed system with results of other researcher's work is given in Table 10.3.

From Table 10.3, it is observed that, in the proposed system, the strong stability can be achieved up to a stress level of 1700 VU. However, in case of [14], poor stability was achieved with a failure rate of 60.28% at stress level of 100 VU. Similarly, for [16], poor stability was achieved with a failure rate of 50% at stress level of 100 VU. Similarly, for

Table 10.3 A comparative assessment of the proposed service-oriented computing–based system with other authors' deployment methodology.

No. of virtual user	FR% of proposed study	FR% of earlier study done by other researchers		
		[14]	[16]	[11]
50	0	Not tested	0	0
100	0	60.28%	50%	0
500	Not tested	87.9%	Not tested	0
800	0	Not tested	Not tested	Not tested
1000	Not tested	92.06%	Not tested	37%
1500	0	Not tested	Not tested	75%
1700	0	Not tested	Not tested	Not tested
1800	35%	Not tested	Not tested	Not tested

Ref. [11], poor stability was achieved with a failure rate of 37% at a stress level of 1000 VU. As such, from the overall comparative assessment, it can be concluded that the proposed system, by using CBLB web server is comparatively better in stability and efficiency than the systems developed by other researchers. So, even in high usage of the system, the PwCOV is highly stable for processing medical data store. It can be also concluded that during high usage of PwCOV, the expectation of SOC service delivery through CBLB web server can be achieved efficiently. However, beyond the processing limit, the PwCOV may throw failure records. The limitation beyond the capacity may by due to the fact that with increased stress level, utilization of server resources increases gradually. These resources primarily used for SOAP request processing, generating HTTP POST methods and necessary BL with data base engine. Hence, beyond the capacity limit of such resource, the CBLB web server may not work efficiently for delivering response properly. So, after a specific stress level of usage, the PwCOV can have failure rate. From the results of the proposed work, it can be concluded that the SOC-based system is feasible for medical data processing for a specific load of users. The tolerance of processing SOAP request is applicable for the stress level of upto 1700 VU. Beyond that the failure response may generate with poor system performance. It is also observed that better scalability can be achieved by segregating the role among WS instead of amalgamating the service into one WS.

7. Conclusion

This work introduces a novel methodology for evaluating the SOC-based system that can process disease-related data by using CBLB web server. From the study of the prototype service in CBLB web server, it is observed that the PwCOV is feasible, effective, and applicable. The service along with Tomcat web server and JAX-WS can process SOAP request up to a stress level of 1700 VU. Beyond that, the SOAP processing error can generate. And so, it throws HTTP failure records. The overall processing and efficiency limit of the proposed SOC-based system is observed to be better than other methodology followed by other group of researchers. The correlation of system metric exists. The correlation can also influence the overall performance of the proposed PwCOV for medical data processing. The response time of the system is influenced by the throughput and CPU utilization. The normality of data distribution for the recorded system metrics is observed. From the overall assessment, it can be concluded that segregating the role among services is better than amalgamating them into a single service. The proposed deployment methodology can provide better performance and scalability for pandemic disease data processing against massive growth of consumers over web. The results of the proposed work can help researcher and medical practitioner to gain the viability and insight about CBLB-based web server for medical data processing. However, an in depth evaluation of the proposed SOC-based system by considering different system metrics, test cases, and environmental test configuration is necessary to highlight different aspects of COVID-19 data processing system through the paradigm of SOC with CBLB web server.

Acknowledgment

The authors are thankful to the All India Council of Technical Education (AICTE), Govt. of India, New Delhi for financial support toward the work (Grant No. 8023/BOR/RID/RPS (NER)-84/20102011, March 31, 2011).

References

[1] H. Leite, I.R. Hodgkinson, T. Gruber, New development: 'healing at a distance'- telemedicine and COVID-19, Publ. Money Manag. (2020), https://doi.org/10.1080/09540962.20201748855.

[2] L.H. Wang, Y. Wang, D. Ye, Q. Liu, A review of the 2019 novel coronavirus (COVID-19) based on current evidence, Int. J. Antimicrob. Agents 55 (6) (2020), https://doi.org/10.1016/j.ijantimicag.2020.105948.

[3] A. Cortegiani, G. Ingoglia, M. Ippolito, et al., A systematic review on the efficacy and safety of chloroquine for the treatment of COVID-19, J. Crit. Care 57 (2020) 279−283, https://doi.org/10.1016/jjcrc.2020.03.005.

[4] M.F. Vaida, V. Todic, M. Cremene, Service oriented architecture for medical image processing, Int. J. Comput. Assist. Radiol. & Surg. 3 (3−4) (2008) 363−369, https://doi.org/10.1007/s11548-008-0231-8.

[5] M. Herrmann, M.A. Aslam, O. Dalferth, Applying semantics (WSDL, WSDL-S, OWL) in service oriented architectures (SOA), in: Proceedings of 10th International Protégé Conference, 2007.

[6] Microsoft Healthcare, April 10, 2019. Available from: https://www.microsoft.com/en-us/enterprise/health.

[7] SOAP, April 05, 2019. Available from: http://www.w3.org/TR/soap12-part1/.

[8] A. Bora, M.K. Bhuyan, T. Bezboruah, Investigations on hierarchical web service based on java technique, Proc. World Congr. Eng. (2013) 891−896.

[9] A. Bora, T. Bezboruah, Testing and evaluation of a hierarchical SOAP based medical web service, Int. J. Database Theory & Appl. 7 (4) (2014) 169−188.

[10] A El Saddik, Performance measurement of web service based application, IEEE Trans. Instrum. & Meas. 55 (5) (2006) 1599−1605.

[11] S. Medhi, T. Bezboruah, Investigations on implementation of e-ATM web services based on. NET technique, Int. J. Inf. Retr. Res. (IJIRR) 4 (2) (2014) 42−51.

[12] S. Medhi, A. Bora, T. Bezboruah, Security impact on e-ATM windows communication foundation services using certificate based authentication and protection: an implementation of message level security based on. NET technique, Int. J. Inf. Retr. Res. (IJIRR) 6 (3) (2016) 37−51.

[13] M. Rahmani, A. Azadmanesh, H. Siy, Architectural reliability analysis of framework-intensive applications: a web service case study, J. Syst. Software 94 (2014) 186−201.

[14] M.C. Matthew, K. Laskey, F. McCabe, P. Brown, R. Metz, Reference model for service oriented architectures, OASIS Work. Draft 11 (2005).

[15] A. Bora, T. Bezboruah, Some aspects of QoS for high performance of service oriented computing in load balancing cluster based web server, in: Handbook of Research on Recent Developments in Intelligent Communication Application 21, IGI- Global, 2016, pp. 566−603. https://doi.org/10.4018/978-1-5225-1785-6.

[16] M. Kalita, T. Bezboruah, Investigations on implementation of web applications with different techniques, IET Softw. 6 (6) (2011) 474–478.

[17] H. Bryhni, E. Klovning, O. Kure, A comparison of load balancing techniques for scalable web servers, IEEE Network 14 (4) (2000) 58–64. https://doi.org/10.1109/65.855480.

[18] A. Bora, S. Medhi, T. Bezboruah, Investigations on failure and reliability aspects of service oriented computing based on different deployment technique, Lecture Notes in Electrical Engineering, in: Advanced Computational and Communication Paradigms 475, 2018, pp. 535–543. https://doi.org/10.1007/978-981-10-8240-5_60.

[19] D.A. Menasce, V.A.F. Almeida, Capacity Planning for Web Services - Metrics, Models and Methods, Prentice Hall, 2002.

[20] H.K. Singh, T. Bezboruah, Performance metrics of a customized web application developed for monitoring sensor data, in: Proceedings of IEEE 2nd International Conference on Recent Trends in Information Systems, 2015, pp. 157–162.

[21] A.L. Goel, Software reliability models: assumptions, limitations, and applicability, IEEE Trans. Softw. Comput. SE-11 (12) (1985) 1411–1423. https://doi.org/10.1109/TSE.1985.232177.

[22] S. Iserhardt-Bauer, P. Hastreiter, T. Ertl, K. Eberhardt, B. Tomandl, Case study: medical web service for the automatic 3D documentation for neuroradiological diagnosis, Proc. IEEE Vis. (2001) 425–581. https://doi.org/10.1109/VISUAL.2001.964542.

[23] R. Anzbock, S. Dustdar, Modeling and implementing medical web services, Data Knowl. Eng. 55 (2005) 203–236. https://doi.org/10.1016/j.datak.2005.03.009.

[24] V.D. Castro, M. LópezSanz, E. Marcos, Business process development based on web services: a web information system for medical image management and processing, in: Proceedings of IEEE International Conference on Web Services, 2006, pp. 807–814.

[25] C.S. Wu, I. Khoury, H. Shah, Optimizing medical data quality based on multiagent web service framework, IEEE Trans. Inf. Technol. Biomed. 16 (4) (2012) 745–757.

[26] M. Zieba, Service-oriented medical system for supporting decisions with missing and imbalanced data, IEEE J. Biomed. & Health Inform. 18 (5) (2014) 1533–1540.

[27] C. Andrikos, G. Rassias, P. Tsanakas, I. Maglogiannis, Real-time medical collaboration services over the web, in: Proceedings of 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2015, pp. 1393–1396.

[28] C. Mata, A. Oliver, A. Lalande, P. Walker, J. Marti, On the use of XML in medical imaging web-based applications, IRBM 38 (1) (2017) 3–12.

[29] M. Tang, Z. Zheng, G. Kang, J. Liu, Y. Yang, T. Zhang, Collaborative web service quality prediction via exploiting matrix factorization and network map, IEEE Trans. Network & Serv. Manag. 13 (1) (2016) 126–137.

[30] J. Liu, Y. Zhang, C. Xing, Medical big data web service management platform, in: Proceedings of IEEE 11th International Conference on Semantic Computing, 2017, pp. 316–321.

[31] S. Maheswaria, G.R. Karpagam, Performance evaluation of semantic based service selection methods, Comput. Electr. Eng. 71 (2017) 966–977.

[32] S. Medhi, A. Bora, T. Bezbaruah, Investigations on some aspects of reliability of content based routing SOAP based windows communication foundation services, Int. J. Inf. Retr. Res. (IJIRR) 7 (1) (2017) 17–31.

[33] S.G. Mougiakakou, C.S. Bartsocas, E. Bozas, N. Chaniotakis, D. Iliopoulou, I. Kouris, S. Pavlopoulos, A. Prountzou, M. Skevofilakas, A. Tsoukalis, K. Varotsis, A. Vazeou, K. Zarkogianni, K.S. Nikita, SMARTDIAB: a communication and information technology approach for the intelligent

monitoring, management and follow-up of type 1 diabetes patients, IEEE Trans. Inf. Technol. Biomed. 14 (3) (2010) 622−633.

[34] T. Yamada, K. Suzuki, C. Ninagawa, Scalability analysis of aggregation web services for smart grid fast automated demand response, in: Proceedings of IEEE International Conference on Industrial Technology, vol. 205, 2018, pp. 1285−1289. https://doi.org/10.1109/ICIT.2018.8352363 (1).

[35] Drug Index, Passi Publications, India, 2012.

[36] Mercury LoadRunner, July 29, 2018. Available from: https://qageek.files.wordpress.com/2007/05/loadrunner_tutorial.pdf.

[37] NIST/SEMATECH E-Handbook of Statistical Methods, January 29, 2016. Avaialable from: www.itl.nist.gov/div898/handbook/apr/section1/apr162.htm.

[38] M.L. Shooman, Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design, John Wiley & Sons, 2002. https://doi.org/10.1002/047122460X.