International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August 2019

Scalability Enhancement for Cloud-based Applications using Software Oriented Methods

Muhammad Ehsan Rana, Usman Farooq, Wan Nurhayati Wan AB Rahman

Abstract: ***Scalability refers to the ability of a system to handle resource utilization in a constant and smooth fashion when high or low volume of data is applied. It is among the key attractions for migration to a cloud based infrastructure. Most of the previous studies in this area are based on the enhancement of cloud scalability in terms of hardware resources and network infrastructure. However in this case the cost of additional hardware resources and expansion of network infrastructural components to improve the cloud scalability is a major hurdle. Improving scalability of software on cloud platform by improving the software design is very less explored area. This paper focuses on two major concepts that involve measuring of software scalability using different methods and secondly exploring the software design based approaches to improve scalability. At the end, researchers have also explored the use of software design patterns to enhance scalability and flexibility in software applications on available cloud platforms especially Platform-as-a-Service (PaaS).***

Index Terms: Software Scalability, Scalability Measurement Methods, Scalability Enhancement Methods, Cloud Infrastructure, Platform-as-a-Service (PaaS), Software Design Patterns

I. INTRODUCTION

Cloud computing is very effective approach to develop efficient software applications with less cost. Cloud services are provided by service providers and being managed by them. Most of the time service providers are failed to offer the best services such as availability, performance and scalability which are highly in demand by clients these days. Cloud computing mostly refers to on-demand service over the internet. There are two main things here, first is the application which is distribute by the internet as a service and second is hardware which refers to the data center. The combination of these both allows a software developer to deploy his/her application on to this big platform [1].Software scalability is mostly referred to the capability of a software application to offer a rational performance even when data volume or resources are high.

Revised Manuscript Received on August 30, 2019.

* Correspondence Author

Muhammad Ehsan Rana*, Asia Pacific University of Technology & Innovation, Malaysia & Universiti Putra Malaysia.

Usman Farooq, Asia Pacific University of Technology & Innovation, Malaysia.

Wan Nurhayati Wan AB Rahman, Universiti Putra Malaysia.

Usually cloud software scalability becomes difficult to achieve because of the unknown type of services used by customer especially when unpredictable volumes of service requests approach the cloud platform. Due to the lack of scalability, cloud service providers or companies lose their customers and revenues. Most of the existing work done on scalability improvement is based on hardware upgrading. As this phenomenon of adding more resources is a static approach, hardware oriented scalability structure requires the shutting down of systems to enhance scalability which stops all the services.

II. SCALABILITY VS. PERFORMANCE

Software application's performance is very onerous area because complexity of software solutions is increasing every day and to take care of the best performance is a challenging venture. Most of the research in this paper revolves around scalability which differs from performance. Performance has following three main factors.

- Response time: how much time a system takes for a request to complete its process.
- Throughput: the count of number of processed requests per second.
- System availability: probability of a functioning system, when it's needed. In other words, it is percentage of time when system is operational. Goal of high system availability is to minimize downtime or recovery from outage.

On the other hand, scalability is to make sure that the performance is constant no matter how many end users are using the system concurrently. This means greater performance will automatically handle high scalability. But this is not always the case sometimes hardware resources are not enough to handle big number of requests even for the good performance software applications so improvement in hardware can also increase the scalability [2]. Study shows that using software improvement techniques can also be used to improve the scalability of software especially software design patterns [3]. Ability of code to communicate or handle requests efficiently on larger system in which nodes are constantly increasing is called scalability of the code. Running a code in efficient and optimized way on increasingly larger systems is called scaling.

2. Scalability on Cloud Platform

Scalability is among the main features of cloud environment to motivate for cloud migration. High scalability in cloud is typically achieved by adding more hardware resources or by improving network infrastructure. Because it is a dominant feature of cloud,

which means when more resources are added its capacity to handle the increasing amount of data volume or its capacity to tackle the throughput performance becomes better. Good scalability and elasticity are the main reasons that leads people to use cloud infrastructure [4]. High elasticity can be achieved by scaling up the code of software application by different manners.

There are two main types of scalability namely vertical scalability and horizontal scalability. To achieve vertical scalability power of nodes should be increased. On the other hand, to achieve a horizontal scalability new node should be added into the system [5].Achieving high scalability is very important on cloud as scalable cloud environment is much more effective for running enterprise software applications. Utilization of resources in a content and steady manner is mostly needed for system availability without effecting all other users in multiple usage environments or platforms [6]. Software developers need to put a lot of effort to create and deploy software solutions on cloud using PaaS environment that offer dynamic and effectual support for software developers. Scalability improvement on cloud can also be beneficial for the productivity of the software developers and it can save time and money [7]. High scalability is also very important because it can process big data in a very effective way [8].

3. Review of Scalable Architectures for Platform-as-a-Service (PaaS) Cloud

PaaS is a commonly used service model in cloud computing from last ten years especially in terms of market share because it helps different organizations to deploy their services or software solutions which are built in different programming languages. It also provides support for different databases, runtime environments and several frameworks to run their services in a systematic way. Major PaaS platform providers that include Amazon AWS, Microsoft Azure, Google App Engine and IBM Bluemix have facilitated software development by providing software developers fault-tolerant cloud services such as, key value, objects, databases and background multithreading. Some of the ongoing PaaS platforms have big automations in which it will automatically scale up the software applications [9].

Cloud computing platforms are based on virtual environments. The main components of a virtual environment include software and hardware platform in the form of servers (PaaS resources), resource management node and database servers [10]. It is found that replication of databases is an issue while scaling of PaaS platform. Figure 1 shows that at container level, similar component can run on different instances to achieve better fault tolerance and scalability. Some mechanisms should always be available on platform for consistency of data replication between different components. For better scalability on database level, copies of data can be hosted on instances of database management system (DBMS) [11].Platform as a service usually pacts with the accumulation of software solutions which are found on internet. It also deals with the management of software applications in terms of better performance throughput and scalability. Scalability and elasticity can be achieved on the cloud platform by different methods.

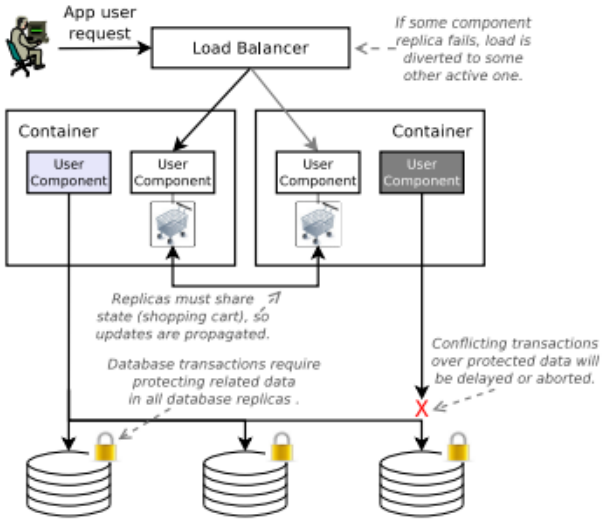

Figure 1: Database Replication in PaaS [11]

III. SCALABILITY MEASUREMENT METHODS

Measuring software scalability is an important factor because it helps software developers to know which area of their project can be optimized. Software scalability can be measured using different frameworks and scientific laws, it can also be measured with numerous automated testing tools. Benefits of these tools are formulation of automated result sets in graphs and numbers. To quantify software scalability, there are several methods available and some of them are following:

3.1 Little's Law

To apply this law, let's consider a small example of a banking system. There can be number of request types to this banking system like a request of a client to view recent transactions. Let's suppose a single client request takes 10ms to complete the request on server side. End to end response can take more time but to keep the example simple, let's just consider only server-side time. In next turn, suppose 50 users want to view their transaction then it will take of average 500ms server processing time [2].

Some of the terms used in this scalability measure process are defined as follows [2].

- Throughput: This is the rate on which a truncation is being processed by the system.
- Resource Usage: The amount of usage by different resources such as disk, memory CPU etc.
- Cost: This is the price per transaction.

Now measuring resource usage is bit easy but throughput and cost needs further explanation. Here little's law comes into the place to calculate throughput in both cases:

- X=N/R

Where N is number of users and if average user spends R seconds then X is throughput. This law can be applied on a system, a disk or a web application [2].

Table 1: Throughput per Second

Concurrent User	Time(ms)	Throughput(tps)
1	10	100
50	500	100
100	1200	83.333
150	2200	68.182
200	4000	50

Based on these numbers in Table 1, it is hard to determine that how good the application scales because there should be a comparison so that we can judge how good or bad the application scales. The comparison example would be a liner scalable version of our application by which application would do the same work no matter how many users exists.

Table 2: Linearly Scalable Throughput per Second

Concurrent User	Time(ms)	Throughput(tps)
1	10	100
50	500	100
100	1000	83.333
150	1500	68.182
200	2000	50

Table 1 shows that our application is not linear scalable when users are more than 50. Here keep in mind that scalability analysis is not same as performance analysis: (i.e. a slow application can be able to scale). But here we are interested in how well application manages increase amount of load. On the other hand, Table 2 shows that a theoretical application should keep or maintain the transaction per second. It doesn't matter if response time increases but amount of work done should be the same [2]. In conclusion, table 1 showed latency from client's point of view and table 2 showed work done per unit from server's point of view.

3.2 Measuring Cloud-based Application Scalability by Stress Testing Tools

Stress testing is form of testing by which we can determine the performance of the application. The result of stress testing shows that how much maximum load an application can handle. The owner of the application can set the limit of the load after the result shown by stress test [2], [12].

Following are the testing tools:

- Load Runner
- Perceiver
- HP ALM
- Perfmon Logs

These tools are very handy for performing test cases. Table 3 describes the results obtained by using these tools.

Table 3: Tools and Results

Tools	Performance
Load Runner 11.0	It is used to capture user's business processes and generates an auto performance test script. It manages, organize and monitors load test. Creates virtual users to generate load. Compares and generates graphs for performance results.
Perceiver	It is used to figure out bottlenecks in performance, used to monitor memory and CPU utilization
HP ALM	It is used to run and schedule tests according to requirement of client for multiple users
Perfmon Logs	To find out number of request hits on server

3.3 Scalability Analysis using Universal Scalability Law

Universal Scalability law is combination of the following:

- Initial linear scalability when load is increasing
- Cost of shared resources
- Diminishing returns due to contention
- Relative capacity

$$C(N) = \frac{N}{1 + \alpha (N - 1) + \beta N (N - 1)}$$

USL [13]

Where N is the scalability of system application in terms number of parallel users. α is the contention penalty and β represents coherency penalty. These all parameters are indistinct over $0 \leq \alpha, \beta < 1$ [13].

IV. SCALABILITY ENHANCEMENT METHODS

Maximizing number of hardware resources i.e. adding extra processing power, memory etc. is one of the most commonly used approach for scalability enhancement. However scalability can also be improved by improving the software architecture or by improving both software architecture and hardware resources. This research focuses on the use of software improvement approaches for scalability enhancement. The benefit of using these approach is to reduce the cost and infrastructural limitations involved in the hardware improvement [2].

There are number of methods available for scalability enhancement, as per the limitation of the subject, we tried to figure out the most appropriate approaches using software engineering techniques for scalability improvement. Some of these techniques which are related to the topic are discussed below:

6.1 Multithreaded Applications

Cloud environment is very high in demand for software solution according to very recent researches. Because of the high demand load balancing is a big issue for many organizations. According to research by increasing number of threads can enhance the scalability in terms of memory bandwidth. Many algorithms with multithreaded approach are being used to improve load balancing [14]. Multithreaded applications are very useful for gaining load balance, high performance and scalability [15].

6.2 Qualitative Comparison of Software Oriented Methods for Scalability Enhancement

CloudScale is a method which directly deals with the scalable cloud applications and services. Electronic health record (EHR) system is a scenario being used to demonstrate the cloudscale model. It describes how to improve the complexities involved in the cloud applications in terms of scalability. This method is being supported by three integrated tools and scalability description language. It contains complete system life cycle,

- First step of this model is identifying the requirement process and focus of this step is to describe work scaling path and load. For example, in EHR system requires a record to show up in less than a second.
- Second step of this model called construction and analysis which is used to specify a system.
- Next step contains the analysis of the modeled system to verify identified requirements.
- After above steps implementation will be performed and then deployment on the cloud environment.

Monitoring process is also enabled during system operation. Collection of all the measurements and performance is gathered here. Based on the quality, if there is a change required in system requirements than it will return to process cycle. Scalability Patterns shows how to build a good scalable cloud system while anti-patterns are opposite of the patterns. In EHR system, scalability patterns are used to improve the scalability [16]. AppScale is an opensource application software that implements PaaS, it allows cloud applications, easy to deploy and easy to scale over cloud. AppScale is also compatible with Google App Engine (GAE).Figure 2 shows that AppScale has implemented a multitier distributed stack with auto scaling, load balancing and deployment it also includes the API adaptors. Software Developers will deploy AppScale on cloud environment and then they can upload their applications to the platform using AppScale [9]. Both approaches are software oriented which supports our research topic. As focus of these frameworks are not limited to hardware, so we can say that there is high probability of improvement in scalability using software engineering approaches.

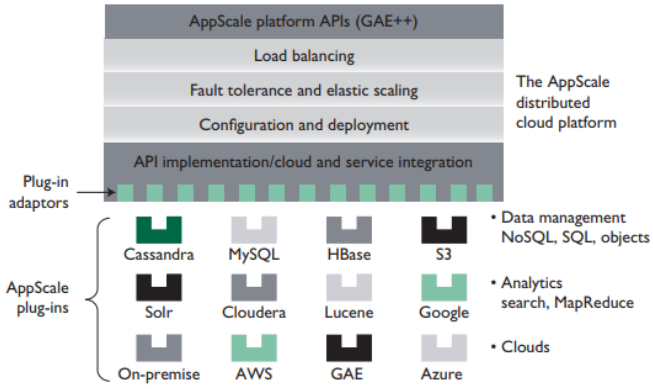

Figure 2: AppScale on Cloud Platform [9]

Eucalyptus is a cloud API framework which provides implementation of AWS APIs and its implementation is done on top of Eucalyptus. It provides tools on cloud ecosystem which offers possibility of communication with AWS. In comparison to OpenStack API, eucalyptus provides additional features such as, auto scaling of resources and elasticity in terms of load balancing. It also offers a tool to monitor utilization of resources and applications with CloudWatch [17]. Scaling of software applications is not a simple task. Purposed approach has achieved a component-based structure to scale software applications on cloud. This service allows transparent resource allocation at the same time components are distributed on multiple computing nodes. It automatically decides when to scale out and scale in the application's component. Java and OSGi model are used in this prototype. It contains fully automated process which doesn't require any manual step. Application will remain portable and it can be moved easily from one host to cloud [5].

4. Impact of Software Architecture and Design on Scalability

Software design and architecture has a significant impact on scalability. The way software is architected or designed will significantly influence the quality factors including scalability of a system.

7.1 Role of Object-Oriented Paradigm in Scalability Enhancement

Object-oriented approach is used to design computer software applications in an efficient and effective way. This technique revolves around modeling software components as real-world objects. Attributes holds the data of objects whereas functionality is managed in methods also known as behaviors. The main components in object-oriented design are encapsulated in a single entity known as class. Object oriented program is not concerned with the details of program rather, its main concern is the structure of the program [18]. Object-oriented paradigm is used for systematic development of complex systems, such as management information systems (MIS), complex enterprise systems, health care information systems and many more [19]. According to [20], more benefits can be achieved by combining object-oriented approach with parallel computing. A C++ class for robust optimization generates number of objects and each of them runs on single processor. Object-orientation is one of the most popular approaches used in software development industry nowadays. The reason behind its success and popularity lies in its inherent efficiency and flexibility of design [21]. Another research states that object-oriented approach provides very good results in fault prediction and performance [22]. Object-oriented applications are object driven which are well-defined and easy to communicate in order to achieve a specific solution for a problem. Reusability is also a key factor of object-oriented approach which diminishes the need of unstructured development, testing and new set of behaviors [23].

7.2 Use of Software Design Patterns for Scalability Enhancement

Software Design Patterns are reflection of real systems to set common programming practices. Design Patterns provide a reusable solution for a problem which occurs commonly. They provide a structural and behavioral model for a given problem. Patterns are commonly used in software development and they offer help in enhancing software quality [24]. Good Object-oriented design provides reusability, extensibility and maintainability. Patterns provide general rule, and a solution with good qualities of object-oriented design. Each design pattern has its own unique feature which handles the problem by providing a relationship between objects and classes and it carries out the solution [25].Patterns help to improve the design of software in such a way that we can avoid complexity of a software design.They provide a common standardized practice to follow in case of resolving a specific design issue by combining best industry practices in the solution. Design patterns can also be used as a design refinement technique to improve an application in terms of scalability, maintainability,

efficiency, testability and reusability [3]. Patterns are applied in most common object-oriented programming languages. An interoperable cloud environment using several software design patterns may provide interface to many cloud infrastructures for software developers without even knowing their specific details. To facilitate this approach an adapter can be introduced, which directly communicates with both application programming interfaces (APIs) and cloud by providing a seamless experience for developers because of the unified interface [17].

V. CONCLUSION

Achieving higher scalability of software applications on cloud is a challenging task as volume of data is growing rapidly. Adding more hardware resources is always a quick and useful option but this approach is usually costly. This research showed that software architectural improvement approaches are preferable choices for scalability enhancement. There are techniques available that can be applied to offer scalability improvements without the need for enhancement in hardware resources. Software scalability measurement methods provide desired metrics such as resource sharing, utilization and throughput by which we can verify or conform the scalability outcomes after applying different software scalability enhancement approaches. Purpose of finding multiple testing methods is to prove and compare results. We found numerous scalability improvement tools and approaches including multithreaded applications, CloudScale, AppScale, Eucalyptus, object-oriented paradigm and software design patterns. Researchers found that the use of software design patterns improve scalability, elasticity and reusability in a cost effective and efficient way. In future researchers intend to formulate a model using software design patterns that can be applied on large-scale cloud applications to enhance scalability.Muhammad Ehsan Rana possesses 20+ years of experience in academic management, lecturing and quality assurance. He holds Masters in Computer Science and Masters in Mathematics degrees and has a vast project supervision experience. He is associated with Asia Pacific University of Technology & Innovation (APU) Malaysia since December 2008. His research interest include software engineering, cloud computing, software architecture, IoT and blockchain. He has authored or co-authored many journal papers and conference publications in the areas specified.Usman Farooq is a software Engineer and Researcher in the field software engineering at APU. He has worked on several industrial projects which includes software requirement gathering, design, development and testing. His research area is software performance and scalability measurement and improvement using different methods. His research also includes improvement of software applications on cloud using software design patterns. He has worked on the projects to automate the manual file system for reputed organization. His research and development projects includes website builder to create quick websites without writing a single line of code.

Wan Nurhayati WAN AB RAHMAN is an Associate Professor at Department of Software Engineering and Information System, Faculty of Computer Science and IT, Universiti Putra Malaysia. Her research interest is towards exploration of quality factors in software and web engineering development. She graduated her Ph.D. and Master of Science both in Computer Science from Salford University, United Kingdom. She has published 50 papers including ISI index journals in the related research field. She has supervised about 20 research students for their Ph.D. and masters. She has received recognition from UPM for Ph.D. graduate on time under her supervision.

REFERENCES

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M., (2010) A view of cloud computing. Communications of the ACM, 53(4), pp.50-58.
2. Barish, G., (2002). Building scalable and high-performance Java Web applications using J2EE technology. Addison-Wesley Professional.
3. Nagy, A. &Kovari, B. (2015) Programming language neutral design pattern detection. In 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI '15). Budapest, Hungary: IEEE, pp. 215.
4. Agrawal, D., El Abbadi, A., Das, S. and Elmore, A.J., (2011) Database scalability, elasticity, and autonomy in the cloud. In International Conference on Database Systems for Advanced Applications (DASFAA '14). Hong Kong, China: Springer. pp. 2-15.
5. Kächele, S. and Hauck, F.J., (2013) Component-based scalability for cloud applications. In Proceedings of the 3rd International Workshop on Cloud Data and Platforms (CloudDP '13). Prague, Czech Republic: ACM. pp. 19-24.
6. Xiong, H., Fowley, F., Pahl, C. and Moran, N., (2014) Scalable architectures for platform-as-a-service clouds: Performance and cost analysis. In European Conference on Software Architecture (ECSA '14). Vienna, Austria: Springer International Publishing. pp. 226-233.
7. Sun, H., Wang, X., Yan, M., Tang, Y. and Liu, X., (2013) Towards a scalable PaaS for service oriented software. In International Conference on Parallel and Distributed Systems (ICPADS '13). Seoul, South Korea: IEEE pp. 522-527.
8. Kashyap, S., Min, C. and Kim, T., (2015) Scalability in the Clouds! A Myth or Reality? In Proceedings of the 6th Asia-Pacific Workshop on Systems (APSys '15). Tokyo, Japan: ACM. p. 5.
9. Krintz, C. (2013) The AppScale Cloud Platform: Enabling Portable, Scalable Web Application Deployment. IEEE Internet Computing, vol. 17, no. 2, pp. 72-75.
10. Tian, W., Su, S. & Lu, G. (2010) A Framework for Implementing and Managing Platform as a Service in a Virtual Cloud Computing Lab. In Second International Workshop on Education Technology and Computer Science (ETCS '10). Wuhan, China: IEEE, pp. 273.
11. Vaquero, Luis &Rodero-Merino, Luis &Buyya, Rajkumar. (2011). Dynamically Scaling Applications in the Cloud. ACM SIGCOMM Computer Communication Review, pp. 45-52
12. R. Khan and M. Amjad, "Smoke testing of web application based on ALM tool," 2016 International Conference on Computing, Communication and Automation (ICCCA), Noida, 2016, pp. 862-866.
13. T. Heyman, D. Preuveneers and W. Joosen (2014), "Scalar: Systematic Scalability Analysis with the Universal Scalability Law," 2014 International Conference on Future Internet of Things and Cloud, Barcelona, 2014, pp. 497-504.
14. Al Nuaimi, K., Mohamed, N., Al Nuaimi, M. and Al-Jaroodi, J., 2012, December. A survey of load balancing in cloud computing: Challenges and algorithms. In Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on (pp. 137-142). IEEE.
15. Chen, K.Y., Chang, J.M. and Hou, T.W., 2011. Multithreading in Java: Performance and scalability on multicore systems. IEEE Transactions on Computers, 60(11), pp.1521-1534.
16. Brataas, G., Stav, E., Lehrig, S., Becker, S., Kopčak, G. &Huljenic, D. (2013) CloudScale: scalability management for cloud systems. In Proceedings of the 4th ACM/SPEC International Conference on performance engineering (ICPE '13). Prague, Czech Republic: ACM. pp. 335.
17. Markoska, E., Ackovska, N., Ristov, S., Gusev, M. &Kostoska, M. (2015) Software design patterns to develop an interoperable cloud environment. In 23rd Telecommunications Forum Telfor (TELFOR '15). Belgrade, Serbia: IEEE, pp. 986.

18. Alic, D., Omanovic, S. &Giedrimas, V. (2016) Comparative analysis of functional and object-oriented programming. In 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO '16). Opatija, Croatia: IEEE. pp. 667.
19. Lo, C. & Huang, S. (2010) Applied object-oriented programming technology to ICT applications development. In Proceedings of SICE Annual Conference on Society of Instrument and Control Engineers (SICE '10). Taipei, Taiwan: IEEE. pp. 3336.
20. Pierucci, S., Buzzi Ferraris, G. (2010) A Combination of Parallel Computing and Object-oriented Programming to Improve OptimizerRobustness and Efficiency.In 20th European Symposium on Computer Aided Process Engineering (ESCAPE20). Milano, ITALY: ScienceDirect. pp. 337-342.
21. Zhang, W., Tian, P., Liu, J. & Li, J. (2009) A Method of Software Reliability Test Based on Relative Reliability in Object-Oriented Programming. Asia-Pacific Conference on Information Processing (APCIP '09). Shenzhen, China: IEEE. vol.1, no. 117, pp. 454.
22. C. Catal, B. Diri and B. Ozumut. (2007) "An Artificial Immune System Approach for Fault Prediction in Object-Oriented Software," 2nd International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX '07), Szklarska, pp. 238-245.
23. Mohan, P. & Daniel, B.K. (2006) Towards Object-Oriented Design Patterns for Reusability of Learning Objects. In Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06). Kerkrade, Netherlands: IEEE, pp. 1025.
24. G. Baumgartner, K. Lauferand V. F. Russo. (1996) "On the Interaction of Object-OrientedDesign Patterns and Programming Languages," 2nd International Conference on Dependability of Computer Systems Computer Science Technical Reports. Paper 1276.
25. Sahoo, S.K. (2011) Social object - a software design pattern. In IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS '11). Beijing, China: IEEE, pp. 580.

AUTHORS PROFILE

Muhammad Ehsan Rana possesses vast experience in academic management, lecturing and quality assurance. He holds Masters in Computer Science and Masters in Mathematics degrees and has a vast project supervision experience. He is associated with Asia Pacific University of Technology & Innovation (APU) Malaysia since December 2008. His research interest include software engineering, cloud computing, software architecture, IoT and blockchain. He has authored or co-authored many journal papers and conference publications in the areas specified.

Usman Farooq completed his Masters qualification from Asia Pacific University of Technology & Innovation (APU). He is currently working in software industry where he has worked on several industrial projects which includes software requirement gathering, design, development and testing. His research area is mainly focused on software engineering and cloud computing. He has worked on the projects to automate the manual file system for reputed organizations. His research and development projects also include web development.

Wan Nurhayati WAN AB RAHMAN is an Associate Professor at Department of Software Engineering and Information System, Faculty of Computer Science and IT, Universiti Putra Malaysia. Her research interest is towards exploration of quality factors in software and web engineering development. She graduated her Ph.D. and Master of Science both in Computer Science from Salford University, United Kingdom. She has published 50 papers including ISI index journals in the related research field. She has supervised about 20 research students for their Ph.D. and masters. She has received recognition from UPM for Ph.D. graduate on time under her supervision.