ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (973-980), Month: April - June 2015, Available at: www.researchpublish.com

Simulation and Metrics Framework for Guiding Innovation in Non-Proprietary Cloud Architectures

[1]Dr. Charles D. Madewell, [2]Dr. Danette Lance

***Abstract*: The importance of innovation to the continued success of an organization in this competitive environment is well-known and discussed. Persistent innovation can be the key to continued success and longevity. Innovation done correctly, also tends to bring about greater efficiency, competitive advantage, and enhanced profits. Thus, developing ways to understand and guide innovation so that emerging technologies can be implemented is a greatly needed capability for organizations. This is especially true for the new and highly competitive cloud services provider arena. This research samples seventy-seven report articles published in peer-reviewed research journals and created a simulation and metrics framework that can be used to help guide innovation. This framework facilitates organizations to predict the overall performance of their architecture and choose which components of the architecture merit use.**

***Keywords*: Performance, Architecture, Cloud, Innovation, Framework, Semantic.**

I. INTRODUCTION

Kotter (2012) discusses the value of major change and the ability to lead change by pointing out that major change tends to help organizations adjust to shifting business environments as well as improving their standing with their competitors. Change can lead to positioning for a brighter future for businesses. Kotter (2012) also points out that the major error that businesses have historically made is to become complacent in leading change. Complacency can lead a business to failure in a highly competitive environment such as cloud computing. Tidd and Bessant (2014) point out that innovative firms tend to grow twice as fast as their complacent competitors. Innovation can be used to improve efficiency of operations, enhance utilization of resources, increase profitability, reduce costs, gain competitive advantage, and add value to stakeholders. Thus, innovation is very important to the continued success of organizations. Tidd and Bessant (2014) also offer a process for innovation as:

1. Search for Innovation Initiatives
2. Select Innovation Initiatives
3. Implement Innovation Initiatives
4. Capture the Benefits of the Innovation Initiative.

The research shown in this article specifically facilitates the first two steps of the Tidd and Bessant (2014) process shown above.

The simulation and metrics framework developed in this research is directly applicable for use with new technology innovations that provide revolutionary improvement. It is not a framework that is geared for use with incremental innovation. More specifically, the framework presented in this article is most useful for emerging technologies where very little previous data on performance in real infrastructures exists. Thus, the framework is best used in the up-front conceptual design phase to determine which path forward to undertake. Past research and frameworks have been focused on use with communication, computer, and sensor architectures. The framework in this article extends the use of past research to the use of cloud computing architectures which have very similar needs. More specifically, this framework is geared toward architectures which are very time-sensitive and the component selection is very much focused on the performance in time. This goes hand-in-hand with the needs of cloud computing which relies heavily on processes and transfers being done in an efficient and acceptable timeframe.

Cloud computing typically exists in three types of settings. These are: public, private, and hybrid clouds. Typically, three service offerings are also utilized. These are: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The benefits of the use of cloud computing is often highlighted as offering: greater flexibility, easier scalability, reduced complexity, decreased need for capital expenditures, and more focus on core business functions.

The term component is used within this framework to mean any hardware, software, algorithm, embedded code, or combination of these items. A component is defined as any system that performs a process in which time is of great importance. Examples of components include: processors, servers, routers, switches, and such like. The framework in this article is very useful with IaaS, PaaS, and SaaS. The simulation and metrics framework in this article facilitates organizations in predicting the overall performance of their architecture and choose which components of their architecture bring about the most competitive advantage. This is accomplished by vendors supplying system-level models of components based upon early empirical information gathered in pilot testing. If a component vendor has not completed at least this level of pilot testing, then their component should not be considered by this framework. Thus, it is advantageous for component vendors to innovate quickly and bring forth early empirical data based upon pilot tests that they fund and perform. Once component vendors have their data available to share with cloud service providers, this simulation and metrics framework can be utilized to help select the most innovative components of the future-focused cloud infrastructure. This methodology also allows for cloud service providers to determine which component vendors would be most opportune to purchase (in advance) the technical data rights or licenses. This affords cloud service providers the great opportunity to provide non-proprietary solutions to their customer base. Having stated that, this article presents the simulation and metrics framework that was developed in research and analysis of seventy-seven reports published in peer-reviewed research journals. The remaining sections of this article will discuss past works, sample data, research methodology, results discussion, conclusions, and future work.

II. PAST WORKS

Past works covered in this article include three broad areas. The first covers the topic of why simulation is a good tool to use for analyzing architectures and infrastructures. The second covers the difficulties of analyzing information technology systems. Lastly, the third covers the areas of research conducted specifically to cloud computing.

Brugger (2007) pointed out that much work has been done in the past relating to queuing and protocol analysis. He noted that latency and throughput alone often fail to provide a holistic view of a network. Therefore, he recommended using more metrics and conducting a more holistic view and proposed the use of simulation and more empirical methods. Sawaya III (2007) stated that real data is great to have but is not completely necessary for modeling and researching distributed networks. Instead he emphasized more robust strategies using agent-based simulation. His study focused on the impact of shared information and used empirical data injection into agent-based simulations. His study also highlighted the usefulness of simulation for network analysis. Drake, Smith, and Peters (1995) used simulation as a tool for the planning and scheduling of systems. They proposed and on-line simulation tool which can be used for real-time decision making giving managers and controllers a decision vector. Aksin, Armony, and Mehrotra (2007) used simulation to study the modern call center. They noted the value of simulation in this application as being able to handle variability, uncertainty, and unpredictability or arrival rates and service processes. Furthermore, they stated simulation provides a means to model and understand network overloads. They also noted that simulation is good for determining and balancing loads of components or services.

Paxson and Floyd (1997) discussed the complexities of simulating the internet. They focused much of their discussion on difficulties in modeling the internet due to the morphing of the internet architecture and the changing size of the internet. They referred to the internet as a moving target. Much of their discussion focused on routing, links, bandwidth, protocols, and traffic changes. When simulation was used, they recommended holding all parameters constant and varying only the one parameter you want to study. Their research concluded that measurement, experimentation, analysis, and simulation are all needed for the evaluation of the internet but they note that simulation has the most promise. Kasunic (2001) tackled the area of system interoperability which has many similarities to architectures and infrastructures. He determined the attributes applicable in the area of interoperability of systems to include: connectivity, capacity, system overload, underutilization, under-capacity, data latency, information interpretation, and information utilization. He noted three very important points related to simulation. These were: careful analysis from well-instrumented simulation is needed, scenario-based simulation and assessment must be conducted, and multivariate analysis needs to be explored. This point on multivariate analysis directly conflicted with the single variable methodology recommended by Paxson and Floyd (1997) however.

Frey and Hasselbring (2010) gave a six step concept for migrating legacy systems to the cloud. Within their discussion of the six steps, they called out selection, evaluation in target architecture, and transformation all of which are covered in our framework. Zhang, Berre, Roman, and Huru (2009) developed a seven step methodology for migrating legacy software applications to the cloud. Their methodology included steps for architecture representation, web service generation, cloud computing platform selection, and deployment. This methodology was very similar to the six steps of Frey and Hasselbring (2010). Mohagheghi, Berre, Sadovykh, Barbier, and Benguria (2010) presented a method of working to ensure interoperability of legacy applications into cloud platforms. Their methodology was based upon the use of model-driven approaches to check for interoperability in the model before actual implementation. Hu and Klein (2009) discussed the area of security and its effect on latency when migrated to the cloud. They developed estimates of performance penalties caused by the encryption of data. Khajeh-Hosseini, Greenwood, and Sommerville (2010) discussed migrating whole information technology systems to IaaS cloud services. Although their study was interesting, it was based on cost and not performance metrics. Hao, Yen, and Thuraisingham (2009) discussed the area of migrating dynamic data services to the cloud. Much like Khajeh-Hosseini and et. al. (2010), their analysis was based upon cost and not performance. They did note, however, that it is very important to understand the infrastructure and computational resources necessary before migrating to the cloud. The simulation and metrics framework developed in our research addresses many of the shortcomings of these cloud based research methodologies listed above.

III. METHODOLOGY

This research uses a mixed-mode research method using both qualitative and quantitative analysis. The sample data set was seventy-seven research reports published in peer-reviewed research journals. This sample set spanned seven major report categories which are shown below in Figure 1.0.

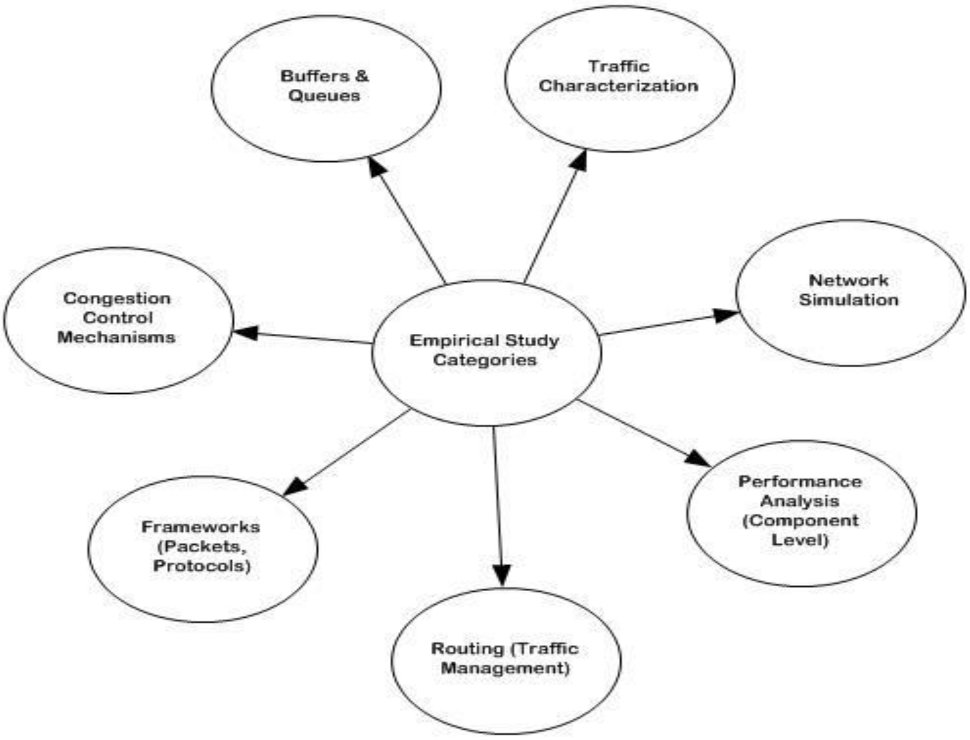


Figure 1.0, Report Categories

The final sample set consisted of seventy-seven samples that included eleven samples for each of the seven major report categories. The final sample set is shown in Figure 2.0.

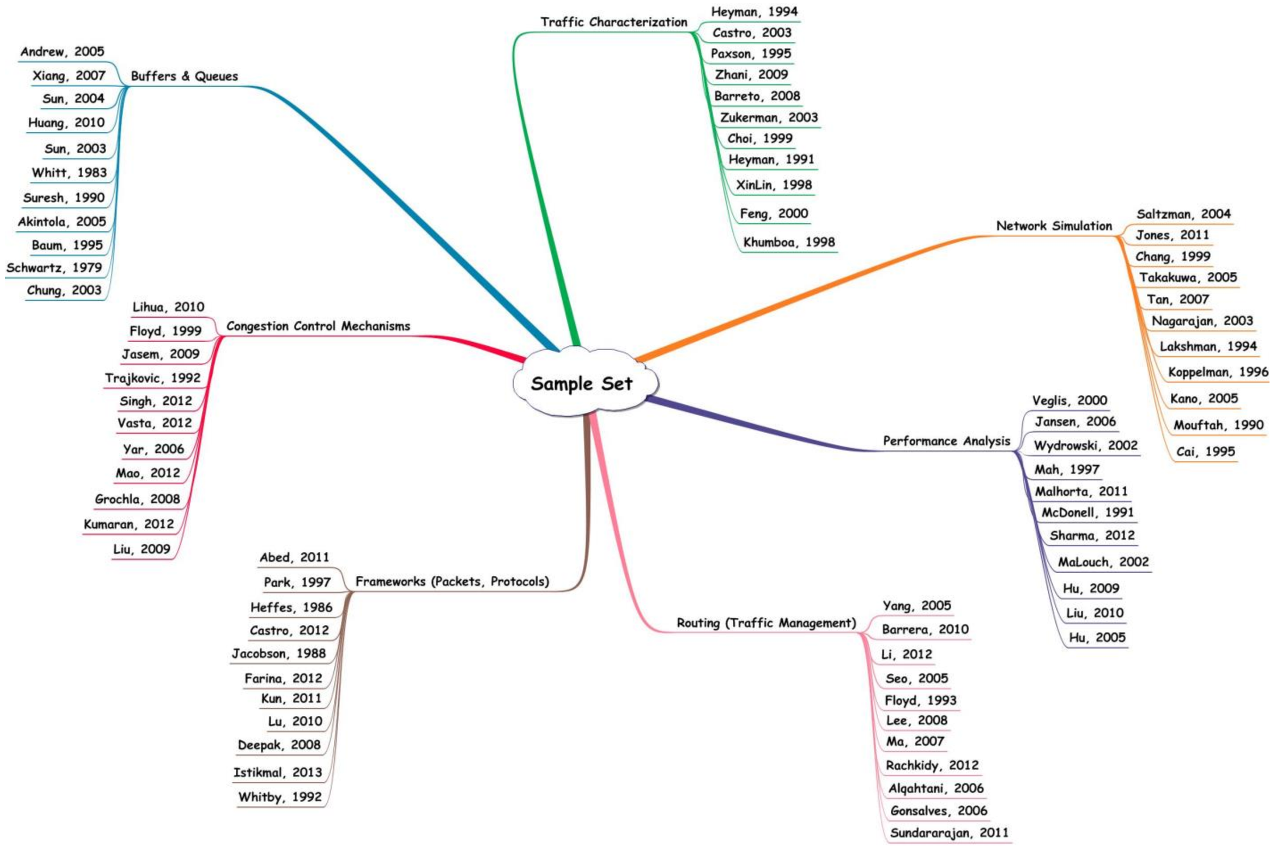


Figure 2.0, Actual Sample Set

The overall methodology of this research was:

1. Research nomenclature
2. Categorize nomenclature and create semantic mapping
3. Exercise the semantic mapping on all samples
4. Conduct the quantitative correlational study.

The first step of the process was very time-consuming in that it required thorough and detailed reading of each of the seventy-seven sample reports in order to understand the breadth of nomenclature used for metrics as well as analytical methods utilized. Once that was complete, the nomenclature was categorized using thematic coding (Boyatzis, 1998) and a qualitative method based upon frequency count. Metric terms with like meanings were categorized. The name of the category was chosen by using the metric term with the highest frequency count. The semantic mapping was created by repeating this for all metric categories. Then, the semantic mapping was exercised on all seventy-seven samples so that like nomenclature was used in the entire sample set. Finally, the qualitative (correlational study) portion of this experiment was done and the results were recorded for each metric category and analytical method used.

Triangulation was used as the method for providing validity for the qualitative portion of this research. Robustness and validity was provided in the sample set triangulation which spanned:

- Samples from years 1984 to 2013
- Seven (7) report categories
- Twenty-two (22) countries of origin
- Multiple prestigious universities (both domestic and international)

- Twenty-two (22) academic departments
- One hundred and eighty-nine (189) academic researchers
- Thirty-five (35) technical areas
- Eight (8) government labs
- Multiple federally funded or National Science Foundation funded projects.

Seventy-seven samples also provided the statistical significance needed for the quantitative (correlational study) portion of this research. The hypothesis and alternate hypothesis of this research is:

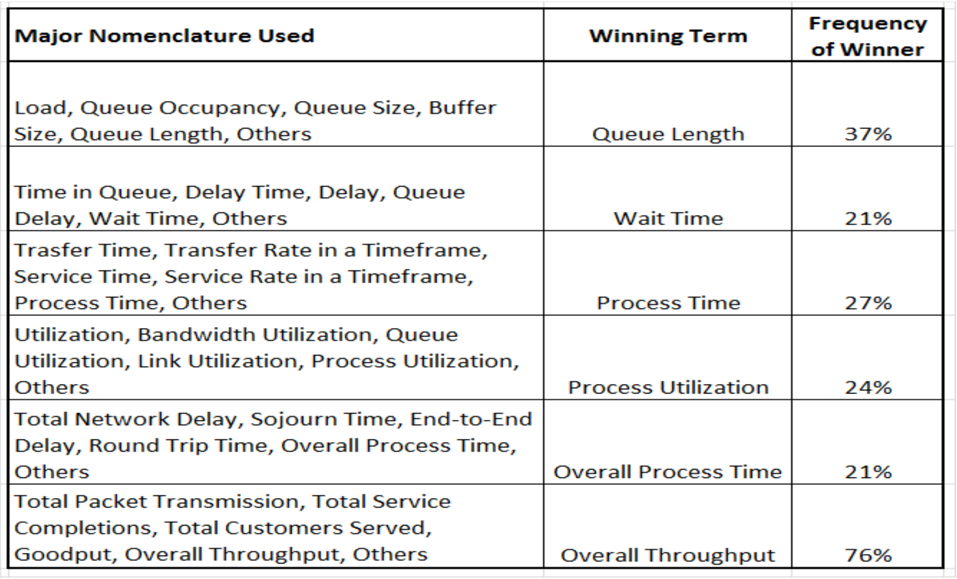
Hypothesis: There exists a set of metrics and analytical methods that has a correlation coefficient of greater than or equal to 0.55 to the baseline case.

Alternate Hypothesis: A set of metrics and analytical methods does not exist having a correlation coefficient of greater than or equal to 0.55 to the baseline case.

The proof criterion for these hypotheses was a positive Pearson correlation coefficient of 0.55 or higher with a 95% confidence interval. With seventy-seven samples and a 0.05 uncertainty level, the Pearson correlation coefficient was found to have plus or minus standard deviation of 0.207. This was found by interpolating the confidence interval between 75 and 82 samples in the Pearson's column of Table 9 in the Naval Postgraduate School document, *Sample Size for Correlation Estimates* (Salar, 1983, pp.44). Only metrics or analytical methods found to have a minimum value of a positive 0.55 correlation coefficient were reported.

IV. RESULTS AND DISCUSSION

Each of the seventy-seven samples shown above in Figure 2.0 was read in detail for definition of terms and intended use. This was a very time comprehensive and time-consuming process. Metric and method data from each sample was then input in numerical format into the collection sheets. Metrics of varying nomenclature were thematically grouped into similarly defined bins. A frequency count was then calculated for each unique phrasing of the metric nomenclature and the phrasing with the highest frequency was chosen as the winner. All phrasing of nomenclature for similarly defined metrics were mapped to the winning phrase with the highest frequency within each metric bin. This semantic mapping was specifically developed to be executed on the sample set in order to normalize varying metric phrasings in preparation for the correlational analyses. The results of this thematic coding and semantic mapping are shown below in Figure 3.0.

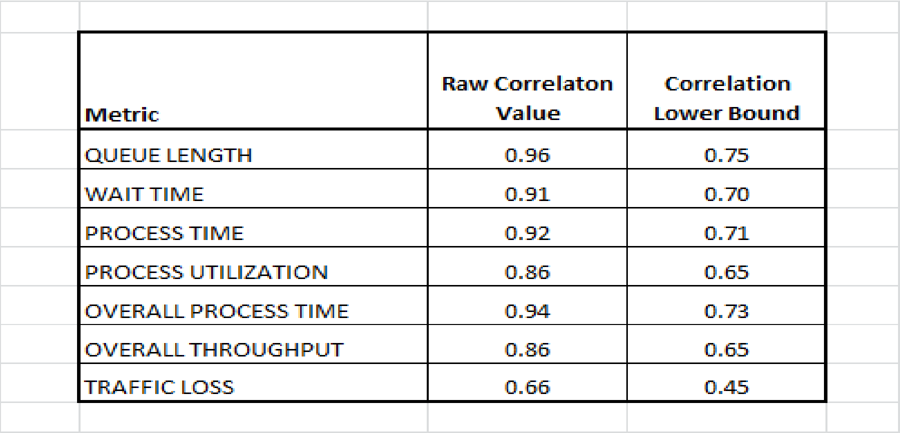


Major Nomenclature Used	Winning Term	Frequency of Winner
Load, Queue Occupancy, Queue Size, Buffer Size, Queue Length, Others	Queue Length	37%
Time in Queue, Delay Time, Delay, Queue Delay, Wait Time, Others	Wait Time	21%
Trasfer Time, Transfer Rate in a Timeframe, Service Time, Service Rate in a Timeframe, Process Time, Others	Process Time	27%
Utilization, Bandwidth Utilization, Queue Utilization, Link Utilization, Process Utilization, Others	Process Utilization	24%
Total Network Delay, Sojourn Time, End-to-End Delay, Round Trip Time, Overall Process Time, Others	Overall Process Time	21%
Total Packet Transmission, Total Service Completions, Total Customers Served, Goodput, Overall Throughput, Others	Overall Throughput	76%

Figure 3.0, Thematic Coding and Final Semantic Mapping

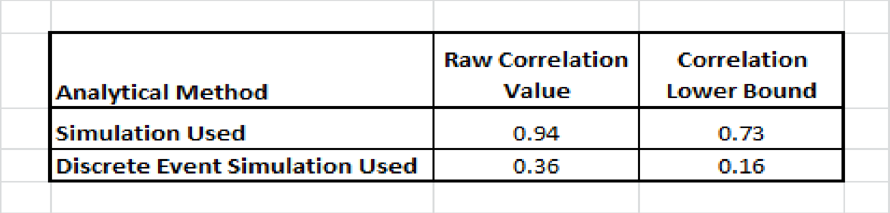
The semantic mapping found in this research and shown above in Figure 3.0 was then executed on the entire sample set to categorize the metrics into bins. Next, the sample set was coded with a zero (0) when a metric or method was not present and a one (1) when the metric or method was present. The sample set was then compared to the baseline set which was coded with all ones to represent the ideal case where each metric and method is present. The Pearson correlation coefficient was then calculated individually for each metric and each method by conducting a comparison between the sample set and baseline set. The final correlation coefficients for the metrics and methods were then recorded and are shown below in Figure 4.0 and Figure 5.0, respectively.

Of all the metrics found in this research, six metrics were found to be consistent and meet the proof criteria of 0.55 for the lower bound correlation coefficient. The metrics that met the proof criteria were: queue length, wait time, process time, process utilization, overall process time, and overall throughput. Queue length, wait time, process time, and process utilization are more component-based metrics while overall process time and overall throughput are more overall architecture performance metrics. Traffic loss is shown as an example metric that did not meet the proof criteria. Traffic loss can be calculated by subtracting the traffic within system (in queue or in process) plus overall throughput from the total traffic input into the system. Traffic loss is therefore a secondary metric since it is a metric that can be calculated from primary metrics. These metrics are shown in Figure 4.0.



Metric	Raw Correlaton Value	Correlation Lower Bound
QUEUE LENGTH	0.96	0.75
WAIT TIME	0.91	0.70
PROCESS TIME	0.92	0.71
PROCESS UTILIZATION	0.86	0.65
OVERALL PROCESS TIME	0.94	0.73
OVERALL THROUGHPUT	0.86	0.65
TRAFFIC LOSS	0.66	0.45

Figure 4.0, Metric Correlation Results



Analytical Method	Raw Correlation Value	Correlation Lower Bound
Simulation Used	0.94	0.73
Discrete Event Simulation Used	0.36	0.16

Figure 5.0, Analytical Method Correlation Results

With respect to the analytical methods, this research found that multiple analytical methods were used in the analysis of architectures and components. The main methods included: test, closed form mathematical solutions/mathematical modeling, computer simulation, or some hybrid of these three. Of all these analytical methods recorded, only one stood out as dominant and this method was computer simulation. All other analytical methods were infinitesimal as compared to the frequency of use of computer simulation. Within computer simulation, discrete-event simulation was often specifically mentioned. This is also highlighted in Figure 5.0.

V. CONCLUSIONS

This research found that there was a very consistent and credible set of metrics as well as an analytical method that can be used as a framework in the evaluation of architectures. This simulation and metrics framework can be used for the analysis of new architectures, components, or combinations thereof. This makes the framework very useful for guiding innovation especially in the very early stages of development of emerging technologies. This framework is also very useful for analyzing time critical architectures.

To help illustrate the utility of this framework, an example scenario is discussed next on how a cloud service vendor could solicit a request for proposals (RFP) for emerging technologies which would be evaluated prior to purchasing or acquiring. Along with this RFP, the cloud service vendor would supply a specification that outlines the specific requirements that must be met along with all interoperability and interface constraints. The cloud service vendor would distribute the technical reference architecture for their cloud service and request vendors interested in new work send models of the performance of their components. The cloud service vendor's overall requirements of the system would state that the system shall receive all messages in a specified format from one or more sensors, process the message to pinpoint a particular target, transfer the output of processor unit to a server unit, run algorithms to optimize the use of a set of four final resource nodes, and send a command to one of the four final resource nodes to perform the terminal function. The functional block diagram (simplistic example) for their technical reference architecture is shown below in Figure 6.0.

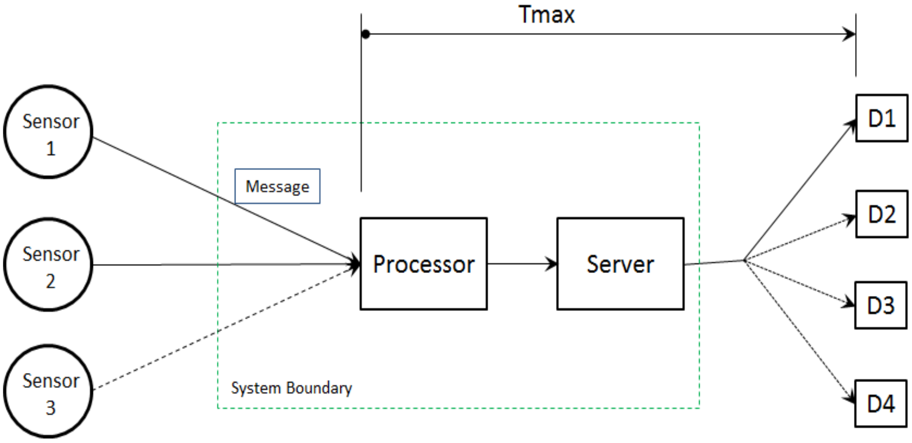


Figure 6.0, Notional System View from Cloud Service Vendor RFP

The cloud service vendor requirements would also state that the message traffic bandwidth into and out of the system is Gigabit Ethernet and therefore there are no system concerns on message traffic bandwidth. The main performance requirement stated by the cloud service vendor requirement is the total time from entry into the process to command reached at one of the four final resource node (D1, D2, D3, or D4) is less than or equal to Tmax. Again, Tmax would be required due to the time criticality of the system. Since the component vendor would be expected to provide processor and server components in this case, Tmax would therefore represent the metric of overall process time. For architecture completeness, the cloud service vendor requirement would further specify that the latency between server and final resource node should be assumed to be 0.68 seconds.

Knowing these facts, the vendors could also construct a simulation of the cloud architecture and further develop their components to perform in an optimal state. In fact, it would be to the component vendor's advantage to do so. Once the vendors develop their component system(s) they would then perform testing to provide statistically significant system models of the performance of their component. The system model required would simply be either an empirical model or a cumulative distribution function (CDF) model that simulates the input, processing time, and output performance. Then, the cloud service vendor would take those component models from various vendors, add them into their own technical reference architecture simulation model, and run comparative analysis between component vendor alternatives. The comparative analysis would be based upon the six metrics found in this research. Once the analysis is done, the cloud service vendor would have great awareness which emerging technology for components performs better and can choose to invest in that technology. In this manner, the simulation and metrics framework can be used to guide innovation. For more comprehensive documentation of the framework as well as its utilization, readers are referred to [17].

VI. FUTURE WORK

There are two prime areas that future work can be conducted pertaining to this research. The first pertains to the robustness of the semantic mapping and the second is in reference to the final metric set. The qualitative analysis potion of this research in which the thematic coding and categorization of metric nomenclature was accomplished was very manual and time-consuming. This made it prohibitive to analyze sample size of much more than eighty samples. Taking much larger samples and utilizing automated (computer-based) contextual analytics may make the semantic mapping more robust. This is the first recommendation for future research. The second recommendation is to exercise the semantic mapping and correlational analysis on a much larger sample set (thousands of samples) to observe if more metrics emerge by meeting the proof criteria. Again, for this to work efficiently, an automated, computer-based method would need to be employed in order to properly code the metrics in each sample with a zero (not present) or a one (present).

REFERENCES

[1] Kotter, J. (2012). Leading change. Boston, Massachusetts: Harvard Business Review Press.

[2] Tidd, J. & Bessant, J. (2014). Managing innovation: integrating technological, market and organizational change (5^{th} ed.). West Sussex, United Kingdom: John Wiley & Sons Ltd.

[3] Brugger, S. (2007). An approach for computer network comparison [Whitepaper]. University of California-Davis. Retrieved from http://www.bruggerink.com/~zow/GradSchool/brugger_approach_netcompare.pdf

[4] Sawaya III, W. (2007). Using empirical demand data and common random numbers in an agent-based simulation of a distribution network. In Proceedings of the 2007 Winter Simulation Conference, 1947-1952.

[5] Drake, G., Smith, J., & Peters, B. (1995). Simulation as a planning and scheduling tool for flexible manufacturing systems. In Proceedings of the 27th conference on Winter Simulation, 805-812. IEEE Computer Society.

[6] Aksin, Z., Armony, M., & Mehrotra, V. (2007). The Modern Call Center: A Multi-Disciplinary Perspective on Operations Management Research. Production and Operations Management, 16(6), 665-688.

[7] Paxson, V., & Floyd, S. (1997). Why we don't know how to simulate the Internet. In Proceedings of the 29th conference on Winter Simulation, 1037-1044. IEEE Computer Society.

[8] Kasunic, M. (2001). Measuring systems interoperability: Challenges and opportunities. Pittsburg, Pennsylvania: Carnegie-Mellon University Software Engineering Institute.

[9] Frey, S., & Hasselbring, W. (2010). Model-based migration of legacy software systems to scalable and resource-efficient cloud-based applications: The CloudMIG approach. In CLOUD COMPUTING 2010, The First International Conference on Cloud Computing, GRIDs, and Virtualization, 155-158.

[10] Zhang, W., Berre, A., Roman, D., & Huru, H. (2009). Migrating legacy applications to the service cloud. In 14th Conference companion on Object Oriented Programming Systems Languages and Applications (OOPSLA 2009), 59-68.

[11] Mohagheghi, P., Berre, A., Sadovykh, A., Barbier, F., & Benguria, G. (2010). Reuse and migration of legacy systems to interoperable cloud services-the REMICS project. In Proceedings of Mda4ServiceCloud, 10.

[12] Hu, J., & Klein, A. (2009). A benchmark of transparent data encryption for migration of web applications in the cloud. In Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on, 735-740. IEEE.

[13] Khajeh-Hosseini, A., Greenwood, D., & Sommerville, I. (2010). Cloud migration: A case study of migrating an enterprise it system to IaaS. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, 450-457. IEEE.

[14] Hao, W., Yen, I., & Thuraisingham, B. (2009). Dynamic service and data migration in the clouds. In Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International, 2, 134-139. IEEE.

[15] Boyatzis, R. (1998). Transforming Qualitative Information: Thematic Analysis and Code Development. Thousand Oaks, CA: Sage, ISBN: 0-7619-0961-3

[16] Salar, K. (1983). Sample size for correlation estimates. Naval Postgraduate School Monterey CA. Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA219810

[17] Madewell, C. (2014). Technical metrics for identifying key processor components for acquisition in open system architectures and non-proprietary infrastructures. Retrieved from ProQuest Dissertations and Theses Database.

Author's Biographies:

Dr. Charles D. Madewell is an independent researcher and author covering topics on information technology, architecture analysis, sensors architectures, communication networks, cloud computing, data science, emerging technology, innovation, and strategy.

Dr. Lance worked in the information technology industry for fifteen years before joining academia and moving on to academic administration. She enjoys working with research and graduate students as well as teaching information technology and business courses at all levels.