Cloud-based Decision Support Systems and Availability Context: The Probability of Successful Decision Outcomes

Stephen Russell
Department of Information Systems & Technology Management
The George Washington University
russells@gwu.edu

Victoria Yoon, Guisseppi Forgionne
Information Systems Department
University of Maryland, Baltimore County
yoon@umbc.edu, forgionn@umbc.edu

Abstract

In an age of cloud computing, mobile users, and wireless networks, the availability of decision support related computing resources can no longer guarantee five-nines (99.999%) availability but the dependence on decision support systems is ever increasing. If not already, the likelihood of obtaining accurate deterministic advice from these systems will become critical information. This study proposes a probabilistic model that maps decision resource availability to correct decision outcomes. Grounded in system reliability theory, the probability functions are given and developed. The model is evaluated with a simulated decision opportunity and the outcome of the experimentation is quantified using a goodness of fit measure and ANOVA testing.

Keywords: Context Aware Computing, Availability Awareness, Decision Support Systems

1. Introduction

The availability of pervasive networking, inexpensive storage, and high performance computing has created the foundation for a broad range of new approaches capable of delivering on the promise of cloud computing. The "cloud" can be defined as the environment where computing resources are hosted in and used from the distributed Internet environment. Cloud computing extends the notion of desktop computing to the scalability and virtualization of distributed processing servers on the Internet. Within the cloud, an application is built using resources from multiple services and potentially from multiple locations. Cloud computing is typically implemented as a software-as-a-service model. This removes the burden of software updates and server maintenance from business and other users. One of the most significant benefits of cloud computing is that it provides a way to increase capacity or add capabilities dynamically as they are needed without investing in new infrastructure, training new personnel, or licensing new software. For many companies, the true value of cloud computing beyond the "pay as you go" model is the time to value ratio and reduced risk, compared to in-house implementations.

A user of cloud computing services doesn't necessarily care about how they are implemented, what technologies are used, or how they are managed. The most significant concern is that there is access to it and that the service/application has a level of reliability necessary to meet the functional requirements (Buyya et al. 2008). In this context, reliability may be one of the most significant issues facing cloud computing. Very few applications have been able to achieve

Springer To be published by Springer-Verlag Berlin Heidelberg in Lecture Notes in Business Information Processing. Lecture Notes in Business Information Processing LNBIP

100% availability. Moreover, while a cloud computing application may be 100% reliable, users' access to it may not be. This issue is complicated in the face of mobile users and access using 3G wireless networking technologies.

Consider the following example. A user gets on a train and begins conducting a session with a cloud application-based decision support system. As the train begins to move, the environmental computing conditions begin to change. If the train were to go through a tunnel, the user's connectivity may temporarily be interrupted; potentially regaining the connection when the train emerges from the tunnel. If a user were made to know that the tunnel was coming and connectivity may be lost, the user could adjust their interaction with the decision support system (DSS), or even plan their usage of the system in a manner that avoids complete or disastrous interruption, e.g. downloading data/information and interacting locally. While it may not be possible to know exactly when and for how long a session may be interrupted, it may be possible to know the likelihood of interruption occurrences and durations. The work of Russell et al. (2008) has demonstrated that knowledge of computing resources' availability can affect the support provided by applications and even affect decisions associated with that use. Russell et al's research indicates that it is important to extend awareness of computing resources' such as data, network connectivity, or software applications to system users and not just implement it as an internal hardware or software algorithm.

This work extends Russell's et al's work as the basis for use with cloud computing decision support applications. Conceptually, the research in this paper seeks to develop a probabilistic model that can be used in conjunction with collected availability context data. To further the understanding of the effect of availability context information on decision making, this paper proposes a probabilistic model to describe the relationship between decision outcome accuracy and evaluates the model using a simulation. In Section 2, a discussion of existing reliability and availability-related technologies is discussed, followed by the introduction of a probabilistic reliability model for decision support systems. In section 2.3, a probabilistic model that maps decision outcome to DSS availability is presented. Section 3 explains the simulation experiment that evaluates the decision outcome model. Section 4 presents the results of this experiment followed by conclusions in Section 5.

2. System Reliability, Availability, and Decision Making

Many people confuse or interchange the concepts of system reliability and availability. Before examining availability in depth, it is helpful to have an understanding of the distinction between the two concepts. These concepts are most frequently discussed in a hardware or equipment context, but also have applicability to software. Most often associated with component or system failure, reliability is a measure of the likelihood that a system or process will perform its designed function for a specified period of time. Availability is a relative measure of the *extent* that a system can perform its designed function (Bhagwan et al. 2003). As a relative measure, availability includes metrics such as delays, congestion, and loading. To illustrate, the reliability of a system could be determined by the number of failures and the amount of time between them. So, if a system was broken twice as much as it was working, it would have 50% reliability. This same system would be unavailable 50% of the time. However, a system does not have to be broken to be unavailable. Consider if a system was so busy processing data that it could not handle any additional tasks. The system is not broken, it is just temporarily busy. Simply put, system availability includes system reliability, as well as delay-oriented interruptions.

Reliability and availability both affect the systems that business people have become dependent on and there has been significant research on quantifying and minimizing system outages of any sort. Much of this research has centered on computer system components or hardware. Reussner et al. (2003) use rich architecture definition language (RADL) to predict component reliability through compositional analysis of usage profiles and of environment component reliability. Mikic-Rakic et al. (2005) propose a fast approximating solution for relating how software systems' environmental deployment (wired, mobile, grid, etc.) that will affect its availability. Henson (2006) suggests a method to improve hard disk reliability by dividing the hard disk file system data into small, individually repairable fault–isolation domains while preserving normal file system semantics. Dai et al. (2003) propose a model for a centralized heterogeneous distributed system and examines the distributed service reliability which is defined as the probability of successfully providing a service in a distributed environment. All of these prior works concentrate on the system itself rather than the impact of low availability. The emphasis on hardware components is typical of system availability research and seldom do these types of studies simultaneously address software service availability.

With the recent interest in web services and service composition there has been a renewed research effort concentrating on software availability. In the web service domain, hardware is seen as an underlying component, on which software functions run. These independent services are loosely coupled and assembled to perform more complex functions. As a result, the availability of software is critical to web service use and composition. Notwithstanding its general consideration as an underlying component, even in a web service context, there is an implicit emphasis on systems hardware. For example, Salas et al. (2006) proposed a method for providing an infrastructure that replicates web services across hardware on a wide area network. Sung et al. (2007) put forward dynamic cluster configuration using a server agent and a service description server as a solution to improve computer service availability. Other research in this area adopts the use of context information, such as physical location about the service hardware (Ibach et al. 2004) or using network bandwidth reservation (Xu et al. 2003) to improve availability. Research in web service availability has primarily addressed the issue of being able to compose the set of services that are necessary to fulfill a process or function. Like research in hardware availability, little attention is given to the impact of availability on outcomes resulting from system usage.

The approach proposed in this study addresses the two limitations noted above (a focus on hardware only and the exclusion of outcome impact). This prior research provides a solid foundation for quantifying, improving, and addressing system reliability (and subsequently availability) but these works have not extended this information to decision making outcomes. By mapping system availability to decision outcomes, users of cloud computing-based DSS may be made aware of the likelihood that a successful result can be obtained from an engagement with the system, within the time constraints of the decision opportunity.

2.1 Existing Availability-Related Technologies

In a decision support context, availability should be considered not only from a systems viewpoint, but also from the perspective of decision-related resources. These resources may be models, data, services, agents, processing, output devices, other decision makers, or even the decision maker requesting support. While not all decision support systems and scenarios require external or distributed resources to provide guidance to a decision maker, most contemporary DSS utilize the benefits provided by computer networks. The introduction of networking and distributed resources adds another dimension to the issue of resource availability. As discussed above, there are many solutions to determine or quantify if hardware or a software service is reliably "on-line." However, availability goes beyond this on-off notion and encompasses more than resource online/offline - operating/failed status information.

The obvious question is: *how might details about resources' availability be obtained*? Research from other domains provides answers to this question. The first domain is high-availability computing. Research in this area has already identified methods to monitor and evaluate hardware related statuses such as power (Chakraborty et al. 2006; Rahmati et al. 2007), network characteristics (Roughan et al. 2004; Shahram et al. 2006), computer components (Brown et al. 1999; Weatherspoon et al. 2005), processing/computing load (Zhoujun et al. 2007), and storage (Blake et al. 2003).

The second domain provides status of resources that can be considered software services. Software services provide a layer of abstraction for a full range of programmable functions and data. Research in the area of web service composition and quality of service (QoS) can provide solutions delivering awareness knowledge for these types of resources. Quality of service is often defined as the probability that a network or service will meet a defined provision contract. This probability could be used by agents to forecast the likelihood of resource interruption as well as potentially quantitatively predict outage durations. There is a significant amount of research studying applications using QoS and QoS monitoring for service level agreements, adaptation to changing system conditions, and web service composition (Ali et al. 2004; Loyall et al. 1998; Menasce 2004; Thio et al. 2005). Web service composition is a particularly active research area, rich with solutions for service availability, because of the critical nature of this information for process scheduling and execution planning (Peer 2005; Pistore et al. 2004).

A third domain provides availability information on human users of decision support systems. From the perspective of "users as a resource," human computer interaction research has provided several availability-oriented solutions. Most of the efforts have focused on detecting *if a user is* and not necessarily *when the user will become* online and available (Begole et al. 2004; Danninger et al. 2006; Muhlenbrock et al. 2004). However, there are probabilistic models that can provide forecasts for humans' presence and availability. Horvitz et al. (2002) developed a prototype service intended to support collaboration and communication by learning predictive models that provide forecasts of users' presence and availability. To accomplish this, they collected data about user activity and proximity to multiple devices and combined this with analyzed content of users' calendars.

The research in these three domains provides reasonable methods for obtaining quantitative availability information regarding decision resources such as hardware, network, and software services (e.g. web server, database servers, and business logic applications), as well as collaborators and systems users. Because the research conducted in these other domains delivers viable solutions for this problem, the probability model proposed in this work does not focus on this issue. Instead, the model is grounded in system reliability theory, extended to decision-related resource availability, and focused on how correct decision outcome may be a function of these resources' availability.

2.2 Decision-related Computing Resource Availability as a System Reliability Problem

Particularly in the case of DSS, decision resource availability may be analogous to system reliability. This is because a decision resource may be the DSS itself, another system, data from some storage-system, a network communication medium, an output device (e.g. monitors, printers, etc.), collaborating system users, or even decision makers themselves. One way to view a decision resource is as a hierarchical structure of interacting functions. This hierarchical structure may have *1 – n* levels and when the resource is a computing device, these levels would encompass hardware, firmware, and software. The underlying concept is that a decision-resource, via its hierarchy, relates to a number of sub-resources at lower levels of the hierarchy. The functions in a resource hierarchy act and interact together to provide the resource at the top of the hierarchy. Upper level functions depend on lower levels to ensure their reliability and availability.

Figure 1 illustrates a generalized hierarchy. While every level may not be necessary for every resource, the level dependencies are evident. Consider an example where a DSS provides guidance by presenting information on a graphical map, e.g. a spatial DSS. This DSS may require data from a remote website that converts street addresses to latitude and longitude. This remote website/data would be considered a resource. This resource depends on a hierarchy of dependent functions to be able to serve its purpose. Examining Figure 1, without power/electricity, there is no connectivity or anything else above power. Without connectivity the data in storage cannot be delivered. Without storage, processing cannot occur; there is nothing to process. Without processing, the software cannot operate, and if the software (web server, address-to-lat/long converter, or host operating system) fails, the resource cannot respond to the DSS's request. There is an implicit dependency from top to bottom but not the other way. It is possible to not have connectivity, yet have power available and so on.

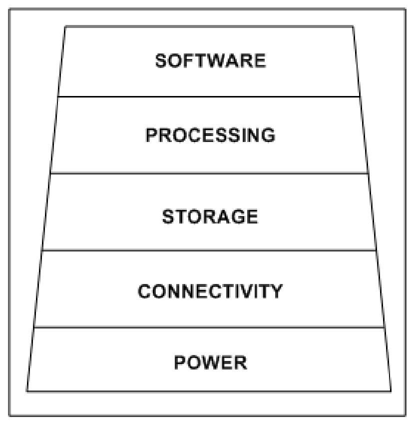

Figure 1. Decision resource hierarchy

Typically the availability of components, software, and systems is given in terms of its likelihood of being available; not the inverse (the likelihood it is *not* available). In the reliability practice there is a concept called 5 nines that refers to uptime (availability). If a system is said to have 5 nines reliability it is available 99.999% of the time. This translates to being *unavailable* 31.5 seconds per year. This measure is complicated in systems where there are dependencies, as is the case with a decision resource. The hierarchical dependence seen in a decision resource would suggest that the probability of a resource being available can be viewed as the product of the levels in its hierarchy. Each item in the hierarchy is independent unto itself but dependent on the level below it. Therefore, if each hierarchy level's availability is given in terms of it its potential availability (the likelihood it is available), probability theory can be applied. Because the function at each level in the hierarchy is independent in terms of its availability and dependent on lower levels in terms of its unavailability, the multiplicative rule of probability applies. Equation (1) shows the generalized form.

$$P(A \cap B) = P(A)P(B) \tag{1}$$

Building on this theory and applying it to Figure 1, let W=Power, C=Connectivity, G=Storage, O=Processing, and T=Software. The probability of availability for each level in the hierarchy can be given according to Table 1. Because availability is most commonly expressed in positive terms (e.g. uptime instead of downtime) the focus of Table 1 is on the probability that each level *is* available. Due to the dependency on lower levels, the probability that each level *is not* available would be additive. Table 1 shows the probability that each level in the hierarchy is determined by multiplying the previous levels' probability by the current level's probability and the overall availability of the decision resource is given by the probability of the topmost level. This leads to the generalized equation shown in (2) where $P(R)$ represents the probability of the decision resource (R), and F represents the function at each level, for all levels $i=1$ through n.

LEVEL	FUNCTION	FUNCTION PROBABILITY	LEVEL PROBABILITY
1	POWER	*P(W)*	*P(W)*
2	CONNECTIVITY	*P(C)*	*P(W) * P(C)*
3	STORAGE	*P(G)*	*P(W) * P(C) * P(G)*
4	PROCESSING	*P(O)*	*P(W) * P(C) * P(G) * P(O)*
5	SOFTWARE	*P(T)*	*P(W) * P(C) * P(G) * P(O) * P(T)*

Table 1. Hierarchy level availability probability

$$P(R) = \prod_{i=1}^{n} P(F_i) \tag{2}$$

Extending the generalized resource availability equation in (2), all of the resources necessary for decision support can be accounted for similarly as a set of dependent resources. In this case, all the resources are necessary for the system to provide direct advice and therefore dependent in the context of the DSS provided solution. It follows that the availability of the system is given by Equation (3), where $P(S)$ denotes the availability of all the decision resources and $P(F)$ is the resource level availability, for all levels $i=1$ to n, for all resources $r=1$ to t.

$$P(S) = \prod_{r=1}^{t} \left(\prod_{i=1}^{n} P(F_i) \right)_r \tag{3}$$

Figure 2 shows a DSS that has 3 dependent resources. To illustrate with the earlier example, Resource A would represent data stored in a data warehouse. Resource B is the address to latitude and longitude web service, and Resource C would be the local processor that provides the graphical mapping. The availability of the DSS's guidance for this decision problem would be: *.85 * .50 * .75 = .319*, or 31.9% availability.

Figure 2 however, does not account for redundant resources. A redundant resource duplicates the functionality of another resource. Introducing redundant resources generally increases the overall system availability because dependencies are distributed across multiple resources. In this context parallel resources are introduced. The availability of redundant resources (or redundant levels within a resource) is given by Equation (4), where $P(RR)$ represents the availability of the redundant resource, for all resources comprising the redundant resources *r=1* to *t*. Redundant levels of a resource hierarchy follows this same formula replacing $P(R)$ with the probability of the redundant level's function availability.

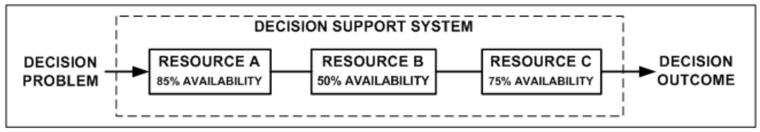

Figure 2. A DSS with dependent resources

$$P(RR) = 1 - \prod_{r=1}^{t} (1 - P(R_r)) \tag{4}$$

A generalized equation can be given to describe DSS that have redundant or alternate resources as shown in Equation (5). In equation (5), probabilities for redundant resources $P(X)$, are calculated separately from non-redundant resources $P(R)$. As previously discussed, the multiplicative rule of probability applies for all resources. If there are no redundant resources, this portion of the problem is removed from the equation. Similarly, if all the resources are redundant, the part of the equation that determines the availability of non-redundant resources is removed. Each redundant resource's *unavailability* is calculated over resource *j=1* to *v*. Then that value is subtracted from 1; giving the availability of the composite redundant resource. That result is then multiplied by other redundant resources *q=1* to *u* to determine the availability of all redundant resources.

$$P(S) = \left(\prod_{r=1}^{t} P(R_r)\right)\left(\prod_{q=1}^{u}\left(1 - \left(\prod_{j=1}^{v}\left(1 - P(X_i)\right)\right)\right)_q\right) \tag{5}$$

Figure 3 extends the DSS in Figure 2 with redundancy in resource B. Illustrating again with the earlier example, Resource A would represent data stored in a data warehouse and Resource C is the local processor providing graphical mapping. Resource B remains the address to latitude and longitude service; only this time there is more than one provider from where the address to lat/long conversion can be obtained. To determine the probability of DSS guidance availability $P(S)$, the probability of the redundant resource must first be determined and then applied as a single value to the remainder of the system. Because all of the redundant resources must fail before the availability of that resource is reduced, the calculation's use of the likelihood of the resource being *unavailable* takes this into account. The availability probability of the redundant resource (B) in Figure 3 is: *1 - ((1-.50) * (1-.50)) = .75*. Once this is calculated, the problem is the same as the previous example, except .75 is used for resource B: *.85 * .75 * .75 = .478*, or 47.8% availability.

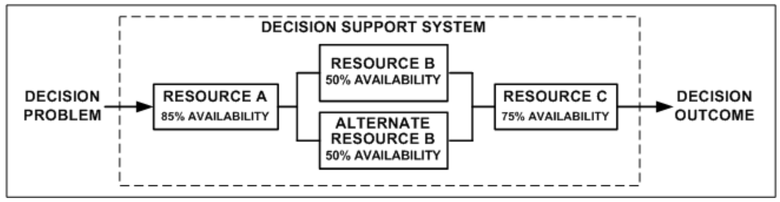

Figure 3. A DSS with dependent and redundant resources

The overall resource availability $P(S)$ provides a deterministic value for the likelihood that the DSS can supply advice based on the necessary resources. In a data-oriented decision support context, where the DSS provides guidance that is assumed to be correct based on this data, the overall resource availability can be directly mapped to the probability of accurate decision guidance outcomes.

2.3 Decision Outcome as a Function of Resource Availability: A Probabilistic Model

The equation shown in the previous section is part of the necessary probabilities to predict outcome success. Equation 5 will provide the likelihood of decision resource availability. This assumes that if available, the DSS will provide a correct and exact answer to the decision problem. However, it is possible for the decision maker to make the correct choice without assistance from the DSS. Therefore, the complete probability of successful outcome must include the probability the decision maker makes a choice without assistance from the DSS. Equation (6) defines the probability of a correct decision outcome $P(C)$ as: the probability decision-resources are available $P(S)$, plus the probability that the decision maker selects the correct answer from the possible alternatives $P(A)$ without the decision support system.

$$P(C) = P(S) + P(A) \tag{6}$$

Uncertainty about the likelihood of success can be quantified and provided to the decision maker, before beginning their interaction with the DSS using Equation (6). There is a second benefit of this probability model that would exist when the decision opportunity is time constrained. Consider the case when a resource is unavailable for some limited time duration. Ordinarily limited unavailability would introduce additional uncertainty in the decision making process because the decision maker does not know when or if the unavailable resource will become available. The decision maker has to decide whether or not to wait for the resource to become available or to proceed without the assistance of the DSS. Given the probabilistic model shown in Equation (6), the decision maker could make that wait-don't wait decision with insight into the resource's availability. The uncertainty could be further reduced, if the specific resource probability (as opposed to the probability of all the resources) was provided to the decision maker.

3. Experimental Evaluation

More than ever, decision makers depend on data to make and justify their decisions and the general assumption is that the data is accurate and the DSS will provide a correct answer (Amaro et al. 2005; Covin et al. 2001). As mobile and cloud computing environments increasingly become the norm and system resources become more likely to be distributed (e.g. service oriented architectures, computing grids, and distributed databases) the relevance of the research in resource availability will be progressively more significant. As a result, this study raises the research question: does the availability of decision-related resources map to decision outcomes in a probabilistic manner? Based on the above discussion and the probabilistic model presented in Section 2.3, the following alternate hypothesis is formulated:

> Decision-related resource availability maps to accurate decision outcomes according to the following probability: $P(C) = P(R) + P(A)$, where $P(C)$ is the

probability of correct decision outcome, $P(R)$ is the probability decision-resources are available, and $P(A)$ is the probability that the decision maker chooses the correct answer without the decision support system.

To evaluate the probabilistic model, it is desirable to have a scenario where the DSS provides deterministic (go/no-go, yes/no, or singular answer) guidance in support of a decision opportunity. Further, the decision needs to be based on selection from possible data alternatives, where the data resource may be unavailable. For purposes of evaluating the above hypothesis, a simulation of a stock trading decision was modeled. Stock purchasing was chosen because it is representative of the decision opportunities identified above and easy to understand for a broad range of audiences.

A decision problem was constructed where a stock is purchased from the list of Standard & Poor's 500 stocks (S&P 500). The decision maker has a simple strategy for deciding which stock to purchase. In equity trading, there is a concept that volume precedes price (Fontanills et al. 2001) and this is the purchasing strategy that the decision maker employs. The DSS has the capability to identify from the list of 500 which stock has the highest volume for the time of purchase and this is the correct advice provided. To provide this advice, the DSS requires a resource that specifically identifies the stock with the highest volume for the purchase period. To choose a stock the decision maker requests the highest volume stock for the DSS and always takes the provided advice, if available. If the DSS is unable to provide advice, the decision maker selects a stock from the list of 500 stocks.

A precise and explicit model of the decision problem and simulation was programmed in Matlab. This software provided a robust programming environment where the simulation could be created and evaluated. The resource that provided the high-volume stock selection was coded with 5 hierarchy levels according to Figure 1. Each of these levels was coded with a probability between 0 and 100% that would be determined randomly at run time. The equation shown in (5) was coded to determine the overall resource availability probability. This probability was compared to an "outage" variable whose value was also set randomly. If the outage variable value was lower than the resource availability probability, the resource was considered unavailable. The decision maker was also coded as part of the simulation and always took the advice offered by the DSS. When the advice was not available, the simulated decision maker chose a stock randomly from the list of 500.

A run of the simulation consisted of the generating the availability probabilities, the outage value, and a single decision outcome. For each run, the availability status of the resource and subsequently the DSS advice, was recorded, with the correct stock and the stock selected by the decision maker. Several executions of the simulation were made of varying run sizes from one hundred to one million.

4. Results

The results were collected and analyzed using SPSS. Correct decision outcomes were coded as one and incorrect as zero. The resource availability status was coded similarly: one for available and zero for unavailable. The probability that the resource was available was determined for each run-size and then applied to the probability model to forecast the expected decision outcome accuracy. For example, in the case of the run of 100 decisions, the resource was available only 3% of the time. Applying this value to the probability model shown in Equation (6) leads to an expectation of 3 correct outcomes, given by: (3/100) + (1/500)*100 = 3.2/100. Since the measurement of correct outcomes must be an integer value, the model results were rounded to the whole number: 3 correct and 97 incorrect outcomes in the run-size with 100 runs. This same calculation was performed for each of the run-size sets and used as input to Pearson's Chi-Square Goodness of Fit Test for each. Pearson's Chi-Square Goodness of Fit Test evaluates how close observed values are to those that would be expected from a model (Chernoff et al. 1954). Table 2 shows the results of this test, with the expected column being the calculated values from the availability model.

RUN-SIZE	PERCENT AVAILABLE	OUTCOME	EXPECTED	OBSERVED	RESIDUAL	CHI-SQUARE	ASYMP. SIG.
100	3.00%	*Correct*	3	4	1	.344	.558
		Incorrect	97	96	-1		
1,000	2.80%	*Correct*	30	32	2	.137	.711
		Incorrect	970	968	-2		
10,000	3.29%	*Correct*	349	345	4	.048	.827
		Incorrect	9,651	9,655	-4		
50,000	3.08%	*Correct*	1,639	1,630	9	.051	.821
		Incorrect	48,361	48,370	-9		
100,000	3.09%	*Correct*	3,285	3,254	31	.302	.582
		Incorrect	96,715	96,746	-31		
500,000	3.11%	*Correct*	16,565	16,532	33	.068	.794
		Incorrect	483,435	483,468	-33		
1,000,000	3.12%	*Correct*	33,158	33,103	55	.094	.759
		Incorrect	966,842	966,897	-55		

Table 2. Pearson's Chi-Square goodness of fit results

If the computed Chi-Square value is large (generally greater than 1), then the observed and expected values are not close and the model is a poor fit to the data. As is evident in Table 2, the test statistic values are small for every run-size indicating the model is a good fit. The Chi-Square test was also used to evaluate the hypothesis. In a Chi Square Goodness of Fit Test, a small significance indicates that the observed distribution does *not* conform to the hypothesized distribution (Plackett 1983). In all of the runs the asymptotic significance was above an alpha = .05 level of significance indicating that the distributions are the same; supporting the alternate hypothesis. A second analysis of the expected and observed results' distribution was conducted using a one way ANOVA test. The results of the ANOVA test yielded a between groups sum of squares of .038 with an alpha level = 1. This result is consistent with and supports the goodness of fit test.

5. Conclusion

This study proposes that cloud computing applications would benefit if availability-related context information could be known. The first step in this research was to determine if it is possible to map system availability to system usage/benefit. As this study illustrates, there is a probabilistic relationship between decision support-related computing resource availability and correct decision outcomes when the decision is structured, the data is correct, and the DSS provided guidance is deterministic. When availability is less than guaranteed, the ambiguity in resource availability inserts additional uncertainty in the system usage process. By providing a measure of the likelihood of a correct outcome and details about the availability of individual computing resources, users can make informed choices regarding the potential for support from cloud based computing applications.

While the model demonstrated in this paper provides a tool for quantifying the likelihood of correct outcomes, the real benefit of the model may be realized when availability information is extended to the decision maker. In this sense, the model should be incorporated in client hardware and software that utilize cloud computing services. The model is also dependent on collected availability context data. As such, it should be tied to other context-related technologies such as location. This approach would allow the model to be used for predictive purposes. The research in this paper represents a reasonable first step in addressing availability issues related to cloud computing, but as it is implemented as a simulation it has the limitations of the simulation scenario. A future study is planned to operationalize the model within a mobile client. This future study will incorporate network sensing with GPS for location data and extend the model with a real world cloud computing application.

6. References

1. Ali, A.S., Rana, O., and Walker, D.W. "WS-QoC: Measuring Quality of Service Compliance " International Conference on Service Oriented Computing (ICSOC04), New York, NY, USA, 2004.
2. Amaro, H., Blake, S.M., Morrill, A.C., Cranston, K., Logan, J., Conron, K.J., and Dai, J. "HIV Prevention Community Planning: Challenges and Opportunities for Data-Informed Decision-Making," *AIDS and Behavior* (9:2) 2005, pp 9-27.
3. Begole, J.B., Matsakis, N.E., and Tang, J.C. "Lilsys: Sensing Unavailability," 2004 ACM conference on Computer supported cooperative work, Chicago, Illinois, USA, 2004.
4. Bhagwan, R., Savage, S., and Voelker, G. "Understanding Availability," Peer-to-Peer Systems II (IPTPS 2003), Berkeley, CA, USA, 2003, pp. 256-267.
5. Blake, C., and Rodrigues, R. "High availability, scalable storage, dynamic peer networks: pick two," The 9th conference on Hot Topics in Operating Systems (HOTOS'03), USENIX Association, Lihue, HI, USA, 2003, pp. 1-1.
6. Brown, A., Oppenheimer, D., Keeton, K., Thomas, R., Kubiatowicz, J., and Patterson, D.A. " ISTORE: introspective storage for data-intensive network services," The IEEE Seventh Workshop on Hot Topics in Operating Systems, Rio Rico, AZ, USA, 1999, pp. 32-37.
7. Buyya, R., and Sulistio, A. "Service and Utility Oriented Distributed Computing Systems: Challenges and Opportunities for Modeling and Simulation Communities," in: *41st Annual Simulation Symposium (ANSS-41)*, Ottawa, Canada, 2008, pp. 68-81.
8. Chakraborty, S., Yau, D.K.Y., Lui, J.C.S., and Dong, Y. "On the Effectiveness of Movement Prediction to Reduce Energy Consumption in Wireless Communication," *IEEE Transactions on Mobile Computing* (5:2) 2006, pp 157 - 169.
9. Chernoff, H., and Lehmann, E.L. "The use of maximum likelihood estimates in $\chi 2$ tests for goodness-of-fit," *The Annals of Mathematical Statistics* (25) 1954, pp 579-586.
10. Covin, J.G., Slevin, D.P., and Heeley, M.B. "Strategic decision making in an intuitive vs. technocratic mode: structural and environmental considerations," *Journal of Business Research* (52:1) 2001, pp 51-67.
11. Dai, Y.S., Xie, M., Poh, K.L., and Liu, G.Q. "A study of service reliability and availability for distributed systems," *Reliability Engineering & System Safety* (79:1) 2003, pp 103-112.

12. Danninger, M., Kluge, T., and Stiefelhagen, R. "MyConnector: analysis of context cues to predict human availability for communication," 8th international conference on Multimodal interfaces, Banff, Alberta, Canada, 2006.
13. Fontanills, G.A., and Gentile, T. *The Stock Market Course* John Wiley & Sons, New York, NY, USA, 2001.
14. Henson, V., Ven, A.v.d., Gud, A., and Brown, Z. "Chunkfs: Using divide–and–conquer to improve file system reliability and repair," 2nd Workshop on Hot Topics in System Dependability (HotDep '06), Seattle, WA, USA, 2006.
15. Horvitz, E., Koch, P., Kadie, C.M., and Jacobs, A. "Coordinate: Probabilistic Forecasting of Presence and Availability," The Eighteenth Conference on Uncertainty and Artificial Intelligence, Edmonton, Alberta, Canada, 2002, pp. 224-233.
16. Ibach, P., and Horbank, M. "Highly available location-based services in mobile environments," International Service Availability Symposium, Munich, Germany, 2004.
17. Loyall, J.P., Schantz, R.E., Zinky, J.A., and Bakken, D.E. "Specifying and Measuring Quality of Service in Distributed Object Systems," First International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '98), Kyoto, Japan, 1998.
18. Menasce, D.A. "Composing Web Services: A QoS View," *IEEE Internet Computing* (Vol. 8:06) 2004, pp 88-90.
19. Mikic-Rakic, M., Malek, S., and Medvidovic, N. "Improving Availability in Large, Distributed Component-Based Systems Via Redeployment," in: *Component Deployment*, Springer, Berlin, Germany, 2005, pp. 83-98.
20. Muhlenbrock, M., Brdiczka, O., Snowdon, D., and Meunier, J.L. "Learning to detect user activity and availability from a variety of sensor data," Second IEEE Annual Conference on Pervasive Computing and Communications (PerCom 2004). , Orlando, FL, USA, 2004, pp. 13-22.
21. Peer, J. "Web Service Composition as AI Planning - a Survey," University of St. Gallen.
22. Pistore, M., Barbon, F., Bertoli, P., Shaparau, D., and Traverso, P. *Planning and Monitoring Web Service Composition*, 2004, p. 106.
23. Plackett, R.L. "Karl Pearson and the Chi-Squared Test," *International Statistical Review* (51:1) 1983, pp 59-72.
24. Rahmati, A., and Zhong, L. "Context-for-wireless: context-sensitive energy-efficient wireless data transfer," 5th International Conference on Mobile systems, Applications and Services San Juan, Puerto Rico, 2007, pp. 165 - 178.
25. Reussner, R.H., Schmidt, H.W., and Poernomo, I.H. "Reliability prediction for component-based software architectures," *Journal of Systems and Software* (66:3) 2003, pp 241-252.
26. Roughan, M., Griffin, T., Mao, M., Greenberg, A., and Freeman, B. "Combining routing and traffic data for detection of IP forwarding anomalies," *ACM SIGMETRICS Performance Evaluation Review* (32:1) 2004, pp 416-417.
27. Russell, S., Forgionne, G., and Yoon, V. "Presence and Availability Awareness for Decision Support Systems in Pervasive Computing Environments," *International Journal of Decision Support System Technology* (1:1) 2008.
28. Salas, J., Perez-Sorrosal, F., Martinez, M.P., and Jimenez-Peris, R. "WS-replication: a framework for highly available web services," in: *15th International World Wide Web Conference*, ACM, Edinburgh, Scotland, 2006.
29. Shahram, G., Shyam, K., and Bhaskar, K. "An evaluation of availability latency in carrier-based wehicular ad-hoc networks," in: *Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access*, ACM Press, Chicago, Illinois, USA, 2006.
30. Sung, H., Choi, B., Kim, H., Song, J., Han, S., Ang, C.-W., Cheng, W.-C., and Wong, K.-S. "Dynamic Clustering Model for High Service Availability," in: *Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07)*, IEEE Computer Society, Sedona, AZ, USA, 2007.

31. Thio, N., and Karunasekera, S. "Automatic measurement of a QoS metric for Web service recommendation," 2005 Australian Software Engineering Conference, Brisbane, Australia, 2005, pp. 202-211.
32. Weatherspoon, H., Chun, B.-G., So, C.W., and Kubiatowicz, J. "Long-Term Data Maintenance in Wide-Area Storage Systems: A Quantitative Approach," University of California, Berkely, Electrical Engineering & Computer Sciences Department, Berkely, CA, USA.
33. Xu, J., and Lee, W. "Sustaining Availability of Web Services under Distributed Denial of Service Attacks," *IEEE Transactions on Computers* (52:2) 2003, pp 195-208.
34. Zhoujun, H., Zhigang, H., and Zhenhua, L. "Resource Availability Evaluation in Service Grid Environment," in: *2nd IEEE Asia-Pacific Service Computing Conference (APSCC 2007)*, IEEE Computer Society, Tsukuba Science City, Japan, 2007.