EVALUATING SURVIVABILITY AND COSTS OF THREE VIRTUAL MACHINE BASED SERVER ARCHITECTURES

Meng Yu[1], Alex Hai Wang[2], Wanyu Zang[1] and Peng Liu[2]
[1]*Western Illinois University, IL, Macomb, U.S.A.*
[2]*Pennsylvania State University, PA, University Park, U.S.A.*

Keywords: Security modeling, Survivability, Security architecture, Software security, Data center.

Abstract: Virtual machine based services are becoming predominant in data centers or cloud computing since virtual machines can provide strong isolation and better monitoring for security purposes. While there are many promising security techniques based on virtual machines, it is not clear how significant the difference between various system architectures can be in term of survivability.
In this paper, we analyze the survivability of three virtual machine based architectures — load balancing architecture, isolated service architecture, and BFT architecture. Both the survivability based on the availability and the survivability under sustained attacks for each architecture are analyzed. Furthermore, the costs of each architecture are compared. The results show that even if the same set of commercial off the shell (COTS) software are used, the performance of various service architectures are largely different in surviving attacks. Our results can be used as guidelines in the service architecture design when survivability to attacks is important.

1 INTRODUCTION

Virtual machine technology provides strong isolation and better monitoring capability at the virtual machine monitor level. Once attacks happen, though it is possible, it is hard for the attacker to break into the virtual machine monitor to compromise other virtual machines or avoid being monitored. Therefore, virtual machine technology is widely used in cloud computing and data centers as a preliminary approach in various service architectures.

When using the same set of commercial off the shell (COTS) software and the virtual machine technology, the service architectures can be very different. Accordingly, the security characteristics of each architecture will be different too. In this paper, we use three virtual machine based architectures as examples to evaluate such differences with regard to survivability and costs. While we evaluate three specific architectures in the paper, our techniques used for evaluation are general enough to be applied on other architectures.

There have been many techniqeus for evaluating attacks or defense, such as attack graphs (Sawilla and Ou, 2008), attack tree (Mauw and Oostdijk, 2005), stochastic activity network (Sanders et al., 2001), stochastic petri-Net (Marsan, 1990), reliability block diagram (RBD) (Sahner et al., 1996b), queuing networks and Continuous Time Markov Chain (CTMC) (Tijms, 1994; Sahner et al., 1996a). The aforementioned techniques have been used to evaluate some server architectures, like web service architectures (Gokhale et al., 2006), or data flow software architecture (Padilla et al., 2008) with regard to the reliability, availability, and performance. Models that can be used to evaluate the dependability and security have been summarized in (Nicol et al., 2004). However, survivability of virtual machine based architectures have not been investigated yet, especially the architecture with COTS service components.

The major contribution of the paper includes: 1) evaluating the impact of various architecture designs on both the static and dynamic survivability; 2) studying several survivability related metrics in each architecture; and 3) comparing the costs of these architectures to see how much we need to pay for a specific survivability level. To our best knowledge, our work is the first work to analyze the survivability, and performance for the virtual machine based architectures.

The paper is organized as follows. We review related work in Section 2. In Section 3, we describe three virtual machine based architectures to be evaluated. We analyze the survivability statically in Section 4 and analyze the dynamic behaviors of each ar-

Yu M., Hai Wang A., Zang W. and Liu P. (2010).
EVALUATING SURVIVABILITY AND COSTS OF THREE VIRTUAL MACHINE BASED SERVER ARCHITECTURES.
In *Proceedings of the International Conference on Security and Cryptography*, pages 478-485
DOI: 10.5220/0002994604780485

chitecture in Section 5. In Section 6, we compare the costs of different architectures. We conclude the paper in Section 7.

2 RELATED WORK

Data centers using virtual machine (VM) consolidation are taking over old computer rooms run by individual companies. The reasons include space and energy bills. According to a report from research firm Gartner, millions of virtual machines have been or are being deployed in data centers around the world, and virtualization is becoming a dominant indispensable technology for IT departments.

Due to resource and service consolidation, data centers are becoming the "backbone" of the infrastructure for IT operations in companies, governments, and military. Accordingly, two top requirements for modern data centers are *business continuity* and *information security*. Although these two requirements clearly show the importance of data center protection, from the security viewpoint, consolidating services and resources does not automatically consolidate the corresponding security mechanisms. Without security consolidation, the cost of protection can be much higher that it should be, and more importantly, blindly reusing the separate security equipment and tools associated with the services/resources being consolidated could even "create" new security holes. Unfortunately, current security consolidation has been lagging behind service/resource consolidation in the data center industry.

Replicated systems can provide better fault tolerance, or intrusion tolerance when attacks are unknown attacks that can be modeled as Byzantine faults (Castro, 2001; Schneider, 1990; Bernstein et al., 1987; Seguin et al., 1979; Jajodia and Mutchler, 1990; Alvisi et al., 2001; Malkhi and Reiter, 1998; Castro, 2001). While the performance was a concern of these approaches, recent advance of research leads to more pratical performance, e.g., in Zyzzyva (Kotla et al., 2007). the peak throughput achieved by Zyzzyva is within 35% of that of an unreplicated server that simply replies to client request over an authenticated channel. With replications, compromised nodes can be removed through a voting mechnism. Therefore, the attacker have to compromise replicas more than a thredhold number, usually more than $\lfloor \frac{n-1}{3} \rfloor$ replicas, of a replicated system with n replicas to disable the system. However, an attacker can easily compromise all replicas through a common vulnerability shared by all replicas, especially when all replicas are running homoenous environments, e.g., the same operating systems or web servers.

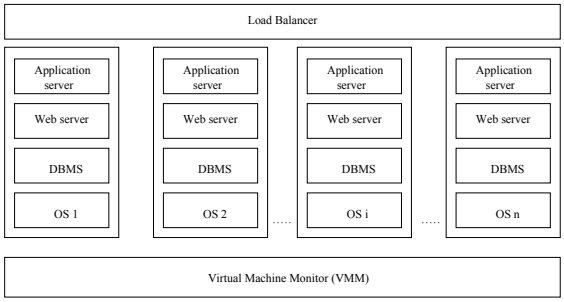

Figure 1: Load Balanced Server Architecture (LBSA).

The basic idea of diverse replication using Virtual Machines has been described in (Chun et al., 2008). The combination of diversification and replication is capable of defeating unknown attacks with practical costs. Proper configured diverse replication will be immune to attacks based on single vulnerability and can remove compromised node even if less then $\lfloor \frac{n-1}{3} \rfloor$ nodes are compromised. However, the method has not been implemented or evaluated with respect to the effectiveness or performance.

Current evaluation techniqeus for attacks or defense, such as aforementioned attack graphs (Sawilla and Ou, 2008), attack tree (Mauw and Oostdijk, 2005), queuing networks and Continuous Time Markov Chain (CTMC) (Tijms, 1994; Sahner et al., 1996a), and so on, have ben used to evaluate some server architectures. Models that can be used to evaluate dependability and security have been summarized in (Nicol et al., 2004). However, survivability of virtual machine based architectures have not been investigated yet, especially the architecture with COTS service components.

3 THREE VIRTUAL MACHINE BASED ARCHITECTURES

In this section, we describe three virtual machine based service architectures - Load Balanced Server Architecture (LBSA), Isolated Component based Server Architecture (ICSA), and Byzantine Fault Tolerant (Castro and Liskov, 1999) Server Architecture (BFTSA).

The LBSA architecture is shown in Figure 1. This is the most popular architecture to provide high availability services. In this architecture, a set of services are installed on each virtual machine of a virtual machine cluster. The users' requests are directed to alive virtual machines with load balancing. In a secure configuration, the virtual machine environments are diversified, which guarantees that a single instance of

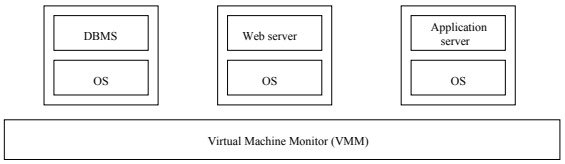

Figure 2: Isolated Component based Server Architecture (ICSA).

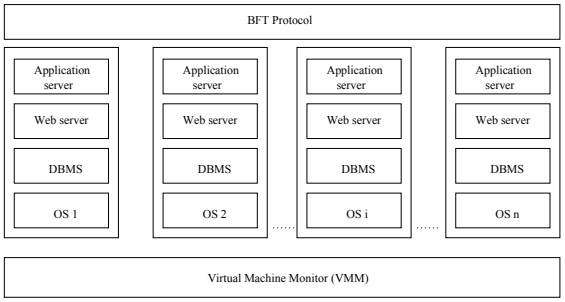

Figure 3: Byzantine Fault Tolerant Server Architecture (BFTSA).

attack cannot compromise all virtual machines in the cluster. Moreover, even if one virtual machine is compromised, the load balancer can detect the inconsistency, give up or reboot the compromised virtual machine with a different diversified configuration. Since virtual machines provide good isolation, we assume that a single instance of attack cannot compromise all virtual machines of the cluster.

The ICSA architecture is shown in Figure 2. In this architecture, each service of a service architecture is installed on a virtual machine. The number of virtual machines is determined by the number of services. In this figure, we have three services thus we have three virtual machines. Once a service is compromised, it is confined in its virtual machine and can be recovered through a reboot of the service. The virtual machine isolation mechanism reduces the number of reboots in the situation of attacks.

The BFT architecture is shown in Figure 3. A Byzantine Fault Tolerant (BFT) algorithm (Castro and Liskov, 1999) is required for this architecture to handle users' requests. Once a user's request is received, the BFT algorithm goes through three phases: pre-prepare, prepare, and commit to remove replies from nodes with arbitrary faults. The BFT algorithm can tolerate $\lfloor \frac{n-1}{3} \rfloor$ compromised servers.

Note that in all architectures, virtual machines should be configured with different diversification, which ensures that no more than one virtual machine can be compromised by one instance of attack.

4 EVALUATING SURVIVABILITY BASED ON STATIC ANALYSIS

4.1 Basic Assumptions

In this paper, we assume that the defense mechanisms used in the virtual machines are based on diversification (Chun et al., 2008). Thus, a single instance of attack cannot compromise more than one virtual machine in any architecture no matter what technique the attacker uses, such as buffer overflow, stack overflow, format string, and etc. However, each instance of attack can have a positive probability to compromise a virtual machine, thus any component inside the virtual machine. Furthermore, attacks can include finite steps on the target, so virtual machines in an architecture may be compromised one by one until all are compromised but not in a single step.

We also assume that the defense mechanisms have an automated recovery procedure included. Once a virtual machine is detected as compromised, the recovery mechanism can recover the virtual machine with a different diversification through reboot or micro-reboot.

We define the survivability of a system under specific condition as follows.

Definition 1 (Survivability). *We define the survivability of a system under a specific condition as the probability that the system can meet the following two requirements.*

Availability. *The system can provide replies to all requests, and*

Service Integrity. *The replies meet the functional specification of the system and all service components of the system are functional.*

For example, a web service architecture needs three level of servers: web server, application server, and DBMS server. When a system is under attacks, as long as the system can provide replies (availability) and the replies are generated by the three levels of servers correctly (service integrity), we say that the system can survive the attacks. Otherwise, the whole system is compromised.

4.2 Analytical Results

The LBSA architecture is good at providing availability. A system under such architecture will provide services unless all virtual machines are crashed. However, the service integrity is another story. If any virtual machine is compromised by the attacker, the service integrity is compromised (note that we are

discussing attacks instead of "fail-stop" failures), because the load balancer cannot tell a virtual machine demonstrating arbitrary faults from other normal virtual machines. Therefore, a single compromised virtual machine leads to a compromised cluster.

Under the ICSA architecture, once a virtual machine is compromised, the installed service will be gone. Thus, both the availability and service integrity are compromised. As the results, the whole system is compromised.

For both LBSA and ICSA architectures, assume that a system has n virtual machines. Note that for ICSA, n is usually the same as the number of services unless multiple services are installed on the same virtual machine. If the virtual machines are not diversified, the probability breaking into the system through a single vulnerability, denoted by P_b, will be the same as the probability breaking into a single replica, denoted by P_r, because the vulnerability is shared by all replicas. Thus, the survivability of the system $P_s = 1 - P_b = 1 - P_r$. With disjoint diversification in a space S, where no attack can compromise more than one variation at a time and S contains all possible variations of the diversification, P_s can be calculated by Equation 1.

$$\begin{aligned} P_s = & \quad 1 - \sum_{i=1}^{m} \binom{m}{i} (P_r)^i (1-P_r)^{m-i} \\ = & \quad 1 - \sum_{i=1}^{m} \binom{m}{i} (\frac{n}{|S|})^i (1-\frac{n}{|S|})^{m-i} \end{aligned} \quad (1)$$

where $n \leq |S|$ (n nodes cannot have more than $|S|$ variations), and $m > 1$ (we need to compromise at least 1 replicas) is the total number of intrusion attempts.

With k independent diversification approaches, e.g., diversified API, memory randomization, etc., the diversification space will be $S_1, S_2, \ldots, S_k$ respectively. Thus, for a sequence of m intrusion attempts,

$$P_s = 1 - \sum_{i=1}^{m} \binom{m}{i} (\frac{n}{\prod_{j=1}^{k} |S_j|})^i (1 - \frac{n}{\prod_{j=1}^{k} |S_j|})^{m-i} \quad (2)$$

Under the BFTSA architecture, we assume a BFT (Castro, 2001; Kotla et al., 2007) server group with n replicated virtual machines. Because BFT protocol can tolerate $\lfloor \frac{n-1}{3} \rfloor$ compromised virtual machines, a system will be compromised if more than $\lfloor \frac{n-1}{3} \rfloor$ virtual machines in the cluster are compromised. Therefore, the survivability can be calculated by Equation 3.

$$\begin{aligned} P_s = & \quad 1 - \sum_{i=\lfloor \frac{n-1}{3} \rfloor + 1}^{m} \binom{m}{i} (P_r)^i (1-P_r)^{m-i} \\ = & \quad 1 - \sum_{i=\lfloor \frac{n-1}{3} \rfloor + 1}^{m} \binom{m}{i} (\frac{n}{|S|})^i (1-\frac{n}{|S|})^{m-i} \end{aligned} \quad (3)$$

where $n \leq |S|$ (n nodes cannot have more than $|S|$ variations), and $m > \lfloor \frac{n-1}{3} \rfloor$ (we need to compromise at least $\lfloor \frac{n-1}{3} \rfloor + 1$ replicas) is the total number of intrusion attempts.

Similarly, with k independent diversification approaches, for a sequence of m intrusion attempts,

$$P_s = 1 - \sum_{i=\lfloor \frac{n-1}{3} \rfloor + 1}^{m} \binom{m}{i} (\frac{n}{\prod_{j=1}^{k} |S_j|})^i (1 - \frac{n}{\prod_{j=1}^{k} |S_j|})^{m-i} \quad (4)$$

where $n \leq \prod_{j=1}^{k} |S_j|$, and $m > \lfloor \frac{n-1}{3} \rfloor$.

In the above discussion, we assumed the ideal situation where 1) the success of one intrusion attempt is independent of other attempts; 2) any combination of diversification techniques is valid, and 3) BFT replication is used. Note that randomization techniques make the probability breaking into a system to be independent between attacks in a sequence. However, the following challenges may significantly change the form of Equation 4. a) Due to the resource limit, full BFT replication may be not feasible in a heavily loaded system, where the threshold, $\lfloor \frac{n-1}{3} \rfloor$, to break into the system will be different. b) The probability of success will become accumulative, conditional, or others, in a sequence of attacks if the diversification is not randomized for each attempt of attack. c) The characteristics of the attacks, e.g., steps included in the attack, may change the probability of success between attempts to be conditional. d) A random combination of different diversification techniques does not automatically lead to a valid and independent defenses, which will change $\prod_{j=1}^{k} |S_j|$ in Equation 4. For an example, a simple combination of two different stack randomization techniques may lead to incorrect stack frames. Thus, there will be many combination in $\prod_{j=1}^{k} S_j$ not valid for deployment. As the result, our defense space will actually be smaller. Furthermore, when the space of a diversification technique is very small, multiple replicas may need to share the same variation, which further weakens the capability of defense.

When all diversification are valid, the comparison of survivability is shown in Figure 4. In the figure, BFTSA shows better survivability and the survivability gets better when increasing the total number of nodes.

An interesting thing is that the survivability of BFTSA is not monotonic, which is due to the characteristics of BFT protocol (Castro and Liskov, 1999). The BFT protocol can tolerate $\lfloor \frac{n-1}{3} \rfloor$ compromised virtual machines in a cluster with n virtual machines. In the example, when we have 7, 8, or 9 virtual

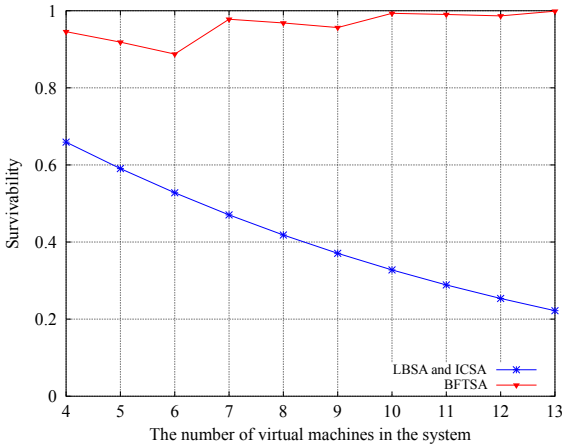

Figure 4: Static analysis of survivability with $m = 5$ and $|S| = 50$.

machines in the cluster, the cluster can tolerate the same number, $\lfloor \frac{n-1}{3} \rfloor = 2$, of compromised virtual machines. However, increased number of virtual machines in the cluster also increases the probability of successful attack in one attempt. Therefore, if the increasing number of virtual machines does not increase $\lfloor \frac{n-1}{3} \rfloor$, the survivability decreases.

5 EVALUATING SURVIVABILITY UNDER SUSTAINED ATTACKS

In this section, we analyze the behaviors of each architecture under sustained attacks.

5.1 State Transition Diagrams

The state transition diagrams of LBSA and ICSA are shown in Figure 5. In the figure, the attacker compromises the virtual machines one by one with rate λ. The system can recover virtual machines or services in the virtual machine (through reboot, micro-reboot, or other approaches) with rate μ. A state G_i, where $0 \leq i \leq n$, indicates that the system has i virtual machines not compromised and $n-i$ virtual machines compromised. In LBSA or ICSA, once a virtual machine is compromised, the service integrity is gone. Thus, the only normal state is G_n where no virtual machine is compromised.

The state transition diagrams of BFTSA is shown in Figure 6. In the figure, the state and state transition rates are the same as the one of LBSA and ICSA. However, since BFTSA has a BFT protocol to eliminate compromised nodes of the cluster, it can tolerate $\lfloor \frac{n-1}{3} \rfloor$ compromised virtual machines. Therefore, the normal states are $\{G_n, G_{n-1}, \ldots, G_{n-\lfloor \frac{n-1}{3} \rfloor}\}$.

5.2 The Continuous Time Markov Chain (CTMC) Model

We assume that λ and μ in Figure 5 and Figure 6 meet the Poisson distribution. Based on the assumptions about parameters and the above discussion, the state transition of our model becomes a finite states Continuous-Time Markov Chain (CTMC) (Tijms, 1994; Sahner et al., 1996a) that can be characterized by a *state transition matrix* $\mathbb{Q} = (q_{i,j})$ and initial state probability vector $\underline{\pi}(0)$, where $q_{i,j}$ is the transition rate from i to j and $q_{i,i} = -\sum_{j \neq i} q_{i,j}$.

The state transition matrix of Figure 5 and Figure 6 is as follows.

$$\mathbb{Q} = \begin{array}{c} \\ G_0 \\ G_1 \\ \vdots \\ G_n \end{array} \begin{array}{c} \begin{array}{cccccc} G_0 & G_1 & G_2 & \ldots & G_{n-1} & G_n \end{array} \\ \begin{pmatrix} -\mu & \mu & 0 & \ldots & 0 & 0 \\ \lambda & -\lambda-\mu & \mu & \ldots & 0 & 0 \\ \vdots & & & \ldots & & \vdots \\ 0 & 0 & 0 & \ldots & \lambda & -\lambda \end{pmatrix} \end{array} \quad (5)$$

The initial state of system is at G_n, Thus

$$\underline{\pi}(0) = \underbrace{(0, 0, \ldots, 1)}_{n} \quad (6)$$

With the CTMC model, both steady state and transient state can be calculated. The steady state of a system is the state that all features of the system do not change any more after running a long period of time. It may not exist at all for a specific CTMC. Fortunately, most real systems do have their steady states. Once a n by n generator matrix $\mathbb{Q}$ is given, the steady-state probability vector $\underline{\pi}$ is determined by Equation 7.

$$\underline{\pi}\mathbb{Q} = \underline{0}, \sum_{1 \leq i \leq n} \pi_i = 1. \quad (7)$$

The comparison of steady states of different architectures is shown in Figure 7. In the figure, X-axes shows the recovery rate μ and the compromise rate λ has a fixed value 1. While μ increases from 1 to 10, all architectures demonstrate increasing survivability. However, BFTSA shows much better survivability and it is very sensitive to the increment of μ. The survivability of BFTSA reaches values close to 1 much earlier than LBSA and ICSA. Furthermore, the survivability of BFTSA is very close to 1 after μ is greater than 2.

The impact of attack rate λ is shown in Figure 8. In the figure, we keep the recovery rate μ at the same value 5. When we increase the attack rate λ from 1 to 20, the survivability of all architectures decrease. The BFTSA architecture demonstrates better responses to the increasing attack rates.

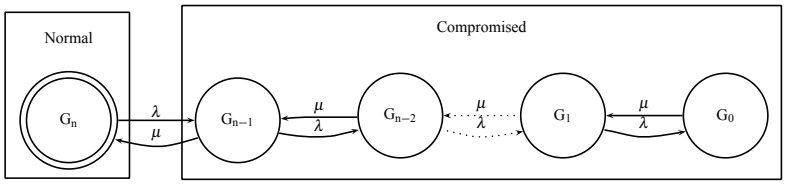

Figure 5: The state transition diagram of LBSA and ICSA.

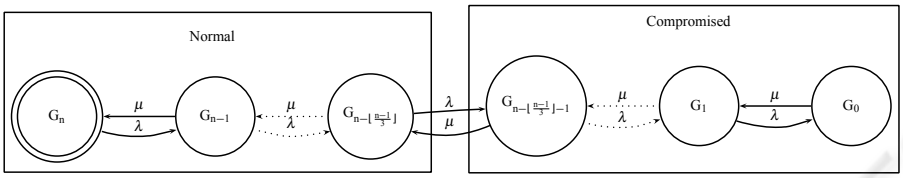

Figure 6: The state transition diagram of BFTSA.

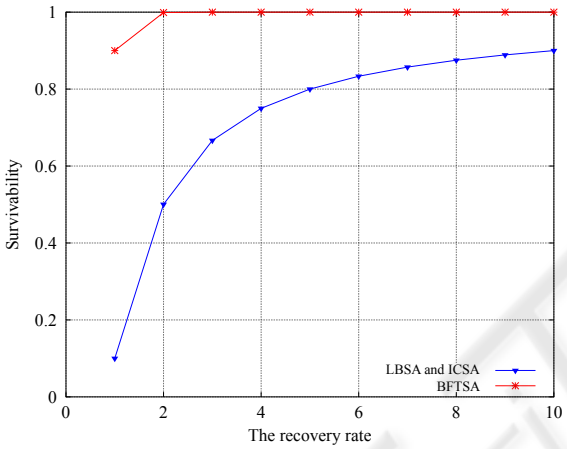

Figure 7: Comparison of steady state survivability while $\lambda = 1$ and $n = 10$.

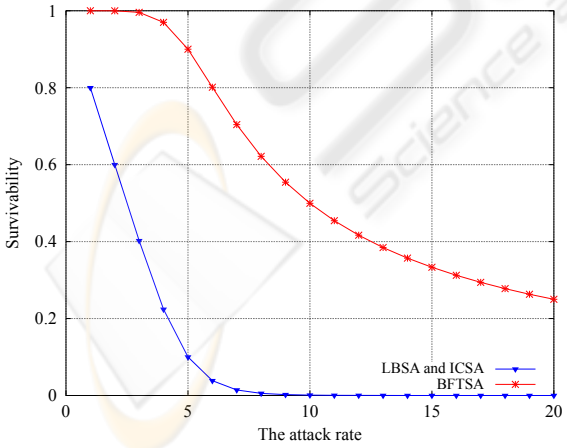

Figure 8: Comparison of steady state survivability while $\mu = 5$ and $n = 10$.

The CTMC model can also tell the transient behaviors of a system. A transient state of a system is the state of the system at a specific moment. The evaluation of transient states shows how quickly a system goes to its steady state, and how much time is spent on each state. A system may satisfy us with its steady states but disappoint us with its transient behaviors, e.g. taking too long to enter the steady state. Transient behaviors also tell us what may happen if a system suffers a short term of high attacking rate.

Given a generator matrix $\mathbb{Q}$ and initial state probability vector $\underline{\pi}(0)$, transit state probability $\underline{\pi}(t)$ at time t is determined by Equation 8.

$$\frac{d}{dt}\underline{\pi}(t) = \underline{\pi}(t)\mathbb{Q} \tag{8}$$

In Figure 9, we show the transient survivability of LBSA and ICSA. In the figure, when time increases from 0 to 8, the survivability decreases fastly to around 0.75. In other words, we can see that the system enters the steady state very quick.

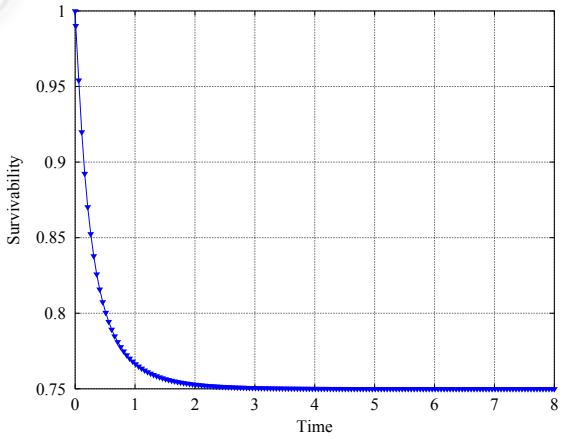

Figure 9: The transient behaviors of LBSA and ICSA while $\lambda = 1$, $\mu = 4$, and $n = 10$.

The transient behaviors of BFTSA is shown in Figure 10. According to the figure, while the steady

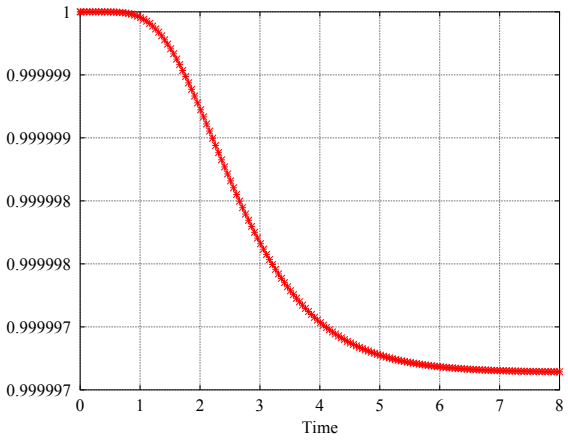

Figure 10: The transient behaviors of BFTSA while $\lambda = 1$, $\mu = 4$, and $n = 10$. Y-axes is survivability.

state survivability is higher than LBSA and ICSA, the system enters the steady state slower.

Given the state transition matrix, when needed, the cumulative time $\underline{l}(t)$ spent on each state at time t is given by Equation 9.

$$\frac{d}{dt}\underline{l}(t) = \underline{l}(t)\mathbb{Q} + \underline{\pi}(0) \tag{9}$$

An example of accumulative time spent on the "normal" state of the LBSA architecture is shown in Figure 11.

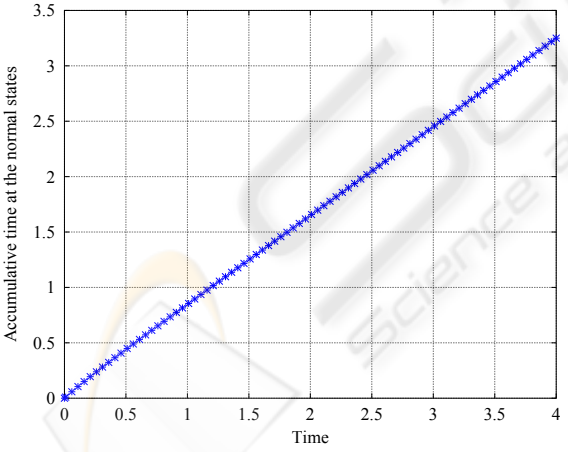

Figure 11: The accumulative time spent on the normal state of LBSA while $\lambda = 1$, $\mu = 5$, and $n = 10$.

6 EVALUATING THE COSTS

In this section, we discuss the costs of each architecture considering the processing costs, memory costs, and communication costs.

Table 1: Evaluation metrics comparison of three architectures.

Metrics	LBSA	ICSA	BFTSA
Processing costs	n	1	n
Memory costs	n	1	n
Communication costs	$O(1)$	$O(1)$	$O(n^{\frac{3}{2}})$
Intrusion tolerance	0	0	$\lfloor\frac{n-1}{3}\rfloor$
Fail-safe fault tolerance	$n-1$	0	$\lfloor\frac{n-1}{3}\rfloor$

Both LBSA and BFTSA need diversified replications. When we replicate the services in virtual machines, the processing costs and memory needed by replication will be n times more than the needs of ICSA.

When we consider the communication costs, for each request, ICSA does not duplicate and forward the request. Similarly, while the load balancer in LBSA forwards the request to lightly loaded virtual machine, it does not duplicate the request to other virtual machines either. However, BFTSA requires a more complex communication protocol. The BFT protocol (Castro and Liskov, 1999) goes three phases and in each phase, one or all nodes in the cluster needs to broadcast to others. The communication complexity of BFT is $O(n^{\frac{3}{2}})$ (Castro and Liskov, 1999).

We summarize the costs and the capability of intrusion tolerance or fault tolerance in Table 1. In the table, intrusion tolerance indicates how many compromised nodes can be tolerated by a specific architecture. The fail-safe fault tolerance indicates how many fail-safe nodes can be tolerated, where fail-safe nodes do not demonstrate arbitrary behaviors as a compromised node does under attacks.

7 CONCLUSIONS

In this paper, we compared the survivability and costs of three virtual machine based architectures. Our studies show that even with the same COTS software, a different architecture can have significant impact on the survivability of the whole system. According to our analytical results, replicated architecture with BFT protocol and diversification is much better than simple replication and isolation with regard to survivability. However, the costs of BFT protocol is high. The analytical methods described in this paper, such as the static analysis and the dynamic CTMC based analysis, can be used to analyze the survivability of other architectures. The results of this paper can also be used as guidelines in architecture design when survivability is crucial to the system.

ACKNOWLEDGEMENTS

We thank Dr. Yan Yang for the discussions of some contents of the paper. We also thank all anonymous reviews comments which helped us to greatly improve the quality of the paper. Meng Yu was supported by NSF grant CNS-0905153. Peng Liu was supported by NSF CNS-0905131, AFOSR FA9550-07-1-0527 (MURI), and ARO MURI: Computer-aided Human Centric Cyber Situation Awareness.

REFERENCES

Alvisi, L., Malkhi, D., Pierce, E., and Reiter, M. K. (2001). Fault detection for byzantine quorum systems. *IEEE Transactions on Parallel and Distributed Systems*, 12(9):996–1007.

Bernstein, P. A., Hadzilacos, V., and Goodman, N. (1987). *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, Reading, MA.

Castro, M. (2001). *Practical Byzantine Fault Tolerance*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. Also as Technical Report MIT/LCS/TR-817.

Castro, M. and Liskov, B. (1999). Practical byzantine fault tolerance. In *The Third Symposium on Operating Systems Design and Implementation (OSDI '99)*, pages 173–186, New Orleans, USA.

Chun, B.-G., Maniatis, P., and Shenker, S. (2008). Diverse replication for single-machine byzantine-fault tolerance. In *ATC'08: USENIX 2008 Annual Technical Conference on Annual Technical Conference*, pages 287–292, Berkeley, CA, USA. USENIX Association.

Gokhale, S. S., Vandal, P. J., and Lu, J. (2006). Performance and reliability analysis ofweb server software architectures. In *PRDC '06: Proceedings of the 12th Pacific Rim International Symposium on Dependable Computing*, pages 351–358, Washington, DC, USA. IEEE Computer Society.

Jajodia, S. and Mutchler, D. (1990). Dynamic voting algorithms for maintaining the consistency of a replicated database. *ACM Trans. Database Syst.*, 15(2):230–280.

Kotla, R., Alvisi, L., Dahlin, M., Clement, A., and Wong, E. (2007). Zyzzyva: speculative byzantine fault tolerance. *SIGOPS Oper. Syst. Rev.*, 41(6):45–58.

Malkhi, D. and Reiter, M. (1998). Byzantine quorum system. *Distributed Computing*, 11(4):203–213.

Marsan, M. A. (1990). *Stochastic Petri nets: an elementary introduction*, pages 1–29. Springer-Verlag New York, Inc., New York, NY, USA.

Mauw, S. and Oostdijk, M. (2005). Foundations of attack trees. In *International Conference on Information Security and Cryptology ICISC 2005. LNCS 3935*, pages 186–198. Springer.

Nicol, D. M., Sanders, W. H., and Trivedi, K. S. (2004). Model-based evaluation: From dependability to security. *IEEE Transactions on Dependable and Secure Computing*, 1(1):48–65.

Padilla, G., Gao, T., Yen, I.-L., Bastani, F., and de Oca, C. M. (2008). An early reliability assessment model for data-flow software architectures. *Mexican International Conference on Computer Science*, 0:9–19.

Sahner, R. A., Trivedi, K. S., and Puliafito, A. (1996a). *Performance and Reliability Analysis of Computer Systems*. Kluwer Academic Publishers, Norwell, Massachusetts, USA.

Sahner, R. A., Trivedi, K. S., and Puliafito, A. (1996b). *Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package*. Kluwer Academic Publishers, Norwell, MA, USA.

Sanders, W. H., S, W. H., and Meyer, J. F. (2001). Stochastic activity networks: Formal definitions and concepts.

Sawilla, R. E. and Ou, X. (2008). Identifying critical attack assets in dependency attack graphs. In *ESORICS '08: Proceedings of the 13th European Symposium on Research in Computer Security*, pages 18–34, Berlin, Heidelberg. Springer-Verlag.

Schneider, F. B. (1990). Implementing fault tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys*, 22(4).

Seguin, J., Sergeant, G., and Wilms, P. (1979). A majority consensus algorithm for the consistency of duplicated and distributed information. In *IEEE International Conference on Distributed Computing Systems*, pages 617–624, New York.

Tijms, H. C. (1994). *Stochastic Models*. Wiley series in probability and mathematical statistics. John Wiley & Son, New York, NY, USA.