Large System Decomposition and Simulation Methodology Using Axiomatic Analysis[†]

L. Spenner[‡], P. Krier, M. Thornton, S. Nair, S. Szygenda, and T. Manikas
High Assurance Computing and Networking Laboratories
Department of Computer Science and Engineering, Southern Methodist University, Dallas, TX, USA
{lspenner, pkrier, mitch, nair, szygenda, manikas}@lyle.smu.edu

***Abstract*—Large, established systems are essentially impossible to replace and must be improved incrementally to incorporate disaster tolerance. The question of which components need to be upgraded or have added redundancy requires the system to be wholly analyzed or simulated. Unfortunately, extremely large systems are not amenable to comprehensive simulation studies due to the large computational complexity requirements. This research presents a new method that results in a dramatic decrease in simulation time from that required by an entire system simulation to the sum of the total time of simulations for decomposed subsystems. Axiomatic analysis principles of information and independence are used as part of the methodologies to determine the best decomposition boundaries between subsystems. This allows the smaller subsystems to have maximum independence so that the entire system response can be reconstructed from a linear combination of the individual subsystem simulation responses.**

***Index Terms*—Axiomatic Analysis, Disaster Tolerance, Large System Decomposition Methodology.**

I. Introduction

LARGE systems are challenging to model effectively due to their unwieldy size and complexity. Simulating a large system can require significant time and considerable computing resources to be successful. A method for decomposing the system into smaller, more computationally practical subsystems is developed to alleviate this difficulty. The ability to simulate subsystems of the larger system and then combine the results to achieve overall system simulation response enables the assessment of the incorporation of redundancy or other modifications in key system components to be assessed for the system as a whole before they are deployed. We are particularly motivated to explore and develop this method for the purposes of increasing disaster tolerance through the modification of existing large systems such as the communications infrastructure. This approach, termed as axiomatic analysis, is described and experimental results are provided that demonstrate the effectiveness of the methodology.

II. Background

A. Disaster Tolerance

Disaster Tolerance is the ability of a system to continue providing services in the presence of a disaster. A disaster is generally defined as an unexpected catastrophic event that causes significant damage or failure. A disaster is considered to differ from a fault in that a fault occurs when a system experiences a single point of failure, while a disaster is predicated on multiple points of failure within a small time frame or a single point of failure that cascades into multiple system wide failures. The models used to simulate and validate disaster tolerance system characteristics must incorporate this difference in disasters as opposed to a classical fault model. Additionally, fault tolerance is generally concerned with the probability of a fault occurring without concern to an actor, an internal user, or an external agent, consciously exploiting the vulnerability. Disaster tolerance must consider the possibility of an intelligent agent intentionally injecting a threat event in a non-random manner as well as a natural act such as a hurricane or flood.

The disaster tolerance of a system is also differentiated from the concept of the disaster recovery of a system. Disaster tolerance is the ability of the system to continue operation, either fully or partially, without interruption or failure during a disaster situation. Disaster recovery is the ability of a system to restart operations after the disaster has interrupted services [1]. A disaster tolerant system will continue to operate with some degree of normality in the presence of a disaster. By analogy to classical fault tolerance, a redundant router might go down causing the fail-over router to experience a significant increase in traffic, but the system is still able to maintain a level of performance. A fault recovery system might not be able to respond at all for a specified period of time, but will recover without user intervention.

This research is focused on disaster tolerance with respect to critical infrastructure and similar large systems. These systems tend to be complex due to their interconnectedness with other large systems, all of which have been developed over time without the benefit of a top-down design approach. This lack of an overall comprehensive design approach lends

[†] Supported by the Office of Naval Research, Project N000140910784
[‡] Supported by the DoD Information Assurance Scholarship Program

itself to greater potential for overwhelming failure in the face of a disaster. The intricate, non-obvious subsystem interdependencies of large systems makes the analysis and simulation of these systems very difficult. As an example, the U.S. electric power grid is a large system that evolved over time and was not conceived of as a single entity prior to implementation. This paradigm of an evolving system prevents the use of many techniques used in classical fault tolerance where analysis and subsystem interdependencies can be identified prior to implementation. This is a chief reason that disaster models are necessarily different from classical fault models and indeed are a superset of fault models.

The fact that disaster models are different from fault models and that the systems of interest are extremely large entities that have evolved over time motivated the development of the axiomatic approach for system simulation. In order to enhance a large system for the purpose of increasing disaster tolerance, it is required that a methodology to simulate such systems both in normal operating mode and in the presence of disaster events is formulated.

B. Axiomatic Analysis

In order to model complex systems, we have developed the idea of Axiomatic Analysis (AA), which was originally motivated by the theories of Axiomatic Design (AD) [2]. AA leverages independence and information minimization design axioms to decompose a complex system into relatively independent subcomponents. The identified axioms are:

Independence: Decomposed subsystems must be determined such that their overall dependencies among one another are minimized.

Information Minimization: Decomposed subsystems should be as simple as possible. Information flow across subsystem boundaries should be minimized.

These axioms are the guiding principles we use to decompose a complex system that cannot be modeled in its entirety into more computationally manageable subsystems. Since these subsystems are relatively independent, the linear combination of the subsystem simulation responses will provide the behavior of the entire system. The first step in AA is to evaluate the entire system to determine boundaries between independent subsystems. Once these subsystems are chosen, each is modeled independently. In the final step, the subsystem simulation responses are combined to estimate total system response [3][7].

In order to implement the above axioms for decomposition, an appropriate system metric must be identified. As an example, data bandwidth might be appropriate for a large data communications network such as the Internet. The amount of data being transferred among various Internet system components can be used to measure independence as well as information content due to the well known channel capacity relationship between bandwidth and information content. Alternatively, for a large power distribution network, the appropriate metric may be the real power transfer (as opposed to apparent power) in units of power (Watts) over various interconnections.

III. Large System AA Procedure

Once the appropriate system metrics are identified, those parameters may be measured and the system may be envisioned as a large interconnected graph adjacency matrix with edge weights corresponding to the measurements. Clearly, for our large systems of interest, it will not be plausible to build a complete model with all possible measurements. However, this approach will allow for a tradeoff to occur in terms of number of measurements versus model accuracy. Furthermore, the resulting adjacency matrix is generally quite sparse allowing for efficient data structures to be employed for its representation [4][9].

Using the guiding principles of axiomatic analysis, a matrix is formed that relates the chosen measured metrics among the system components. This matrix can be considered to be a weighted adjacency matrix for a graph where vertices represent system components and edges are weighted by the identified metric. The system matrix is then decomposed allowing for the identification of relatively independent subsystems with minimal information transfer among the subsystems [5][6]. The procedure employed for AA is:

1. Permute system matrix to transform it to a representation that is close to lower triangular
2. Decompose permuted system matrix to identify subsystems with the most independence
3. Simulate decomposed subsystems independently
4. Combine independent subsystem simulation responses to estimate overall large system simulation response

A. Permute System Matrix

Methods for permuting matrices have been well defined in previous work and those approaches are used here. The system matrix is permuted to attempt to transform it into a lower triangular matrix to be used for decomposition. Lower triangular forms are desirable since they automatically expose subsystem independence. In a perfectly independent subsystem, the permuted adjacency matrix would take on the form of the identity matrix indicating all components are totally independent. Triangular forms show the relative independence of large system components. For large systems, the matrix will become too large to represent explicitly and will be sparse in that it will not be possible to obtain measurements of all metrics. Although it is not possible to always form a completely lower triangular matrix, transforming the system matrices into a form that is close to triangular is beneficial for exposing system component independencies.

B. System Matrix Decomposition

The permuted matrix from the previous step is analyzed for decomposition. Blocks are found and used to determine the subsystem boundaries and included components. A tradeoff is performed at this step in that fewer yet relatively large decomposed subsystems yield more accurate overall results but incur increased simulation runtime. Finer grained decompositions are individually faster to simulate and require fewer computational resources, but overall system response may not be as accurate due to interdependencies among the subsystems that are ignored. Several methods of decomposition were explored and those that performed best are explained in detail in a later section of this paper.

C. *Subsystem Simulation*

Subsystems identified in the previous step are simulated. These independent simulations may be performed sequentially or in parallel. This allows for a significant decrease in runtime as compared to simulating the entire system at one time. Indeed, it may not even be possible to simulate the entire system. Furthermore, the particular simulation engine employed can be chosen based on the type of system being analyzed. From this perspective, the AA approach is more of a large system simulation framework rather than a particular simulation technique. For the example systems described here, we used the OPNET commercial simulation tool [10], but any appropriate simulation software could have been used.

D. *Total System Reconstruction*

In a perfectly independent system decomposition, the principle of superposition could be employed allowing the total system response to be generated as the arithmetic sum of the subsystem simulation responses. Since there will be some degree of subsystem interdependence, a weighted sum of subsystem responses or other adaptive approaches can be employed in this step allowing for the entire system response to be realized as a linear combination of decomposed subsystem responses. The linear combination is justified through the use of the independence axiom. In the future we intend to explore nonlinear decomposition and reconstruction methods that may yield increased simulation efficiency.

IV. Validation of AA Procedure

In order to evaluate the effectiveness of the AA approach for large system simulation, we chose an example system that was small enough such that the entire system could be simulated and then applied the AA procedure. By comparing the overall required CPU runtimes and the accuracy of the total system simulation response and that obtained from the AA approach, we can evaluate the effectiveness of the approach.

A data communications network was selected as the candidate system to be used for validation of the AA approach. This network system contains 25 servers and more than 500 terminals. The connections for the network are handled by at least 5 routers and 23 switches. The network is spread geographically across 4 buildings. This network is simulated using the OPNET tool and bandwidth data is collected. Furthermore, we also had measured bandwidth information for this experiment that allowed us to calibrate our total system simulation. Figure 4.1 shows the example system before any of the decomposition methods have been applied.

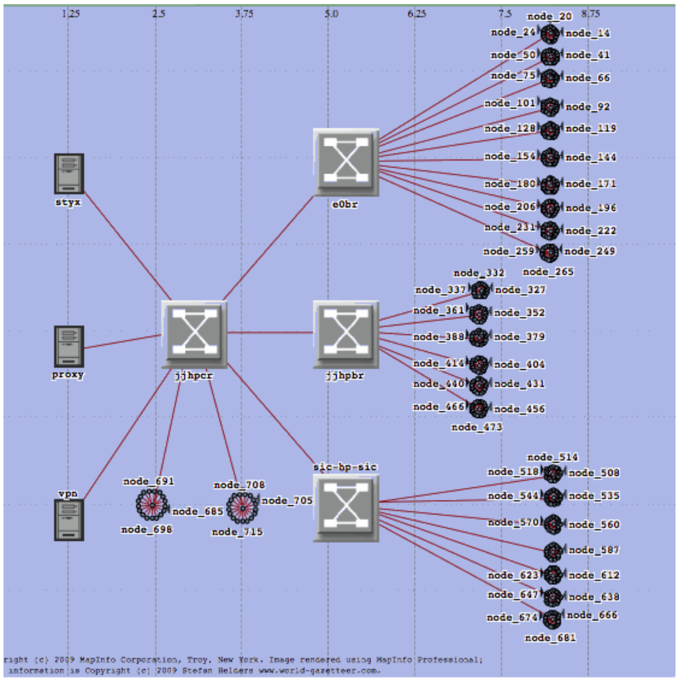

Figure 4.1: Example system as created in OPNET

The data collected through the simulation of the system is used to create the system matrix with edge weights representing bandwidth where rows represent senders of packets and columns represent receivers of packets. An example of this matrix is shown in Table 3.1. An "X" denotes the direct connections between the individual network components while the "∞" is used to indicate the connection between the component and itself. This simple adjacency matrix is non-weighted and enhancements can be made to the decomposition process by weighting interconnections with the metric if interest; bandwidth in this example.

Table 3.1: Example system matrix with connections highlighted

	sw1	ser1	ser2	ser3	sw2	wks1	wks2	wks3
sw1	∞	X	X	X	X	0	0	0
ser1	X	∞	0	0	0	0	0	0
ser2	X	0	∞	0	0	0	0	0
ser3	X	0	0	∞	0	0	0	0
sw2	X	0	0	0	∞	X	X	X
wks1	0	0	0	0	X	∞	0	0
wks2	0	0	0	0	X	0	∞	0
wks3	0	0	0	0	X	0	0	∞

A. *System Decomposition Matrices*

The concepts of AA are used to evaluate the matrix created for the example network. This matrix displays several blocks of data that are largely independent from each other. This independence indicates appropriate places to decompose the system. Using the decomposition points from the matrix, the smaller system components are modeled separately.

The matrix created for the example system can easily be decomposed in terms of the geographical location of the network components. It is recognized that three of the buildings in the network are logical subsystems for this network, reflecting the top-down approach used by the original designer of this network. This geographically related method for decomposing the system would not necessarily be possible when dealing with extremely large, more complex systems. Figure 4.2 shows the network partitions based on the geographic decomposition.

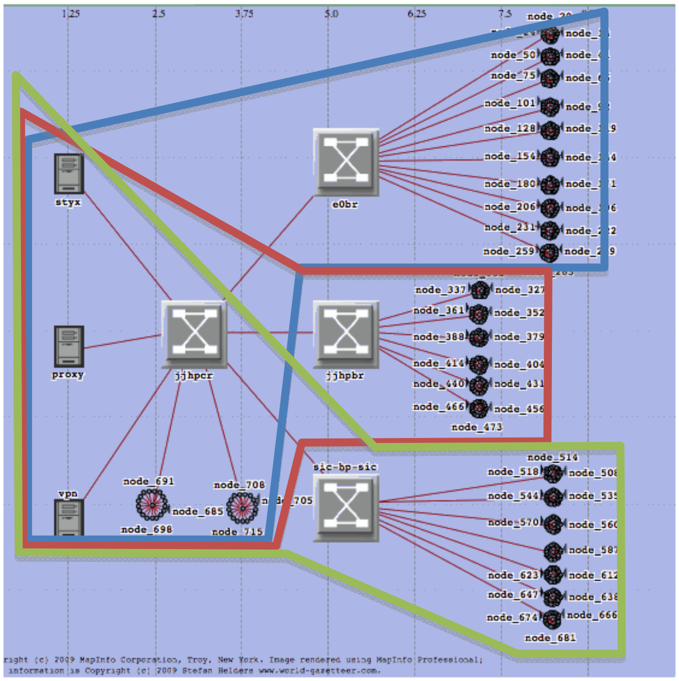

Figure 4.2: Example system with geographic decomposition

Another decomposition method explored is a sliding window matrix reduction. The sorted matrix was reduced in size by summing matrix elements within a square window to generate a new element in the reduced matrix. Performing a matrix reduction makes sense for dealing with extremely large matrices, decreasing the number of elements to be examined. In the case of the example matrix, an 8 x 8 window was used to reduce the matrix size. When the window operations had been performed on the matrix, the matrix was decomposed on the window boundaries. Each subsystem contains exactly the same number of components which has benefits and deficits as shown later in the results. This method results in the system decomposition as shown in Figure 4.3. The consistent window size for the matrix reduction breaks the system into equally sized subsystems and in future work we plan to explore methods that dynamically adjust window size during the decomposition process.

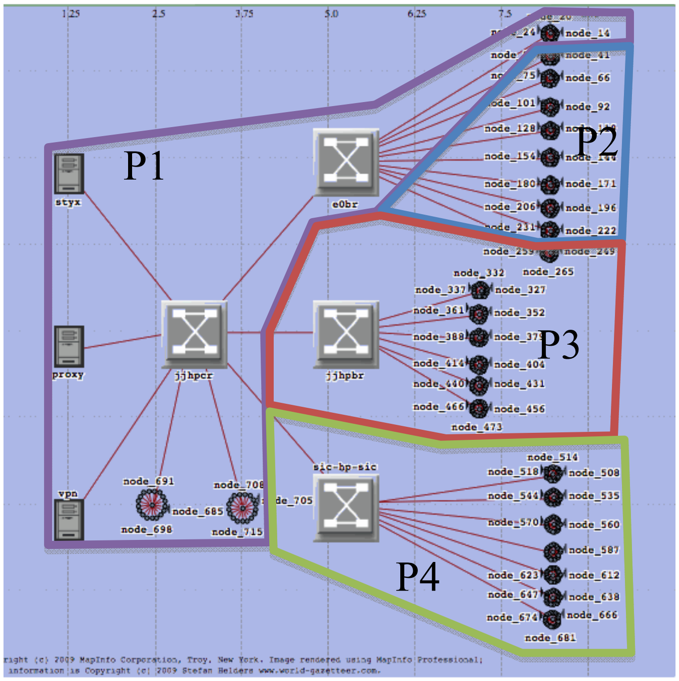

Figure 4.3: Example system with 8x8 window decomposition

V. Results

With a decomposition based on a strict application of constant partition size, the subsystems are broken into parts that are less independent and, in some cases, disconnected as seen in Figure 4.2. In this figure, Partitions 1 through 4 are highlighted. Partitions 1.2, 1.3, and 1.4 are combinations of simple partitions 2, 3, and 4 with partition 1 of the same method, respectively.

The amount of traffic between adjacent nodes is compared to determine how accurately the decomposed systems are modeling the entire system. In most cases the amount of traffic between adjacent nodes in the decomposed systems summed to the traffic seen in the entire system. As expected, this was not true in the case where the decomposed systems were not entirely independent. For instance, all traffic to a specific server from each of the buildings must travel through a single switch. When considered independently, this switch is sufficient to handle all traffic from a decomposed system. In the entire system however, the switch becomes overwhelmed with traffic and begins to drop packets. When decomposing a system, it is necessary to identify independent subsystems; conversely, no subsystem will be completely independent. If the subsystems were completely independent, it would indicate several entirely separate systems, not a single interconnected system.

Our results show that the decomposition of the network into smaller independent pieces is effective in both modeling the behavior of the entire network and reducing the length of the simulation runtimes. We established statistical data for the decomposed network to be comparable to the whole network simulation. The data may be viewed in Tables 5.1 and 5.2. These bandwidth error data are less than a 0.1% of the bandwidth simulation results of the total system. The error data collected indicate that decomposing a large system based on axiomatic principles can allow the modeling and simulation of very large systems with accuracy.

Table 5.1: Average error for partitions created based on geography

Partition Name	Average Error (kbit/s)
Building 1	1.6230
Building 2	1.9199
Building 3	1.8452

The first set of partitions based on a reduced matrix, listed in the first four entries of Table 5.2, creates differently connected subsystems from the original system than the other partitioning methods. This variation in the connectedness explains the difference in the error for this partitioning when compared with the geographically based partitioning and combined reduced matrix partitioning. It should be noted that the error for the first single partition method is still less than 0.1% of the total bandwidth of the system.

Table 5.2: Average error for partitions created from reduced matrices

Partition Number	Average Error (kbit/s)
1	2.3081
2	2.4004
3	2.1045
4	2.1406
1.2	1.6914
1.3	1.6401
1.4	1.6851

The timing data in Table 5.3 show the significant decreases in computer simulation runtime associated with decomposing the system. In Table 5.3, the first column indicates the decomposition method used, the second column indicates the total runtime by summing together the simulation runtimes of each decomposed subsystem, and the third column indicates the overall runtime decrease when compared to the simulation time of the entire system.

Table 5.3: Partitioned system total runtime reduction

Partition Method	Total Runtime	Runtime Reduction
Whole System	16:49:10	-
Single	3:53:05	76.90%
Combined	12:11:38	27.50%
Geographical	11:02:23	34.36%

Simulating the decomposed system partitions in series versus simulating the entire system as a whole considerably reduced run times. The single partitions from the reduced matrix were the smallest subsystems to be simulated, thus, the fastest. The run times for the decomposed systems could be further improved by running the subsystem simulations in parallel, as opposed to in series.

VI. Conclusions and Future Work

The ability to simulate a large system is crucial in the area of disaster tolerance since it is necessary to simulate the large system in normal operating model, in the presence of a disaster, and with modifications to the normal system designed to enhance disaster tolerance. From this point of view large system simulation is a chief computational challenge in disaster tolerance engineering.

Decomposing large systems is shown to drastically reduce the runtime of simulation for a large system. Partitioning methods that exploit the most commonly connected portions in multiple partitions result in a reduction in errors. These combined partitions do not experience the same massive reductions in simulation time, but achieve a closer resemblance to the original large system. The application of both the connectivity and the bandwidth of the example system to the partitioning method allowed the development of more realistic partitions, as opposed to the traditional approach of only considering connectivity.

We plan to explore further refinements to our partitioning methods to determine if we can improve our results in large system simulations. This in turn has the potential to make large system simulation more accessible to industries requiring more complete system data. We will also experiment with nonlinear decomposition methods such as those based on hierarchy instead of the flat approach described here. We expect that such decomposition methods may better reflect the structure of large systems and thus produce more accurate simulation results but will require more processing in the final step of combining individual simulated subsystem responses to determine total system response.

References

[1] S. A. Szygenda and M. A. Thornton, "Disaster Tolerant Computer and Communication Systems," *International Conference on Cybernetics and Information Technologies, Systems and Applications* (CITSA 2005), July 14-17, 2005, pp. 171-173.

[2] N. P. Suh, **Axiomatic Design: Advances and Applications**, Oxford University Press, Oxford Series on Advanced Manufacturing, New York, New York, May 2001.

[3] D. Easton, M. A. Thornton, and V. S. S. Nair, "Axiomatic Design Process for Disaster Tolerance," *Proceedings of the 11th World Multi-Conference on Systemics, Cybernetics and Informatics* (WMSCI), July 8-11, 2007.

[4] D. Easton, M. A. Thornton, V. S. S. Nair, and S. A. Szygenda, "A Methodology for Disaster Tolerance Utilizing the Concepts of Axiomatic Design," *IIIS Journal of Systemics, Cybernetics and Informatics*, vol. 6, no. 4, 2008.

[5] S. Riyavong, "Experiments on Sparse Matrix Partitioning," Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS), 2003.

[6] G. Karypis and V. Kumar, "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *Society for Industrial and Applied Mathematics Journal of Scientific Computing*, vol. 20, No. 1, pp. 359-392, 1998.

[7] M. A. Harper, M. A. Thornton, and S. A. Szygenda, "Disaster Tolerant Systems Engineering for Critical Infrastructure Protection," *IEEE Systems Conference*, April 9-12, 2007, pp. 2-8.

[8] C. M. Lawler, M. A. Harper, S. A. Szygenda, and M. A. Thornton, "Components of Disaster Tolerant Computing: Analysis of Disaster Recovery, IT Application Downtime & Executive Visibility," *International Journal of Business Information Systems*, vol. 3, no. 3, 2008, pp. 317-33.

[9] Y. Y. Haimes, "Roadmap for Modeling Risks of Terrorism to the Homeland," *Journal for Infrastructure Systems*, pp. 35-41. June 2002.

[10] OPNET Technologies, Inc., http://www.opnet.com/.