Multi-Objective Design Methodology using Decision & Causal Bayesian Belief Networks for Complex Data Centers focused on Risk and Cost

By Jayneel Jaydev Patel

B.S in Computer Engineering, May 2005, Pennsylvania State University
M.S. in Software Engineering, May 2011, Pennsylvania State University

A Dissertation submitted to

The Faculty of

The School of Engineering and Applied Science
of The George Washington University
in partial satisfaction of the requirements
for the degree of Doctor of Philosophy

January 31, 2014

Dissertation directed by

Shahram Sarkani
Professor of Engineering Management and Systems Engineering
and

Thomas Andrew Mazzuchi
Professor of Engineering Management and Systems Engineering

The School of Engineering and Applied Science of The George Washington University certifies that Jayneel Jaydev Patel has passed the Final Examination for the degree of Doctor of Philosophy as of December 6th, 2013. This is the final and approved form of the dissertation.

Multi-Objective Design Methodology using Decision & Causal Bayesian Belief Networks for Complex Data Centers focused on Risk and Cost

By Jayneel Jaydev Patel

Dissertation Research Committee:

Thomas Andrew Mazzuchi, Professor of Operations Research and of Engineering Management, Dissertation Co-Director.

Shahram Sarkani, Professor of Engineering Management and Systems Engineering, Dissertation Co-Director.

E. Lile Murphree, Professor Emeritus of Engineering Management and Systems Engineering, Committee Member.

John Bischoff, Professorial Lecturer in Engineering Management and Systems Engineering, Committee Member.

Jason Dever, Professorial Lecturer in Engineering Management and Systems Engineering, Committee Member.

Acknowledgements

I wanted to extend special thanks to Dr. Shahram Sarkani and Dr. Thomas Mazzuchi for their guidance and support in my doctoral research. Sincere thanks to the EMSE office for its assistance throughout the doctoral process. Furthermore, I would also like to thank my friends and family for their support and their patience throughout my doctoral research process.

Abstract

Multi-Objective Design Methodology using Decision & Causal Bayesian Belief Networks for Complex Data Centers focused on Risk and Cost

A multi-objective energy efficient and flexible data center design has been of interest for all in the industry that are familiar with data center costs. A survey of 416 CIO's conducted by the CIO Data center Strategies Survey, suggested that the data center operating costs are steady at 25% of the entire IT budget [21]. Much of this cost is directly attributed to constant scaling of data centers to address the computational and network demands. Additionally, the constant demand restructure negatively affects the non-functional costs of maintaining the data center. Many of the existing data center enhancement strategies include unilaterally adding physical servers & switches, virtualizing servers (including cloud computing), and intelligently scheduling tasks.

To date, there have been very few data center design frameworks that accommodate multiple decision factors to address design objectives while also forecasting for future enhancements. The objective of this research is to design a framework using decision networks that can help create and enhance cost effective data centers. The total cost is measured in relation to the energy consumed by the entire data center network. The total cost is fed back in the system for future design decisions. The decision network is developed using Netica [22], and the data center energy is simulated using a simulation tool called GreenCloud [8].

Key words: data center design, decision networks, data center operations, multi-objective design

Table of Contents

Acknowledgements……... iii

Abstract……..…… iv

Table of Contents……... v

List of Figures……..……..... ix

List of Tables……...……... xi

List of Equations……..…… xii

Chapter 1 - Introduction……..……1

Chapter 2 – Review of literature..……3

2.1 Data center decision factors…….4

2.2 Heating and cooling……...6

2.3 Virtualization and automation……...7

2.4 Systems design……..9

2.5 System models…….11

Chapter 3 – Methodology……....14

3.1 Rationale for using decision network……..……..14

3.2 Domain knowledge and causal relationships……..15

3.3 Design model demand……....19

3.4 Data centers…..……...….20

3.4.1 Data center setup……..……..…..21

3.4.2 Data center expansion……...23

3.4.3 Data center costs, risks, and designs……..…23

3.4.4 Types of data center demands……..25

3.4.5 Data center decision……..……....26

3.4.6 Data center governance…….26

3.4.7 Current data center limitations……..…27

3.5 Data center states…….27

3.6 Costs…….…….31

3.6.1 Operational cost……..32

3.6.2 Transition cost……..…….....34

3.7 Risk…….……...……35

3.8 State transition..……...……....38

3.9 Decision model……..……..39

3.10 Demand estimates…….……42

3.10.1 Monte Carlo method……..……..43

3.10.2 Holt's method……44

3.11 Causal Bayesian Belief Network……..……...45

3.12 Decision network……..……..46

3.12.1 Decision Network Netica model……...…48

3.12.2 Overall nets……....48

3.12.3 Beliefs for each decision question……...……..50

3.13 Decision utility…….……...52

3.13.1 Utility values……...……56

3.14 Sensitivity analysis……..……..58

3.15 Influence analysis..…….……...60

3.16 Developing the model…….……..…61

3.16.1 Building causal Bayesian belief network……..……....64

3.17 Simulation..…….66

3.17.1 Setup……...……66

3.17.2 GreenCloud……....67

3.17.3 Simulating server and network demand……....67

3.17.3.1 Geometric Brownian…….68

Chapter 4 - Results and discussion……75

4.1 Capacity reduction results……..80

4.2 Future work……...…82

References……..……..…86

List of Figures

Figure 1: Correlation between decision factors and data center capabilities……..…18

Figure 1a: No visibility on which decision factor affected the data center demand……18

Figure 1b: Visibility on direct correlation of decision factors to the data center demand……18

Figure 2: Typical high speed multi-tier data center model……....23

Figure 3: Server and network state efficiencies……..29

Figure 4: Setting up smaller data center states for the study from one large (current) state….....30

Figure 5: System states and costs……31

Figure 6: Data center states……..32

Figure 7: Visual illustration of the data center states…….33

Figure 8: Risk assessment table……...37

Figure 9: Chance and decision nodes……..38

Figure 10: Proposed decision model……....39

Figure 11: N-Squared chart of the decision network operating model……..…41

Figure 12: Demand estimates using Monte-Carlo……..…43

Figure 13: Demand estimates using Holt's……..…45

Figure 14: Minimalistic data center decision network……....48

Figure 15: Causal Bayesian belief network…….51

Figure 16: Utility structure of the deterministic decision network…….54

Figure 17: Steps in the decision model design & system inception……....61

Figure 18: Model construction steps and the algorithm used to govern the BBN.....65

Figure 19: Geometric Brownian growth chart for server and network simulation…….69

Figure 20: Data center load……...70

Figure 21: Energy conservation of servers…….….71

Figure 22: Server and network energy costs……..74

Figure 23: Decision model and data center demand…….75

Figure 24: Ideal data center state narrowed down using BBN…….80

Figure 24a: First few iterations…….…80

Figure 24b: Next iterations……....80

Figure 24c: Final iterations……...80

List of Tables

Table 1: Data center history……6

Table 2: Operational cost……...32

Table 3: State transition cost……35

Table 4: Demand estimates using Holt's method……...44

Table 5: Overall net values for the decision network……49

Table 6: Belief table for the decision network……50

Table 7: Traditional and decision model approaches……52

Table 8: Utility table and values for the decision network……56

Table 9: Entropy reduction values of the BBN proposed above……..60

Table 10: Decision network and GreenCould……66

Table 11: State server and network configurations……67

Table 12: Decision table and rationale……72

Table 13: Data center simulation of the states……73

List of Equations

Conditional probability - Bayes Theorem

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

Each variable is conditionally independent of all other nodes in the network

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i|x_{i+1}, ..., x_n) = \prod_{i=1}^{n} P(x_i|Pa(x_i)) \tag{2}$$

The number of possible directed acyclic graphs in the decision network.

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \quad \begin{matrix} n \geq 2 \\ f(0)=1 \\ f(1)=1 \end{matrix} \tag{3}$$

Entropy for variable Q before any findings for the nodes in the system

$$H(Q) = -\sum_{q} P(q) log_2 P(q) \tag{4}$$

Conditional entropy in sensitivity analysis

$$H(Q|F) = \sum_{f} P(f) H(q|f) \tag{5}$$

Entropy reduction I in sensitivity analysis

$$I = H(Q) - H(Q|F) = \sum_{q} \sum_{f} P(q, f) log_2 \frac{P(q, f)}{P(q)P(f)} \tag{6}$$

1. INTRODUCTION

Governance of information technology (IT) is based on several technical and business decision factors. This paper demonstrates system engineering methodology using decision networks to aggregate decision factors and decision rules that impact data center management. The goal of the proposed design is to control (&lower) data center maintenance cost while managing data center risks. The model is applicable to institutions that employ their own data centers. This model will not apply directly to institutions that outsource their data center needs. The data center design model proposed in this paper does three things. First, the model allows the data center design to evolve with the changes in the network and server demands by employing decision networks. Second, the model design additionally allows the institutions to employ a systematic approach for scaling their servers and network switches to save costs. Third, the design model takes multiple objectives and decision factors cohesively to govern the overall data center architecture in order to reduce life cycle costs. Although we focus on two main decision factors, server and network demand, and 3 decision rules, the design is scalable to include other decision factors as the institutions may see fit.

One of the major goals of any data center manager is to appropriately scale the data center to keep it cost effective. Some designs allow the data center to respond to continued growth by adding servers, network switches, or virtualizing existing physical resources [2]. Data centers are rarely downsized due to the fear of experiencing unforeseen high network and/or sever demands. However, when data centers are downsized, the institutions either dynamically shutdown servers or physically remove the servers [23]. Since the most important principle for any data center is its availability (as opposed to flexibility or maintainability), in most cases the data center is overdesigned

and left running at their max capacity to avoid any downtime due to network or computational congestion. Such designs are hence risk averse (risk of experiencing downtime) but also high in cost for the institution.

This paper explores the opportunity to use the principles of Bayes theorem and the multi-objective decision networks to design a data center management model that helps control cost, address uncertainties in the decisions, and manage data center risks. Bayes theorem uses the principle of conditional probability. Probabilistic approaches in system design and maintenance are heavily used in systems engineering framework [3] and also in data center fault detection strategies [30] but not as significantly in the area of data center design & maintenance strategies. Hence the application of decision theories to data center design and maintenance are not very far off from what is practiced in the industry, hence making the application of decision network for data center maintenance an ideal research area.

This paper first introduces the systems engineering model proposed through our research. The paper then describes each aspect of the model. We begin by explaining the elements of a typical data center architecture followed by a comparison of the typical design approach to the proposed model. We briefly cover how the server and network estimates were calculated. Furthermore, the paper covers the decision network in detail explaining each element and aspect of the decision making process. The paper then discusses the steps required to setup the proposed model while offering justification for each step. Finally the decision network and the data center are simulated using Netica and GreenCloud respectively, while the results are discussed in the final section of the paper.

2. REVIEW OF LITERATURE

According to my observations and the knowledge on data centers, originally when data center architectures were first designed, the data center systems were kept static. That is, the server and the network bandwidths were rarely changed. The server and network capacities were estimated at the maximum anticipated demand and left alone. Overtime the design began to change as the computing demand started to increase, floor space became limited & expensive, and power management became expensive. Based on our professional experience, as the computing demand grew companies instituted more servers to meet the demand. As a result floor space began to become sparse and system cooling became a real problem when more servers began to increase the temperature of the data center. In my opinion, cooling issue shed some light on the lack of ventilation at the data centers. Hence, the concept of raised floors, proper ventilation, and geographic locations became significant in the data center design decision process. Geographic locations with generally cooler temperatures became ideal sites to build data centers. Furthermore, based on my own observations over the years, institutions began to explore ways estimate computing capacity as accurately as possible to minimize the effect of over deploying servers on the data center floor.

With the explosion of dot net technology and websites the data center demand became heavy. In my opinion the decision makers then did not have the task of balancing server capacity and resulting server heating problem. However, the high demand for constant web traffic introduced a strong need for availability. Based on my observation, since the data center demand estimation process was not mature, data centers often underestimated the demand. Underestimation of capacity resulted in outages in data center availability. Hence the data center decisions began balancing decision factors

among data center capacity, data center cooling, and data center space. In my opinion decisions on data center are made by the business team while there were no formal decision methodologies adopted in the data center decision process. As data centers began to evolve so did the decision factors used to govern them. Below is a list of decision factors that most data centers today institute according to the Philadelphia technology park [33].

2.1 Data Center Decision Factors

- Real Estate
- Core infrastructure
- Data Center Operations
- Availability and cost
- Proximity to fiber and choice of providers
- Space size and scalability
- Commitment - contract vs. lease
- Time to market
- Heating, venting, air conditioning (HVAC)
- Security components (manned, access, surveillance)
- Power infrastructure
- Fire suppression
- Security
- Maintenance

- Monitoring
- Staffing

The decision factors originally started with only functional factors such as servers and networks. The decision factors began to include non-functional factors such as cooling and power distribution during the dot com burst in the 90s. In the 2000s the primary decision factors shifted from functional to non-functional factors since the non-functional factors became more expensive than the functional factors in the data center design. Hence the data center design methodologies began concentrating on the non-functional factors just as much as the functional factors.

Typical data centers are of 3 to 4 tiers in size. Each tier of the data center brought specific value in reliability as part of the original design. The tiers brought first level of data center scalability to the design.

Tier 1: Applies to small businesses. The decisions are directly reflected in the data center design.

Tier 2: This was built as an extension of Tier 1 that could be turned off whenever underutilized.

Tier 3: Originally designed to support the internal framework for companies that had 24x7 help desk services.

Tier 4: Businesses based on e-commerce and online transactions utilized 4 tier framework. The framework allowed businesses that were client-facing to have the business available for business 24x7 for online transactions.

Below is a table that summarizes the evolution and benefits that each data center tier added to the business decision. [28]

	Tier 1	Tier 2	Tier 3	Tier 4
Number of delivery path	Only 1	Only 1	1 Active/1 Passive	2 Active
Redundant components	N	N+1	N+1	2(N+1) or S + S
Support space to raised floor ratio	20%	30%	80-90%	100%
Initial watts/SF	20-30	40-50	40-60	60-80
Ultimate watts/SF	20-30	40-50	100-150	150+
Raised floor height	12 Inches	18 Inches	30-36 Inches	30-36 Inches
Floor loading pounds/SF	85	100	150	150+
Utility voltage	208, 480	208,480	12-15kV	12-15kV
Months to implement	3	3 to 6	15 to 20	15 to 20
Year first deployed	1965	1970	1985	1995
Construction $/SF raised floor	$450	$600	$900	$1100+
Annual IT downtime due to site	28.8hours	22hours	1.6hours	0.4hours
Site availability	99.67%	99.75%	99.98%	99.99%

Table 1: Data center history

2.2 Heating and cooling

Data center cooling is a secondary parameter in data center decisions. However, data center cooling can become a primary decision parameter for some data centers that experience constant change in their systems. The architecture of the data center is

carefully designed during the initial design stage to address data center cooling. The cooling factors are addressed by creating just the right amount of space all around the servers. It is trivial to calculate the server space between the servers by simply monitoring the temperature around the servers and spacing them accordingly. The data center space under the servers is created through raised floors. The methodology introduced in mid 1900's, raised floor not only provides ventilation for the servers but also allows for adequate power management. Raised floors allow the ability to manage power distribution by running cables under the raised floors. There are many ventilation and cooling techniques now applied to the data center designs.

2.3 Virtualization and automation

Virtualization starts by taking all physical resources and adding them up in a common pool. When a system is requested the virtual machine is created by allocating a specific portion of physical resources in a virtual space. Virtualization can be tricky because Virtual system have nothing physical to track. The machines themselves are Virtual which are created when users log in the framework and a user session is created. Once the users log off the system resources are returned back to the pool. Security is of a big concern in Cloud Computing. Since the sessions are difficult to track, they are difficult to make secure. Furthermore, since virtual machines are built on top of the physical resources it adds an additional level of complexity in the system.

Physical virtualization was first made possible by IBM mainframe systems by introducing the Logical Partitions (LPar) technology. LPars allow the mainframe systems to build dynamic partitions in the system. The user can build a completely new system in the

newly created partition. Once the use for the system is done, the user can easily break down the partition to return system resources back to the system pool. LPars are capable of not only running separate operating systems but also different operating systems. One LPar can have z/OS while other can have Linux installed. LPars can also be of different sizes. One LPar can be 100GB while the other of 400GB for example.

There are many virtualization products on the market that allow institutions to virtualize their server farms. VMWare is one of the most popular virtualization product offered in the Unix and Windows environment. Beyond the LPar technology, mainframe also offers its own version of virtualization. This virtualization unlike VMWare is offered in Linux only. While IBM's LPar technology is semi-permanent, the virtualization technology is extremely dynamic. However the virtualization technology at IBM can only accommodate Linux operating system. Could computing takes the concept of virtualization from virtualizing physical resources to virtualizing the environment. Hence, the entire system adds another layer of complexity to the design. Additionally the system security at the cloud level is even harder to address than at the system level since cloud computing is still at its early stages. While the engineers continue to address security concerns around virtualization, work towards securing cloud computing still continues.

In the area of distributed computing, advancements have been made to automate the way servers are activated and shut down. With the advancements in the network technology the data center is equipped with the ability to dynamically shutdown servers (DNS). The data center management was initially governed manually in the 90's. During the past decade the data center governance was made automated through a software controlled system. The governing software system was evolved to employ policy based

decisions and server activation techniques. However, the policy based decisions were developed using simple algorithms.

Policy based decisions over the past 5 years began to include probability based approach to determine faults in the system before they would occur. The probabilistic approach was applied to other areas of the data center such as cooling and ventilation. The system would determine which servers to turn on based on the estimated temperature values in the system. However, the systems have never had a total probabilistic approach in governing the data center until now. The research in this paper introduces the use of decision networks, a probabilistic approach, to govern the data center. Decision networks have been applied to various different fields. Many systems use a software application to build and apply decision networks to govern the system. However there has been a need for a system that can encompass all decision factors, weigh the risks, and reduce the operational cost.

2.4 Systems design

There has been a tremendous amount of progress in IT for the past forty years. Most of the progress in the data center technology has been around specific areas of the datacenter. These areas include, network switching, priority queuing, scheduling, dynamic shutdown, carving out new partitions dynamically, virtualization, power distribution unit (PDU), and cooling facility design (CFD). However, very little progress has been made to address the overall data center design and model.

Furthermore, quite a few dynamic design approaches have been implemented with data center design. However, all of these approaches address a single objective at a time.

These approaches include using priority queuing in the network routes, dynamic scheduling of the tasks, virtualizing the environment so that the underlying physical resources is shared, carving out new physical partitions or resources for scaling up the design, and monitoring the cooling facility devices (CFDs) and scheduling tasks accordingly. What has been lacking is a framework that can take all decision factors into account to help govern the design and maintenance of the entire system. Hence, I used decision networks to achieve this very goal.

Employment of decision network in the application of systems engineering is becoming stronger. Areas such as problem localization, systems decision based on time expanded decision networks, and thermal monitoring are to name a few. Research has also been carried out to apply the concepts of decision networks to create and assess a situation using simulation characters in an artificial intelligent scenario.

IT's approach to a problem has been to focus and to address on a specific pain point at a time. So, the enhancements and designs implement a focused approach that can help fix the specific problem being experienced. While these approaches work great in the modern day data centers, the solution still ends up being extremely specific which makes the overall system highly coupled and less cohesive. As a result, the risk in the system exponentially increases. Any one change in the system is likely to make another part of the system suffer. To compensate for this difference more changes are hence introduced in the system which in turn has the possibility of increasing its risk even more. Keeping the risks in mind, a framework for designing and developing a data center design with multiple objectives that tied to the decision process became necessary.

2.5 System models

The system models have evolved over time. Not only the system design has evolved but also has the approach on the design itself. The system design at first was stagnant. The purpose of the design was to make sure that all quantitative system requirements were met. As the process evolved so did the need for qualitative measures to be involved in the design. Some of the designs included single objective that pairs with the system's primary objective.

Over the period of time many different system architectures have been developed. Each design helps address a specific area of interest or approach in a design. Some of these methods include Design for manufacturing and assembly (DFMA), Group technology, and Taguchi methods. Quality focused designs have used a Quality function deployment (QFD) approach. Furthermore, fault detection and tolerant design use the Failure mode effects analysis (FMEA). Hence, several different tools have also been developed to assist with the system design such as computer aided design (CAD), computer aided engineering (CAE), computer aided manufacturing (CAM), and product data management (PDM) [26].

As systems began to interact with each other systems so did the designs and the design methodologies. The concept of unified modeling language migrated from systems engineering to computer and software engineering as well. Hence this helped address the primary objective of integration of systems. Through UML the components of the systems themselves could be mapped as a way of capturing the entire system. The approach hence migrated to building a central model including a process model using UML that encompasses systems of systems.

Once the complex systems were integrated or designed a system measurement and analysis using objected oriented process and simulations was utilized to measure performance. The process typically takes sensors, processors, and communication systems into consideration for performance assessment. As systems became more complex and multiobjective it wasn't enough to maximize just one aspect of the system while ignoring other aspects. Engineers needed a way to maximize one aspect of the system while minimizing the other. Hence the system models began to use Pareto's principle for multiobjectivity design strategy.

Eventually the designs led to the use of probabilities to focus on particular aspects of the system such has forecasting fault detection, system readiness levels (SRLs), technology readiness levels (TRLs), and integration readiness levels (IRLs). As a result the system designs led to decision networks. This form of system design can include not only the objective design strategies but can also use multiple subjective user needs in the mix. Using probabilities as the foundational aspects of the design, the system can minimize risk in uncertainties while maximizing results.

The foundation of decision networks rely on the principles of Bayes theorem. Bayes theorem uses conditional probabilities to determine the probability of an event or a state while taking into account the given information regarding that state. When using singular criteria for evaluation probabilistic values are used to break down the information into a decision tree. However, decision trees don't accommodate for the uncertainties or the relationships between parameters quite so well. As a result, Bayes theorem helps address the uncertainties in the design by taking their probability into account while determining the probability of the state at question.

Bayes theorem is used to understand the impact of many states on one another in a network, a principle also known as the Bayesian network. Bayesian theorem uses the probability density function in relation to the parent values in the system. In decision network the output of the network is assessed against the utility values which help determine the satisfaction of the primary stakeholder or the objective in the system. In a deterministic environment the utility values or the satisfaction results are predetermined based on the preferences. In this design the satisfaction rates are strongly correlated to the system risks. When adding new parameters to the design it is imperative to make sure that the model is a directed acyclic graph (DAG) and that it satisfies the local Markov property with respect to DAG. In other words, the decision network can't have a looping dependency in its design.

Since the major limitation in data center design process was being unable to take multiple objectives in a risk averse decision process, I expanded the study of decision network from the systems engineering domain to the IT domain. The reason behind using an iterative approach in the decision network system model was to allow the model to continue to provide most cost effective and multi-objective solutions during the data center maintenance stage. This system therefore is flexible in include new factors in the design after inception. At minimum there can be the following number of nodes:

The numbers of nodes in the system are equal to the number of qualitative levels raised to the power of number of factors.

> **Numbers of Nodes (n)** = qualitative levels: (3 - high, medium, low) ^ factors: (2 - servers and network) = 3^2 = 9 possible states.

Number of total arcs = n^n (number of possible choices the system can decide from)

Each stage in the decision network consists of decision node and a group of chance nodes. Collectively they represent an instance of the data center. For bandwidth and computational changes, the model allows the datacenter to expand (at minimum) in two different areas. These levels include network changes to address the bandwidth needs and server changes to address the computational needs.

3. METHODOLOGY

3.1 Rationale for using decision network

Data centers go through numerous updates on weekly and sometimes daily basis. Institutions perform their own change risk assessment, change management, and impact assessment. However, the design is still not fool proof. Mistakes get made all the time during data center changes thus adding more to the maintenance cost. The cost of operating and maintaining a data center is high enough. The decisions that drive the data center attempt to low the cost but due to not having a method to weight each decision factor against each other prevents the data center managers to not only lower cost to its maximum potential but also weigh risks quantitatively in the decision process.

Decision networks allow its users to do just that. It uses conditional probabilities to weigh decisions against each other to provide the best decision with the information given at hand. Accuracy of the decision certainly depends on the accuracy of the parameters fed in, as its true for any decision process. Using decision network to design data center maintenance strategy seems like a perfect fit since the decision networks can be used to assess risk while trying to address the main objective which is to reduce cost. The network allows the parameters and its values to work holistically. Current data center maintenance strategies lack a system that helps reduce cost while balancing risks. Since the data center are no long static and accept change on weekly and sometimes daily basis, a more holistic and comprehensive tool therefore was necessary to drive the data center maintenance strategy.

3.2 Domain knowledge and causal relationships

As mentioned earlier, data center capacities are influenced by business and technical decision factors. However, most domain knowledge for IT systems contain technical decision factors since business decisions are often filled with subjective reasoning, personal preferences, and factors that are not directly related to the IT design [23]. For this reason, we used causal Bayesian belief network (BBN) to holistically capture the business and IT decision factors. Causal reasoning and conditional probabilities allowed us to use statistical knowledge of the system mapped using the causal relationships. Hence, our data center capacity reduction model encompasses statistical relationships and wholly causal knowledge that exists in the data center domain. A common risk in any design is that the design itself introduces uncertainties in the framework. Hence, in order to address the uncertainties (and to reduce them), we performed sensitivity analysis on each relationship node in the model. More details on the sensitivity analysis are discussed later in the paper.

Causal analysis methods have been commonly used to analyze technical and non-technical associative learning [16]. Causal Bayesian belief networks have been used to capture the relationships and influences among the systems within the different domains. A typical approach applied to data center design, maintenance, and lean capacity planning focuses on implementing an improvement plan on a specific area of the data center. The common approach is to explore new ideas and technologies that can help address a single area of the data center design that often acts as a bottle neck. Furthermore, most of the factors to be addresses are based on non-functional requirements of the design such as heating & cooling, floor space, and staffing [7]. Following are the pros and cons of the BBN v/s the focused approach.

Focused Approach:

- Pro:
    - The study and the change are much focused.
    - Good for small quick changes or ongoing chronic issues
- Con:
    - Improvement is on a single objective.
    - Functional and non-functional requirements are separately looked at
    - No or little visibility on the effects to the rest of the system
    - High overhead due to possible unused data capacity
    - System upgrades and maintenance need to be specially carried out and impact assessment need to be performed each time.

Causal Bayesian Belief Network Approach:

- Pro:
    - The study can focus improvement on the whole system
    - Learn long term influence of each parameter on the rest of the system
    - Good for long term cause and effect study
    - Inexpensive way to study
    - Low overhead due to the ability to develop lean designs
    - System upgrades and maintenance could be included as a part of the design
- Con:
    - Requires probability values on all the nodes of the system
    - Overall improvement/changes could become expensive

Business risks play a major role in IT decision making. Often the system is oversized to prevent availability related risks in the design. Hence, it is often difficult to understand if the present system was grown to a certain size because of IT needs or because of the fear of unavailability of the systems. Furthermore the management team may not know which factors it should monitor more closely out of the list of factors that attribute to data center demand. Lastly, the management and the IT team may also not know the level of uncertainty that exists in the design and how strong the correlations are between the decision factors and the end decisions.

The focused approach helps control the end output but does not have a way to correlate the knowledge of the system and its factors for a holistic approach. Hence, by using focused approach an institution may not be able to change the system significantly without altering and possibly adding another risk in the system. One way to address this issue is by collecting domain knowledge to better understand which factors have influenced the data center demand the most and will continue to do so going forward [29].

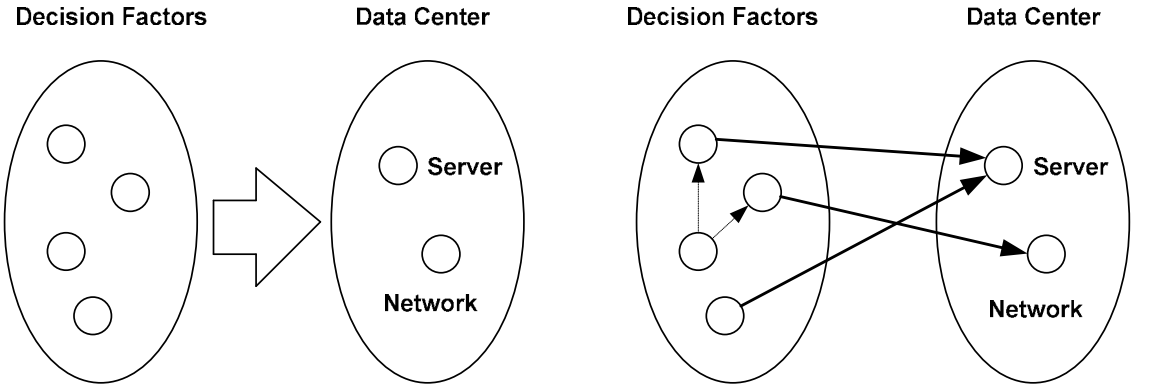

Fig 1a: No visibility on which decision factor affected the data center demand

Fig 1b: Visibility on direct correlation of decision factors to the data center demand

Figure 1: Correlation between decision factors and data center capacities

In order to understand the significance of different decision factors we used sensitivity analysis in our design. Sensitivity analysis can help reduce uncertainty in the system and can help with the study of influence of each parameter on the end decision of data center capacity change. Figure 1a shows how multiple factors collectively influence data center demand. However, not all factors may influence the decision equally and some may not even influence it at all.

Figure 1b shows that the proposed approach can allow the management to narrow down which factors influenced the data center decisions and their significance by the probability weight. The dotted lines represent the influence of one decision factor to the other. Hence the proposed approach can help figure out the factors that should be tracked further and monitored for future capacity planning.

3.3 Design model demand

The demand of server capacity is very different from network capacity. The network demand can be directly correlated to the demand coming in whereas the demand of the server capacity is evaluated based on the MIPS usage. High bandwidth could use low computational capacity and similarly low bandwidth demand could be computationally heavy. Therefore, computational forecasting is an independent entity. There are 3 types of demands. These demands include constant, trending, and seasonally fluctuating [29]**.** Seasonally fluctuating demands will be based on the past seasonal demands observed. Hence, for the first year of datacenter inception the seasonal data will not be available. Therefore trending demand could be used in its place.

Demand is estimated using the Holt's method. Since the states and the capacity levels are discrete values it is important to pick discrete value estimations instead of employing a method that provides with continuous estimated values. Each stage also provides a level of buffer zone. Servers are designed so that they don't ideally run at their 100% capacity. However if necessary the servers have the ability to run at its maximum capacity for certain spikes in the demand. Therefore, the servers themselves add a level of buffer zone to their design. Additionally the states are selected so that the system has a buffer zone to operate under.

The buffer zone built in the design could accommodate for at least two standard deviations from the maximum value. The Gaussian graph can be generated from the systems if the institution chooses. Since each environment is unique, the system construction will therefore require an iterative process to determine the ideal construction of each state. As a result the first year's data will not only be used to solidify the capacities but also the spans of each state. Since the design structure is risk averse, small demand peak situations are expected to be absorbed by the design. For unexpected high demand peaks the traffic would be transferred to the backup systems that are used seldom.

3.4 Data centers

Traditionally data centers are setup as static designs in which servers and networks are seldom removed. Some of the designs have focused to include dynamic shut downs. Dynamic shut down technology offers data center managers the ability to use a data center monitoring tool through which they can shut down a part of the data center. Often

the dynamic shutdown governed through artificial intelligence built in the data center monitoring software.

Initial data center monitoring systems were simply data acquisition systems. The software performed a pulse check on all the system to make sure they were up and running. The pulse check allowed the data center managers to make sure that their systems were available. As the acquisition technology matured so did the need for more information from data center monitoring system. The data acquisition systems began logging respective values at each aspect of the data center. With the new system, the data center managers had the visibility to the data center performance. The new data acquisition system allowed the data center managers to data center decisions real time. Before that system managers either never made any data center changes (by setting the data center system at its max capability regardless of the demand) or made seasonal changes ahead of time. Over the last decade the data center monitoring system grew from just a data acquisition system to a system that can automatically respond and adjust the data center architecture based on functional and non-functional parameters.

- Functional Data Center Parameters

  Functional parameters for the data centers are also its primary decision factors. The parameters include server capacity and network bandwidth. Secondary functional parameters include data center cooling, power management, and in some cases data center cooling. The primary parameters could be governed through the dynamic shutdown technology.

- Non-Functional

The non-functional parameters for data centers include floor space, security, fire suppression, data center operations, time to market, and maintenance are just to name a few. The non-functional parameters affect the data center decision process however, these parameters are not continuously measured. These parameters are carefully evaluated during the datacenter design. The architecture is adjusted again when functional parameter design decisions are weighed against the non-functional parameters.

3.4.1 Data center setup

A typical high speed multi-tier data center at minimum includes 4 levels. These levels start with the core L3 switches. Next, the design contains aggregate switches which are typically twice the number of core switches. The next level contains the access switches. These switches depend on the number of servers and the number of aggregate switches chosen in the design. The final layer contains servers [1]. The number of servers, and their work rate, depend on the computational capacity per server and the aggregate unit of servers per server rack. Typically these servers are rated as 1U, 10U, and 40U servers [29].

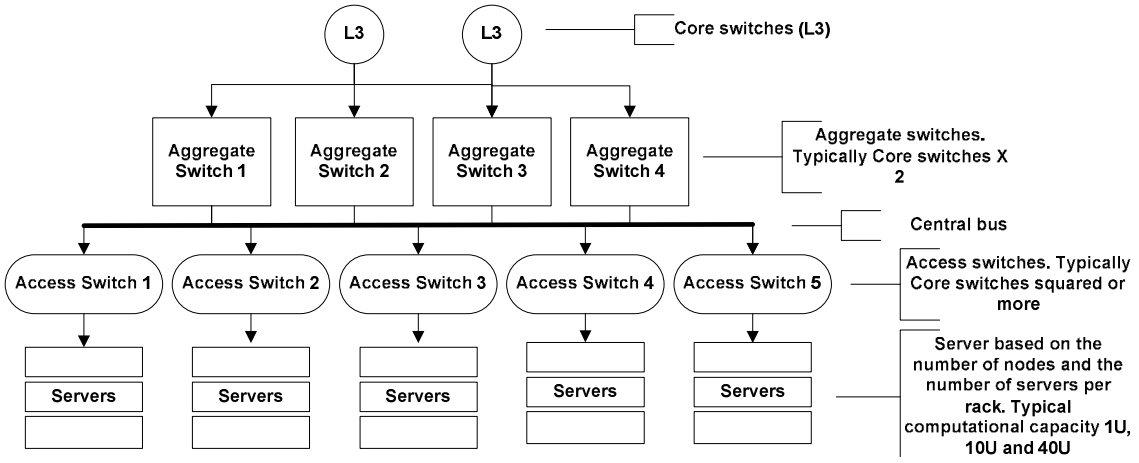

Figure 2. Typical High Speed Multi-Tier Data center Model

3.4.2 Data center expansion

To keep up with the changes in demand, the newer data center growth strategies include one of two things: Physically increasing network switches and servers (horizontal expansion) or increasing the use of the existing physical servers by virtualizing them (vertical expansion) [24]. Some of the newer data center virtualization approaches include the principle of cloud computing. However, it is important to note that the limiting factor for any virtualization is still the amount of physical resources available. Therefore, the scope of this research was to introduce effective physical resource scaling approach by leveraging decision network.

3.4.3 Data center costs, risk, and designs

Lowering data center cost has been of interest for all who are familiar its expenses. Majority of the data center costs come from data center electricity and routine

maintenance. If the data center capacity is not planned effectively to begin with then the cost to upkeep the data center can be even higher. Some of the other causes of high data center costs come from floor space square footage cost, cooling costs, and costs related to inefficiencies in power distribution.

If the data center capacity is not planned effectively to begin with then the cost to upkeep the data center can be even higher than originally anticipated. Over time as the complexities in the data centers increase so do the costs. Network switches and servers roughly consume the same amount of energy when underutilized as they do when they run at their full capacity [9]**.** Hence, running them at a low capacity makes them extremely inefficient. Therefore, planning a data center at its maximum capacity regardless of the demand makes the design sub-optimal.

Data center development plans are usually risk averse. That is, engineers plan for the maximum capacity that they can anticipate to accommodate for any spike in the network or server demands. Since the network switches and the servers get charged the same for electricity whether they run at a full capacity or low, running them at a low capacity makes them extremely cost inefficient. Hence, planning a data center at its maximum capacity regardless of the demand makes it a sub-optimal design. To accommodate for this some of the traditional designs have employed the principle of dynamic shutdown in which the servers that are not being used left in a suspend mode consuming less electricity.

To keep up with the changes in demand the newer data center growth strategy includes one of the two things: Physically increasing network switches and servers (horizontal expansion) or increasing the use of the existing physical servers by virtualizing it (vertical

expansion). Hence, the limiting factor for any virtualization is still the amount of physical resource available.

As always, there should be a failover plan designed as part of any data center design for natural disasters, planned transition during upgrades, or transition due to unforeseen fault, failures, or high volumes. The failover data center is usually underutilized. The failover data center that accompanies the design methodology proposed in this paper will be utilized when there is an unexpected high volume of traffic that the primary design is unable to handle.

3.4.4 Types of data center demands

There are two primary types of demands in any data center: network demand and computational demand. Network heavy data centers are designed to exchange high bandwidth of information and may require minimal amount of computational capacity. Examples of such scenario include content and media driven websites in which the network bandwidth is the primary concern. Even though the requests are not computationally heavy, they need to be processed on large volumes and at fast rate. Computationally heavy data centers could include low bandwidth of network flow but are designed to address high computational demands. Examples of such scenario include scientific research labs and medical data centers. There are also data centers that require both high network bandwidth and high computational capacity. An example of such data center is the financial investment and trading systems.

3.4.5 Data center decision

Data center decisions are influenced by the employees all levels in an organization. System Engineers setup servers and network. They are closest to the actual systems and understand the seasonal fluctuations in data center demand. Software engineers recommend system changes based on the software they have written that is about to be elevated to production. The system and network recommendations are made based on binary heap analysis, application response time, software's cyclomatic complexities, tangles, lines of codes, and software fats [20]. Business team influences the data center decisions through their business strategies. New business initiatives can anticipate new clients and hence new data center demands. IT architects can influence the decision based on data center cost around non-functional requirements such as data center cooling, power distribution, and floor space. Lastly, the IT management influences the data center decisions from the point of view of costs, labor, and new technology upgrades.

3.4.6 Data center governance

We address two different types of data center decision choices. Institutions either make routine changes to their data centers or leave them alone. In their cases the capacity planning plays a big role. We address both of these approaches in our research. The first approach uses decision network to constantly assess which state to move to so that each data center alteration brings the best overall result while managing risk. The second approach takes an existing overgrown data center and applies causal Bayesian belief network to assess which final state to institute.

3.4.7 Current data center limitations

Presently there are no methodologies in place that allow for dynamic data center design. That is, no methodologies allow the data center to be dynamically updated while addressing all risks in the decision process to lower costs. The most expensive aspect of any data center are the non-functional parameters such as cooling, space, and power distribution [11]. However, the non-functional requirements are directly depended on the functional requirements such as servers and network. The methodologies either address estimating server and network bandwidth for one time setup or employ dynamic shutdowns of servers to lower costs. In other cases, the methodologies address design and/or governance of non-functional requirements. Hence, there has been a strong need for a methodology that can allow the data center managers to govern functional parameters to lower cost of non-functional parameters while addressing all risks in the design.

3.5 Data center states

There are two different types of capacity demands in a data center. First is server capacity and second is network capacity. It could be argued that there are many other levels of demand but the counter argument would result in the fact that at the system level there are only two levels of capacities that are needed to fulfill a transactional request – server and network capacities. The demand of server capacity is very different from network capacity. The network demand can be directly correlated to the network traffic received whereas the demand of the server capacity is evaluated based on computational capacity of Million Instructions Per Second (MIPS), which is a

standardized way of measuring large computational capacities [32]. Server capacities produce a trend and experience less variance than network demand [32]. High bandwidth could use low computational capacity and similarly low bandwidth demand could be computationally heavy. Therefore, computational forecasting is an independent entity. Because network demand is directly correlated to the traffic, network demand can be constant, trending, or seasonally fluctuating [29].

Computational demand is estimated using the Holt's method. Each stage of the system also provides a buffer zone. That is, servers are designed to run at less than their 100% capacity continuously. Typical servers are designed to run at 64.4% of max capacity so that they have the bandwidth to accommodate computational peaks [16]. Nevertheless, even when the servers welcome high computational demands, the institutions are accustomed to run servers at 64.4% max capacity. The buffer zone built in the design could accommodate for at least two standard deviations from the maximum value. Since each environment is unique, the system construction will therefore require an iterative process to determine the ideal construction of each state. Since the design structure is risk averse, small demand peak situations are expected to be absorbed by the design. For unexpected high demand peaks, the traffic would be transferred to the backup data centers, which are usually underutilized.

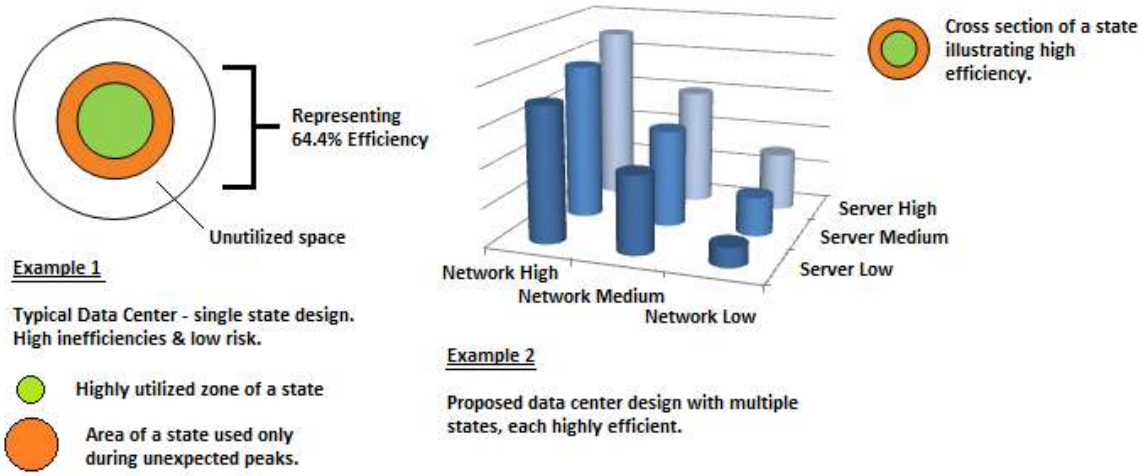

Figure 3. Server & Network State Efficiencies

Data center server and network demands can vary by type and functionality. The two types of demands are internal demands (server to server) and external demands (business to business or client to business) [30].

The demand functionalities at the detail level could include batch jobs, internal database calls, content deployment, rules engines, file nets, etc. For the purpose of simplicity we will simply categorize demand as server demand and network demand. Furthermore, we have characterized data center demand at three levels, high, medium, and low, for server and network capacities. Based on the data center capacity and design, these levels could equate to any quantitative measure. That is, low server for one institution could mean 4 servers while for another it could mean 40 servers. The principle behind characterization of capacity however remains the same. Therefore, any given data center could be broken down to nine separate states. The states could be further divided into more states depending on the institutions and their designs.

For the purpose of developing a data center capacity reduction plan, we took the initial data center state and marked it as green. Institutions that are interested in making their data centers lean view their data centers as server and network heavy. We made the assumption here that both – servers and network were heavy in capacity. The current data center state (showed in green) was theoretically broken down to different capacity states as shown with orange states. The purpose of this step was to clearly define the end state which could result through cost effective measures after running analysis on the decision factors on the overall design. Since the purpose of this approach was to reduce data center capacity, we wanted to pick states lower in capacity than the current data center state so that we could remove the calculated waste that could've possibly resulted from excessive capacity planning.

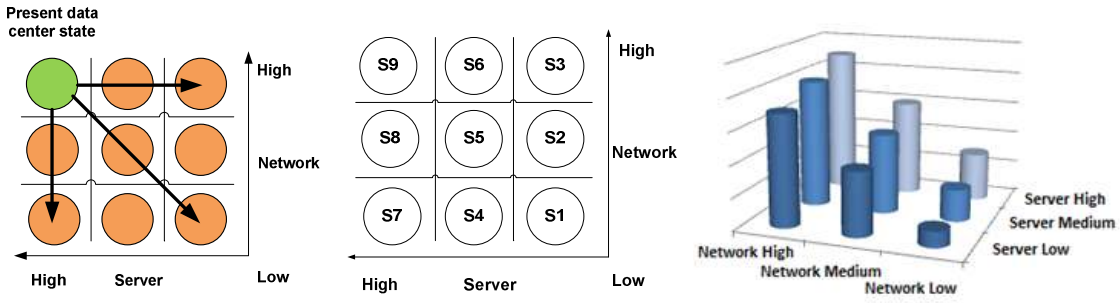

Figure 4: Setting up smaller data center states for the study from one large (current) state

One of the greatest challenges in any data center design is being able to assess most accurate capacity levels for server and network. Since most institutions still carryout data center capacity planning once or twice during the entire life cycle, there is high opportunity to clean up waste in such designs. Re-coordinating and re-performing data center capacity planning could be expensive and pointless since the cost of performing

such task can easily outweigh its benefits. Data center maturity and growth allows the institutions to understand how the data center has evolved and what factors have led to its development. By using the approach mentioned in this paper we can address the questions such as which decision factors do we attribute to data center's growth and how do we track it going forward?

3.6 Costs

One of the most important aspects of any engineering system is the cost associated with designing, developing, and most importantly maintaining the system. The data center in proposed model is represented as a single state that includes a specific level of server capacity and network bandwidth. Costs associated between and within the states are represented by arcs. The arc looping back to the same state represents the state's ongoing operational cost. The weight of the arc from one state to another is defined by its transition cost calculated with respect to the amount of work needed to transition from one state to another.

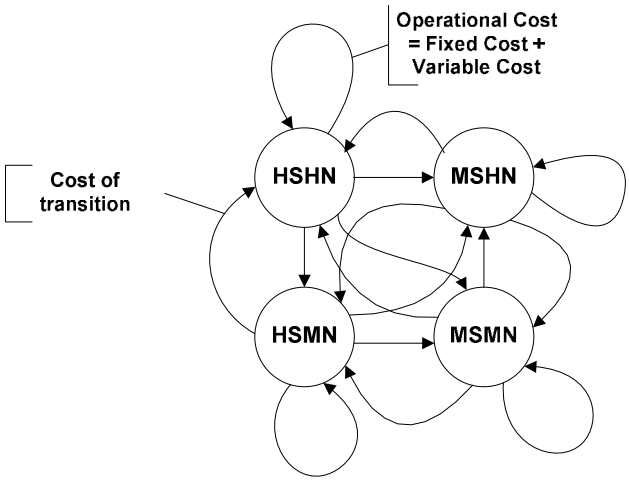

Figure 5. System states and costs

Where: **HS** is high capacity server, **MS** is medium capacity server, **HN** is high bandwidth network, **MN** is medium bandwidth network.

3.6.1 Operational Cost

Operational cost can vary based on the scale of the data center. The cost includes fixed cost and variable cost at each state in the system. During our research we focused on the variable cost since that plays a factor in the decision process. In this model we have presented a way to categorize data center scales and associated costs in an attempt to understand the relationship between computational and network demands and their costs.

State (Qualitative)	Operating cost	State(Translated)
LS,MN; LS,LN; MS,LN	Low	S4, S1, S2
LS,HN; MS,MN; HS,LN	Medium	S7, S5, S3
MS,HN; HS,HN; HS,MN	High	S8, S9, S6

Table 2. Operational cost

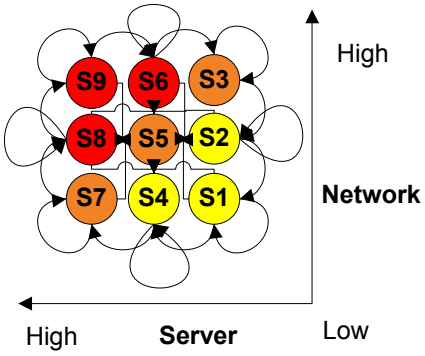

Figure 6. Data Center States

Where:

LS = Low Sever State, **MS** = Medium Server State, **HS** = High Server State

LN = Low Network State, **MN** = Medium Network State, **HN** = High Network State

Note: For visual clarity, not all arcs are displayed in figure 7.

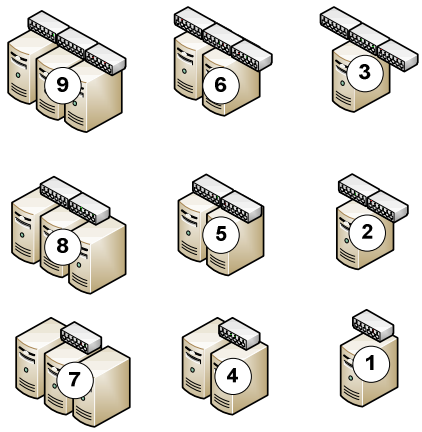

Figure 7: Visual Illustration of the Data Center States

3.6.2 Transition cost

Transition cost can vary based on two criteria. First criterion is whether the data center is being upgraded, scaled up, or scaled down. A second criterion is the capacity and bandwidth difference between current and future state. Greater the difference, more it will cost to transition the states. There are inherent risks involved when scaling down a data center. Since system availability is one of the highest priorities of all data centers, scaling down servers or network may mean that institutions may not be prepared to accept unexpected high volumes of demand. However, using strong forecasting models

and probability approach through decision network, we are able to not only minimize the risks but also address uncertainties in the decision process.

There are two scenarios in which a data center upgrade is expected. First scenario includes technology advancement that results in institutions upgrading and redefining all data center states. Second, if the institution realizes that they will never return to some of the lower states, redefining of the states will make the model not only more realistic but also usable. The other scenarios of the system change can include scaling up or scaling down the system. The cost of transition is based on the number of changes the system needs to carry out to make a successful transition between states.

In table 3, the blue boxes represent the state to which the system wants to transition during scale down. Yellow, orange, and green states represent computationally and/or network bandwidth higher than the blue state. White boxes are either lower in server capacity or network bandwidth when compared to the blue box. Hence, during scale down white states will not be applicable in the decision process. For the purpose of our experiment, yellow boxes represent 1 unit of work needed to get to the blue state. Orange boxes represent 2 units of work needed to get to the blue state. Finally, green boxes represent 3 or more units of work needed to get to the blue state. Unit of work is a measure of effort needed to alter one data center state to represent another with minimal change. The change includes reducing server capacity from high to medium or medium to low. Similarly the change can also include altering network bandwidth from high to medium or medium to low. These weights however could be replaced with actual cost values.

State 9

HS,HN	MS,HN	LS,HN
HS,MN	MS,MN	LS,MN
HS,LN	MS,LN	LS,LN

State 6

HS,HN	MS,HN	LS,HN
HS,MN	MS,MN	LS,MN
HS,LN	MS,LN	LS,LN

State 3

HS,HN	MS,HN	LS,HN
HS,MN	MS,MN	LS,MN
HS,LN	MS,LN	LS,LN

State 8

HS,HN	MS,HN	LS,HN
HS,MN	MS,MN	LS,MN
HS,LN	MS,LN	LS,LN

State 5

HS,HN	MS,HN	LS,HN
HS,MN	MS,MN	LS,MN
HS,LN	MS,LN	LS,LN

State 2

HS,HN	MS,HN	LS,HN
HS,MN	MS,MN	LS,MN
HS,LN	MS,LN	LS,LN

State 7

HS,HN	MS,HN	LS,HN
HS,MN	MS,MN	LS,MN
HS,LN	MS,LN	LS,LN

State 4

HS,HN	MS,HN	LS,HN
HS,MN	MS,MN	LS,MN
HS,LN	MS,LN	LS,LN

State 1

HS,HN	MS,HN	LS,HN
HS,MN	MS,MN	LS,MN
HS,LN	MS,LN	LS,LN

Table 3. State transition cost

HS: High computational capacity

MS: Medium computational capacity

LS: Low computational capacity

HN: High network bandwidth

MN: Medium network bandwidth

LN: Low network bandwidth

3.7 Risk

Risk is an extremely important parameter in data center design and operation. There are different types and levels of risks in all data center designs. In every design, it is of utmost importance that the risk of compromising information security and data integrity

are maintained at all times. Based on our combined knowledge on data centers, we know that security risk is handled at the infrastructure level. That is within first firewall, de-militarized zone, second firewall, and finally the load balancer. Hence since our design strategy is around the capacity, which is beyond the load balancer, and since we are not proposing to change the security infrastructure itself, we propose to keep the security risk as a constant in the design. Therefore, we don't mention security risk when talking about data center capacity rather we will focus on risk of downtime in data center availability.

There is extremely heavy dependency on the data center due to the high demand of information availability. Doctors need patient information at their fingertips. Investors need market information real time to be able to make better investment decisions. Social networks demand fast information streaming. Hence, there is no doubt that the availability of the data center acts as a heartbeat for the institution. Therefore, the decisions that drive the data center demand need to be extremely well thought out and can't threaten the data center availability in any context [5][6]. Such designs are said to be risk averse.

One of the major decision factors in the decision utility is whether the data center state being considered it self introduces any risk in the design. For example consider the table below. Each "home" state is marked in blue. The risk factors in the decision are considered as follows:

1. If the decision is to move up in the design then it is least risky, which in this case is moving from red to blue.

2. If the decision is to move from down from green to blue then it is less risky.
3. If the decision is to move from green to red then it is most risky.

Please note that the decision can also be to remain at the same state and the decision goes through the same evaluation as some of the rest of them.

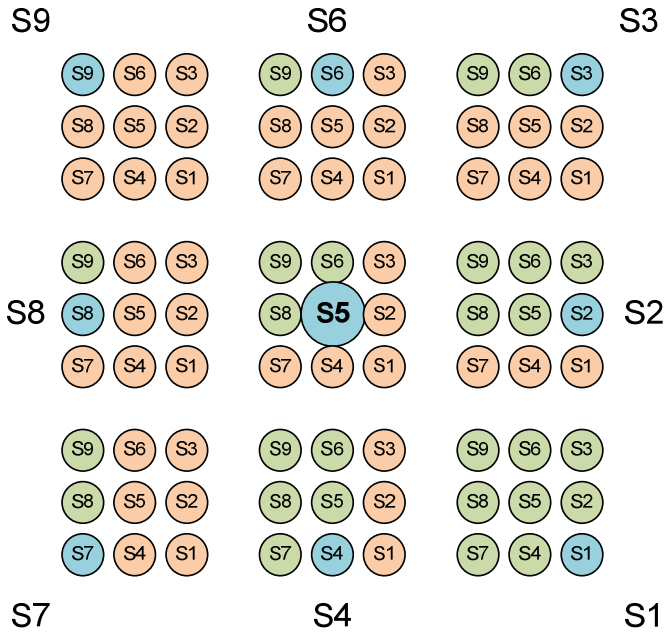

Figure 8: Risk assessment table

The grid above provides a gauge on how the risk is assessed. The states that are marked red are done so because they provide lower support when compared to the blue state in server capacity, network bandwidth, or both. Green states are least risky because they provide more support than the blue states in server capacity, network bandwidth, or both.

3.8 State transition

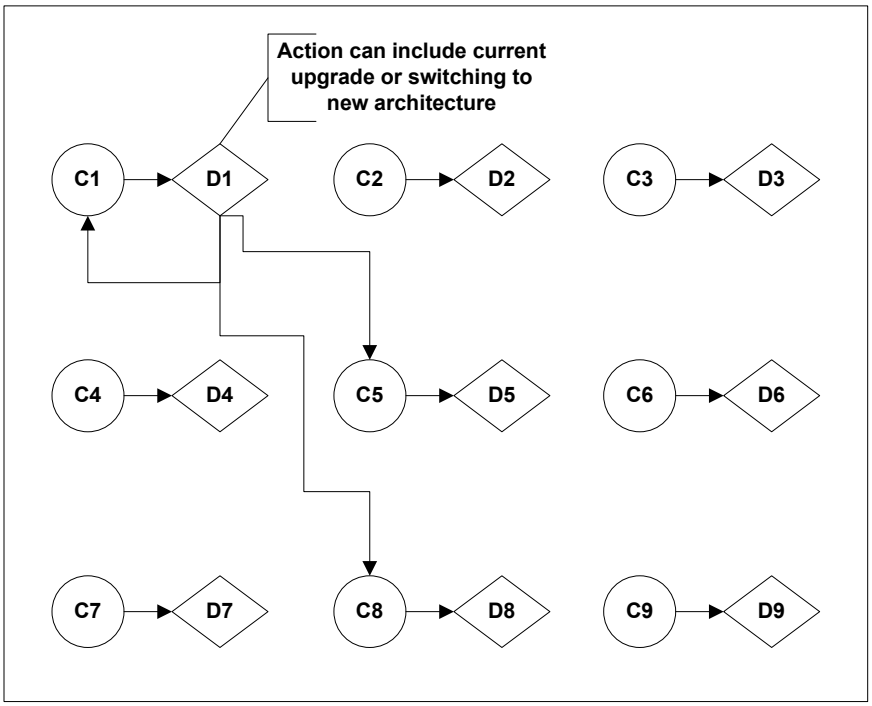

Figure 9: Chance and decision nodes

Above is an example of how the state transition would occur in the decision network. Each state consists of change node (shown as a circle) and the decision node (shown as a diamond box) above. In order to keep the decision network states consistent with data center states, nine states are shown. However, there is no limit to how many states are drafted in the design. The decision node determines which next state to select based on the input from the current chance nodes. The decision then feeds the output as input for the next chance nodes. The decision node can determine that a state transition should not occur. This action is showed in the arc looping from decision 1 (D1) to change 1 (C1).

3.9 Decision model

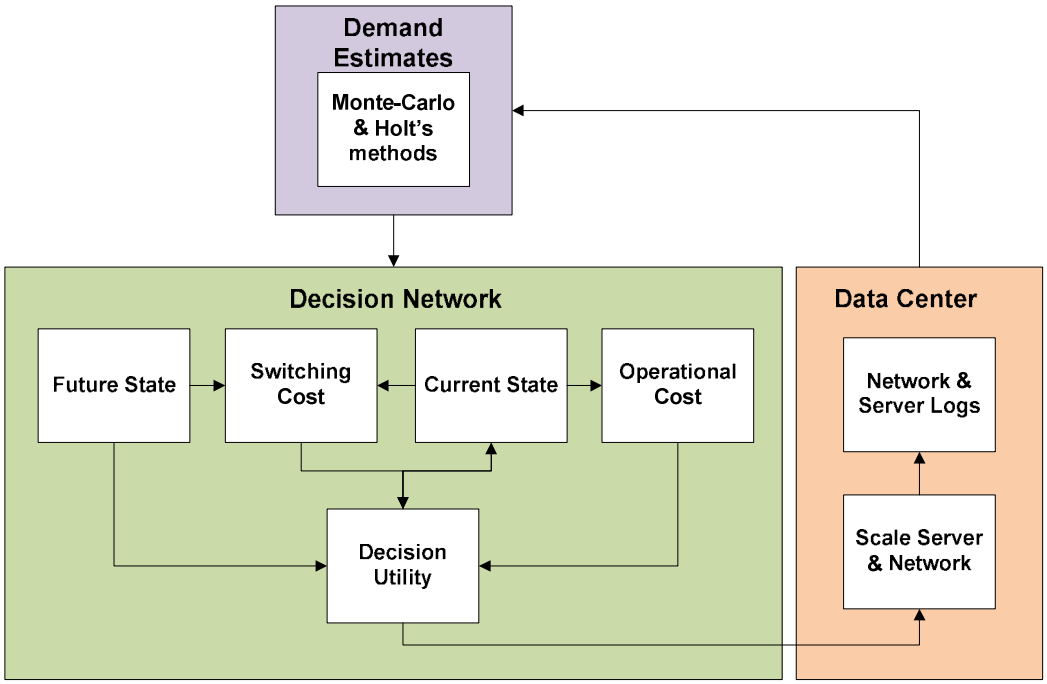

Figure 10: Proposed Decision Model

The objective of the systems engineering model in figure 1 is to control (& lower) the variable cost of data center design and maintenance while managing risks and uncertainties in data center management decision process. The effectiveness of the system is measured through cost savings that result from scaling down servers and network switches. The model is tested with numerous simulations. The risk criteria of the model is measured by calculating the number of times the model underestimated server and/or network demands, forcing the demand to be directed to the backup data center. Furthermore, once the model was setup and the probability values for the decision nodes were trained using simulation data, the model was tested against a demand pattern.

In the proposed approach the system consists of the demand estimation methods, the data center architecture, and the decision network to govern the data center changes. The elements of the data center consist of servers, network access switches, aggregate switches, and core switches as shown in figure 3. These elements are part of a typical data center design [1]. The output of the decenter elements include server and network demand logs. The governing body of the system is the decision network. The elements of decision network at minimum are current data center state, calculated future data center state, current and future operational costs, data center state transition (modifying) cost, decision node, and the decision utility. Decisions are suggested based the server and network demand estimates presented through Monte-Carlo approach for network and Holt's method for server capacity. The model also assumes that the decision factors and the decision rules stay constant throughout the data center life cycle.

Although there are several factors that contribute to the total cost of data center maintenance, there is a direct correlation between functional (server and network) demand and non-functional (heating, cooling, etc.) demands [16]. This correlation could be monitored through the overall amount of energy consumed by the data center. In some data centers the non-functional costs could be fixed cost, for e.g. buying the entire data center floor space regardless of the actual amount used. In other cases the non-functional costs could be variable costs, for e.g. renting only the area of data center floor space that is being utilized.

Since the variable cost of non-functional elements is directly depended on the functional elements (number of servers and network switches) [9], the cost of functional elements was closely monitored by observing the energy consumed by the data center during the simulation. Since we didn't have access to the actual data center that could be

constantly scaled for our experiment, the total cost of energy consumption for data center was simulated through GreenCloud.

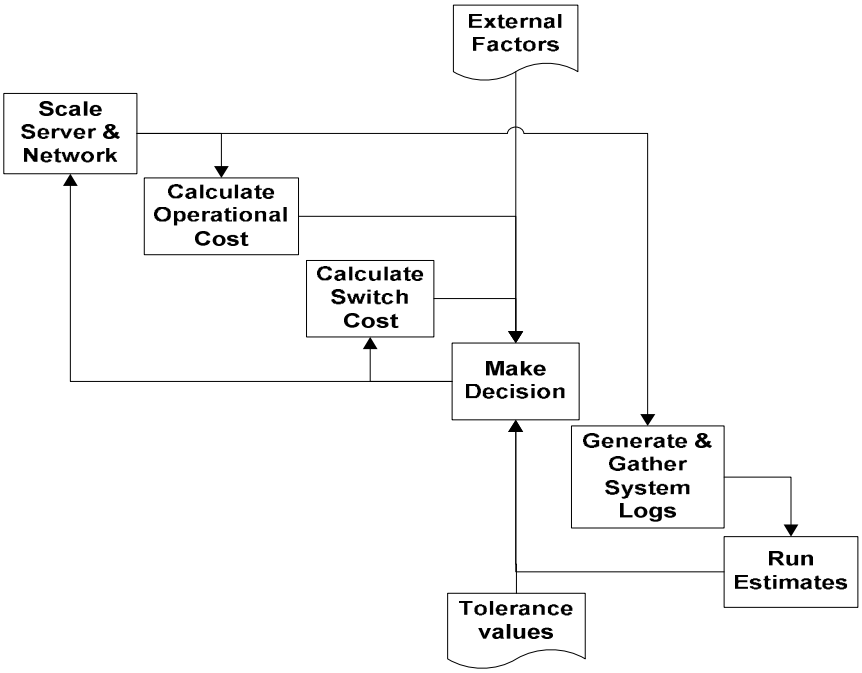

Figure 11. N-Squared chart of the decision network operating model

The decision model uses certain key factors and decision rules in the decision making process. Once the decision has been made the accepted decision updates the very factors that it used in making the decision at the first place. Hence, the process prepares the system for its next decision. In our experiment the decisions were made on monthly server and network simulated logs. Figure 2 shows an illustrative high level operating model N-Squared chart of the decision network functional interactions.

Working from top down, first function is scaling server and network capacities. The overall system conceptually starts with a basic number of servers and network switches. This function is executed again once the decision has been made on what the capacity planning should be at each decision point. Second function is calculating the operational

cost. Calculating operational cost requires input of servers and network capacities. Since fixed cost don't vary and the variable cost of non-functional areas such as cooling are directly relative to the functional areas, only server and network costs are measured. Third function is calculating cost of transitioning data center states, which also uses server and network capacities as its input. Fourth function is making the decision. Decision function uses operational and transition costs, server and network capacities, anticipated server and network demand estimates, and cost tolerance values pre-determined by the institution managing the data center. In some cases other external factors important to the institution could be used in the decision making process, either added as a decision factor or applied as a decision rule. Fifth function is generating and gathering server and network logs. Sixth and final function is calculating future capacity estimates using the server and network logs.

3.10 Demand estimates

The decision network uses at minimum server and network usage/demand estimates to determine its next state while balancing other decision factors. Hence, estimating server and network demand plays a key role in developing an accurate decision model. Network bandwidth estimates were made using the Monte-Carlo method. Monte-Carlo method provides a strong base for probabilistic network model [27]. Hence, it has been widely used in Bayesian Belief Network and Decision Networks [18][28][25][13][31]. Monte-Carlo could've been used for both server and network estimates. However since the server capacity growth has a trend, Holt's method is an ideal forecasting method. Unlike network demands, server demands tend to stay constant and grow in demand [1]. Therefore, server capacity estimates were done using the Holt's method.

3.10.1 Monte Carlo method

```
sigma(1) = 1; %Initialize
T = 100;
a0 = .5;
a1 = .7;
b1 = .3;
for t = 1:1:T
    x(t) = sigma(t)*randn(1,1);
    sigma(t+1) = sqrt(a0+b1*sigma(t)^2+a1*x(t)^2);
end
x(1:floor(T/10))=[];
x = x./std(x); %Normalize x
```

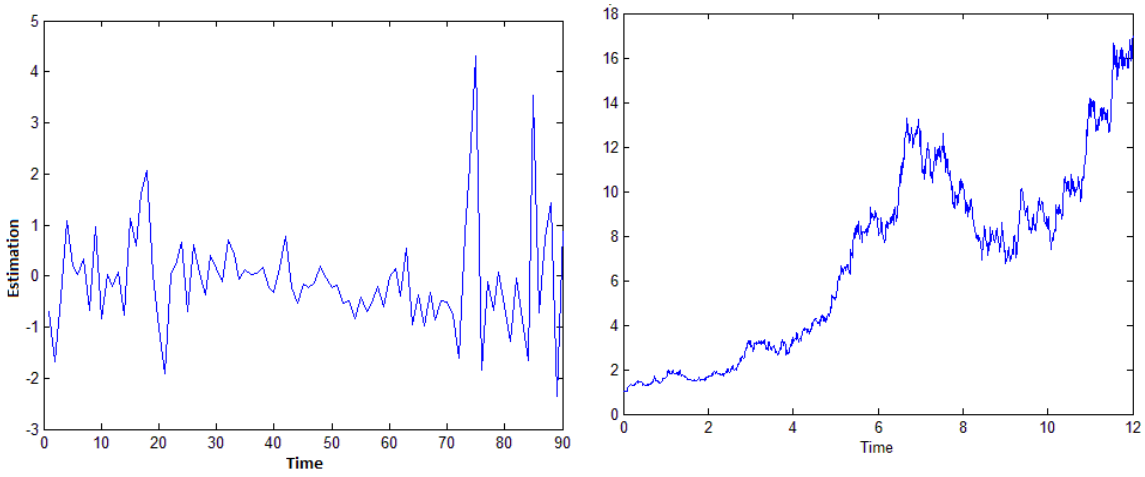

Figure 12: Demand estimates using Monte-Carlo

3.10.2 Holt's method

Holt's method is an extension of exponential smoothing and is designed for data with trends and presents stronger results than linear regression and Brown's linear exponential smoothing method [32]. Existing research done using these two methods for server capacity estimates and network demand under probabilistic model gave us sufficient reasoning and justification to use these methods for our research. To keep the scope of the paper on the decision network itself, these theories are not discussed in detail. However, the references for these methods will provide enough justification for their user in our research over other methods. Both methods, Monte-Carlo and Holt's, have been tested using real data in the research mentioned in the references. Below is a table of values generated through Holt's method.

Time Period	MIPS x10^3	Level	Trend	Forecast		
1	**95200**	95200.0	0.0	**--**		
2	**90800**	90800.0	-4,400.0	**95200.0**		
3	**151600**	151600.0	60,800.0	**86400.0**	**alpha =**	**1.000**
4	**249600**	249600.0	98,000.0	**212400.0**	**beta =**	**1.000**
5	**257800**	257800.0	8,200.0	**347600.0**	**MSE =**	**11190105714**
6	**502200**	502200.0	244,400.0	**266000.0**		
7	**713500**	713500.0	211,300.0	**746600.0**		
8	**836900**	836900.0	123,400.0	**924800.0**		

Table 4: Demand estimates using Holt's method

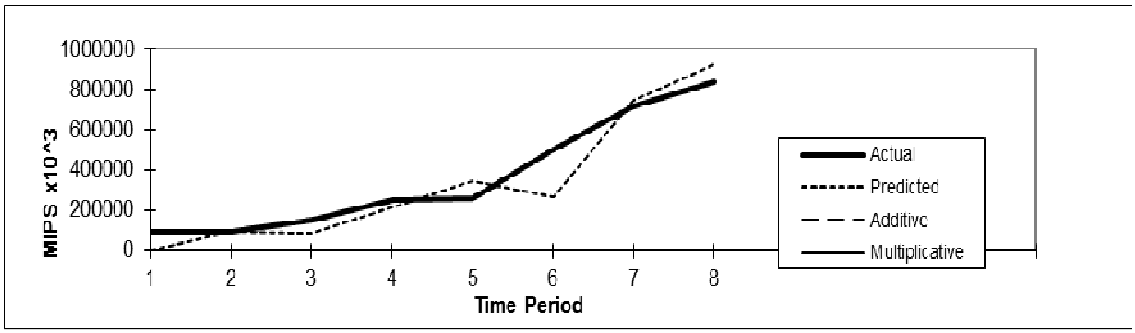

Figure 13: Demand estimates using Holt's

After generating the values in the table the demand estimates were forecasted to study and anticipate the growth in the server capacity over a length of time.

3.11 Causal Bayesian Belif Network

Bayesian Belief Network (BBN) is a collection of mutually exclusive factors represented as nodes influencing other nodes in the network. Each node could have several variables that represent the particular state. The probabilistic approach to inference based network relies on prior knowledge of the system and observed probabilistic values for each variable in the nodes. The network is constructed on the principles of Bayes Theorem. Below, the probability of factor A is conditional on the given information for B. Hence, the conditional probability of A given B is probability of A times probability of B given A, all divided by the probability of B.

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (1)$$

Equation 1 shows the essence of Bayes theorem between two variables. However, it can accommodate multiple factors. Each additional decision factor in BBN contributes at an exponential rate of decision possibilities. Hence, the institutions should account for computing power when developing a BBN of an incredibly large scale. The number of possible directed acyclic graphs/possible decisions for nodes *n* in BBN is n^n [17].

3.12 Decision network

Decision network is a directed acyclic graph representing conditional probability distribution. The conditional probabilities are propagated throughout the network. Decision network therefore, is able to perform under high uncertainty. Equation 2 below shows that each variable is conditionally independent of all other nodes in the network [19].

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i | x_{i+1}, ..., x_n) = \prod_{i=1}^{n} P(x_i | Pa(x_i)) \quad (2)$$

Decision networks are constructed of chance nodes illustrated by brown boxes in figure 14. Chance nodes represent probability of a particular variable. The probability values are derived from data center's historic data if available. If the data center is new the probability values are equally weighted to begin with. As the architecture mature so will the model allowing for more accurate probability values. The arcs represent influence of one variable on the other in a system. There can be one-to-many, many-to-one, and

many-to-many relationships [10]. The chance nodes use influence of one node on another to calculate the probability of each node. The underlying principle of calculating probabilities uses Bayes theorem of conditional probabilities. Bayes theorem calculates probabilities of events within a node by using the information from the influencing nodes.

The blue box represents the decision node. The decision node represents all decisions possible in a system given different probabilities and circumstances. Finally, the diamond box is the utility node that takes user preferences into account by assigning different satisfaction weights to all possible decisions in a system. Satisfaction weights denote decision maker's level of satisfaction on each possible outcome [4]. Since the decision maker's satisfaction is based on the institutions risk tolerance and rules, the decision network consider these values in its outcome. As a result, we get a system that not only incorporates all factors that influence the decision but also subjective information from the user in a form of satisfaction decision weights. For the purpose of this research the satisfaction weights were assigned based on the utility algorithm mentioned later in the paper. The weights were in increments of 10 satisfaction units between 0 and 100. Most risky and most expensive output was assigned the satisfaction weight of 0 while least risky and least expensive outcome was assigned the satisfaction weight of 100.

3.12.1 Decision network Netica model

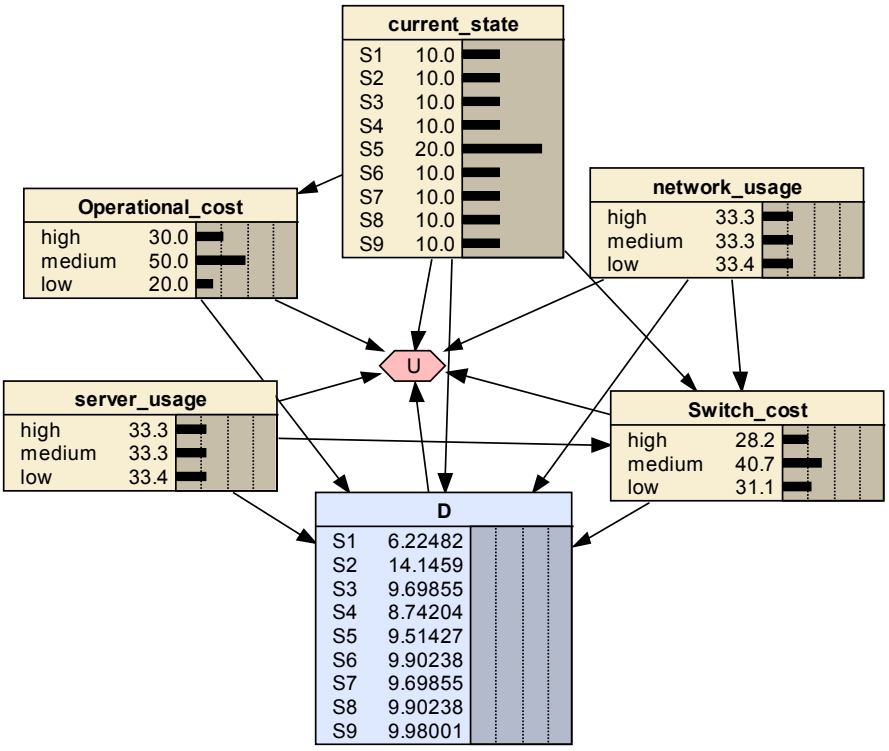

Figure 14: Minimalistic data center decision network

3.12.2 Overall Net

8	**Nodes**	(not including constants)
1	Decision nodes	
1	Utility nodes	
0	Constant nodes	

16	**Links**	(not disconnected)
0	Disconnected links	
0	Time delay links	
0	Directed cycles	(without delays)
9	Loops	(disregarding link directions)
1	Separate networks	(ignoring constants)
530	**CPT Numbers**	(total number of conditional probabilities)
729	Decision conditions	
0	**Findings nodes**	(not including constants)
0	Negative or likelihood findings	

Table 5: Overall net values for the decision network

3.12.3 Beliefs for each decision question

	S1	S2	S3	S4	S5	S6	S7	S8	S9
D	6.2248	14.146	9.6986	8.742	9.5143	9.9024	9.6986	9.9024	9.98
current_state	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.1	0.1
	high	**medium**	**low**						
Operational_cost	0.3	0.5	0.2						
Transition_cost	0.28227	0.40667	0.31106						
network_usage	0.333	0.333	0.334						
server_usage	0.333	0.333	0.334						
	Rent	**Buy**							
Space	0.5	0.5							

Table 6: Beliefs table for the decision network

Similarly a Bayesian belief network focused on three main criteria of time to market, development status, and operational costs would look something like the following:

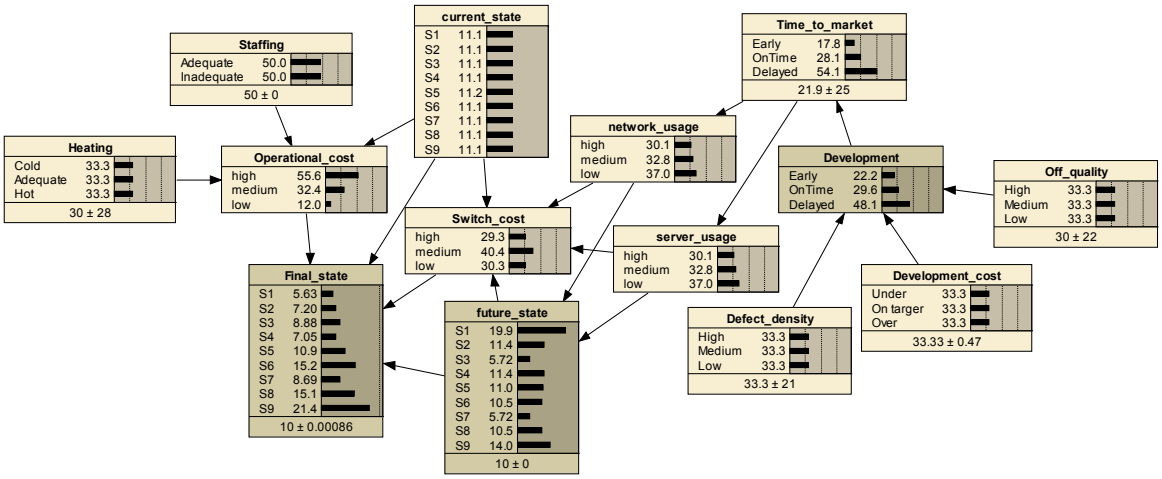

Figure 15: Causal Bayesian belief network

The decisions on data center changes are made based on the past design considerations and future goals of the organization with the information best available at the time. Companies employ their own decision processes. The table below compares the model proposed in this paper to the traditional design.

Traditional/Typical Approach	Model using Decision Network
• Non-functional requirements such as heating and cooling are addressed independently.	• Non-functional requirements such as heating and cooling are addressed through decision network.
• Design decision for capacity planning is based on maximum anticipated demand.	• Design decision for capacity planning is based on routine estimation values obtained through data center logs.
• Data center rarely rescaled unless it is scaled up to add more servers or network switches.	• Data center is regularly rescaled on a condition that the benefits of rescale outweigh the cost and rescale risk
• Less reliance on the backup data center	• More possible reliance on the back up data center

• System upgrades and maintenance are specially planned and executed	• System upgrades and maintenance are planned along with routine scaling
• High overhead due to wasted capacity space.	• Low overhead for keeping the design as close to the demand as possible.

Table 7. Traditional and Decision model approaches

3.13 Decision utility

When there are several decision factors in the play, the influence of these factors on the operational cost, transition cost, network bandwidth, and server capacity are evaluated through the decision network. Since all decisions in any architecture and institution can be correlated to cost, the influence diagrams can be expanded to any length as long as the influence of all factors on the costs are accurately determined. The flexible decision network works in a holistic manner while allowing the operational cost, transition cost, network bandwidth, and server capacity to be its primary decision parameters. The overhead of estimation and transition is therefore directly correlated to how accurately the institution is able to assess the influence of decision factors to its costs.

Each additional decision factor in the decision network contributes the number of decision possibilities at an exponential rate. Hence, caution should be practiced in developing a decision model and in choosing the number of decision factors for the model. The number of possible directed acyclic graphs in the decision network could be represented with the mathematical function for *n* nodes below [17].

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \quad \begin{matrix} n \geq 2 \\ f(0)=1 \\ f(1)=1 \end{matrix} \tag{3}$$

Decision Utility allows the user to build decision rules off of decision factors in the model with the use of satisfaction weights. Decision network generates a table with all possible outcomes of a decision. In a deterministic model, the satisfaction weights assigned to all the values in the utility table allows the user to receive a decision that is in adherence to the decision rules. Following were the decision rules adopted in the design:

Rule 1. If the servers and network switches need to be scaled up to handle anticipated large demand then ignore all cost factors. This rule allows the institutions to prevent data center outage, hence managing the expectation of data center availability.

Rule 2. If the cost of transitioning data center states while scaling down is lower than or equal to the current operational cost then support the decision to scale down the data center.

Rule 3. If there is significant cost saving in scaling down servers and/or network and it is worthwhile to pay high cost in transitioning the data center state then suggest transitioning the state.

Rules 1 and 2 are simple cases where data center transition is suggested even when just one decision factor is satisfactory. In rule 1, the main decision factor is whether the anticipated next step is to scale up the data center. In rule 2, the main decision factor is the transition cost. Rule 3, has uncertainties for which the probabilities in the decision network play a key role in the decision process. Factors such as *significant cost* and *worthwhile to pay high cost* are highly subjective. For such factors the satisfaction weights, probability values, and institution's cost/expense threshold values contribute heavily in the decision process. The term *worthwhile* in decision utility means that the

utility will compare current operational cost and future operational cost. The difference/gap between the two costs get weighed against company's risk tolerance in order to determine if the company should make the decision to transition to next state.

The system checks if the transition cost is lower than current operational cost. If the transition cost is not lower than a certain value, the system checks to see how big the difference is between future operational cost and the current operational cost. The system is allowed to transition states by spending high cost if there is significant long term savings. Figure 8 illustrates the decision concept.

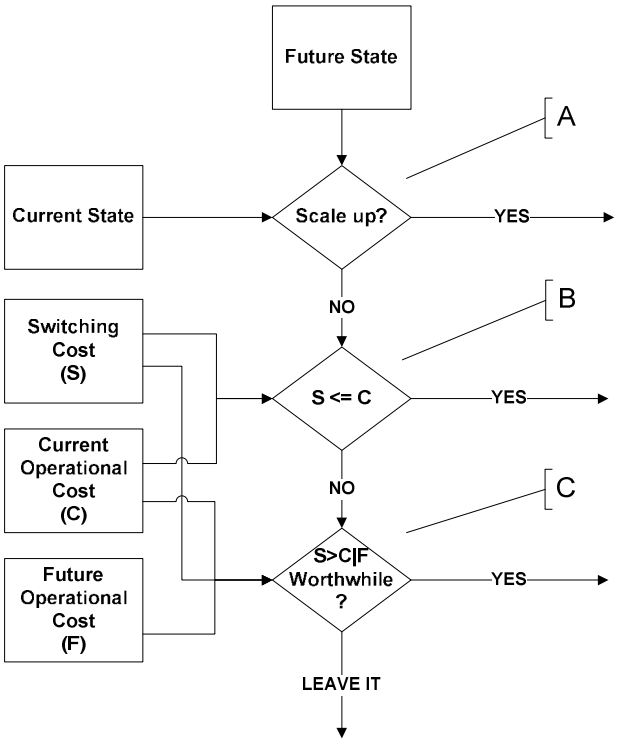

Figure 16. Utility Structure of the deterministic decision network

A. Satisfaction utility for which the model suggests to scale-up is assigned the highest value.

B. Knowing the institution's risk and cost tolerant values, since the future operational cost will be lower than current operational cost (because of down-scale), the satisfaction values of system transition are assigned 'high' if transition costs are lower than or equal to current costs.
C. We already know that future operational cost will be lower than current operational cost. The satisfaction utility values are applied in order of institution's priorities and risk tolerance. The decision to transition is satisfactory if it's worthwhile to invest money now to transition the states. It may even mean investing extra money now for long term cost savings. The gap between the current and future operational cost is weighed against institution's risk tolerance. Higher the tolerance, larger the cost difference will be. Lower risk tolerance will allow for smaller cost difference.

The decision network is most useful in data centers that experience large variability in demand since that creates more opportunities to scale down the data center. Different periodic server and network demands make the utility function work hard to determine the best data center state. If the server and network demand are constant the data center has no reason to tamper with its state unless the institution is performing upgrades. The life cycle cost savings from using decision network is relative to the institution. Since the actual cost could greatly differ between institutions, sensitivity of life cycle cost to the decision rules could therefore also greatly differ between institutions.

3.13.1 Utility values

Decision	Current	Server	Network	Operational	Transition	Utility
S4	S8	medium	low	high	high	**70**
S4	S8	medium	low	high	medium	**80**
S4	S8	medium	low	high	low	**90**
S4	S8	medium	low	medium	high	**60**
S4	S8	medium	low	medium	medium	**70**
S4	S8	medium	low	medium	low	**80**
S4	S8	medium	low	low	high	**10**
S4	S8	medium	low	low	medium	**20**
S4	S8	medium	low	low	low	**30**
S1	S2	Low	low	high	high	**70**
S1	S2	Low	low	high	medium	**80**
S1	S2	Low	low	high	low	**90**
S1	S2	Low	low	medium	high	**40**
S1	S2	Low	low	medium	medium	**50**
S1	S2	Low	low	medium	low	**90**
S1	S2	Low	low	low	high	**10**
S1	S2	Low	low	low	medium	**20**
S1	S2	Low	low	low	low	**30**

Table 8: Utility table and values for the decision network

Above is an example of the utility values assigned in the decision network. For a case when the current state of the data center is state S8 which is high server and medium

network state, the decision to move to state S4 is depended, at minimum, on the factors such as estimated server demand, network bandwidth demand, transition cost, and the future operational cost. We know that anticipated server demand is medium and network is low – which translates to state S4.

One of the rules of this design is to make the system risk averse. Hence, risky moves will be made least satisfactory in the utility node. Therefore, for every state that is lower in demand in any nature compared to the medium anticipated server demand and low network demand, the utility values are set to 0 since those states are risky for the data center.

The decision comes into play when the data center is sitting at an expensive state (operationally) and the anticipated server or network demand is lower than the current state. Now that the system knows that it is dealing with the right state medium server + low network = S4 it will try to justify its move from S8 to S4. The justifications are employed by assigning certain weights. Again, we want to save cost so it is most satisfactory to move to S4 from S8 if the current operational cost is high for some reason and transition cost is low. We know that S4 is anticipated to have low operational cost from our grid.

The transition cost from S8 to S4 would be normally medium cost because of the two hops – S8 to S5 or S7 and then from S5 or S7 to S4. However, if the user already has maintenance is planned and the costs of transition the states could be merged with the maintenance cost making it low, it will be the ideal condition for the data center manager. Hence, making it most satisfactory.

The least satisfactory decision will be if the current operational cost is already low despite the fact that the data center is sitting at state S8 which is normally is a high cost and additionally if the system would have to pay high cost to transition for some reason due to other factors then the decision to move would be least satisfactory. Another similar example is given above between state S2 and state S1.

3.14 Sensitivity analysis

Sensitivity analysis is studying entropy of the final decision to its factors. Sensitivity analysis in Bayesian belief network allows the user to determine just how much a particular node in the network is influenced by the findings at other nodes. In the concept of lean design and eliminating waste, it is crucial to know which nodes had the most influence in the final decision and which ones didn't so that critical factors could be controlled and/or monitored. Furthermore, the analysis data could provide information critical to study the effects of crucial nodes on the data center demand.

Netica allows its user to monitor variance reduction and entropy reduction among other factors. We were able to monitor variance reduction in the network as the expected reduction in the variation of the real value of node D that has d states influenced by node E that has e states. The entropy in the network determines the amount of uncertainty present in the system. The level of uncertainty in the system is measured with respect to the influence a specific node experiences against the rest of the nodes in the system. Entropy for variable Q before any findings for the nodes in the system is defined as following: [15]

$$H(Q) = -\sum_{q} P(q) log_2 P(q) \tag{4}$$

Conditional entropy of the system is the entropy of Q in relation to the variable F applied after new findings. Hence, conditional entropy is defined as following: [15]

$$H(Q|F) = \sum_{f} P(f) H(q|f) \tag{5}$$

Finally, entropy reduction I is determined as the expected reduction in mutual information in Q with respect to findings of F. Entropy reduction I is represented with the following equation. [15]

$$I = H(Q) - H(Q|F) = \sum_{q} \sum_{f} P(q, f) log_2 \frac{P(q, f)}{P(q) P(f)} \tag{6}$$

Following is a snapshot of one of our sensitivity analysis sets collected at the decision stage. As the probability values matured so did the relationship between the nodes. As a result we were able to monitor the final state of the design closely and were able to narrow down the ideal states in the design.

Sensitivity of 'Final_state' to a finding at another node:

Node	Mutual Info	Percent	Variance of Beliefs
future_state	1.25416	41.2	0.2232459
current_state	0.82736	27.2	0.1124167
Switch_cost	0.11802	3.88	0.0019265
Staffing	0.00005	0.00172	0.0000006
Heating	0.00025	0.00813	0.0000014
Operational_cost	0.01748	0.575	0.0004276
Final_state	3.04220	100	0.7559220
Defect_density	0.06330	2.08	0.0014391
Development_cost	0.00383	0.126	0.0000941
Off_quality	0.02590	0.851	0.0006961
Development	0.13448	4.42	0.0041061
Time_to_market	0.16903	5.56	0.0050200
network_usage	0.49525	16.3	0.0221384
server_usage	0.49444	16.3	0.0221864

Table 9: Entropy reduction values of the BBN proposed above

3.15 Influence analysis

Influence analysis can allow us to study the best and the worst case values of decision nodes. This allows the decision makers to assess the risk factors in the system when the decision parameters sway to extreme directions due to unforeseen circumstances. In influence analysis, the resulting scenarios are compared to normative model to understand the degree of change from the initial state. While performing influence analysis and sensitivity analysis, it is important to note that when the individual nodes are changed the sensitivity values (entropy reduction or variance) result to zero.

Using the data from the sensitivity and influence analysis we see that server and network usage has the most impact on the decision of future state. This information may seem trivial to anyone who is familiar with data center demands and strategies. However, in fact the influence analysis information shows strong confidence in the BBN

methodology by generating the same results as its predecessor methodologies. The BBN methodology shows that it can show similar results for all nodes and decision factors.

3.16 Developing the model

There are few basic steps in developing and instituting decision model proposed in this paper. Like any design, successful models and decisions are based on how well the institution knows the technology that it houses, its incremental capacity, and its related costs. Following are the stages for developing a decision model that can fit any data center environment. It is important to note that there can be many levels of qualitative and quantitative measures involved in a decision process. Based on the institution and its needs, the design can very well have a healthy range of decisions factors and decision rules at any given time from which the system decides which state to recommend.

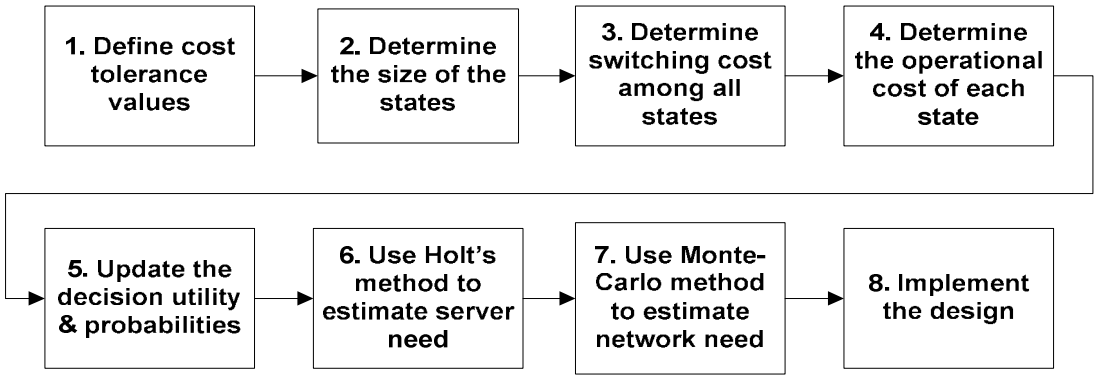

Figure 17. Steps in the decision model design & system inception

Each institution must take time to uncover its data center design decision making factors. At minimum there will be two factors, the server and the network, since these two factors together make up the data center itself. Other factors could include floor space (rent v/s buy), heating and cooling, frequency of updates, etc.

1. Each institution has its own cost tolerance values. This value reflects how liberal or conservative the institution is around each element of IT. To define the tolerance values the institutions first need to determine the fixed costs and variable costs of the system. The tolerance values help determine how much cost the institution believes it should absorb if necessary during a state transition. If the tolerance value is high then the model is more likely to recommend fewer state transitions. Similarly if the tolerance value is low, the model would result in more state transitions in order to save cost.
2. Determine computational units of work
    - In a distributed environment the minimum increment will be one server for computational capacity and one network switch for bandwidth capacity
    - The decision for the unit of work will directly reflect the institution's tolerance for risk. The smaller the unit of work, the riskier the decisions whereas the larger the unit of work, less risk tolerant is the institution. It is also important to note that making too many transitions may also be expensive depending on the institution's operational cost and transition cost.
    - The idea is also to maximize the effort at each state. That is, determining states where the server and network capacities get the opportunity to run at their maximum potential.

3. Determine transition cost of the system based on the fixed and the variable cost at each state. Larger the difference between each state, more savings will be experienced long term.
4. Determine operational cost of network and server for each independent state. At each state it takes certain fixed cost and the variable cost to run the system. Since we want to pick the states in which the computation and bandwidth capacities are maximized, there will be little variability in the variable cost. Hence, the variable cost will become fixed cost at that particular state.
5. Update the decision network and the probability values. However, using the qualitative values such as high, medium, and low instead of actual quantitative values will allow the design to be more flexible. As the architecture grows, the qualitative states can be reassigned to new quantitative values, maintaining the integrity of the framework. This also allows us to maintain the decision network conceptually and practically without many alterations. However, institutions may very well choose to use quantitative values. The weights in the utility node are predefined by the decision makers. Once decision makers consider institution's priorities and asses risk tolerance for decision factors, the utility nodes are assigned different weights by the decision makers.
6. Use Holt's method to estimate the future computational capacity. Use the existing system logs to determine a pattern with low variance which can help result in stronger decisions. Holt's method is an exponential smoothing method applied to discrete events. The Holt's method has the smallest standard deviation value of residue, making it a strong technique for server capacity forecasting [14].

7. Use Monte-Carlo method to estimate future network bandwidth. Use the existing network logs to determine a low variant pattern for stronger decisions.
8. Implement the design.

3.16.1 Building causal Bayesian belief network

We applied an algorithmic approach to monitor Bayesian network, factors, decisions, and sensitivity factors to determine what factors influenced the change in the data center at each decision stage. Constructing and monitoring a BBN is a highly iterative process. Before BBN is constructed it is essential to explore and capture the causality and the domain knowledge affecting the system [12]. Building of BBN starts with defining variables in each node representing the domain knowledge while the nodes themselves present the causality. We tested the BBN using historical data to assess if the model was ready. When the model was not ready we re-evaluated the variables and the model construction. Once the model was ready we applied an algorithmic approach to run it.

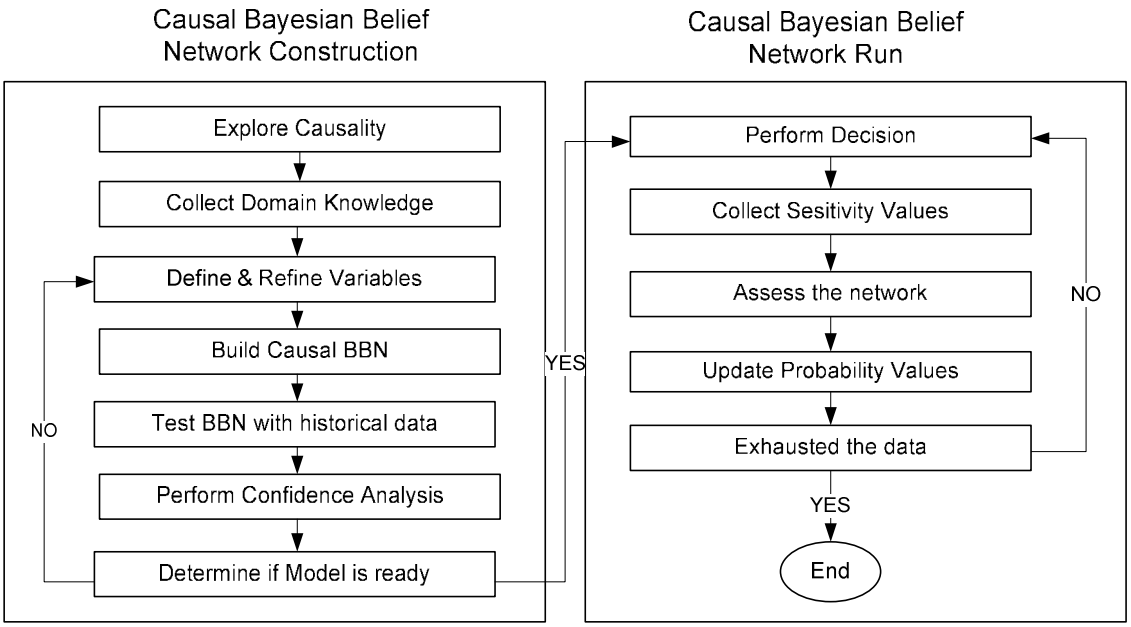

Figure 18: Model construction steps and the algorithm used to govern the BBN

The algorithm allows us to monitor the decisions and collect sensitivity values as well. The algorithm starts with performing the decision best suited by the BBN and the probability values. It then collects the sensitivity values and asses the entire BBN. The probability values of the decision factors are updated where necessary and institution's data or simulation data is checked for next set of values. The algorithm processes a full cycle of decision making in BBN. It collects sensitivity values to further accurately update the influence analysis, assesses the entire network during the cycle to reflect correct data center state, and updates the probability values before moving on to the next set of decision values.

We used Netica to model the Bayesian belief network. Notice that some of the nodes start with factors that have equal probability values. Once the network was trained through Geometric Brownian motion the probability values were updated in the design.

The reason for this approach was simple, we didn't have proprietary data center demand values to use. When using this approach, the institutions would be expected to use their past real data along with any additional decision factors that they see fit. The framework and the approach will however still remain the same.

3.17 Simulation

3.17.1 Setup

The design setup takes core switches, aggregate switches, and access switches into account. This impacts the minimum number of servers that a system should have. While designing the system it is important to avoid bottlenecks as they can lead to lower performance and heating issues. Hence the system should not only be conceptually sound but must also practically functional. GreenCloud provided an excellent opportunity to rigorously test and validate decision network. GreenCloud also offers more flexibility than MDCSim and CloudSim, which are the other two data center simulators available in academia [8].

Decision Network (Qualitative values)	GreenCloud (Quantitative values)
Low Server Capacity	8 Servers
Medium Server Capacity	40 Servers
High Server Capacity	80 Servers
Low Network Bandwidth	1 Core switch & 1 Access switch
Medium Network Bandwidth	1 Core switch & 4 Access switches
High Network Bandwidth	2 Core switches & 4 Access switches

Table 10. Decision Network and GreenCould

3.17.2 GreenCloud

Table 4 shows an example of how different states could be setup with network and server configurations in GreenCloud. Although we have used only two factors, server and network, the user can add as many factors as the institution desires. The first box shows 3 different levels of server setup: low, medium, and high. Next, the network box includes Core switches and Access switches. Aggregate switches are simply equal to twice the number of core switches so they haven't been included in the table. Network combinations provide states for low, medium, and high network bandwidth. When the network and server configurations are combined together we get 9 unique states.

Server			Network			Combined			Unique States		
80	80	80	2C4A	1C4A	1C1A	80,8	80,4	80,1	S9	S6	S3
40	40	40	2C4A	1C4A	1C1A	40,8	40,4	40,1	S8	S5	S2
8	8	8	2C4A	1C4A	1C1A	8,8	8,4	8,1	S7	S4	S1

Table 11. State server and network configurations

High:	80 servers	2 Core 4 Access	= 8
Medium:	40 servers	1 Core 4 Access	= 4
Low:	8 servers	1 Core 1 Access	= 1

3.17.3 Simulating server & network demand

Obtaining proprietary data center logs is impractical. Since the model developed is versatile to accommodate any range of estimated growth, we can use the Geometric Brownian method to generate a growth chart as an example to demonstrate capability of

the model. Geometric Brownian method is used in simulating data center network traffic patterns, providing us with an excellent opportunity to use the Brownian method during our study [32]. Geometric Brownian method has also been specifically used with decision network to simulation a demand pattern [25]. Although it can have unpredictable characteristics, demand simulation has a trend. Because of the trend demand will have larger variance over longer period of time.

3.17.3.1 Geometric Brownian

```
function [Y] = geometric_brownian()
N = 100;
T = 5;
r = .3;
sigma = .3;
t = (0:1:N)'/N;   % t is the column vector [0 1/N 2/N ... 1]
W = [0; cumsum(randn(N,1))]/sqrt(N);  % cumsum function is running sum of N(0,1/N) variables.
t = t*T;
W = W*sqrt(T);
X = (r-(sigma^2)/2)*t + sigma * W;
Y = exp(X);
plot(t,Y);
title(['Geometric Brownian motion with diffusion cofficient=' num2str(sigma),' Interest rate=' num2str(r) ])
xlabel(['Time'])
```

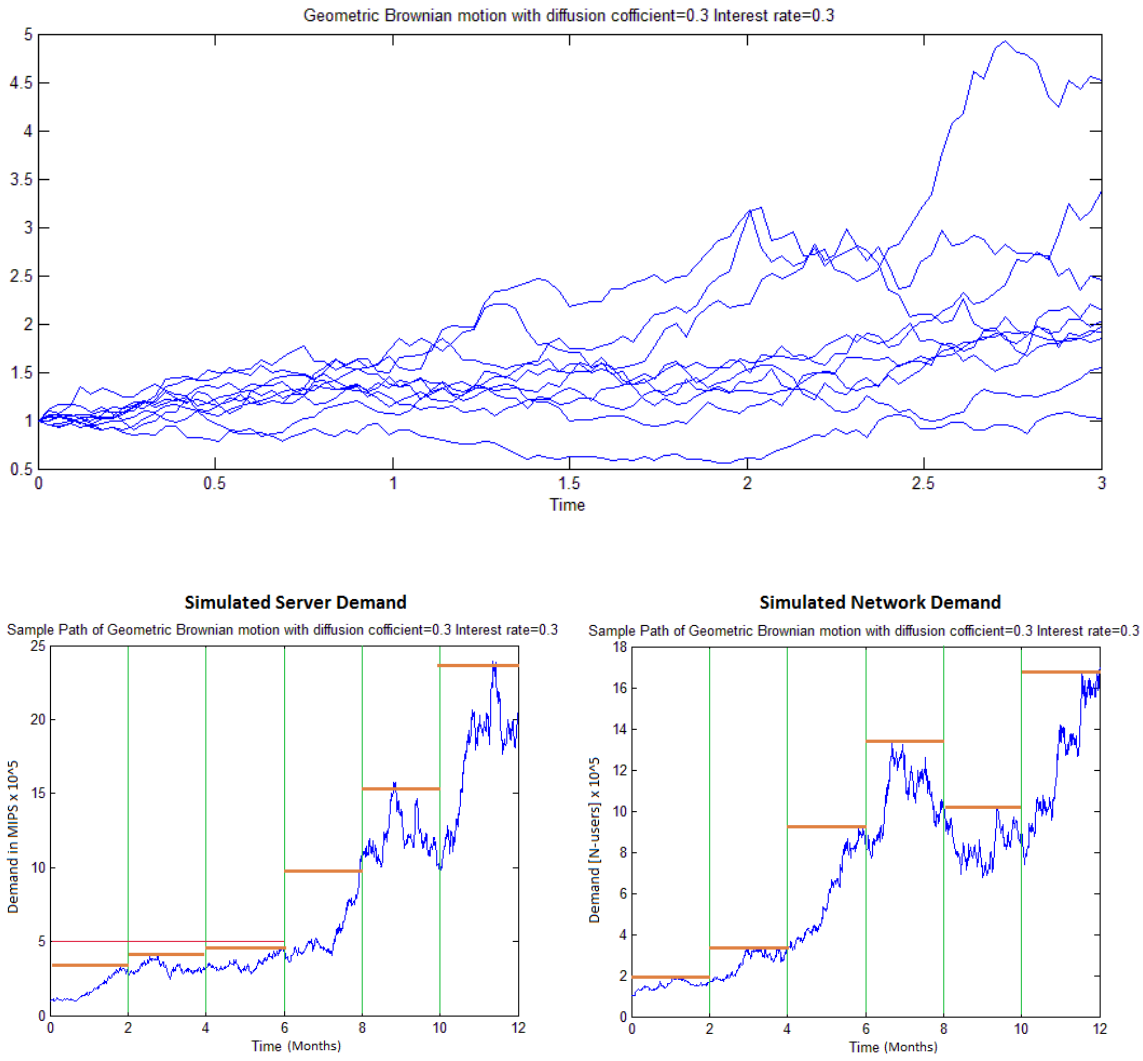

Figure 19: Geometric Brownian growth chart for server and network simulation

We ran numerous simulations to build server and network log data. Figure 10 is an example of a 1 year run of server and network demand. The estimates were detailed for each month by applying Holt's method on the discrete server values and Monte-Carlo on the continuous network values. The network used the resulting final estimates to determine the next possible states. The decision utility determined whether the transition

of state should occur. Numerous simulations allowed the decision network to zero-in on correct probability numbers for all the decision factors.

The figure below shows how the data center was able to maintain a steady amount of load. Since the purpose of the test was to see how the states performed during the maximum anticipated demand applicable to each estimated demand, the data center simulation was coded to push to maximum capacity. The y-axis in the figure below represents percentage of total data center capacity. As you can see the demand was maintained to push the data center at its, or close to, 100% capacity. The x-axis is shown below is time in hours.

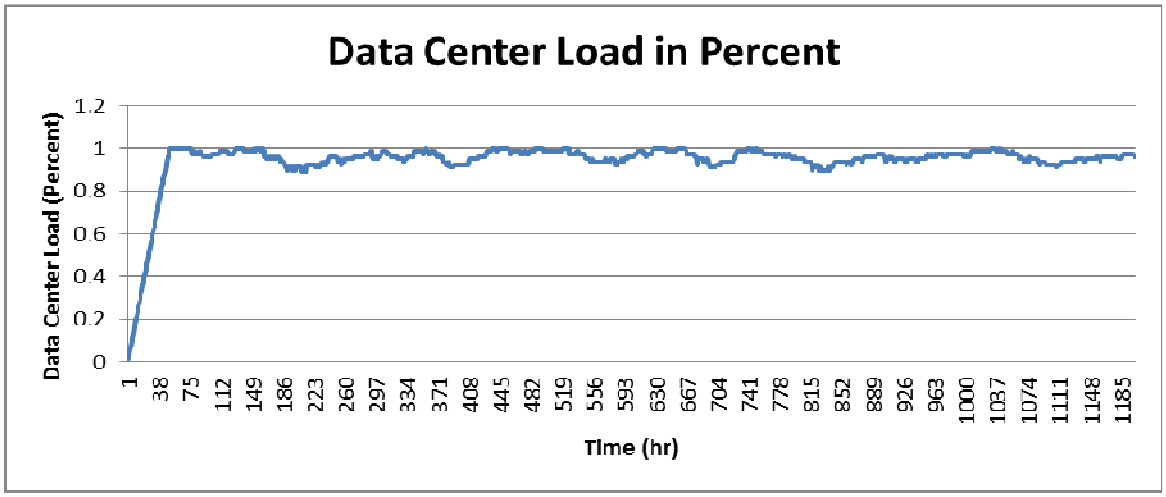

Figure 20: Data Center Load

The figure below show how the servers conserve energy once they are done with their tasks. This figure shows that running a task that is suitable for 8 servers on 80 servers at their max capacities can severly increase operational cost by running more severs longer. Since the task is suitable for 8 servers, by running it on 80 server will simply

make more servers run the task that could be accomplished in the same amount of time on 8 servers. However, running more servers will consume more energy in the system. Furthermore, it will take 80 server longer time to activate and deactivate for use than 8 servers. Network energy conservation is not provided since network is never shutdown. Network is used at all time to activate and deactivate servers dynamically.

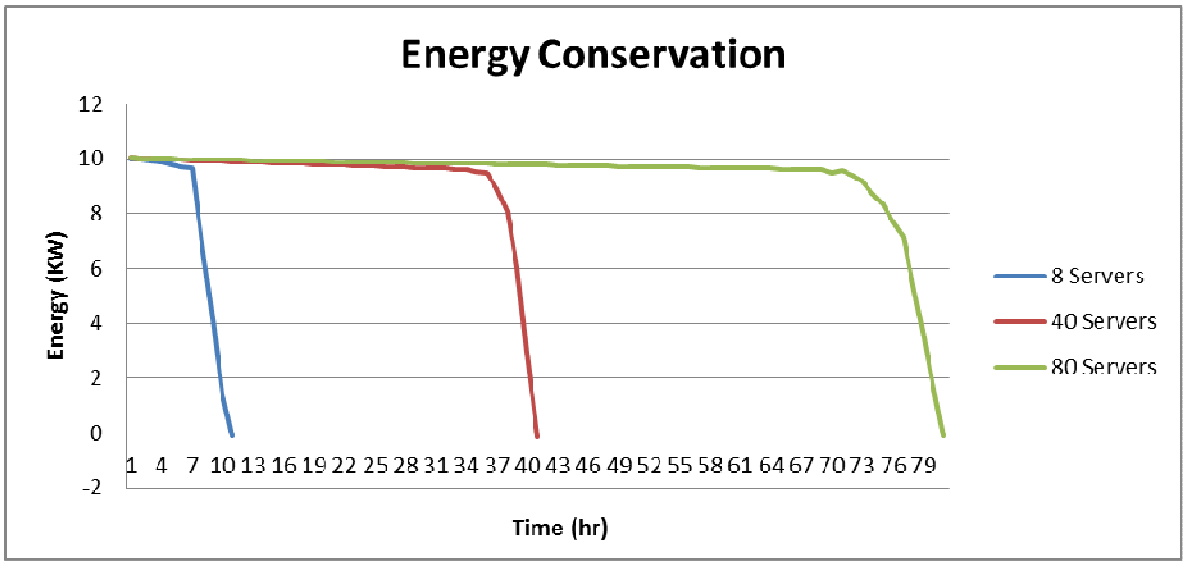

Figure 21: Energy conservation of servers.

Table 6 shows a typical simulation run on the decision model. The scenarios display scale up, upgrade, scale down, and as-is decisions. The estimate column displays the estimated results of running Monte-Carlo and the Holt's methods on each of Geometric Brownian simulated growth patterns. The sequential decisions are made at monthly server and network log intervals. The *Current State* column shows data center state at which decisions were made. The *Possible Next State* column shows what the next state of the data center could be based on the demand estimates. The *Future Operational Cost* reflects the operational cost of the possible next state. The metric that is also used

in the decision process but not shown here is *Current Operational Cost* since it was intuitive. The *Transition Cost* column shows the cost of scaling down data center state based on the current and possible next states. The *Final Decision* takes all criteria into consideration, utilizes the probabilities produced at each node, applies the utility algorithm for the deterministic approach, and recommends the next data center state. The rationale column briefly explains the reasons behind each final decision.

Estimate	Current State	Possible Next State	Future Operational Cost	Transition Cost	Final Decision	Rationale & Action
LSLN	-	S1	Low	Low	Initiate	
LSMN	S1	S2	Low	NA	S2	Scale up– therefore ignore everything else
LSLN	S2	S1	Low	Low	S2	Stay on S2. Operational cost is already low. Not worth transitioning.
MSHN	S2	S6	High	NA	S6	Scale up- therefore ignore everything else
LSHN	S6	S3	Medium	Low	S3	Inexpensive to move & significant savings
HSMN	S3	S8	High	NA	S8	Upgrade! - Redefine server's medium state
HSLN	S8	S7	Medium	Low	S7	Scale down and upgrade. Operational cost saving after the transition. Redefine server's low state
HSHN	S7	S9	Medium	NA	S9	Scale up– therefore ignore everything else
MSLN	S9	S4	Low	High	S4	Onetime expense of high cost to drop to low operational cost. Worth it! Redefine medium state.
MSHN	S4	S6	High	NA	S6	Scale up– therefore ignore everything else
MSMN	S6	S5	Medium	Low	S6	Not worthwhile to transition – stay at S6
HSHN	S5	S9	High	NA	S9	Scale up– therefore ignore everything else
LSMN	S9	S2	Low	High	S2	Onetime expense of high cost to drop to low operational cost. Worth it!

Table 12. Decision table and rationale

Some of the steps mentioned in the table include the action of *upgrade* paired with another action. Realistically, you can have a case where you are down-scaling the systems while upgrading the technology around it. This happens when the technology advancement allows the data center to consolidate servers without giving up on performance. To get a realistic visualization of the simulation, the server and network

configurations were setup in the GreenCloud data center simulator. The results of the run are showing in table 7. The data center ran at 94.4% of its maximum capacity. There were two total runs executed. During the first run, the servers were left running at their full capacity even after the tasks were done executing as shown in column labeled *Normal*. During the second run, dynamic shutdown policy was employed in which the servers were suspended to conserve energy after their tasks were done executing as shown in column *Dynamic Shut Down*. The task rate, MIPS (capacity), and the total number of tasks per run varied to simulate a 94.4% maximum data center capacity.

	Server(KWatts/hr)		Network(KWatts/hr/unit)			Process			
Decision	Normal	Dynamic Shut Down	Core	Aggregate	Access	Tot Tasks	MIPS	Data center Capacity in %	Process Rate (MBits/s)
S1	78.8	75.8	94.1	188.3	5	925	8000000	94.4	6.4
S2	78.8	75.8	191.9	383.7	19.6	925	8000000	94.4	6.4
S2	78.8	75.8	191.9	383.7	19.6	925	8000000	94.4	6.4
S6	393.8	378.7	191.9	383.7	39.6	4625	40000000	94.4	32
S3	78.8	75.8	191.9	383.7	39.6	925	8000000	94.4	6.4
S8	787.5	757.2	94.1	188.3	20.6	9250	80000000	94.3	64
S7	787.5	757.2	94.1	188.3	6	9250	80000000	94.3	64
S9	787.5	757.2	191.9	383.7	40.2	9250	80000000	94.3	64
S4	393.8	378.7	94.1	188.3	5.4	4625	40000000	94.4	32
S6	393.8	378.7	191.9	383.7	39.6	4625	40000000	94.4	32
S6	393.8	378.7	191.9	383.7	39.6	4625	40000000	94.4	32
S9	787.5	757.2	191.9	383.7	40.2	9250	80000000	94.3	64
S2	78.8	75.8	191.9	383.7	19.6	925	8000000	94.4	6.4

Table 13. Data center simulation of the states

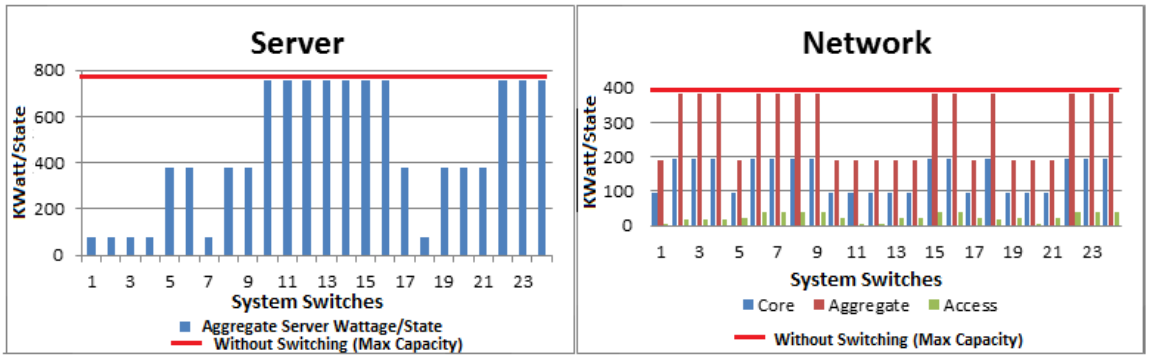

Figure 22: Server and Network Energy Costs

Figure 22 shows the total server energy usage in presence of decision model compared to the maximum energy the data center would have used in absence of decision model. Similarly the network graph shows breakdown of energy consumed by network switches in presence of decision model compared to the aggregated energy the network switches would have used otherwise.

4. RESULTS AND DISCUSSION

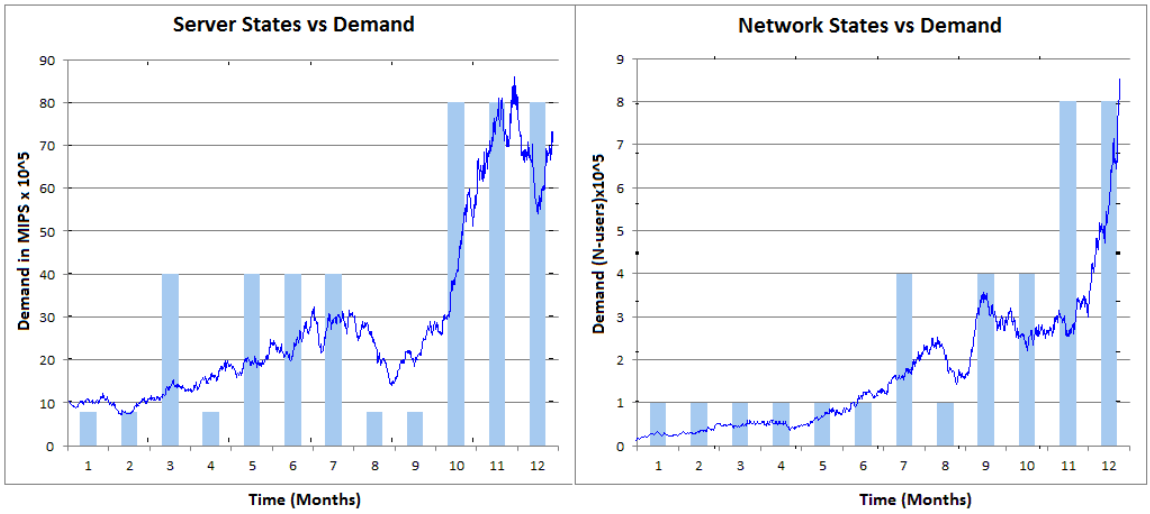

Figure 23: Decision Model and Data Center Demand

The model was successful in its initial design. The simulation data showed that the model was able to estimate the server and network demand without needing to rely on the backup data center. Hence, the availability metric of the performance was met by 100%. We set out to prove 4 major areas in effective data center capacity planning using the Bayesian Belief Network. First, we can tie a comprehensive list of decision factors together in the same model. Second, we can come up with a data center capacity solution that is optimal while taking all decision factors into account. Third, we handle uncertainties in the decision factors. Fourth, we manage risk while carefully balancing it with the overall cost.

Through our observation and knowledge on data centers we were able to assess which factors directly influence the data center capacity planning. We know that since there is a direct correlation between the demand of server and network resources and the

capacity at hand. Therefore, the design first started out with a base decision network with server and network nodes. Since our decision also depended on the overall cost of the data center design and maintenance we added transition cost and future (recommended) data center state. The initial model is shown in figure 14. When tested with simulation data, the base model itself was functional and shown to be effective. We took the model to the next level and factored in other three main areas of decision factors such as market conditions, development process, and internal business decisions. Hence we expanded the model as shown in figure 15. We refined and tested the model again with the iterative process shown in figure 18 and found it to be successful (and more accurate) in its results as shown in figure 19. We were able to show that the data center design model could handle multiple factors and provide an optimal result while accommodating all decision factors.

We handled uncertainties in the design by running sensitivity analysis on the belief network. The results of which were fed back in the system so that the next data center state accounted for the uncertainties in the decision process in order to propose an optimal solution. This process is also shown in figure 18. The uncertainties were only one part of the risk handled by the model. Uncertainties represented risks in the model and/or decision factors themselves. While the second type of risk in the design was in capacity planning.

Second type of risk was directly correlated to the decision itself and not the process. Hence, we needed a different way to handle the second type of risk in the design. The risk matrix that we came up with, shown in figure 8, effectively helped address the risk in the decision of changing data center state/configuration. The resulting risk assessment was built in the decision utility, shown in figure 16, where the utility weighed the risk

against the cost savings that would be experienced if the data center state was changed. The final results shown in figure 23 reflect the overall risk assessment (risk type one and type two) in the model while there was no compromise in the data center availability.

As the model matured so did the probability values in all the chance nodes, thus improving the ability to correctly anticipate the correct data center demands. The simulation allowed us to learn that as the data center matures, the decision network is more effective in not only predicting stable values for the data center states but also in anticipating the values that are closer to the demand. Model's ability to recommend cost effective and low risk states depended on the system's ability to estimate server and network capacities accurately.

The proposed design model shows promise in being able to handle and balance decision factors for the design. Further study is needed to test the design under specific environment with actual cost values. The results of the model will differ between institutions depending on institution's priorities, cost values, cost tolerance values, and decision rules. Further study is also needed to determine the sensitivity of the model's parameters to its overall design and results. Since GreenCould simulated data center algorithmically, the simulator itself didn't add any extra overhead to the overall design. Ideal candidates for this model are institutions that experience variability in demands thus giving them an opportunity to down-scale their design to save cost. Decision network's ability to address uncertainties in the decision process with the use of conditional probabilities allows the model to effectively down-scale a data center with manageable risks.

When the most important factor of a system model is it availability, businesses tend to adopt designs that are low in risk. Unfortunately, low risks designs often end up being

high in cost. The probability theories and decision networks allow the system to address uncertainties efficiently thus keeping the risk low while also lowering costs. The data center design proposed in this paper allows the institutions to tackle their costs by facing and minimizing risks. Above all, the institutions can employ this methodology while still maintaining their best practices. Decision model gives institutions an opportunity to incorporate all of their cost related points of pain in a methodology that also manages risks and uncertainties in decision making process.

Further research is still needed to understand the true relationship between the amount of data center log data available and the model's ability to correctly estimate and recommend the data center state. Further research is also necessary to study the relationship between the number of risk factors and the models ability to deliver accurate results while still addressing new uncertainties in the decision network. The risk criteria of the model depend on the backup data center that is rarely utilized. Further study of the model and estimation methods could aid the decision makers on how to best utilize the backup data center. Study is still needed to see how the model responds under conditions in which the decision factors continually change.

The geographical factors were not included in the study. For example, cooling data center would play a significant role in the decision factor in Arizona than in Michigan due to the climate conditions and associated cost factors. It will cost the company more to cool data center in Arizona than in Michigan. The user can use the model proposed in either conditions by simply allocating different cost value for data center maintenance and hence changing decision utility weights. The final results from the decision network will be applicable to its respective assigned weights customized for the geographic location that it represents.

Additional candidate for further research includes analyzing the flexibility and sensitivity of the decision network to the business decisions and its users. Business teams go through the phase of storming, forming, norming, performing, and in many cases deforming. Which means once the business objective is carried out the business teams would be dismantled and a new team could be formed. Hence, a new need could be generated in the model as a result. The model would need to be further analyzed to see how the results get influenced when the current state of the data center is based on the past decision weights while the future state is based on the new decision weights. An error correction factor may be necessary in that case.

In the area of capacity reduction, the simulation results indicate that while accommodating all necessary decision factors that influence data center capacity planning, the data center that was once designed to run at the maximum capacity has an opportunity to eliminate computational demands while maintaining network demands. Sensitivity analysis allowed us to narrow down the decision factors that actually influenced the decision in data center capacity planning and growth. As a result we were also able to question states S7, S8, and S9. It appears that based on the simulation, data S7, S8, S9 would've appeared to be a waste in the design. Hence, server capacities were possibly overestimated in the original design. Further research would need to occur at institutions to determine in such situations just why the data centers capacities were over estimated at the first place.

4.1 Capacity reduction results

We picked an assumed data center state with 80 servers, 2 core network switches, and 4 aggregate network switches. The simulation results show that at some point in time the data center was influenced to grow beyond a certain size. Simulation for our study replaced the actual data center knowledge that would normally be available at an institution for its research. Following the response of the nodes to the simulation data we were able to see a small development in the data center demand trend. The data center demands began at low and medium server capacities and low to high network capacities as shown in figure 7a. As the simulation matured, the capacity stayed steady at medium server capacity and medium to high network capacity shown in figure 7b. Towards the end of the simulation the data center capacity settled at medium server and high network, resulting in state 6 (S6) as shown in figure 7c.

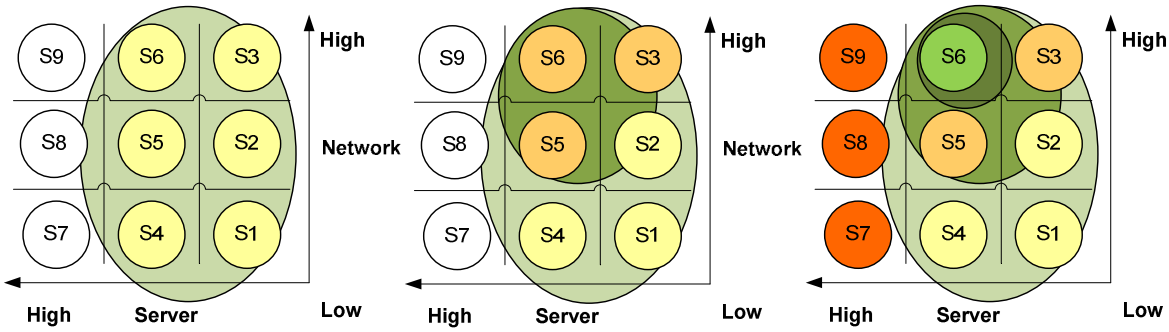

Fig 24a: First few iterations **Fig 24b: Next** iterations **Fig 24c:** Final iterations

Figure 24: Ideal data center state narrowed down using BBN

Based on the values generated in the simulation for the decision factors, the approach presented with the results displayed in the figure above. At each stage of decision

analysis performed, a sensitivity analysis was also carried to understand the influence of each on the data center capacity. The parameters narrowed down through sensitivity analysis allowed us to determine which final data center state was most advisable by the network. We started with a wide range of states from state 1 (S1) to state 6 (S6). As the algorithm progressed the states were further narrowed down three states, S3, S5, and S6.

Finally, as the simulation ended we were left with strong decision factors with reduced uncertainty pointing at state 6 (S6). Based on the simulation and decision probabilities, the server capacity in the data center was overestimated. If this method of data center capacity reduction approach runs out of data before a single state is concluded, then it would be wise to pick the state with maximum capacity from the remaining pool. For example if the design was left with states mentioned in figure 7b – S2, S3, S5, & S6, then it would be wise to select S6 to keep risks at minimum in the final design. If S2 was selected then institution may not prepare itself for medium server and high network demand conditions.

The point of this exercise is to not get hung up on particular results but to rather focus on the overall approach. The end results will differ between each institution based on the decision factors and the numeric values presented. Many companies have adopted the philosophy of and are comfortable with instituting static data center capacities for long term. However such institutions may be overestimating their data centers.

By letting the causal BBN learn the probability values with historic data and studying the results, we were able to assess the readiness of BBN. We analyzed not only the probability and sensitivity values in the results but also the risk factors at each decision level. Each step in the decision process correlated to the risk requirements and IT

requirements to ensure availability of the data center. Hence the GBM provided us with a trend in data center usage and the final results with a level of confidence that by reducing the data center capacity to the resulting state will not affect the data center availability.

4.2 Future work

Future work in this area would include studying the effects of technology changes within the system. We made an assumption that the technology would remain the same before and after the capacity reduction technique has been applied. Furthermore, the next step would also include separating different technologies such as host batch, enterprise data, java batch (if exists), midtier, user interface, file nets etc. We combined major demands into two main categories computations for server capacity and bandwidth for network capacity.

Future work would also include refining the decision utility to make risk assessment better aligned with cost. Presently the current and future operational costs are directly weighted against institution's risk tolerance values. Hence the decisions are made on value rather than the utility of value itself. Since the results show that current utility approach is functional and in-tuned with the currently design, refining the utility further will result in closer data center state approximations. That said, the next step in the decision utility design would include building utility for the current operational cost, future operational cost, and risk tolerance value before they are weighed against each other to determine the next data center state.

Current model takes business objectives into account, however the decision process employed in the model is strictly around data center IT factors such as network demand and server demand. In my opinion, business decisions influence IT decisions, but at times IT factors influence new business strategies. Currently model only incorporates influence of business decisions on IT factors. From my experience of building decision models in financial strategy role, I have noticed that business decisions widely vary based on individual probability distribution and its influence on an independed decision model. Currently the model doesn't incorporate individual decision models with the decision utility. Hence, there is lot of room for future work in the area of incorporating business decisions with the decision utility, building influence of IT on business, and building a way to incorporate business decision models in the current design.

Future work on the current model also includes expanding and refining the risk aspect in the model. That is, refining current risk matrix further by incorporating other risk factors. Presently the risk matrix chart only shows risk of experiencing down time in the data center by transitioning to a different data center state. The current risk is assessed in a binary fashion. That is, the risk is present (shown in red) and the risk is not present (shown in green). However, risk assessment in more sophisticated designs incorporate probability based risk assessment. Since the purpose of current risk matrix was to show conceptually the idea of building and incorporating the bare minimum level of risk factors in the design (i.e. data center availability), the current design successfully demonstrates the concept. Future research could refine the risk matrix to incorporate more factors in probabilistic manner.

Risk aspect in the design could also be built to have its own model that incorporates security risk, transactional risk – risk of losing transactional data while the data center is

being transitioned to the next state, and risk of incorporating multiple technologies along with multiple processes in the design. Other area of risk is to incorporate human error in the building out data center transitions. From my own personal observation, I have noticed data center technician make errors in following an already established process. Such risks also affect the data center availability and should be incorporated in the future design.

From the cost perspective, there could also be an assessment on how widely the overhead cost ranges between the data center states and the data center transition steps. It is assumed that larger data center states will have higher overhead cost. Similarly, it is also assumed that bigger data center state transitions will also have higher transition related overhead costs. Hence, the cost model could be expanded to incorporate, study, and control the data center overhead costs at each stage of the decision making process during the actual flow.

Future work on the design could also include a way to adapt and manage different technologies and processes in the design as they become available. The data center states will get heavily influenced by the advancement in the technology. What gets achieved by 8 servers and network switches could be accomplished with fewer servers and switches. Furthermore, the transactional capacity and the ability to enhance data center availability gets refined with time. Current design assumes that the data center server and network technology remains the same. However, future design could be expanded to incorporate new technologies during the data center transition and dynamically rebuild the data center states with the new technology. Furthermore data storage servers, a large server that's sole responsibility is to store and retrieve data, are

not presently incorporated in the design. Future work in developing next steps of data center states would include incorporating data store in the design as well.

Finally, presently the design doesn't incorporate multiple decision makers. The utility of the design takes in several decision factors, however the design is still made by one entity, the decision utility. The future design of this model could include the influence of different independent models such as risk, cost, development, business strategy, etc. where the decision are made by different decision makers/heads of each individual model. Therefore, having demonstrated the ability and the flexibility of the design, the future possibilities of expanding current research holds many avenues.

REFERENCES

1. Barroso, Luiz André, and Urs Hölzle. *The Datacenter as a Computer an Introduction to the Design of Warehouse-Scale Machines*. San Rafael, Calif. (1537 Fourth Street, San Rafael, CA 94901 USA): Morgan & Claypool Publishers,, 2009. Text (PDF xi, 107 p ill).

2. Bartolomei, Jason E., Daniel E. Hastings, Richard de Neufville, and Donna H. Rhodes. "Engineering Systems Multiple-Domain Matrix: An Organizing Framework for Modeling Large-Scale Complex Systems." *Systems Engineering* 15, no. 1 (2012): 41-61.

3. Brady, Jim. "Systems Engineering and Cost as an Independent Variable." *Systems Engineering* 4, no. 4 (2001): 233-41.

4. Crowston, Wallace B. S., Carnegie-Mellon University. Graduate School of Industrial Administration, and United States. Office of Naval Research. *Decision Network Planning Models*. Management Sciences Research Report No 138. Pittsburgh: Management Sciences Research Group, Graduate School of Industrial Administration, Carnegie-Mellon University, 1968.

5. Dicdican, Ruth Y., and Yacov Y. Haimes. "Relating Multiobjective Decision Trees to the Multiobjective Risk Impact Analysis Method." *Systems Engineering* 8, no. 2 (2005): 95-108.

6. Engel, Avner, and Miryam Barad. "A Methodology for Modeling Vvt Risks and Costs." *Systems Engineering* 6, no. 3 (2003): 135-51.

7. Harris, Jason, and NetLibrary Inc. *Green Computing and Green It Best Practices on Regulations and Industry Initiatives, Virtualization, Power Management, Materials Recycling and Telecommuting*. Australia?: Emereo Pty? ,, 2008.

8. Kliazovich, D., P. Bouvry, Y. Audzevich, and S. U. Khan. "Greencloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers." Paper presented at the Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, 6-10 Dec. 2010 2010.

9. Kliazovich, D., P. Bouvry, and S. U. Khan. "Dens: Data Center Energy-Efficient Network-Aware Scheduling." Paper presented at the Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom), 18-20 Dec. 2010 2010.

10. Martinez, Lauro J., Nilesh N. Joshi, and James H. Lambert. "Diagramming Qualitative Goals for Multiobjective Project Selection in Large-Scale Systems." *Systems Engineering* 14, no. 1 (2011): 73-86.

11. Bell, G., “Data center decisions.” Building Operating Management, no. 12 (2006): 53, 51-54.

12. Prieto, A. G., D. Gillblad, R. Steinert, and A. Miron. "Toward Decentralized Probabilistic Management." *Communications Magazine, IEEE* 49, no. 7 (2011): 80-86.

13. Sinha, Agam N., Frank L. Willingham, and Michael Hermes. "An Architecture Framework for Decision Support Systems (Dsss) in Air Traffic Management." *Systems Engineering* 4, no. 1 (2001): 1-12.

14. Pinnow, Osterburg, and Hanisch L. “Forecasting demand of potential factors in data centers.” Informatica Economica, no. 13 (2009): 1, 9-15.

15. Marcot, Bruce G. "Metrics for Evaluating Performance and Uncertainty of Bayesian Network Models." *Ecological Modelling* 230, no. 0 (2012): 50-62.

16. Abts, Dennis, and John Kim. *High Performance Datacenter Networks Architectures, Algorithms, and Opportunities*. San Rafael, Calif. (1537 Fourth Street, San Rafael, CA 94901 USA): Morgan & Claypool,, 2011. Text (PDF ix, 99 p).

17. Mowery, David, and Nathan Rosenberg. "The Influence of Market Demand Upon Innovation: A Critical Review of Some Recent Empirical Studies." *Research Policy* 8, no. 2 (1979): 102-53.

18. Parnell, Gregory S., Ronald E. Metzger, Jason Merrick, and Richard Eilers. "Multiobjective Decision Analysis of Theater Missile Defense Architectures." *Systems Engineering* 4, no. 1 (2001): 24-34.

19. Steinder, and Sethi A. S. “Probabilistic fault localization in communication systems using belief networks.” IEEE/ACM Trans. Netw. no. 5 (2009): 809-822.

20. Offutt, J. "Quality Attributes of Web Software Applications." *Software, IEEE* 19, no. 2 (2002): 25-32.

21. Stenzel, Joe, and MyiLibrary. *Cio Best Practices Enabling Strategic Value with Information Technology*. Hoboken, N.J.: Wiley,, 2011.

22. http://www.norsys.com/netica.html

23. Boursas, Latifa, Mark Carlson, Wolfgang Hommel, Michelle Sibilla, and Kes Wold. *Systems and Virtualization Management Standards and New Technologies*. Dordrecht: Springer,, 2008. http://www.gallaudet.eblib.com/patron/FullRecord.aspx?p=416927.

24. Cafaro, Massimo, and Giovanni Aloisio. *Grids, Clouds and Virtualization*. Dordrecht: Springer,, 2011. http://www.gallaudet.eblib.com/patron/FullRecord.aspx?p=645659.

25. Silver, Matthew R., and Olivier L. de Weck. "Time-Expanded Decision Networks: A Framework for Designing Evolvable Complex Systems." *Systems Engineering* 10, no. 2 (2007): 167-88.

26. Ogren, Ingmar. "On Principles for Model-Based Systems Engineering." *Systems Engineering* 3, no. 1 (2000): 38-49.

27. Giudici, Paolo. "Markov Chain Monte Carlo Methods for Probabilistic Network Model Determination." [In English]. *Journal of the Italian Statistical Society* 7, no. 2 (1998/08/01 1998): 171-83.

28. Lock, Russell. "Developing a Methodology to Support the Evolution of System of Systems Using Risk Analysis." *Systems Engineering* 15, no. 1 (2012): 62-73.

29. Gai, Silvano, Roger Andersson, and Tommi Salli. *Cisco Unified Computing System (Ucs) a Complete Reference Guide to the Data Center Virtualization Server Architecture*. Indianapolis, Ind.: Cisco Press,, 2010. http://library.georgetown.edu/search/i?=9781587141980

30. Piao, Park, and Lee E. “Problem localization using probabilistic dependency analysis for automated system management in ubiquitous computing.” Internet Research, no. 2 (2009): 136-152.

31. Tan, Weiping, Jose Ramirez-Marquez, and Brian Sauser. "A Probabilistic Approach to System Maturity Assessment." *Systems Engineering* 14, no. 3

(2011): 279-93.

32. Hsu, Yu-Sheng, and Cheng-Hsun Wu. "A Generalization of Geometric Brownian Motion with Applications." *Communications in Statistics: Theory & Methods* 40, no. 12 (2011): 2081-103.

33. www.philadephiatechnologypark.com