PARAMETRIC ESTIMATION OF LOAD FOR AIR FORCE DATA CENTERS

Derek P. Molle, Civ, USAF

AFIT-ENV-MS-15-M-170

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-MS-15-M-170

PARAMETRIC ESTIMATION OF LOAD FOR AIR FORCE DATA CENTERS

THESIS

Presented to the Faculty

Department of Systems Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Derek P. Molle, BS

Civ, USAF

March 2015

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENV-MS-15-M-170

PARAMETRIC ESTIMATION OF LOAD FOR AIR FORCE DATA CENTERS

Derek P. Molle, BS

Civ, USAF

Approved:

John M. Colombi, Ph.D. (Chairman)

John J Elshaw, Ph.D. (Member)

LtCol Brent Langhals, USAF, Ph.D. (Member)

AFIT-ENV-MS-15-M-170

Abstract

The Office of Management and Budget (OMB) has tasked Federal agencies to develop a Data Center Consolidation Plan. Effective planning requires a repeatable method to effectively and efficiently size Air Force Base-level data centers. Review of commercial literature on data center design found emphasis in power efficiency, thermal modeling and cooling, and network speed and availability. The topic of sizing data center processing capacity seems undeveloped. This thesis provides a better, pedigreed solution to the data center sizing problem. By analogy, Erlang's formulae for the probability of blocking and queuing should be applicable to cumulative CPU utilization in a data center. Using survey data collected by 38th Engineering Squadron, a simulation is built and correlation between the observed survey measurements and simulation measurements, and the Erlang, Gamma, and Gaussian-Normal distributions is found.

For a sample dataset of 70 servers over 14 hours of observation and a supposed 0.99999 requirement for traffic to be passed or otherwise unimpeded, Erlang distribution predicts 10 CPU cores are required, Gamma distribution predicts 10 CPU cores are required, Gaussian-Normal distribution predicts 9 CPU cores are required, Erlang B formulae predicts 14 CPU cores are required, and Erlang C formulae predicts 15 CPU cores are required.

AFIT-ENV-MS-15-M-170

Acknowledgments

I would like to thank Dr. John M. Colombi, my faculty advisor, for his guidance, support, and seemingly endless patience throughout the course of this thesis effort. I also need to thank Dr. Brent T. Langhals and Dr. John J Elshaw for their advice regarding various improvements to the text.

Derek Molle

Table of Contents

List of Figures

List of Tables

Page

PARAMETRIC ESTIMATION OF LOAD FOR AIR FORCE DATA CENTERS

I. Introduction

General Issue

From the 1950's through the 1970's, the mainframe data center was the only effective means of computing. Starting in the 1980's and until recently, the cost of individual computing has continued to drop. An overabundance of isolated functional and program managed data centers emerged. Recently, there has been a trend towards consolidation of these data centers to gain economies in staffing, power, environmental control, reliability, and computing power. This is due in large part to advances in high-speed networking technology and processor virtualization, which allow for sharing of processing across a pool of hardware computing resources. Sizing this pool of computing resources remains a challenge and is the focus of this thesis.

Federal mandates through the Federal Data Center Consolidation Initiative (FDCCI) require a 75% reduction in data centers across all federal departments by FY15. Specific to the Department of Defense (DoD), a 40% reduction is expected, which equates to reducing from 772 data centers to 428. DoD Core Data Center (CDC) initiatives have a target objective date of FY18. The discrepancy between the FDCCI and CDC timelines is best explained by the complexity and cost involved in data center consolidation [1].

The Office of Management and Budget (OMB) has tasked Federal agencies to develop a Data Center Consolidation (DCC) Plan in support of FDCCI [2]; a Presidential Directive memo was later issued reinforcing this task [3]. To accomplish this, the

Department of Defense is virtualizing all servers with few exceptions. The preferred approach is using “cloud” technologies, which is a “model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources” [4]. The issue is the cloud needed to house all these virtual servers has not yet been built, designed, or sized.

Problem Statement

One problem that arises in data center consolidation is the sizing of a cloud's virtual environment – effectively, determining the number of central processing units (CPUs). Sizing is accomplished by determining the load, which are time requests from virtual machines for CPUs. The duration of these requests is a random variable, related to the myriad of enterprise applications being used. Load is then derived from the probability distribution for the number of CPU resources occupied simultaneously. Using this probability distribution, the optimal size for a data center virtual environment can be found that minimizes waste (idle CPUs) while avoiding shortages (no free CPUs when requested).

One organization that is providing base data center consolidation support is the Air Force Space Command’s (AFSPC) 38th Engineering Squadron, Tinker AFB, OK. When tasked with sizing new data centers, a group there found there was no published formulae or models that could be used to meet the task. Initially the team looked at sizing new requirements based on existing average CPU utilization and scaling that against 60 percent of capacity of the cumulative environment. The 38th design team established 60 percent as a reasonable buffer against peaks in traffic on a basis of experience with data center applications. They believed this would provide a sufficient margin to account for any transient peaks in processing requirements. This thesis provides a better, pedigreed solution to the data center sizing problem.

Research Focus and Investigative Questions

Spurred by the major initiatives of the Department of Defense to consolidate data centers in support of the Federal Data Center Consolidation Initiative legislated by Congress, this thesis answers the following:

How should the Air Force size physical processing of a proposed data center?

It is important that the new data centers, the new cloud, be designed and sized to support the existing applications that are already fielded. It is also necessary to be able to project and compare costs of varying cloud implementations, such as Infrastructure as a Service (IaaS). Infrastructure as a Service would assume that DoD would outsource the operation, ownership and maintenance of all computing infrastructure and equipment of the data center. Effectively, DoD would pay a service provider on a per-use basis.

An analogy is hypothesized between the Erlang distribution describing the load of human callers in a telephony system and of virtual machines (hosts) requesting CPU time from a Hypervisor. The Erlang distribution is a continuous probability distribution related to other parametric exponential and Gamma distributions. While used originally by A.K. Erlang to estimate the number of phone calls made to a telephone switch, it has general applicability to many traffic engineering and queuing problems. This thesis will validate the use of the Erlang distributions by applying the same general methodology used to historically size telephony systems to size data center virtual environments. To answer the overall research question, a few investigative questions must first be posed and answered.

1. What are the prevalent metrics for computer processing emphasized by current practice, or found in the academic body of knowledge?
2. How well does an Erlang distribution approximate data center CPU load?
3. How should processing in data centers or IaaS projects be sized?

Methodology

To attempt to answer these investigative questions a review of the technology involved is conducted followed by statistical analysis of measurements taken by a 38th CEIG survey team as well as analysis of a simulation based upon measurements taken during that survey. Correlation between the observed survey measurements, as well as simulation measurements, and the Erlang, Gamma, and Gaussian-Normal distributions will be calculated to test the validity of the proposed sizing solution.

Assumptions/Limitations

Limitations influencing this study include the cost and complexity of testing on a production network, the critical nature of the enterprise data center systems, and the limits of our ability to measure load and generate perfectly realistic traffic. This thesis will assume that Windows® Performance Monitor can provide accurate measurements of load, the open source tool LookBusy can generate realistic traffic, and that these results from a prototype lab running Windows® Hyper-V are sufficiently typical for all hypervisors [5]. This thesis will also assume that the survey data, upon which analysis is based, is typical of an Air Force data center.

Implications

Federal mandates through the Federal Data Center Consolidation Initiative (FDCCI) require a 75% reduction in federal data centers by FY15, with DoD expecting a 40% reduction. The discrepancy between the FDCCI and CDC timelines is best explained by the complexity and cost involved in data center consolidation. The Office of Management and Budget (OMB) has tasked Federal agencies to develop a Data Center Consolidation (DCC) Plan in support of FDCCI. This thesis will provide a repeatable method to effectively and efficiently size base-level data centers.

Preview

In the next chapter background information on data centers, virtualization, metrics for data centers, and some tools for statistical analysis will be introduced. In subsequent chapters, a methodology for statistical analysis of survey results and for simulation of the survey data in a virtual environment, then results from one such survey and analysis of those results as well as analysis of simulation results based upon the survey.

II. Literature Review

Chapter Overview

The purpose of this chapter is to overview technologies, metrics, and statistical analysis tools to provide a background for the thesis question.

Technology

History of Technology

The electronic computer has evolved over the years. The industry started with mainframe computers for laboratories and large firms in the 1940s. In the 1960s the integrated circuit led to the development of the minicomputer and later the microprocessor enabled the personal computer. In the 1969s the ARPANET was established beginning the era of the networked computer [6]. It was in the 1970s that virtualization began in earnest with the development of time-share computing on mainframe computers. Virtualization is defined as a technology which “enables several operating systems and applications to run on one physical server or ‘host.’ Each self-contained virtual machine (VM) is isolated from the others, and uses as much of the host’s computing resources as it requires. [7]”

Virtualization Technology

As time-share computing was further developed modern virtualization technology emerged. It is implemented using a supervisory program referred to as a hypervisor virtual machine manager. The hypervisor abstracts the hardware and presents common interfaces to virtual machines. By sharing common storage it becomes possible for the virtual machines to almost instantly be migrated, over the network, between physical computers [8].

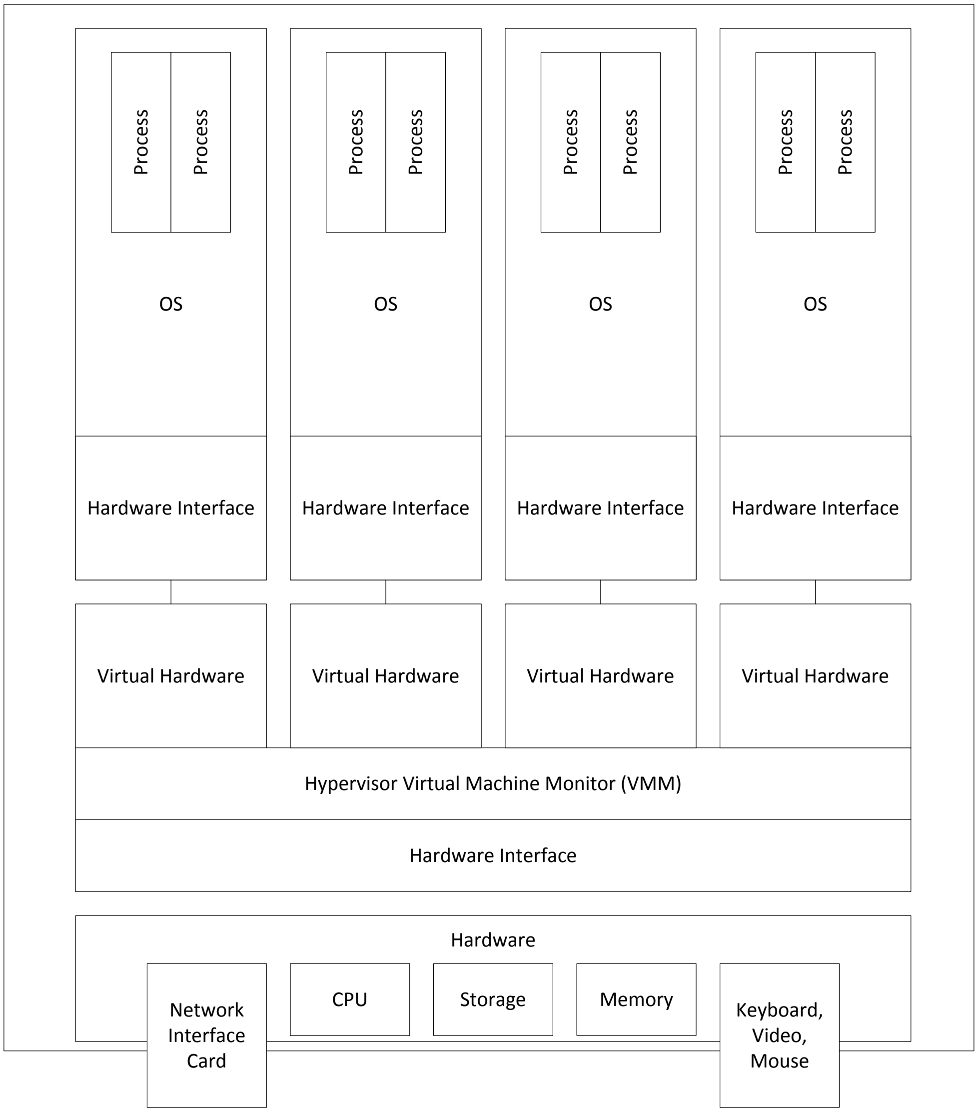


Figure 1. Hardware Abstraction by the Hypervisor

Migration is used to balance load across hosts and to prevent loss of service when hardware goes into maintenance or fails. This creates a common pool of resources across a common network of computers. Policies for management of resources in this common pool are addressed Grit and Wood in "Virtual machine hosting for networked clusters:

Building the foundations for "autonomic" orchestration," and "Black-box and gray-box strategies for virtual machine migration," [9] [10]. Microsoft's Performance and Resource Optimization (PRO) feature for Hyper-V Virtual Machine Manager (VMM) allows administrators to configure target utilization for host hardware in virtual environments. In VMM 2008 the default target CPU utilization is 90 percent [11]. While some details of implementation differ between hypervisors, and there are performance differences between hypervisors, these features are common across all modern hypervisors [5]. The cumulative effect of these features underpinning virtualization is that a small set of hardware can run what used to require a large set of hardware while at the same time offering higher redundancy than before. Below, figures 2 and 4 help illustrate these architectural advantages of virtualization.

Data Center Technology

For the purposes of this paper, the central technological aspect of the data center is the ability to process and store data. This capability is often expressed in terms of Higher Performance Computer (HPC) systems. Hussain and Malik identify three types of HPC architecture: Cluster, Grid, Cloud. The HPC cluster is a group of computers with redundant interconnections that form a highly available system [12]. The cluster is centrally managed and interconnected across a LAN environment. One example would be a website that is load-balanced across multiple web servers in a data center. Frequently the load balancing is accomplished by sharing an IP address across the cluster. The load balancing function is sometimes performed by the cluster itself.

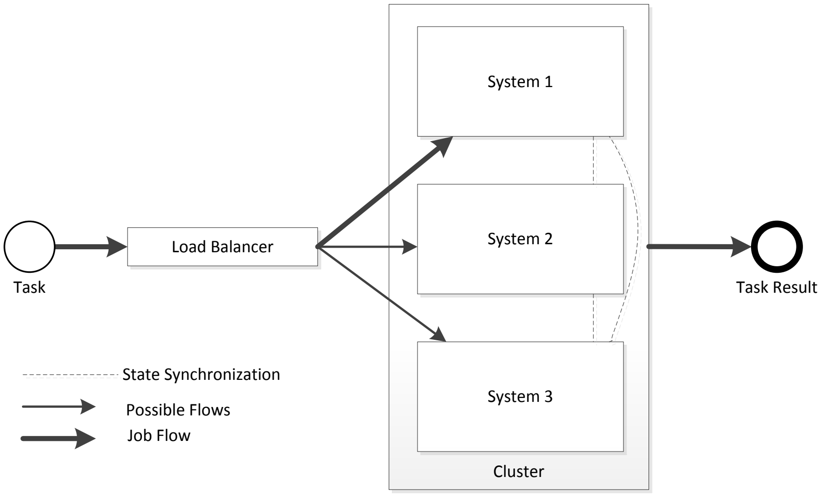


Figure 2. Simple HPC Cluster Architecture

The HPC Grid is a group of computers that use the Internet to spread calculations out across low cost commodity components. A grid does not exist in a single data center but instead in across multiple data centers and frequently homes, classrooms, and office floors where it can use spare compute to complete calculations for the grid. The distributed nature of the grid limits the workload it can take on. Typically grid computing is used to handle non-interactive workloads that can be broken into self-contained chunks to be processed by grid members. The same chunk may be sent to multiple grid members for result verification or to mitigate against the loss of a grid member.

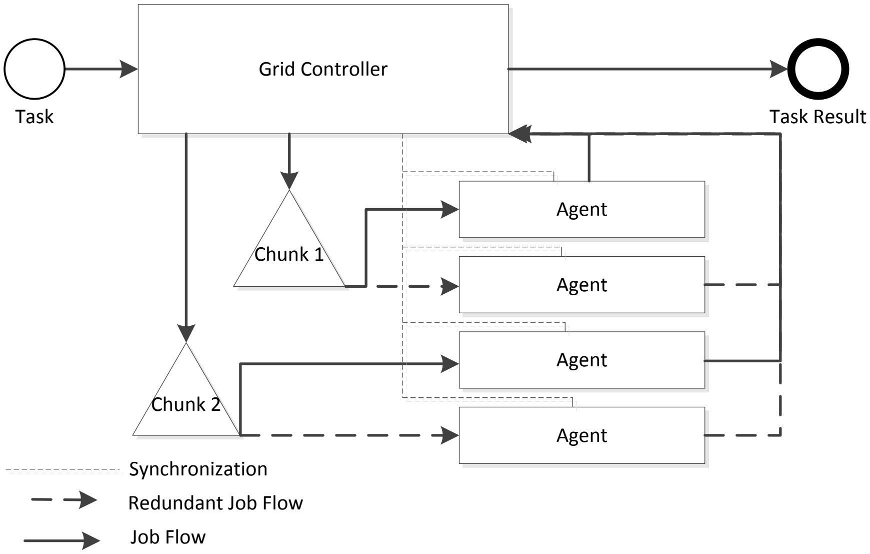


Figure 3. Simple HPC Grid Architecture

HPC cloud computing is an amalgamation of grid and cluster computing. The cloud can exist in a single data center or across data centers, under a single administrative domain or across many, as a private commodity or as a public service.

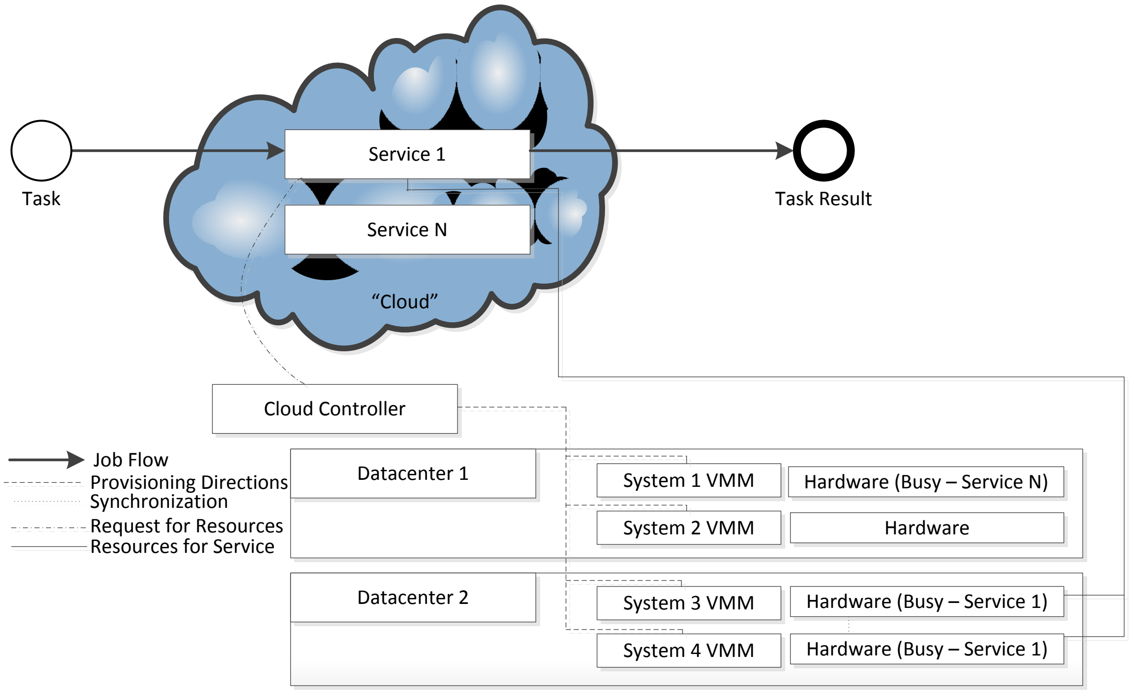


Figure 4. Simple HPC Cloud Architecture

The HPC Cloud architecture is shown in Figure 4. Users are presented with persistent service while the workload is shared across physical data center resources. The system VMM controls scheduling of resources on each system. The single most unique feature of the cloud computing model is that it is entirely virtualized. Where computers participating in a HPC cluster or HPC grid can be virtualized they can also run on bare metal. The self-service and elasticity characteristics of cloud computing requires a common pool of resources to exist in the data center that can be provisioned on request.

Metrics

Traditional Data Center Metrics – Power, Ping, and Pipe

Review of the literature finds remarkably little by the way of similar or alternative solutions to the problem of sizing a data center's compute environment. Instead there is a great emphasis by recent authors on the "power, ping, and pipe" [13] associated with data center design. Other work describes the goals of "power and pipe" in terms of high power usage effectiveness, PUE, as achieved through efficient power distribution and cooling systems. Methods for achieving "Ping," referring to the need for a responsive and resilient internetwork in the data center, have been well discussed in other work [14] [15] [16] [17].

Load Testing Metrics

There is an existing body of knowledge on the use of software load generators and the analysis and identification of important performance counters. Methods vary from emulation of production networks, stress testing of individual web applications, to arbitrary load generation in commercial cloud environments such as the Amazon Elastic Cloud. [18] [19] [20] [21] [22].

Data Center Metrics

In order to study data center capacity and utilization it is necessary to establish metrics and collection methods. In studying a 1500 node HPC cluster [23] identified SNMP counters, Sampled Flow, and Deep Packet Inspection methods to measure traffic patterns and performance. These are network focused metrics. SNMP is capable of measuring CPU information but it is not implemented in USAF Windows Server deployments. Instead, performance counters and WMI are available to provide reliable information about CPU utilization. Metrics are measured over the interval between two measurement instants [24].

Tools for Statistical Analysis

This section will develop and introduce concepts of statistical analysis by way of descriptive statistics including terms involved in calculations and distributions used by inferential statistics. The purpose of descriptive statistics is to describe and summarize the characteristics of a sample. To accomplish that, measures of mean, median, standard deviation and variance are used. These measures can then be applied to distributions to infer properties of the underlying system. To describe how well a statistical distribution fits a sample dataset the correlation coefficient can be calculated by equation 1.

Eq. 1

where:

Correl(X,Y) = the correlation coefficient of sets X and Y
x = Sample of X
$\bar{x}$ = Mean of X
y = Sample of Y
$\bar{y}$ = Mean of Y

The Central Limit Theorem is a theory that states, convolution of a sufficiently large number of independent random variables will be approximately normally distributed. This distribution is also referred to as the Gaussian-Normal distribution and shown as equation 2.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \qquad \text{Eq. 2}$$

where:

$P_{(x)}$ = Probability Density Function of Gaussian-Normal Distribution
x = Sample value
μ = Mean
σ = Standard Deviation

The Gaussian-Normal distribution is one of the simplest random function used for time-series modeling allowing direct calculation from just mean and standard deviation [25].

Erlang’s Formula

This section will develop and introduce Erlang’s formulas. The Erlang function is an equation initially published as a solution to telephony problems in Elektroteknikeren Vol 13 (1917) by Agner Krarup Erlang [26]. It is founded in the theory of "statistical equilibrium" by which , for a very large number of calls the individual characteristics of each call does not affect the group characteristic as an increase in one call duration will be balanced by a decrease in some other call duration. Erlang supposes the probability of finding all telephone lines engaged (a blocking condition as described by B) can be approximated by the total number of lines and the average number of calls per time unit (traffic intensity) [27]. These dimensionless units of traffic intensity have come to be known as Erlangs. The blocking probability, P_{b}, is also known as the Grade of Service.

Eq. 3

where:

P_b = Probability a request for service is blocked
A = Normalized load or mean traffic intensity
N = Number telephone lines

As an alternative to blocking excess traffic, the Erlang C formula describes the probability that a call is placed on hold and waits for service in a queue. To simplify the mathematics of it, it is assumed that callers will wait indefinitely.

Eq. 4

where:

P_w = Probability a call has to wait and is buffered
A = Normalized load or mean traffic intensity = call frequency x call duration
N = Number of telephone lines

Under certain conditions P_w can be calculated from P_b, N, and A without an additional round of summation. Where N > A then

Eq. 5

where:

P_w = Probability a call has to wait and is buffered
P_b = Probability a request for service is blocked
A = Normalized load or mean traffic intensity
N = Number of telephone lines

Erlang's formula for solving certain problems in telephony has been applied to computer networks before. Chromy and Baronak [28] applied Erlang distributions to ATM and IP network traffic. They found success for Erlang B in the case of ATM networks and Erlang C for IP networks, noting that ATM and IP handle delay differently. In their work, the Erlang equations were not modified but the parameters were. Specifically, A was used to describe Link utilization and N to describe Bandwidth.

Table 1 : Chromy and Kavacky's common parameters of synchronous and asynchronous networks

Synchronous Network		Asynchronous Network	
B [%]	Lost calls ratio	B [%]	Loss rate
C	Probability of waiting for service	C	Probability of delay
A [Erl]	Total offered traffic	A [%]	Link utilization
N	Number of channels (links)	N [Mbit/s]	Bandwidth

Bonald and Thomas focused on IP traffic and similarly found the probability that internet traffic (IP traffic) , which should reduce flow rate to avoid buffering, is bounded by the Erlang C formula [29]. Bonald and Thomas, similar to Chromy and Kavacky, identified an explicit performance relationship involving only link capacity (N) and expected demand (A).

Erlang’s Distribution

This section will introduce the Erlang Distribution and the Gamma Distribution.

$$\frac{-}{} \qquad \text{Eq. 6}$$

where:

$f(x;\alpha, \beta)$ = Probability Density Function of Gamma Distribution
x = Sample Value
$\alpha = \mu^2 / \sigma^2$ = Distribution Shape
$\beta = \sigma^2 / \mu$ = Distribution Rate

The Erlang Distribution, equation 7, is a special case of the Gamma Distribution, equation 6, where the shape parameter α is a positive integer. This allows the denominator of the distribution to be calculated by factorial rather than by the Gamma function.

$$\frac{-}{} \qquad \text{Eq. 7}$$

where:

$f(x;\alpha, \beta)$ = Probability Density Function of Erlang Distribution
x = Sample Value
$\alpha = \mu^2 / \sigma^2$ = Distribution Shape, a positive integer
$\beta = \sigma^2 / \mu$ = Distribution Rate

The NIST published Engineering Statistics Handbook cites the Erlang Distribution as being frequently used in queuing theory applications [25].

Summary

The idea of sharing a common pool of computing resources is not a new one. Old time-share methods have been replaced by virtualization technology which is able to abstract and present tasks to generic hardware in a data center. The HPC data center as a cloud offers users real-time service, with resources on demand. To answer the investigative questions, quantitative metrics are needed. Review of commercial literature finds emphasis in power efficiency, thermal modeling and cooling, and network speed and availability. The topic of sizing data center processing capacity seems undeveloped. By analogy Erlang's formulae seems adaptable to the problem, however, of all of the reviewed literature only one reference to the applicability of Erlang's methodology to CPUs could be found [30]. Other statistical tools from descriptive and inductive statistical analysis can be used to further analyze.

III. Methodology

Chapter Overview

As discussed earlier, there are significant similarities between the telephony problems of 1917 and the modern virtualized computing environment. The greatest possibility for error lay in the analogy of call frequency and duration to CPU utilization. To validate the adaptation of the Erlang formula to problems of the modern data center a medium sized virtual environment was built containing several virtualized application servers and numerous load generating "client" virtual machines. Performance monitors were used to track average CPU utilization from the perspective of the Guest operating system, as well as the System/Processor Queue Length. Processor Queue Length is proportional to the amount of time threads were awaiting physical compute resources and is typically less than 12 [31]. This emulated data will be processed to fit parameters to for the Erlang formula and distribution, as well as the Gamma and Gaussian-Normal distribution using correlation coefficients.

Erlang Formulae Modified For CPU Loading

Since originally the Erlang formulae were intended for sizing telephone circuits, the definitions for some variables must be modified to fit with a virtual data center environment rather than a telephony environment. Based on work by Chromy and Kavacky the definitions of some variables involved in Erlang’s formulae should be modified using Table 2.

Table 2 : Common parameters of synchronous Telephony and Virtual Data Center

Telephony		Virtual Data Center	
P_b [%]	Lost calls ratio	P_b [%]	Loss rate
P_w	Probability of waiting for service	P_c	Probability of delay
A [Erl]	Total offered traffic	A [%]	Mean CPU utilization
N	Number of channels (links)	N	Number of logical CPU (cores)

Applying the modified parameters from table 2 to equation 3 and 4 results in equations 8 and 9 below.

Eq. 8

where:

P_b = Probability a request for service is blocked
A = % Mean CPU Utilization
N = Number of logical CPU available to the hypervisor

Eq. 9

where:

P_c = Probability a request for service has to wait and is buffered
A = % Mean CPU Utilization
N = Number of logical CPU available to the hypervisor

Equations associated with distributions (Eq. 2, 6, 7) are also modified such that x = N and μ = A therefore $\alpha = A^2 / \sigma^2$ and $\beta = \sigma^2 / A$. The CDF of the Gaussian-Normal, Gamma and Erlang distributions is produced by equation 10.

Eq. 10

where:

f(x) = Cumulative Distribution Function
P(x) = Probability Density Function

In practice, all of these functions are calculated by Excel.

Survey Data Collection

When tasked with sizing new data centers as part of a data center consolidation effort, a group at 38th Engineering Squadron, Tinker AFB, OK sent engineering teams to survey the existing Air Force data center infrastructure at prospective sites. Over the course of several days NIPRNET connected systems were identified, the system owners identified and interviewed, and performance information was collected using Windows Performance Monitor from the production application servers. Twenty-seven CPU Utilization samples were taken in 30 minute intervals over the course of 14 hours for each of 70 identified application servers beginning at 10 am. Metrics are measured over the interval between two measurement instants [24]. An example of the survey data is shown in table 3. To protect the OPSEC of the servers their names have been replaced with numbers. This CPU Utilization information, included in Appendix B, is used to generate scripts for simulation of the consolidated data center and for direct statistical analysis in Chapter 4.

Table 3 : Example of Survey Data

Server	%utilization	Time	Server	%utilization	Time	Server	%utilization	Time
005v	0	0	006v	0	0	007v	1	0
005v	0	30	006v	0	30	007v	28	30
005v	0	60	006v	0	60	007v	26	60
005v	0	90	006v	0	90	007v	24	90
005v	1	120	006v	0	120	007v	2	120

Experimental Equipment Setup

A Dell PowerEdge R720 with eight 10,000 RPM, 900 GB SAS harddrives, twenty-two 1600MHz 16GB DIMMs, and two 2.70GHz Xeon E5-2697 CPUs is used as the host system. Each Xeon CPU provides the host 12 cores. After Hyper-threading, the host operating system is presented 48 logical processors. There is a L1 cache of 1.5MB, a L2 cache of 6MB, and a L3 cache of 60MB. Harddrives 2 – 6 are combined into one 3353GB logical drive with Windows 2012 soft RAID-5. Harddrive 1 contains data unrelated to the project. Harddrive 7 is used to store installation files. Harddrive 8 is the system drive of the host operating system. The host system also has a 3000GB Fusion-io drive2 installed and recognized by the host operating system as disk 0, it is used as part of a tiered storage pool to overcome disk IO limitations found during early testing.

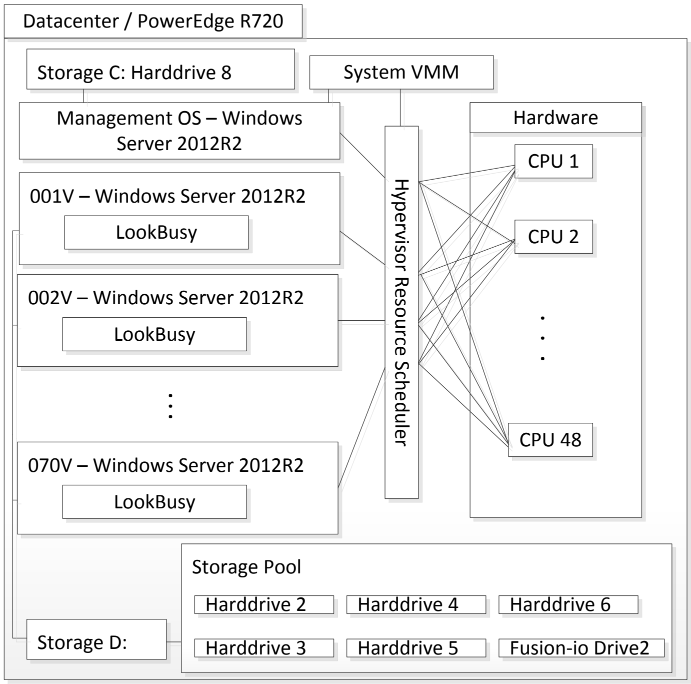


Figure 5. Experimental Equipment Setup and Configuration

Software Configuration

The software configuration, and key aspects of its interaction with hardware, is depicted in figure 5. Windows Server 2012R2 is used as the host operating system. The Hyper-V hypervisor is enabled allowing virtualization of additional guest operating systems (OS) and applications. The RAID-5 partition is labeled “D.” One virtual switch named “NIC0 Bridged” is created and associated with all guest OS. The host operating system is able to use this virtual switch to communicate with guest OS.

The guest operating systems are Windows Server 2012R2. One virtual machine was created with 8GB of RAM, 2 virtual CPU, and a virtual 60GB harddrive on the D drive. After installation of Windows Server 2012R2, Cygwin, Lookbusy, and cpu-load-generator.py are loaded and a scheduled task is created to start the simulation. The template virtual machine is shutdown and cloned to simulate an Air Force Data Center.

Simulation Generation

To create the data center workload trace twenty-seven samples, one per half hour over a fourteen hour period, were taken from 70 systems in the Air Force Data Center. These were then enumerated into trace files for use by cpu-load-generator.py. The simulation was configured to change state every 9 seconds with sampling every 3 seconds to satisfy Shannon's Theorem. Three simulation sets were created:

- Replay of survey data
- Random sampling of survey data set per application
- Random sampling of survey data set per unit time

The first simulation is intended to show validity of the simulation in a consolidated virtual environment and is expected to closely match the survey data which was taken from a distributed physical environment. The second simulation is intended to simulate the data center with an assumption that load between applications is not dependant on time. The third simulation is intended to simulation the data center with an assumption that load between applications is dependent on time. A python script was developed for the purpose of generating the second and third simulation sets and is included in Appendix A.

Workload Generation

The cpu-load-generator.py script was developed by Dr. Beloglazov as an open source tool to generate CPU load according to a configuration profile or workload trace. It uses Devin Carraway’s open source tool Lookbusy to make an arbitrary number of CPUs arbitrarily busy. While Dr. Beloglazov used a web service to assign and trigger load profiles, in this simulation task scheduler is used with load profiles assigned by powershell script.

Each clone is configured with a CPU load profile based on measurements taken by the 38ES survey team. This allows the virtual environment to either replay real activity seen in a physical Air Force Data Center or run simulation scenarios described in the previous section.

Data Collection

The Windows Performance Monitoring tool, Perfmon, was used to collect data. Perfmon allows for the recording of hundreds of different counters. In the simulation phase, this study used several counters including [32]:

Table 4 : Windows Performance Monitoring Metrics

Metric	Description	Example
<Hyper-V Hypervisor Virtual Processor\CPU Wait Time Per Dispatch>	The average time (in nanoseconds) spent waiting for a virtual processor to be dispatched onto a logical processor	12699.25
< Hyper-V Hypervisor Virtual Processor(_Total)\% Total Run Time >	Shows the time the cpu is not idle across the Hyper-V environment	5.243332
<System\Processor Queue Length>	Shows the number of threads waiting to be serviced	0
< Hyper-V Hypervisor Virtual Processor(_Total)\% Hypervisor Run Time>	Slice of the % Total Run Time used by the Virtual Machine Manager	0.052604
< Hyper-V Hypervisor Virtual Processor(_Total)\% Guest Run Time>	Slice of the % Total Run Time used by guest virtual machines	5.190729
<System\File Write Operations/sec>	Shows the combined rate, in incidents per second, of file system write requests to all devices on the computer	61.32717
<System\File Read Operations/sec>	Shows the combined rate, in incidents per second, of file system read requests to all devices on the computer	60.66057
<System\File Control Operations/sec>	Shows the combined rate, in incidents per second, of file system operations that were neither read nor write operations.	271.6394

During data collection in the production environment, a PowerShell script was used to sample the Perfmon value for <System\% Processor Time> [33].

Analysis Methodology

There are five sections of data to be analyzed. Raw survey data, cumulative survey data, cumulative replay simulation data, cumulative application-based simulation data, and cumulative time-based simulation data. The raw survey data consists of CPU utilization measurements taken by the 38 ES survey team on a per application server basis. Cumulative survey data consists of CPU utilization measurements summed across all 70 application servers in the data center for each sample in time. The sum of all samples at t=0 is the cumulative CPU utilization at t=0, the sum of all samples at t=30 is the cumulative CPU utilization at t=30, and so on. All three Cumulative simulation data sets consist of total CPU utilization measurements from the hypervisor supporting the simulation. The cumulative simulation data sets offer a view into resource utilization in a fully virtual environment.

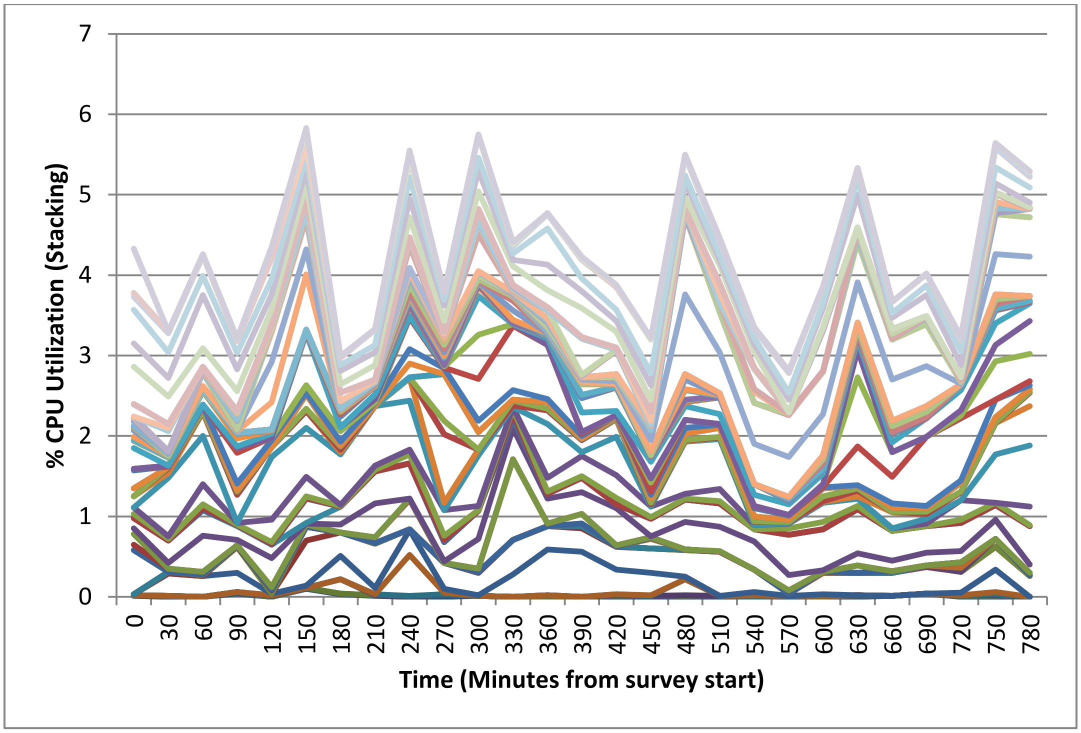


Figure 6. Cumulative CPU Utilization of Survey data, Stacked Line Plot

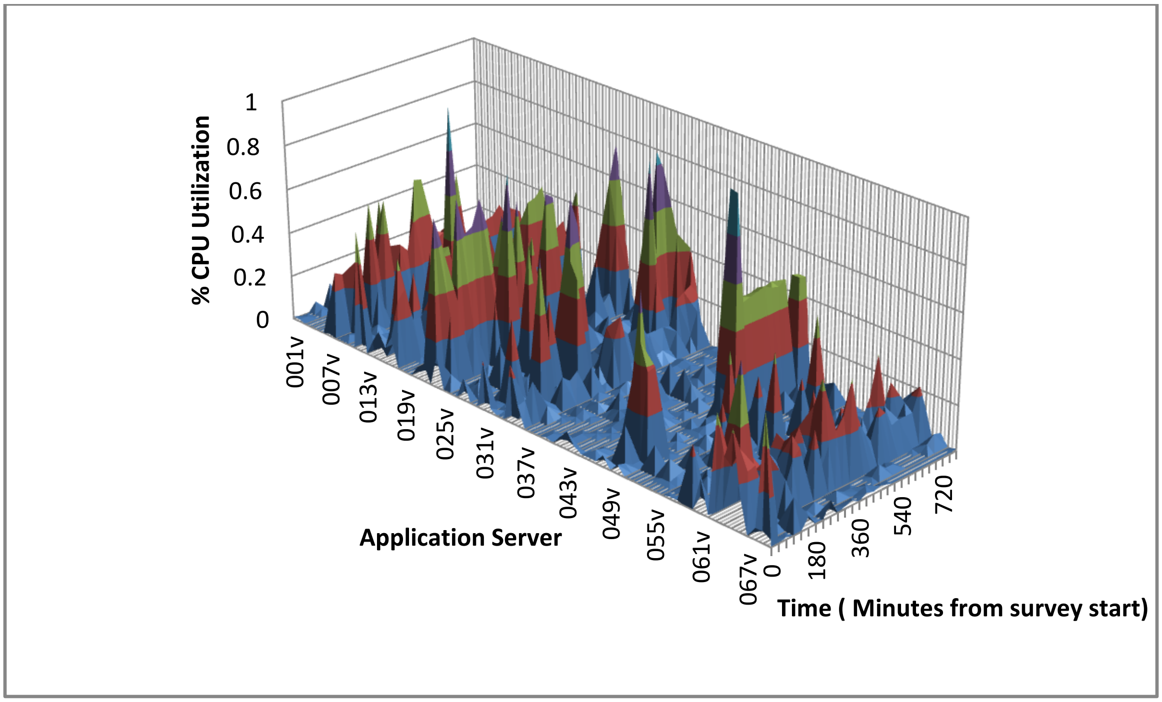


Figure 7. Surface Plot of Survey Data

The analysis methodology is based on chapter 2 findings for Erlang's formulae and more generally for statistical analysis from the Engineering Statistics Handbook. For each data set, descriptive statistical values for minimum, maximum, mean and standard deviation are calculated. A histogram is plotted then compared by correlation with Erlang's formulae then with the Erlang, Gamma and Gaussian-Normal distributions. As noted above, the Erlang distribution will be calculated by rounding the α parameter to the nearest integer.

IV. Analysis and Results

Chapter Overview

In this part the results obtained by calculation from measurements obtained during survey, as well as during simulation and compare with Erlang, Gamma, and Gaussian-Normal distributions are presented.

Descriptive Statistical Analysis

Appendix B includes an anonymized version of Windows Performance Monitor <System\% Processor Time> data collected during the survey of an Air Force Data Center. Server names were replaced with numbers from 001v to 070v.

Since the Performance Monitor results were collected into a csv format, Excel was used to calculate descriptive statistics that summarize the raw survey data. Table 5 compares these statistics. All values, except for sample count, are expressed in terms of physical CPU cores.

A median value of zero for Raw Survey Data indicates that at least 50% of samples reported no server activity, emphasizing possible efficiencies to be found in virtualization and consolidation by showing that over 50% of the time data center assets are simply waiting for work.

Table 5 : Comparison of Descriptive Statistics for Each Data Set

Survey Data		Cumulative Survey Data		Cumulative Replay Simulation Data	
Minimum	0	Minimum	2.81	Minimum	2.59
Maximum	1	Maximum	5.83	Maximum	14.1
Mean	0.0629	Mean	4.24	Mean	4.60
Median	0	Median	4.23	Median	4.63
Samples	1838	Samples	27	Samples	2850
Standard Deviation	0.136	Standard Deviation	0.937	Standard Deviation	0.814
		Cumulative App-based Simulation Data		Cumulative Time-based Simulation Data	
		Minimum	2.16	Minimum	2.19
		Maximum	9.3	Maximum	8.4
		Mean	4.59	Mean	4.56
		Median	4.55	Median	4.52
		Samples	5401	Samples	5401
		Standard Deviation	0.651	Standard Deviation	0.810

For most cumulative data sets, the minimum, mean, and standard deviation of CPU utilization are similar. The cumulative app-based simulation data set shows a smaller standard deviation when compared with the other simulation scenarios and results in a larger α value and smaller β value per definitions for equations 6 and 7 as well as slightly lower predictions for hardware requirements.

Analysis of Fit of Erlang's Formulae

The second part of analysis uses equations 8 and 9 to attempt to size a data center. The descriptive parameters of mean CPU utilization, A, is used together with various numbers of CPU cores available for traffic, N, then compared with the inverse cumulative distribution of the observed datasets.

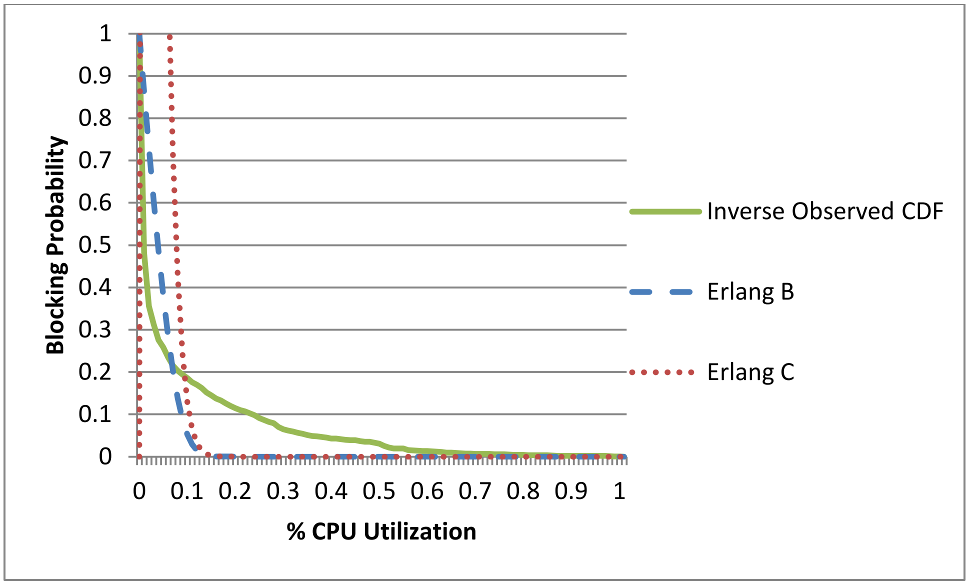


Figure 8. Comparison of Erlang B and C with Survey Data

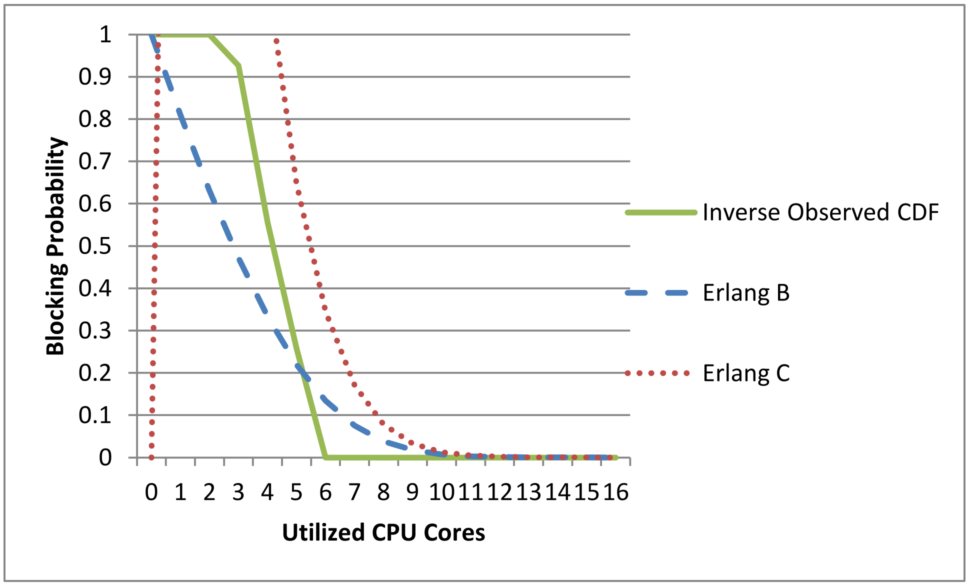


Figure 9. Comparison of Erlang B and C with Cumulative Survey Data

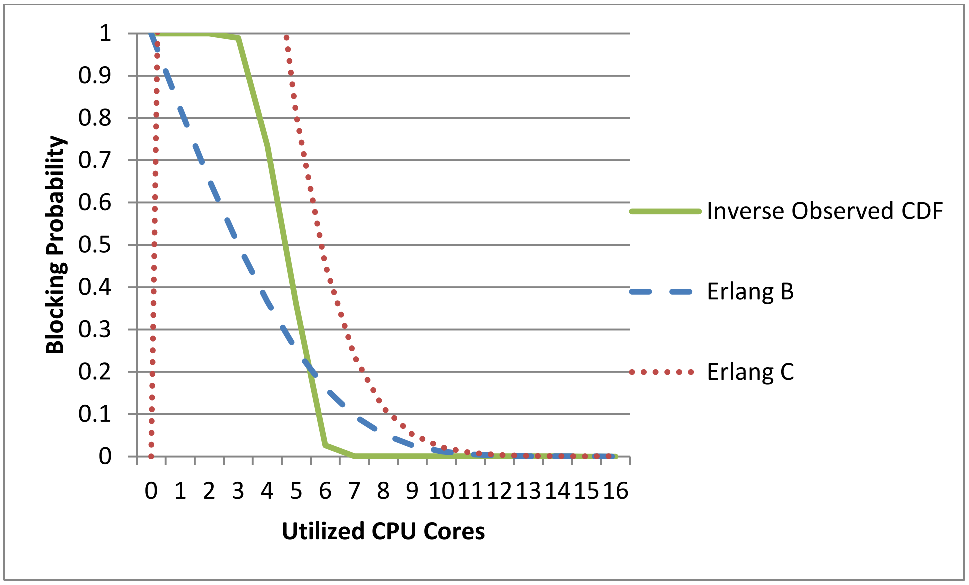


Figure 10. Comparison of Erlang B and C with Cumulative Replay Simulation Data

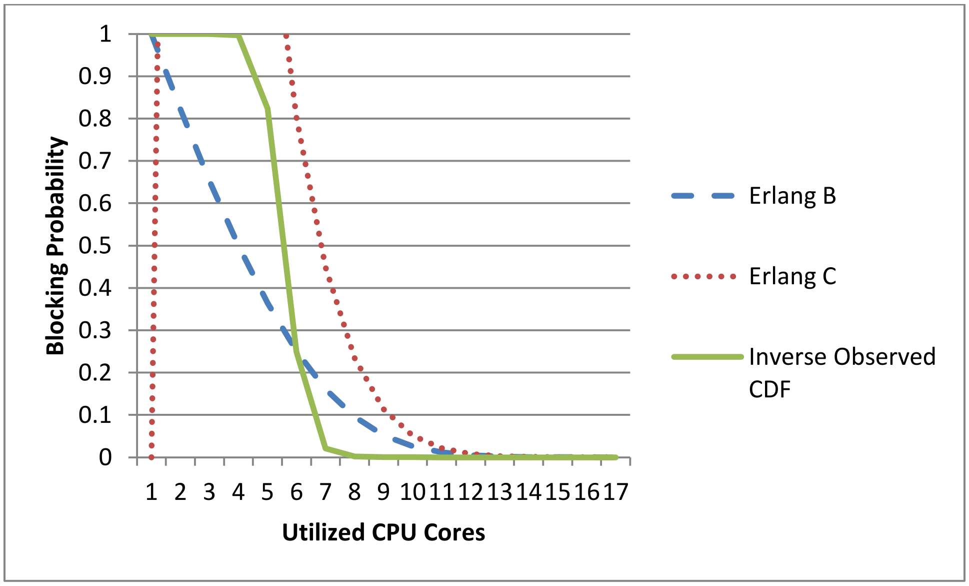


Figure 11. Comparison of Erlang B and C with Cumulative App-based Simulation Data

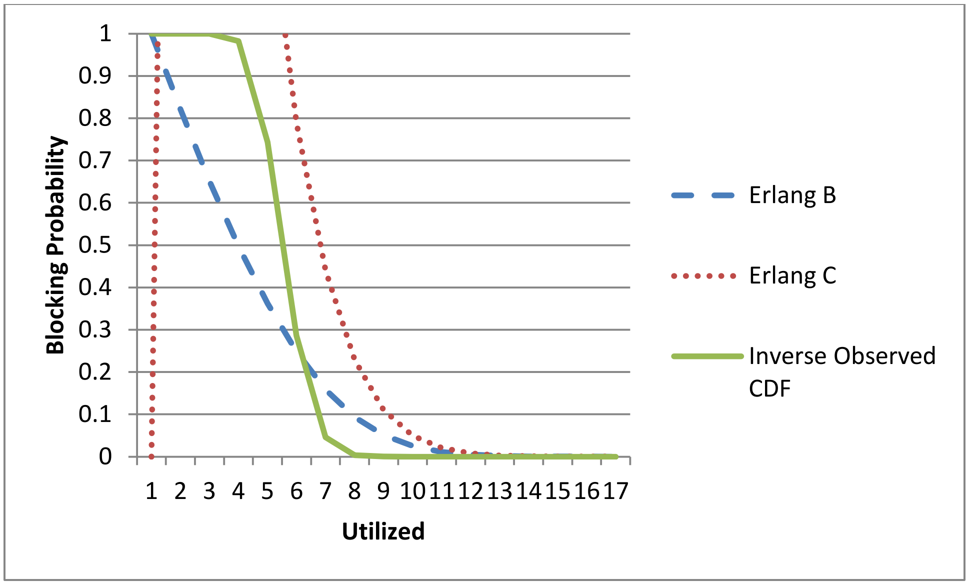


Figure 12. Comparison of Erlang B and C with Cumulative Time-based Simulation Data

The Erlang B and Erlang C formulae do not appear to be a good fit in figures 8 - 12. Calculation of correlation coefficients in table 6 shows some degree of correlation.

Table 6 : Comparison of Erlang Formulae Correlation Coefficients for Each Data Set

	Erlang B	Erlang C
Cumulative Survey Data	0.950820839	0.937472778
Cumulative Replay Simulation Data	0.941291397	0.921250068
Cumulative App-based Simulation Data	0.931956494	0.911316961
Cumulative Time-based Simulation Data	0.941643605	0.922631406
Raw Survey Data	0.877987087	0.791652648

Other inferential statistical methods discussed in Chapters 2 and 3 are next pursued.

Inferential Statistical Analysis

The third part of analysis uses descriptive parameters calculated in the previous section as inputs to known statistical distributions and will use correlation coefficients to find best fitting distributions. Beginning with the raw survey data, visualized in Figure 13, it is shown that the Gamma distribution is a very close match to the observed data. It is also apparent that the Erlang distribution is not a good fit here. This is caused by the distribution shape parameter for the Erlang distribution, α, which in this case was less than 0.5 to begin with, being rounded to the nearest positive integer of 1. As per equations 6 and 7, the distribution shape is a product of μ^2 / σ^2, or A^2 / σ^2 when translated by the modified parameters table, table 2.

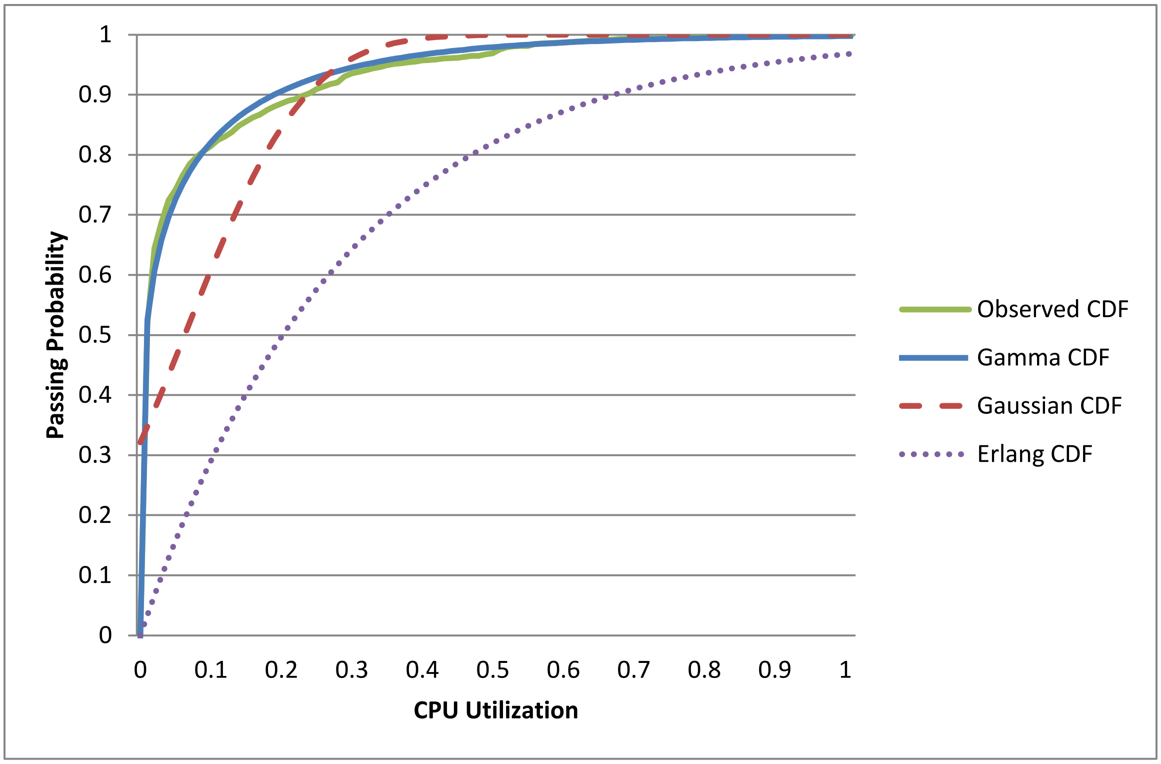


Figure 13. Raw Survey Data – Comparison of Observed, Gamma (*α = 0.215 , β = 0.292)*, Gaussian-Normal (μ = 0.063, σ = 0.136) and Erlang (*α = 1 , β = 0.292)* Distributions

The Erlang, Gamma, and Gaussian-Normal distributions are compared with the observed replay simulation data in Figures 14, 15, and 16. The same distributions are compared with the application based simulation data in Figures 17, 18, 19 and again with the time based simulation data in Figures 20, 21, and 22. Each comparison was split out into its own figure to improve readability. Each of the three distributions match so closely, however, it is still difficult to differentiate between the two curves.

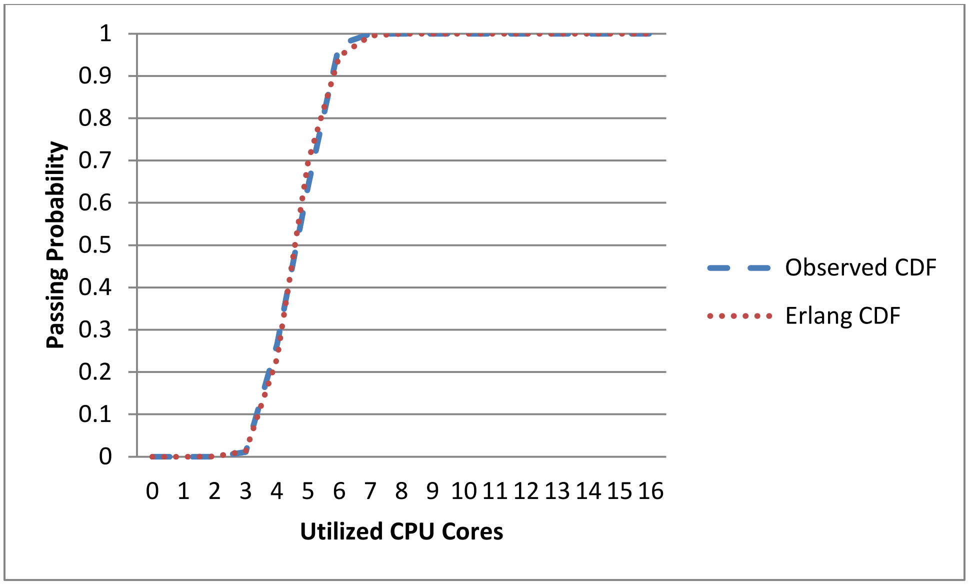


Figure 14. Cumulative Replay Simulation Data – Observed CDF vs Erlang Distribution CDF (*α = 32 , β = 0.14419)*

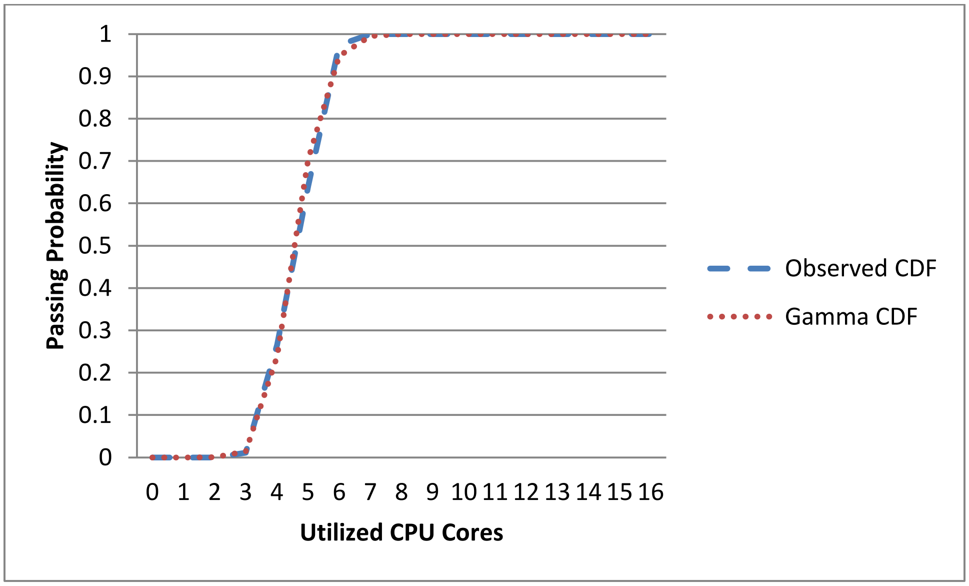


Figure 15. Cumulative Replay Simulation Data – Observed CDF vs Gamma Distribution CDF (*α = 31.901, β = 0.14419)*

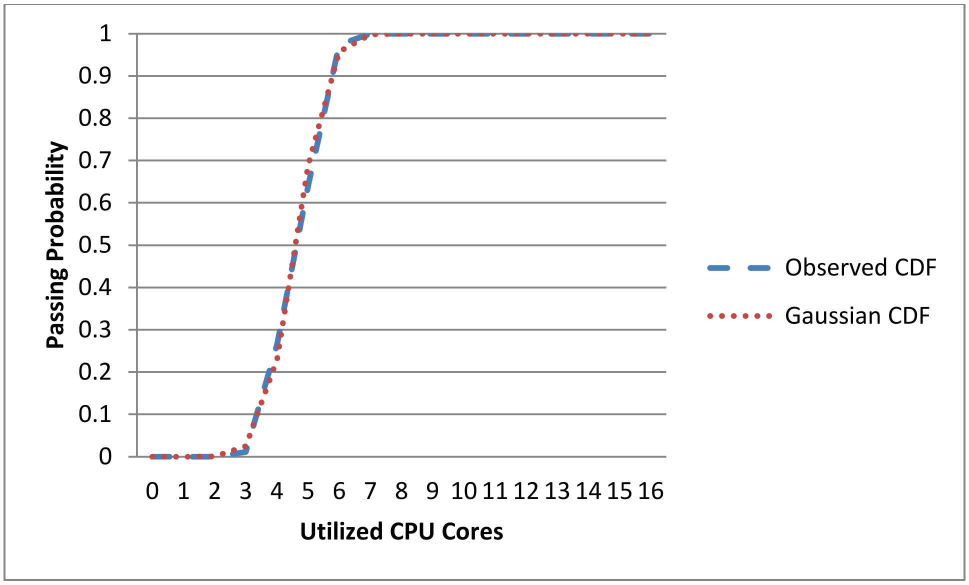


Figure 16. Cumulative Replay Simulation Data – Observed CDF vs Gaussian-Normal Distribution CDF (μ = 4.5999, σ = 0.81441)

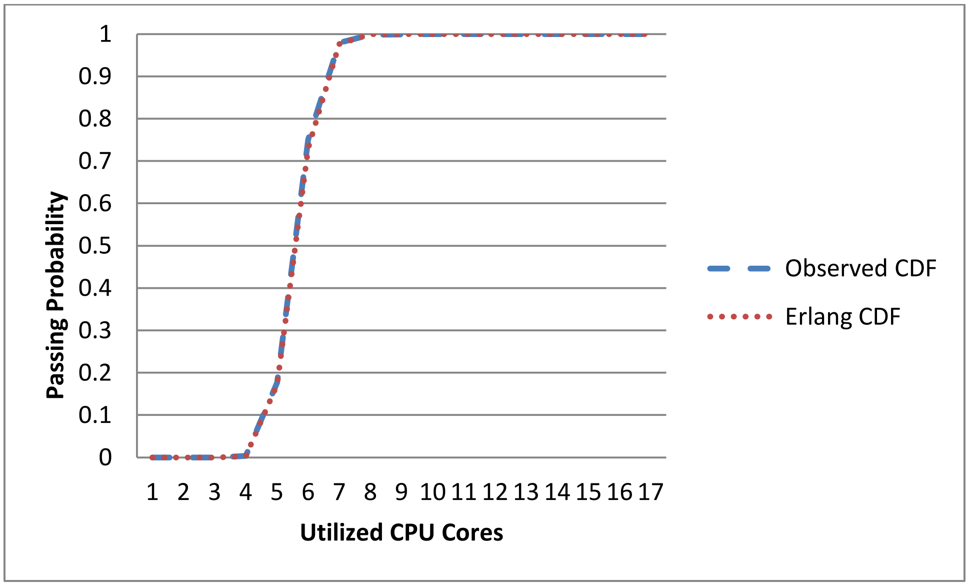


Figure 17. Cumulative App-based Simulation Data - Observed CDF vs Erlang Distribution CDF ($\alpha = 50$, $\beta = 0.092395$)

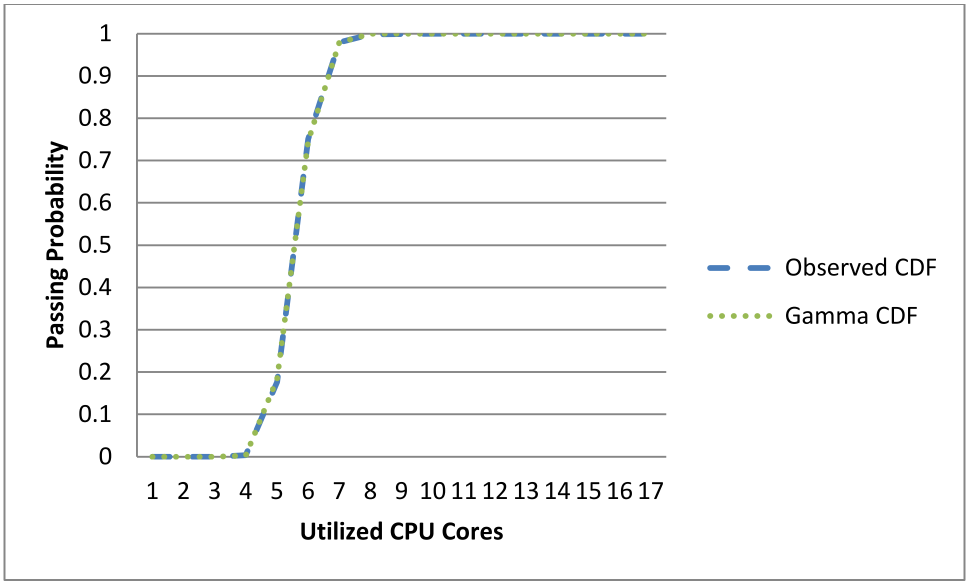


Figure 18. Cumulative App-based Simulation Data – Observed CDF vs Gamma Distribution CDF (α = 49.664, β = 0.092395)

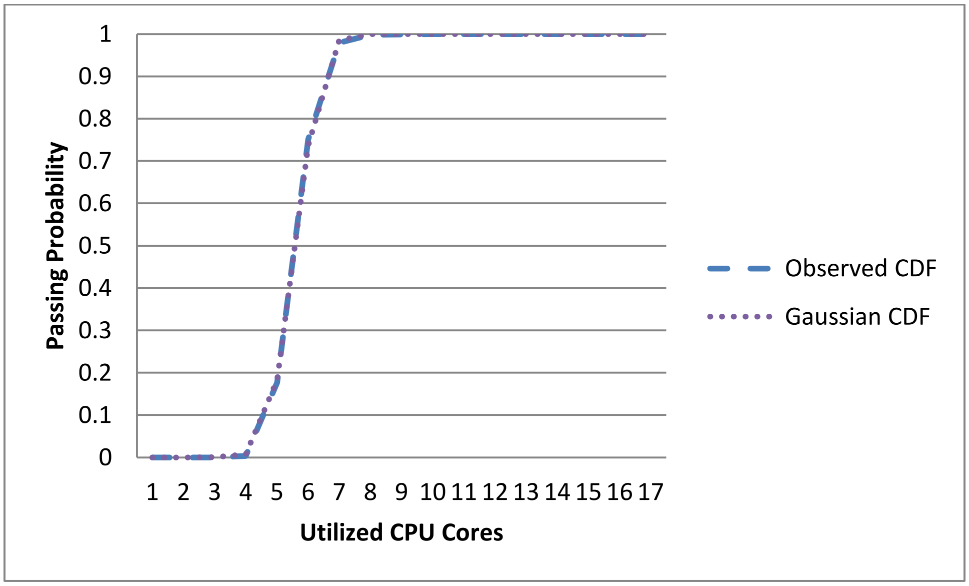


Figure 19. Cumulative App-based Simulation Data – Observed CDF vs Gaussian-Normal Distribution CDF (μ = 4.5888, σ = 0.65114)

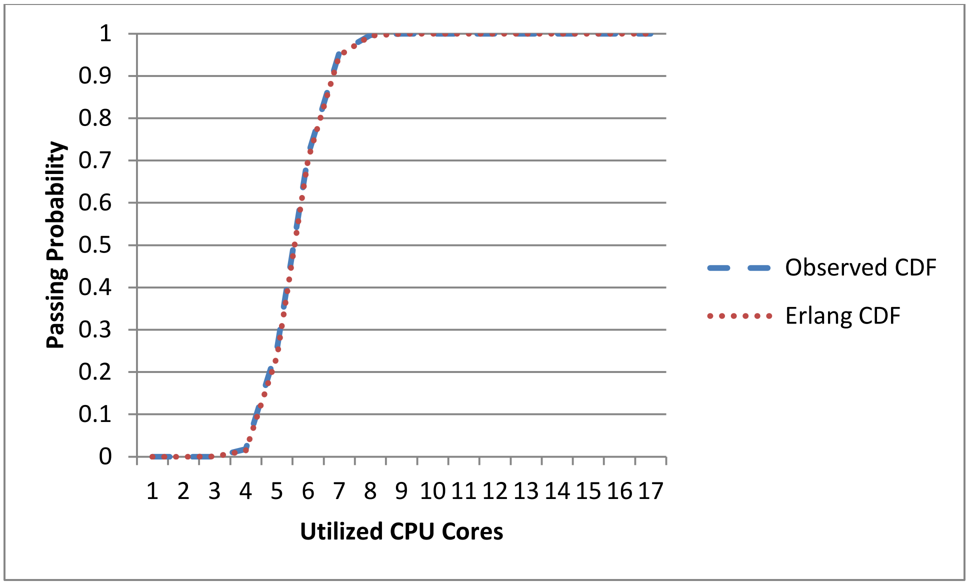


Figure 20. Cumulative Time-based Simulation Data - Observed CDF vs Erlang Distribution CDF (α = 32, β = 0.14377)

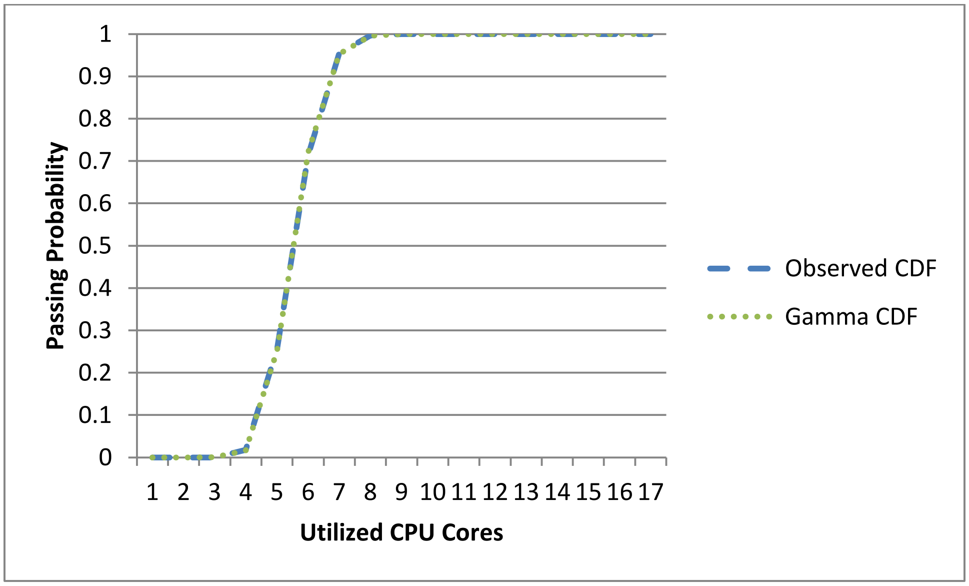


Figure 21. Cumulative Time-based Simulation Data – Observed CDF vs Gamma Distribution CDF (α = 31.727, β = 0.14377)

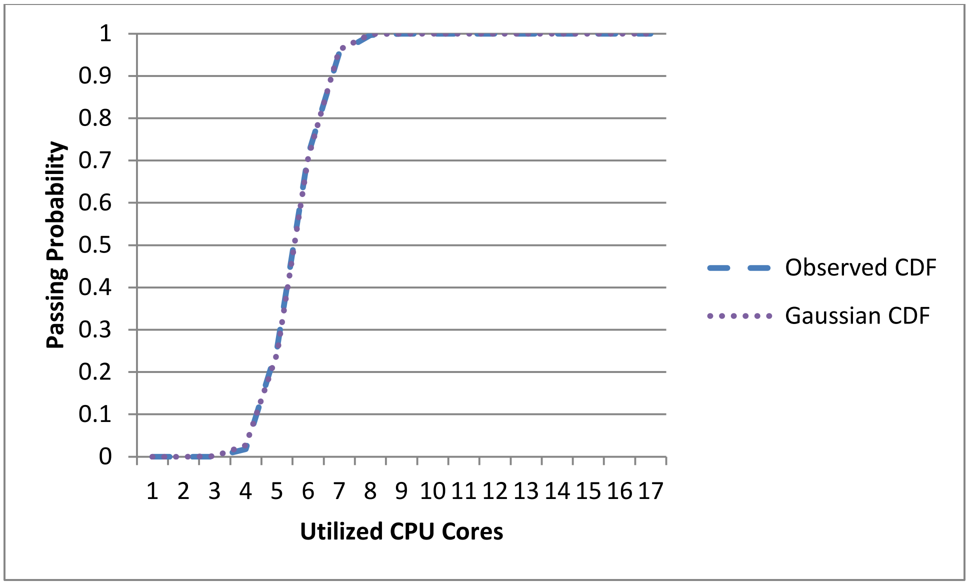


Figure 22. Cumulative Time-based Simulation Data – Observed CDF vs Gaussian-Normal Distribution CDF (μ = 4.5229, σ = 0.80984)

In order to overcome the limitations of the graphs, correlation coefficients are shown in table 7. It is shown that for cumulative data sets, all three distributions are good fits and that the Gaussian-Normal distribution provides the best fit. For the raw survey data it is shown that the Gamma distribution provides the best fit.

Table 7 : Comparison of Distribution Correlation Coefficients for Each Data Set

	Erlang CDF	Gamma CDF	Gaussian CDF
Cumulative Survey Data	0.998495255	0.999039326	0.998980715
Cumulative Replay Simulation Data	0.999155381	0.999106357	0.999349484
Cumulative App-based Simulation Data	0.999935781	0.999986389	0.999953034
Cumulative Time-based Simulation Data	0.999936780	0.999991168	0.999937383
Raw Survey Data	0.8272*	0.99664592	0.862088045

* Could not be calculated because α was too small after rounding

In trying to understand these results, it is important to recall the Central Limit Theorem. While individual application workloads do not appear to be best fit by a Gaussian-Normal distribution, the hypervisor is essentially performing convolution of the various distributions representing each individual application in the data center. As such, for a sufficiently large number of applications virtualized in a data center, the distribution will tend towards Gaussian-normal.

Contrasting with Sixty Percent Rule

The Sixty Percent Rule discussed in Chapter 1, and used by the 38th CEIG team during early planning efforts, multiplies mean utilization measured during the survey (4.24) by 1.6 then rounds up to the nearest integer. Table 8 compares.

Table 8 : Comparison of Estimated CPU Requirements

	CPU Cores Required (5 9s)	CPU Cores Required (3 9s)
Cumulative Survey Data - Gaussian	9	8
Cumulative Replay Simulation Data - Gaussian	9	8
Cumulative App-based Simulation Data - Gaussian	8	7
Cumulative Time-based Simulation Data - Gaussian	9	8
Cumulative Survey Data - Gamma	10	8
Cumulative Replay Simulation Data - Gamma	9	8
Cumulative App-based Simulation Data - Gamma	8	7
Cumulative Time-based Simulation Data - Gamma	9	8
Cumulative Survey Data - Erlang	10	8
Cumulative Replay Simulation Data - Erlang	9	8
Cumulative App-based Simulation Data - Erlang	8	7
Cumulative Time-based Simulation Data - Erlang	9	8

Cumulative Survey Data - Erlang B	14	13
Cumulative Replay Simulation Data - Erlang B	15	13
Cumulative App-based Simulation Data - Erlang B	15	13
Cumulative Time-based Simulation Data - Erlang B	15	13
Cumulative Survey Data - Erlang C	15	13
Cumulative Replay Simulation Data - Erlang C	15	14
Cumulative App-based Simulation Data - Erlang C	15	14
Cumulative Time-based Simulation Data - Erlang C	15	13
Sixty Percent Rule	7	7

Evidently, the Sixty Percent Rule matches up with Gaussian, Erlang and Gamma distribution predictions for a data center with 3 nines reliability under the Application based simulation scenario. Under all other scenarios, including calculations based off of the survey data itself, the Sixty Percent Rule under estimates the hardware requirement.

Investigative Questions Answered

1. What are the prevalent metrics for computer processing emphasized by current practice, or found in the academic body of knowledge?

    In research, the majority of the academic body of knowledge centers around optimization of data center (electrical) power, (network) ping, and (cooling) pipes. It has perhaps been the prevalence of Moore's Law, popularly thought of as processor power doubling every 18 months, that has left the topic of estimating processing requirements in neglect.

2. How well does an Erlang distribution approximate data center CPU load?

A quirk of the Erlang distribution is a parameter of the distribution must be a positive integer. During the analysis of the collected data, the Gamma distribution’s CDF (which is the same as the Erlang distribution sans the integer requirement) was used and found to strongly correlate to both the simulation and survey datasets. For the simulation datasets, with over 2000 samples, a normal distribution was found to high slightly higher correlation coefficient than a gamma distribution, 0.9993 compared to 0.9991. For the survey data, with only 27 samples, the reverse was true, gamma distribution CDF correlation coefficient 0.9990 compared to normal distribution CDF of 0.9989. This is not a significant difference in correlation. In cases where there are fewer virtual servers in the environment it should be expected for the gamma distribution to be more reliable whereas in environments with many virtual servers, such as the 70 found in the 38th ES survey and simulated as part of this thesis, the Gaussian-Normal distribution has been shown to be reliable as well.

3. How should processing in data centers or IaaS projects be sized?

Since the Erlang/Gamma distribution was shown to be a good fit for the observed data, IaaS and data center projects may use Erlang’s distribution, Gamma distribution, or Gaussian-Normal distribution to help size necessary processing capacity. In situations where a small number of servers are being virtualized, the Gamma distribution was found to be the most accurate. Where a large number of servers are being virtualized, the Gaussian-Normal distribution should be used to minimize waste capacity.

V. Conclusions and Recommendations

Chapter Overview

In this chapter, conclusions and recommendations for action and future research are presented.

Conclusions of Research

By analogy, Erlang's formulae for the probability of blocking and queuing were expected to be applicable to cumulative CPU utilization in a data center. One assumption of the Erlang formulae is that the traffic essentially follows an exponential distribution. While this was found to be true for individual applications, it was not so once the hypervisor convoluted dozens of virtual applications against limited physical resources. The research shows the distribution of cumulative CPU utilization across a large number of applications (70 application servers in the survey and simulation) can be described by the Erlang/Gamma distribution and Normal distribution. While it appears that Erlang's formulae, with minor modification discussed in chapter 3, can be used for CPU loads, a normal distribution is more accurate.

This is significant to Air Force Data Center Consolidation efforts because it allows planners to estimate how much consolidation can take place using existing resources and what new resources will be required to reach the end state. Without a method for parametric estimation of load in Air Force datacenters, planners are forced to make wild guesses at the requirement. This leads to either over purchasing hardware to avoid the risk of not having enough, which wastes money, or under purchase hardware for lack of justification, which reduces reliability and increases latency of Air Force

Enterprise applications running in the datacenter like Active Directory Services and Microsoft Exchange.

Recommendations for Action

Statistical analysis, taking advantage of either Gamma or Gaussian-Normal distribution, should be used to test if processing capacity is a likely cause of delay when sizing and troubleshooting virtual environments. As a practical example, if an availability of 5 nines is required for a base level data center then mean cumulative CPU utilization plus 5 standard deviations calculates the processing requirement before any additional redundancy requirements are introduced.

Recommendations for Future Research

Future research should include analysis of changes to cumulative CPU utilization induced by hypervisor co-scheduling processes and methods for predicting mean CPU utilization in virtual environments based on measurements of the same application on a physical host. Future research might also include investigation of the use of Markov chains as statistical models to better predict CPU utilization.

Bibliography

[1] Congressional Research Service, "Department of Defense Implementation of the Federal Data Center Consolidation Initiative: Implications for Federal Information Technology Reform Management," 23 April 2013. [Online]. Available: http://fas.org/sgp/crs/natsec/R42604.pdf. [Accessed 10 July 2014].

[2] V. Kundra, "Federal Data Center Consolidation Initiative," 26 February 2010. [Online]. Available: http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/federal_data_center_consolidation_initiative_02-26-2010.pdf. [Accessed 11 July 2014].

[3] B. H. Obama, "Memorandum on Disposing of Unneeded Federal Real Estate—Increasing Sales Proceeds, Cutting Operating Costs, and Improving Energy Efficiency," 10 June 2010. [Online]. Available: http://www.gpo.gov/fdsys/pkg/DCPD-201000483/pdf/DCPD-201000483.pdf. [Accessed 11 July 2014].

[4] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST, Gaithersburg, MD, 2011.

[5] H. Fayyad-Kazan, L. Perneel and M. Timmerman, "Benchmarking the Performance of Microsoft Hyper-V server,VMware ESXi and Xen Hypervisors," *Journal of Emerging Trends in Computing and Information Sciences,* vol. 4, no. 12, pp. 922-933, 2013.

[6] F. Malerba, R. Nelson, L. Orsenigo and S. Winter, "'History-friendly' models of industry evolution: the computer industry," *Industrial and Corporate Change,* vol. 8, no. 1, pp. 3-40, 1999.

[7] VMWare Inc., "Virtualization Basics, What is Virtualization: VMware," VMWare, 2015. [Online]. Available: http://www.vmware.com/virtualization/virtualization-basics/what-is-virtualization.

[8] M. Nelson, B.-H. Lim and G. Hutchins, "Fast transparent migration for virtual machines," *Proceedings of the annual conference on USENIX Annual Technical Conference (ATEC '05),* 2005.

[9] L. Grit, D. Irwin, A. Yumerefendi and J. Chase, "Virtual machine hosting for networked clusters: Building the foundations for "autonomic" orchestration," *VTDC '06 Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing,* vol. 0, p. 7, 2006.

[10] T. Wood, P. Shenoy, A. Venkataramani and M. Yousif, "Black-box and gray-box strategies for virtual machine migration," *Proceedings of the 4th USENIX conference on Networked systems design & implementation (NSDI'07),* 2007.

[11] Microsoft, "Customizing PRO," Microsoft Corporation, 23 October 2009. [Online]. Available: https://technet.microsoft.com/en-us/library/cc956018.aspx.

[12] H. Hussain, S. U. R. Malik, A. Hameed, S. U. Khan, G. Bickler, N. Min-Allah, M. B. Qureshi, L. Zhang and e. al, "A survey on resource allocation in high performance distributed computing systems," *Parallel Computing,* vol. 39, no. 11, pp. 709-736, 2013.

[13] C. D. Patel and A. J. Shah, "Cost Modeling for Planning Development and Operation of a Data Center," *Hewlett-Packard Laboratories Technical Report,* Vols. HPL-2005-107, 2005.

[14] T. Benson, M. Zhang, A. Anand and A. Akella, "The case for fine-grained traffic engineering in data centers," *INM/WREN'10 Proceedings of the 2010 internet network management conference on Research on enterprise networking* , p. 2, 2010.

[15] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review* , vol. 38, no. 2, pp. 69-74, 2008.

[16] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang and S. Lu, "Dcell: a scalable and fault-tolerant network structure for data centers," *ACM SIGCOMM Computer Communication Review* , vol. 38, no. 4, pp. 75-86, 2008.

[17] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel and S. Sengupta, "VL2: a scalable and flexible data center network," *Communications of the ACM* , vol. 54, no. 3, pp. 95-104, 2011.

[18] J. Devine, "An Empirical Evaluation of Methods for Improving Efficiency in Xen Virtual Machine CPU Scheduling," Allegheny College, Meadville, 2010.

[19] A. Mashtizadeh, E. Celebi, T. Garfinkel and M. Cai, "The Design and Evolution of Live Storage Migration in VMware ESX," *USENIXATC'11 Proceedings of the 2011 USENIX conference,* p. 14, 2011.

[20] H. Malik, Z. M. Jiang, B. Adams, A. E. Hassan, P. Flora and G. Hamann, "Automatic Comparison of Load Tests to Support the Performance Analysis of Large Enterprise Systems," *CSMR '10 Proceedings of the 2010 14th European Conference on Software Maintenance and Reengineering,* pp. 222-231 , 2010.

[21] W. Voorsluys, J. Broberg, S. Venugopal and R. Buyya, "Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation," *CloudCom '09 Proceedings of the 1st International Conference on Cloud Computing* , pp. 254 - 265, 2009.

[22] Z. M. Jiang, A. E. Hassan, G. Hamann and P. Flora, "Automatic Identification of Load Testing Problems," *IEEE International Conference on Software Maintenance,* pp. 307 - 316, 2008.

[23] S. Kandula, S. Sengupta, A. Greenberg and P. Patel, "The nature of data center traffic: measurements & analysis," *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference,* pp. 202-208, 2009.

[24] A. Vasan, A. Sivasubramaniam, V. Shimpim, T. Sivabalan and R. Subbiah, "Worth their Watts? - An Empirical Study of Datacenter Servers," IEEE, 2009.

[25] B. Hembree, J. Prins, P. Spagon, P. Tobias and C. Zey, Engineering Statistics Handbook, ITL - National Institute of Standards and Technology, 2003.

[26] A. K. Erlang, "Solution of some Problems in the Theory of Probabilities of Significance in Automatic Telephone Exchanges," *Elektrotkeknikeren,* vol. 13, 1917.

[27] D. Jagerman, "Some Properties of the Erlang Loss," *Bell System Technical Journal,* vol. 53, p. 525–551, 1974.

[28] E. Chromy and I. Baronak, "The first Erlang formula and traffic description in Asynchronous networks," *International Conference on Signals and Electronic Systems,* pp. 435-438, 2008.

[29] R. Bonald and J. W. Thomas, "Internet and the Erlang formula," *ACM SIGCOMM Computer Communication Review,* vol. 42, no. 1, pp. 23-30, 2012.

[30] V. Iverson, Teletraffic ENgineering Handbook, Lyngby: Technical University of Denmark, 2001.

[31] Microsoft, "Observing Processor Queue Length," 2009. [Online]. Available: http://technet.microsoft.com/en-us/library/cc940375.aspx. [Accessed 09 June 2014].

[32] Microsoft, "System Object: Core Services," Microsoft Corporation, 28 March 2003. [Online]. Available: https://technet.microsoft.com/en-us/library/cc737309%28v=ws.10%29.aspx.

[33] M. B. Friedman, "Performance by Design - Measuring Processor Utilization in Windows and Windows applications," 3 October 2011. [Online]. Available: http://performancebydesign.blogspot.com/2011/10/measuring-processor-utilization-in.html. [Accessed 2014 9 26].

[34] A. Singh, M. Korupolu and D. Mohapatra, "Server-storage virtualization: integration and load balancing in data centers," *In Proceedings of the 2008 ACM/IEEE conference on Supercomputing,* no. 53, p. 12, 2008.

[35] A. Corradi, M. Fanelli and L. Foschini, "VM consolidation: A real case based on OpenStack Cloud," *Future Generation Computer Systems,* vol. 32, no. March, pp. 118-127, 2014.

[36] F. Malerba, R. Nelson, L. Orsenigo and S. Winter, "'History-friendly' models of industry evolution: the computer industry," *Industrial and Corporate Change,* vol. 8, no. 1, pp. 3-40, 1999.

[37] J. Jiang, T. Lan, S. Ha, M. Chen and M. Chiang, "Joint VM placement and routing for data center traffic engineering," *INFOCOM, 2012 Proceedings IEEE,* pp. 2876 - 2880, 2012.

[38] M. Chen, H. Zhang, Y. Su, X. Wang, G. Jiang and K. Yoshihra, "Effective VM Sizing in Virtualized Data Centers," *Proceedings of the 12th IFIP/IEEE Symposium on Integrated Network Management*, pp. 594 - 601, 2011.

[39] W. John, S. Tafvelin and T. Olovsson, "Trends and differences in connection-behavior within classes of internet backbone traffic," in *Proceedings of the 9th international conference on Passive and active network measurement*, Berlin, 2008.

[40] P. Gum, "System/370 Extended Architecture: Facilities for Virtual Machines," November 1983. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5390401. [Accessed 14 July 2014].

Appendix A: Testbed Setup and Configuration

Cygwin was installed to allow for the use of the open source load generator LookBusy. A python script originally written by Anton Beloglazov was used to control the LookBusy load generator. Cygwin needed the Archive, Python, WEB->wget, DEVEL->gcc, and DEVEL->make packages included for successful installation and operation.

Survey Dataset

This file is used as an input to the simulation-generator.py script

LB Utilization.csv

Simulation-generator.py

```
# Copyright 2015 Derek Molle
# Usage of the works is permitted provided that this instrument is
#retained with the works, so that any entity that uses the works is
#notified of this instrument. Disclaimer: The works are without
#warranty.
#This script uses LB Utilization.csv to generate two folders of
#simulation files for use by cpu-load-generator.py: asim where a
#simulation is generated using application behavior as basis. tsim
#where a simulation is generated using data center behavior at each
#moment in time as a basis
import csv
import random
#number of simulations to generate for both application and time
#methods
numsimulations = 100
with open('LB Utilization.csv', 'rb') as surveydatafile:
      csvdata = csv.DictReader(surveydatafile)
      data = {x.strip():[y] for x,y in csvdata.next().items()}

      for rows in csvdata:
            for x,y in rows.items():
                  data[x.strip()].append(y)
      print len(data['Time'])
      timedistributions = {x.strip():[] for x in set(data['Time'])}
      appdistributions = {x.strip():[] for x in set(data['App'])}
      for t, u, a in zip(data['Time'],data['Utilization'],data['App']):
            timedistributions[t.strip()].append(u)
            appdistributions[a.strip()].append(u)

      LBAppfiles = {x.strip():[] for x in set(data['App'])}
```

```
      LBTimefiles = {x.strip():[] for x in set(data['App'])}

      for x in LBAppfiles:
            LBAppfiles[x] = [random.choice(appdistributions[x]) for _
in xrange(numsimulations*27)]

      for _ in xrange(numsimulations):
            for t in set(data['Time']):
                  for x in LBTimefiles:

      LBTimefiles[x].append(random.choice(timedistributions[t]))
      path = "asim\\"
      for x in LBAppfiles:
            file=open(path+x+".txt",'w')
            for y in LBAppfiles[x]:
                  file.write("%s\n" % y)
      path = "tsim\\"
      for x in LBTimefiles:
            file=open(path+x+".txt",'w')
            for y in LBTimefiles[x]:
                  file.write("%s\n" % y)
```

cpu-load-generator.py

```
# Copyright 2012 Anton Beloglazov
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
# See the License for the specific language governing permissions and
# limitations under the License.

""" A tool for generating a set of subsequent CPU utilization levels.
"""

from optparse import OptionParser, Option, IndentedHelpFormatter
import os
import subprocess
import time


def process(interval, utilization_list, ncpus):
    ncpus_str = str(ncpus)
    for utilization in utilization_list:
        utilization_str = str(utilization)
        print "\nSwitching to " + utilization_str + "%"
        p = subprocess.Popen(['lookbusy',
                              '--ncpus', ncpus_str,
```

```
                              '--cpu-util', utilization_str])
        time.sleep(interval)
        p.terminate()


class PosOptionParser(OptionParser):

    def format_help(self, formatter=None):
        class Positional(object):
            def __init__(self, args):
                self.option_groups = []
                self.option_list = args

        positional = Positional(self.positional)
        formatter = IndentedHelpFormatter()
        formatter.store_option_strings(positional)
        output = ['\n', formatter.format_heading('Positional
Arguments')]
        formatter.indent()
        pos_help = [formatter.format_option(opt) for opt in
self.positional]
        pos_help = [line.replace('--', '') for line in pos_help]
        output += pos_help
        return OptionParser.format_help(self, formatter) +
''.join(output)

    def add_positional_argument(self, option):
        try:
            args = self.positional
        except AttributeError:
            args = []
        args.append(option)
        self.positional = args

    def set_out(self, out):
        self.out = out


def main():
    parser = PosOptionParser(
        usage='Usage: python %prog [options] INTERVAL SOURCE',
        description='Generates a set of subsequent ' +
                    'CPU utilization levels read from a file. ' +
                    '                                         ' +
                    'Copyright (C) 2012 Anton Beloglazov. ' +
                    'Released under Apache 2.0 license.')
    parser.add_positional_argument(
        Option('--INTERVAL', action='store_true',
               help='interval between subsequent CPU ' +
               'utilization levels in seconds'))
    parser.add_positional_argument(
        Option('--SOURCE', action='store_true',
               help='source file containing a new line ' +
               'separated list of CPU utilization levels ' +
               'specified as numbers in the [0, 100] range'))
```

```
    parser.add_option('-n', '--ncpus', type='int', dest='ncpus',
default=1,
                      help='number of CPU cores to utilize [default:
1]')

    (options, args) = parser.parse_args()

    if len(args) != 2:
        parser.error('incorrect number of arguments')

    try:
        interval = int(args[0])
    except ValueError:
        parser.error('interval must be an integer >= 0')
    if interval <= 0:
        parser.error('interval must be an integer >= 0')

    filename = args[1]
    if not os.access(filename, os.R_OK):
        parser.error('cannot read file: ' + filename)

    utilization = []
    for line in open(filename):
        if line.strip():
            try:
                n = float(line)
                if n < 0 or n > 100:
                    raise ValueError
                utilization.append(int(n))
            except ValueError:
                parser.error('the source file must only ' +
                             'contain new line separated ' +
                             'numbers in the [0, 100] range')

    if interval <= 0:
        parser.error('interval must be an integer >= 0')

    process(interval, utilization, options.ncpus)


if __name__ == '__main__':
    main()
```

lb.c - Lookbusy

LB.c.docx

Appendix B: Survey Information

Filename	%util	Time	Filename	%util	Time	Filename	%util	Time	Filename	%util	Time
001v	1	0	002v	1	0	003v	0	0	004v	0	0
001v	0	30	002v	1	30	003v	0	30	004v	0	30
001v	0	60	002v	0	60	003v	0	60	004v	0	60
001v	4	90	002v	2	90	003v	0	90	004v	0	90
001v	0	120	002v	0	120	003v	0	120	004v	1	120
001v	10	150	002v	1	150	003v	0	150	004v	0	150
001v	3	180	002v	1	180	003v	0	180	004v	18	180
001v	1	210	002v	1	210	003v	0	210	004v	0	210
001v	0	240	002v	0	240	003v	0	240	004v	0	240
001v	0	270	002v	3	270	003v	0	270	004v	0	270
001v	1	300	002v	0	300	003v	0	300	004v	0	300
001v	0	330	002v	0	330	003v	0	330	004v	0	330
001v	0	360	002v	0	360	003v	0	360	004v	2	360
001v	0	390	002v	0	390	003v	0	390	004v	0	390
001v	0	420	002v	1	420	003v	0	420	004v	1	420
001v	0	450	002v	1	450	003v	0	450	004v	0	450
001v	0	480	002v	1	480	003v	0	480	004v	1	480
001v	0	510	002v	1	510	003v	0	510	004v	0	510
001v	0	540	002v	0	540	003v	0	540	004v	0	540
001v	0	570	002v	1	570	003v	0	570	004v	0	570
001v	0	600	002v	0	600	003v	0	600	004v	1	600
001v	0	630	002v	2	630	003v	0	630	004v	0	630
001v	1	660	002v	0	660	003v	0	660	004v	0	660
001v	4	690	002v	0	690	003v	0	690	004v	0	690
001v	0	720	002v	1	720	003v	0	720	004v	1	720
001v	0	750	002v	1	750	003v	0	750	004v	1	750
001v	0	780	002v	0	780	003v	0	780	004v	0	780
005v	0	0	002v	0	810	003v	0	810	004v	0	810
005v	0	30	006v	0	0	007v	1	0	008v	0	0
005v	0	60	006v	0	30	007v	28	30	008v	0	30
005v	0	90	006v	0	60	007v	26	60	008v	0	60
005v	1	120	006v	0	90	007v	24	90	008v	32	90
005v	0	150	006v	0	120	007v	2	120	008v	0	120
005v	0	180	006v	0	150	007v	3	150	008v	56	150
005v	1	210	006v	0	180	007v	29	180	008v	29	180
005v	1	240	006v	0	210	007v	8	210	008v	55	210
005v	0	270	006v	51	240	007v	31	240	008v	1	240

005v	0	300	006v	2	270	007v	5	270	008v	32	270
005v	0	330	006v	0	300	007v	1	300	008v	28	300
005v	0	360	006v	0	330	007v	28	330	008v	43	330
005v	0	390	006v	0	360	007v	57	360	008v	29	360
005v	1	420	006v	0	390	007v	56	390	008v	29	390
005v	0	450	006v	0	420	007v	31	420	008v	28	420
005v	20	480	006v	1	450	007v	28	450	008v	30	450
005v	0	510	006v	0	480	007v	3	480	008v	33	480
005v	0	540	006v	0	510	007v	0	510	008v	55	510
005v	0	570	006v	2	540	007v	4	540	008v	28	540
005v	0	600	006v	0	570	007v	0	570	008v	3	570
005v	0	630	006v	0	600	007v	2	600	008v	27	600
005v	0	660	006v	0	630	007v	0	630	008v	28	630
005v	0	690	006v	0	660	007v	0	660	008v	29	660
005v	0	720	006v	0	690	007v	0	690	008v	33	690
005v	1	750	006v	0	720	007v	3	720	008v	26	720
005v	0	780	006v	3	750	007v	28	750	008v	28	750
005v	1	810	006v	0	780	007v	0	780	008v	26	780
009v	0	0	006v	19	810	011v	0	0	008v	28	810
009v	2	30	010v	0	0	011v	0	30	012v	55	0
009v	1	60	010v	0	30	011v	0	60	012v	0	30
009v	0	90	010v	1	60	011v	1	90	012v	0	60
009v	0	120	010v	1	90	011v	1	120	012v	0	90
009v	17	150	010v	4	120	011v	0	150	012v	0	120
009v	0	180	010v	0	150	011v	0	180	012v	0	150
009v	0	210	010v	0	180	011v	0	210	012v	0	180
009v	0	240	010v	1	210	011v	0	240	012v	0	210
009v	0	270	010v	0	240	011v	0	270	012v	0	240
009v	0	300	010v	0	270	011v	1	300	012v	0	270
009v	0	330	010v	0	300	011v	0	330	012v	0	300
009v	0	360	010v	0	330	011v	0	360	012v	0	330
009v	2	390	010v	0	360	011v	1	390	012v	0	360
009v	1	420	010v	0	390	011v	0	420	012v	3	390
009v	0	450	010v	0	420	011v	0	450	012v	0	420
009v	0	480	010v	0	450	011v	0	480	012v	12	450
009v	0	510	010v	1	480	011v	0	510	012v	0	480
009v	0	540	010v	0	510	011v	0	540	012v	0	510
009v	0	570	010v	0	540	011v	0	570	012v	0	540
009v	0	600	010v	0	570	011v	0	600	012v	0	570
009v	0	630	010v	0	600	011v	0	630	012v	0	600
009v	0	660	010v	0	630	011v	0	660	012v	0	630

009v	1	690	010v	0	660	011v	1	690	012v	2	660
009v	0	720	010v	0	690	011v	5	720	012v	0	690
009v	0	750	010v	0	720	011v	0	750	012v	0	720
009v	0	780	010v	10	750	011v	0	780	012v	0	750
009v	0	810	010v	1	780	015v	13	0	012v	0	780
013v	0	0	014v	7	0	015v	0	30	016v	7	0
013v	0	30	014v	4	30	015v	0	60	016v	7	30
013v	0	60	014v	3	60	015v	0	90	016v	45	60
013v	0	90	014v	0	90	015v	0	120	016v	7	90
013v	0	120	014v	3	120	015v	0	150	016v	36	120
013v	0	150	014v	3	150	015v	0	180	016v	1	150
013v	0	180	014v	0	180	015v	0	210	016v	10	180
013v	0	210	014v	7	210	015v	1	240	016v	42	210
013v	0	240	014v	37	240	015v	0	270	016v	0	240
013v	0	270	014v	0	270	015v	0	300	016v	2	270
013v	0	300	014v	4	300	015v	0	330	016v	37	300
013v	0	330	014v	100	330	015v	0	360	016v	39	330
013v	0	360	014v	3	360	015v	0	390	016v	31	360
013v	0	390	014v	12	390	015v	0	420	016v	27	390
013v	0	420	014v	1	420	015v	0	450	016v	46	420
013v	1	450	014v	1	450	015v	0	480	016v	1	450
013v	0	480	014v	0	480	015v	1	510	016v	34	480
013v	0	510	014v	0	510	015v	0	540	016v	30	510
013v	0	540	014v	0	540	015v	0	570	016v	35	540
013v	0	570	014v	4	570	015v	0	600	016v	19	570
013v	1	600	014v	0	600	015v	0	630	016v	2	600
013v	0	630	014v	9	630	015v	0	660	016v	15	630
013v	0	660	014v	0	660	015v	0	690	016v	13	660
013v	0	690	014v	0	690	015v	0	720	016v	16	690
013v	7	720	014v	0	720	015v	0	750	016v	14	720
013v	0	750	014v	0	750	015v	0	780	016v	24	750
017v	13	0	014v	3	780	019v	0	0	016v	10	780
017v	28	30	018v	0	0	019v	0	30	020v	0	0
017v	32	60	018v	0	30	019v	0	60	020v	0	30
017v	17	90	018v	0	60	019v	0	90	020v	2	60
017v	17	120	018v	1	90	019v	0	120	020v	0	90
017v	1	150	018v	0	120	019v	0	150	020v	1	120
017v	22	180	018v	30	150	019v	0	180	020v	1	150
017v	40	210	018v	0	180	019v	0	210	020v	0	180
017v	44	240	018v	0	210	019v	0	240	020v	0	210
017v	24	270	018v	0	240	019v	0	270	020v	0	240

017v	35	300	018v	0	270	019v	0	300	020v	3	270
017v	12	330	018v	0	300	019v	0	330	020v	0	300
017v	6	360	018v	0	330	019v	0	360	020v	6	330
017v	18	390	018v	0	360	019v	0	390	020v	0	360
017v	5	420	018v	0	390	019v	0	420	020v	0	390
017v	22	450	018v	0	420	019v	0	450	020v	0	420
017v	28	480	018v	0	450	019v	0	480	020v	0	450
017v	29	510	018v	0	480	019v	0	510	020v	0	480
017v	15	540	018v	0	510	019v	0	540	020v	0	510
017v	50	570	018v	0	540	019v	0	570	020v	0	540
017v	51	600	018v	0	570	019v	0	600	020v	0	570
017v	55	630	018v	0	600	019v	0	630	020v	0	600
017v	37	660	018v	0	630	019v	0	660	020v	0	630
017v	33	690	018v	0	660	019v	0	690	020v	0	660
017v	35	720	018v	0	690	019v	0	720	020v	0	690
017v	18	750	018v	0	720	019v	0	750	020v	0	720
017v	48	780	018v	0	750	019v	0	780	020v	0	750
017v	26	810	018v	0	780	023v	0	0	020v	0	780
021v	5	0	022v	8	0	023v	73	30	020v	0	810
021v	2	30	022v	3	30	023v	60	60	024v	14	0
021v	5	60	022v	25	60	023v	0	90	024v	6	30
021v	1	90	022v	2	90	023v	78	120	024v	30	60
021v	2	120	022v	28	120	023v	61	150	024v	35	90
021v	2	150	022v	24	150	023v	63	180	024v	12	120
021v	1	180	022v	1	180	023v	74	210	024v	21	150
021v	2	210	022v	5	210	023v	61	240	024v	3	180
021v	11	240	022v	6	240	023v	0	270	024v	0	210
021v	5	270	022v	32	270	023v	63	300	024v	28	240
021v	1	300	022v	5	300	023v	0	330	024v	9	270
021v	6	330	022v	2	330	023v	67	360	024v	7	300
021v	3	360	022v	17	360	023v	5	390	024v	2	330
021v	2	390	022v	25	390	023v	48	420	024v	17	360
021v	8	420	022v	28	420	023v	0	450	024v	15	390
021v	2	450	022v	13	450	023v	65	480	024v	22	420
021v	1	480	022v	6	480	023v	62	510	024v	3	450
021v	3	510	022v	15	510	023v	0	540	024v	0	480
021v	0	540	022v	5	540	023v	1	570	024v	2	510
021v	8	570	022v	6	570	023v	0	600	024v	4	540
021v	9	600	022v	24	600	023v	0	630	024v	1	570
021v	4	630	022v	9	630	023v	0	660	024v	1	600
021v	0	660	022v	3	660	023v	6	690	024v	3	630

021v	0	690	022v	3	690	023v	0	720	024v	20	660
021v	4	720	022v	24	720	023v	60	750	024v	6	690
021v	3	750	022v	0	750	023v	76	780	024v	10	720
021v	1	780	022v	23	780	028v	10	0	024v	39	750
025v	0	0	027v	0	0	028v	0	30	024v	49	780
025v	6	30	027v	1	30	028v	5	60	029v	0	0
025v	0	60	027v	1	60	028v	2	90	029v	0	30
025v	0	90	027v	4	90	028v	0	120	029v	0	60
025v	0	120	027v	0	120	028v	18	150	029v	0	90
025v	0	150	027v	3	150	028v	1	180	029v	0	120
025v	3	180	027v	4	180	028v	1	210	029v	0	150
025v	0	210	027v	0	210	028v	1	240	029v	0	180
025v	0	240	027v	0	240	028v	58	270	029v	0	210
025v	85	270	027v	17	270	028v	22	300	029v	0	240
025v	0	300	027v	0	300	028v	1	330	029v	0	270
025v	0	330	027v	6	330	028v	8	360	029v	0	300
025v	0	360	027v	2	360	028v	1	390	029v	0	330
025v	0	390	027v	1	390	028v	2	420	029v	0	360
025v	0	420	027v	1	420	028v	5	450	029v	0	390
025v	0	450	027v	2	450	028v	8	480	029v	0	420
025v	1	480	027v	1	480	028v	11	510	029v	0	450
025v	0	510	027v	1	510	028v	6	540	029v	0	480
025v	0	540	027v	1	540	028v	3	570	029v	0	510
025v	0	570	027v	0	570	028v	10	600	029v	0	540
025v	1	600	027v	6	600	028v	3	630	029v	0	570
025v	7	630	027v	3	630	028v	5	660	029v	0	600
025v	1	660	027v	0	660	028v	4	690	029v	0	630
025v	0	690	027v	3	690	028v	11	720	029v	0	660
025v	0	720	027v	1	720	028v	6	750	029v	0	690
025v	0	750	027v	1	750	028v	3	780	029v	0	720
025v	17	780	027v	1	780	032v	2	0	029v	0	750
030v	0	0	031v	22	0	032v	0	30	029v	0	780
030v	0	30	031v	1	30	032v	1	60	033v	0	0
030v	0	60	031v	1	60	032v	38	90	033v	1	30
030v	0	90	031v	8	90	032v	3	120	033v	0	60
030v	0	120	031v	9	120	032v	9	150	033v	6	90
030v	0	150	031v	2	150	032v	13	180	033v	0	120
030v	0	180	031v	5	180	032v	1	210	033v	0	150
030v	0	210	031v	3	210	032v	39	240	033v	0	180
030v	17	240	031v	18	240	032v	1	270	033v	0	210
030v	0	270	031v	7	270	032v	52	300	033v	2	240

030v	0	300	031v	14	300	032v	80	330	033v	1	270
030v	0	330	031v	12	330	032v	67	360	033v	55	300
030v	0	360	031v	4	360	032v	3	390	033v	2	330
030v	0	390	031v	3	390	032v	0	420	033v	0	360
030v	0	420	031v	1	420	032v	4	450	033v	2	390
030v	0	450	031v	5	450	032v	10	480	033v	0	420
030v	0	480	031v	7	480	032v	1	510	033v	11	450
030v	0	510	031v	4	510	032v	2	540	033v	0	480
030v	0	540	031v	10	540	032v	1	570	033v	0	510
030v	0	570	031v	5	570	032v	1	600	033v	1	540
030v	0	600	031v	1	600	032v	48	630	033v	0	570
030v	0	630	031v	1	630	032v	33	660	033v	3	600
030v	0	660	031v	5	660	032v	86	690	033v	86	630
030v	0	690	031v	3	690	032v	77	720	033v	31	660
030v	0	720	031v	3	720	032v	0	750	033v	0	690
030v	0	750	031v	22	750	032v	5	780	033v	3	720
030v	0	780	031v	5	780	032v	30	810	033v	48	750
034v	0	0	035v	26	0	036v	9	0	033v	34	780
034v	0	30	035v	0	30	036v	10	30	037v	0	0
034v	0	60	035v	1	60	036v	17	60	037v	0	30
034v	0	90	035v	2	90	036v	10	90	037v	0	60
034v	0	120	035v	5	120	036v	3	120	037v	6	90
034v	65	150	035v	0	150	036v	2	150	037v	0	120
034v	5	180	035v	0	180	036v	13	180	037v	0	150
034v	4	210	035v	6	210	036v	10	210	037v	0	180
034v	0	240	035v	0	240	036v	13	240	037v	0	210
034v	0	270	035v	1	270	036v	0	270	037v	0	240
034v	48	300	035v	0	300	036v	12	300	037v	0	270
034v	0	330	035v	0	330	036v	1	330	037v	2	300
034v	0	360	035v	8	360	036v	2	360	037v	0	330
034v	0	390	035v	24	390	036v	18	390	037v	0	360
034v	1	420	035v	5	420	036v	29	420	037v	0	390
034v	5	450	035v	21	450	036v	6	450	037v	0	420
034v	0	480	035v	17	480	036v	5	480	037v	0	450
034v	0	510	035v	12	510	036v	21	510	037v	1	480
034v	0	540	035v	14	540	036v	13	540	037v	0	510
034v	0	570	035v	13	570	036v	4	570	037v	0	540
034v	0	600	035v	13	600	036v	5	600	037v	0	570
034v	34	630	035v	13	630	036v	4	630	037v	0	600
034v	0	660	035v	12	660	036v	11	660	037v	1	630
034v	0	690	035v	23	690	036v	0	690	037v	0	660

034v	7	720	035v	25	720	036v	6	720	037v	0	690
034v	20	750	035v	27	750	036v	16	750	037v	0	720
034v	41	780	035v	22	780	036v	1	780	037v	0	750
038v	0	0	039v	0	0	036v	6	810	037v	0	780
038v	0	30	039v	0	30	040v	1	0	037v	0	810
038v	0	60	039v	0	60	040v	1	30	041v	0	0
038v	0	90	039v	0	90	040v	0	60	041v	0	30
038v	1	120	039v	0	120	040v	0	90	041v	0	60
038v	0	150	039v	1	150	040v	0	120	041v	0	90
038v	3	180	039v	3	180	040v	0	150	041v	0	120
038v	0	210	039v	0	210	040v	1	180	041v	0	150
038v	2	240	039v	3	240	040v	0	210	041v	0	180
038v	0	270	039v	0	270	040v	0	240	041v	0	210
038v	0	300	039v	1	300	040v	1	270	041v	3	240
038v	0	330	039v	0	330	040v	0	300	041v	9	270
038v	2	360	039v	0	360	040v	0	330	041v	2	300
038v	4	390	039v	0	390	040v	1	360	041v	4	330
038v	0	420	039v	1	420	040v	0	390	041v	0	360
038v	1	450	039v	1	450	040v	0	420	041v	0	390
038v	0	480	039v	1	480	040v	0	450	041v	0	420
038v	1	510	039v	0	510	040v	1	480	041v	0	450
038v	0	540	039v	0	540	040v	0	510	041v	12	480
038v	1	570	039v	0	570	040v	0	540	041v	1	510
038v	1	600	039v	0	600	040v	0	570	041v	0	540
038v	2	630	039v	1	630	040v	0	600	041v	1	570
038v	0	660	039v	0	660	040v	0	630	041v	0	600
038v	0	690	039v	1	690	040v	0	660	041v	1	630
038v	0	720	039v	0	720	040v	0	690	041v	1	660
038v	1	750	039v	1	750	040v	0	720	041v	1	690
038v	1	780	039v	1	780	040v	0	750	041v	0	720
042v	4	0	043v	8	0	040v	0	780	041v	0	750
042v	1	30	043v	0	30	044v	3	0	041v	0	780
042v	1	60	043v	0	60	044v	0	30	045v	1	0
042v	1	90	043v	0	90	044v	0	60	045v	0	30
042v	0	120	043v	1	120	044v	0	90	045v	0	60
042v	0	150	043v	0	150	044v	0	120	045v	0	90
042v	0	180	043v	1	180	044v	1	150	045v	0	120
042v	0	210	043v	2	210	044v	0	180	045v	0	150
042v	4	240	043v	3	240	044v	2	210	045v	0	180
042v	2	270	043v	5	270	044v	3	240	045v	0	210
042v	1	300	043v	0	300	044v	4	270	045v	8	240

042v	0	330	043v	12	330	044v	1	300	045v	5	270
042v	2	360	043v	0	360	044v	12	330	045v	2	300
042v	13	390	043v	1	390	044v	0	360	045v	5	330
042v	3	420	043v	1	420	044v	0	390	045v	1	360
042v	0	450	043v	0	450	044v	0	420	045v	0	390
042v	1	480	043v	12	480	044v	0	450	045v	0	420
042v	1	510	043v	0	510	044v	7	480	045v	0	450
042v	0	540	043v	0	540	044v	1	510	045v	0	480
042v	2	570	043v	0	570	044v	0	540	045v	1	510
042v	7	600	043v	0	600	044v	1	570	045v	0	540
042v	1	630	043v	0	630	044v	0	600	045v	0	570
042v	1	660	043v	0	660	044v	0	630	045v	5	600
042v	0	690	043v	0	690	044v	0	660	045v	5	630
042v	1	720	043v	0	720	044v	0	690	045v	7	660
042v	6	750	043v	0	750	044v	0	720	045v	6	690
042v	6	780	043v	0	780	044v	0	750	045v	0	720
046v	1	0	047v	1	0	044v	0	780	045v	6	750
046v	3	30	047v	0	30	048v	5	0	045v	0	780
046v	2	60	047v	0	60	048v	0	30	049v	0	0
046v	0	90	047v	0	90	048v	3	60	049v	0	30
046v	0	120	047v	0	120	048v	2	90	049v	17	60
046v	0	150	047v	0	150	048v	34	120	049v	1	90
046v	1	180	047v	0	180	048v	69	150	049v	50	120
046v	0	210	047v	0	210	048v	0	180	049v	31	150
046v	9	240	047v	0	240	048v	0	210	049v	0	180
046v	6	270	047v	0	270	048v	3	240	049v	1	210
046v	4	300	047v	0	300	048v	0	270	049v	0	240
046v	2	330	047v	0	330	048v	0	300	049v	3	270
046v	0	360	047v	0	360	048v	0	330	049v	0	300
046v	2	390	047v	0	390	048v	3	360	049v	1	330
046v	0	420	047v	0	420	048v	0	390	049v	0	360
046v	5	450	047v	0	450	048v	0	420	049v	3	390
046v	0	480	047v	0	480	048v	0	450	049v	3	420
046v	0	510	047v	0	510	048v	0	480	049v	14	450
046v	0	540	047v	0	540	048v	0	510	049v	99	480
046v	0	570	047v	0	570	048v	0	540	049v	51	510
046v	5	600	047v	0	600	048v	0	570	049v	50	540
046v	6	630	047v	0	630	048v	0	600	049v	50	570
046v	7	660	047v	0	660	048v	0	630	049v	51	600
046v	7	690	047v	0	690	048v	0	660	049v	50	630
046v	0	720	047v	0	720	048v	0	690	049v	51	660

046v	6	750	047v	0	750	048v	0	720	049v	50	690
046v	0	780	047v	0	780	048v	0	750	049v	1	720
050v	4	0	051v	0	0	048v	0	780	049v	50	750
050v	3	30	051v	0	30	052v	0	0	049v	49	780
050v	1	60	051v	0	60	052v	0	30	053v	0	0
050v	1	90	051v	0	90	052v	0	60	053v	0	30
050v	38	120	051v	0	120	052v	11	90	053v	0	60
050v	39	150	051v	0	150	052v	0	120	053v	1	90
050v	0	180	051v	1	180	052v	0	150	053v	0	120
050v	0	210	051v	0	210	052v	0	180	053v	0	150
050v	2	240	051v	0	240	052v	0	210	053v	0	180
050v	0	270	051v	0	270	052v	0	240	053v	0	210
050v	0	300	051v	0	300	052v	1	270	053v	0	240
050v	1	330	051v	0	330	052v	5	300	053v	0	270
050v	3	360	051v	0	360	052v	0	330	053v	0	300
050v	3	390	051v	0	390	052v	6	360	053v	1	330
050v	4	420	051v	1	420	052v	0	390	053v	2	360
050v	12	450	051v	0	450	052v	4	420	053v	0	390
050v	98	480	051v	0	480	052v	0	450	053v	0	420
050v	50	510	051v	0	510	052v	0	480	053v	0	450
050v	51	540	051v	0	540	052v	10	510	053v	3	480
050v	52	570	051v	1	570	052v	14	540	053v	0	510
050v	53	600	051v	0	600	052v	0	570	053v	0	540
050v	51	630	051v	0	630	052v	0	600	053v	0	570
050v	50	660	051v	0	660	052v	8	630	053v	0	600
050v	52	690	051v	0	690	052v	1	660	053v	0	630
050v	2	720	051v	0	720	052v	8	690	053v	0	660
050v	50	750	051v	0	750	052v	2	720	053v	0	690
050v	49	780	055v	0	0	052v	1	750	053v	0	720
054v	0	0	055v	0	30	052v	10	780	053v	6	750
054v	0	30	055v	1	60	052v	2	810	053v	0	780
054v	1	60	055v	2	90	056v	0	0	057v	1	0
054v	8	90	055v	3	120	056v	0	30	057v	0	30
054v	0	120	055v	2	150	056v	0	60	057v	2	60
054v	0	150	055v	0	180	056v	0	90	057v	0	90
054v	9	180	055v	0	210	056v	0	120	057v	1	120
054v	0	210	055v	5	240	056v	0	150	057v	5	150
054v	2	240	055v	3	270	056v	0	180	057v	0	180
054v	0	270	055v	46	300	056v	0	210	057v	0	210
054v	1	300	055v	1	330	056v	28	240	057v	7	240
054v	1	330	055v	10	360	056v	0	270	057v	1	270

054v	3	360	055v	1	390	056v	0	300	057v	13	300
054v	0	390	055v	29	420	056v	0	330	057v	2	330
054v	0	420	055v	1	450	056v	0	360	057v	0	360
054v	6	450	055v	0	480	056v	1	390	057v	1	390
054v	3	480	055v	1	510	056v	0	420	057v	1	420
054v	1	510	055v	3	540	056v	0	450	057v	1	450
054v	0	540	055v	1	570	056v	0	480	057v	1	480
054v	0	570	055v	1	600	056v	0	510	057v	0	510
054v	0	600	055v	2	630	056v	0	540	057v	25	540
054v	5	630	055v	0	660	056v	0	570	057v	1	570
054v	0	660	055v	0	690	056v	0	600	057v	50	600
054v	1	690	055v	1	720	056v	0	630	057v	0	630
054v	0	720	055v	10	750	056v	0	660	057v	6	660
054v	8	750	055v	1	780	056v	0	690	057v	1	690
054v	0	780	055v	13	810	056v	0	720	057v	3	720
054v	0	810	059v	0	0	056v	0	750	057v	1	750
058v	1	0	059v	25	30	056v	0	780	057v	1	780
058v	0	30	059v	1	60	056v	0	810	057v	1	810
058v	0	60	059v	0	90	060v	0	0	061v	15	0
058v	1	90	059v	0	120	060v	4	30	061v	5	30
058v	5	120	059v	0	150	060v	0	60	061v	1	60
058v	2	150	059v	0	180	060v	0	90	061v	0	90
058v	0	180	059v	0	210	060v	0	120	061v	0	120
058v	0	210	059v	0	240	060v	15	150	061v	0	150
058v	1	240	059v	0	270	060v	0	180	061v	11	180
058v	0	270	059v	1	300	060v	2	210	061v	1	210
058v	2	300	059v	3	330	060v	0	240	061v	1	240
058v	2	330	059v	0	360	060v	0	270	061v	2	270
058v	0	360	059v	0	390	060v	14	300	061v	1	300
058v	45	390	059v	0	420	060v	0	330	061v	0	330
058v	0	420	059v	0	450	060v	4	360	061v	0	360
058v	0	450	059v	0	480	060v	1	390	061v	0	390
058v	1	480	059v	0	510	060v	2	420	061v	1	420
058v	0	510	059v	0	540	060v	4	450	061v	11	450
058v	2	540	059v	0	570	060v	0	480	061v	0	480
058v	0	570	059v	4	600	060v	11	510	061v	13	510
058v	0	600	059v	0	630	060v	0	540	065v	42	0
058v	2	630	059v	0	660	064v	29	0	065v	32	30
058v	6	660	059v	0	690	064v	23	30	065v	24	60
058v	0	690	059v	0	720	064v	66	60	065v	20	90
058v	0	720	059v	0	750	064v	27	90	065v	19	120

058v	1	750	059v	0	780	064v	18	120	065v	15	150
058v	0	780	059v	0	810	064v	18	150	065v	7	180
058v	0	810	063v	46	0	064v	17	180	065v	9	210
061v	15	0	063v	34	30	064v	16	210	065v	27	240
061v	5	30	063v	23	60	064v	23	240	065v	7	270
061v	1	60	063v	25	90	064v	19	270	065v	16	300
061v	0	90	063v	19	120	064v	26	300	065v	8	330
061v	0	120	063v	19	150	064v	8	330	065v	45	360
061v	0	150	063v	10	180	064v	32	360	065v	16	390
061v	11	180	063v	18	210	064v	21	390	065v	14	420
061v	1	210	063v	25	240	064v	13	420	065v	15	450
061v	1	240	063v	14	270	064v	18	450	065v	8	480
061v	2	270	063v	22	300	064v	19	480	065v	7	510
061v	1	300	063v	23	330	064v	6	510	065v	8	540
061v	0	330	063v	21	360	064v	9	540	065v	9	570
061v	0	360	063v	36	390	064v	16	570	065v	7	600
061v	0	390	063v	20	420	064v	24	600	065v	20	630
061v	1	420	063v	15	450	064v	41	630	065v	6	660
061v	11	450	063v	15	480	064v	14	660	065v	12	690
061v	0	480	063v	22	510	064v	26	690	065v	17	720
061v	13	510	063v	20	540	064v	15	720	065v	20	750
062v	1	0	067v	0	0	064v	11	750	065v	19	780
062v	0	30	067v	0	30	064v	6	780	069v	55	0
062v	0	60	067v	0	60	068v	5	0	069v	0	30
062v	0	90	067v	1	90	068v	2	30	069v	0	60
062v	0	120	067v	1	120	068v	1	60	069v	0	90
062v	0	150	067v	16	150	068v	2	90	069v	13	120
062v	0	180	067v	0	180	068v	0	120	069v	0	150
062v	0	210	067v	0	210	068v	0	150	069v	1	180
062v	1	240	067v	0	240	068v	1	180	069v	0	210
062v	0	270	067v	0	270	068v	1	210	069v	0	240
062v	0	300	067v	0	300	068v	3	240	069v	0	270
062v	1	330	067v	0	330	068v	0	270	069v	0	300
062v	0	360	067v	0	360	068v	0	300	069v	0	330
062v	1	390	067v	0	390	068v	3	330	069v	0	360
062v	0	420	067v	0	420	068v	1	360	069v	1	390
062v	0	450	067v	0	450	068v	1	390	069v	2	420
062v	0	480	067v	0	480	068v	0	420	069v	0	450
062v	0	510	067v	0	510	068v	0	450	069v	0	480
066v	16	0	067v	0	540	068v	1	480	069v	0	510
066v	24	30	067v	1	570	068v	5	510			

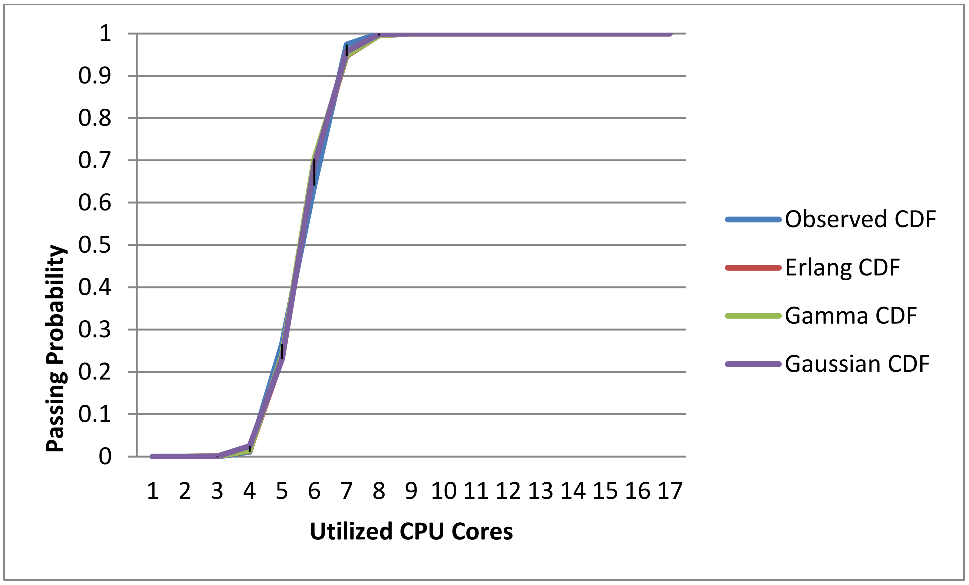
066v	25	60
066v	15	90
066v	26	120
066v	15	150
066v	9	180
066v	19	210
066v	23	240
066v	10	270
066v	28	300
066v	9	330
066v	18	360
066v	23	390
066v	28	420
066v	42	450
066v	24	480
066v	16	510
066v	11	540
066v	24	570
066v	20	600
066v	13	630
066v	14	660
066v	14	690
066v	14	720
066v	24	750
066v	13	780
070v	0	0
070v	0	30
070v	1	60
070v	0	90
070v	0	120
070v	5	150
070v	0	180
070v	0	210
070v	7	240
070v	0	270
070v	1	300
070v	2	330
070v	0	360
070v	2	390
070v	1	420
070v	2	450
067v	0	600
067v	0	630
067v	0	660
067v	1	690
067v	1	720
067v	0	750
067v	1	780
067v	2	810
068v	2	540
068v	2	570
068v	0	600
068v	0	630
068v	0	660
068v	0	690
068v	3	720
068v	6	750
068v	5	780

070v	1	480
070v	0	510
070v	0	540
070v	0	570
070v	0	600
070v	0	630
070v	0	660
070v	0	690
070v	0	720
070v	0	750
070v	1	780

Appendix C : CDF Tables

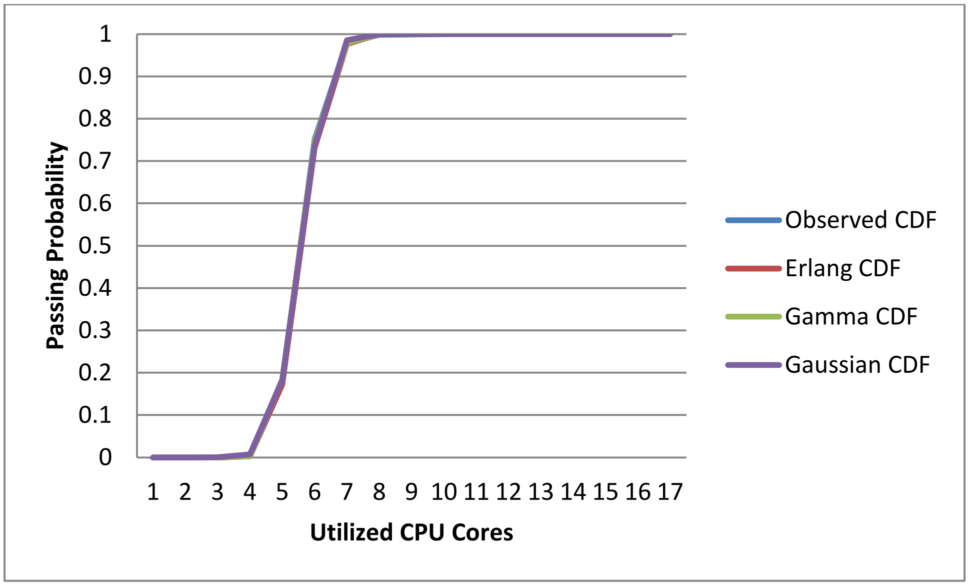
Cumulative Replay Simulation Data

Cores in Use	Observed CDF	Erlang CDF	Gamma CDF	Gaussian CDF
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0.0007
3	0.0112	0.0135	0.0142	0.0247
4	0.2660	0.2328	0.2384	0.2307
5	0.6400	0.6981	0.7041	0.6884
6	0.9740	0.9464	0.9482	0.9572
7	0.9996	0.9952	0.9954	0.9984
8	0.9996	0.9998	0.9998	1
9	0.9996	1	1	1
10	0.9996	1	1	1
11	0.9996	1	1	1
12	0.9996	1	1	1
13	0.9996	1	1	1
14	0.9996	1	1	1
15	1	1	1	1
16	1	1	1	1



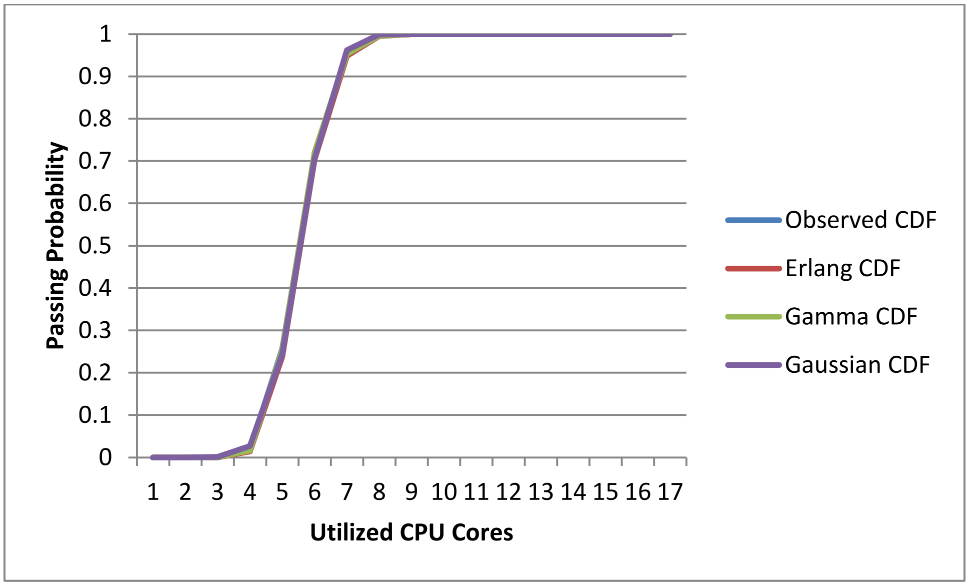
Cumulative App-based Simulation Data

Cores in Use	Observed CDF	Erlang CDF	Gamma CDF	Gaussian CDF
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0.0000
3	0.0033	0.0026	0.0030	0.0073
4	0.1766	0.1718	0.1845	0.1829
5	0.7508	0.7302	0.7455	0.7361
6	0.9791	0.9761	0.9785	0.9849
7	0.9981	0.9993	0.9994	0.9999
8	0.9993	1.0000	1.0000	1
9	0.9998	1	1	1
10	1.0000	1	1	1
11	1.0000	1	1	1
12	1.0000	1	1	1
13	1.0000	1	1	1
14	1.0000	1	1	1
15	1	1	1	1
16	1	1	1	1



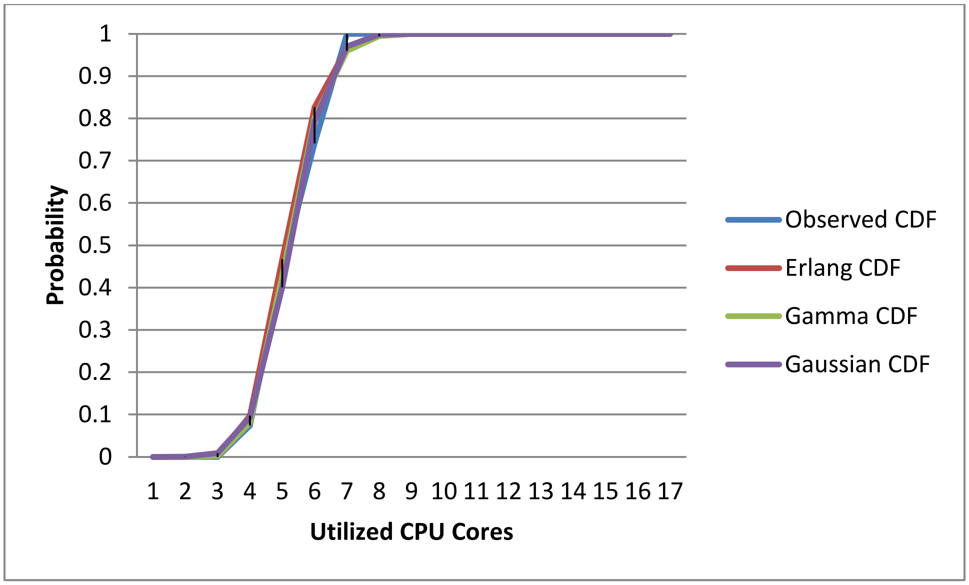
Cumulative Time-based Simulation Data

Cores in Use	Observed CDF	Erlang CDF	Gamma CDF	Gaussian CDF
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0.0008
3	0.0176	0.0140	0.0160	0.0269
4	0.2568	0.2376	0.2533	0.2440
5	0.7139	0.7040	0.7203	0.7059
6	0.9539	0.9484	0.9531	0.9621
7	0.9967	0.9955	0.9960	0.9987
8	0.9996	0.9998	0.9998	1
9	1.0000	1	1	1
10	1.0000	1	1	1
11	1.0000	1	1	1
12	1.0000	1	1	1
13	1.0000	1	1	1
14	1.0000	1	1	1
15	1	1	1	1
16	1	1	1	1



Cumulative Survey Data

Cores in Use	Observed CDF	Erlang CDF	Gamma CDF	Gaussian CDF
0	0	0	0	0
1	0	0	0	0
2	0	0.0023	0.0016	0.0085
3	0.0741	0.0976	0.0809	0.0936
4	0.4444	0.4667	0.4278	0.4007
5	0.7407	0.8262	0.8007	0.7927
6	1	0.9667	0.9593	0.9702
7	1	0.9958	0.9946	0.9984
8	1	0.9996	0.9995	1
9	1	1	1	1
10	1	1	1	1
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1



Vita

Derek Molle was born in Omaha, Nebraska where he lived his early life, graduated from High School in 2006, and graduated from the University of Nebraska Lincoln at Omaha in 2010. While at the University of Nebraska, he began working on a Bachelor of Science in Computer Engineering only to leave the institution four years later with a Bachelor of Science in Electronics Engineering and a wink of knowledge from both fields. After completing the University, Derek was employed by Department of the Air Force as a civil servant.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 27-03-2015	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From – To)* June 2013 – March 2015

TITLE AND SUBTITLE Parametric Estimation of Load for Air Force Datacenters	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Molle, Derek P., Civ, USAF	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENV)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENV-MS-15-M-170

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Vinh Phung, 38ES/ENOC
5813 Arnold St, Building 4064
Tinker AFB OK 73145-8120
COM: 405-734-7461, vinh.phung@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)
38ES/ENOC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The Office of Management and Budget (OMB) has tasked Federal agencies to develop a Data Center Consolidation Plan. Effective planning requires a repeatable method to effectively and efficiently size Air Force Base-level data centers. Review of commercial literature on data center design found emphasis in power efficiency, thermal modeling and cooling, and network speed and availability. The topic of sizing data center processing capacity seems undeveloped. This thesis provides a better, pedigreed solution to the data center sizing problem. By analogy, Erlang's formulae for the probability of blocking and queuing should be applicable to cumulative CPU utilization in a data center. Using survey data collected by 38th Engineering Squadron, a simulation is built and correlation between the observed survey measurements and simulation measurements, and the Erlang, Gamma, and Gaussian-Normal distributions is found.
For a sample dataset of 70 servers over 14 hours of observation and a supposed .99999 requirement for traffic to be passed or otherwise unimpeded, Erlang distribution predicts 10 CPU cores are required, Gamma distribution predicts 10 CPU cores are required, Gaussian-Normal distribution predicts 9 CPU cores are required, Erlang B formulae predicts 14 CPU cores are required, and Erlang C formulae predicts 15 CPU cores are required.

15. SUBJECT TERMS

computer centers; data center; data center consolidation; resource management; resource utilization; servers; stochastic; virtualization; virtual machines

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. John Colombi, AFIT/ENV
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	82	937-255-3636 x3347, john.colombi@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18