2015

Mobile Cloud Computing: Offloading Mobile Processing to the Cloud

Jesus Zambrano
University of North Florida, kingping.soft@gmail.com

MOBILE CLOUD COMPUTING:
OFFLOADING MOBILE PROCESSING TO THE CLOUD.

by

Jesus Zambrano

A thesis submitted to the
School of Computing
in partial fulfillment of the requirements for the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
SCHOOL OF COMPUTING

April 2015

The thesis "Mobile Cloud Computing: Offloading Mobile Processing to the Cloud" submitted by Jesus Zambrano in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Sciences has been

Approved by the thesis committee: Date

______________________________ ________________

Dr. Sanjay P. Ahuja
Thesis Advisor and Committee Chairperson

______________________________ ________________

Dr. Roger Eggen

______________________________ ________________

Dr. Robert F. Roggio

Accepted for the School of Computing:

______________________________ ________________

Dr. Asai Asaithambi
Director of the School

Accepted for the College of Computing, Engineering, and Construction:

______________________________ ________________

Dr. Mark A. Tumeo
Dean of the College

Accepted for the University:

______________________________ ________________

Dr. John Kantner
Dean of the Graduate School

ACKNOWLEDGEMENT

I want to thank my grandparents; Tristan and Carmen Acevedo for always being my greatest motivation and my mother Carmen, my example to follow and for having left me the biggest teachings in my life, this achievement is yours.

I would also like to take this opportunity to thank my thesis advisor Dr. Sanjay Ahuja for his valuable guidance, support, and time he dedicated to me through all this process; thank you Dr. Ahuja. I also thank Dr. Roger Eggen and Dr. Bob Roggio who agreed to be on my thesis committee and provided great feedback in this research process. I also want to thank Jim Littleton for his valuable help and guidance in writing this thesis. Working with these four distinguished UNF faculty members has been an honor and a privilege. I also thank School of Computing and the ISA for all the support.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The current proliferation of mobile systems, such as smart phones, PDA and tablets, has led to their adoption as the primary computing platforms for many users. This trend suggests that designers will continue to aim towards the convergence of functionality on a single mobile device. However, this convergence penalizes the mobile system in computational resources such as processor speed, memory consumption, disk capacity, as well as in weight, size, ergonomics and the user's most important component, battery life. Therefore, this current trend aims towards the efficient and effective use of its hardware and software components. Hence, energy consumption and response time are major concerns when executing complex algorithms on mobile devices because they require significant resources to solve intricate problems.

Current cloud computing environments for performing complex and data intensive computation remotely are likely to be an excellent solution for off-loading computation and data processing from mobile devices restricted by reduced resources. In cloud computing, virtualization enables a logical abstraction of physical components in a scalable manner that can overcome the physical constraint of resources. This optimizes IT infrastructure and makes cloud computing a worthy cost effective solution.

The intent of this thesis is to determine the types of applications that are better suited to be off-loaded to the cloud from mobile devices. To this end, this thesis quantitatively and

qualitatively compares the performance of executing two different kinds of workloads locally on two different mobile devices and remotely on two different cloud computing providers. The results of this thesis are expected to provide valuable insight to developers and architects of mobile applications by providing information on the applications that can be performed remotely in order to save energy and get better response times while remaining transparent to users.

Chapter 1

INTRODUCTION

Mobile devices such as smart phones and tablets are increasingly the means by which users are accessing services on the Internet. In fact, by 2015, more users will access the Internet through mobile computational devices than by desktops or other wired devices [Hamblen11]. This will lead to a convergence of functionality on a single mobile device (such as phone + mp3 player + camera + Web browser + GPS + mobile apps + sensors) [Ranganathan10]. However, this conjunction penalizes the mobile system both with respect to computational resources such as processor speed, memory consumption, disk capacity, and in weight, size, ergonomics and the component most important to users, battery life [Satyanarayanan09].

Energy and response times are two key design considerations across a spectrum of computing solutions, from supercomputers and data centers to handheld phones and other mobile computers [Ranganathan10]. Low power consumption and energy efficiency have been critical concerns for developing electronic devices such as personal computers and mobile devices, and this importance seems to be increasing [Miettinen10].

Currently, most companies have been using the Internet to provide information and services online. Users are accustomed to doing everything on the Internet and taking advantage of its simplicity and efficiency. Therefore, most of the mobile device providers

have been designing and creating mobile systems capable of supporting the demands of the newest Internet based applications. Moreover, corporations such as banks, social networks companies, video and music providers, instant messaging networks and most of the software development companies have seen a huge advantage in creating mobile apps to avoid the use of mobile web browsers and go directly to the services the users desire.

Since the adoption of flat rate tariffs on cellular networks, cost no longer seems to be an obstacle [Kelényi10]. In addition, the newest wireless technologies such as 4G and Wi-Fi supporting faster internet connections, wireless Internet-connected devices, including mobile devices, are becoming more common each day. However, with this increasing development in mobile technology, more applications now require intensive use of the processing and communicating capabilities of mobile devices. Therefore, mobile apps are directly affected by two main factors: battery life limited by energy consumption and processing time limited by poor mobile resources.

Recent trends aim towards the integration of huge variety of applications and different components within a mobile device, which means more processing as well as less battery life. In addition, the limited hardware capability in current mobile devices is an obstacle to supporting the increasing high-processing demands of the latest applications and of future developments.

Mobile devices have spread out and have become ubiquitous technologies that people use on a daily basis. No matter which brand, they have become necessary accessories for all kinds of users.

While the most energy-efficient setup for many current mobile applications is local computing, there clearly are workloads that can benefit from moving to remote infrastructures [Miettinen10] due to these workloads demanding higher computing resources. An obvious solution to the resource constraint on mobile devices is to offload computation. Therefore, offloading mobile processing could mean an effective solution to overcome the limited resources on mobile phones. However, when using this approach, the critical aspect for mobile clients is the trade-off between energy consumed by computation and the energy consumed by communication [Miettinen10]. In addition, there are many concerns when offloading data to remote infrastructures such as privacy, security, and reliability on the wireless network and the remote infrastructure service.

Currently, there is a new set of services being widely implemented in the computer field that are able to perform these high-processing tasks remotely with a lower cost of energy consumption on the mobile system and a shorter response time; namely, Cloud Computing Services.

1.1 Android OS

Android, one of the most popular mobile operative systems powers hundreds of millions of mobile devices around the world. As many other successful technologies, it was developed by Google in partnership with Open Handset Alliance. The mobile OS utilizes a Linux based kernel; it implements its permission based security model and most of its code is under Apache License. Unlike on other mobile operating systems like Apple's iOS, Palm's WebOS or Symbian, Android applications are written in Java and run in virtual machines [Brahler10]. Therefore, it has a broad Java library support (java.io, java.security, java.net, java.sql). In addition, Android, instead of using a standard Java virtual machine, uses its own VM, the Dalvik VM, which is not compatible to the standard Java virtual machine.

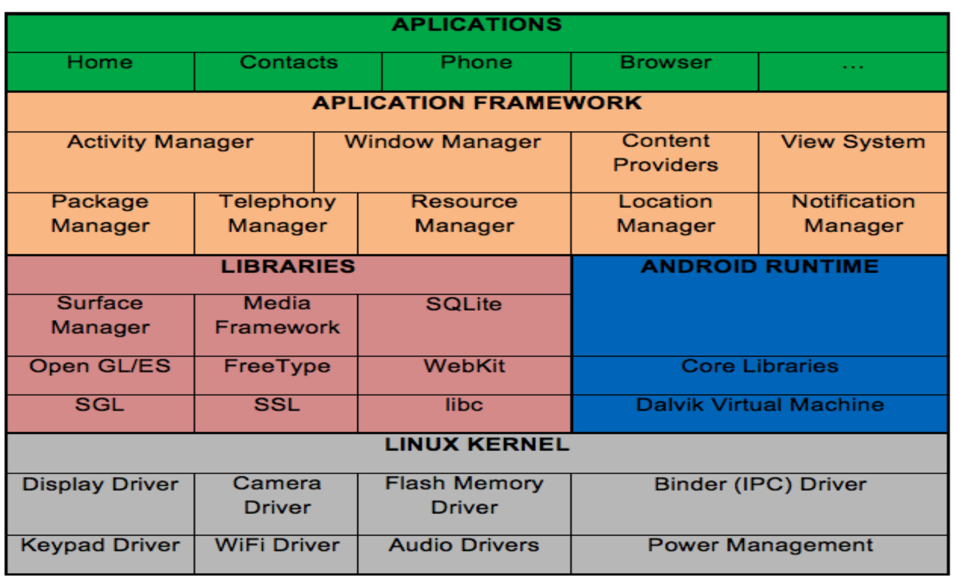

Figure 1: Android Architecture

1.1.1 The Dalvik VM

The Android VM instead of using standard byte code it owns a different byte code format adjusted to the needs of Android target devices. The byte code is more compact than usual Java byte code and its file extension is .dex (Dalvik Executable). The Dalvik runs classes compiled by a Java language compiler transformed into the .dex format by the included "dx" tool. It relies on the Linux kernel for underlying functionality such as threading and low-level memory management [Ehringer10].

To address memory constraints and allow for fast startup times, Dalvik shares core, read-only libraries between VM processes using a concept called Zygote. The sharing is done securely by giving the VM processes permission to read the code but not edit it.

1.1.2 The Zygote

The Zygote is an approach that enables both sharing of code across VM instances and providing fast startup time of new VM instances. The Zygote design assumes that there are a significant number of core library classes and corresponding heap structures used across many applications. Generally, these core library classes are read-only and are therefore a good candidate for preloading and sharing across processes. In other words, there are data and classes that most applications use but never modify. These characteristics are exploited to optimize sharing of this memory across processes [Ehringer10].

Once the Zygote has initialized, it will sit and wait for socket requests coming from the runtime process indicating that it should fork new VM instances based on the Zygote VM instance. By spawning new VM processes from the Zygote, the startup time is minimized.

In traditional Java VM design, each instance of the VM will have an entire copy of the core library class files and any associated heap objects. Memory is not shared across instances.

Android's security architecture ensures no application, by default, has permission to perform any operations that would adversely affect other applications, the operating system, or the user. This includes reading or writing the user's private data (such as contacts or e-mails), reading or writing another application's files, performing network access, keeping the device awake, etc.

1.1.3 Android Garbage Collector

Due to limited resources on Android, garbage collection may be invoked frequently and may take more time than on other systems [Husted11]. The garbage collector (GC) on Android is run by the Dalvik and it is executed in each VM separately, therefore each VMs heap is garbage collected independently [Brahler10].

The GC plays a role in removing objects from memory, when the objects generated by an application in execution are no longer in use. Manual memory management on Android is no needed, the GC is automatically called by the Dalvik VM and frees unused objects; it automatically reclaims unreachable objects, or reachable but not used objects [Espinar11].

In addition, it does not mean Android software engineers can ignore memory management; they should be especially mindful of memory usage on mobile devices, where memory is more constrained.

Android architecture manages memory in a per-app cap size basis, which means, every app running has its own memory heap size. This automatic memory assignation is device-dependent. Android devices have a per-app cap range from 16MB to around 48MB and future devices will likely have even more available [Dubroy11]. Depending on how much big the per-app heap size is the GC will take longer or shorter time freeing memory. The GC is called depending on how full the app heap is and if the app is close to its memory limit. However, the GC will automatically try freeing memory even though it might not be able to keep the app up causing that the app eventually crashes.

1.2 Cloud Architectures

Many companies are adopting cloud computing services to store their data and to use cloud-based infrastructure to execute applications and perform heavy workloads.

Generally, those applications require vast amounts of resources to perform complex algorithms with the most efficient response time.

Cloud computer services offer hardware infrastructure and software models necessary to support application transactions and to provide storage, performance, security and maintenance in a pay-per-use basis while reducing IT infrastructure and personnel costs to business. Cloud computing utilizes advanced high-performance server systems with large amounts of memory, storage and multiple processors all working on a collaborative model.

The concept of deployment models leads to the classification of Public clouds, Private clouds and Hybrid clouds. Public clouds offer storage and other resources on a pay-per-use basis. Private clouds offer the infrastructure needed either through the internal organization or by third party vendors; however, private cloud pose certain risks relating to scalability, maintenance, and investments. Hybrid clouds, a combination of Public and Private clouds serve the benefits of multiple deployment models and degrees of fault tolerance [Bhagavathi13].

Cloud computing model services include Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). IaaS provides virtual platforms capable of running different virtual machines, storage, firewall, IP addresses, load balancers etc. [Bhagavathi13]. In PaaS, provides a complete computer platform including a choice of operating system, programming environment, database and servers, and the platform's

resources can scale automatically to handle the demands of the application. SaaS, provides an on-demand programming environment and database, which allows users to run their applications without the overhead of providing IT support and maintenance.

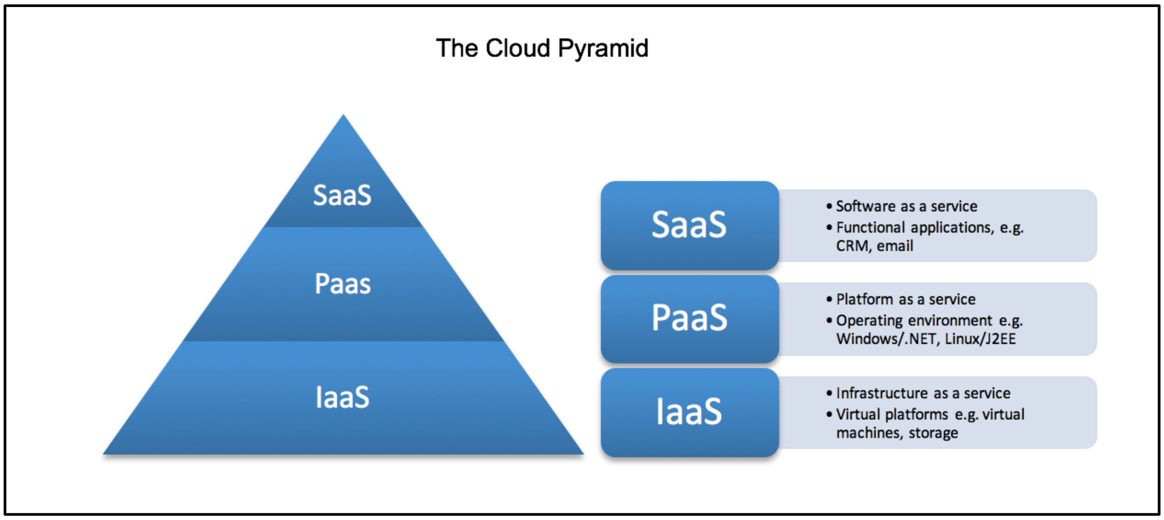

Figure 2: Categories Architecture of Cloud Computing Services

In cloud computing, the core technology that has made it possible is hardware virtualization. "Virtualization is a technology that combines or divides computing resources to present one or many operating environments using methodologies like hardware and software partitioning or aggregation, partial or complete machine simulation, emulation, time-sharing, and many others" [Nanda05].

A virtualization layer provides an infrastructural support using the lower-level resources to create multiple virtual machines that are independent and isolated from each other within a single physical server. Each virtual server could have its own operating system

installed in it. Many virtual servers can operate simultaneously and independently of each other.

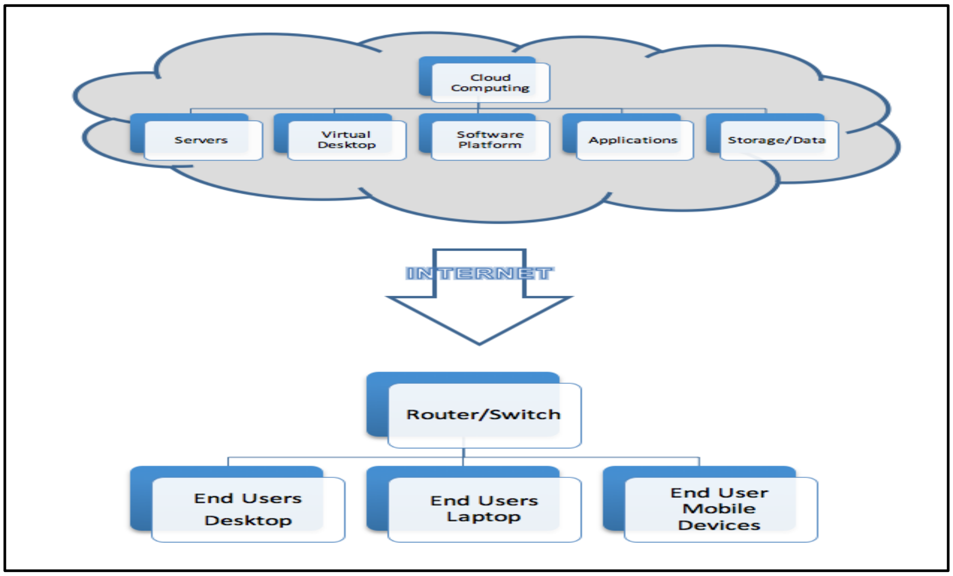

Figure 3: Generic Cloud Computing Architecture

1.2.1 Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 is an IaaS cloud service that provides resizable computing capacity in Amazon Web Services Cloud (AWS). In EC2 it is possible to launch as many virtual servers as you need, configure security and networking, and manage storage. EC2 enables to scale up or down to handle changes in requirements or spikes in popularity, reducing needs to forecast traffic.

Amazon EC2 provides a web-based interface that allows users to configure their environment, virtual machines, operative systems, memory, storage and even the data

center location so virtual machines can be deployed closer to the service target. Amazon EC2 also supports several instance types including micro, high CPU, high memory, cluster GPU, cluster compute, high memory cluster and high I/O instances [Bhagavathi13].

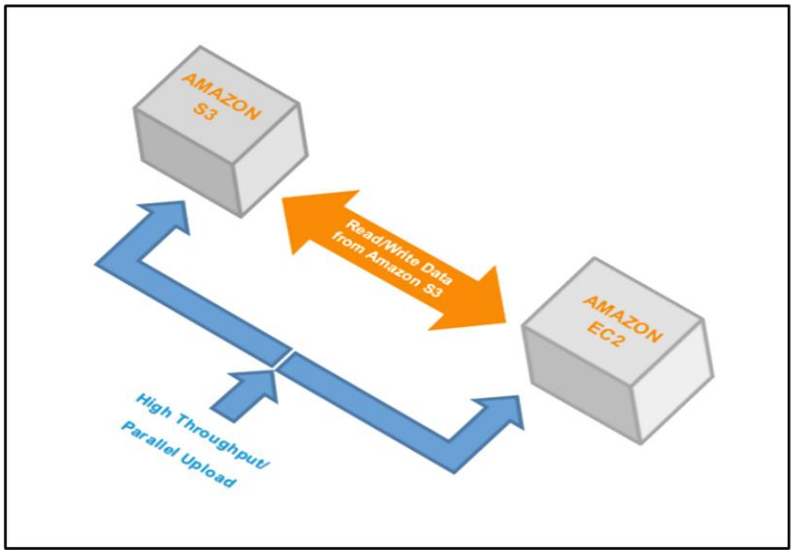

Figure 4: Amazon Web Services Architecture

1.2.2 Microsoft Windows Azure

Azure is an Internet-scale cloud computing and services platform hosted in datacenters created by Microsoft Corp. An open and flexible cloud platform enables to quickly build, deploy, and manage applications across a global network of Microsoft-managed datacenters. It delivers a 99.95% monthly SLA and enables to build and run highly available applications without focusing on the infrastructure [Microsoft13].

Windows Azure provides automatic OS and service patching, built in network load balancing and resiliency to hardware failure. It supports a deployment model that enables you to upgrade your application without downtime. Elastically grow or shrink your resource usage based on your needs. Windows Azure is available in multiple datacenters around the world, enabling bossiness to deploy their applications close to your customers [Microsoft13].

It features a comprehensive set of storage, computing, and networking infrastructure services that reside in Microsoft's network of datacenters [Sysfore11].

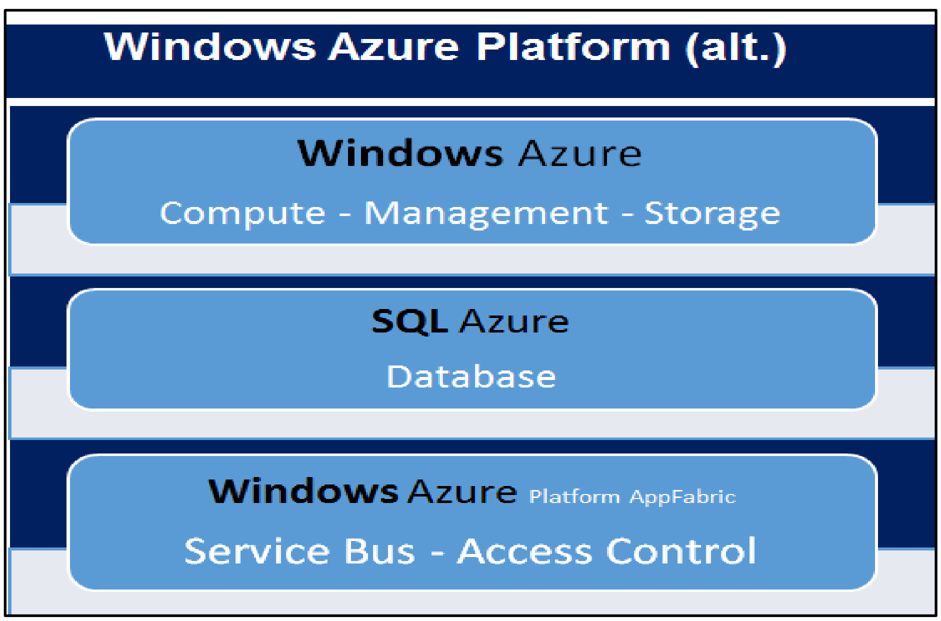

Figure 5: Microsoft Windows Azure Architecture

As well as Amazon EC2 it has a web-based console that allows user to create and manage cloud services as virtual machines, as well as to configure networking, create security policies, administer storage, replication and so on.

Chapter 2

LITERATURE REVIEW

The intention of this chapter is to focus on documenting the contribution of other researchers and to expand the understanding of concepts, models, and patterns of computation on mobile devices. The most important references surveyed are listed below.

2.1 Studies on Energy Efficiency of Mobile clients in Cloud Computing

Miettinen and Nurminen (2010), both workers at the Nokia Research Center, in their research paper, "Energy efficiency of mobile clients in cloud computing", discuss the potential saving of mobile client energy when offloading certain mobile workload to the cloud. They also discuss the trade-off between energy consumed by computation and the energy consumed by the additional communication. They show how the trade-offs are highly sensitive to the exact characteristics of the workload, data communication patterns and technologies used, and discuss the implications for the design and engineering of energy efficient mobile cloud computing solutions.

2.2 Studies on Benefits of Offloading Mobile Computation

Kumar and Yung-Hsiang (2010) in "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" state that mobile cloud computing can enhance the computing capability of mobile systems and it is a possible solution for extending battery lifetimes of mobile systems. They also discuss some factors in which mobile cloud computing may not be beneficial:

- Privacy and security: Encrypting and decrypting data require more processing as well as battery life.
- Reliability: Depended on the wireless network and the cloud service.

2.3 Studies on Virtual Cloud Computing provider for Mobile Devices

Huerta-Canepa and Dongman (2010) in their "A virtual cloud computing provider for mobile devices" paper, present the guidelines for a framework to create virtual mobile cloud computing providers. The main goal of this work is to show the feasibility of a new scheme of sharing mobile resources to perform common workloads among different mobile users. The power of this approach increases when there is a high availability of nearby devices with common goals/activities**.**

The three documents previously mentioned describe the feasibility of offloading mobile tasks to powerful remote infrastructures where higher energy savings and lower response times can be achieved. Miettinen and Nurminen provide an understanding about the

trade-off between energy consumed by computation and the energy consumed by the additional communication. This forms a basis to understand the main factor when offloading workloads. Kumar and Yung-Hsiang analyzed some factors in which mobile cloud computing may not be beneficial such as when applications deals with sensitive data and they should encrypt and decrypt it before and after transmissions. They also analyzed the scenario of dependence on the wireless network and the cloud service. The framework described by Huerta-Canepa and Dongman in their paper "A virtual cloud computing provider for mobile devices" serves as a starting point for designing a methodology to measure energy consumption and response time parameters by using two different commercial mobile devices and two different cloud providers.

Chapter 3

RESEARCH METHODOLOGY

This research evaluates the performance on a mobile device; a Samsung S3 phone, and Amazon EC2 together with Microsoft Windows Azure cloud services as platforms for data-intensive computation. The study performs and analyzes three experiments in order to obtain response times and energy consumption on the mobile device when performing two different kinds of workloads; light-communication / intensive-computation and intensive-communication / intensive-computation. These studies reflect what workload characteristics are more suitable to offload to the cloud. These two different workloads represent two generic tasks defined by the below table

Light-Communication. Intensive-Computation.	Calculates the next prime number of a given long number on both phone and clouds
Intensive-Communication. Intensive-Computation.	Finds a matched value for a given index in a text file on both the phone and clouds.

Table 1: Workloads

An android app enables both local processing as well as offloading processing to the cloud VMs. The mobile app as well as the server applications installed on the VMs are written in java using Eclipse IDE. Additionally, both client and server apps use java sockets to establish transmit and receive data from/to the mobile device.

Little Eye, a specialized software to obtain energy consumption, was used to take measurements on the mobile device. It shows graphical and quantitative values that reflect the energy consumption behavior in mAh. In addition, to obtain response times, the mobile app records the time while executing the workload locally and off-loading it to the cloud.

In the first study, response times and energy consumption measurements are compare locally to the response times and energy consumption remotely when offloading workloads to the cloud. The second study, determines a breakpoint when the mobile device cannot handle the data transmissions due to device's heap size saturation. This test executes the varying of a file size until the app crashes. In the third study, Amazon EC2 and Microsoft Windows Azure VMs are compared using the previous taken response times and energy consumption measurements.

The main goal is to identify what kinds of workloads are more suitable to off-load to the cloud instead of performed locally. In order to get a more accurate measurement of these indicators, the tests conduct a five-time regime under the same conditions to compute a more precise average. Additionally, these measurements are organized into tables and graphs to be compared and analyzed.

This equation involves the quantitative factors that can be measured and are decisive when processing locally on the mobile device or while transmitting to and receiving from the cloud.

$$ECloud < ELocal.$$ [Miettinen10] and

$$RTcloud < RTlocal$$

ECloud: Energy consumed when off-loading mobile processing to the cloud and receiving the result back into the mobile device.

ELocal: Energy consumed when processing workload on the mobile device.

RTcloud: Response time when off-loading mobile processing to the cloud and receiving the result back into the mobile device.

RTlocal: Response time when processing workload on the mobile device

This condition has several factors affecting both sides of the equation [Kumar10]:

D: Amount of data transferred in bytes between the mobile device and the cloud

C: Computation requirements

B: Network Bandwidth

Suppose the energy consumption for receiving data on the mobile device is the same as when transmitting data; even though in reality data transmission takes more battery life than when the mobile device is receiving data.

Having all the factors that affect the above conditions, there is energy saving if ELocal is greater than ECloud. The graph below shows how the variables workload, computation and bandwidth affect the decision whether or not offload data on the cloud [Kumar10].

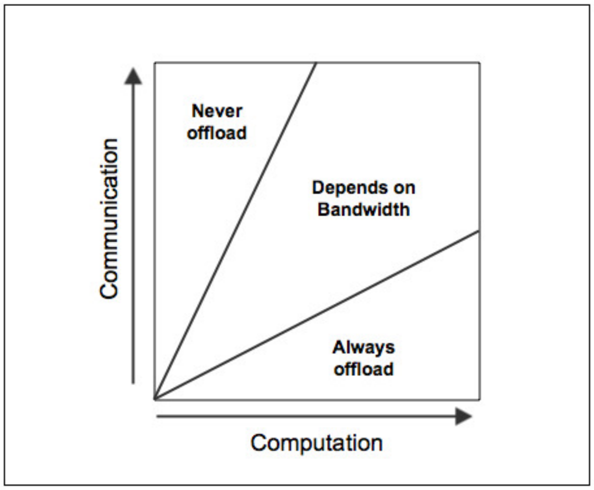

Figure 6: Decision graph with variables Workload, Computation, and Bandwidth

Chapter 4

TESTBED SETUP

4.1 Setting up the Android Development Environment

The Android development environment is composed of three different software components used.

- Eclipse Standard 4.3.1 Edition
- Android SDK (Software Development Kit)
- ADT Plugin for Eclipse (Android Development Tool)

The developing framework uses Eclipse, the Android SDK, and the ADT plug-in. The Android SDK is a package that provides API libraries and development tools necessary to build, test and debug apps for Android [Android13]. The ADT plug-in for Eclipse allows setting up Android projects, creating an application UI, adding packages based on the Android Framework API, and providing an emulator to test the Android apps locally in the development machine.

4.2 Creating Virtual Machines on the Amazon EC2 Cloud Service

By using the Amazon Web Services web-based console it is possible to configure and create three different virtual machine sizes; small, medium and large on the EC2 platform. Additionally, the JRE 7 kit and Eclipse Standard 4.3.1 run on each virtual machine in order to perform the Java server application in charge of running the algorithms for executing the mobile workloads.

4.3 Creating Virtual Machines on the Microsoft Windows Azure Cloud Service

Windows Azure as well as Amazon EC2 offers a web-based console to manage VMs on its cloud service. Azure creates a cloud service that contains the VMs created. In addition, Windows Azure requires that for every communication port an *end point* be created for transmitting and receiving data. This mechanism allows the load balancer to grant communications through it.

Chapter 5

HARDWARE AND SOFTWARE SPECIFICATIONS

5.1 Software Specifications

- Eclipse 4.3.1 as development framework with Java Runtime Environment JRE 7).
- The Android Software Development Kit (SDK).
- Android Development Tools (ADT).
- Use Remote Desktop Connection application for accessing, managing and installing software on the Amazon EC2 and Windows Azure different VMs.

5.2 Hardware Specifications

- Samsung Galaxy S3 as a mobile client described in the table below:

	Samsung Galaxy S3
Processor	Samsung ARM Cortex-A9. Quad Core 1.4Ghz
Operative System	Android 4.1.2
Memory	1 GB
Storage	12 GB
Battery	2100 mAh

Table 2: Mobile client specifications

- Use of three different VMs configuration on each cloud provider described in the table below:

	Amazon EC2	Microsoft Azure
Number of cores	1 Core	1 Core
Processor	Intel Xeon Family	Intel Xeon Family
Compute Unit	1 C.U	1 C.U
Operative System	Microsoft Windows Server 2012 Base 64bits	Microsoft Windows Server 2012 Datacenter
Memory	1.6 GB	1.75 GB
Internal Storage	160 GB	70 GB

Table 3: Comparison between Amazon EC2 and Microsoft Azure Small VMs

	Amazon EC2	Microsoft Azure
Number of cores	1 Core	2 Cores
Processor	Intel Xeon Family	Intel Xeon Family
Compute Unit	2 C.U	2 C.U
Operative System	Microsoft Windows Server 2012 Base 64bits	Microsoft Windows Server 2012 Datacenter 64 bits
Memory	3.7 GB	3.5 GB
Internal Storage	160 GB	135 GB

Table 4: Comparison between Amazon EC2 and Microsoft Azure Medium VMs

	Amazon EC2	Microsoft Azure
Number of cores	4 Cores	4 Cores
Processor	Intel Xeon Family	Intel Xeon Family
Compute Unit	5 C.U	5 C.U
Operative System	Microsoft Windows Server 2012 Base 64bits	Microsoft Windows Server 2012 Datacenter
Memory	15 GB	7 GB
Internal Storage	160 GB	285 GB

Table 5: Comparison between Amazon EC2 and Microsoft Azure Large VMs
Data Link Communication Specifications

	Wi-Fi	4G HSPA
Service Provider	Bright House Internet	T-Mobile
Download	30 Mb/s	10 Mb/s
Upload	2MB/s	1 Mb/s

Table 6: Comparison between Wi-Fi and 4G Internet service

Chapter 6

RESULTS AND DISCUSSION

This study evaluates and compares the performance and energy consumption of processing two different workloads; light-communication / intensive-computation and intensive-communication /intensive-computation both locally and remotely to two different cloud providers. Throughout this research, p-values are obtained through statistical analysis of the collected data using T-TEST function available in Microsoft Excel 2010. Statistical analysis resulting in a p-value of less than 0.05 is significant.

The response times (in milliseconds) and energy consumption (in mAh), for Amazon EC2 and Microsoft Windows Azure are presented in graphs to assist with analyzing trends. The graphs compare the Amazon EC2 and Microsoft Windows Azure cloud services executing light-communication / intensive-computation and intensive-communication / intensive-computation on three different virtual machines using Wi-Fi and 4G. For each graph, the y-axis represents the response times and energy consumption values achieved during the test and the x-axis represents the workload characteristic.

6.1 Quantitative Comparison

6.1.1 Local vs. Remote Processing Comparison

Tables 7 and 8, and Figures 7 and 8 present the response times for Amazon EC2 and Windows Azure for light-communication / intensive communication workload

Number	Local	Remote					
	Phone	Amazon EC2					
		SVM		MVM		LVM	
		Wi-Fi	4G	Wi-Fi	4G	Wi-Fi	4G
0	4105	4269	4256	4136	5022	4013	5008
9	4118	4298	4482	4171	4879	4137	5543
9223	5150	5491	5567	5342	5545	5042	6364
9223372	6186	6074	5342	6115	5986	6233	6756
9223372036	6619	9449	5742	6622	6296	6766	7556
9223372036854	10579	11369	7157	7190	7998	7191	8314
9223372036854775	77706	14799	24467	11465	13875	10994	15667
9223372036854775643	1531440	187700	199654	59181	63325	54329	57006
P-value		0.164502	0.164564	0.163618	0.164341	0.163546	0.165324

Table 7: Local vs. EC2 - Response times for light-communication / intensive-computation

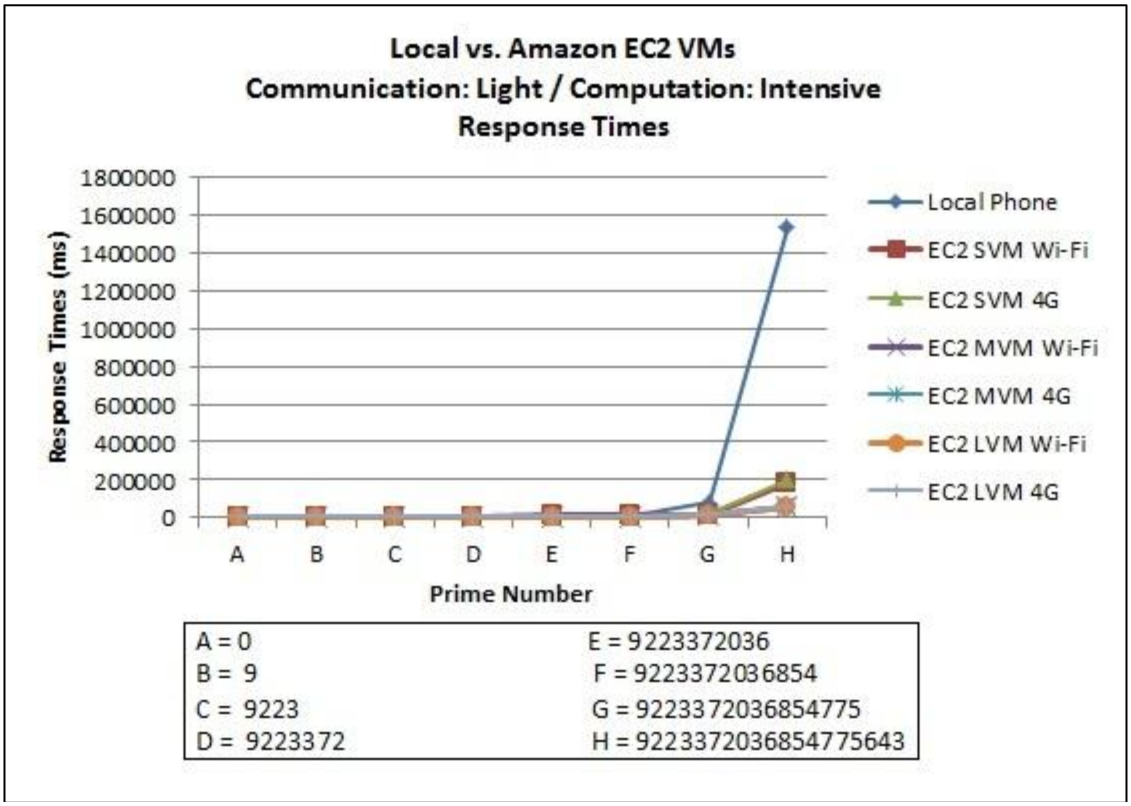

Figure 7: Response Local vs. Amazon EC2 - Response times for light-communication / intensive-computation.

Number	Local	Remote					
	Phone	Windows Azure					
		SVM		MVM		LVM	
		Wi-Fi	4G	Wi-Fi	4G	Wi-Fi	4G
0	4105	4183	4823	4054	5074	4009	5011
9	4118	4191	5282	4047	5179	4035	5087
9223	5150	5292	6942	5235	6335	5089	6395
9223372	6186	5920	5264	6184	6510	5824	6592
9223372036	6619	8859	7346	6075	6535	5903	6482
9223372036854	10579	10094	7634	6298	7284	6523	6935
9223372036854775	77706	15963	246120	11344	15955	14568	12535
9223372036854775643	1531440	184700	201684	61945	65110	56784	59495
P-values		0.164283	0.21147	0.163293	0.164871	0.163766	0.164284

Table 8: Local vs. Windows Azure - Response times for light-communication / intensive-computation.

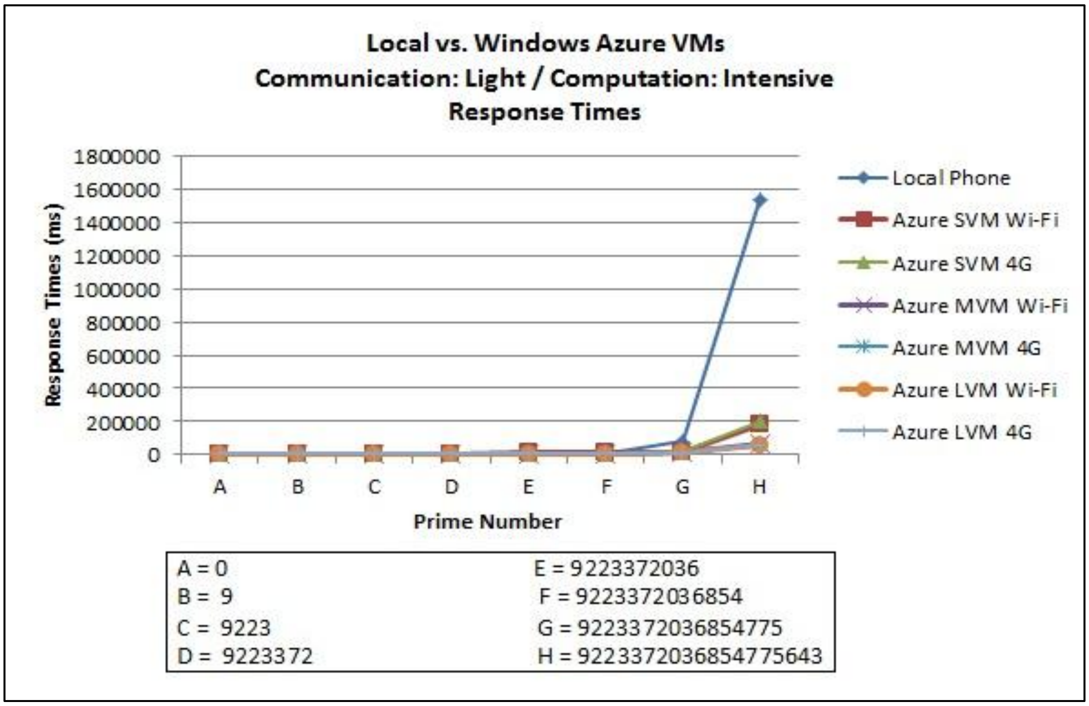

Figure 8: Local vs. Windows Azure - Response times for light-communication / intensive-computation.

Tables 9 and 10, and Figures 9 and 10 present the energy consumption for Amazon EC2 and Windows Azure for light-communication / intensive communication workload

Number	Local	Remote					
	Phone	Amazon EC2					
		SVM		MVM		LVM	
		Wi-Fi	4G	Wi-Fi	4G	Wi-Fi	4G
0	0.36	0.33	0.47	0.39	0.49	0.38	0.45
9	0.37	0.34	0.48	0.42	0.50	0.4	0.48
9223	0.38	0.50	0.51	0.42	0.51	0.42	0.49
9223372	0.47	0.59	0.59	0.50	0.60	0.52	0.55
9223372036	0.72	0,62	0.61	0.52	0.58	0.58	0.62
9223372036854	0.77	0.66	0.76	0.55	0.61	0.57	0.62
9223372036854775	6.07	0.97	1.15	0.90	1.24	0.7	1.10
9223372036854775643	112.22	10.12	11.20	3.17	4.17	3.02	3.48
P-Value		0.165676	0.16414	0.162968	0.164735	0.162711	0.164308

Table 9: Local vs. Amazon EC2 - Energy consumption for light-communication / intensive-computation

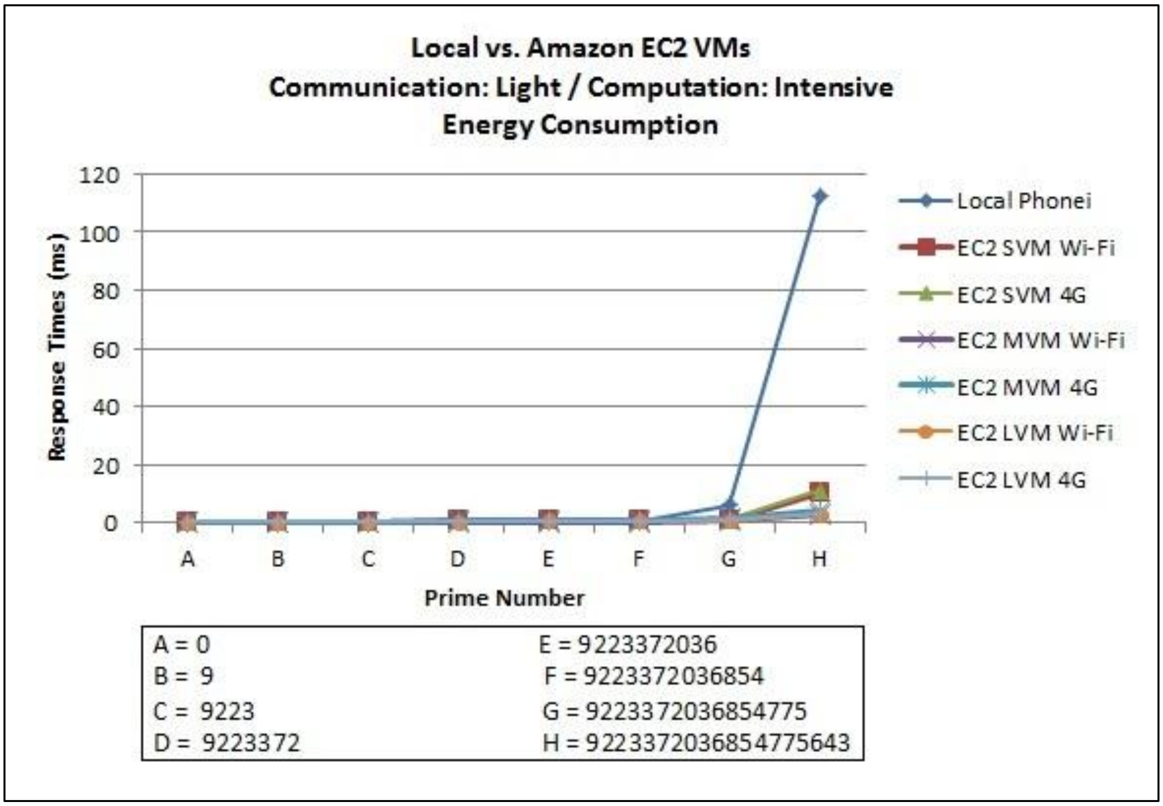

Figure 9: Local vs. Amazon EC2 - Energy consumption for light-communication / intensive-computation

Number	Local	Remote					
	Phone	Windows Azure					
		SVM		MVM		LVM	
		Wi-Fi	4G	Wi-Fi	4G	Wi-Fi	4G
0	0.36	0.32	0.5	0.39	0.42	0.41	0.42
9	0.37	0.33	0.49	0.42	0.43	0.42	0.42
9223	0.38	0.54	0.52	0.42	0.49	0.42	0.44
9223372	0.47	0.54	0.56	0.50	0.52	0.48	0.50
9223372036	0.72	0.59	0.58	0.52	0.54	0.58	0.56
9223372036854	0.77	0.64	0.82	0.55	0.68	0.67	0.68
9223372036854775	6.07	0.89	1.21	0.90	1.19	1.28	1.26
9223372036854775643	112.22	9.21	13.34	3.17	3.89	3.14	3.66
P-value		0.162639	0.162639	0.16418	0.164166	0.164207	0.164204

Table 10: Local vs. Windows Azure - Energy consumption for light-communication / intensive computation

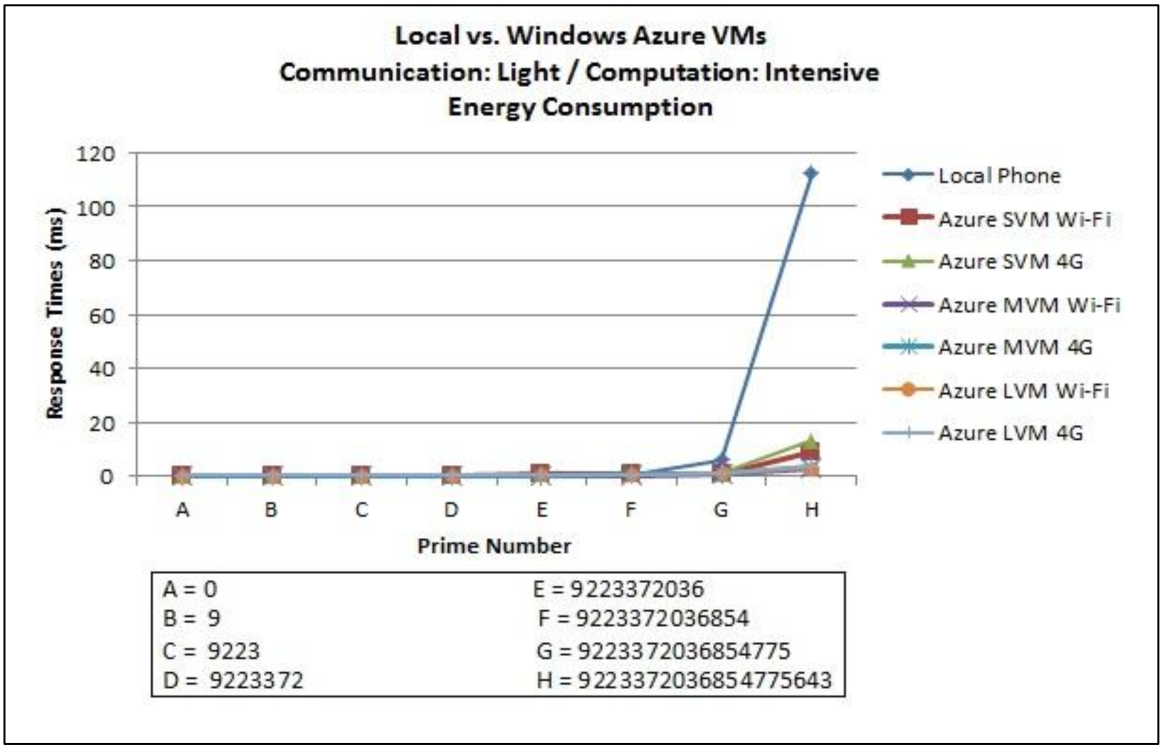

Figure 10: Local vs. Windows Azure - Energy consumption for light-communication / intensive computation

Tables 11 and 12, and Figures 11 and 12 present the response times for Amazon EC2 and Windows Azure for intensive-communication / intensive communication workload

Number	Local	Remote					
	Phone	Amazon EC2					
		SVM		MVM		LVM	
		Wi-Fi	4G	Wi-Fi	4G	Wi-Fi	4G
0	51263	39196	47529	38338	46091	35117	46221
1.000	53351	39745	48212	38481	47627	35430	47039
20.000	55140	42380	49165	39931	48813	37619	47484
40.000	55796	43185	49960	39013	47814	37909	50638
80.000	59839	43483	50081	41375	49963	37755	49570
150.000	61355	43842	49470	42881	50916	37934	49734
160.000	∞	44408	50790	43354	50604	39121	48923
P-value		0.00	0.00	0.00	0.00	0.00	0.00

Table 11: Local vs. Amazon EC2 - Response times for intensive-communication / intensive computation.

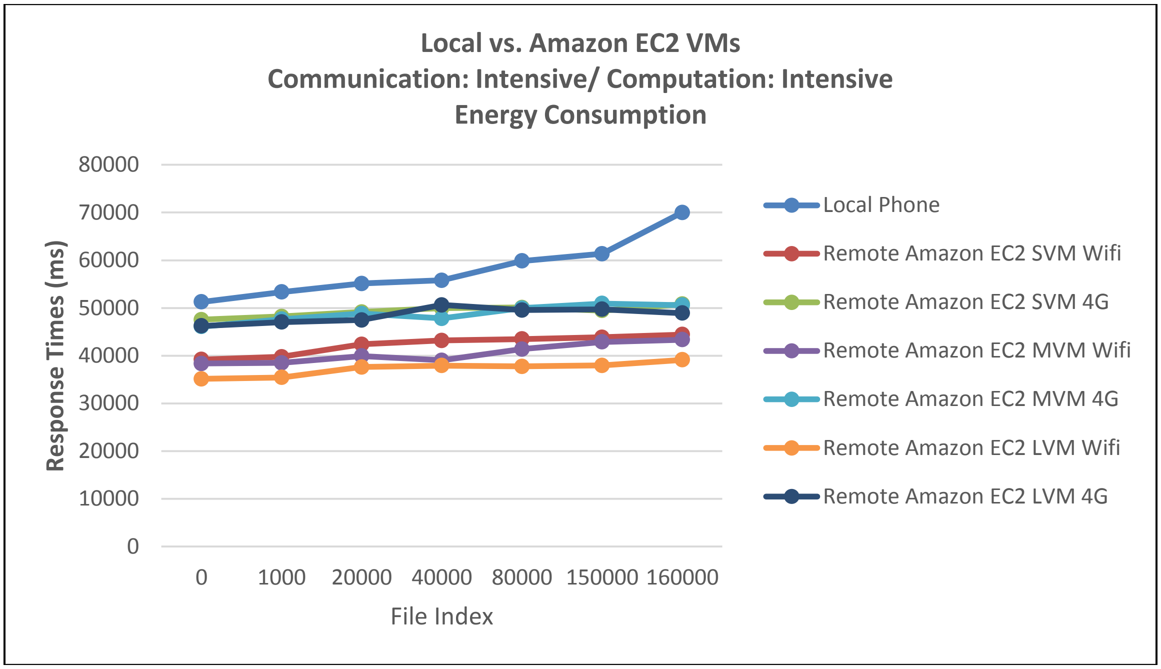

Figure 11: Local vs. Amazon EC2 - Response times for intensive-communication / intensive computation.

Number	Local	Remote					
	Phone	Windows Azure					
		SVM		MVM		LVM	
		Wi-Fi	4G	Wi-Fi	4G	Wi-Fi	4G
0	51263	40365	50529	36746	44606	39034	47467
1.000	53351	38045	49363	39938	46679	40382	47927
20.000	55140	40806	49345	40936	49374	40592	46038
40.000	55796	42185	47646	41683	48834	39887	49437
80.000	59839	42803	51563	41784	47944	40945	48205
150.000	61355	39477	52356	38366	51984	39430	47661
160.000	∞	43367	53058	40757	50475	38592	48024
P-value		0.00	0.00	0.00	0.00	0.00	0.00

Table 12: Local vs. Windows Azure - Response times for intensive-communication / intensive computation

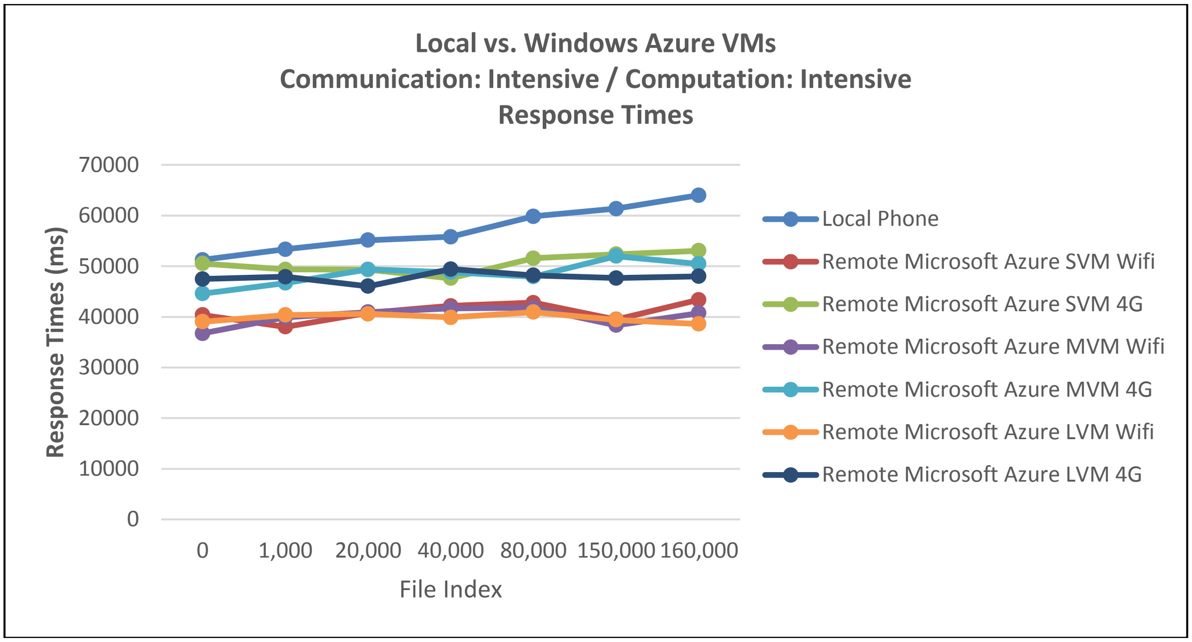

Figure 12: Local vs. Windows Azure - Response times for intensive-communication / intensive computation

Tables 13 and 14, and Figures 13 and 14 present the energy consumption for Amazon EC2 and Windows Azure for intensive-communication / intensive communication workload

Number	Local	Remote					
	Phone	Amazon EC2					
		SVM		MVM		LVM	
		Wi-Fi	4G	Wi-Fi	4G	Wi-Fi	4G
0	3.95	2.30	4.04	2.18	3.64	2.07	3.48
1.000	4.07	2.31	4.05	2.18	3.63	2.09	3.51
20.000	3.82	2.50	4.05	2.34	3.62	2.16	3.53
40.000	3.99	2.45	4.03	2.35	3.54	2.2	3.65
80.000	4.15	2.53	4.05	2.42	3.78	2.22	3.82
150.000	4.27	2.61	4.06	2.44	3.86	2.25	3.85
160.000	∞	2.64	4.05	2.44	3.84	2.28	3.89
P-value		0.00	0.46963957	0.00	0.00	0.00	0.00

Table 13: Local vs. Amazon EC2 - Energy consumption for intensive-communication / intensive-computation

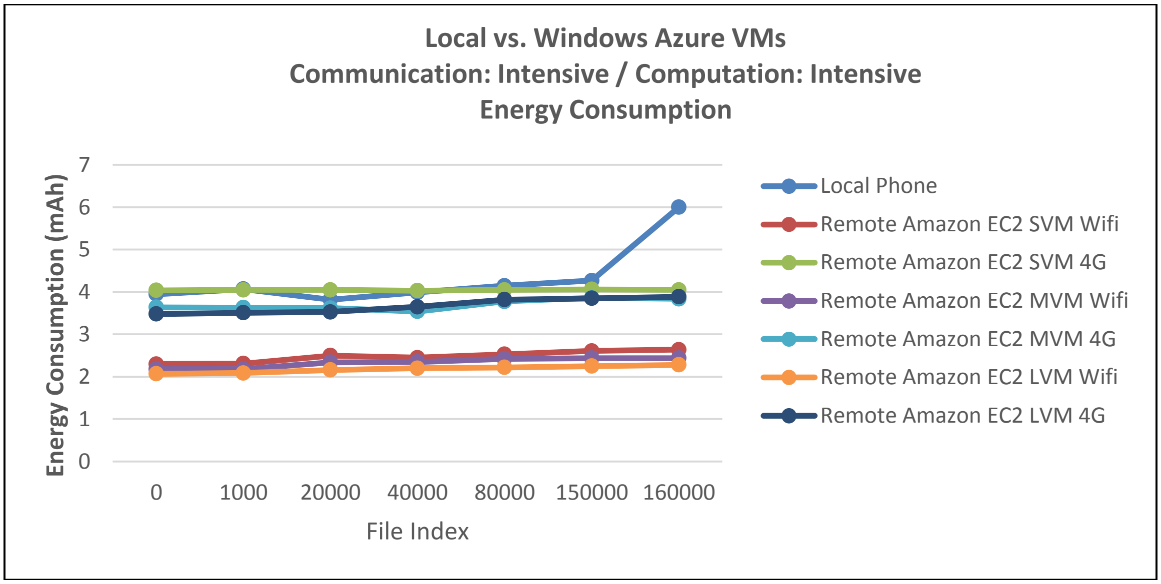

Figure 13: Local vs. Amazon EC2 - Energy consumption for intensive-communication / intensive-computation

Number	Local	Remote					
	Phone	Windows Azure					
		SVM		MVM		LVM	
		Wi-Fi	4G	Wi-Fi	4G	Wi-Fi	4G
0	3.95	2.37	4.1	2.22	3.66	2.2	3.52
1.000	4.07	2.22	4.08	2.23	3.62	2.22	3.54
20.000	3.82	2.39	4	2.19	3.6	2.24	3.48
40.000	3.99	2.4	4.06	2.28	3.56	2.18	3.58
80.000	4.15	2.24	4	2.32	3.72	2.22	3.72
150.000	4.27	2.3	4.1	2.38	3.84	2.24	3.9
160.000	∞	2.51	4.05	2.36	3.74	2.18	3.98
P-value		0.00	0.4070	0.00	0.00	0.00	0.00

Table 14: Local vs. Windows Azure - Energy consumption for intensive-communication / intensive communication

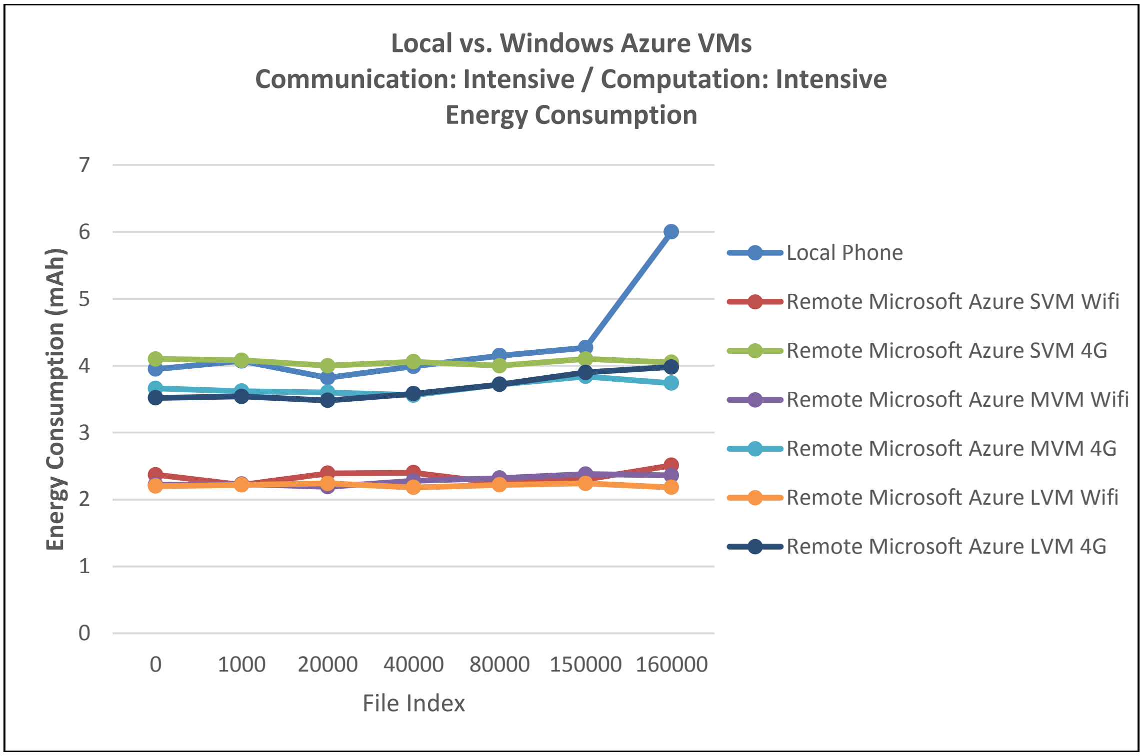

Figure 14: Local vs. Windows Azure - Energy consumption for intensive-communication / intensive communication

6.1.2 Mobile Memory Saturation Breakpoint

Tables 15,16,17,18 and 19 and Figures 15, 16, 17, and 18 present the response time for Amazon EC2 for intensive-communication / intensive communication workload

Index	Wi-Fi (ms)	4G (ms)
0	26,823	32,869
11,000	28,056	34,650
22,000	30,113	34,997
33,000	31,241	35,873
44,000	31,471	36,062

Table 15: Amazon EC2 response time on a large virtual machine when transferring 1MB data file

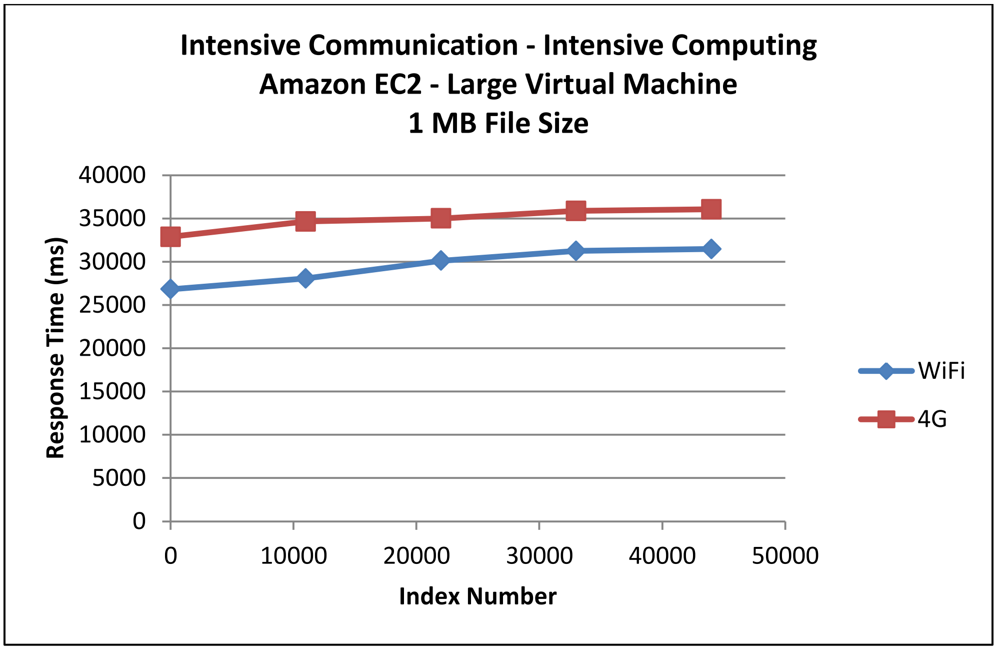

Figure 15: Amazon EC2 response time on large virtual machine when transferring 1MB data file.

Index	Wi-Fi (ms)	4G (ms)
0	62,127	84,783
106,250	63,360	85,962
212,500	66,544	101,436
318,750	66,941	110,928
425,000	68,252	112,940

Table 16: Amazon EC2 response time on a large virtual machine when transferring 10MB data file

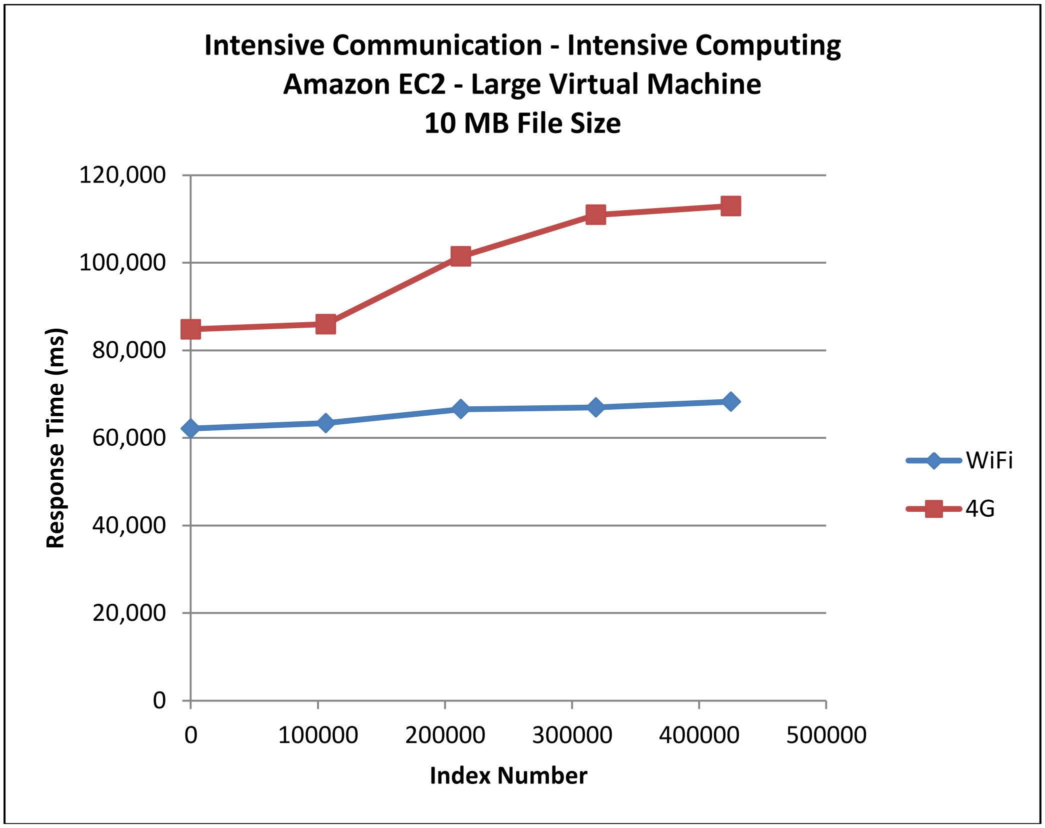

Figure 16: Amazon EC2 response time on large virtual machine when transferring 10MB data file.

Index	Wi-Fi (ms)	4G (ms)
0	117,812	181,736
263,750	120,120	194,514
527,500	123,949	200,063
791,250	125,477	207,060
1,055,000	127,019	209,877

Table 17: Amazon EC2 response time on a large virtual machine when transferring 25MB data file

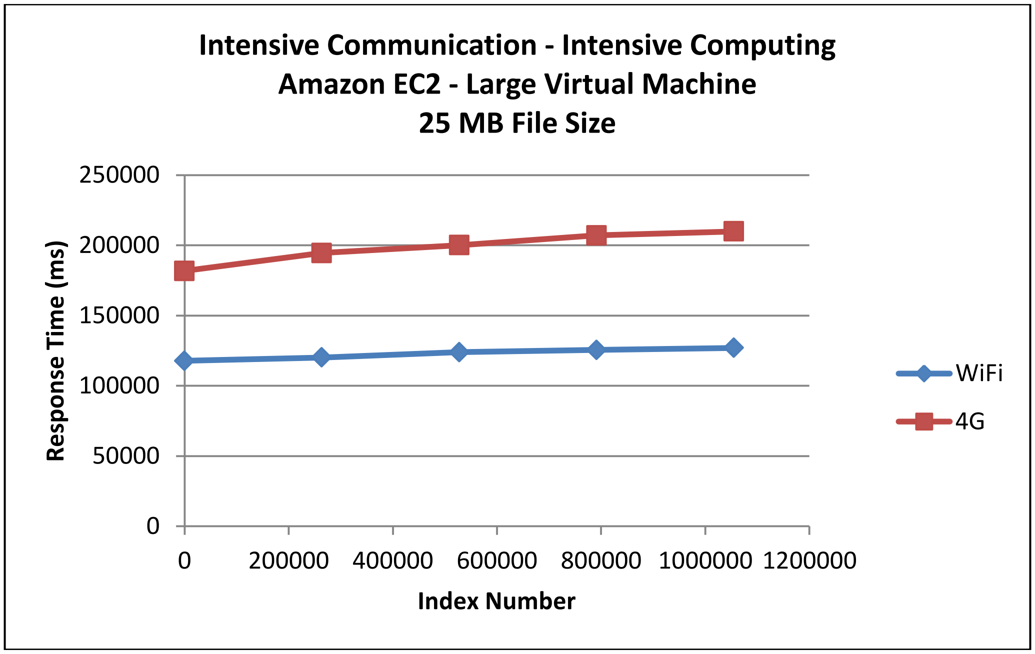

Figure 17: Amazon EC2 response time on large virtual machine when transferring 25MB data file.

Index	Wi-Fi (ms)	4G (ms)
0	190,948	311,993
440,000	194,162	318,144
880,000	196,169	330,293
1,320,000	202,631	336,543
1,760,000	205,083	344,596

Table 18: Amazon EC2 response time on a large virtual machine when transferring 42.5MB data file

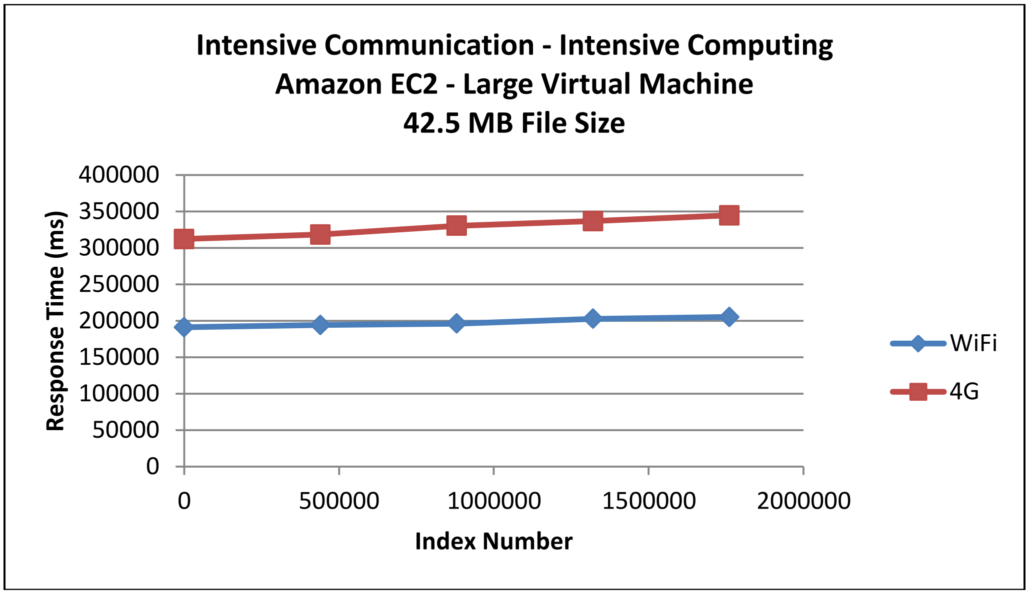

Figure 18: Amazon EC2 response time on large virtual machine when transferring 42.5MB data file.

Index	Wi-Fi (ms)	4G (ms)
0	∞	∞
463750	∞	∞
927500	∞	∞
1391250	∞	∞
1855000	∞	∞

Table 19: Amazon EC2 response time on a large virtual machine when transferring any file above 42.5MB data file.

6.1.3 Amazon EC2 vs. Microsoft Windows Azure

Tables 20, 21 and 22 present the energy consumption for Amazon EC2 and Windows Azure for intensive-communication / intensive-communication workload

Number	Small Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	4269	4183	4256	4823
9	4298	4191	4482	5282
9223	5491	5292	5567	6942
9223372	6074	5920	5342	5264
9223372036	9449	8859	5742	7346
9223372036854	11369	10094	7157	7634
9223372036854775	14799	15963	24467	246120
9223372036854775643	187700	184700	199654	201684
P-value	0.252645		0.335075	

Table 20: Amazon EC2 vs. Windows Azure - Response times for light-communication / intensive-computation on Small VMs.

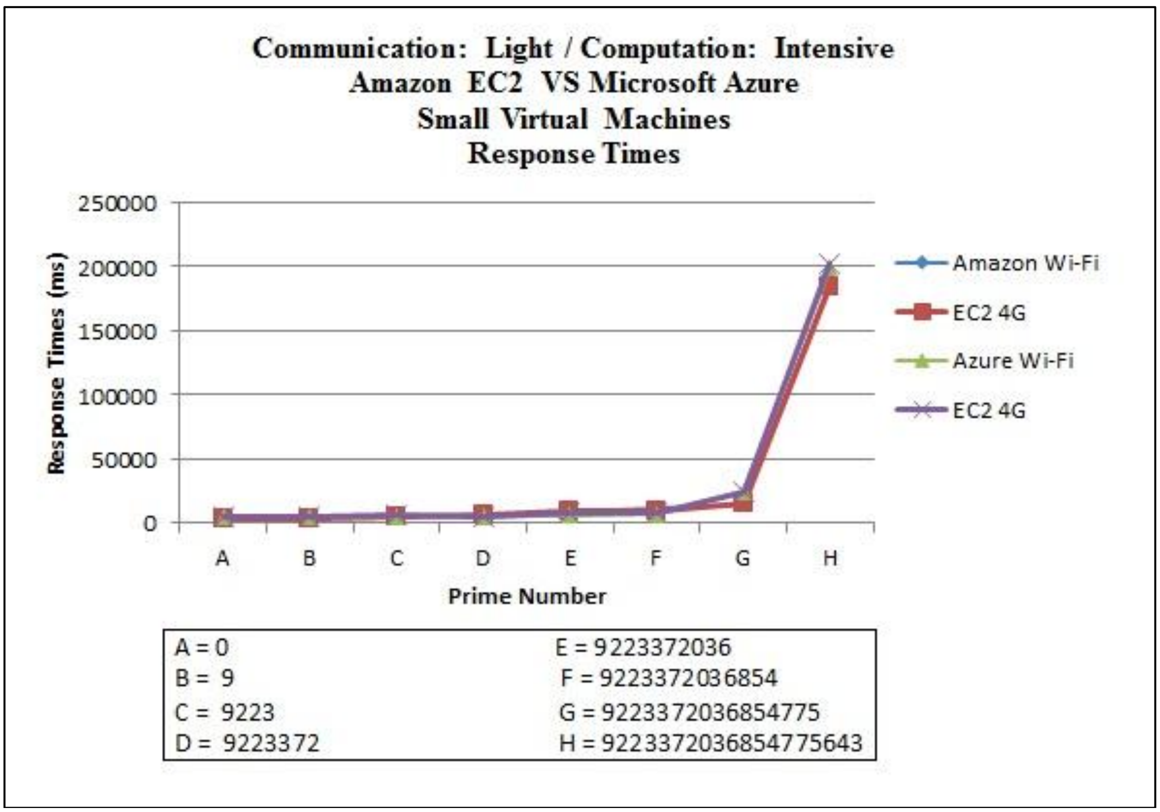

Figure 19: Amazon EC2 vs. Windows Azure - Response times for light-communication / intensive-computation on Small VMs.

Number	Medium Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	4136	4054	5022	5074
9	4171	4047	4879	5179
9223	5342	5235	5545	6335
9223372	6115	6184	5986	6510
9223372036	6622	6075	6296	6535
9223372036854	7190	6298	7998	7284
9223372036854775	11465	11344	13875	15955
9223372036854775643	59181	61945	63325	65110
P-value	0.384641		0.046007	

Table 21: Amazon EC2 vs. Windows Azure - Response times for light-communication / intensive-computation on Medium VMs.

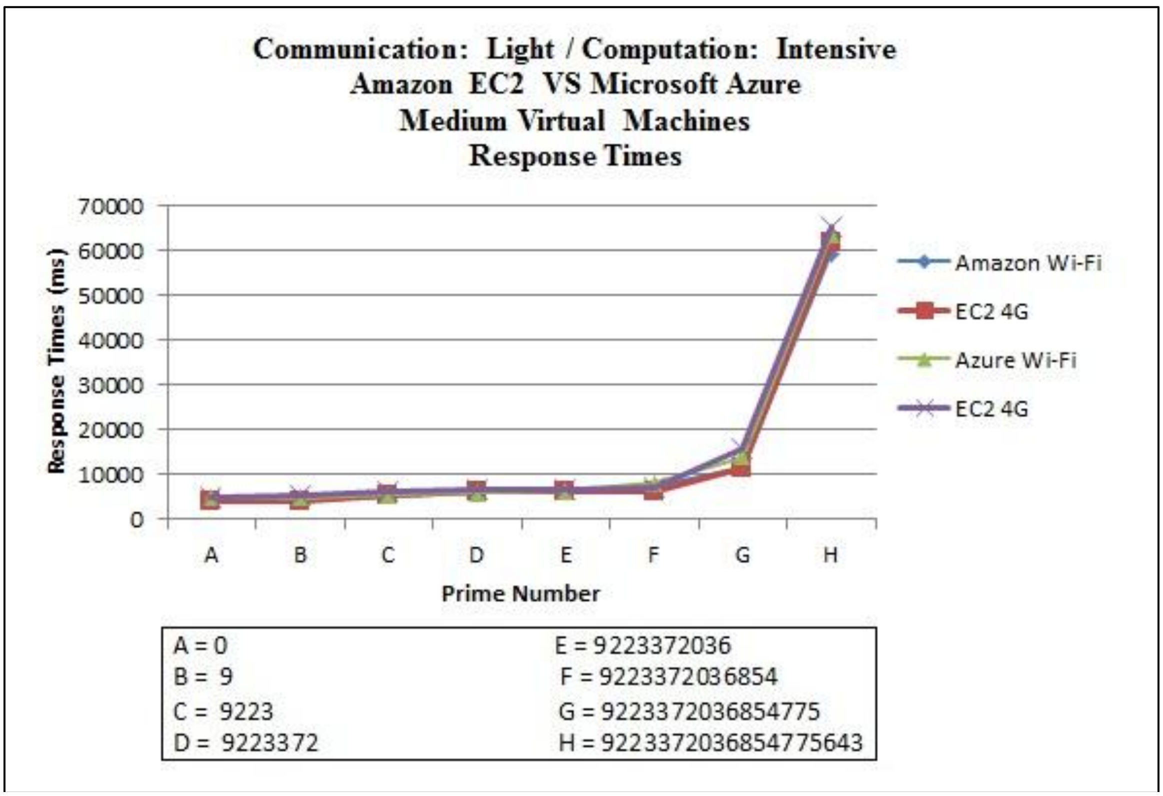

Figure 20: Amazon EC2 vs. Windows Azure - Response times for light-communication / intensive-computation on Medium VMs.

Number	Large Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	4013	4009	5008	5011
9	4137	4035	5543	5087
9223	5042	5089	6364	6395
9223372	6233	5824	6756	6592
9223372036	6766	5903	7556	6482
9223372036854	7191	6523	8314	6935
9223372036854775	10994	14568	15667	12535
9223372036854775643	54329	56784	57006	59495
P-value	0.202748		0.219365	

Table 22: Amazon EC2 vs. Windows Azure - Response times for light-communication / intensive-computation on Large VMs.

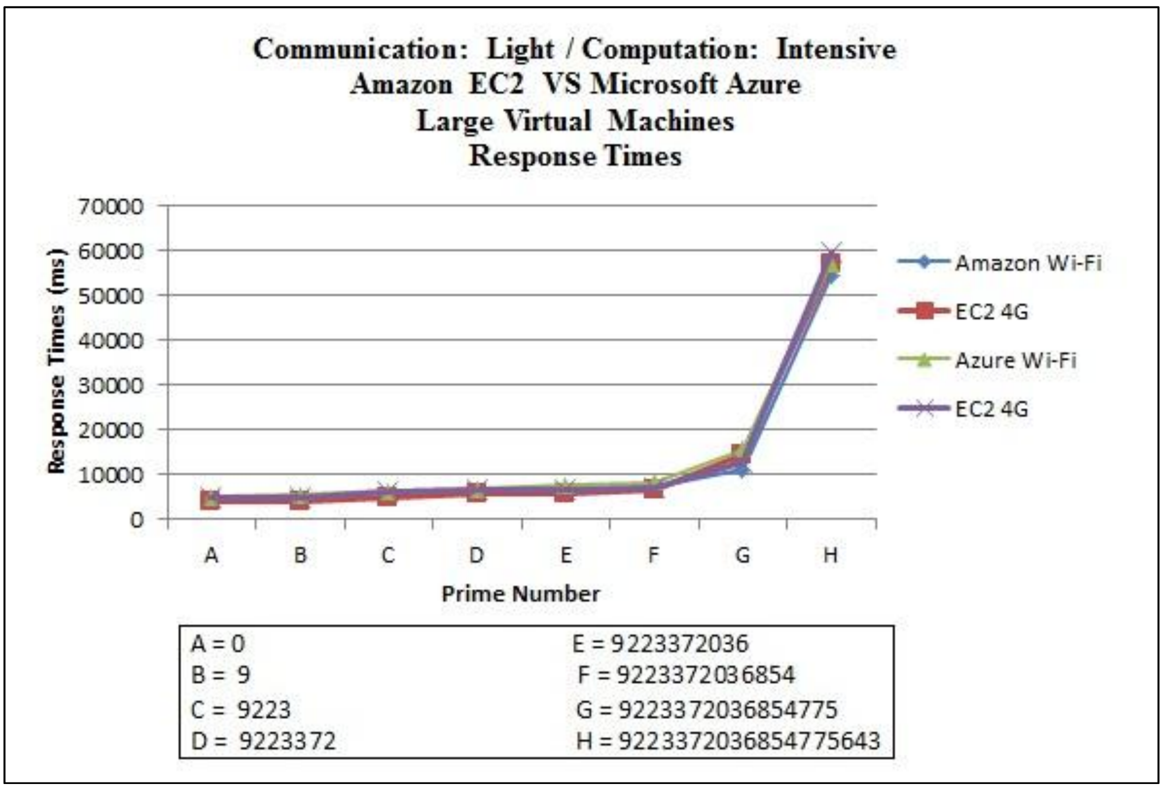

Figure 21: Amazon EC2 vs. Windows Azure - Response times for light-communication / intensive-computation on Large VMs.

Number	Small Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	0.33	0.32	0.47	0.5
9	0.34	0.33	0.48	0.49
9223	0.50	0.54	0.51	0.52
9223372	0.59	0.54	0.59	0.56
9223372036	0.62	0.59	0.61	0.58
9223372036854	0.66	0.64	0.76	0.82
9223372036854775	0.97	0.89	1.15	1.21
9223372036854775643	10.12	9.21	11.20	13.34
P-value	0.13479		0.162584	

Table 23: Amazon EC2 vs. Windows Azure – Energy consumption for light-communication / intensive-computation on Small VMs

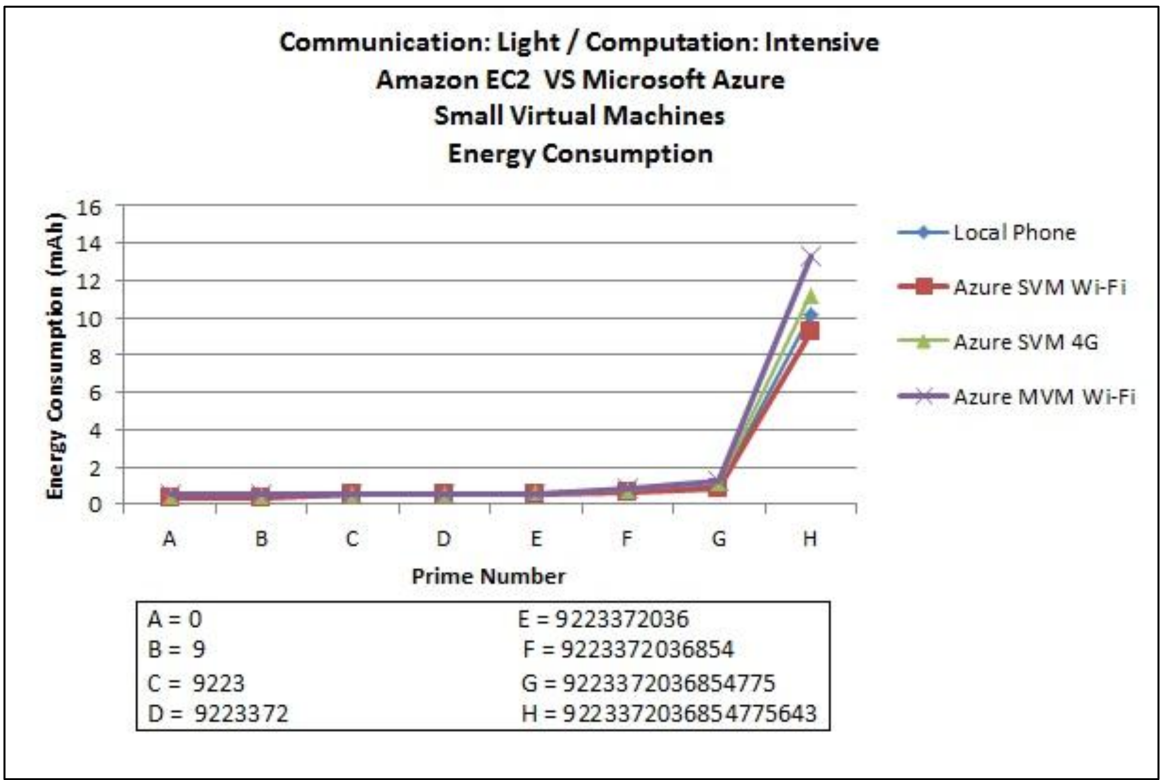

Figure 22: Amazon EC2 vs. Windows Azure – Energy consumption for light-communication / intensive-computation on Small VMs.

Number	Medium Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	0.39	0.41	0.49	0.42
9	0.42	0.44	0.50	0.43
9223	0.42	0.44	0.51	0.49
9223372	0.50	0.48	0.60	0.52
9223372036	0.52	0.50	0.58	0.54
9223372036854	0.55	0.57	0.61	0.68
9223372036854775	0.90	0.90	1.24	1.19
9223372036854775643	3.17	3.23	4.17	3.89
P-value	0.108419		**0.046527**	

Table 24: Amazon EC2 vs. Windows Azure – Energy consumption for light-communication / intensive-computation on Medium VMs

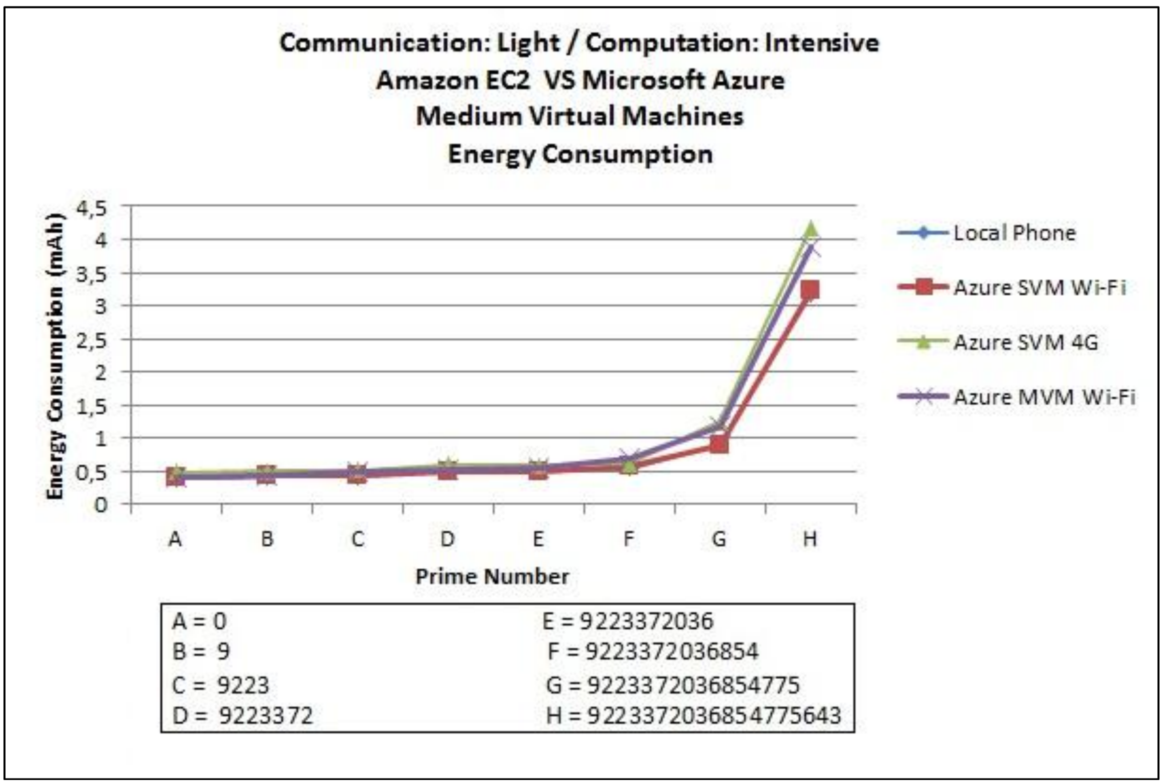

Figure 23: Amazon EC2 vs. Windows Azure – Energy consumption for light-communication / intensive-computation on Medium VMs

Number	Large Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	0.38	0.41	0.45	0.42
9	0.4	0.42	0.48	0.42
9223	0.42	0.42	0.49	0.44
9223372	0.52	0.48	0.55	0.50
9223372036	0.58	0.58	0.62	0.56
9223372036854	0.57	0.67	0.62	0.68
9223372036854775	0.7	1.28	1.10	1.26
9223372036854775643	3.02	3.14	3.48	3.66
P-value	0.09823		0.308452	

Table 25: Amazon EC2 vs. Windows Azure – Energy consumption for light-communication / intensive-computation on Large VMs

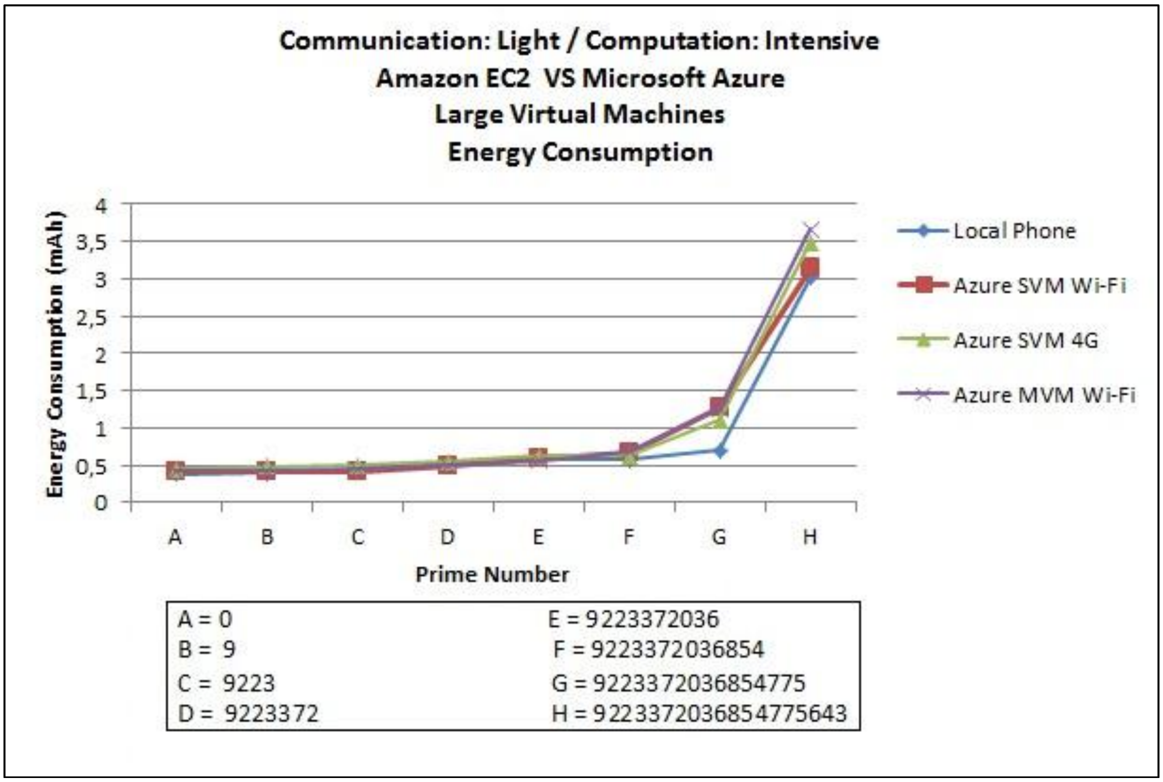

Figure 24: Amazon EC2 vs. Windows Azure – Energy consumption for light-communication / intensive-computation on Large VMs

File Index	Small Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	39196	40365	47529	50529
1.000	39745	38045	48212	49363
20.000	42380	40806	49165	49345
40.000	43185	42185	49960	47646
80.000	43483	42803	50081	51563
150.000	43842	39477	49470	52356
160.000	44408	43367	50790	53058
P-value	0.23799764		0.17184667	

Table 26: Amazon EC2 vs. Windows Azure – Response Times for intensive-communication / intensive-computation on Small VMs

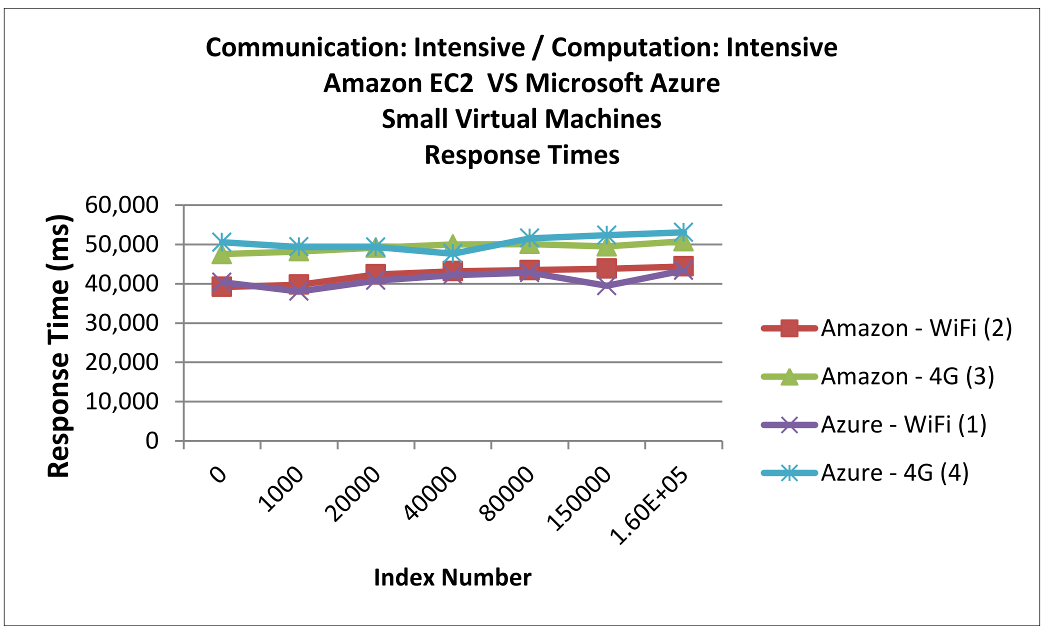

Figure 25: Amazon EC2 vs. Windows Azure – Response Times for intensive-communication / intensive-computation on Small VMs

File Index	Medium Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	38338	36746	46091	44606
1.000	38481	39938	47627	46679
20.000	39931	40936	48813	49374
40.000	39013	41683	47814	48834
80.000	41375	41784	49963	47944
150.000	42881	38366	50916	51984
160.000	43354	40757	50604	50475
P-value	0.32715623		0.28804057	

Table 27: Amazon EC2 vs. Windows Azure – Response Times for intensive-communication / intensive-computation on Medium VMs

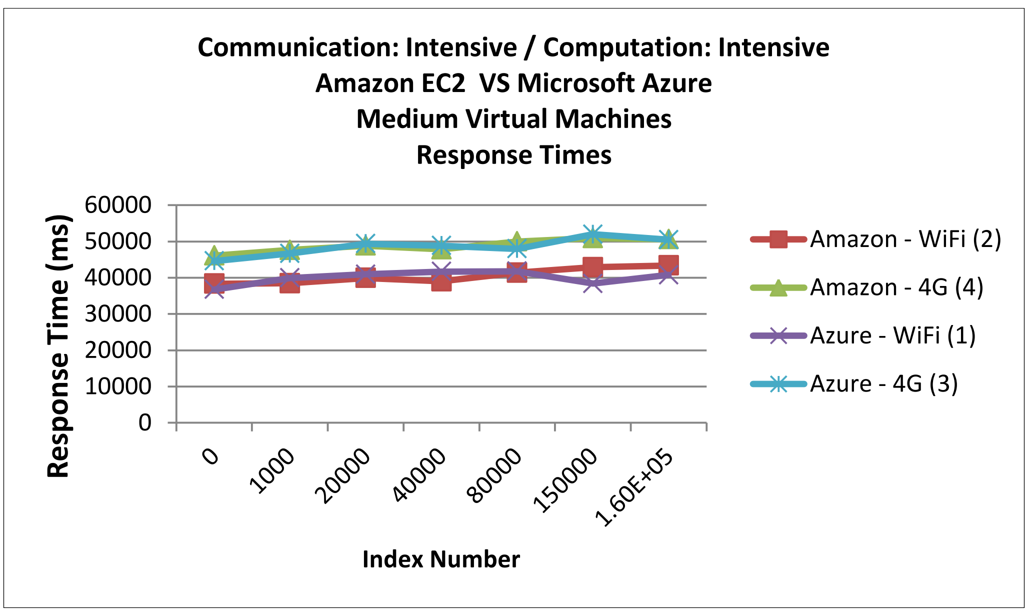

Figure 26: Amazon EC2 vs. Windows Azure – Response Times for intensive-communication / intensive-computation on Medium VMs

File Index	Large Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	35117	39034	46221	47467
1.000	35430	40382	47039	47927
20.000	37619	40592	47484	46038
40.000	37909	39887	50638	49437
80.000	37755	40945	49570	48205
150.000	37934	39430	49734	47661
160.000	39121	38592	48923	48024
P-value	**0.002546**		0.0975374	

Table 28: Amazon EC2 vs. Windows Azure – Response Times for intensive-communication / intensive-computation on Large VMs

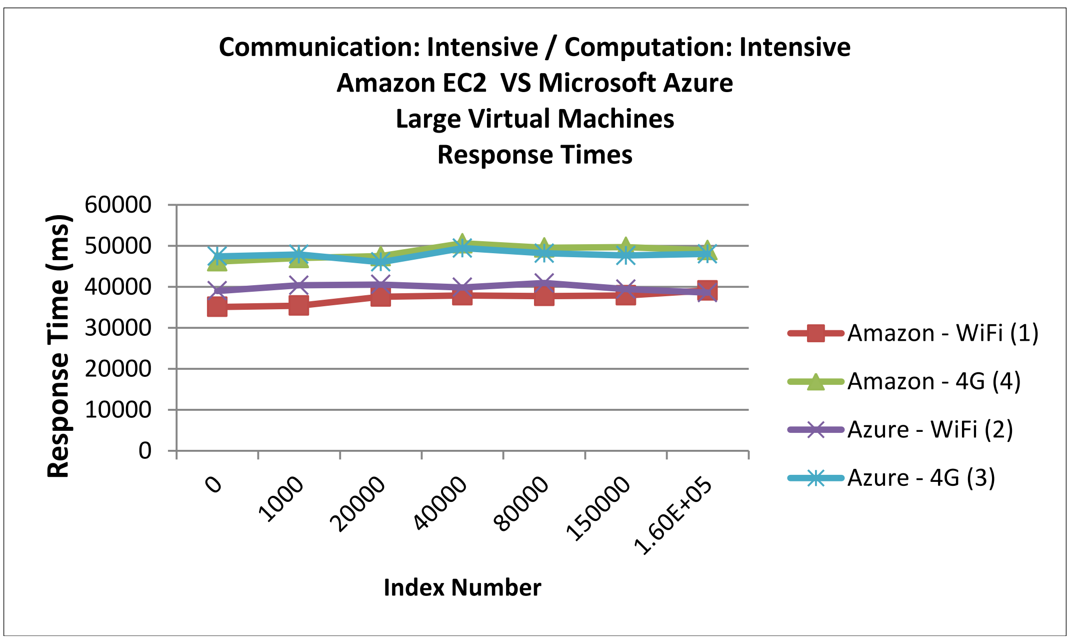

Figure 27: Amazon EC2 vs. Windows Azure – Response Times for intensive-communication / intensive-computation on Large VMs

File Index	Small Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	2.30	2.37	4.04	4.1
1.000	2.31	2.22	4.05	4.08
20.000	2.50	2.39	4.05	4
40.000	2.45	2.4	4.03	4.06
80.000	2.53	2.24	4.05	4
150.000	2.61	2.3	4.06	4.1
160.000	2.64	2.51	4.05	4.05
P-value	**0.02080521**		0.31139294	

Table 29: Amazon EC2 vs. Windows Azure – Energy Consumption for intensive-communication / intensive-computation on Small VMs

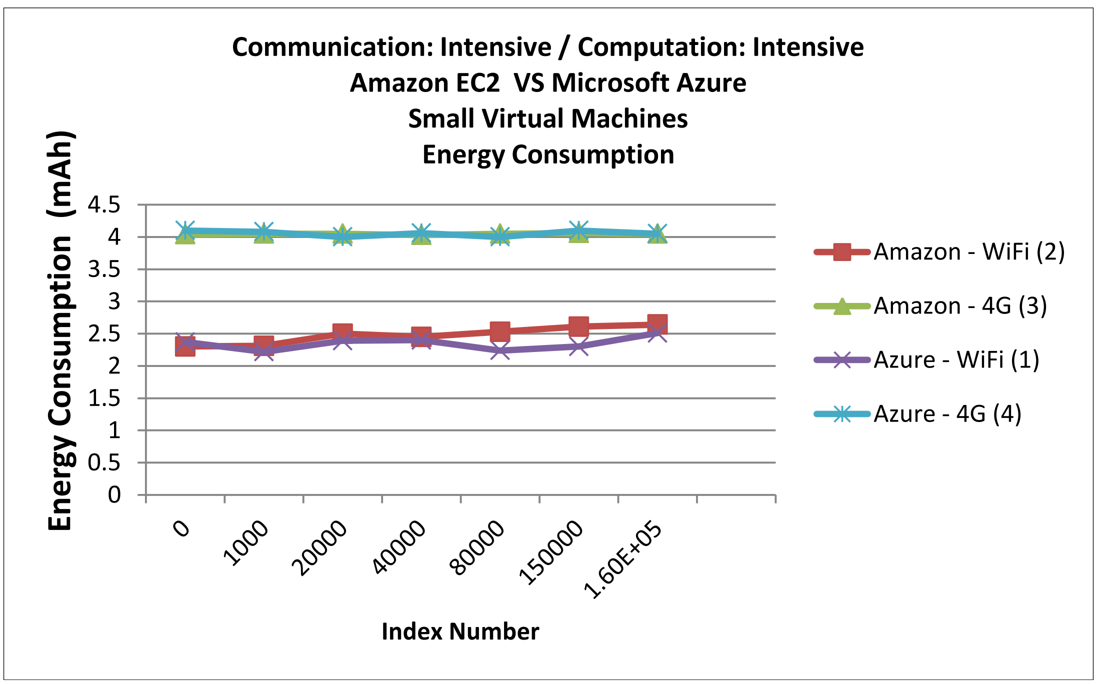

Figure 28: Amazon EC2 vs. Windows Azure – Energy Consumption for intensive-communication / intensive-computation on Small VMs

File Index	Medium Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	2.18	2.22	3.64	3.66
1.000	2.18	2.23	3.63	3.62
20.000	2.34	2.19	3.62	3.6
40.000	2.35	2.28	3.54	3.56
80.000	2.42	2.32	3.78	3.72
150.000	2.44	2.38	3.86	3.84
160.000	2.44	2.36	3.84	3.74
P-value	0.324849		0.05187388	

Table 30: Amazon EC2 vs. Windows Azure – Energy Consumption for intensive-communication / intensive-computation on Medium VMs

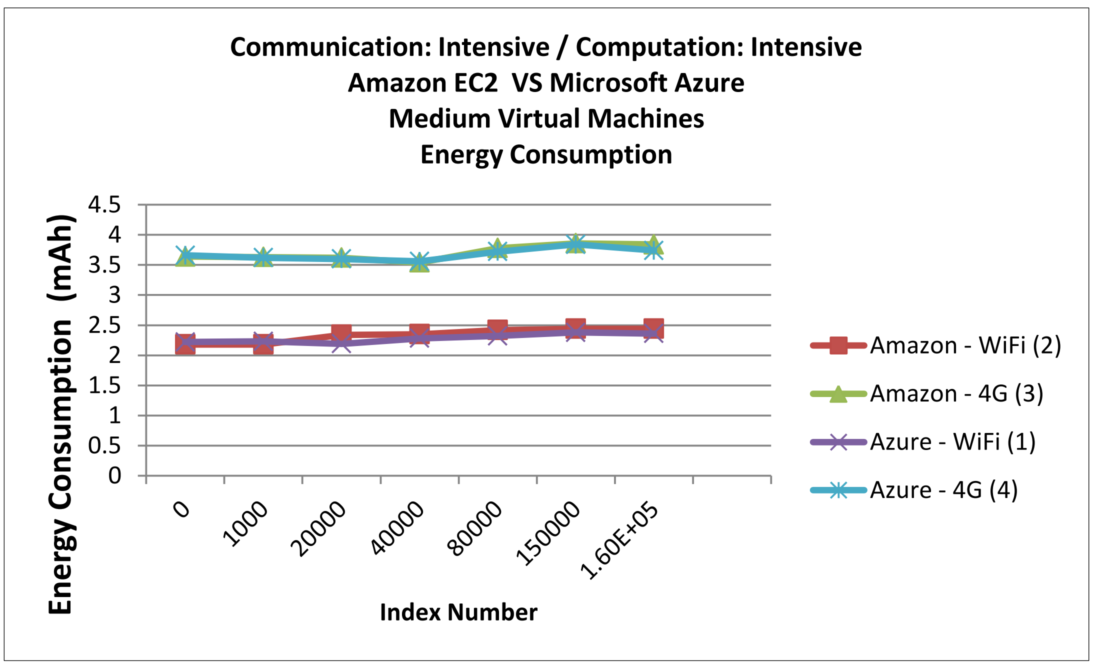

Figure 29: Amazon EC2 vs. Windows Azure – Energy Consumption for intensive-communication / intensive-computation on Medium VMs

File Index	Large Virtual Machines			
	Wi-Fi		4G	
	Amazon EC2	Windows Azure	Amazon EC2	Windows Azure
0	2.07	2.2	3.48	3.52
1.000	2.09	2.22	3.51	3.54
20.000	2.16	2.24	3.53	3.48
40.000	2.2	2.18	3.65	3.58
80.000	2.22	2.22	3.82	3.72
150.000	2.25	2.24	3.85	3.9
160.000	2.28	2.18	3.89	3.98
P-value	0.19588181		0.47971679	

Table 31: Amazon EC2 vs. Windows Azure – Energy Consumption for intensive-communication / intensive-computation on Large VMs

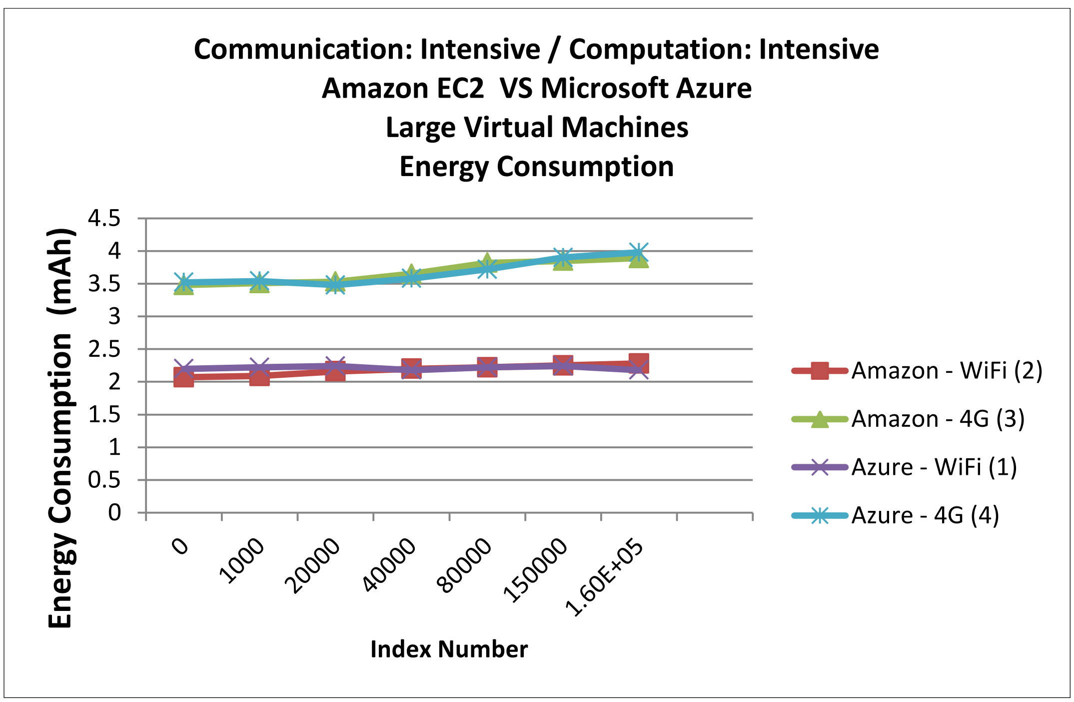

Figure 30: Amazon EC2 vs. Windows Azure – Energy Consumption for intensive-communication / intensive-computation on Large VMs

Chapter 7

CONCLUSIONS

The following is a discussion of the results presented in chapter 6, organized by study. Mobile devices as well as Amazon EC2 and Microsoft Windows Azure cloud services were tested using an Android app to perform locally and offload remotely mobile processing.

7.1 Workloads studies

7.1.1 Light Communication / Intensive Computation workload

The goal of this experiment was to find the next prime number of a given long number as fast as possible and saving as much energy as possible. The test combined testing this workload locally and offloading the processing to Amazon EC2 and Microsoft Windows Azure.

While most of the p values were not significant ($p>0.05$) for overall data set when comparing local computation with Amazon EC2 and Azure, there was a significant difference in computing time and energy consumption when the workload was more computing intensive. The reason is that there was a communication penalty in response times and energy consumption when offloading this workload to the cloud.

After using Wi-Fi and EC2 when the workload was more computing intensive, the response time was up to 28 times faster and there was a 270% of energy saving on the mobile device. The results indicate the suitability of the cloud for computing-intensive workloads.

7.1.2 Intensive Communication / Intensive Computation workload

The goal of this experiment was to find a matched value for a given index in a text file. The test combined testing this workload locally and offloading the processing to Amazon EC2 and Microsoft Windows Azure.

Most of the p values were significant ($p<0.05$) for overall data set when comparing local computation with Amazon EC2 and Azure. Therefore, there was a significant difference in computing time and energy consumption when offloading the workload.

After using Wi-Fi and EC2, the response time was up to 1.6 times faster and there was 53% of energy saving on the mobile device. The results indicate the suitability of the cloud for computing-intensive workloads.

Additionally, when executing the workload locally the app crashes due it having reached its memory heap size assigned by the Dalvik VM.

Results indicate the suitability of the cloud for computing-intensive workloads.

7.2 Mobile Memory Saturation Breakpoint

The goal of this experiment was to find a breaking point where the mobile app cannot transmit data over to the cloud due the workload size was too great compared to the mobile memory heap size and the Dalvik VM kills the app.

After offloading a varying intensive-communication / intensive-computation workload to Amazon EC2 using Wi-Fi and 4G, a breakpoint found showed that there are certain workload sizes not suitable to be offloaded because loading them into memory could exceed the assigned memory heap size causing the app crashes. After transmitting different file sizes, the results show that the app performs well when transmitting file sizes below 42.5 MB over any communication link. When trying to transmit files over 42.5 MB the app misbehaves; it crashes due to reaching its memory heap size. Moreover, when the app tries to allocate more memory the Android memory manager kills the app creating an *out of memory error*. Therefore, the memory heap size assigned by the Dalvik VM for an app is a very important factor when deciding to offload the workload.

Additionally, it is vital to clarify that heap size is device-dependent and the Dalvik VM increases it on demand until it reaches its maximum assigned memory size. Although it is possible to manually set the maximum heap size in runtime, it is strongly not recommended because it may degrade the overall system performance as well as cause other apps to become unstable.

7.3 Amazon EC2 and Microsoft Windows Azure

The goal of this experiment was to compare response times and energy consumption measurements on Amazon EC2 and Microsoft Windows Azure.

While most of the p values were not significant ($p>0.05$) for the overall data set when comparing Amazon EC2 to Microsoft Windows Azure, there was a slightly significant difference in computing time and energy consumption when offloading to Amazon EC2.

Based on the previous measurements, Microsoft Windows Azure response times were slightly higher than the ones on Amazon EC2. Therefore, it also means that energy consumption tended to be higher when performing processing on Microsoft Windows Azure.

7.4 Future Research

This study is limited to compare mobile local processing to different VM sizes on Amazon EC2 and Microsoft Azure using Wi-Fi and 4G communication links.

An extension to this study on mobile processing could include evaluating other mobile operative systems such as iOS, Windows Phone, BlackBerry on different mobile devices configurations as well as evaluating additional cloud services such as Google Cloud

Engine, IBM SmartCloud and so on. Additionally, different Android mobile devices can be included in the research in order to determine their influence in the experiment.

Since this study covers offloading mobile processing to the cloud, it could serve as a reference for future studies involving the development of any kind of middleware able to make transparent decisions on whether perform mobile tasks locally on mobile devices or remotely on cloud providers based on response times and energy consumption, file size, computation and bandwidth.

REFERENCES

Electronic sources

[Android13]
"Get the Android SDK", http://developer.android.com/sdk/index.html, last accessed October 02, 2013.

[Bhagavathi13]
Bhagavathi, K., "Performance Evaluation of Data-Intensive Computing in the Cloud", Theses and Dissertations, University of North Florida, School of Computing, Florida 2013.

[Brahler10]
Brahler, S., "Analysis of the Android Architecture", University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association, Baden-Wuerttemberg 2010.

[Bulmer09]
Bulmer, S., "5 Ways Cloud Computing Will Revolutionize Business", http://wisepreneur.com/innovation/5-ways-cloud-computing-will-revolutionize-business, last accessed September 15, 2013.

[Dubroy11]
Dubroy, P., "Memory Management for Android Apps", Google I/O Development Conference, Mountain View, CA, 2011.

[Ehringer10]
Ehringer, D., "The Dalvik Virtual Machine Architecture", http://show.docjava.com/posterous/file/2012/12/10222640-The_Dalvik_Virtual_Machine.pdf, last accessed Sept 08, 2013.

[Espinar11]
Espinar, P., "Android Memory management", http://prezi.com/k_k4gil_ecas/android-memory-management, last accessed September 13, 2013

[Huerta10]
Huerta, C. and Dongman L., "A Virtual Cloud Computing Provider for Mobile Devices", ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond. MCS'10, 2010, San Francisco, California, 2010.

[Husted11]
Husted, N., Qureshi, S., Tariq, D., and Gehani, A., “Android Provenance: Diagnosing Device Disorders”, 5th USENIX Workshop on the Theory and Practice of Provenance, April 2013.

[Kelényi10]
Kelényi, I., and Nurminen J.K., “CloudTorrent – Energy-Efficient BitTorrent Content Sharing for Mobile Devices via Cloud Services”, Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE, 2011.

[Kumar10]
Kumar, K. and Yung-Hsiang Lu, “Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?”, IEEE Computer Society, West Lafayette, PA, 2010.

[Microsoft13]
“Microsoft Corp”, http://www.windowsazure.com/en-us/overview/what-is-windows-azure/, last accessed September 12, 2013.

[Miettinen10]
Miettinen, A.P., and Nurminen J.K., “Energy efficiency of mobile clients in cloud computing,” 2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '10), Boston, MA, June, 2010

[Ranganathan10]
Ranganathan, P., “Recipe for Efficiency: Principles of Power-Aware Computing”, Communications of the ACM 53, Palo Alto, CA, 2010.

[Satyanarayanan09]
Satyanarayanan, M., Bahl, P., Caceres, R., and Davies, N., “The Case for VM-based Cloudlets in Mobile Computing”, IEEE Pervasive Computing, Vol. 8, No. 4, October-December, 2009.

[Sysfore11]
Sysfore, “Windows Azure Platform: An Overview”, http://www.sysfore.com/PDF/Windows_Azure_Platform_An_Overview.pdf, last accessed October 02, 2013.

VITA

Jesus Zambrano has a Bachelor of Engineering from Universidad Nacional Experimental del Táchira, Venezuela in Computer Sciences and Engineering in 2009, and expects to receive a Master of Science in Computer and Information Sciences from the University of North Florida, July 2015. Dr. Sanjay Ahuja of the University of North Florida is Jesus's thesis advisor. Jesus has been an Application Developer in Mobile based products at Nationwide, Columbus, Ohio for six months. Then he was as Android Software Developer at iMobile3, Jacksonville, Florida. He currently works as a Senior Android Developer in Grability, Bogota, Colombia. Jesus also has 1+ years experience at Banfoandes Venezuela using JSP, PHP, and PostgreSQL. Jesus's academic work has included use of Java, C, C++, Microsoft Visual Studio.Net, Oracle, SQL, JavaCorba, RMI, and networking.

Jesus likes to incorporate his computer skills in real life situations needing the use of Mobile Apps and Cloud Computing. His main goal is to become a respected Software professional who strives to produce excellence in every area useful for the society, and to maintain ethical, high quality standards and respectful approaches to working with the vast power of computing.