
Software Requirements Specification

for

vTimeSeries

Version 1.1

Prepared by

Group Name: Null Terminators

Vladimir Adam
Nick Cross
Ryan McGinley
Cesar Polanco
Karanbir Toor

4126140
3855921
4510285
4428207
4395141

vladam91@gmail.com
njcross1990@gmail.com
kirikenz@gmail.com
xbox.inann@gmail.com
karanbirsinghtoor@gmail.com

Instructor: Chandra Krintz

Course: CS189A

Lab Section: *Friday, 12:00-12:50*

Teaching Assistant: *Stratos Dimopoulos*

Date: 2/12/2013

Contents

CONTENTS	II
REVISIONS	II
1 INTRODUCTION	1
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	2
1.5 REFERENCES AND ACKNOWLEDGMENTS	2
2 OVERALL DESCRIPTION.....	3
2.1 PRODUCT PERSPECTIVE	3
2.2 PRODUCT FUNCTIONALITY	3
2.3 USERS AND CHARACTERISTICS	4
2.4 OPERATING ENVIRONMENT.....	4
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	4
2.6 USER DOCUMENTATION.....	5
2.7 ASSUMPTIONS AND DEPENDENCIES.....	5
3 SPECIFIC REQUIREMENTS	5
3.1 EXTERNAL INTERFACE REQUIREMENTS	5
3.2 FUNCTIONAL REQUIREMENTS	6
3.3 PRD TABLES.....	7
3.4 BEHAVIOUR REQUIREMENTS	8
4 OTHER REQUIREMENTS	11
APPENDIX A – DATA DICTIONARY	11
APPENDIX B - GROUP LOG	11

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.00	Null Terminators	Initial version created for review by mentor.	2/8/2013
1.01	Null Terminators	Based on feedback from mentor	2/11/13
1.1	Null Terminators	Based on feedback from Professor, added PRD	3/6/13

1 Introduction

1.1 Document Purpose

The purpose of this document is to provide a detailed and complete specification of vTimeSeries. This project is two-fold; a system to manage the collection and visualization of statistics collected from hosts and virtual machines. The second system will provide some insight for system administrators relating to questions about VM performance management, host capacity, and power management. The document will follow a similar format, first describing the system of statistic management and then discussing the question addressing portion of the program.

The rest of the Software Requirement Specification will go over the different aspects of the project; section 2 will give a broader, more generalized view of the project. The overview will cover some of the benefits associated with the project as well as the general idea behind what the software will try to accomplish. Section 3 will go into the specifics of the software, explaining how each portion will be implemented and what constraints were faced.

1.2 Product Scope

vTimeSeries is a software application aimed toward users, such as system administrators and managers. These users will utilize the product to answer queries regarding host statistics and visualize these statistics to observe trends. vTimeSeries will be an independent program, but it will work with the already existing DRS; vTimeSeries will expand on the length of time of the data collected and the amount used in decision making.

The first problem vTimeSeries is solving is data visualization. Currently statistics are only available in numerical form. This makes it very difficult to observe trends or quickly asses and compare performance across a data center. vTimeSeries will ameliorate the situation by providing real time graphs and tools to compare various statistics.

Another problem that vTimeSeries will address is that of predictive load balancing. The current system used by VMware, their DRS system, performs dynamic load balancing in a reactive manner. The purpose of the program is to facilitate predictive actions for the DRS, allowing data centers to be better prepared for the load throughout the day. This system will also reduce the amount of on-the-fly load balancing that needs to occur.

The project will also be used by system administrators to analyze the behavior of their virtualized environment as well as reason about it. Graphical representations will allow users to examine the data and answer what-if queries they might have. The module will also be able to look for periodic trends and report those to the admins.

1.3 Intended Audience and Document Overview

This document is intended for Team Null Terminators, VMware, as well as faculty, students, and teaching assistants associated with UCSB Capstone and CS189A and CS189B.

This document will have information that may be deemed pertinent for each member of the

intended audience, but it is likely that certain sections will be of more interest to each type of audience member. Those interested in the functionality of the software should read section 2 (Overall Description) in order to learn more about the capabilities of the software. Those interested in the internal workings of the software should read section 3 (Specific Requirements). Finally, those interested in the development process should read through the entirety of this document.

1.4 Definitions, Acronyms and Abbreviations

API – Application Programming Interface
CPU – Central Processing Unit
CS – Computer Science
JRE – Java Runtime Environment
DPM – Distributed Power Management
DRS – Distributed Resource Scheduler
GUI – Graphical User Interface
Gb – Gigabit
NFS – Network File System
RAM – Random Access Memory
SDK – Software Development Kit
SRS – Software Requirements Specifications
VC – Virtual Center
VM – Virtual Machine

1.5 References and Acknowledgments

A. Gulati, A Holler, M. Ji, G. Shanmuganathan, C. Waldspurger, X. Zhu. (Date N/A [remove this]). *VMware Distributed Resource Management: Design, Implementation, and Lessons Learned* [Online]. Available: <http://labs.vmware.com/publications/gulati-vmtj-spring2012>

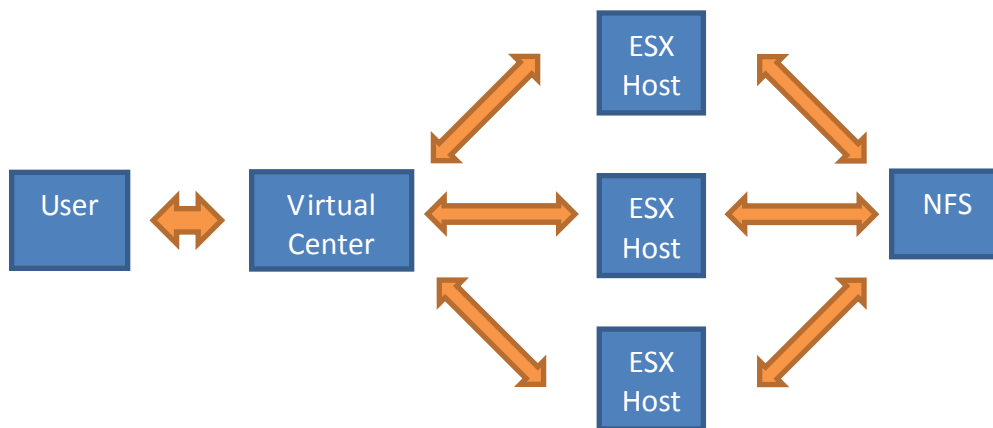
2 Overall Description

2.1 Product Perspective

The DRS feature from VMware currently uses small quantities of short term data, up into the tens of minutes to make decisions about initial placement of VMs on hosts and to perform load balancing to avoid any hot-spots at the host level. Although this is useful for certain set of operations, there is a class of management operations which can benefit from long-term analysis of performance data. This analysis will be based on time-series analysis that captures correlations and statistical multiplexing among workloads in order to provide better answers. vTimeSeries will create the tools needed to accomplish this goal.

vTimeSeries can be divided into two parts. The first part consists of collection and visualization of data from the ESX hosts. The data will be collected from VMware's vCenter software through the use of their SDK. This collected data will then be stored on a database, where an interface will provide it to the rest of our program when necessary. When providing any visualization, the database will also be queried and will then provide information to the necessary GUI components.

The second part uses time-series analysis to influence the decision making from admins and answer certain what-if queries from them. The analyzer will be a separate module, which will interact with the database through the interface. After receiving the data that it has requested, the analyzer will perform a variety of analysis techniques on the data. These techniques will include a moving average, Fourier Transform, and the Savitzky-Golay smoothing filter.



2.2 Product Functionality

- Provide Visual representations of data analysis to user
- Make VM migration recommendations based on time-series analysis, including moving average
- Coherent performance data will be presented to answer what-if queries about host management

2.3 Users and Characteristics

Administrators

- Will use software frequently
- Will use information provided via visualization to answer what-if queries
- Examples of possible queries:
 - Can I power off a cluster/host in order to save power?
 - Should I add more hosts in order to prevent overloading?
 - How many more VMs can I add to this cluster/host?
 - Can I migrate VM A from host B to host C?
 - Should I migrate VMs off of this host?
- Need to have extensive knowledge of VMs, vCenter
- Needs to know trade-offs of VM migration and powering VMs on and off

VMware, VMware employees and its affiliates

- Will view and use source code to further test and add
- Need to know about DRS
- Need in depth knowledge of n-house software
- Need extensive knowledge of vCenter, vSphere, DRS and Time Series

2.4 Operating Environment

We will be using Github for our repository. The code will be developed on our personal computers and deployed on the vCenter computer. We will also run the code on a separate machine and connect to vCenter remotely.

The supervising computer will use vCenter, installed on Microsoft Windows Server 2012. There will be three hosts, each running VMware ESX hypervisor. For simulating load on these ESX hosts, we will use load simulating software provided by VMware called VMMark. Each of the ESX hosts will connect to an NFS server running a variant of FreeBSD to host data needed by the virtual machines. Keeping the data on network storage allows for smooth migration of VMs.

2.5 Design and Implementation Constraints

Hardware Specifications

- 3 Servers (ESX Hosts)
 - 2x Quad Core @2.66 GHz, 16 GB of RAM, 2x 72GB HDD
- 1 Server (NFS)
 - 2x Dual Core @3.00 GHz, 4 GB of RAM, 6x 72GB HDD
- 1 Server (vCenter)
 - 2x Dual core @3.00 GHz, 4 GB of RAM, 4x 72GB HDD
- 1 Gigabit Network Switch (Managed)

Software Specifications

- vSphere
- vCenter
- ESX Hypervisor
- FreeNAS OS
- Google Web Toolkit SDK

Programming Specifications

- Requires JRE to be installed on user machine

Other

- VMs need to operate for a significant amount of time in order to provide data to analyze
 - This means that the cluster will have to run for a set amount of time before vTimeSeries can perform its tasks.

2.6 User Documentation

A user manual explaining what the functionality and usage of the visualization software will be required. Additionally, the team developing the software would potentially need to be available in case of questions or problems with the software once it starts being used by other users/administrators. Our source code will be heavily documented and follow vCenter's SDK and API documentation. This will make it easier for other people to understand and continue development on our product if needed.

2.7 Assumptions and Dependencies

We are assuming that the user will have a working data center that has a managing computer running vCenter with one or more clusters participating in it. We are also assuming that the user has some hardware to run our module, which will consume storage and CPU to store and analyze data. We are further assuming that the simulated workload provided through VMMark is representative of a normal user's virtual workload. Finally, we are assuming that the computers are running the correct VMware software and the correct version that supports the API used by our module.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- Graphs showing various performance metrics for VMs, hosts, and clusters in a data center
- Graphs showing aggregated metrics at VM, host, and cluster level
- Detailed statistics about various metrics

3.1.2 Hardware Interfaces

vTimeSeries will communicate with the cluster using vCenter's API calls provided through the vCenter web service SDK. Three ESX hosts will use a NFS server as a shared storage to synchronize and manage the files. The hosts, NFS, and the vCenter workstation will all be connected through a 1 Gb switch.

3.1.3 Software Interfaces

The cluster will consist of 3 ESX hosts all running VMWare's vSphere software. The NFS will be running FreeNAS and the managing machine will be running Windows 2012 Server edition. The application software used on the administrator/management server will be vCenter, which provides support for host migration and API calls to collect data. We will be using Eclipse on our personal computer to write the Java code that will be used in conjunction with vCenter to collect data and make decisions. We will also be using vCenter Client that allows us to login in to the vCenter computer remotely from anywhere.

3.1.4 Communications Interfaces

N/A

3.2 Functional Requirements

1) Opening vTimeSeries

- On opening up the program, on a supported browser, the user will be prompted to login and provide data center information.
- After login, plots displaying VM Happiness at the cluster level will be displayed with the time scale set to one week.

2) Selecting a data set

- Upon successfully opening the program the "splash screen" will be displayed. This will enable the user to select a cluster that they wish to investigate visually.
 - The active data set will by default display VM Happiness but this can be changed by the user to display any other available statistic
- Any time after the splash screen, the user will be able to click on any of the assorted line graphs displayed per page in order to use the corresponding data set as the currently active data set.
 - The line graphs displayed per page will always consist of the available information at the lower levels of data; a cluster will allow you to select from graphs displaying information at the host level, etc.
- Alternatively, the user will be able to select from a quick selection menu in the top right corner. This will display the selected data set as the active data set, exactly as if the user had navigated to it through standard means.

3) Answering what-if queries

- The user will be able select a question from a list of possible queries to the system. The graphs will then be modified and display data in order to predict the outcomes of certain actions that the user has "asked" to take.
- This combination of data will produce hypothetical graphs which will show the user the result of their action. The user will then look at this and see if it would be viable to perform the action.
 - For example, a user will be able to judge if adding a new VM to a host would be prudent given the resulting load.

4) Changing the time scale

- Upon successfully opening the program the "splash screen" will be displayed. Users can then modify the time axis, by selecting the range and what particular days to represent.

5) Live Graph Dates

- New stats are available from the data center every 20 seconds. All graphs need to be updated in sync with this interval, thus providing real time data.
- 6) Comparing statistics
- When looking at an active data set. The user will be able to compare the present data with other data of their choosing.
 - Users can compare the same statistic across different hardware.
 - Users can also compare different statistics across the same or different hardware.
 - Users add a statistic to compare by selecting it from the comparison tab.
 - The comparison tab presents the comparable statistics in a hierarchical fashion.

3.3 PRD Tables

Navigation

Requirement	Description	Hours	Priority
QuickSearch	This menu allows the user to quickly navigate through the web interface.	8	D
Small Graph Linking	This allows users to use the child graphs at the bottom of each page to get to the main screen for that child.	4	M

Graphing

Requirement	Description	Hours	Priority
Display Correct Stats	Will graph a series of archived statistics from the database onto the web interface.	18	M
Add/Remove Entity	Allows the user to add a particular entity's statistic to the graph.	12	I
Add/Remove Stats to Graph	Allows the user to select which performance counters (statistics) are displayed on the graph. This adds an item to the legend and a line to the graph.	8	I
Specify Time Range	Allows the user to specify a range within 24 hours for the x-axis.	4	I

Peripherals

Requirement	Description	Hours	Priority
Average in Days	Allows the user to choose the days which are going to be averaged into the 24 hour period in the graph.	10	D
Specify Date Range to Average	Allows the user to specify the dates between which the graphs will display the average.	10	D
Display Small Graphs	Shows the user the top 4 sub-entities and bottom 4 sub-entities by statistic.	8	M
Real-Time Updates	Updates the graphs in real time with incoming performance data.	10	D
Hypothetical Graphs	Allows the user to graph hypothetical VM migrations (removals or additions).	12	F
Affinity & Migration Recommendations	Makes recommendations to the user on which VMs are out of phase.	16	F
Small Graph Stat Updates	Updates which small graphs (of sub-entities) are displayed to the user, based on the current top-used and bottom-used entities.	10	D

3.4 Behaviour Requirements

3.4.1 Use Case View

Use Case: View VM statistics from a specified VM

Actors: System Administrator, VMware Developer

Precondition: Our module has fetched has recorded time-series data.

Flow of Events:

Basic Path:

1. Actor selects data to be visualized.
2. Actor selects the time frame within which he/she would like to observe data
3. Actor observes graph.

Use Case: View Host statistics from a specified host.

Actors: System Administrator, VMware developer

Precondition: Our module has fetched some time-series analyzed data

Flow of Events:

Basic Path:

1. Actor selects data to be visualized.
2. Actor selects the time frame within which he/she would like to observe data
3. Actor observes graph.

Use Case: View Cluster statistics from a specified cluster

Actors: System Administrator, VMware developer

Precondition: Our module has fetched the appropriate time-series analyzed data

Flow of Events:

Basic Path:

1. Actor selects data to be visualized.
2. Actor selects the time frame within which he/she would like to observe data.
3. Actor observes graph.

Use Case: Compare VM statistics between currently held statistics and another source of data

Actors: System Administrator, VMware developer

Precondition: Our module is currently displaying previously selected data

Flow of Events:

Basic Path:

1. Actor selects location of other source's data from a drop down menu.
2. Actor selects the data they want compared with original data.
3. Actor observes comparison graph.

Alternate Path:

1. Actor selects location of other source's data from a drop down menu.
2. Actor selects the data they want compared with original data.
3. Actor decides to instead view other data
4. Actor removes the other source's selected data
5. Actor selects the other data to add
6. Actor observes comparison graph.

Use Case: What if query: Can we add another VM to this host?

Actors: System Administrator, DRS

Precondition: ESX host has several VMs running on it. Actor has a potential VM to add. Actor has specified weight for time and power.

Flow of Events:

Basic Path:

1. Actor queries whether potential VM can be added.
2. Yes response is given.
3. Actor adds potential VM to host.

Alternative Path:

1. Actor queries whether potential VM can be added.
2. No response is given.
3. Actor does not add potential VM to host.

Use Case: What if query: Can we power off a host in a given cluster?

Actors: System Administrator

Precondition: A cluster of hosts are running as one aggregate pool of resources. Actor has specified a weight for time.

Flow of Events:

Basic Path:

1. Actor queries whether a host X can be turned off in the cluster
2. Yes response is given.
3. VMs migrate to other hosts in cluster and X is turned off.

Alternative Path:

1. Actor queries whether a host X can be turned off in the cluster.

2. No response is given.
3. Host X remains on.

Use Case: Process Specific Statistics from ESX host(s) using Time Series

Actors: DRS/Developer

Precondition: ESX hosts are running on a cluster with active VMs for a sufficient amount of time

Flow of Events:

Basic Path:

1. Gather specific statistic using vCenter API calls
2. Use library calls and internal calls to locate trends using Time Series
3. Successfully extract useful and readable data.

Alternate Path:

1. Gather specific statistic using vCenter API calls
2. Use library calls and internal calls to locate trends using Time Series
3. Use other analysis to extract useful and readable data.

Use Case: Predictively turn on hosts

Actors: DRS

Precondition: ESX host(s) are running on a cluster with active VMs for (x) amount of time.

Flow of Events:

Basic Path:

1. DRS inquires on a specific host and whether it can be turned on.
2. vTimeSeries provides suggestion based on previous data.
3. DRS follows through with suggestion and powers host on.

Alternate Path:

1. DRS inquires on a specific host and whether it can be turned on.
2. vTimeSeries provides suggestion based on previous data.
3. DRS observes use decline and ignores suggestion.

Use Case: Predictively turn off hosts

Actors: DRS

Precondition: ESX host(s) are running on a cluster with active VMs for (x) amount of time.

Flow of Events:

Basic Path:

1. DRS inquires on a specific host and whether it can be turned off.
2. vTimeSeries provides suggestion based on previous data.
3. DRS follows through with suggestion and powers host off.

Alternate Path:

1. DRS inquires on a specific host and whether it can be turned off.
2. vTimeSeries provides suggestion based on previous data.
3. DRS observes use surge and ignores suggestion.

4 Other Requirements

Appendix A – Data Dictionary

Appendix B - Group Log

Notes Taken during our first meeting with Ajay 1/23/13

- no low level “kernel hacking”
- understand what a virtual machine consumes
- how to answer questions about a VM farm
- new frontier is to manage VM environment
- UCSB is on VMap program (free licenses)
- ESX as hypervisor (monitor) [Ajay: ESX is all caps]
- ESX only needs a light OS as host operating system
- commands available are VM specific
- many are prefixed with ESX
- PXE boot [Ajay: this is basically booting a machine over network]
- boot a few machines on ESX, then connect to them through management software
- Virtual Center/vSphere
- management software then has hosts added
- provides full summary of host
- create VMs, look at VM console, power on and off
- At this point, cluster VMs, allow different traits/options
- want to run vSphere on an internal machine in same vicinity
- want to collect stats on VMs, each resource has about 200+ stats
- want to answer questions performed on them (data analytics)
- use time series analysis, or something else, up to our discretion
- questions: I want to create 5 VMs on x hosts, can I do so?
- detect trend automatically; detect periodicity
- Example: You can add a VM with the opposite trend to normal, allowing for use
- need to implement time series ourselves, can use standardized libraries
- need to download workloads and run them on VMs
- want to create largest available space
- will also do analysis on whether or not we can shut something down
- vCenter has a specific port to listen on
- Ideally gives some sort of information at the beginning without being prompted.

1/25/13 Meeting

- Discussed literature provided (and authored) by our mentor Ajay.
- Covered DRS algorithm in detail.
- Covered concepts (resource pooling, virtualization)
- Covered studies

Scrum Standup 1/28/2013

Ryan McGinley

- Sent an email to Chandra presenting different servers.

- Don't know where we're keeping the hardware. Ask Chandra in class.
- Ajay "upped the requirements". "Buy 32 gigs."
- Store is able to satisfy the requirements for the project.
- Talk to Chandra about location and pricing

Nick Cross

- Sent email for vMap software, no response.

Vladimir Adam

- Talked to software supervisor "for entire campus", need to email and ask about it
- Only have vSphere, need to find other licenses.
- Charene Strobach - possibly calling tomorrow.
- Find out by Wednesday morning, verify so that we can talk to Ajay about it.

Karanbir Toor

- Ajay responded, meet Wednesday at 9:30 AM.
- Mac Users should install windows on their computers.

Cesar Polanco

- Should we get Windows on a mac or is it not necessary?
- We might need licenses, ask if necessary.
- Given our budgets, this is the best we can do.

Scrum Standup 1/29/2013

Cesar Polanco

- Set up the sprint for the week.
- Will look into the SDK.

Ryan McGinley

- Purchased the server
- Will inform Ajay about this.

Karanbir Toor

- Found other mentor will return on the 7h.
- Will find out about the licences.

Vladimir Adam

- Found out UCSB is not part of vMap.
- Will find if we can renew vMap subscription.

Nicholas Cross

- Found literature on time-series analysis.
- Will look into what development environment is best.

Topics to cover for meeting with mentor:

- Inform him about what we purchased
- Ask about other mentor returning from vacation
- Ask for licences to VMware software
- What next steps should we take this week for a sprint
- Confirm our conclusions about about Ajay's article
- What sort of literature to read for TSA?
- Decided on SDK

1/30/13 meeting with Mentor + standup

Ryan:

- Bought servers to have monitoring software as well as vSphere.
- RAM not limited

Vlad: People are giving them a roundabout to vMap.

- vSphere is available, but the school does not provide more vm software.

Karan:

- Hi Ajay. very clear paper

- Do you want us to follow the same algorithm?
- Anything to read on TSA?

Ajay: Should be good enough for the start; NFS needs more disks.

- Re-arrange disks to provide for the NFS box
- Place Linux on box and use it for NFS since I/O focuses there.
- Talk to any professor doing Virtualization Research and they will have a better idea.
- Somebody in the CS department should know
 - UCSB may have to pay 200 to be part of the program (vMap)
- Apply algorithm to the step, but also apply to longer term questions.
- We want to apply Time-Series Analysis to longer term questions
- Ajay will look something up, but feel free to look for information as well
- Look up a “compression” algorithm to keep stored data similar
- We will want to pull some of the “ESX top” information
- Stats in a virtualized environment are different from hardware (flat linux) stats
- Look into getting an IDE (Maven and Ent?) similar to Eclipse
- trying to get some code for us (boiler plate) in order to get started
- Multiple meetings should be fine once the other mentor returns.
- Emails should be more frequently responded to as well.

Scrum Standup 1/30/2013

Ryan McGinley

- Looked into increasing RAM on servers. Will cost an extra \$300 per server.
- Will decide on OS for NFS.

Karanbir Toor

- Read more literature on time-series analysis. Paper on time-series applied to VM migration.
- Will set up Git repository.

Cesar Polanco

- Sent a reminder email for software licences.
- Will continue to bug mentor about software licences.

Vladimir Adam

- Asked mentor if we can get a test bench or some other way to simulate load.
- Will pursue more information about the simulated workload.

Nicholas Cross

- Investigated into getting licenses from the school.
- Will continue to look for the licenses.

Scrum Standup 1/31/2013

Ryan McGinley

- Correspondence with Tim in order to verify the useability of the servers.
- Will email mentor to verify that the hardware will suffice.

Karanbir Toor

- Decided on purchasing a physical board instead of maintaining a virtual one.
- Will look to find better price.

Vladimir Adam

- Continued to look into finding more about the test benches.
- Will continue to find out more about testing materials.

Cesar Polanco

- Heard back about the lack of vMap.
- Have to follow up on person who will give us licenses.

Scrum Meeting 2/1/2013

Karanbir Toor

- Purchased cork board for sprint planning. Everyone owes me \$10.

- Will find time to set up Github.

Cesar Polanco

- Downloaded Windows 2012 server.
- Will investigate the different types of IDEs that would be beneficial to use.

Vladimir Adam

- Burned the Windows server cd.
- Will look into what else we need in terms of software.

Nicholas Cross

- Still have not gotten licenses.
- Will look into getting 30 day trial licenses since regular licenses seem to be unavailable at this time.

Scrum Meeting 2/5/2013

Karanbir Toor

- Set up Github.
- Will begin Burndown chart.

Ryan McGinley

- Set up servers.
- Will set up NFS/FreeNAS

Cesar Polanco

- Set up servers.
- Will look into test cases.

Vladimir Adam

- Set up servers.
- Will look into setting up VMWare stuff.

Nicholas Cross

- Set up servers.
- Began looking through test cases

Wed, Feb 6, 2012 Meeting with Mentor + standup

- OS on Servers windows on vCenter, Linux NFS, 3 ESX server
- Racks?
- NFS will be mounted on ESX host
- Free NFS widely supported easy
- Linux NFS version 3 protocol okay?
- Java template source code still waiting on it
- Feel free to send reminder email
- Mac OS is fine
- 30 day free trial software vCenter and ESXi
- Renew subscription must be renewed so Steve may be trying to get in contact with someone
- Google capstone we will share reading resources
- Download VMware SDK
- 1 gigabit switch
- makes some windows and Linux VMs
- iometer, benchmark, many software options to create benchmark
- Then clone VM's
- Sprint would be to set up all the software on the Servers
- Sprint after that would be to establish VM's
- independent testing of NFS with normal Linux computer
- Next mentor back at the end of this week
- Find software trial here
- <http://www.vmware.com/products/vcenter-server/overview.html>

Scrum Meeting 2/7/2013

Ryan McGinley

- Installed Windows 2012 Server Edition on vCenter machine.
- Will install NFS.

Cesar Polanco

- Set up ESX hosts.
- Will work on SRS

Karanbir Toor

- Purchased wood to build a rack for the servers.
- Will work on SRS.

Nicholas Cross

- Set up ESX hosts.
- Will work on SRS.

Vladimir Adam

- Set up ESX hosts.
- Will work on SRS.

Scrum Meeting 2/8/2013

Vladimir Adam

- Downloaded licensed software.
- Will burn DVDs with VMWare software on them.

Ryan McGinley

- Set up NFS system.
- Will configure systems.

Karanbir Toor

- Worked on SRS.
- Will work on feedback to finalize SRS.

Nicholas Cross

- Worked on SRS.
- Will work on feedback to finalize SRS.

Cesar Polanco

- Worked on SRS.
- Will wait on feedback from mentor to finalize SRS.