**Senior Design - Design Document**

**Objectives:**

 This project seeks to provide a server that can exceed the speeds of normal hard disc servers while maintaining a cost lower than that of a solid state drive. In general, hard disc drives can be bought at a low cost, but have poor performance. They are becoming outdated and being phased out because of their more archaic technology and design. However, the alternatives are much more expensive. High capacity solid state drives are much faster and more sophisticated, but also much more expensive. This project seeks to take the advantages of both of these options and put them together. By using a much smaller solid state drive that is used sparingly as a cache for important files on the hard disc. By using the solid state drive intelligently, performance can be greatly boosted while keeping costs low.

**Users:**

**Server Client –** Any client who connects to this server and uses some sort of storage on the server will indirectly be using this project.

**Server Administrator –** This is the person who is in charge of installing this project into their server and making it available to their client.

**Server Client:**

**Use Case – Interacting with the Server’s Persistent Storage**

 Any time a user interacts with the persistent storage of a server that is running this project, they are indirectly using the custom driver that is handling the backend management of their data. By doing any of the following operations, they are using this project:

* Reading a file
* Writing to a file
* Creating a new file
* Creating an account
* Updating account information

Despite the numerous places where this project is used, the user experience is totally unchanged, except for faster performance. The user does all of these operations as if nothing were different from before. The complexity of the backend storage devices and caching algorithms is completely abstracted away from the average user. The client to the server should not be able to tell that they are using the cached system as opposed to just a normal server.

In particular, the server running this project should be using Xen, a paravirtualization tool. Each client should be interacting with a virtual machine, which already abstracts away parts of the hardware. Because of this, each virtual machine doesn’t even handle its own disc requests, it merely passes along all requests to the privileged virtual machine, Domain 0.

**Main Functional Components: Xen**

**Server Administrator:**

**Use Case – Installation**

 The one use case for the server administrator comes when the system itself needs to be installed. There are some steps that must be taken in order for this system to run at optimal efficiency, or even work at all.

1. Install a backing hard drive and at least one solid state drive.

This project uses a large hard disc drive to store all of the data being used by the system, then certain files can be copied onto the solid state drive for faster performance. Because of this, at least one hard drive must be installed, then a solid state drive should be installed at the same time. Multiple solid state drives can be used for different levels of caching, and the larger the size, the better performance for the virtual machines, but these decisions can be limited by cost.

1. Install and run Xen.

Xen is a previously discussed paravirtualization tool, meaning that a number of virtual machines run simultaneously while sharing a single pool of hardware resources. The Xen hypervisor distributes these resources passively among all virtual machines, but all active input/output operations are passed to a single, privileged virtual machine. This machine, Domain 0, contains all of the necessary drivers and protocols for interacting with hardware devices such as hard drives. From here, virtual machines can be configured to interact with the hardware in a number of ways, but the method used by this project is the blktap library.

1. Install custom blktap driver.

The caching algorithms and ability to use multiple hard drives is all built into a custom driver that can be included in the blktap Xen code. Once the code has been downloaded and included in the Xen code, it is simple to install and configure new virtual machines to use the blktap driver. Once this is done, the driver will accept I/O requests from other virtual machines, and then decide where the data is located and whether or not it should be cached.

 Once these steps have been performed, the driver should run in the background and should not require any active interaction. Server clients can continue to use the virtual machines, which will work though Xen, and then the driver will perform the necessary algorithms and storage to keep track of all the data on both the hard disc and cache drives.

**Main Functional Components: Xen, Blktap Driver**

**Relationships of Functional Components:**



This diagram shows how the modified operating systems of the guest virtual machines send requests to the privileged Domain 0, which has the blktap driver, and interacts with the hardware. Through this, it interacts with both the hard drive and solid state drive.

**Requirements:**

**Xen –** There are no major modifications to Xen itself, so the main requirement is for it to just do what it’s supposed to do. It must successfully create and manage virtual machines, and pass input/output requests to Domain 0. At this point, it must use the blktap driver correctly and return the correct data to the proper virtual machines.

**Blktap Driver**

**Functional Requirements –** The blktap driver does most of the work and therefore has a number of functional requirements. It must keep track of what data is currently being stored on the cache drives, and where it is located. It must decide what data should be added to the cache drive. It must be able to access the blocks on both the solid state and hard disc drives.

**Nonfunctional Requirements –** In addition to the things that the driver needs to do functionally, there are additional requirements that must be taken into account. The cost of adding the solid state drives cannot be excessive, and the performance increase over a standard hard disc drive must be significant. In addition, the installation process of Xen and the driver must be simple and easy to do.