**Functional Specifications**

**Hard Drive Caches for Virtual Machines**

**Goals:**

This project seeks to implement a system that utilizes multiple hard drives in order to optimize both cost and performance. With the costs of larger capacity solid state drives still being exceedingly high, a cost efficient solution is necessary. That solution is in this project, which seeks to use smaller capacity solid state drives in order to reduce cost, but still provide high speed and performance. Therefore, the main goal of this project is to provide a system that has a lower cost than a system that uses a solid state drive as well as better performance than a standard hard drive.

One of the smaller goals that is part of this is keeping the final solution cost efficient. In order to do this, the system needs to operate on as small of a solid state drive as possible. The cost of the solid state drives will increase exponentially with higher capacity. If they are required to be large and still expensive, then the positive benefits of this project will be not be worth the unavoidable performance drop from a pure solid state drive.

Similarly, this project needs to provide a significant performance increase over a simple hard drive. The main benefit of a standard hard drive is that they are cheap. Since this project requires adding on at least some extra costs to a standard hard drive, the speed up needs to be significant. While the costs need to be low in order to provide a better option than pure solid state drives, the performance still needs to be higher in order to be better than pure hard drives.

The main internal goal of this project is to provide an algorithmic solution that makes it worth using. On the previously discussed spectrum of solid state and hard drives, solid state drives have better performance and hard drives have better cost. In this project, which, when over-simplified, is just a middle ground between the two, the algorithmic accomplishments are going to be what makes it the best option. If this project simply finds a half way point in both cost and performance between the two options, than there is no real benefit. If the performance of this system can almost match the performance of a solid state drive, with minimal additional costs, then it could have enormous potential for a number of applications.

**Non-Goals:**

This project does not seek to improve performance and cost for personal computers or laptops. This project operates on a larger scale, with servers that host multiple virtual machines. By monitoring and allocating solid state space to these virtual machines, the performance of the whole system can be improved. With personal computers with a single user, there is no need for this. Hard drive access is generally random and not highly repetitive, so there would be little performance increase. In addition to this, solid state drives for personal use are relatively cheap and becoming more common, limiting the benefits of this system. The ideal solution for personal use has already been offered by some companies like Apple, which simply puts the operating system on a solid state drive and everything else on a hard drive. For most personal computers, this is sufficient.

Another goal that is not within the scope of this project is defining or abstracting away the number of hard drives. It will assume the same number of hard and solid states drives is used for any system implementing the algorithms, and require the user to install these drives on their own. It will not try and detect new hardware and adapt or change the way the algorithm works. This would require a lot more algorithmic complexity for not much gain. If the user of this system simply has the same size hard drives or ratio of solid state space to hard drive space, then it is simply a matter of them making sure that all of them are connected to their server.

**User Experience:**

Let’s say that you’re in charge of the next big start-up. You have your game changing product, you’ve designed your website. You have a plan and you’re ready to get started and break into the market. You probably have capital and investments, but you need to be using them wisely. Now let’s say you’re setting up an office and technology infrastructure to support and host your website. This is going to be the backbone of your company, the user experience is going to be all about their interactions with your website and/or mobile application. Your servers are running multiple virtual machines in order to handle all the requests and transactions that are going to drive your sales. You are presented with three options.

Your first option is to spend thousands and thousands of dollars and use a massive solid state drive to back up your website and everything hosted there. It’s a large expense, perhaps large enough to put you in serious debt before your business even gets started, but it will provide your users with a faster experience. However, you may have crippled yourself with the unnecessary costs of multiple large solid state drives to handle all the requests. Not only doesn’t that, but the kind of traffic going through your servers doesn’t fully take advantage of these drives. If further expansions are necessary, the costs of installing larger drives will increase exponentially.

Your second option is to get a single massive hard drive for your server. Hard drives are an older and more reliable technology, cheap and plentiful, but it may have adverse effects on your sales. When it comes to your website, user experience is paramount, so speed and efficiency is important. If someone is trying to create an account or make a purchase and there’s lag or a slow response time from the server, they might be dissatisfied and not want to use it in the future. This is a risk you take with a hard drive, especially with larger scale systems.

Your third option is to implement our system. In order to do this, you can buy a cheaper hard drive to hold all of your content initially, as well as a few smaller solid state drives, and then run Xen on your paravirtualized servers. With Xen and our custom driver both installed, your servers can now take optimal advantage of the smaller solid state drives that cost a fraction of the price and still provide your users with a quick and responsive experience. By doing this, you have avoided both the problems of a high initial cost to overcome and the risk of losing future business.

**Main Components:**

**Hard Drive:**

In order to ensure that no data is lost, a backing hard drive is necessary. This hard drive will have enough capacity to hold all the content being hosted on the server. Once the hard drive is installed, it becomes an important part of the algorithmic process, as files can be copied from the hard drive onto the faster solid state drives.

**Solid State Drives:**

The solid state drives in the system are the driving factor for performance. Using a couple of smaller capacity solid state drives will give most of the performance benefits as if a full solid state drive was being used. The use of smaller capacity solid state drives is crucial, as they can increase in price exponentially. Here are some statistics, all based on Intel prices:

* 800 GB Solid State Drive - $650
* 1200 GB Solid State Drive - $1080
* 2000 GB Solid State Drive - $4700

The inclusion of a few smaller capacity solid state drives will not greatly increase the cost of the system, however if you try and use one big drive, it will cost you greatly.

**Xen:**

Xen is the virtualization software that our system is built for. Virtualization is the process of taking a number of virtual computers and making them run on a single set of physical hardware. In the case of this project, there might be a number of virtual computers handling requests and accessing files on the server, but there is only a single processor and hard drive backing these machines. The virtual machine manager, in this case Xen, is in charge of distributing the single pool of resources amongst the many virtual machines. Xen does all of the work of scheduling the virtual machines and making sure that they are capable of accessing the backing hard drive.

One aspect of Xen that makes it well suited for this project is that it uses paravirtualization as opposed to full virtualization. Because of this, each of the virtual machines is slightly modified to not make all of the operating system processes go through the virtual machine manager. Background and passive processes such as memory management and scheduling are still performed by the hypervisor, but all input and output operations, like reading and writing from a hard drive, are sent through a single virtual machine. This special virtual machine, Domain 0, has all of the drivers and libraries necessary for interacting with the underlying hard drives and solid state drives.

**Custom Driver:**

The algorithmic complexity and implementation lies within a custom driver that is installed within Xen. As opposed to just accessing the hard drive directly through the standard, pre-installed drivers, users can use our driver instead. The driver keeps track of both the hard and solid state drives and decides what files should be moved for faster access and which virtual machines can make the best use of the limited solid state space. This driver can be easily added to Xen and virtual machines can be configured to use this driver and improve their performance.