TerraServer Bricks – A High Availability Cluster Alternative

Tom Barclay

Jim Gray

Wyman Chong

October 2004

Technical Report

MSR-TR-2004-107

Microsoft Research
Advanced Technology Division
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Table Of Contents

TerraServer Bricks – A High Availability Cluster Alternative

Tom Barclay. Jim Gray, Wyman Chong
{TBarclay, Gray, Wymanc }@microsoft.com

Microsoft Research, 455 Market St., Suite 1690, San Francisco, CA 94105
http://research.microsoft.com/barc

Abstract

Microsoft® TerraServer stores aerial, satellite, and topographic images of the earth in a SQL database available via the Internet since June 1998. It is a popular online atlas, combining twenty-two terabytes of image data from the United States Geological Survey (USGS). Initially the system demonstrated the scalability of PC hardware and software – Windows and SQL Server – on a single, mainframe-class processor [Barclay98]. Later, we focused on high availability by migrating to an active/passive cluster connected to an 18 terabyte Storage Area Network (SAN) provided by Compaq Computer Corporation [Barclay04]. In November 2003, we replaced the SAN cluster with a duplexed set of "white-box" PCs containing arrays of large, low-cost, Serial ATA disks which we dub *TerraServer Bricks*. Our goal is to operate the popular TerraServer web site with the same or higher availability than the TerraServer SAN at a fraction of the system and operations cost. This paper describes the hardware and software components of the TerraServer Bricks and our experience in configuring and operating this environment for the first year.

1 Overview

The TerraServer is designed to be accessed by thousands of simultaneous users using Internet protocols via standard web browsers. During a typical day, TerraServer is visited by 50k to 80k unique visitors who access 1.2 to 2.7 million web pages containing 2 to 12 200x200 pixel image "tiles". In addition 2 million SOAP web service requests are made to the Terraservice on a busy day. This is 10x increase since September 2003.

Users access the data through one of three applications – 1) a HTML based web application, 2) a programmatic SOAP/XML based web service, or 3) an image based web application commonly referred to as a "web map server". The applications enable users to query the TerraServer image repository in a number of ways. Results are imagery *tiles* presented as compressed Jpeg or GIF file; or meta-data presented as an HTML or SOAP/XML document.

All TerraServer data is managed by Microsoft SQL Server 2000. The USGS delivers meta and raster data as large graphics files in TIFF or proprietary image formats. The TerraServer load application shreds the input imagery into small, 200x200 pixel *tiles* and stores each tile in a *blob*, (Binary Large Object, a.k.a. image data-type) column in a SQL Server database. TerraServer currently contains more than 406 million tiles extracted from the high-resolution raster data-sets provided by the US Geological Survey. We call each of these different data products a *theme*:

DOQQ (Digital Ortho Quarter Quadrangles): 280,000 USGS aerial images at one-meter resolution – a total of 18 terabytes of uncompressed imagery has been received to date,

DRG (Digital Raster Graphics): 60,000 USGS topographic maps – a total of 2 terabytes of compressed imagery has been received to date,

Urban Area: 61,000 USGS high-resolution (.3 meter per pixel), natural-color imagery files – a total of 4 terabytes of uncompressed imagery has been received to date (45 cities).

The total imagery received to date is 24 terabytes. There is substantial redundancy and overlap within the data files, thus when compressed, the imagery and necessary meta-data consumes 3.8 terabytes within the SQL database(s).

TerraServer data easily partitions in two dimensions – 1) meta-data can be separated from imagery data, and 2) imagery data can be partitioned into geographic regions.

Meta-data can be physically separated from imagery data due to the architecture of HTML documents. An HTML web page containing imagery does not directly encapsulate image file into the HTML document. Instead, an HTML document describes where an image is to appear relative to the text and other images, and how to retrieve the image, i.e. the image's URL. Thus, TerraServer applications that generate HTML or SOAP/XML meta-data document need only the relatively small metadata and can be partitioned from the web pages and web service methods that produce imagery data.

Imagery data can be further partitioned by image *scene*. A *scene* is all the imagery from a single *theme* that forms a seamless-mosaic – for our data it is a single UTM zone (see Figure 1). Each scene is a single coordinate reference system. The web application presents imagery a single scene at a time on a web page. The TerraServer application requires the imagery data for a single scene be contained in only one SQL Server database. A single SQL Server database can contain one or more scenes.

All TerraServer's image data is in the UTM NAD83 projection system. The UTM projection method divides the earth into sixty, six-degree wide regions known as zones as diagramed in Figure 1, UTM Zone Boundaries.

The DOQQ theme overlaps zones 5, 6, and 10 thru 20. The DRG theme overlaps zones 1 thru 19, and 60. The Urban Area theme overlap zones 6, and 10 thru 19. Each of these zones can be a separate partition, so in the extreme, the TerraServer easily partitions into 45 physical SQL Server databases. This gives fine grain units of management and failover.

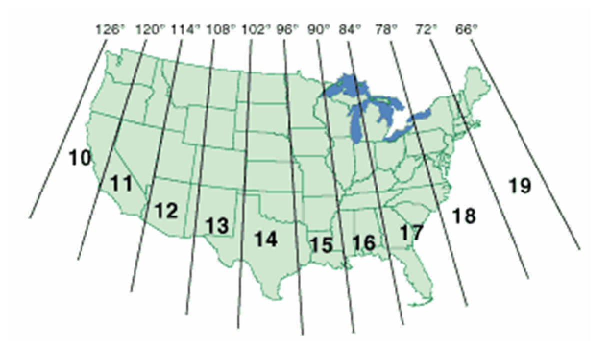

Figure 1: UTM Zone Boundaries

When launched in June 1998, all TerraServer data was stored on a single server and SQL Server database [Barclay98]. In September 2000 we eliminated this single point of failure by migrating to an active/passive four-node Windows 2000 Data Center Edition SAN cluster [Barclay04]. The design partitioned the TerraServer data into three[1] databases:

1. DRG theme's imagery data and all imagery themes' meta data.
2. DOQQ theme's imagery data for odd numbered UTM zones (5, 11, 13, 15, 17, and 19)
3. DOQQ theme's imagery data for even numbered UTM zones (6, 10, 12, 14, 16, and 18)

The Windows 2000 Data Center Edition cluster ran on large Compaq 8-way processors connected to an eighteen terabyte Storage Area Network (SAN) provided by Compaq StorageWorks. Except for a couple of operational mistakes, this Compaq/Windows Data Center Cluster was available 99.99% of the time [Barclay04]. However, the cluster configuration had a number of drawbacks:

Expensive – the configuration cost $1.9m[2] in September 2000. Prices have dropped substantially since but are still relatively high compared to other hardware alternatives. Hosting charges for the system were also high because it occupied 6 racks in the data center.

Complex – to achieve the performance and high availability goals, the equipment was sophisticated and non-trivial to configure and support. Maintenance operations were complicated and error-prone as we discovered during a SAN software upgrade that shut the system down for 17 hours [Barclay04].

Long Failover Events – each TerraServer database partition was six terabytes of disk. The minimum time to failover a database resource (108 disk drives) to another node was 30 seconds and averaged 45 seconds.

Required Tape – The online system had single copy of the data. A tape library was required to backup and store the data offsite in case the system was physically destroyed or, in the more likely scenario, in case of an operations or software fault. The SAN was so expensive; we couldn't afford an online copy of the data.

In many enterprise applications, a SAN's high cost and complexity can be tolerated because of the ROI the application provides to the organization. However, most internet applications have razor thin profit margins. It is difficult if not impossible to host a profitable internet business on SAN hardware. Yahoo and Google give good examples of this. They buy very low-cost hardware configured redundantly to achieve high availability. They do not depend on system software or hardware components to handle failure cases. Instead, they program "around failures" at the application or in the "middle-ware" that their staff implements. As a result, they have very high application availability implemented and deployed at a very low cost.

In contrast MSN and many Microsoft customers have traditionally deployed SQL Server, and Microsoft Clustering applications that expect the underlying hardware and system software to handle failure conditions transparently to the application. This is changing, MSN search has a brick design, and MSN Hotmail is making the transition from expensive backend SAN servers to commodity servers similar in design to TerraServer Bricks.

The TerraServer Brick architecture described in this paper is an experiment to build a low-cost hardware and software environment, similar in approach to a Yahoo, Google, MSsearch and the new Hotmail architecture. The goals of the design were simply the converse of the drawbacks of the TerraServer Cluster and SAN, specifically:

Inexpensive – the configuration should cost one-tenth of the TerraServer SAN-Cluster price and the hosting charges should be at least three times less.

Simple – the components should be commodity equipment that requires no special training or skills to maintain beyond those of a competent Windows administrator.

[1] When we were operating the four-node cluster, the TerraServer application offered only two data themes – DOQQ and DRG.

[2] The initial system cost $1.4m in September 2000, it was expanded by 6TB of storage and ADIC Scalar 1000 tape library in 2001 bringing the total hardware cost to $1.9m.

Brief Failover Events – the application should sense and failover to another resource within seconds rather than minutes. The new design should exceed the availability delivered by the TerraServer SAN-Cluster and SAN deployed from October 2000 through October 2003 [Barclay04].

No Tape – Magnetic tape is expensive, needs special software, and recovery times are measured in hours or days. Like paper tape, punch cards, and floppy disks, we believe it is time to retire tape from modern configurations and replace it with, mirrored systems and geoplexing.

To meet these requirements, we designed two types of small, rack mounted servers described in Table 1: (1) web servers and (2) storage bricks. Each server was assembled by a local manufacturer, Silicon Mechanics [SilMech], from low-cost, readily available components. To simplify installation, operation, and maintenance, each system was identical depending on its purpose.

Table 1: Key properties of the two kinds of bricks.

	Web Brick	**Storage Brick**
Number	5 x 1U	7 x 3U
cpus	2 Xeon 2.4Ghz Hyper-Threaded	2 Xeon 2.4Ghz Hyper-Threaded
RAM	2 GB	4 GB
controllers	Built in	3ware 8500-8
disks	2 80 GB SATA	16 WD SATA 250 GB 5200 RPM
network	Dual GbpsE	Dual GbpsE
OS	Windows2003	Windows2003
S/W	IIS 6 , .NET V1.1	SQL Server 2000
Price	$2,100	$10,300

We deployed the configuration next to the TerraServer SAN-Cluster in November 2003. The TerraServer end user applications were modified to handle failure cases and transfer processing to a redundant node. We operated them side-by-side for a month and then retired the TerraServer SAN-Cluster. We have been running the TerraServer web applications exclusively on the TerraServer Brick configuration since mid-November 2003. This paper describes the TerraServer Brick Architecture and our experience operating it for the last year.

2 TerraServer Brick Architecture

The TerraServer web site is composed of:

- a redundant farm of web bricks,
- a mirrored array of storage bricks,
- a redundant LAN linking the web and storage bricks,
- a remote IP keyboard, video, mouse (KVM) switch,
- and, remote IP power distribution units (PDU).

Each web brick is identical. It has storage capacity to host the TerraServer web application, web service, web map server components and the disk space for a month of web log files. The TerraServer web applications are written in C# and depend on IIS 6 and ASP.NET that are included with Windows Server 2003.

Web bricks are inexpensive, so they are over-provisioned. TerraServer applications can comfortably operate under normal load with 50% of the web servers functioning.

Each of the seven Storage Bricks has identical hardware and software – but the data is partitioned and replicated among them. They run SQL Server 2000 Standard Edition. The web applications make requests to T-SQL stored procedures hosted in SQL Server 2000 databases supporting the TerraServer database schema and data.

We have the choice of configuring the disks using no redundancy (JBOD), using hardware or controller-based redundancy, or software based redundancy (RAID-1 or RAID-5). We deployed the majority of the Storage Bricks with controller-based RAID-1 (mirroring) redundancy. To see the differences, one Storage Brick has software-based RAID-1, and one has a mix of controller-based RAID-1 volumes and JBOD. Each redundancy option offers different pluses and minuses discussed later in this paper.

A sixteen drive Storage Brick is typically configured with eight RAID-1 mirrored volumes. The first volume is partitioned into two logical drives, C: and D:, The C: drive contains the operating system software, SQL Server 2000 software, and TerraServer application files. The other seven volumes are dedicated to TerraServer SQL Server database data.

These disk volumes can store up to 232GB of SQL Server data files. A single seven volume Storage Brick can store 1,624GB or 1.5TB of data. As of October 2004, the TerraServer databases consume 3.8 TB of disk space.

2.1 Bunches of Bricks

The image data (3.8 terabytes) will not fit on a single brick which has 1.5 TB of RAID-1 storage available to SQL Server. Hence the image data must be partitioned across multiple storage bricks. In contrast, the metadata is only 25 GB and so can be replicated at each server. Section 2.2, describes in detail how TerraServer imagery data is partitioned across multiple servers. This section discusses how a set of Storage Bricks are grouped and presented to the application.

Storage Bricks are organized as an array of shared-nothing partitioned databases – RAPS (reliable array of partitioned servers) in the terminology of [Devlin] here called a *bunch*. Bunches do not use shared-disks or any formal clustering (pack) software such as Microsoft Cluster Services (MSCS) to form a group of bricks. Each Storage Brick runs an independent copy of the Windows 2003 Server operating system and SQL Server database software – the least sophisticated "standard" editions of both offerings.

To avoid confusion, we call a set of Storage Bricks that contain a complete copy of the TerraServer data a *Bunch of Bricks* or simply a *Bunch.* Others might call them *clusters* – but cluster is a loaded term connoting a formal entity running specialized software such as MSCS. Bunches have simple system software and have application-level fault-tolerance and application-level system mirroring.

For availability and data preservation, we clone a bunch's data (array of data partitions) on a second or third bunch. Minimally two bunches are deployed to have a redundant set of data. But additional redundant bunches can be deployed depending on performance demands and data preservation paranoia.

Figure 2 depicts how Storage Bricks and Bunches scale. Storage Bricks can be added to each Bunch at any time. And an additional Bunch can be added to the set of Bunches at any time too.

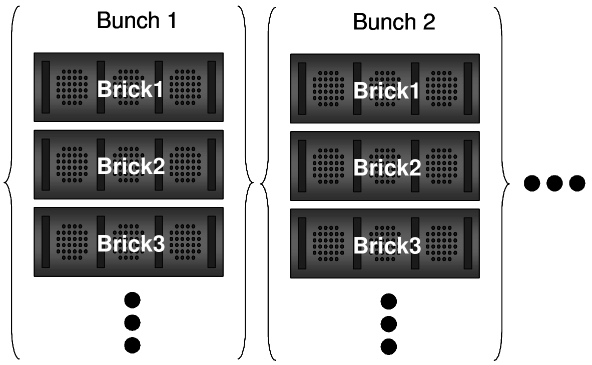

Figure 2: Scaling Bricks and Bunches

TerraServer Brick Architecture lacks traditional tape or disk backups. To provide data protection we require that at least two copies each TerraServer SQL Server database be available at most times. Should one disk volume have a double disk failure, then there is at least one additional copy available to the application.

This rule can easily be broken when there are only two bunches in a single configuration. If a volume is lost on one of the Storage Bricks, then the application will continue to operate correctly accessing the data on the *clone* Storage Brick in the other Bunch. But now there is only one operational copy of those databases. What if that volume on the surviving clone also failed?

The mathematical probability of data loss is less than once in a million years. However, Murphy's Law suggests that it will happen to us in the first few months.

We extended the Bunch and Clone architecture with the standard *pair-and-spare 2N+1* redundancy strategy to minimize the exposure of having only one copy of a set of databases should a volume fail. Figure 3 depicts a two Bunch configuration that includes a spare *Backup* Storage Brick. This brick has the same hardware and system software as a Storage Brick. The backup brick's disk organization is different recognizing that it has a different purpose.

The Backup Brick does not require mirrored disks. Instead, it can be configured as "just a bunch of disks" (JBOD), as two-or-more disk multi-volume set, or as multi-disk stripe set.

When a volume fails on a Storage Brick, the SQL Server databases on the surviving clone are backed up to one of the volumes on the Backup Brick. SQL Server 2000 supports an on-line backup capability that operates while users are reading and writing data to the database being backed up. We refer to this as a *just-in-time-backup* that is performed anytime a database volume fails.

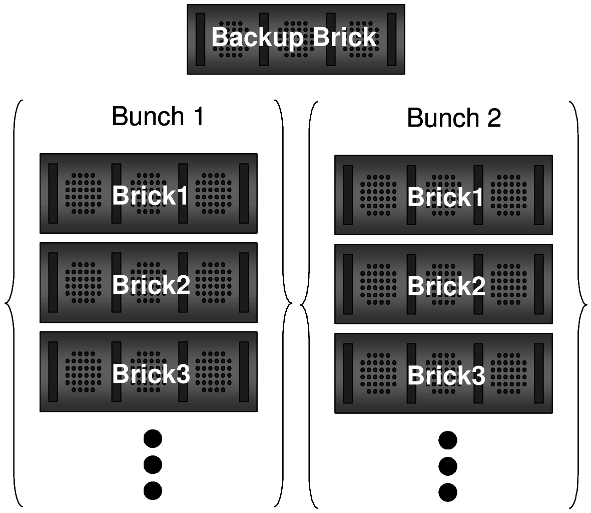

Figure 3: Two Bunches with a Backup Brick

Though not drawn in the diagram, more than one Backup Brick can be deployed in a configuration. Also, Backup Bricks can be deployed in configurations that have more than two bunches.

A Backup Brick can perform other functions. We use the Backup Brick in the data-loading process to update the on-line storage bricks.

A Backup Brick can also be pressed into service as a regular Storage Brick. This can be attractive should a storage brick fail entirely to the point of needing to be replaced. The data from the surviving node can be migrated to the Backup Brick, and the application re-directed to accessing the Backup Brick instead of the failed Storage Brick.

2.2 Partitioning Strategy

There are two goals in determining how to partition TerraServer data:

1. Minimize the impact to the application should a disk volume fail and data is lost.
2. Balance the I/O load across all storage bricks in a bunch.

2.2.1 *Minimizing Impact of Data Loss*

We wanted to minimize data inaccessibility should any component in a storage brick fail. The data partitioning approach we employed is designed to reduce the amount of data that is inaccessible due to a failure of any single component within a storage brick. Since we have eliminated tape as backup strategy, we also needed to be concerned about data loss due to multiple component failures.

Most non-disk hardware failures effectively cause all the data on storage brick to be inaccessible. However these failures usually do not involve any data loss. These failures include network card, memory, RAID controller, or processor failure. Our architecture depends on a second Storage Brick containing an exact copy of the data on the failed brick to serve data to the TerraServer applications running on the Web Bricks.

The disks in the Storage Bricks are configured using RAID-1 mirroring. We always mirror two 250gB disks into one logical volume. We do not stripe data across volumes (RAID-0 or RAID-10). A single disk failure in a mirrored volume would be handled transparently by the 3Ware controller or operating system RAID software and data on the effected volume would continue to be accessible. If both disks in a single mirrored volume fail at the same time, then the data on the failed volume is inaccessible and databases using that volume are lost.

To minimize the granularity of failure, we create SQL Server databases that contain the smallest amount of data possible. The ideal situation is for a single imagery database to fit on a single RAID-1 volume and each RAID-1 volume to contain only one imagery database. If a RAID-1 volume is lost, then only one SQL database is impacted. Unfortunately, the nature of the TerraServer application does not allow the data to be conveniently chopped into units the size of single RAID-1 volume.

The TerraServer application requires that all the imagery data for a single scene of a single data theme be stored together in a single database. A *scene* is defined to be one logical mosaic of tiles. The geo-spatial projection of the data theme dictates the scope of a single scene. The current data themes are in the UTM projection where a scene is equivalent to a UTM zone.

Figure 4 depicts how the SQL Server databases for each data theme and their sizes per each UTM zone. The three databases in the top left contain data from the UTM zones that cover Alaska, Hawaii, and Puerto Rico. Because these are smaller areas containing little data, we lump the data from multiple zones into a single SQL Server database. Thus, USGS DOQQ data for zones 5, 6, and 20 are in a single database. USGS DRG data for zones 0 through 9 are in a single database.

The numbers in each database icon identify the total size consumed for that partition including tabular data, imagery data, indices, and spare space. TerraServer SQL databases are 92% full, and many are over 95%. None of the databases fit snugly within a 232GB disk volume. Many DOQQ databases require two disk volumes. While two or three TOPO (DRG) databases can fit on a single disk volume. This situation changes with the newer 400GB disk drives.

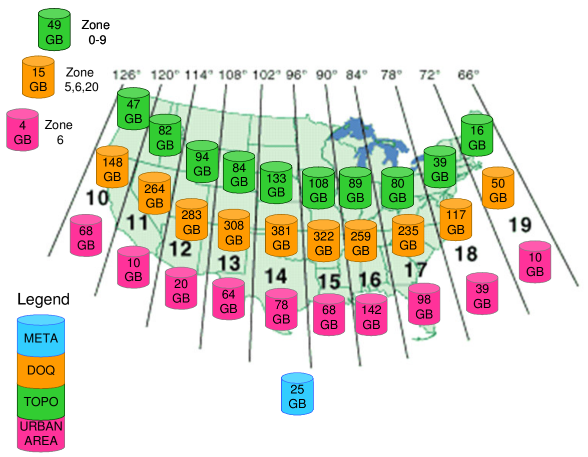

Figure 4: TerraServer Database Partitions

Our strategy in placing databases is to use the minimum number of disk volumes for each database. We place the largest databases first and leave 15% available on the last volume allocated. Once the large databases are placed, we repeat the process for the smaller databases. Again, we try to avoid completely filling a disk in order to allow for database growth.

In November 2003, a single copy of the TerraServer meta and imagery data was 333[3] million rows consuming 3.0 TB of the 3.3 TB of space assigned to the SQL Server databases (90% full). As of September 2004, the databases have grown by 66 million rows to 406 million and 700 GB to a total of 3.5 TB of data stored in SQL Server databases totaling 3.8 TB (90% full).

2.2.2 *Balancing I/O Load Across a Bunch*

The current size of the TerraServer data requires three storage bricks in each bunch. We arrange the databases between each brick in the bunch so that the I/O load is spread evenly over all volumes of all Storage Bricks.

As we learned from the TerraServer initial launch [Barclay98], it is possible for user request rate to far exceed SQL Server's ability to return results. When this occurs, it appears that the TerraServer web application is hung – web requests time out and the users see many red-x symbols where images should appear.

It is difficult to predict the data users will be interested in viewing. Generally, users tend to view imagery near where they live. Many users access TerraServer for professional reasons, thus we can assume that 8am to 5pm

[3] Actually the number of rows is double the 333 million. There are 333 million imagery rows, and 333 million meta-data rows.

in their time zone is their most active time. Monitoring the network traffic supports this assumption. The peak network traffic occurs around noon, and remains within 80% of the peak for approximately 12 hours per day. The daily low is 25% of the peak. TerraServer off-peak load is significant, over 100 requests per second, but the system is configured to handle peak loads.

Occasionally, news stories or internet sites talk about or link to TerraServer in a prominent way. These events can triple daily traffic over a normal busy day.[4] These peak usage days are primary concern for I/O load balancing.

The tendency of users to view data where they live or associated with some event helps locality and caching. There is a wave of traffic across the country during the course of a day where the majority of the data accessed is on the East Coast, then the Eastern Midwest, then Western Midwest, then Mountain Time zone, and finally the West Coast.

SQL Server's data caching algorithms significantly reduce the I/O rate when, for example, thousands of New Yorkers are looking at data fetched from disk once and read from the cache thereafter. We help the cache algorithm by giving it more memory to work with. To do this, we place TerraServer databases such that all bricks contain some data for each time zone. Generally, two UTM zones cover a single time zone. So for each data theme, we place one UTM zone from a time zone on one Storage Brick, and the second UTM zone from the same time zone on a second Storage Brick. We continue this round-robin until all the data is placed across all the bricks.

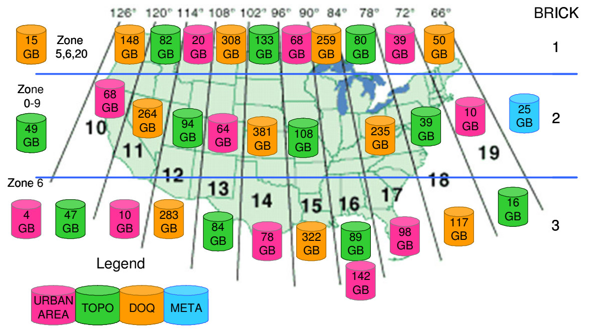

Figure 5: Partition Placement Per Storage Brick

Figure 5 shows the placement of TerraServer databases on each Storage Brick. The top row of databases is placed on Storage Brick 1. The middle row of databases is placed on Storage Brick 2. The last row of databases is placed on Storage Brick 3.

The placement of two Urban Area databases, Zone 16 and Zone 17, violated these placement rules for two reasons. First, Urban Area data was added after the initial design was laid out. We do not know the Urban Area data volume for each UTM zone. Second, Storage Brick 2 hosts the 25GB Metadata database. The Metadata database is the busiest of all the TerraServer databases; so we favor filling the other Storage Bricks first.

2.3 Resource Failover

In the TerraServer Brick Architecture, the TerraServer applications are responsible for detecting and handling resource failures. The TerraServer applications are:

1. The HTML web application (http://terraserver-usa.com)
2. The SOAP/XML web services (http://terraservice.net/terraservice2.asmx and http://terraservice.net/landmarkservice.asmx)
3. The two web map servers – OpenGIS compliant map server, and the TerraServer map server – available at http://terraserver-usa.com/webservices.aspx)
4. The TerraServer data loading and administration program – a .NET WinForms based application.

2.3.1 TerraServer End User Applications

SQL Server is the only stateful resource that matters in the TerraServer applications. The TerraServer applications, which are 1 through 3 above, are stateless and disposable. An IP sprayer distributes traffic among them. They perform their functions by accessing one of the SQL Servers on a Storage Brick and formatting the results to be returned to the client.

The applications are unaware that there are multiple Storage Bricks and multiple Bunches of Bricks. The database partitioning is designed so that once connected to the proper SQL Server database, the T-SQL stored procedure has access to all the data and methods required to fulfill the application request. The process of connecting to a SQL Server database is where data partitioning and resource failover issues are handled within the TerraServer end user applications.

All TerraServer end user applications connect to a TerraServer SQL Server database using the *TerraServerConnection class.* It manages the *SqlConnection* object for the TerraServer application classes. Logically, TerraServer applications open the meta-data database, the gazetteer database, or the imagery database. At open time, TerraServer applications are aware of the data theme(s) of interest and possibly the location – scene or longitude – of interest. The TerraServerConnection class uses this information to determine the list of Storage Bricks that host a copy of the database containing the data. This is a comma separated list of SQL connection strings where each connection string identifies the Storage Brick, the SQL database, and logon information.

Figure 6 contains the pseudo-code showing how the TerraServerConnection class iterates through the SQL

[4] The busiest days on TerraServer are usually Monday and Tuesday. Saturday and Sunday are usually 50% of the Monday and Tuesday volume. There is a steady slide downward from Tuesday to Saturday.

connection list and opens a SQL connection. If an open attempt succeeds, the class returns a TerraServerConnection object that can execute SQL commands. If the class iterates through the entire list without successfully connecting to a database, then the class throws an error to the caller indicating that the TerraServer database is currently unavailable.

```
private void Open(string connectCSV)
{
  string[] conLst = connectCSV.Split();
  foreach(string con in conLst) {
   tsCon._sqlCon = new SqlConnection(con);
      try {
         tsCon._sqlCon.Open();
         return true;
      }
      catch{}
  }
  throw new Exception(“Can’t connect”);
}
```

Figure 6: TerraServer Web Connection Pseudo-Code

The TerraServerConnection class uses the *ConfigurationSettings* class provided by the .NET Framework to obtain the list of SQL connection strings from the configuration file associated with every .NET application. The application modules are hosted within the ASP.NET web framework so the *web.config* file is found in the root directory of each web site and virtual directory.

ASP.NET web applications can add their own key and value pairs to a section of the web.config file known as “appSettings”. The TerraServer web application extensively uses the entries in the appSettings section to customize the TerraServer application. The list of database connection strings is one class of variables stored in the appSettings section.

TerraServer end user applications call a method of the TerraServerConnection class associated with the type of database the application wishes to access. The database type categories are:

Gazetteer – exists once per TerraServer web instance
Meta-data – typically exists once per theme-scene pair.
Imagery – typically exists once per theme-scene pair.

The Meta-data and Imagery databases look for variables in the web.config file associated with Theme and Scene combinations. If no variables are found, then the TerraServerConnection methods look for a variable associated with a specific theme. If that fails, then the TerraServerConnection methods look for a variable identifying a single meta-data or imagery database.

Table 2 lists the variable name templates the TerraServerConnection methods use to locate the above database types. The variables are listed in search order by type.

The {0} and {1} strings are placeholders for the Theme and Scene numbers of the data to be returned. The placeholders are replaced with the appropriate values by TerraServerConnection class members.

Table 2: TerraServer Connection Config Variables

Type	**Variable Name Template**
Gazetteer	GazetteerDb
Metadata	MetaTheme{0}Scene{1}DB
Metadata	MetaTheme{0}DB
Metadata	MetaDataDb
Imagery	ImageryTheme{0}Scene{1}DB
Imagery	ImageryTheme{0}Db
Imagery	ImageryDb

The values associated with the variables are fully formed Sql Connection strings. A simple connection string is a list of name value pairs separated by semi-colon strings. The following is an example of the simplest connection string that is used within the TerraServer application:

```
Server=tsImgE1;Database=ImgT1S10;Trusted_Connection=yes;
```

The TerraServer web application defines one variable for each logical database depicted in Figure 3. The following is the definition of DOQQ Theme UTM Zone 10 database on one of the TerraServer web bricks:

```
<add key=”ImageryTheme1Scene10” value=
”Server=Brick11; Database=ImgT1Z10;
Trusted_connection=yes;,
Server=Brick21; Database=ImgT1Z10;
Trusted_connection=yes;”>
```

The TerraServerConnection Open methods split the value string into an array of connection strings. An open attempt is made on server “Brick11” first, and if it fails, a second attempt is made on server “Brick21”. By convention, the first digit in the brick name identifies the bunch. The second digit in the brick name identifies the brick within the bunch. Thus, “Brick11” identifies the server as “Brick 1 of Bunch 1”, and “Brick21” identifies the server as “Brick 1 of Bunch 2”.

The web.config files on 50% of the web bricks reference the Bunch 1 bricks first and the Bunch 2 bricks second in each database variable. The other 50% of the web bricks reference the Bunch 2 bricks first and the Bunch 1 bricks second in each database variable. This effectively balances the transaction load across all the database bricks.

2.3.2 TerraServer Administration Application

The primary TerraServer administration tasks are:
1. Adding new data.
2. Performing software maintenance – primarily installing security-patches.
3. Monitoring system performance.
4. Responding to user inquiries.
5. Performing hardware maintenance.

In addition, the host datacenter staff provides routers, connection to the Internet, and facilities.

Tasks 1 thru 5 above are handled on a part-time basis by two of us (Barclay and Chong.) Barclay is also evolving the TerraServer code base to support new themes and with innovations like [JAIN03].

Most administration tasks are performed with the TerraServer administration program. It manages and monitors data loading tasks and database maintenance activities on the Storage Bricks. Many users believe TerraServer is a read-only database. In reality, the USGS delivers several terabytes of new imagery each year. Since January 2004, we have received 60,000 images totaling 4 TB of Urban Area imagery and 13,000 files totaling 1 TB of DOQQ imagery. The data loading process is divided into two components – the tiling component and pyramid component. The tiling component processes the meta-data provided by the USGS and shreds input imagery into 200x200 pixel tiles. The pyramid component distributes the new tiles to the on-line imagery databases, creates the image pyramid, and updates the tile meta-data which makes the new data visible to the end user applications.

The administration program automates a number of common maintenance functions – database creation, database expansion, database backup, and database restoration. These functions can also be performed by using SQL Server's Enterprise Manager and Query Analyzer tools. The TerraServer administration program simplifies the administration tasks by automatically repeating the task on all clone bricks.

The administration program is aware of the topology of the Storage Bricks and Bunches of Bricks. In fact, it is the administration program that configures the Storage Bricks and Bunches. The data loading component ensures that data is redundantly distributed to the Storage Bricks that are supposed to be identical copies of each other.

The TerraAdmin program depends on the Admin SQL Server database for the definition of the Storage Bricks, Bunches, Web Bricks, and TerraServer Meta and Imagery databases. The TerraAdmin program uses the relationships between the Brick, TerraDb, and ThemePartition tables to determine the Storage Bricks and SQL databases that need to be updated as new imagery arrives from the data providers.

The TerraAdmin program detects when a failure occurs on a Storage Brick. If a load task is executing, it is aborted and placed in a failed state. The TerraAdmin program will resume execution of a failed load repairs have been made and the Storage Brick is returned to service. The aborted job is continued from the interrupted transaction such that successful updates to Bricks are repeated and done for the first time on the recovered Brick. This method ensures that all Storage Bricks contain identical data.

2.4 Media Failure Recovery

Each TerraServer Storage Brick has mirrored volumes RAID-1. The Brick's data remains available if a single disk fails. However, if both disks of a pair fail, then databases on that volume are lost.

If a double disk error occurs, requests to that Web Brick are automatically re-routed to the surviving Storage Brick by the connection class (as described in Section 2.3.1). Thus, applications are not aware that a failure has occurred. However, there is now only one copy of the database(s) available on the surviving brick which breaks our rule that there should always be two copies of each database.

When a double disk failure occurs, the TerraServer administrator uses SQL Server's on-line backup utility to backup a copy of each database that was affected by the failed disk volume. The backup operation is executed on the storage brick containing the surviving databases. The backup file is saved to one or more of the backup disk volumes located on the Backup Brick. The administrator performs the backup using one of three methods:

- SQL Enterprise Manager (GUI)
- SQL Query Analyzer (command line)
- TerraServer Administration program (GUI)

The following is an example backup command using the SQL Query Analyzer:

```
Backup database IT1z10 to disk=
     '\\BackupBrick\BackupG\It1z10.bck'
  with stats=10
```

This command copies the surviving database to a single backup disk volume. If the database is too large to fit on one volume, then something like the following command copies a database to multiple disk volumes on the Backup Brick. The SQL Server on-line backup utility writes to all backup devices in parallel, thus increasing I/O throughput:

```
Backup database IT1z14 to
  disk='\\BackupBrick\BackupH\It1z14a.bck',
  disk='\\BackupBrick\BackupI\It1z14b.bck',
  disk='\\BackupBrick\BackupJ\It1z14c.bck'
    with stats=10
```

The Storage and Backup Bricks are connected via a Gigabit Ethernet switch that also carries traffic between the web servers and Storage Bricks. The following table shows the throughput of three different backup operations:

Table 3: TerraServer Online Backup Performance

Db	Backup Size	Elapsed Time	MB per Second	Backup Disks
It4z10	62.8 GB	01:40:20	11.2	1
It1z13	287 GB	04:15:47	20.1	2
It1z15	302.8 GB	02:46:38	32.5	4

While the backup operation is in progress, the TerraServer administrator attempts to recover the failed disk devices. These steps often can recover a disk volume avoiding a trip to the data center. If the disk volume cannot be recovered, then the TerraServer administrator schedules a visit to the data center as soon as it is convenient.[5]

Once the failed volume is back in service, the lost databases are restored from the Backup Brick. As with the backup operation, the TerraServer administrator can use one of three tools to restore the data – (1) Enterprise Manager, (2) SQL Query Analyzer, or (3) the TerraServer Admin program. The following is an example Restore command executed from within SQL Query Analyzer:

```
Restore database It4z10 from disk=
      ‘\\tk2terrabkp11\BackupG\It4z10.bck’
 with stats=10,
 move ‘tsData0’ to ‘h:\It4z10\tsData0.mdf’,
 move ‘tsAdmin0’ to
‘h:\It4z10\tsAdmin0.ndf’,
…
```

The “move” clauses allow the TerraServer administrator to place the physical database files in different device and directory locations than where they were originally placed. The clauses are unnecessary if the files are going to the same device and directory name.

The database restore operations are repeated for each database that needs to be recovered on the replaced/repaired disk volume.

To date, the TerraServer Brick configuration has not experienced any disk volume failures. We have tested and measured the performance of backup operations from each storage brick to the backup brick while users are actively accessing the web site and databases. The numbers in the Table 3, TerraServer Online Backup Performance, show the results of some of these tests. We have not performed a restore operation during production operation.

2.5 Local Area Network Architecture

The MSN hosting service provides TerraServer with a *Front-End* Local Area Network (LAN) that load balances TCP/IP connection requests across the available web bricks. The web bricks sit in a demilitarized zone (DMZ) that provides an extra layer of firewall protection from internet attacks. The web bricks translate incoming HTTP and SOAP requests into database requests. The only traffic on the Front End LAN is end users HTML and image mime types.

The TerraServer web application is a client/server architecture. Web Servers are required to host the web page generation software, implemented as ASP.NET programmable web pages. The ASP.NET page classes use ADO.NET to connect to TerraServer meta-data or imagery database via the *Back-End* LAN, execute a stored procedure, and translate the returned data to HTML or an image mime type. The TerraServer Connection class performs the connection task for all web modules. As explained in section 2.3, Resource Failover the Connection class automatically handles failing over to a redundant storage brick should the primary brick be unavailable.

The LAN is also used by SQL Backup and Restore utilities to copy data from on-line storage bricks to the Backup Brick, and from the Backup Brick to recovered disk volumes on repaired Storage Bricks. Also on the Back End LAN are the Load Brick(s) which are used by the TerraServer administrator to update the imagery and meta-data located on the Storage Bricks. In the production configuration, we deployed on Load Brick with 6 terabytes of raw storage physically located on the main Redmond campus, 50 kilometers from the rest of the TerraServer hardware. It is connected to the data center hosting the TerraServer storage bricks and is a member of the Back End LAN.

Thus, the TerraServer Brick architecture requires plentiful and available local area network capacity to reliably perform two tasks: (1) extract meta and imagery data to present to end users, and (2) backup and recover data on failed components – disks or entire storage bricks.

TerraServer has three virtual local area networks (VLANs) depicted in Figure 7:

Back End LAN (BELAN) connects Storage Bricks to the Web Bricks, and all bricks to the Load Brick(s).

Front End LAN (FELAN) connects the Web Bricks to the Storage Bricks.

Management LAN (MLAN) connects a Management Brick to the KVM/IP console and the intelligent Power Distribution Units (PDU/IP).

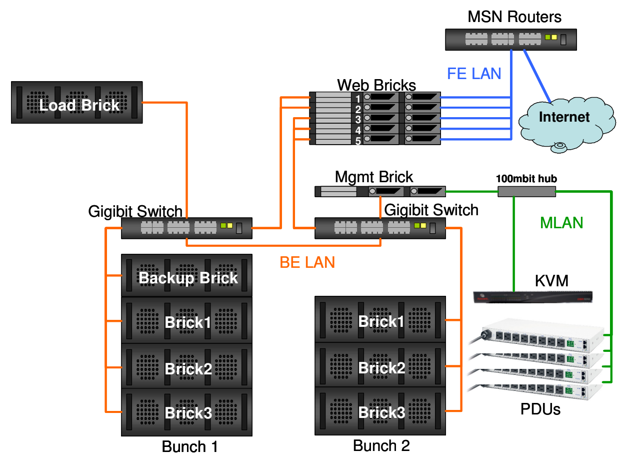

Figure 7: TerraServer Brick Network Topology

The Front End LAN is implemented by network switching gear provided by the MSN data center. The data center is responsible for routing, securing and load-

[5] The data center hosting the TerraServer storage and web brick is 30 miles from the TerraServer administrator’s office and is difficult to reach quickly due to its location and traffic conditions in the Seattle area.

balancing TerraServer IP requests from the public internet to the TerraServer Web Brick farm. Currently, there are five web bricks in the web farm.

The Back End LAN is implemented by two SMC 24-port "Tiger" Gbps Ethernet switches. One switch supports Storage and Web Bunch 1. The second switch supports Storage and Web Bunch 2. The two switches are connected so that Web Bunch1 can access Storage Brick 2, and Web Bunch 2 can access Storage Brick 1.

The Front End LAN connects the web server nodes to the public internet. This has the effect of isolating the database servers located on the Back End LAN from the public internet. The only traffic on the Front End LAN is end user HTML, SOAP/XML, and image mime type traffic.

The Management LAN (MLAN) enables the TerraServer administrator to obtain a virtual connection to the KVM console device and PDU/IP units physically housed in the TerraServer computer racks. This enables the administrator to power on or off storage or web bricks, and connects the computer console monitor remotely as if the administrator was physically next to the rack.

2.5.1 *Front-End LAN Traffic Patterns*

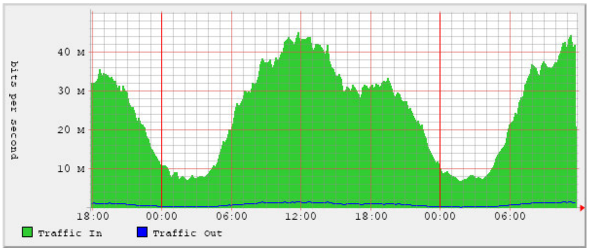

Figure 8: Daily Internet Traffic Pattern

Figure 8 shows the typical daily network traffic pattern on the Front End LAN. The graph is from the perspective of the MSN network router. The blue line represents bits per second from the router to the TerraServer web servers. This is the data sent to TerraServer web pages from client browsers and applications. The green shaded area, represents bits per second transferred to the router from the TerraServer web servers. This is the data sent by TerraServer web pages in response to client requests. TerraServer is busiest between 7am and 10pm where the output rate exceeds 30 mega-bits per second. Between 10pm and 7am, TerraServer output rate slides to below 10 mega-bits per second.

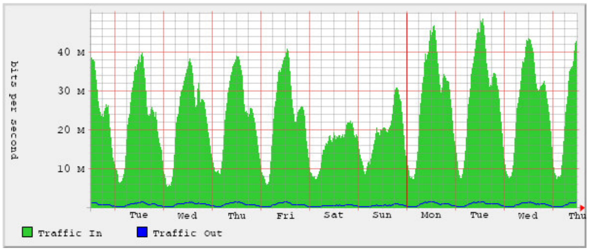

Figure 9: Weekly Internet Traffic Pattern

Figure 9 shows the Internet traffic pattern over a nine-day span. Traffic consistently peaks over 30 mega-bits per second during the weekdays from 7am to 10pm. There is a noticeable and consistent spike at noon and then a smaller spike after 6pm each day. The weekend traffic is approximately 50% of the Monday and Tuesday traffic.

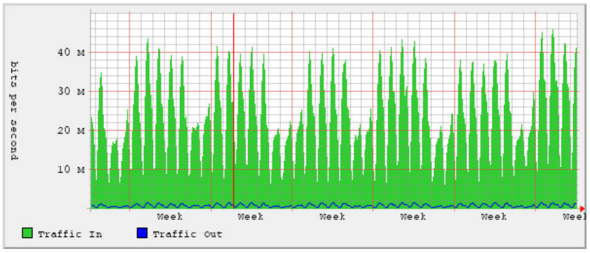

Figure 10: Monthly Internet Traffic Pattern

Figure 10 shows the Internet traffic pattern over a six week period. The graph confirms the relative consistency and predictability of the TerraServer usage pattern. Occasionally there are traffic spikes when the site is publicized, but otherwise traffic is fairly predictable.

2.5.2 *Web Request Traffic*

The previous section showed how system load varies during the day. Table 4 shows how the load is divided among the applications over the last 12 months (Nov 2003 thru Oct 2004) compared to two years ago (Nov 2001 thru Oct 2002).[6]

Table 4: Million web requests per month: 2004 vs 2002

	Page Views		**Tiles**		**Web Svcs**		**Web Maps**	
Mon	**2004**	**2002**	**2004**	**2002**	**2004**	**2002**	**2004**	**2002**
Oct	1.7	0.9	15.2	2.2	0.50	0.04	0.73	0.15
Sep	1.8	0.9	14.2	2.2	0.70	0.02	0.50	0.07
Aug	1.7	0.8	12.5	2.0	1.90	0.02	0.36	0.08
Jul	1.6	0.8	12.8	2.0	1.19	0.02	0.32	0.02
Jun	1.8	0.8	11.8	2.0	1.01	0.01	0.34	0.01
May	1.9	0.8	13.1	2.1	1.48	0.02	0.34	0.02
Apr	1.5	1.0	15.9	2.1	1.11	0.01	0.28	0.02
Mar	1.6	0.8	14.4	4.8	0.43	0.03	0.27	0.02
Feb	1.6	0.8	11.6	4.6	0.50	0.01	0.24	0.01
Jan	1.3	1.6	11.2	5.6	0.27	0.01	0.19	0.01
Dec	1.2	1.5	10.1	4.5	0.55	0.01	0.15	0.01
Nov	1.1	1.9	19.5	5.8	0.97	0.01	0.12	0.01

The Page Views and Tiles columns report the average daily volume of requests for the HTML web application. The Web Svcs columns report the average daily request volume for the SOAP/XML service. The Web Maps columns report the data volume for the Web Map servers.

[6] Unfortunately, web statistics were inadvertently dropped for a quarter of 2003 making a year to year comparison impossible.

We attribute the dramatic growth in the Page View and Tiles of the HTML web application to the addition of the Urban Area data set. In October 2003, we loaded the first city into the database. In early spring we began loading Urban Area in volume. We believe this is what has led the average page views from roughly 800k per day to over 1.5 million per day.

In 2002, the web map server and web services components were very new. Users were trying to determine whether web services were viable in their business. We attribute the low volume to developers experimenting and demonstrating the capability to management. Midway through 2003, we began to see a dramatic climb in both web service calls and web map server requests.[7]

From a system architecture perspective, we are pleased see that the brick design has gracefully handled the usage growth we have experienced over the past twelve months.

2.6 Console Operations and Power Management

The TerraServer SAN Cluster hardware was nearly trouble-free [Barclay04]. We expect to have more hardware issues with the Bricks architecture because the Bricks are low cost components without the sophisticated engineering that went into the SAN Cluster. To minimize the trips to the data center, we deployed a console concentrator and four power distribution units that are accessible from via the MLAN. This enables us to perform all the typical functions normally performed in the data center except physically replacing parts.

The *house-rules of* MSN data centers limit each rack to two 20 amp 208V circuits. Conservative power policies require that equipment not exceed 16 amps per circuit. The TerraServer hardware would fit in one rack. As shown in Table 5, TerraServer equipment consumes roughly 44 amps (~9KW), so MSN datacenter house rules mandate two racks.

Table 5: TerraServer Power Budget (amps @ 208V)

Device	Amps	Devices	Net Amps
Storage Bricks	3.5	7	24.5
Web Bricks	2.3	6	13.8
Ethernet Switches	1.3	2	2.6
Avocent KVM	1.3	1	1.3
Misc.	1.7	1	1.7
Total			**43.9**

To allow remote control of the power to each brick, we installed two Power Tower XL, 8 outlet PDU (Power Distribution Units) manufactured by Server Technology [ServTech] in each rack. The Storage Bricks were plugged into one PDU and other equipment was plugged into the second PDU.

An Avocent DSR-1010-AN remote KVM/IP [Avocent] supports up to 16 computer systems. Each computer's video, keyboard, and mouse port is connected to an Avocent service interface module which converts the signals to an Ethernet message sent to/from the Avocent DSR-1010. The Avocent DSR-1010 is connected to a flat panel monitor, keyboard, and mouse installed in rack two.

An 100 mega-bit Ethernet hub serves as the MLAN network and connects the KVM and PDU gear to a web brick configured to be a "Management Brick".

The Management Brick is also connected to the Back-End LAN (see 2.5, Local Area Network Architecture). We can connect from the Load Brick that has dual home connection to the Microsoft Corporate Network (CorpNet) and the data center's Back-End LAN. Using Terminal Server running a computer in our office, we first use Terminal Server to connect to the Load Brick. From the Load Brick session, we use Terminal Server to connect to the Management Brick located in the racks. We can then connect to the KVM or the PDUs from the Management Brick

The Avocent KVM comes with a GUI that allows a user on the Management Brick to virtually connect to any console managed by the KVM. The user interface is similar to a Windows Remote Terminal Server session. A window is associated with each open console session. GUI buttons can be clicked to communicate special character sequences such as CTRL+ALT+DELETE, and ESCAPE to the console for the remote system.

The Server Technology PDUs supports a command line interface assessable through Telnet protocol. The command line supports simple commands such as ON, OFF, etc. that can be directed at any individual power port. This allows us to cycle the power of any device connected to the PDUs.

Taken together, the IP-based PDU and IP-based KVM allow us to operate the entire system remotely.

2.7 Data Loading Changes

The deployment of the TerraServer Brick architecture required changes to the imagery loading process. Previously, there was a single copy of both the imagery tiles (blobs) and tile meta-data, although they were stored in separate databases. This was a simpler design to implement.

In the brick design, imagery and meta-data is replicated in multiple imagery and meta-data databases. To minimize the impact on other TerraServer applications, the TerraServer load process ensures that imagery tiles are inserted or updated before the tile meta-data is inserted or

[7] We believe most web service applications call the web service to obtain the meta-data necessary to determine the parameters to pass to a web map server request.

updated. The web application is written to query the tile meta-data first. If the web application finds a meta-data row, then the load application guarantees that there is an associated tile row in the imagery databases.

We experimented with SQLserver 2000 Replication technology. Replication would have minimized the changes to the TerraServer load programs. The loaders would continue to update only one database and SQL Server replication would be responsible for copying data to the redundant imagery and meta-data databases. However we did not see an easy way to guarantee that **all imagery** databases would be **updated before any** meta-data database. SQL Server replication has numerous features focused on buffering data should network or other problems exist in transmitting data. These are terrific capabilities in most wide-area distributed environments. But we could not discover a way to guarantee that meta-data would inserted/updated after all imagery databases were updated. SQLserver 2005 mirrored-systems (log shipping) seem to solve this problem.

We choose to re-write the TerraServer load system to use a single "admin" database as the center of the load system, linked servers to define the connection to imagery and meta databases, and distributed transactions to atomically update multiple servers in single transactions. The Admin database maintains a synchronized copy of the online tile meta-data and new, "candidate" tiles received. Figure 11 shows the flow and processing steps involved in an update cycle.

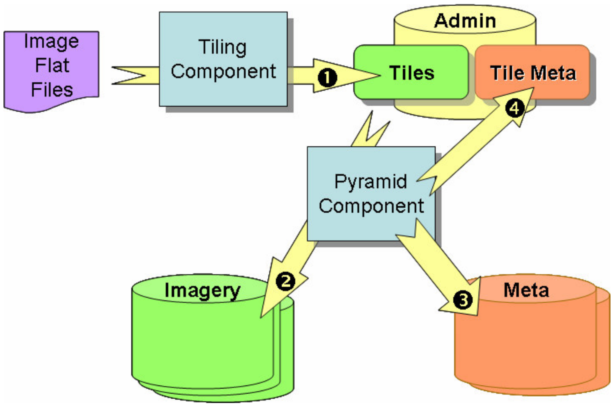

Figure 11: TerraServer Load Process

❶ The tiling component shreds input imagery into tiles stored in the Admin database.

❷ The pyramid component copies tiles to the Imagery databases

❸ The tile meta data is updated to reflect the presence of updated tiles in the Imagery databases.

❹ The tile meta data is updated in the Admin database to match the changes made to the online tile meta data.

The architecture and features of the new TerraServer load system are the subject of a forth-coming Technical Report.

2.8 Architecture Summary

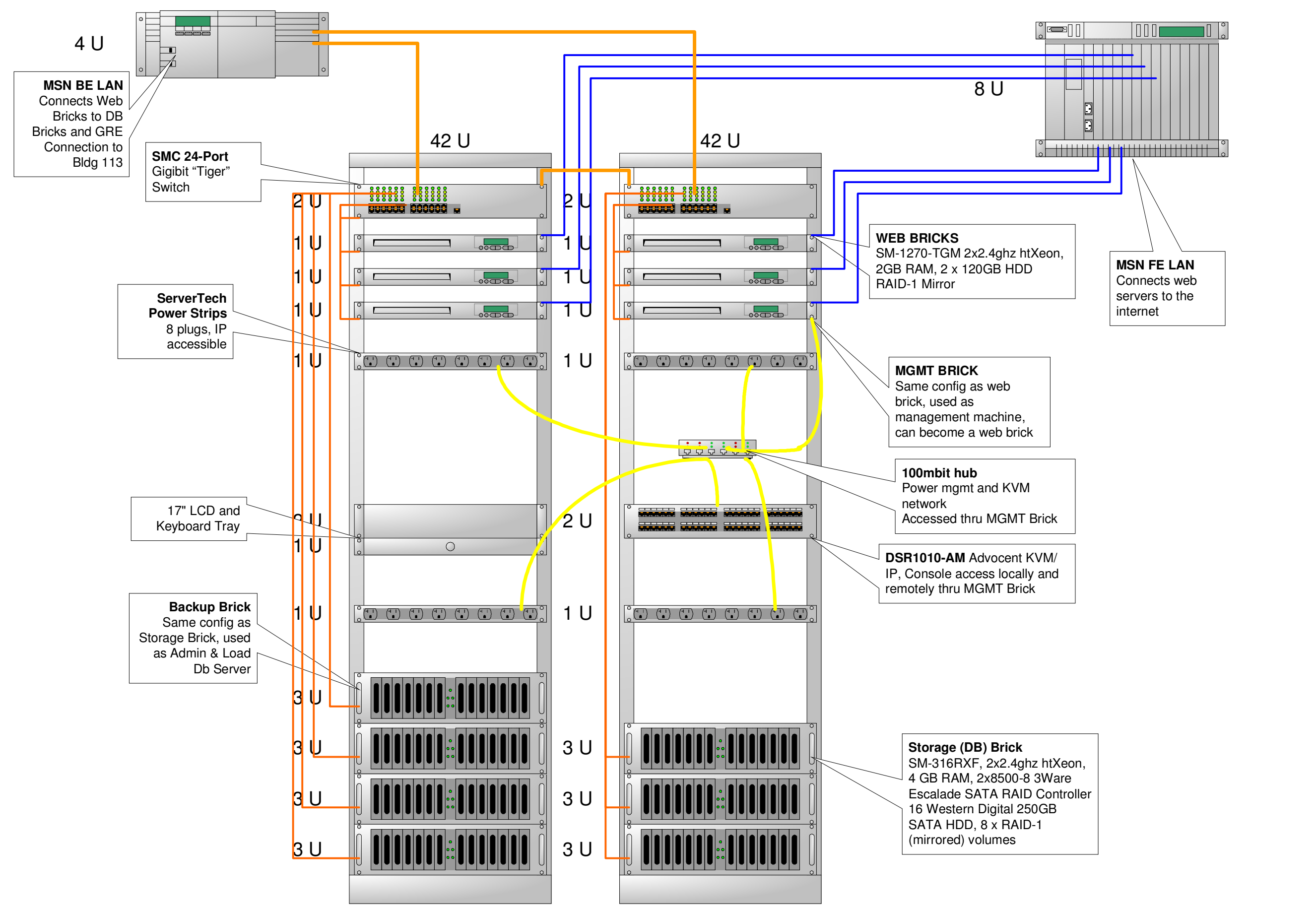
4 U
MSN BE LAN
Connects Web Bricks to DB Bricks and GRE Connection to Bldg 113
SMC 24-Port
Gigibit "Tiger" Switch
42 U
42 U
8 U
2 U
1 U
1 U
1 U
1 U
2 U
1 U
1 U
3 U
3 U
3 U
3 U
2 U
1 U
1 U
1 U
1 U
2 U
1 U
3 U
3 U
3 U
ServerTech Power Strips
8 plugs, IP accessible
17" LCD and Keyboard Tray
Backup Brick
Same config as Storage Brick, used as Admin & Load Db Server
WEB BRICKS
SM-1270-TGM 2x2.4ghz htXeon, 2GB RAM, 2 x 120GB HDD RAID-1 Mirror
MSN FE LAN
Connects web servers to the internet
MGMT BRICK
Same config as web brick, used as management machine, can become a web brick
100mbit hub
Power mgmt and KVM network
Accessed thru MGMT Brick
DSR1010-AM Advocent KVM/IP, Console access locally and remotely thru MGMT Brick
Storage (DB) Brick
SM-316RXF, 2x2.4ghz htXeon, 4 GB RAM, 2x8500-8 3Ware Escalade SATA RAID Controller
16 Western Digital 250GB SATA HDD, 8 x RAID-1 (mirrored) volumes

Figure 12 TerraServer Brick Configuration

Figure 12 shows how the TerraServer hardware is configured. As explained in Table 5, the power budget and datacenter house rules mandated two racks even though the equipment can physically fit into one 42u rack.

We put Bunch1 in rack 1 and Bunch2 in rack 2. Computer names follow the format: 'PPPPPFFF##' Where:

- "PPPPP" identifies the product; we use "Terra" for TerraServer-USA.
- "FFF" identifies the function of the system; we use the following values:
  - SQL – SQL Server database server
  - WEB – Web Server
  - MGR – TerraServer Management server
  - BKP – TerraServer Backup Brick and Admin Server
- "##" identifies the system number; we use the first digit to identify the rack and the second digit to identify the system.

Rack 1 contains the servers TerraWeb11, TerraWeb12, TerraWeb13, TerraBkp11, TerraSql11, TerraSql12, and TerraSql13. Rack 2 contains the servers TerraWeb21, TerraWeb22, TerraMgr21, TerraSql21, TerraSql22, and TerraSql23.

Each rack is configured with its own SMC 24-port Gigabit Ethernet switch and two Server Technology Power Distribution Units. The web bricks and Ethernet switches are powered by one PDU, and the storage bricks are powered by the second PDU in each rack. This configuration enables the TerraServer web site to continue to function if one PDU or an entire rack loses power.

3 Operations Experience

Two prototype TerraServer Storage Bricks were purchased in the fall of 2003 and tested. The testing verified that two bunches of bricks could support the TerraServer I/O load comfortably [Barclay03]. Indeed, based on a stress test, a single bunch could handle four times the Fall 2003 peak load. The production configuration of six storage bricks and five web servers were purchased in October 2003 from Silicon Mechanics [SilMech]. One of the test bricks became the Backup Brick and was installed with the six Storage Bricks in the MSN Tukwila data center. The other test brick became the Load Brick and remained in the TerraServer development and test lab located on the main Redmond campus. This section describes our operational experience installing and operating the TerraServer brick configuration since it entered service in mid-November 2003.

Our expectations were not very high. Many experienced administrators told us to expect many disk failures and problems with our inexpensive, "white-box" PC configurations. Typical advice was "SATA drives will fail all the time", "SATA is not SCSI and can't keep up with the I./O demands", etc. We had previously experienced excellent reliability from our Compaq Cluster and SAN [Barclay04]. We practically never visited the data center to perform maintenance work. We were advised to be prepared to be in the data center frequently to service the bricks, disks, or other components. The advice we received was so severe that we made a substantial investment, over 5% of the total system cost, in remote management capabilities provided by the Advocent KVM/IP and ServerTech IP PDUs fearing that we would be living at the data center.

Our experience has been the exact opposite. The storage bricks and the SATA disk drives have been every bit as reliable as the Compaq Cluster and SAN containing SCSI disks. In three years, approximately thirty-two SCSI drives failed in the Compaq SAN and web farm. Due to triple-disk mirroring, we never experienced any data loss. To date, a total of nine SATA drives have failed and been replaced. Due to dual-disk mirroring, we have not experienced any data loss and have not had to put our "just-in-time-backup" process into action.

In part, our positive experience may be due to careful design. We were careful about disk cooling – SATA drives are rarely cooled with the same care that a SCSI array receives. We experimented with SMART and worked hard to keep our disks below 47º C. The Silicon Mechanics disk packaging keeps the drives below body temperature.

The SATA drives and Storage Bricks are not perfect. The remainder of this section lists all the operations issues and events we experienced since November 2003 operating the Storage Brick configuration. The events can be lumped into the following groups:

1. Disk connection issues.
2. Mysterious disk offline issues.
3. Disk failures due to power failure/surge in the data center.
4. Applying required software patches.
5. Resetting the SQL Server domain password.

3.1 Disk Connection Issues

SATA drives connect to the 3Ware controllers using a fairly thin cable. The small cable significantly enhances the air flow through a system box which is, in part, why SATA drives seem to be more reliable than parallel ATA (PATA) disk drives. Silicon Mechanics, the manufacturer of our storage bricks, takes the additional step of creating custom SATA cables that are the precise length required and then bundling them to improve the cooling air flow.

Figure 13 shows how the SATA cables connect to the disk drives on the left, combined into a bunch with cable ties, and routed to the 3Ware controllers on the right. We found that precise length and bundling caused enough tension on the connector for the disk drive to go off-line. Reducing the tension and re-seating the connector corrected the problem. We experienced this problem five times before we diagnosed it.

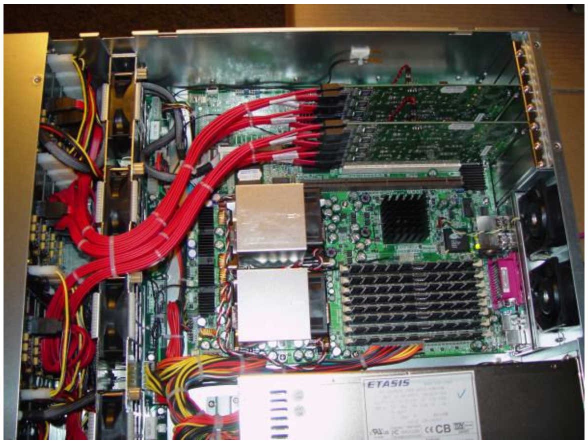

Figure 13: Storage Brick SATA Cabling

The first occurrence happened to our Load Brick located in the Redmond lab. The Load Brick is identical to the Backup Brick. It was purchased in September 2003 and used for testing SATA drives and the Storage Brick configuration [Barclay03]. In January, a disk went off-line. The 3Ware controller utilities failed to get the failed disk back. A complete system shutdown including a power recycle and removing the disk and re-inserting it also failed to revive the disk.[8] We opened the box and noticed the cable tension was bending the connectors on the drives to one side. We loosened the cables, reseated

[8] All non-O/S drives on Storage Bricks are installed with hot-swap trays. Thus we pulled the tray out and re-inserted it to verify the connection there the tray plugs into the system box.

the connector, and restarted the system. The 3Ware controller detected the drive and placed it back in service.

In late February 2004, the same event occurred on two other "live" Storage Bricks. We were installing a Windows 2003 operating system update that required a re-boot of the Storage and Web Bricks. After the systems were re-booted, the bricks were checked for any software and hardware failures. Two disks were off-line on two separate storage bricks. Both were RAID-1 volumes so SQL Server and the TerraServer application were able to continue to access the data on both drives. Through our remote management connections, the TerraServer administrator used the 3Ware tools to re-enable the disks. This did not work and the disks remained off-line.

The administrator visited the data center the next day. On both the off-line drives, the cable tension was bending the connectors on the disk to one side. The administrator released the tension, re-seated the connector, closed the box, and re-started the system. The failed drives were recognized and re-synchronized their mirrors. May was the fifth and last occurrence of the cable tension issue.

Silicon Mechanics has reduced cable tension to eliminate these types of errors. In addition, a new type of cabling system has been developed for SATA drives based on infiniband connectors. This will eliminate cable tension as a potential problem for SATA configurations.

3.2 Mysterious Offline Issues

From time to time an individual disk in a RAID-1 mirrored volume just disappears and goes off-line. There is no evidence in the system event log, or the 3Ware management utility log stating why the disk went off-line.

The TerraServer administrator goes through the following process when he discovers a disk has gone off-line:

1. He uses the 3Ware management utility to re-activate the disk. This never works.
2. He power cycles the storage brick. He connects the ServerTech PDU remotely by using Windows Terminal Server to connect to the Management Brick, then using TelNet to connect to the PDU. He turns the power port off, then on.
3. When the brick reboots, he launches the 3Ware management utility and attempts to reactivate the disk.
4. If the disk does not reactivate, then he schedules a trip to the data-center where the power recycle is repeated, the disk carrier is removed from the system, re-inserted/re-seated, and the system is powered up.
5. If the disk cannot be activated, then he powers off the system, pulls the disk carrier from the system, removes the drive from the carrier, replaces the drive, and restarts the system.

The "mysterious" off-line errors have occurred eight times since November 2003. Four of the eight occurrences the disks were recovered at step 3 above (after the first power recycle). Two of the eight occurrences the disks were recovered after re-seating the disk carrier at step 4 above. The other two occurrences required the disks to be replaced and returned to the manufacturer.[9]

We have two other disk failures where we had to replace the disk drives. In both instances however, there were quite a few disk error messages reported in both the Windows event log and the 3Ware Management utility logs. One failed disk occurred on the Backup Brick. The other failed disk occurred on a Web Brick.

As of early October, we have a third drive that is beginning to report SMART errors. We expect it to fail in the next week or so.

In summary, out of a total of 140 disks, we have had 4 disks fail and be returned for warranty, and we have a fifth that is showing signs of failing soon.

3.3 Data Center Power Cycle

On July 13th, we experienced every professional data center manager's worst nightmare – a tradesman hit the big red emergency power off button immediately "pulling the plug" on one-quarter of the entire data center. What the tradesmen meant to do was turn off a set of the air flow fans, but inadvertently hit the wrong button even though the button is marked with very large warning signs.

The data center personnel followed all the normal procedures required to reset the power in the data center. Since this is an emergency switch, there are processes and procedures to ensure that the center is not on fire and in fact it isn't a real emergency. When power is restored, it is "trickled" to the racks to minimize the power surge caused by powering on so much equipment all at once.

In total, our servers were off-line for a total of 13 minutes. It has been the only time the TerraServer-USA web site has been totally unavailable since switching to the TerraServer Brick configuration.

When the storage bricks were restarted, eight disks in eight separate RAID-1 mirrored volumes did not automatically activate. The TerraServer administrator followed the procedure outlined in section 3.2,. Two of the disks were reactivated at step 3. Three other disks were reactivated at step 4. However, two of the three disks failed approximately a day later and had to be replaced. Three disks did not reactivate even briefly and were immediately replaced (step 5).

In summary, we lost and replaced five disks as a direct result of the data center power recycle (in addition to the 4 failed disks mentioned in the previous section). We categorize these as "non-mysterious" because we attribute them to an external event. However, we are a puzzled as to what actually happened when the power went off or when it was restored that would seem to be so hard on our

[9] We purchased Western Digital SATA drives with a three year warranty.

disk drives. This could be related to not just powering down but powering up, because of the "brownout" bringing up several hundred systems at once.

3.4 Applying System Software Updates

The last several years have been an extremely active time for the distribution of software updates to Microsoft operating systems and applications such as .NET Framework and SQL Server which the TerraServer application is built on. Since the TerraServer Bricks went into production service, there have been eight occurrences where we had to apply software updates to the Storage and Web Bricks. Seven of the eight have required a re-boot.

We use the following procedure when updating the system on the Storage and Web Bricks:

1. The software update is applied to the Backup Brick using the Windows Update application. The system is re-booted. We run a series of tests to verify that Windows, SQL Server, the .NET Framework, and the TerraServer web applications[10] function properly.
2. Each Storage Brick in Bunch 1 is updated with Windows Update application and re-booted. The Bunch 1 web bricks automatically fail over to Bunch 2 storage bricks.
3. Each Storage Brick in Bunch 2 is updated with Windows Update and re-booted. The Bunch 1 and Bunch 2 web bricks automatically fail over to the Bunch 1 storage bricks.
4. Each web brick is updated with Windows Update and re-booted. When each brick restarts, it automatically accesses the default storage bunch. Thus Bunch 1 web bricks access Bunch 1 storage bricks; bunch 2 web bricks access Bunch 2 storage bricks.

This procedure does not interrupt service – it is a rolling upgrade. Some bricks are always up and are providing access to all the data and all the services. To minimize impact on TerraServer end users, we usually try to perform software upgrades in the late afternoon or early evening Pacific Time. TerraServer traffic normally slows down in the late afternoon before it gets an early evening boost.

3.5 Resetting the SQL Server Account Password

Every 60 days, datacenter house rules required that we change the passwords to the domain account used by the SQL Server service. In the past, we used an internal tool to change the password without stopping and restarting the SQL Server instance. Occasionally the tool would silently fail. This would eventually cause SQL Server instances to fail at random times in the future, disabling part or all of the web application. Because we now have a redundant set of bricks and very short failover time, we stop and restart SQL Server as part of the password change process to re-authenticate the server. This also gives us a chance to apply system software patches at the same time. Concurrent password and software updates were twice done at the same time.

We follow the procedure outlined in section 3.4 when changing the SQL Server service password. If we are doing both – applying a software update and changing the password, then we first change the SQL Service account password on the Backup Brick and then apply the software update. We continue by changing the password on the Bunch 1 storage bricks prior to applying the software update, and do the same on the Bunch 2 storage bricks.

If we are just changing the password SQL Server is restarted on the Backup Brick. If this succeeds, the same procedures is followed on the Bunch 1 storage bricks, and then on the Bunch 2 storage bricks. On each web brick, we restart the IIS service to force the TerraServer application to restart. This forces all the web bricks to connect to their default storage brick. As explained in [Barclay04], being able to practice operations procedures on the load brick (which does not affect system availability) is a real luxury. This first step tests the scripts and the process. It helps avoid some of the mistakes we made in managing the *live* TerraServer SAN Cluster.

3.6 Operations Summary

Table 6 summarizes all the operations events in the past year. The Count column identifies the total number of occurrences of the event. The Data Center Trips column identifies the subset of the count that required a trip to the data center to correct the problem found.

Table 6: Operation Event Summary

Event	Count	Data Center Trips
Reset SQL Account Password	6 x 7 [11]	0
Software Upgrades Requiring Reboot	9 x 13 [12]	0
Power-cycle brick to recover disk	4	0
Reseat bent SATA drive connector	5	2
Reseat SATA drive carrier	2	2
Replace SATA drive (normal wear and tear)	4	4
Replace SATA drive (data center power cycle/brownout)	5	2
Failed SATA drives not yet replaced	1	0
Failing SATA drives	1	0
Total	**37**	**8**

[10] The Backup Brick is technically a SQL Storage Brick, however it runs IIS, .NET Framework, and the TerraServer web application for testing purposes.

[11] The SQL password was reset six times on seven storage bricks (=42 times in all).

[12] Nine software updates were performed on thirteen servers – seven storage brick and 6 web bricks (117 reboots in all).

The SATA drive failure rate is higher than the rate we experienced with SCSI disks on the Compaq Storageworks SAN. We lost 9 drives out of 140 SATA drives on the Web and Storage Bricks in one year. This is a 6.4% annual failure rate. In contrast, the Compaq Storageworks SAN and Web servers lost approximately 32 drives in three years out of a total of 194 drives.[13] This is a 5.5% annual failure rate.

The failure rates indicate that SCSI drives are more reliable than SATA. SATA drives are substantially cheaper than SCSI drives. Because the SATA failure rate is so close to the SCSI failure rate gives SATA a substantial return on investment advantage. Based on the warnings from experts, we were prepared to replace every SATA drive once per year. We are ecstatic that we have missed our estimate by over a factor of 10!

We installed fairly expensive KVM ($6000) and PDU equipment ($3000) to enable the administrator to perform power and console operations without having to travel to the data center. This equipment enabled the administrator to diagnose issues that would have required a trip to the data center a total of fourteen times. Five of the fourteen uses avoided a trip to the data center. The additional cost to our project for these devices is $6000.[14] The investment in remote IP KVM and PDU equipment was a major time and cost savings.

4 Conclusion

The TerraServer Brick architecture, server equipment purchased from Silicon Mechanics, and the SATA disk technology has exceeded our expectations in every aspect.

We already knew that SATA disks and white-box PCs could meet the performance requirements because of testing done in October 2003 [Barclay03]. We were frightened into thinking the failure rate of the SATA disk drives would be 100%. The actual annual failure rate has been 6.4% which is reasonably close to the 5.5% SCSI disk failure rate. The SATA drives combined with the reliability and performance of the 3Ware RAID controllers are formidable competitors to SAN technology at a fraction of the cost.

We expected the “white-box” servers to be less reliable and the service to be worse than what we received from Compaq (now HP) on the SAN Cluster. We had a handful of reliability issues[15] and excellent service from Silicon Mechanics that so far is on par with the experience we had with Compaq for over five years.[16]

We also experienced zero “blue screens” or other unexplained system crash. Actually, we didn’t experience any issues with the system software or hardware that resulted in a system crash.

We expected to add a substantial amount of software to the TerraServer application to detect and recover from failed SQL Server instances due to disk failures and other hardware issues. We were very pleased to find out that all the hard work in detecting failed SQL Server connections already exists in the MDAC stack used by the ADO.NET libraries. This allowed us to abandon our complicated, thread-safe class designed to detect and recover from broken connections. Instead we ported the TerraServer application from a single instance architecture to a fault-tolerant, redundant instance architecture with the handful of statements listed in Figure 6 of Section 2.3. The failover times have improved from 45 seconds on the SAN Cluster to a few seconds in the brick design.

[13] We do not have an exact count of the drives replaced on the Compaq SAN because maintenance was often done to the Compaq Storageworks without our presence. Thus, we do not have a verified accounting.

[14] The Avocent KVM costs approximately twice the price of a standard KVM unit. The MSN data centers supply non-IP PDUs which we replaced with the ServerTech IP PDU units.

[15] We categorize the “bent SATA connector” issue a manufacturing issue.

[16] Silicon Mechanics is an experienced but young company in the Seattle area. They are highly focused on service quality. We recommend them highly.

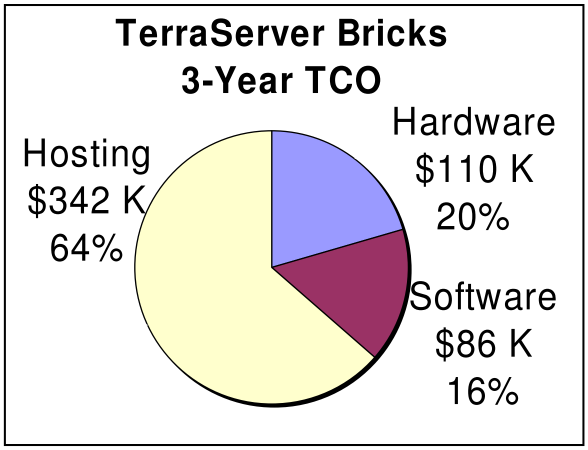

Figure 14: TerraServer Bricks 3-Year TCO

Figure 14 breaks out the three-year cost of ownership for the TerraServer Brick configuration into hardware, software, and hosting. We assume a 10% annual cost increase in hosting charges and assume that we will not be adding or changing hardware configurations or upgrading to new system software releases during the three year period. The software license prices are list prices assuming no discounts, no enterprise agreements, or other special pricing. In practice most customers would pay lower software prices.

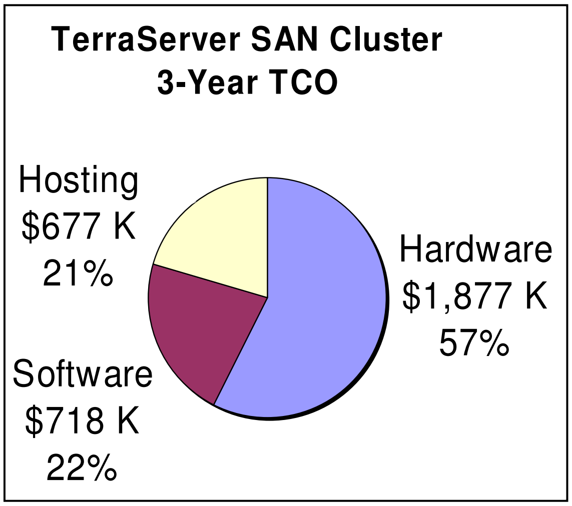

Figure 15: TerraServer SAN Cluster 3-Year TCO

Figure 15 shows the three year cost of ownership of the TerraServer SAN Cluster. This includes the actual hosting cost for the three year period. The software license prices were computed using the same method as in Figure 14 – list prices with no discounts.

Figures 14 and 15 do **not** give an "apples-to-apples" comparison. They compare two systems from different hardware generations, late 2000 versus late 2003. But they do illustrate how dramatically computer technology has changed in four years. Three-year TCO has gone from $3.3M to $0.5M – a 6-fold cost reduction.

In the SAN Cluster, the Hardware and Software components dominated the three-year cost. This is reversed in the TerraServer Bricks configuration where the Hosting component dominates the three-year cost. The cost ratios and total amounts vary for several reasons:

Disk density improvements allowed the TerraServer bricks to house 28 terabytes of 250gB SATA hard drives in one rack versus 18 terabytes of 18gB and 73gB SCSI hard drives and controllers in four racks. This dramatically reduces hardware and hosting charges.

Application-based failover versus hardware and system-software-based failover reduced the cost of the system software license fees and the hardware. The TerraServer SAN Cluster depended on Windows and SQL Server clustering features provided by the higher priced Enterprise Editions of both products. Implementing failover within the TerraServer application allowed us to substitute the more economical Standard Editions of Windows and SQL Server.

Replicated Disk Storage versus Tape Backup media recovery eliminated the cost of the tape library and tape media from the TerraServer Brick configuration. This is a small percentage of the total cost of the SAN Cluster, but not insignificant. It also reduced tape management software license fees – again a relatively small, but not insignificant sum. Maintaining a replicated online disk copy of the data increased data availability while substantially reducing hardware and hosting costs.

In addition to all the benefits listed so far, the brick configuration is simpler and less error-prone to manage. The bricks are all standard equipment directly supported by the operating system. There is no additional maintenance activity required beyond applying Windows and system software upgrades. Having spare equipment like the Backup Brick and the Management Brick reduces our risk of operations and maintenance errors because we have the opportunity to test all upgrades on identical equipment in the exact environment before upgrading the production bricks.

In summary, we conclude, like Yahoo, Google, MSN Hotmail, and MSN Search, that commodity storage and servers are the price/performance choice for high-volume web applications. While we loved the TerraServer SAN Cluster and its ability to detect and handle failures transparently, the price, performance, and reliability benefits of the TerraServer Bricks configuration out-weigh the costs of implementing failover and redundancy logic in the application. We expected to find limitations and missing features in Windows, .NET, and/or SQL Server that high-availability web sites would need to deploy Windows and SQL Server on commodity servers. We were wrong. Windows 2003, .NET 1.1 and SQL Server 2000 have all the engineering robustness and features required for users to deploy highly-available and high-volume web applications with little additional investment in application development.

5 References

[ADIC03] ADIC Scalar 1000 TerraServer case study, http://www.adic.com/us/collateral/Terra_ServerCS.pdf

[Avocent] Avocent Corporation, Huntsville, AL, http://www.avocent.com

[Barclay98] T. Barclay, et. al., The Microsoft TerraServer, Microsoft Technical Report MS TR 98 17, Microsoft Corp, Redmond, WA. http://research.microsoft.com/research/pubs/view.aspx?tr_id=155

[Barclay99] T. Barclay; J. Gray; D. Slutz "Microsoft TerraServer: A Spatial Data Warehouse", June 1999, Microsoft Corp., Redmond WA. http://research.microsoft.com/research/pubs/view.aspx?tr_id=280

[Barclay00] T. Barclay, Victoria Rozycki, "About Microsoft TerraServer – How TerraServer Works", Microsoft Corp., Redmond, WA. http://terraserver-usa.com/About.aspx?n=AboutStoryOverview

[Barclay01] T. Barclay, A. Nayberg, " About Microsoft TerraServer – Backup and Restore Hardware and Software", Microsoft Corp., Redmond, WA, http://terraserver-usa.com/about.aspx?n=AboutQaBackup

[Barclay02] T. Barclay; J. Gray; E. Strand; S. Ekblad; J. Richter "TerraService.NET: An Introduction to Web Services", June 2002, Microsoft Corp, Redmond, WA. http://research.microsoft.com/research/pubs/view.aspx?tr_id=588

[Barclay03] T. Barclay, W. Chong, J. Gray, "A Quick Look at Serial ATA (SATA) Disk Performance", October 2003, Microsoft Corp., Redmond, WA http://research.microsoft.com/research/pubs/view.aspx?tr_id=xxxx

[Barclay04] T. Barclay, J. Gray, "TerraServer SAN and Cluster Experience", July 2004, Microsoft Corp. Redmond, WA

[Davis94] F. Davis, W. Farrell, J.Gray, R. Mechoso, R. Moore, S. Sides, M. Stonebraker., "EOSDIS Alternative Architecture Final Report," Sept., 1994, http://research.microsoft.com/~gray/EOS_DIS/

[Devlin] B. Devlin; J. Gray; B. Laing; G. Spix, "Scalability Terminology: Farms, Clones, Partitions, and Packs: RACS and RAPS", MSR-TR-99-85, http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-99-85

[GEOSTOP03] US Geological Survey, Reston, VA. http://www.geo-one-stop.gov

[Jain03] S. Jain, T. Barclay: "Adding the EPSG:4326 Geographic Longitude-Latitude Projection to TerraServer", August 2003, Microsoft Corp., Redmond, WA. http://research.microsoft.com/research/pubs/view.aspx?tr_id=675

[Kobler95] B. Kobler, J. Berbert, P. Caulk, P. C. Hariharan: "Architecture and Design of Storage and Data Management for the NASA Earth Observing System Data and Information System (EOSDIS)". IEEE Symposium on Mass Storage Systems 1995: 65-76

[Moore] L. Moore, "Transverse Mercator Projections and U.S. Geological Survey Digital Products", U.S. Geological Survey, Professional Paper.

[MSCS03] Microsoft Corp., Redmond, WA. http://www.microsoft.com/windowsserver2003/evaluation/overview/technologies/clustering.mspx

[NATLMAP03] US Geological Survey, Reston, VA. http://nationalmap.usgs.gov

[Robinson95] A. H. Robinson, J. L. Morrison, P. C. Muehrcke, A. J. Kimerling, S. C. Guptill, *Elements of Cartography, Sixth Edition,* John Wiley & Sons, Inc., U.S.A. 1995, ISBN 0-471-55579-7.

[Samet90] H. Samet, *The Design and Analysis of Spatial Data Structures,* Addison-Wesley, Reading, MA, 1990. ISBN 0-201-50255-0.

[ServTech] Server Technology, Inc., Reno, NV, http://www.servertech.com

[SilMech] Silicon Mechanics, Seattle WA, http://www.siliconmechanics.com

[Snyder89] Snyder, J.P., "An Album of Map Projections", U.S. Geological Survey, Professional Paper, 1453, (1989).

[SQL Server] Microsoft SQL Server 7.0 http://microsoft.com/SQL/

[VER01] Veritas NetBackup Data-Center Edition, http://www.veritas.com/news/press/PressReleaseDetail.jhtml?NewsId=9445