Microsoft TerraServer: A Spatial Data Warehouse

Tom Barclay
Microsoft Research
301 Howard St., Suite 830
San Francisco, CA 94105
415 778 8223
tbarclay@microsoft.com

Jim Gray
Microsoft Research
301 Howard St., Suite 830
San Francisco, CA 94105
415 778 8222
gray@microsoft.com

Don Slutz
Microsoft Research
301 Howard St., Suite 830
San Francisco, CA 94105
415 778 8226
dslutz@microsoft.com

ABSTRACT

Microsoft® TerraServer stores aerial, satellite, and topographic images of the earth in a SQL database available via the Internet. It is the world's largest online atlas, combining eight terabytes of image data from the United States Geological Survey (USGS) and SPIN-2. Internet browsers provide intuitive spatial and text interfaces to the data. Users need no special hardware, software, or knowledge to locate and browse imagery. This paper describes how terabytes of "Internet unfriendly" geo-spatial images were scrubbed and edited into hundreds of millions of "Internet friendly" image tiles and loaded into a SQL data warehouse. All meta-data and imagery are stored in the SQL database. TerraServer demonstrates that general-purpose relational database technology can manage large scale image repositories, and shows that web browsers can be a good geo-spatial image presentation system.

Keywords

Geo-spatial, VLDB, image databases, internet.

1. Overview

The TerraServer is the world's largest public repository of high-resolution aerial, satellite, and topographic data. It is designed to be accessed by thousands of simultaneous users using Internet protocols via standard web browsers. TerraServer is an image "tile" server that delivers a set of raster images based on a users search criteria. Once an image of interest is located, users can pan, zoom in, zoom out, or display meta-data about the image they are viewing.

The TerraServer is a multi-media data warehouse. It differs from a traditional data warehouse in several ways: (1) it is accessed by millions of users, (2) the users extract relatively few records (thousands) in a particular session and, (3) the records are relatively large (10 kilobytes). By contrast, classic data warehouses are (1) accessed by a few hundred users via proprietary interfaces, (2) queries examine millions of records, to discover trends or anomalies, (3) the records themselves are generally less than a kilobyte. In addition, classic data warehouse queries may run for days before delivering results. Initial results typically cause users to modify and re-run queries to further refine results.

One thing the TerraServer has in common with classic data warehouses is that both manage huge databases: several terabytes of data. TerraServer's topographic maps cover all of the United States at 2 meter resolution 10 million square kilometers), the aerial photos cover 40% of the United States today (3 million square kilometers) at one-meter resolution, and 1% of the urban areas outside the United States (1 million square kilometers) at 2 meter resolution.

This report describes the design of the TerraServer and its operation over the last 18 months. It also summarizes what we have learned from building and operating the TerraServer.

Our research group explores scaleable servers. We wanted first-hand experience building and operating a large Internet server with a large database and heavy web traffic. To generate the traffic we needed to build an application that would be interesting to millions of web users.

Based on our exposure to the EOS/DIS project [2], we settled on building a web site that serves aerial, satellite, and topographic imagery. We picked this application for four reasons:

1. The web is inherently a graphical environment, and these images of neighborhoods are recognizable and interesting throughout the world.
2. We believed this application would generate the billions of web hits needed to test our scalability ideas.
3. The data was available. The USGS was cooperative, an since the cold war had ended, other agencies were more able to share satellite image data. The thaw relaxed regulations that had previously limited the access to high-resolution imagery on a global basis.
4. The solution as we defined it – a wide-area, client/server imagery database application stored in a commercially available SQL database system – had not been attempted before. Indeed, many people felt it was impossible without using an object-oriented or object-relational system.

This paper describes the application design, database design, hardware architecture, and operational experience of the TerraServer. The TerraServer has been operating for eighteen months now. We have deployed the third redesign of the database, user interface, and process of adding new images to the database.

2. Application Design

TerraServer is accessed via the Internet through any graphical web browser. Users can zoom and pan across a mosaic of tiles within a TerraServer scene. The user interface is designed to function adequately over low-speed (28.8kbps) connections. Any modern PC, MAC, or UNIX workstation can access the TerraServer using a standard web browser – Internet Explorer 3.0 or later, or Netscape Navigator 3.0 or later. If you have never used it, look at the TerraServer web site at http://terraserver.microsoft.com/.

There are four methods by which a user locates an image:

(1) **Coverage Map:** clicking on low resolution map of the world shaded to show where coverage exists,

(2) **Place Query:** entering a place name, e.g. San Francisco,

(3) **Coordinate Query:** entering the longitude and latitude of interest, or

(4) **Famous Place:** selecting a location from a pre-compiled list of places.

A successful search presents the user with a web page containing an HTML table of image tiles fetched from a SQL database. GIF images surrounding the image tile table provide the user with the following basic controls:

- Pan and zoom
- Display image meta data
- Download an image copy
- Control the size of the image table. There are three fixed sizes – Small (400 x 200 pixels), Medium (600 x 400 pixels), and Large (800 x 600 pixels).
- Choose the image "style" or theme. TerraServer stores three image styles -- imagery stored in TerraServer – shaded relief, topographic map, and photograph (aerial or satellite).

We expect and support the use of TerraServer image tiles on remote web sites. Most data on TerraServer is public domain data. Therefore, we deliberately chose simple graphics and storage methods so that users could craft their own web pages that display TerraServer image tiles.

Imagery is categorized into "themes" by data source, projection system, and image "style". Currently, there are four data themes:

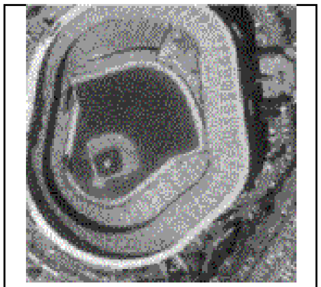

Figure 1. A USGS DOQ Image of 3Com Park near San Francisco

USGS Digital Ortho-Quadrangles (DOQ) are grayscale or color infrared, 1-meter resolution aerial photos. Cars can be seen, but 1-meter resolution is too coarse to show people. Imagery is ortho-rectified to 1-meter square pixels. Approximately 50% of the U.S. has been digitized. The conterminous U.S. is expected to be completed by the end of 2001. Some locations have more than one DOQQ image available varying by image source date or color mode. TerraServer stores the latest grayscale image. If only a color infrared image is available, they it is converted to grayscale before tiling and storing in the database.

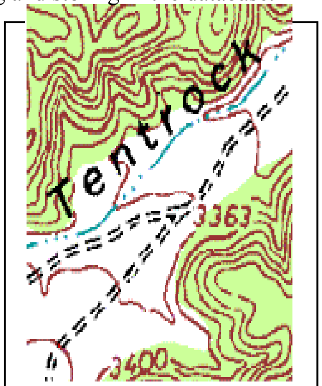

Figure 2. A USGS DRG 2-meter resolution image.

USGS Digital Raster Graphics (DRG) DRGs are the digitized versions of the popular USGS topographic maps. The complete set of USGS topographic maps have been scanned for the conterminous United States and Hawaii. The original images are available in three map scales – 24,000:1 (2.4 meters/pixel), 100,000:1 (10 meters per pixel) and 250,000 meters per pixel. The raster images are re-sampled to nearest power of 2 meters per pixel.

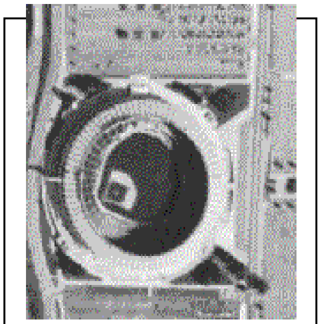

Figure 3. a SPIN-2: 1.56-meter image of Atlanta's Fulton County Stadium.

Aerial Images SPIN-2™ are grayscale 1.56-meter resolution de-classified Russian military satellite images. The images are re-sampled to 2-meter resolution. Terra-Server contains SPIN-2 images of Western Europe, the United States, and the Far East. Unfortunately, there is little coverage of Canada, South America, Africa, and Southeast Asia. The SPIN-2 imagery is rectified, i.e. rotated so north is up, but is not ortho-rectified. That is, the image is not "rubber sheeted" so that each pixel covers a consistent square number of square meters. However, given the height of the satellite, the difference in ground area between individual pixels is small.

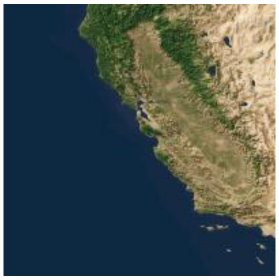

Figure 4: Encarta Virtual Globe shaded relief image of California, 8km / pixel.

Encarta Shaded Relief is natural color, shaded relief map of the globe. The full resolution image detail is approximately 1 kilometer per pixel. The image is a seamless image of the globe between latitude +80º and -80º. The Microsoft Geography Business Unit assembled the image from a public domain combination of weather satellite data and elevation data. The image appears in the Encarta Virtual Globe add-on product to the Encarta Encyclopedia CD title.

2.1 System Architecture

TerraServer is a “thin-client / fat-server” design. The TerraServer has a 3-tier architecture:

Tier 1: ***The Client*** is a graphical web browser or other hardware/software system that supports HTTP 1.1 protocols and HTML 3.2 document structure. TerraServer is built and tested with Netscape Navigator and Internet Explorer on Windows, MacOS, and UNIX.

Tier 2: ***The Application Logic*** is a web server application that responds to HTTP requests submitted by clients by interacting with the Tier 3 database system and applying logic to the results returned.

Tier 3: ***The Database System*** is a SQL Server 7.0 Relational DBMS containing **all image** and meta-data required by the Application Logic tier.

Most web pages are dynamically formed and contain standard HTML 3.2 tags. All imagery is stored within SQL Server “blob fields” and compressed in Jpeg or Gif format. There is one row per image tile. The row contains the meta-data describing the tile’s geographic attributes and one blob field containing the image data. Imagery is presented via <IMG SRC=…> tags without the aid of java applets, or other specialized client side controls. The SRC URL references a script executed on the web server that is fetched from the database and sent back to the browser prefixed with the appropriate mime type. TerraServer supports Netscape Navigator and Internet Explorer V3.0 or later browsers.

“Zoomed out” or sub-sampled imagery is also stored in the database, one-row per tile. Four higher resolution tiles are sub-sampled into one lower resolution tile. The process is repeated for the number of levels in the image hierarchy. We found, and our graphics colleagues have confirmed, that a 7 level image pyramid is the maximum for grayscale imagery. All levels of the image pyramid are pre-computed and stored in the database for the following reasons:

1. We wanted to build the largest physical database that we could.
2. A 7 level image pyramid would require 25,600 tiles to be sub-sampled to create one single 64:1 resolution tile. We do not believe users are willing to wait for this operation to be completed “on-the-fly”.
3. We did not have the resources to develop and support a high performance, server-side sub-sampling and dynamic image generation application.[1]
4. We wanted users to reference TerraServer imagery on their own web pages with a simple <IMG SRC…> tag and not require a client-side control to display and sub-sample the imagery.

The web site is a cluster of machines. A set of servers executes Microsoft Internet Information Server (IIS) web server software that interfaces with the SQL Server databases. The site is designed to support a variable number of web servers for performance (more net cards for increased bandwidth) and reliability reasons (a failed web server does not take down the whole web site). Increasing the throughput of the web site is as simple as adding another web server until the network web and database servers saturate.

The web servers connect to the database servers that host the SQL Server database via a separate internal network. This protects the SQL Servers provides an extra level of security from hackers and a separate private network that does not compete with Internet or other network traffic. Currently, there are two database servers – one for aerial and satellite imagery, and a second for the topographic maps. There are also two 100mbit subnets between the database and web servers.

The number and size of database servers is determined by the popularity of the data and convenience. The bandwidth between database servers and web servers drives the hardware configuration. Topo maps are expected to be popular, so we decided to host them on a separate machine where queries for aerial photography data do not have to compete for bandwidth with queries for topo data. The shaded relief data is small and replicated on both the topo and aerial photography server.

Web pages containing imagery have a consistent layout. Users can control the number of image tiles that appear on a single page. The user’s monitor size and Internet connection speed dictate their choice. Web pages are dynamically created on the web servers due to the millions of combinations of possible web pages.

There are a wide number of choices for dynamic web page construction on Internet Information Server based web servers – ISAPI, Active Server Pages, CGI/Perl, Cold Fusion, etc. We chose Active Server Pages (ASP) for a number of reasons:

- Fast and easy development – ASP host Visual Basic or JavaScript engines. An ASP document can be written and debugged without requiring a compiler or other elaborate development tools.
- The execution time of our ASP scripts was dominated by the SQL execution time and the data transfer time between database and web server. There was little or no performance gain in using a compiled language.[2]

We chose Visual Basic as the scripting language because it had better support for error handling. The Active Data Object (ADO), an OLEDB object, is used to access the SQL Server database engine. The Visual Basic error object could trap the errors raised by the ADO object. Our Visual Basic scripts process URL query strings, access the Imagery SQL database, and dynamically generate HTML pages. One ASP page, Tile.asp, is responsible for retrieving Jpeg or GIF formatted blobs from the database and sending it back to the client. A second ASP page, image.asp, is responsible for executing SQL queries to fetch the meta data necessary to dynamically produce the HTML tags which format an image page. These two web scripts are called 85% of the time.

The Cmap.asp (coverage map), Place.asp, Geo.asp, and Famous.asp implement the four search methods described

[1] The entire TerraServer web and database application was developed by one person.

[2] Originally we built the web application in C and accessed the web server via the ISAPI interface. We abandoned this approach after determining there was no performance gain in our case and a substantial increase in development cost.

previously. Each major function, e.g. Download, Image Information, the Home Page, etc., is implemented by a separate ASP page.

All TerraServer ASP scripts have a common structure. Database access is performed by calling a single SQL Server stored procedure function. The SQL stored procedure returns one or more record sets. The ASP script calls ADO methods to connect to the database server, call one stored procedure, iterate through the returned result set(s), and disconnect. This design achieves the highest performance as we learned during the first few weeks of service.

2.2 Tuning the Application

TerraServer was our group's very first web site. While we had some professional graphics design assistance, we developed the web application by the classic seat-of-the-pants method. Also like most software projects, particularly Internet projects, we were under marketing pressure to release to the web quickly. We learned a lot about our design and products we chose during an all too brief beta period and during the first month of live service.

We initially estimated the application was interesting enough to generate 1 million hits or 250,000 page views a day. Later we increased our estimate to 5 million hits and 1 million page views a day. We configured 4 web servers to support the 5 million hits and 1 million page views per day estimate. Officially, TerraServer went live on June 24, 1998. However, there was an article published on the front of the USA Today Business page on June 22, 1998. The article proved we grossly underestimated the popularity of the web site.

Starting on June 22, our four web servers managed to deliver 35 million web hits and 8 million page views. Millions more were rejected. We quickly grew our site to 10 web servers by the weekend and learned the following:

1. Web server software is really a TP Monitor. Once we realized this point, we used the tuning skills we learned back in the late 70s and 80s to good use. We treated the database server as a scarce resource and used the web server configuration tools to optimally schedule requests to the back end. Prior to this discovery, we unleashed requests from the web servers to the backend via a "fire hose" and were genuinely surprised when the database server ground to a halt.
2. Round trips to the database server are costly. Therefore, do as much as possible in one trip.
3. People look at imagery of where they live. While spending many a sleepless night the first week, we noticed that there was the interesting "sine wave" of Internet connection and disk activity. In highly populated and covered areas, we would notice a precipitous rise in user connections at the start of that time zone's day. Between 5 am and 6am PST or 8 am and 9am EST, the number of user connections would rise steeply. About one hour later, the number of connections continued to rise, but the disk activity began to drop and reach a steady state. Over time, we realized that separate users were requesting the same data as their neighbors. We had 2 GB of physical memory on the database server, about 1.8 GB was SQL Server's memory cache. Thus many of the queries were resolved out of the database cache.

   Over time, we realized the TerraServer web site is busiest in the mornings where we have coverage. Thus our web site is very active from 11pm to 3 am (Europe) and from 5 am to 3 or 4 pm. But it is not very busy around 5 pm because we have very little coverage in the Pacific Rim and East Asia.
4. Our Microsoft.com and msn.com colleagues confirmed some other web usage facts. The internet is busiest on Mondays and Tuesdays. Saturday and Sunday is half the volume of Monday and Tuesday. A steady slide occurs from Wednesday thru Friday. Thus, we do on-line database maintenance on the weekends – on-line backups, table reconfigurations, etc.

2.3 Scenes and Projection Systems

TerraServer is map and image data tiling system. Unlike online mapping web sites, e.g. MapQuest, TerraServer does not re-project the data to match the user's request. Instead, TerraServer displays the image or topographical map data in the projection chosen by the data provider.

TerraServer allows a user to navigate the length and width of an entire scene. A web page contains tiles from only one scene. Lists of links to scenes that overlap the viewed scene are offered to the user. Thus TerraServer really is a collection of seamless scenes and not a single seamless view of earth.

The reason for this is geometry and geography. The earth is a bumpy ellipsoid. Maps and computer monitors are flat. It is impossible to accurately present a spherical object on a flat surface.

Cartographers have addressed this issue by developing projections of the geoid onto flat surfaces [5]. There are many projection systems, each suited to present certain regions or properties. Multiple images in a projection system can often be joined together to form a seamless mosaic within certain boundary conditions. These mosaics either have extreme distortion as they scale out, or they introduce seams.

DOQ and DRG data are projected by the USGS into Universal Transverse Mercator (UTM) projection using the North American Datum (NAD) ellipsoid created in 1983 [7]. UTM is a projection system that divides the earth into 60 wedge shaped *zones* numbered 1 thru 60 beginning at the International Date Line. Each zone is 6 degrees wide and goes from the equator to the poles. UTM grid coordinates are specified as zone number, then meters from the equator and from the zone meridian[3].

The conterminous United States is divided into 10 zones (see Figure 5). Each of these UTM zones is a *scene*. The TerraServer mosaics each scene, but two adjacent scenes are not mosaiced together. Users can pan and zoom within a scene, and can jump from one scene to another.

[3] Actually, UTM grid units can be in inches, feet, meters, or kilometers. The USGS chose meters for most of their assets in the UTM projection. UTM is not used above 80N or 70S [5].

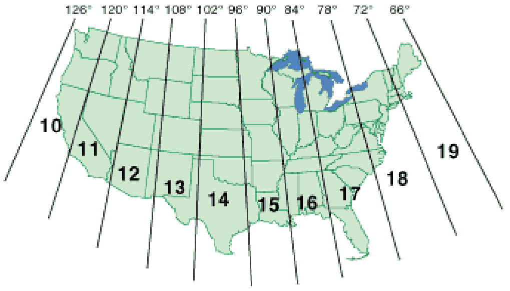

Figure 5: The ten UTM zones in the continental United States.

The SPIN-2 imagery is digitized from Russian satellite photographs. The Russian satellite captures 160km wide by 40km high areas in a single image. The satellite takes one image and then begins the adjacent image, overlapping the last image. The overlap is variable, and when digitized does not line up on a pixel boundary.

To create a seamless mosaic of SPIN-2 imagery, all SPIN-2 imagery would have to be ortho-rectified. This requires precise geo-location of each image, which was not available due to security concerns. Without rectification, if tiles extracted from separate SPIN-2 satellite images are mosaiced, the tile edges are misaligned. Roads, rivers, and other geographic features do not line up. While this may be understandable to GIS experts, it is disorienting and unacceptable to novice users.

Consequently, the TerraServer treats each 160km x 40km SPIN2 image as a separate scene. These scenes are not mosaiced together. Users can pan and zoom within a scene, and can jump from one scene to another.

2.4 TerraServer Grid System

Users can zoom and pan across a mosaic of tiles within a TerraServer scene. The tiles are organized in the database by theme, resolution, scene, and location within a scene in the TerraServer grid system.

TerraServer supports a fixed number of resolutions in powers of 2 from 1/1024 meters per pixel (scale 0) through 16384 meter per pixel (scale 24). One-meter per pixel is scale 10.

For UTM projection data sets, the SceneID is the UTM zone assigned to the original image a tile's pixels were extracted from. For SPIN2 data sets, a unique SceneID is assigned for each scene loaded per theme.

Each TerraServer scene is planar. A tile can be identified by its position in the scene. The tile loading program assigns a relative X and Y tile identifier to each tile as it is loaded.

For UTM projected data, the X and Y tile address is the UTM coordinate of the top-left most pixel in the tile divided by the tile image size in UTM units in meters. The following are the formulas:

$$X = TopLeftUTM\_X / (TilePixWidth \bullet Resolution)$$
$$Y = TopLeftUTM\_Y / (TilePixHeight \bullet Resolution)$$

For SPIN2 scenes, the X and Y tile addresses are relative to the upper left corner of the scene.

The six fields – Resolution, Theme, SceneID, Scale, X, and, Y - form the unique key by which any TerraServer image tile can be directly addressed. Each TerraServer web page contains image tiles from a single Theme, Scale, and SceneID combination. For example, our office building in USGS DOQ theme (T=1), has scene UTM zone 10 (S=10), at scale 1 meter (Z=10) with X=2766 and Y=20913. The URL is: http://terraserver.microsoft.com/tile.asp?S=10&T=1&Z=10&X=2766&Y=20913.

The TerraServer search system performs the conversion from geographic coordinate systems to the TerraServer coordinate system. The TerraServer image display system uses TerraServer grid system coordinates to pan and zoom between tiles and resolutions of the same theme and scene.

2.5 Imagery Database Schema

Each theme has an Source Meta-data table. This table has a row for each image that is tiled and loaded into the TerraServer database. The *OrigMetaTag* field is the primary key. The meta-fields vary widely from theme to theme. Some of the meta fields are displayed by the Image Info Active Server Page (for example http://terraserver.microsoft.com/imageinfo.asp?S=17&T=2&X=17&Y=122&Z=17&W=1&O=c28080a1&P=28+km+SW+of+Orlando%2C+Florida%2C+United+States

All the image tiles and their metadata are stored in an SQL database. One table is maintained for each (theme, resolution) pair so that tiles are clustered together for better locality. USGS DOQ supports resolutions from 1-meter resolution through 64-meter resolution. USGS DRG data supports 2-meter resolution through 128-meter resolution. SPIN supports resolutions from 1-meter to 64-meter.

Each theme table has the same five-part primary key:

- *SceneID* –individual scene identifier
- *X* – tile's relative position on the X-axis
- *Y* – tile's relative position on the Y-axis
- *DisplayStatus* – Controls display of an image tile
- *OrigMetaTag* – image the tile was extracted from

There are 28 other fields that describe the geo-spatial coordinates for the image and other properties. One field is a "blob type" that contains the compressed image.

These tile blobs are chosen to be about ten kilobytes so that they can be quickly downloaded via a standard modem (within three seconds via a 28.8 modem).

2.6 Gazetteer Database Schema

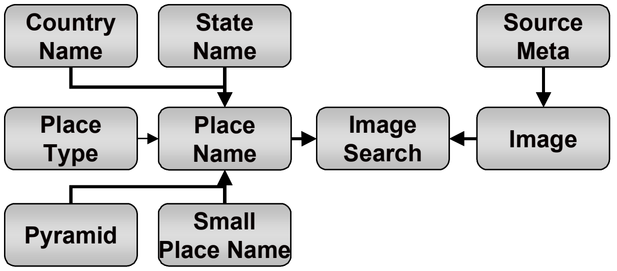

Figure 6: Gazetteer and Image Schema

The Gazetteer lets users find images by name. It contains the names for about 1.5 million places, with many alternate spellings. It is a simplified version of the Gazetteer found in the Encarta Virtual Globe™ and Microsoft Streets™ products.

The Gazetteer Schema is a snowflake database design. *PlaceName* is the center table. It contains the formal name for unique location on earth and maps a unique named location to the TerraServer Grid System. The *AltPlace* table contains all the synonyms of a unique place. The *StateName* and *CountryName* parent tables identify a place's state/province and country. The *AltState* and *AltCountry* tables contain the state/province and country synonyms.

Lookup by place name is surprisingly common (40%). So the user interface was modified to make it even easier. The top of each web page has a simple name lookup field where the user can enter city, state/province, or country separated by commas. The home page has an additional link that takes the user to an "advanced" name lookup web page.

The *find a place* input field allows the user to enter a subset of *PlaceName*, *StateName*, and *CountryName*. The supporting database stored procedure builds a cursor that searches for the name by performing a join on the appropriate tables depending on which fields the user specified. Name searches are performed on the "Alt" tables, which have synonyms and abbreviations for places (USA for example). Formal names matching the search criteria are returned from the *PlaceName, StateName,* and *CountryName* tables.

The *ImageSearch* table forms the association between a named place and a visible image. The *ImageSearch* table identifies the Theme, *SceneID, Scale, X, Y*, and *ImageDate* of a visible image tiles that cover the associated kilometer square cell. The load program inserts rows into the *ImageSearch* table when it has completed the image pyramid for a certain area. The *ImageSearch* table serves as a one-level quad-tree index of the image data [6].

The image display Active Server Page scripts use an additional table, the *Pyramid* table, to display the name and distance to the location closest to the center tile on an image display web page. This table is a two-level quad-tree is used to find population-weighted nearest neighbors of a given latitude and longitude. The SQL stored procedure scans a rectangle of the quad-tree to determine the closest city to latitude-longitude in the center of the web page image. The quad tree is implemented atop a B-tree by giving each quadrangle a name that is a prefix of the key for records in that quadrangle.

In total, the Gazetteer contains about 4 million rows and consumes 3.3 GB of space. Our first design used a fine-granularity (quarter kilometer) quad-tree and so used a hundred times more space (400GB). That design needed no computation during lookup. The current design must examine 50 records on average and do some spherical trigonometry on the coordinates for each record. The new design uses more computation, but it can process a record in 3 microseconds of processor time, so it seems a good tradeoff.

2.7 Database Architecture

The database architecture was chosen to demonstrate the scalability and usability of SQL Server—everything was done in the most straightforward way, with no special tricks. There are two database servers – Imagery and Topographic Map. On the Imagery database server, a single SQL server database was created with two File Groups – Gazetteer and Primary.[4] The Gazetteer File Group is comprised of one, 5 GB file named t2b2gaz0.ndf and placed on volume "G:". The Imagery, Image Search, Load Management, and all other tables are stored in the Primary File Group consisting of many NTFS files. Each file resided on one of the four logical volumes and was 20GB, which is a convenient backup/recovery unit. Initially, 53 files were created to achieve the 1TB database goal. Additional files are added as new imagery is loaded. Currently, there are 71 20GB files. Plans are to grow the database to 2.0 TB. The initial files were placed on two 595GB NT stripe-set volumes and the files added later were placed on two other similar volumes. SQL Server makes all allocation and placement decisions within and among the files.

The Topographical Map database is similarly configured. There are two File Groups – Gazetteer and Primary. The Gazetteer data is replicated from the Imagery database server. There are 42 20 GB files spread over two 559 GB NTFS volumes.

The TerraServer database was created using default settings with two exceptions. A *bulk copy* option was set to improve load times by reducing logging. Also, a *truncate log on checkpoint* option was set. These options preclude media recovery using the log. Instead, Terraserver would restore from an online database backup and reload any data that had been added since that backup.

All tables are clustered on their primary key and a few secondary indexes, mostly in the Gazetteer, were added to support searching for different name combinations and for on-line loading. Retrieving one image tile requires the simplest of SQL statements:

```
Select * from Image where PrimaryKey='value'
```

One set of Gazetteer tables and one *ImageSearch* table serve to locate images by name in all themes. The *Loadjob*, *Scalejob,* and *SearchJob* tables are used to manage the online loading of images. They hold the state of load jobs and are used for monitoring and restart.

2.8 Hardware Architecture

The web site is configured to minimize single points of failure, protect the database from hackers, and scale to support additional users or data over time.

The Tier 2 and Tier 3 software runs on separate computer systems. There is an HTTP firewall in front of the web servers and a packet filter firewall between the web servers and the database server. Having the database server inside the corporate firewall allows us to load data to the TerraServer from within the Microsoft corporate network.

The web site has eight Windows NT servers – 6 web servers and 2 database servers. The USGS aerial imagery is maintained on a Compaq AlphaServer™ 8400 containing 8 440 Mhz Alpha processors and 10 GB of RAM. The processor is attached to 7 StorageWorks™ Enterprise Storage Array 10000 (ESA-10000) cabinets. The disk arrays are based on UltraSCSI technology.

Each ESA-10000 contains 48 9 GB disk drives and 2 HSZ70 dual-redundant RAID-5 controllers. 4 sets of 11 disks each are

[4] SQL Server 7.0 supports a new concept called "File Groups", which replace the previous "Database Device" concept. A File Group is a named entity which lists the physical files that store a specific list of tables in a database.

configured into a single RAID-5 stripe-set and managed as a single logical disk by the HSZ70 controller. 2 drives per cabinet are available as hot spares. Should a disk fail, the HSZ70 controllers automatically swap a spare drive into a RAID set.

Windows NT Server sees each large (85 GB each) disk created by the RAID controllers of each of the seven disk cabinets. It stripes these into 4 large (595 GB) volumes which are then each formatted and managed by the Windows NT file system (NTFS). Each 595 GB volume contains about thirty 20GB files. SQL Server stores its databases in these large files. We chose this 20GB file size since it fits conveniently on one DLT magnetic tape cartridge.

Connected to the AlphaServer 8400 is a StorageTek 9710 automated tape robot. The tape robot contains 10 Quantum DLT7000 tape drives. Legato Networker backup software can backup the entire 1.5 TB TerraServer SQL database to the StorageTek tape robot in 7 hours and 15 minutes – or 17 GB/hour.

The USGS topographic maps are maintained on a Compaq ProLiant 8500 containing 8 550 mhz Pentium III processors and 4 GB of RAM. Two racks contain 140 9GB hard drives. A row of 10 drives is configured into a RAID-5 disk volume by the Compaq SmartArray hardware controller. The RAID-5 sets are connected to the processor via Fiber Channel.

For reliability and performance, the database servers contain three 100 Mbit Ethernet cards and is connected to three separate local area networks. One network card connects the database server to three of the Compaq ProLiant 5500 web servers. A second network card connects the database server to three other Compaq ProLiant 5500 web servers. The third network card connects the database server to the TerraServer image-processing center which is inside the Microsoft corporate network.

The TerraServer web site is housed at the Microsoft Internet Data Center – a well-managed and secure facility with excellent environmental protection (emergency power, good physical security,...), and with high bandwidth to the Internet (about 3.6 Tbps at present).

2.9 Hardware Capacity Planning

It is difficult to size an Internet application in advance. We originally planned for one-million web hits per day, which is far beyond what we actually expected. At the time, other groups were reporting small numbers (e.g. 17 million hits per week for the 1997 winter Olympics.) But, publicity and interest in the site was very high. During the first week, demand was in excess of 30 million web hits per day. Ten times what we expected. This was not a pleasant experience for us or for our users.

Now that the novelty has worn off, demand averages 7 million hits per day with peaks of 15. The web site is configured to support a maximum of 6,000 simultaneous web browser connections (see Table 1). Additional Tier 2 Web Servers could increase this number.

Table 1. TerraServer hardware configuration parameters

Max hits per day	40 million/day
Max SQL queries per day	37 million/day
Max image downloads / day	35 million/day
Bandwidth to Internet	200 Mbps = 2 Terabytes/day
Concurrent connections	6,000 connections
Web front ends	6 4-way 200 Mhz Compaq Proliant 5500, .5GB ram
Database back-end	1 8-way 440Mhz Compaq AlphaServer 8400 10GB ram, 3.2 TB raid5 324 9GB Ultra SCSI disks 1 8-way 550 mhz Compaq ProLiant 8500, 4GB ram, 1.2 TB Raid-5 (140 9GB Fiber Channel disks

3. TerraServer Data Load Process

As with other data warehouses, most of the labor of building the TerraServer consists of data scrubbing and data loading. The TerraServer database is organized to simplify the TerraServer web application that presents image and meta data to end users. The TerraServer design avoids dynamic projection, rotation, and other sophisticated features found in commercial GIS systems. The data loading programs pre-compute the GIS details and present each scene as a seamless mosaic of 200 by 200 pixel tiles. All knowledge of projection systems, re-sampling pixels, edge alignment, merging pixels from multiple images, etc., is implemented in the load programs.

There are two image load programs in the TerraServer system – TerraCutter and TerraScale. *TerraCutter* re-formats imagery received from data sources, tiles it into formats acceptable to the TerraServer web application, and inserts the tiles and metadata into the database. *TerraScale* computes the lower resolution image-pyramid tiles for a theme by sub-sampling the tiles created by TerraCutter.

We implemented a simple job-scheduler system to manage and track the data loading process. Each processing program leaves a "popcorn trail" in the Load Management database so administrators can monitor progress on loading new data.

New imagery is inserted into the TerraServer database on-line while web users browse imagery. The table design and load program insertion order ensures that all the required metadata and imagery is place before the image is made visible to the web application.

3.1 Data Flow

USGS DOQ data is shipped DLT media written in the "tar" format. DOQ files are in a custom USGS format. Meta-data and image pixels are contained in one file. Data is 8-bit grayscale or 24-bit, RGB color infrared. TerraCutter converts color infrared to 8-bit grayscale. DOQ files cover a USGS "standard quarter-quadrangle", which is a 3.75-minute by 3.75-minute square area.

The order of DOQ files on tape is random. Adjacent DOQ files can arrive in any order.

USGS DRG data is shipped on CDROM media. All 1:24,000, 1:100,000, and 1:250,000 scale maps for a square degree are contained on one CDROM. Images are in the GeoTiff format and have a fixed color map of 13 colors. Meta-data and image pixels are separate files. Files for a single square degree are grouped onto one CDROM.

SPIN-2 data is shipped on DLT media written in "NT Backup" format. SPIN-2 files are in a custom "Kodak/Microsoft/Aerial Images" format. Meta-data and image pixels are in separate files. Data is 8-bit grayscale.

TerraServer System Administrators use the appropriate "off-the-shelf" program to download a tape or CDROM to a directory on an image editing system. The current image editing system is a Compaq ProLiant 8500 with 8 550 mhz processors, 4 GB of RAM, and 450 GB of disk.

The TerraServer System Administrators launch the TerraCutter image-editing program against a directory containing the image and meta files downloaded from tape or CDROM. TerraCutter uses the Load Management schema tables to make sure the job has not been processed previously. Or, if a previous run had aborted, TerraCutter will pick up where it had left off. TerraCutter uses the Load Management schema to catch duplicate files sent on previously processed tapes or CDROMs. When a directory has been successfully processed, the download directory is deleted; the tape is physically marked as "processed" and shelved. All further processing – sub-sampling to create lower resolution scales, correlating tiles with named locations, merging pixels between tiles, etc. – occurs within the memory of a custom program or T-SQL database statements.

3.2 Load Management Schema

The TerraServer load system maintains a set of tables in the TerraServer database. These tables are not visible to end users on the Internet. A set of Active Server Page scripts allow the TerraServer System Administrators to schedule and monitor the TerraServer database load process.

A *LoadJob* row is created when a load program is instructed to process a directory or a specific list of imagery received from a data source. The *LoadJob* row describes the on-disk location of the input data, the source tape/CD, the computer system the load program ran on, the load program version, the date the job started, and the job's current status.

Load programs update the *LoadJob* record each time they complete an input file found in the source path and insert a row into the *ScaleJob* table. This is the signal to the TerraScale program that a block of image tiles is ready to have its image pyramid created.

The TerraScale program updates the *ScaleJob* table with its progress information. There is a set of administrative Active Server Pages that TerraServer Administrators use to monitor the progress of image pyramid creation.

3.3 TerraCutter

TerraCutter is a fairly complicated C program. The simple part is formatting tiles suitable for the TerraServer web application and inserting them into the database.

The ground size covered by a pixel must also be fixed to multiples of 1-meter resolution –¼, ½, 1, 2, 4, 8, 16, etc. If necessary, TerraCutter re-samples the input image to the appropriate resolution as the image is read in. As tiles are produced, TerraCutter saves the tile image into a temporary file, computes the Image table metadata fields, and inserts the new tile into the database using ODBC API calls. A single image tile is inserted in the scope of one transaction.

The challenging aspect to the TerraCutter program is handling the image overlap that generally exists within original imagery. For example, USGS DOQ images cover a 3.75 minutes by 3.75 minutes. Generally, there is a 200 to 300 meter / pixel collar surrounding the 3.75 x 3.75 minute rectangle. Thus adjacent DOQ images contain a certain number of duplicate pixels along an edge.

TerraCutter combines pixels from multiple input images into one tile. The merge must ensure geographic alignment so that roads, buildings and other structures that cross tile boundaries do not appear interrupted. TerraCutter does this by carefully computing the starting point - location 0,0 in the image tile. For UTM based data-sets (USGS DOQ and DRG), TerraCutter looks for the first pixel in the input image that has a UTM X and Y address that is evenly divisible by width and height of an image tile. For example, USGS DOQ images are 1-meter resolution, so DOQ tiles start at 200-meter boundaries. DRG images are 2-meter resolution, so DRG tiles start at 400-meter boundaries.

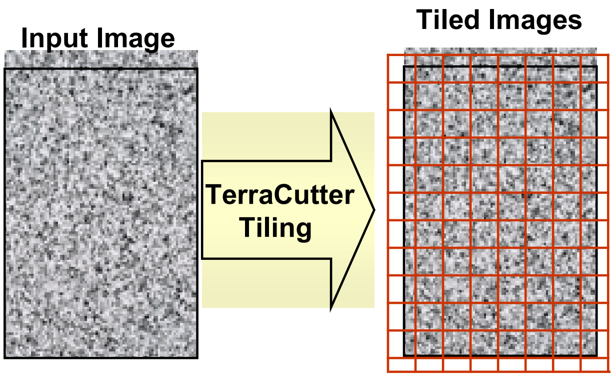

Figure 7: The TerraCutter breaks a large image up into tiles that then can be mosaiced into a scene.

Rounding the starting UTM X and Y coordinate up to width and height of the image tile simplifies aligning layered maps containing multiple TerraServer data-sets. The UTM address for pixel 0,0 in a DOQ Image Tile at 2-meter resolution is the same UTM address for pixel 0,0 in a DRG tile with the same X, Y, zone address.

Input image files of projected data-sets, like USGS DOQ and DRG, will overlap other image files along the edges. TerraCutter must choose which input image to take a duplicate pixel from. The amount of overlap varies from file to file in each data-set. Figure 8 depicts how input imagery files, numbered and outlined with solid thick lines, overlap each other within the UTM coordinate system. The tiles, outlined with light dashed lines within the numbered rectangles, depict the challenge in edge matching.

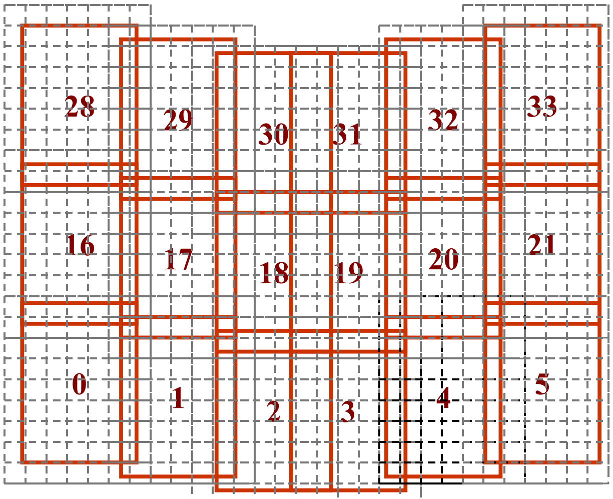

Figure 8. Tiling overlapping UTM images into a seamless mosaic. The dotted lines represent the mosaic grid. Up to four files can contribute pixels to a tile.

DOQ image files typically overlap each other by 100 to 300 pixels. DRG image files can overlap each other by 50 to 1500 pixels. However, only one file will contain "map data" while the others will contain map notes and tick marks found along the border of USGS topographical maps [4].

Conceptually, it is possible for up to four input images to contribute to a single TerraServer tile. To complete all the tiles for a single input image, a total of nine input images are needed – the center image and eight surrounding images. Unfortunately, the projected data sets are not delivered in sorted order. Finding all the adjacent input images would be a tape-shuffling nightmare. Hence, TerraCutter uses an incremental load algorithm.

TerraCutter tiles each input image independently. White space is added around the input image edge to align to the TerraServer grid system and the input data is re-sampled to the appropriate TerraServer resolution. Tiles are then cut and held in memory uncompressed.

After compressing each tile, TerraCutter looks for a tile with the matching Theme, Scale, X, Y, and SceneID properties in the appropriate TerraServer database imagery table. If there is not an existing tile, then TerraCutter inserts the image into the table and sets a "visibility flag" to "visible".

If a tile does exist in the database, TerraCutter compares the "blankness" of the newly cut tile with the "blankness" of the tile in the database. The following table contains the TerraCutter decision tree for handling image overlap:

New Tile Blankness	Db Tile Blankness	Action
0 %	0% – 99%	Discard Db Tile, Insert New Tile
1% – 99%	1% – 99%	Merge pixels from New & Db Tile
1% – 99%	0%	Discard New Tile, Keep Db tile

TerraCutter performs the following steps in one transaction –

(1) check for an existing image,

(2) fetch the image and merge pixels with the in memory copy,

(3) compress the new or merged tile,

(4) insert new tile row,

(5) and delete old tile if necessary.

Using the SQL Server concurrency control, other executing TerraCutters are automatically blocked from modifying the same tile, but can be updating other tiles in the same table. The TerraServer web application performs "dirty reads" of the imagery tables and is not blocked from reading the currently visible row. Thus, we are careful to delete the old tile as the last step so that the web application can get to a valid, but soon to be replaced tile, when TerraCutter is at step 2 or 3.

The program proceeds on to the next tile and repeats the process. When all tiles are cut from an input image file, TerraCutter updates the production status field in the Theme's Original Meta row to indicate that the input image has been completely tiled. TerraServer Administrators monitor the progress of the TerraCutter program through database queries against the Theme Original Meta table.

3.4 TerraScale

TerraScale re-samples the tiles created by TerraCutter to create the lower resolution tiles in the theme's image pyramid. To create a lower resolution tile, TerraScale takes four tiles from the next higher resolution and averages four pixel values into one pixel value. TerraScale repeats this process at every resolution level until it tiles the lowest resolution tile for a given theme

Figure 9 depicts how the highest resolution tiles loaded by TerraCutter contribute to the pixels at lower resolution. We refer to the tiles loaded by the TerraCutter program as the "base scale" or "base tiles".

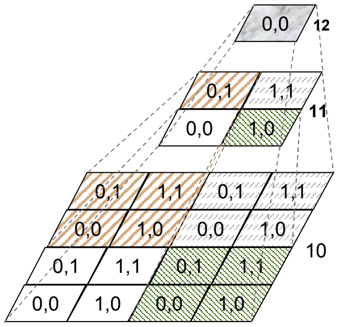

Figure 9: The TerraScale program computes the image pyramid by using averaging or using nearest neighbor algorithms on the four lower-level pixels.

The number of lower resolution tile levels created by the TerraScale program is theme dependent. The USGS DOQ and SPIN-2 data base scale is 1-meter resolution. TerraScale creates 2-meter resolution through 64-meter resolution – a total of seven levels.

The TerraScale program continuously scans the *ScaleJob* table for new work to do. If it finds a row with a "job queued", then it reads the job characteristics and updates the *ScaleJob* row to indicate that the TerraScale program is handling the job. This lets multiple TerraScale programs consume jobs on the same queue.

A *ScaleJob* row describes one complete image pyramid. The TerraCutter inserts or modifies a *ScaleJob* row when it inserts or updates a tile in the bounds defined by the top level of the Theme's image pyramid. In the USGS DOQ case, a single top-level 64m-resolution tile has a 64 by 64 tile 1m resolution base. Several *ScaleJob* rows will be inserted for each DOQ processed by the TerraCutter since a single USGS DOQ does not line up on an image pyramid boundary.

TerraScale begins a loop to create the tiles at the lowest resolution. This is the top of the image pyramid. It recurses down the image pyramid (higher resolution levels) until it fetches the four base tiles in one corner. The four tiles are averaged into one tile and inserted into the database. If the low-resolution tile had been sub-sampled previously, then the old tile is deleted.

The TerraScale program continues to walk up-and-down the image pyramid underneath the lowest resolution tile it is generating. When the top tile is sub-sampled, the ScaleJob is marked as completed and the program moves on to another job.

TerraScale is told which resolution levels are to represent the image pyramid in the search system. As a last step in building an image pyramid for a particular low resolution X,Y value, TerraScale inserts the appropriate rows into the *ImageSearch* tables. A tile is not visible in the TerraServer application until a row is inserted into the *ImageSearch table*.

4. What We Learned

4.1 Initial Results

The TerraServer project began in late 1996. A prototype was demonstrated in May 1997. Aerial Images went live with a demonstration web site in January 1998. The full site went live in June 1998. It has now been operating for over 18 months.

When the web site was launched on June 24, 1998, it was overwhelmed with 35 million "hits". We had clearly under-estimated the popularity of this type of data.

Working with the hardware partners and the SQL Server development team, we configured the hardware and tuned the system software to handle 40 million hits and 300,000 visitors per day.

4.2 Traffic Analysis

TerraServer continues to be a very popular web site. Below are the usage statistics for TerraServer's first year on the web:

Table 2: TerraServer traffic summary July 1998 to July 1999.

Summary	***Total***	***Average***	***Max***
Users	23,104,798	63,128	149,615
Sessions	31,011,284	84,730	172,545
Hits	2,287,259,402	6,624,607	29,265,400
Page Views	367,528,901	1,004,177	6,626,921
DB Querys	2,015,132,166	5,505,826	17,799,309
Image Xfers	1,731,338,052	4,704,723	14,984,365

Since the launch, TerraServer has reached a steady state of 5 to 8 million web hits, 5 to 6 million database stored procedure executions, and 50 GB of image tile downloads per day.

4.3 User Input

We received over 18,000 mail messages from users. We tried to answer each one. Most messages were constructive criticism or praise, but there were substantial complaints as well. The most common complaint was that images were missing. The server has only 30% coverage of the continental US and very spotty coverage outside the US. The second most common complaint was that images did not align. This forced us to go to the "scene" oriented design described here. The third most common complaint centered on the Java applet we wrote: it was difficult for us to get that applet to work on the many different Java Virtual Machines of the common platforms (each platform has many JVM variants).

4.4 System Availability

The Compaq database server and SQL Server 7.0 database management system containing aerial imagery have been extremely reliable. Table 3 summarizes the availability statistics for the period through 1 July 1998 through 1 July 1999. The system went out of service for 3 hours for software upgrades, 2.5 hours to move the server within the data center, and 33 minutes due to a software bug.

Table 3: Availability statistics for TerraServer SQL Server.

	Hours	***%***
Elapsed Time	8760	100.00 %
Availability	8754	99.93 %
Scheduled Availability	8757	99.97 %

The topographical map database server went into service in December 1999. Except for a 30 minute period for a software upgrade, the new server has not be out of service at all.

4.5 Database Size

Table 4 summarizes the database size as of February 2000.

Table 4: TerraServer Databases

	Aerial Imagery Db	*Topo Map Db*
Db Allocation	1.5 TB	1.0 TB
User Data Rows	170.7 million	104.4 million
User Data Bytes	1.1 TB	.75 TB
Log Size	75 GB	25 GB

The database is backed up regularly to the StorageTek 9710 TimberWolf tape robot using SQL Backup integrated with Legato Networker. In on-line mode, the backup consumes approximately 20% of the CPU resources and takes approximately 8.5 hours to complete including tape changes.

Table 5 summarizes the number of uncompressed images received from each organization and compression results:

Table 5: Image Counts

	Comp. Ratio	*Input Images*	*Input Size (GB)*	*Tiles (M)*	*Tile Size (GB)*
Aerial	5	97,500	5,864	133.6	1,136
Topo	5.4	4,335	2,006	39.6	762

4.6 Application Size and Complexity

Table 6 summarizes the size of the TerraServer application source code:

Table 6: Application Source Code.

Item	*Modules*	*Files*	*Lines*
T-SQL Stored Procedures	20	20	5468
Active Server Pages (Tier 2 Web App)	33	41	5727
Load Programs	3	70	39301

One full-time developer and 4 part-time developers built TerraServer. 1 full-time system administrator and 1 full-time data-load specialist maintain the web site.

4.7 New Satellite Imagery Users

Remote sensing and aerial photography have been a niche application due to the high complexity and expense of tools that can view it. TerraServer dramatically reduced the access complexity and cost for simple applications. The application is so simple to use that, high-resolution imagery is now available to the entire Internet population. We and others have been astonished at the wide interest in the data: tens of thousands of people visit the site each day (see Table 2).

4.8 Relational Databases as Image Repositories

Using relational databases to store image pyramids of common graphics file formats, e.g. Jpeg and GIF, forces the separation of storage management from image presentation. The tiled design allows rapid pan and zoom to any part of the image database. It also supports background loading of new images while the current data is being viewed. The database system is able to handle much larger image bases than a file-per-image design used by earlier efforts. Storing tiles individually also allows for easy on-line editing of any portion of an image. Choosing a ubiquitous medium like the Internet and the common web browser as the presentation tool enabled the rapid dissemination of high-resolution imagery to new users and applications.

4.9 The Value of Cooperative Joint Research

Because the project had to use real data, and that data was expensive, it forced us to enlarge the project team beyond database and systems researchers. By including additional companies and organizations, the project goals and requirements expanded. This brought additional skills to the table – geographers, graphics researchers, high-resolution image interpreters (a.k.a. spies), and GIS experts. We were able to blend the knowledge and skills of diverse partners to build a powerful spatial data warehouse and produce a more complete result by solving a wider set of problems than just a database or operating system problem.

4.10 Integration With Encarta Online

TerraServer became part of the Encarta Online web site in May 1999. The Microsoft Encarta product team cross-referenced Encarta Encyclopedia articles with the TerraServer imagery. As users navigate the imagery, hypertext links appear to related Encarta Encyclopedia articles. This vastly improves the richness of the user interface.

5. Future Work

5.1 Layered Maps

We are collaborating with UC Berkeley Digital Library Project, http://elib.cs.berkeley.edu/, on layered maps. The USGS DOQ and DRG data sets are in a common projection system. The TerraServer tiling algorithm cuts tiles so that client applications can identify overlapping tiles from separate themes. We plan to work with the UCB Digital Library team to build a client application which will display TerraServer projected data-sets that are in the same projection as a layered map set. The layered map user interfaces has the same ease-of-use goals as our traditional single-layer HTML interface

5.2 Dense vs. Sparse Coverage

The feedback from users has encouraged us to acquire complete coverage of smaller geographic areas. Thus, in the last 9 months, we have concentrated on adding the USGS DRG topographical map theme. The USGS DRG data-set completely covers the conterminous United States and Hawaii. Thus any U.S. user is guaranteed to find some information about where they live.

We have focused the corpus of the TerraServer image on the United States instead of attempting to load a sparse set of scenes from around the world.

5.3 Distributed Web Sites

The popularity of the web site has encouraged other data providers to offer interesting data sets. These firms generally have a commercial interest in advertising, selling, and distributing their data. They often have data volume, security, and user interface requirements beyond the scope of our research project.

The present TerraServer design allows multiple database servers and multiple web sites to be configured into a single TerraServer web service. Currently, we are experimenting with off-loading the SPIN-2 data to Aerial Images, Inc. located in Raleigh, NC. The Microsoft TerraServer site, http://terraserver.microsoft.com, and Aerial Images site, http://terraserver.com, gazetteers cross reference each other.

The multi-web site service design allows us to locate web sites near the users with the most interest in the data. It also allows individual data providers to tailor the TerraServer web application to their specific requirements and taste. We hope to add other data providers in the future preferably at least one per continent.

6. References

[1] Barclay, T., et. al., The Microsoft TerraServer, Microsoft Technical Report MS TR 98 17, Microsoft Corp, Redmond, WA. http://research.microsoft.com/scripts/pubDB/pubsasp.asp?RecordID=155

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

[3] 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

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

[5] Arther H. Robinson, Joel L. Morrison, Phillip C. Muehrcke, A. Jon Kimerling, Stehen C. Guptill, *Elements of Cartography, Sixth Edition,* John Wiley & Sons, Inc., U.S.A. 1995, ISBN 0-471-55579-7.

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

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

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