Remote Access to Very Large Image Repositories, A High Performance Computing Perspective

Lucian Plesea
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

***Abstract*: The main challenges of using the increasingly large repositories of remote imagery data can be summarized in one word: efficiency. The efficiency of storage, transport, access and evaluation are even more critical when high performance computing environments are involved. In this paper, a number of concrete problems and the chosen solutions are described, based on the construction of a 5TB global Landsat 7 mosaic.**

INTRODUCTION

The Web Map Service (WMS) Global Mosaic is a consistent global image dataset, the result of combining more than 8200 individual Landsat7 scenes, or more than 5 TB of data. A first complete version was assembled in early 2004 and became available on the World Wide Web via a WMS interface in April 2004, on the OnEarth.jpl.nasa.gov portal. A second version of the mosaic, with improved coverage and increased positional accuracy is currently being assembled and will be ready shortly. There are many interesting aspects to this project, but by far the most challenging ones are directly related to the sheer size of the dataset, and how this data is used on high performance computers. Some of these issues will be discussed in detail, and should provide useful insight to anybody that has an interest in using a very large dataset:

- The storage issue is an obvious one. The input dataset was expected to be around 5 TB, or about 550 MB per scene. Assuming that the output mosaic has the same size, a minimum of 10 TB would be required just to store the mosaic.
- The format chosen for storage is critical for datasets of this size because transcoding a very large dataset is an expensive operation. The format chosen had to allow frequent and possibly concurrent updates, and had to guarantee data integrity.
- Access to the data had to be provided via a simple and efficient interface. It was obvious that more than one application had to access both the input and the output data, ranging from simple data extractors to the full complexity mosaic builder.
- Assessment of the output product had to be possible while the processing was still taking place. Waiting a few months just to realize the output is wrong was not a good choice.
- Finally, there was the access to the resulting data. The results had to be immediately accessible.

The problems and solutions discussed here are not the result of a direct initial design. In many cases a lengthy development process took place, in which many choices were made, features were added, the results were tested and the development cycle restarted. In addition to the production of the WMS Global Mosaic, a number of other applications and datasets are used as examples to illustrate particular points.

WMS GLOBAL MOSAIC

The WMS Global Mosaic project started in early 2003, at the same time the first scenes of the GeoCover 2000 dataset were delivered to NASA. A total of 8000 orthorectified Landsat 7 scenes were expected, with each scenes consisting of nine separate image files that characterize the Landsat 7 imaging instrument, the Enhanced Thematic Mapper Plus (ETM+). Each scene is about 700 MB in size, so the total input data size was initially estimated at 5 TB. The first full resolution version of the mosaic was complete in April 2004, when it became available to the public via the WMS server OnEarth.jpl.nasa.gov. Once the full GeoCover 2000 data set was available, a complete coverage mosaic was started. This version of the WMS Global Mosaic has just been completed, and will be replace the initial version on WMS server. The existence of an on-line global mosaic, containing multispectral data and accessible via community standard protocols, can greatly simplify the creation of a data-rich image in any location and at any scale, encouraging the development of applications that can exploit such data-rich sources. It has already proved extremely popular, with more than a dozen known client applications.

The WMS Global Mosaic itself attempts to preserve as much of the spectral information contained in the source data as possible, yet provide a relatively seamless blending of adjacent scenes. It does this by adjusting the contrast for a whole scene at a time, across all of the nine spectral bands. After this, large differences between neighboring scenes were still present, and a recursive filer process reduced these differences to tolerable values. The panchromatic band mosaic image has a resolution of 0.5 arc-second per pixel, while the other six EMT+ bands are limited to one arc-second per pixel. The area covered by the mosaic is 85N to 85S, resulting in an image size of 2,592,000 by 1,224,000 pixels for the panchromatic band. For reference, this mosaic image is 3,600 times larger than the well known NASA Blue Marble MODIS mosaic.

A. Storage

The storage issue was the first big problem. 10 TB of on-line storage was seen as a reasonable minimum. Based on previous experience, the choice of dealing with the complete dataset at all times was greatly preferred over the alternative of only operating on smaller subsets and staging data in and out. The deciding factor here was the cost-to-benefit ratio of using IDE RAID systems. These systems are increasingly seen as alternatives to tape backup systems,

[*] Portions of the work described in this paper were funded by the NASA ESTO-CT Project and the NASA GIO.

and in some cases they do compare favorably with high performance on-line storage systems. In a shared effort with other on-going projects, a 40 TB storage system was built, named Raid Again Storage using Commodity Hardware and Linux (RASCHAL). This system consists of ten Linux systems linked via a 24-port Gigabit Ethernet router, where each Linux system contains two 2 TB hardware RAID5 volumes built from 250 MB IDE drives [Fig. 1]. It was built at JPL in March 2003, and has been in continuous operation. Since it was needed as a storage system immediately and a significant amount of time was spent just to find a working configuration, not many performance tests were accomplished. The hardware RAID card was well supported in Linux, but support for large block devices and large file systems was not very good. The resulting decision was to build a 4 TB RAID0 volume from the two RAID5 hardware devices (the source of the Raid Again acronym), use ReiserFS version 3 for the file system and a development version of the 2.4 Linux kernel. An immediate upgrade to a pre-release 2.6 kernel solved most of the stability issues experienced with the initial configuration. The performance was acceptable, with peak data rates of 20 to 30 MB/sec to a single system being observed. A more recent update of the kernel, combined with minor kernel tuning of disk and network parameters, brought the sustained data rate to 40 MB/second. There was no provision for backup other than relying of the redundancy of the RAID5 hardware and the possibility of duplicating the data. During the two years of operation, this system survived four blackouts, the California power crisis, and multiple power-line spikes generated by heavy construction equipment used to remodel the building. One of the problems encountered was the lack of operational management software for the hardware RAID cards, which was only provided for the stable Linux 2.4 kernel version. The computer security issues were minimized by restricting access to the storage systems to a few chosen host systems and disabling a direct internet connection.

The conclusion here is that using inexpensive hardware for disk servers works well, as long as constant monitoring of the hardware does exists and the Linux systems themselves are treated as embedded disk servers, slaved to a host computer, thus minimizing the system administration task.

B. Data Format

The data format used for both the input and output is an updated version of an existing technology: a very simple tiled file format. This format was enhanced by the addition of indirection via an index file. This type of file is a very common solution when dealing with large images, being present in many established commercial and public domain data formats such as JPEG2000 and TIFF. The tiling increases the efficiency of access to a local area, while the indirection makes it possible to have individually compressed regions. This data format was mostly intended as a temporary working format, with emphasis on flexibility, but lacking many of the features required by a data transfer format. The main reason for a custom file format was again the size of the dataset. Most existing file formats are built around a 32 bit integer size which effectively limits the size of the file and the size of the image. Using such a format, the WMS Global Mosaic would have required a few tens of thousands of component files, making it very hard to manage. One particular requirement for this project was the ability to add or modify the image without having to copy any data. Since the output image was expected to contain more than 1 TB of data, any copy would have been prohibitive. The possibility to roll back to a previous version of the dataset also exists, by simply using an older version of the index file. The index file also makes the file format resistant to computer and network crashes, first by making a change active simply by changing an index entry after the image data has been written, and second, by making it possible to regenerate a corrupted portion of the data without disturbing the whole file. A number of tile compression algorithms are implemented via standard libraries such as zlib, bz2 and libjpeg and other can be added. In the case of the WMS Global Mosaic, the lossless zlib was used, as the best tradeoff between speed and compression ratio. The same format was used to transcode the input data from the source TIFF files, mostly to reduce the size required to store them. They are much harder to manage than the output mosaic, since there are more than 100,000 individual input files.

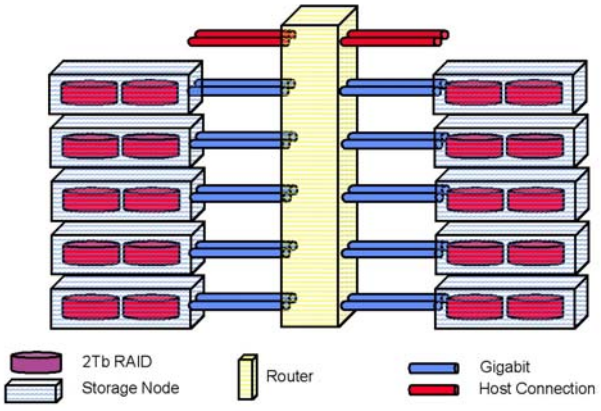

Fig. 1 RASCHAL diagram

The main observation here is that traditional file formats are best suited for transfer and storage of small size datasets, yet in the case of a very large dataset, a simplified custom data format is much more efficient in many cases, especially as a working format.

C. Data Access

The high performance computer that was used to run the application that produced the WMS Global Mosaic was situated remotely from the RASCHAL system used to store the input data. This is a common occurrence when supercomputers are used, and is traditionally solved by copying the dataset to the target computer, and then copying the output dataset back. When very large datasets are involved, this method is not satisfactory, and indeed, it is the main reason for computational grid development. Recognizing this fact, as well as the need to link heterogeneous computing environments, a prototype image specific network access protocol was developed, with the WMS Global Mosaic as the test application. It was initially implemented as a simple extension of the SGI Image Format Library, limited to IRIX systems. The server components have been ported to Linux, and now operate on the RASCHAL systems. Using this protocol, the entire 5 TB input dataset was processed on a large IRIX machine located at NASA AMES, while the source data was stored on RASCHAL at JPL. The output mosaic was sent using the same protocol back to RASCHAL. Compression and decompression of the data can be done either at the server or the client side, making it possible to optimize the data access based on the bandwidth and the latency of a given configuration. In the case of the WMS Global Mosaic processing, the source data was decompressed by the client application, taking advantage of the large number of CPUs available. The results were

sent uncompressed back to the storage server, which would buffer and acknowledge the data, allowing the application to continue without delay, with the data compression done by the Linux network image server, asynchronously from the application. This configuration was able to sustain data rates in excess of 10 MB/second while completely eliminating any data staging. Indeed, the WMS Global Mosaic application can be deployed to a different machine in less than one hour. On a small scale test, there was no significant performance difference between the mosaic application using local NFS file access and the Image Network Protocol. The network image protocol shares most of the basic features of the file format described in the previous section, but it is not limited to it. It effectively insulates the application from the details regarding the effective data location and format, including virtual datasets. For example, the OnEarth server uses an image network server that can scale the nine bands of the WMS Global Mosaic to a common resolution in real time, thus offering access to a virtual dataset. Another such virtual data server combines a vector file containing US roads and an image file into a virtual image.

Accessing remote image datasets via an image specific network protocol can be indeed, very productive. Surprisingly, the biggest obstacle to the use of this technology has proven to be the computer security. Use of encryption is possible, but significantly degrades efficiency, by increasing the latencies and limiting the achievable bandwidth.

D. Access to results

It is very important to be able to assess the output of an application that is processing massive amounts of data while the application is ongoing. In the case of the WMS Global Mosaic, several approaches proved successful. Since the WMS server is already functional, it can be used to request areas of the output mosaic while it is being created. The data can be examined in the context of other maps, even from other servers, allowing easy evaluation of the geolocation accuracy and the image quality. Since only a web browser is needed, the inspection of data can be done from any location, and can be shared with others by a simple web link. One of the unexpected outcomes of the use of a tiled format where each tile is compressed independently is the use of the size field of the index file as a direct measure of data entropy. These fields can be organized as an image, resulting in a major data reduction while providing interesting insight into the dataset itself.

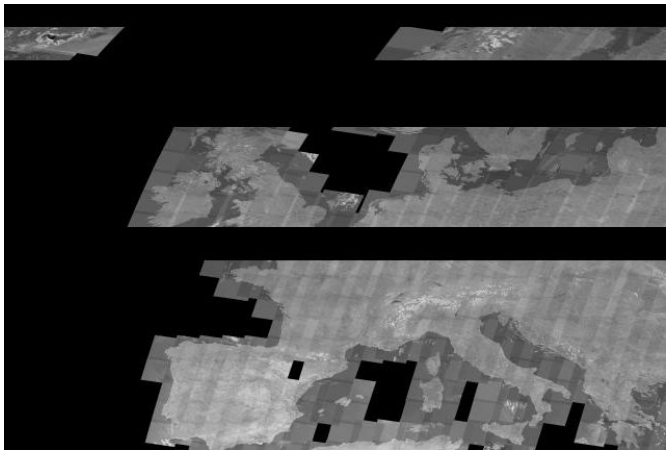

Fig. 3 Entropy image detail, unfinished areas

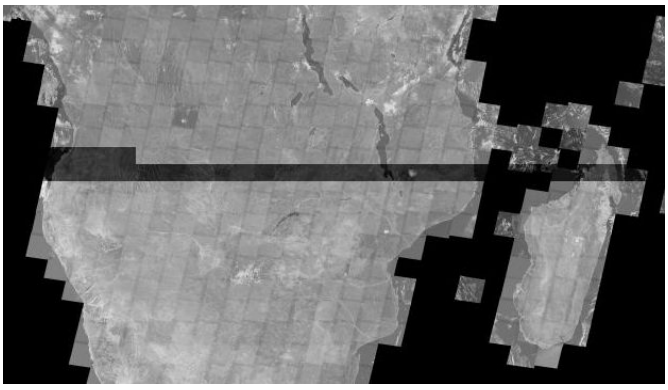

Fig. 2 Darker region due to low resolution data

In Fig. 2, a portion of the entropy of the panchromatic band is shown. The fact that the processing is not complete can be seen in the black areas at the top of the image. A qualitative assessment of the output is immediate, since a brighter pixel means a higher detail level in the corresponding tile. The black areas in the top left corner correspond to the snow covered areas of the land. The brightness of those scenes is very uniform, and sometimes it is even clipped at the maximum available value, producing very small compressed tiles.

In Fig. 3, a darker band in the lower part of the image can be seen. This is the result of an operator error, in which lower resolution input was used, resulting in blurry images. This error is evident in the entropy image, but could otherwise be hard to catch.

Since the processing of a large image dataset takes a very long time, it is essential to be able to gain early access to the output. The use of a file format that allows inspection of an unfinished product is useful, as is a fast method to observe a spatially reduced version of the data.

E. WMS as access

The WMS server itself is fast becoming a data server for a georeferenced large image repository. The OnEarth server architecture pictured in Fig. 4 is an example of this trend by the inclusion of a high volume component. The WMS protocol itself is built on top of HTTP, where a request for an image is a simple URL with the proper parameters. The main parameters are the image size in pixels and the geographical bounding box in a certain earth coordinate system, usually unprojected. The result of a WMS request is a map image, which can be either in a lossy compression format tuned for efficient use of the available bandwidth or in a lossless image format suitable for data access. Unfortunately, the greater flexibility offered by the WMS and the lack of a permanent TCP connection between the client and the server are significant obstacles in building an efficient remote data access system based on this protocol. A high performance client application requires a very low latency and high throughput connection to the data server, the type of connection best served by a permanent TCP connection is best suited.

There are also significant advantages in using WMS as a data access protocol, at least in certain cases. A common case is a client application with moderate data demands that generates a predictable call pattern, usually arranged in a grid. In this case a server side caching system can be used to greatly improve access rates,

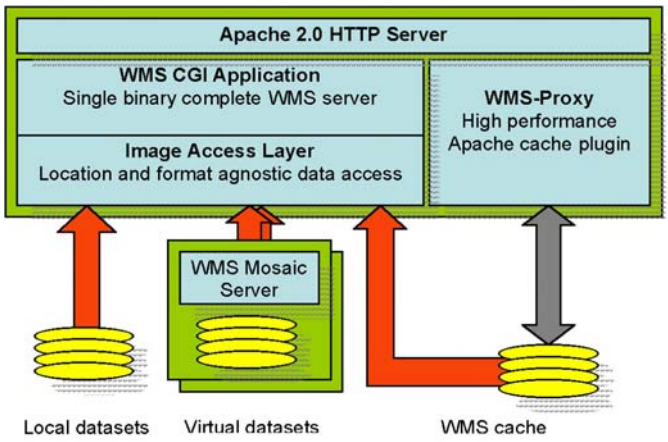

Fig. 4 OnEarth WMS Data Server Architecture

especially when multiple simultaneous clients are active at a given time, requesting data from a single server. The OnEarth WMS server has been able to sustain more than 150 requests per second using such a caching system, while the full WMS server implementation peaks at about 7 WMS requests per second. This type of system does represent a very functional compromise, allowing flexibility in the development of client applications and a providing a simple method to transition to a much higher service level without a significant server upgrade.

The main advantage of an HTTP based data access system is the availability of high performance and scalable server software as well as a well understood and supported computer security model. The use of WMS and Web Coverage Service (WCS) as remote access protocols is a new development with great potential.

CONCLUSION

There are many difficulties in using remote access to large image repositories, with every stage of the application having to be carefully evaluated to preserve the efficiency. In the case of the WMS Global Mosaic, this was possible as direct result of a sustained effort to understand and optimize such a remote access system. One of the most promising recent developments is the emergence of the WMS and WCS as image data access methods. A combination of full featured WMS servers such as OnEarth, collocated with very large image repositories and operating on supercomputing class hardware might represent the future model of both data distribution and supercomputing centers.