

1 Introduction

ER Mapper Image Web Server solutions serve imagery stored in ECW and/or JPEG 2000 files. This differs significantly from the approach used by most other systems, which break the images up into tiles and then store them in a database. This document discusses both approaches and makes recommendations on how a prospective user should assess which is most suitable for them.

2 Serving Image Tiles from a Database

Traditional GIS or Map Serving solutions usually work by querying a database for the image tiles which comprise the area that the user has requested. In order to ensure a quick response, a number of copies of the image are stored in a “pyramid” of increasing resolution. In this way, the database can quickly construct an image for the user at an appropriate resolution, whether they are looking at a low resolution overview of the entire image or a high resolution view of a zoomed-in portion of the image.

2.1 Image Preparation

With a database approach the following needs to be done:

- Different versions of the image need to be generated at the resolutions needed to create the “pyramid” within the database.
- Each of the different versions of the images needs to be cut up into tiles of the desired size.
- The tiles need to be loaded into the database.

3 Serving Images from ECW or JPEG2000 Files

Rather than requesting image data from a database, the Image Web Server reads it directly from an ECW or JPEG 2000 file. These formats use a technique called wavelet compression, which has an inherent pyramid structure (analogous to that described above, but implemented in a way which requires much less space). Because of the way these files are structured, retrieving arbitrary extents at an arbitrary resolution is quick and simple.

One of the advantages of this approach is that it is usually much faster. This is because the Image Web Server was purpose-built for the job of serving images quickly and efficiently, whereas most other implementations are built on top of more general purpose database technology that is not as optimised for image serving. Sources of the speed differences are:

- The imagery is more efficiently compressed in ECW or JPEG2000 files.
- There is no need to stitch together multiple tiles when the user is looking at an area where tiles overlap.
- There is no database overhead in accessing the files directly.

3.1 Image Preparation

With the ECW/JPEG2000 approach, the following needs to be done:

- The images need to be converted from their raw form into ECW or JPEG2000. Often multiple images are mosaiced together into a single large image (with Image Web Server it is often more efficient to serve a single large image than multiple smaller images¹). Note that conversion to ECW or JPEG2000 can be done offline – it doesn't need to be done on the server itself.

¹ The Image Web Server does not have a problem serving small files if this is desired, but there is no need to chop images up into even smaller tiles for the purpose of image serving.

- The images need to be placed in a location accessible by the Image Web Server (usually on a file system local to the server itself).

4 Which approach is better?

When deciding which approach is best for your environment and application, we recommend that you benchmark the different approaches using the list of requirements associated with your particular needs and environment.

ER Mapper is happy to provide advice and support to organisations wanting to benchmark an Image Web Server implementation. We recommend getting similar support from suppliers of alternative technologies when benchmarking their solutions.

Some typical measures you may want to consider when comparing alternatives are listed below:

4.1 *Ease and speed of loading data into the server*

How easy is it to get images into the server? How long does it take? Consider the whole process including any processing needed to get images from their raw form into the server.

4.2 *Functionality*

- What image quality is required? Is compression configurable? If tiles are compressed, are there artifacts where the tiles are joined?
- What sort of user experience is desired (static image/map display or dynamic roaming and zooming)?
- Is there a need for users to work with multiple layers using dynamic transparency and or side-by-side linked views?
- Do specific image protocols need to be supported? (e.g. ECWP, WMS, ArcXML, ImageX, other?)
- Is there a need to integrate with other systems? How easily can this be done?
- What browsers need to be supported?
- Do you need to serve into desktop applications as well as web browsers? If so, which applications? How easy is it for client applications to communicate with the image server?

4.3 *Performance*

- File size: What are the largest images you'll need to handle in the initial implementation? In the future?
- How responsive does your server need to be? E.g. How many "image views per second" need to be displayed (where a new image view is created every time a user zooms or pans).
- How many concurrent users does your server need to handle while maintaining the desired response time?
- How much hardware is needed to achieve your desired performance (number of servers plus how they are configured). Consider the following:
 - The impact on the server CPU as the number of concurrent users increases. How quickly is the CPU saturated?
 - The impact on server I/O as the number of concurrent users increases. How quickly is the I/O saturated?
 - How much RAM is needed to operate at the performance levels you require?

4.4 *Setup and Maintenance*

- How easy is it to set up the image server? What specialist skills are required if any?
- What maintenance is required? Are specialist skills required?

- How easily will it be to integrate the image server into your existing infrastructure?
- How will it fit into your current administration practices (e.g. backups, disaster recovery, etc).

4.5 Cost

Look at the total cost of ownership including:

- Software - What is the licensing model: Per server? Per cpu? Other?
- Hardware costs: How many servers will be needed?
- Setup costs.
- Data loading costs.
- Maintenance costs: What type of support is required? Do you need to have internal expertise?

5 Summary of expected results

It is important for you to make your own assessment of the available alternatives. However, our expectation (based on benchmarks done by our existing clients) is that you will usually find the following:

1. It is easier and quicker to load images into an Image Web Server. The pyramiding/tiling process with a database solution is typically rather slow and can tie up your production server for a considerable amount of time (hence requiring replication and additional associated HW/SW costs to alleviate this problem).
2. Where there is a need for image functionality beyond straight image display, the Image Web Server provides more features and flexibility. Note however that the Image Web Server is designed to be purely an Image Server without GIS or database functions. Where database or GIS functionality is needed, a suitable system will need to operate alongside the Image Web Server.
3. It is easier to build client applications that can read Image Web Server data.
4. Image Web Server significantly outperforms other solutions. This difference can be dramatic where the ECWP protocol is used. This is because ECWP enables massively distributed processing (with much of the work done on the client machine rather than on the server). Where high performance is a requirement, this can result in significantly lower hardware, implementation, and maintenance costs.
5. Data volumes are lower in an Image Web Server solution.
6. Image Web Server is currently implemented as a Windows solution and is designed to be easy to integrate with other systems. A Solaris version will be available in Q3 2006. Other platforms will be supported if there is sufficient demand.
7. Image Web Server will usually be the most cost effective solution in the following circumstances:
 - a. Where ease of setup, data loading, and administration is considered a factor in the total cost assessment.
 - b. Where high performance is a requirement (large images and/or fast response and or many concurrent users).
 - c. Where a highly interactive user experience is needed.
 - d. Where many different protocols need to be supported (i.e. to support multiple browser environments and/or desktop applications).

Where only basic image display is required with a small number of concurrent users, it may be harder to justify the cost of an Image Web Server (though it is worth looking at the lower cost office and corporate versions).

6 Conclusion

In most cases, file based image serving using Image Web Server provides better performance than a system that provides image serving capabilities as part of a GIS or database application. Furthermore, the functionality and flexibility of Image Web Server usually provides better support for advanced image serving requirements.

However different organisations have different needs, and we would encourage prospective users to use benchmarking to verify the type of solution that will best meet their requirements.