Understanding ArcIMS Virtual Servers

By Martin Fleming, ESRI Education Services

A powerful feature in ArcIMS—the Virtual Server—gives an ArcIMS administrator control over how computer hardware resources are utilized. The Virtual Server allows ArcIMS services to take advantage of distributed processing for scalability, performance, and reliability. Modifying the default Virtual Server configuration lets an ArcIMS administrator

- Maximize throughput.
- Reduce response time.
- Constrain memory usage.
- Increase CPU utilization.
- Improve reliability.

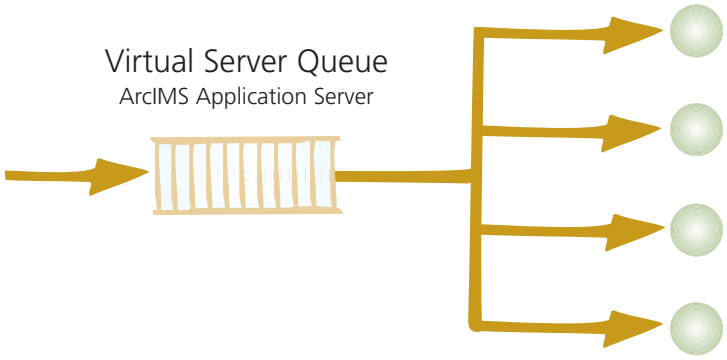

Figure 2: A Virtual Server is a set of instances of a single function type that forms a logical group for the concurrent processing and load balancing of requests.

ArcIMS Architecture

To understand the ArcIMS Virtual Server requires some knowledge of ArcIMS architecture. This architecture incorporates a processing model that uses many components working in cooperation to execute simultaneously and create map data for delivery by the Web server. Components, communicating using standard TCP/IP protocols, can be distributed anywhere on the network and still remain synchronized. Each host computer's operating system controls the execution of ArcIMS processes and manages memory. These ArcIMS processes and the names of their executable files are listed in Figure 1. The ArcIMS Application Server and ArcIMS Spatial Server are the most important processes related to the function of Virtual Servers.

Process	File name
ArcIMS Application Server	Aims_AppServer.exe
ArcIMS Monitor	Aims_Monitor.exe
ArcIMS Tasker	Aims_Tasker.exe
ArcIMS Spatial Server	Aimsserver.exe

Figure 1: ArcIMS process

ArcIMS Application Server

The ArcIMS Application Server handles load balancing for incoming requests and tracks which ArcIMS services are running on which ArcIMS Spatial Servers. The ArcIMS Application Server has no user interface. It is configured using the ArcIMS Administrator application. This is the same application used to create Virtual Servers, Spatial Servers, and instances maintained by the Application Server.

ArcIMS Spatial Server

The core of ArcIMS, the ArcIMS Spatial Server, is a container for instances that access and produce maps and data in an appropriate Web format before a response is sent to a client's browser. The Spatial Server's primary job is to dynamically render the map images that will be displayed on HTML pages.

Using the ArcIMS Administrator, additional ArcIMS Spatial Servers can be added to a local machine or a remote computer running ArcIMS Spatial Server software. Note: Additional ArcIMS Spatial Servers on separate machines require additional ArcIMS licenses. The Spatial Server's workload is far greater than that of any other ArcIMS component. When considering load distribution, the resources required by the Application Server, Monitor, and Tasker can safely be overlooked. The most important factor is the total number of instances inside the Spatial Servers.

Instances

Instances are the fundamental processing unit of the ArcIMS Spatial Server. An instance takes a request and generates a response that can be sent back to a client. Multiple instances mean that multiple requests can be processed concurrently as illustrated in Figure 2. Instances belong to Virtual Servers but exist within an ArcIMS Spatial Server. Instances process one request at a time.

In a typical default installation of ArcIMS, two instances are assigned for each default Virtual Server in the default Spatial Server. Without creating any new Spatial Servers or Virtual Servers, two instances are assigned to each of the six default Virtual Servers (not including any extensions).

Creating a Service

An ArcIMS service is a presentation of spatial data and metadata for a particular use made available through a Web server. The symbology, labeling, and layer draw order for the service are defined by a configuration file that can be an ArcXML file or an ArcMap document if the ArcMap Server extension is used. A service accesses the functionality of ArcIMS Spatial Servers through a Public Virtual Server.

Virtual Servers

A Virtual Server is a set of instances of a single function type that forms a logical group for the concurrent processing and load balancing of requests. A Virtual Server can have instances hosted by one or more ArcIMS Spatial Servers and can place instances on any ArcIMS Spatial Server registered with the ArcIMS Application Server.

Each Virtual Server provides a type of service and is either public or private. Public Virtual Servers are used by ArcIMS services directly and can be Image, Feature, or Metadata Servers. Private Virtual Servers include Geocode, Extract, and Query Servers and are not used by ArcIMS services unless redirected by a request. Route Server and ArcMap Server are optional extensions that can be added to an ArcIMS installation. There are four relationships that are important to remember when creating or modifying a Virtual Server.

- Each ArcIMS service uses one Virtual Server.
- A Virtual Server can use more than one ArcIMS Spatial Server.
- More than one Virtual Server can use an ArcIMS Spatial Server.
- A Virtual Server can be used by more than one ArcIMS service.

For an illustration of these relationships, see Figure 3. MyService1 could use FeatureServer1, ImageServer1, or MyNewImageServer though it is shown using FeatureServer1. In this diagram, MyService2 is using the Virtual Server ImageServer1 that has been modified so it has four instances on SpatialServer_1 and

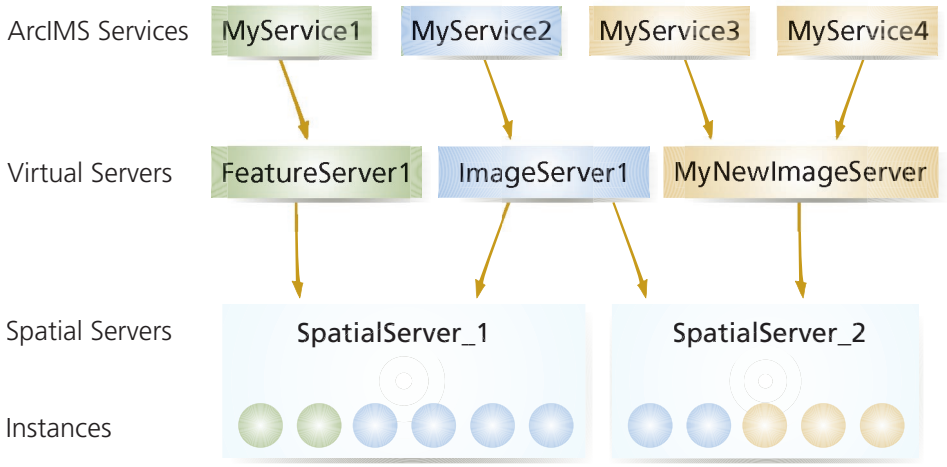

Figure 3: A Virtual Server can have instances hosted by one or more ArcIMS Spatial Servers. The four relationships that are important to consider when creating or modifying a Virtual Server are illustrated here.

two instances on SpatialServer_2, yielding a total of six instances. Creating a new ArcIMS service that utilizes ImageServer1 will add the service to all six instances and the service will be able to handle six simultaneous requests. Note that SpatialServer_1 contains instances of both type image and type feature. Also, two services, MyService3 and MyService4, are using the same Virtual Server (e.g., MyNewImageServer).

Adding a Service

When a service is created, it gets added to each instance individually in the Virtual Server. ArcXML configuration information is sent to the ArcIMS Spatial Server process that contains each instance by the Application Server using a separate TCP port for each instance. Lots of services and instances may mean that this process could take a while as the same messages are repeated for each instance in sequence.

Adding and deleting services are atomic operations. Refreshing a service causes it to be deleted. The service is then added again to each instance. Changing service properties is a similar process—the service is deleted and added again with the new settings applied to each instance. If changing a Virtual Server's properties creates a new instance, the Application Server will automatically add all of the services that use it to the new instance.

How Many Instances Are Needed?

When ArcIMS receives a request directed to a service, a Virtual Server handles it. In addition to a collection of instances, a Virtual Server also has a queue that resides in the Application Server. Its instances exist inside the Spatial Server(s). When a request arrives, it may spend time waiting in the queue until an instance becomes available that can service it. Because each instance in a Virtual Server provides exactly the same service, requests are served in a round robin fashion. The next available instance services the request.

The time between receiving a request and sending a response has two main components—queuing time and instance time. If an administrator increases the number of instances, requests will spend less time waiting in the queue and will be served by an instance sooner. However, this may or may not decrease the response time. The amount of time it takes to process a given request has a lower limit determined by the time spent using resources such as processors and disks. The processing time for a given request is always proportional to the work done. For a given request, an instance always has to carry out the same steps to create the response—fetch data, render symbology, render labeling, and so on. The ArcIMS Spatial Server cannot create a response any faster despite more instances.

Increasing the number of instances can improve throughput if the server has some unused capacity. Using multiple instances allows the computer to take advantage of waiting times that occur while an instance is waiting for I/O processes. Multiprocessing allows the operating system's scheduler to use the computer's resources more efficiently.

The basic unit of execution for Microsoft Windows is the thread. Each process contains one or more threads. Threads can run on any processor in a multiprocessor system, so splitting the ArcIMS Spatial Server into multiple concurrent threads is a way to take advantage of computers that have more than one CPU. ArcIMS uses multiple threads or processes to decrease response time per request where each instance uses a separate thread.

Running programs are sliced and processed by the CPU in an interleaved fashion. The operating system maintains threads or processes in such a manner that each ArcIMS instance "believes" that it is running exclusively. In reality, requests run for a little while, freeze, thaw, and run again. Because the multiple processes are competing for the same hardware resources, performance saturates after utilization of the resources (e.g., CPU, disk, and memory) reaches 100 percent. Increasing the number of instances on a server can improve throughput up to the point at which the server is saturated.

Balancing Throughput and Response Time

Using multiple instances improves throughput and response time, but the number of instances cannot be increased without limit. Every instance requires additional memory and needs to be managed as a thread by the operating system. This increases the amount of overhead required in addition to the actual work being done by the instance. The optimum number of instances varies depending on hardware configuration and service content. Usually, this translates into between two and four instances per CPU for a physical machine.

In a processor sharing queue configuration (i.e., multiple instances for a single CPU), the Spatial Server's service is shared equally among all requests being handled. If there are $n > 1$ requests present, each customer receives service at the rate of $1/n$. When two requests

Continued on page 40

Understanding ArcIMS Virtual Servers

Continued from page 39

are present, the Spatial Server works simultaneously on both instances and devotes an equal amount of its attention to each. If a Spatial Server is servicing a request with one instance and an idle instance on the same machine receives a second request, the first request will experience a sudden drop in its service rate.

If a Virtual Server contains only one instance, the throughput is simply the inverse of the time to service a single request. When a second instance is added to a single processor machine, the throughput is not doubled—it increases by a lesser amount because the server is dividing its time between the requests. While some gains are made due to the efficiency of multiprocessing, response times for individual requests are longer despite the fact that overall throughput has increased.

Under light loads, varying numbers of instances are idle so that arriving requests face little congestion and get serviced at a rate that frequently empties the queue. As the load increases, fewer instances are idle and the server's attention must be divided between more requests. Under a heavy load, all instances will be busy.

Although a heavily loaded server will exhibit a high throughput if many instances are busy servicing requests, the response times will be higher. If the number of instances is increased to a value that far exceeds the number of CPUs available, the server will operate at a throughput close to its saturation value and response times will suffer. Users will experience time-outs while waiting for requests.

The optimum number of instances can't be chosen simply by applying a rule of thumb or predicting the number from a model. An administrator should tune a Virtual Server while it is in production. Requests from the Internet tend to come in bursts, and it is difficult to place a load on a development machine in a manner that realistically simulates this random arrival of requests. The goal is to choose a number of instances that increases the throughput of the server while keeping response times acceptably low.

Conclusion

The Virtual Server is a very successful and elegant feature of ArcIMS architecture. The configuring options available with the Virtual Server give ArcIMS great flexibility in solving performance issues. Virtual Servers can be used to isolate services to improve overall stability if one service uses a data source that has an unreliable connection. Instances in a Virtual Server can be increased to reduce queuing and waiting times or decreased to reduce memory consumption. In a multiple server configuration, requests can be directed to separate hosts to effectively double the throughput as compared with a single server.

For more information about ArcIMS, a two-day instructor-led course, *ArcIMS Administration,* is offered by ESRI Education Services (www.esri.com/training) and two Web-based courses, *Learning ArcIMS 4* and *Customizing ArcIMS 4,* are available from the ESRI Virtual Campus (campus.esri.com).