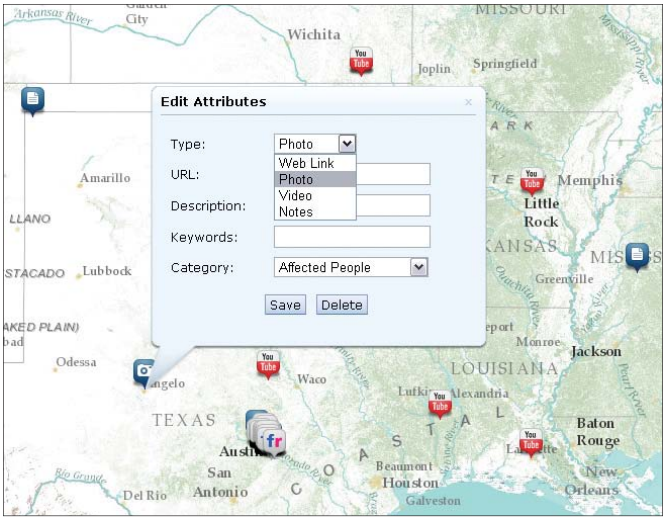
Dealing with High Demand

Building high-capacity mapping applications

By Keith Mann, Esri Writer



↑ The Public Information Map template can be configured to allow the user to add features to the map.

Is your mapping application ready for the big time? Maybe a better way to ask this question is, how well will the computing infrastructure that supports your mapping application stand up to a huge spike in demand?

Wildfires raging in Colorado in 2010 precipitated a real-life test for Esri. In response to the fires, Esri published a public information map (PIM) for wildfires. The wildfire app ran beautifully on a single server machine and seemed to be handling the daily traffic with no problem until a major news organization provided a link to the map in one of its stories. Immediately, demand went through the roof, and the server exceeded its load capacity. It shut down, and the wildfire app went dark.

The application was quickly reborn, but not on the single server machine. This time, it was set up to run in the cloud. The second incarnation of the wildfire app used cloud services to distribute demand across multiple servers. Additionally, the base structure for this type of system was established and configured so that server machines could be added or removed as needed.

In other words, a scalable system for supporting high-capacity mapping applications was created. This system is now used for all Esri disaster response and current events maps (esri.com/news/maps/index.html).

This article provides a brief, high-level view of the thought process used in building the infrastructure for a high-capacity mapping application. Although it's certainly not the only way to do this, it is a tried-and-true method that allows incremental system design while maintaining manual control over scalability.

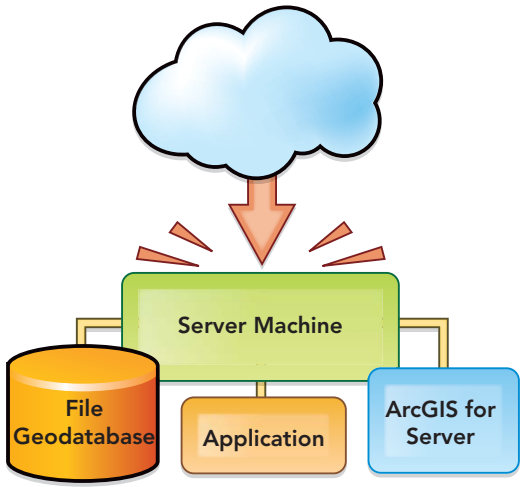
A Review of Amazon EC2

The strategy for adding capacity to the wildfire app uses Amazon Web Services (AWS)—specifically, Amazon Elastic Compute Cloud (EC2) (aws.amazon.com/ec2). Amazon describes EC2 as "a web service that provides resizable compute capacity in the cloud." Essentially, various components of Amazon's compute infrastructure can be rented to rapidly build a high-capacity support system.

The strategy also uses ArcGIS for Server to create and manage the GIS services for the application. Esri provides ArcGIS for Server as an Amazon Machine Image (AMI) running on Amazon EC2. The ArcGIS for Server on Amazon EC2 AMI includes 100 gigabytes of storage. These AMIs can be licensed in the same fashion that ArcGIS for Server is licensed on a local machine.

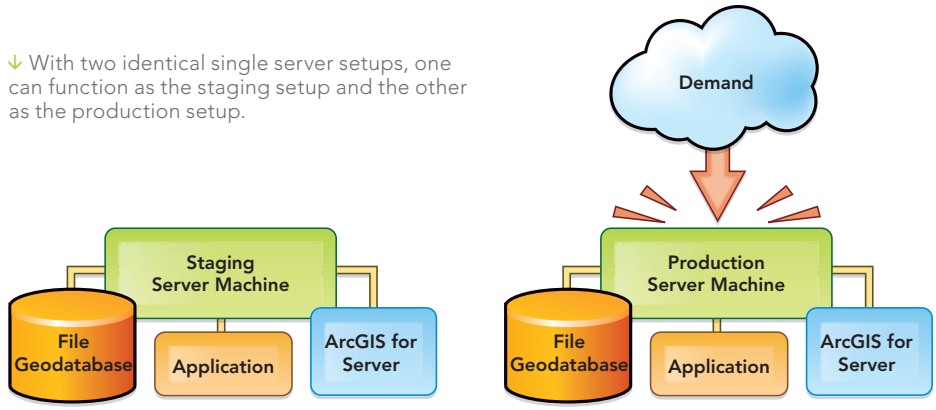
When you start an instance of ArcGIS for Server on Amazon EC2, you choose the ArcGIS for Server AMI, the size and type of machine it will run on, and other services and features such as Elastic Load Balancing and Auto Scaling.

You can upload your data and application to the instance's storage area and configure them with the ArcGIS for Server instance so that you can publish services and add them to the mapping application. Detailed resources for accomplishing these tasks are listed in the online version of this article.



↑ Spikes in demand from the Internet can quickly overload a single server machine setup.

With two identical single server setups, one can function as the staging setup and the other as the production setup.



The Single Server Setup

A common method for deploying a mapping application is to put everything—the mapping application, the GIS server, and the data—on one machine.

This method is often used for internal applications because machine usage is either known or can be safely predicted. However, when the application is exposed to the Internet, it is more difficult to predict the maximum load on the server.

Sudden spikes in demand can quickly overload the processing capability of the server machine and cause it to crash. During the resultant downtime, your customers lose access to the application, and you end up spending hours getting the server up and running, only to have it crash again as soon as users discover its availability.

One way to combat spikes in demand is to use a server machine with more processing power. The problem with this solution is that you end up paying for a server that is underutilized most of the time. This scenario is true whether you're deploying your application on local infrastructure or in the cloud.

The Staging Server and Production Server Setup

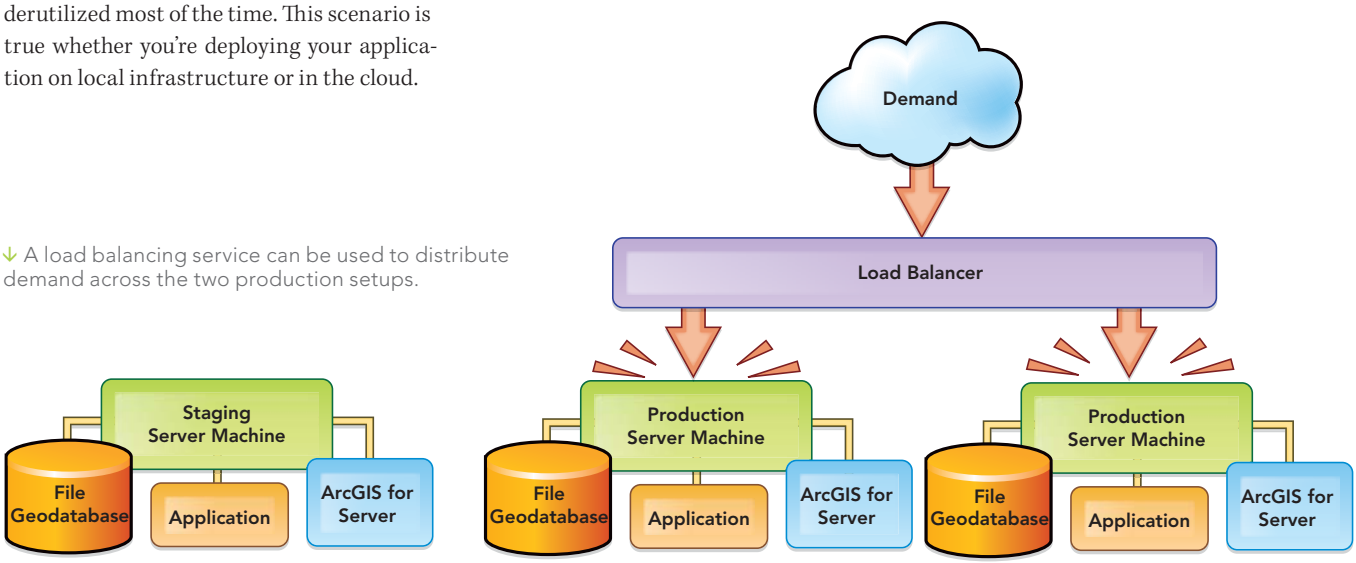
Another way to prepare for server overload situations is to make it easier and faster to spin up a new instance of your mapping application if it goes down. To accomplish this, first create a single server setup. Conceptually, you can think of this as your staging setup. Next, make a copy of your staging setup and *use the copy* as your production setup. End users only access the resources of the production setup. If demand overloads your production server, you make a new copy from the staging setup and use it to replace the crashed production setup.

On Amazon EC2, making a copy simply means starting a new instance from a custom AMI. To create the custom AMI, complete the staging setup and save it as an AMI. The custom AMI encapsulates your entire setup—the application, the data, and the GIS server. Creating a custom AMI can take 20 minutes to an hour. Once it's created, you can generate new instances from the custom AMI very quickly—in approximately 15 to 20 minutes.

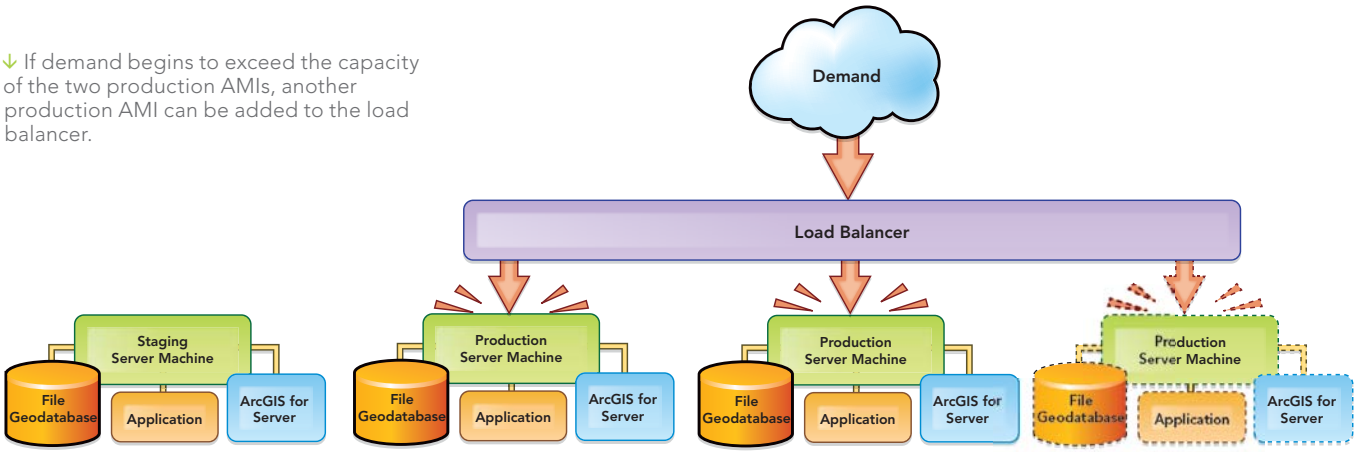
Having a staging setup lets you test your application and make edits before you deploy the production setup. If you want to add functionality to the application or make changes to the data and services, you can do this without disturbing the end users who are using the production setup. When you're ready to deploy the new setup, just kill off the production instance and replace it with the updated production instance.

However, this framework doesn't help you achieve the capacity needed to support large spikes in demand. To do that, you'll need two load-balanced production instances.

A load balancing service can be used to distribute demand across the two production setups.



If demand begins to exceed the capacity of the two production AMIs, another production AMI can be added to the load balancer.



Setting Up the Staging Server and Load-Balanced, Multiple Production Servers

In this scenario, you use the custom AMI created from the staging setup to generate two identical production instances. Next, you employ a load balancing service (called Elastic Load Balancing on Amazon EC2) that will distribute demand across the production instances.

If something goes wrong with one of the production instances, the application continues to work through the remaining production instance. This gives you time to start a new instance of the production instance and configure it with the load balancer.

You can also add additional machines to the load balancing service. If demand increases to a critical threshold—say, 80 percent on both production servers—you can start a new production instance from the custom AMI and add it to the load balancer. When demand subsides to normal levels, you can kill off the third production instance.

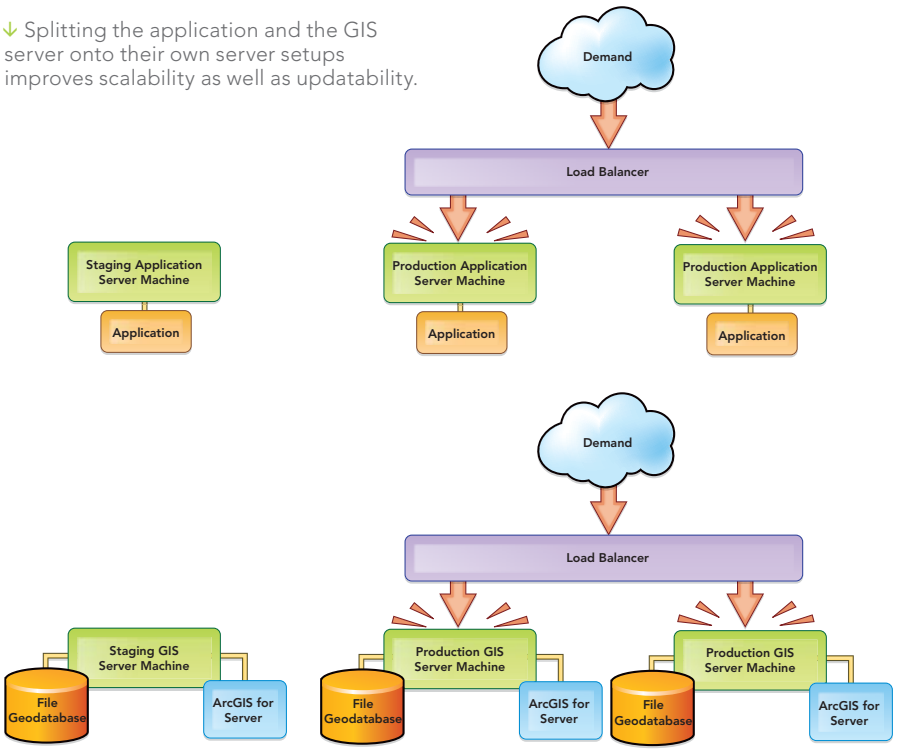
In this scenario, you are achieving a level of scalability that allows you to manually respond to spikes in demand that may affect the performance or viability of your mapping application. While the load balancer can be used to monitor whether the application is responding properly, it cannot check the fine-grained functionality of ArcGIS services. To more closely monitor your services, use ServiceMonitor, a script provided by Esri that checks each service periodically and makes sure it is responding. You can download the ServiceMonitor script at arcscripts.esri.com/details.asp?dbid=15335.

Multiple Staging Servers and Load-Balanced, Multiple Production Servers Setup

With most mapping applications, you'll want to make improvements, add functionality, or change the look and feel of the app on a regular basis. Changes to data and services usually occur much less frequently. You can take advantage of this phenomenon and make your high-capacity system more efficient if you put the application on a different server from the GIS server and data.

In this case, you make custom AMIs of the application server setup and the GIS server setup, create two new instances of each, and load balance the two production application AMIs and the two GIS server AMIs. This scenario provides scalability to your application as well as your GIS server and allows you to work on either while reducing the impact on your end users. Both the application and the GIS server setups can be scaled up and down depending on where the loads are greatest.

Splitting the application and the GIS server onto their own server setups improves scalability as well as updatability.



Including an Enterprise Geodatabase Server for Feature Editing

Some mapping applications allow the end user to perform feature editing such as adding a point, line, or polygon to the map. To have consistency on the production servers running in the load balancer, editing must be done in an ArcSDE database. The ArcSDE database should reside on a separate instance from the production instances. These features are stored in an enterprise geodatabase.

Esri also provides an enterprise geodatabase AMI. Data for editable feature services, is stored in an instance created from this AMI. After putting the data in the enterprise geodatabase, creating the feature service, and configuring the service with the application, you make a custom AMI of the staging data server setup and launch new production instances.

In this example, one staging data server and only one production data server are used to support the application. If possible, use two production data servers, each with a mirrored enterprise geodatabase for redundancy. However, currently, this is not a simple, out-of-the-box solution provided with the Esri enterprise geodatabase AMI—it requires a strong understanding of enterprise database management.

Where to Go from Here

Amazon offers an Auto Scaling service that you can configure with the load balancing service to automatically spin up new production instances when load thresholds reach a specified level. However, you may find that maintaining manual control over scaling of your production instances is preferable.

Amazon's cloud infrastructure and data centers are available in a number of regions around the world. Each region has multiple Availability Zones, designed to protect your instances from failure. Instances can be spread across Availability Zones within a Region so that if one data center goes down, the other instances will continue to function. The Amazon EC2 Service Level Agreement commitment is to provide 99.95 percent availability for each Amazon EC2 Region. As long as you create production instances in different zones within the same region, you can have them all behind the same load balancer.

Conclusion

To support a high-capacity mapping application, you will need a scalable system of identical setups, each with a GIS server; an application; and, possibly, data server machines. If you want to maintain your system in a manner that minimally impacts your end users, consider creating a staging AMI that you can work on while the production AMIs serve your end users. Ideally, there should be two load-balanced production instances in the base structure of the system so that if one fails, the application remains viable. If demand on both production instances exceeds tolerances, another production instance can be manually or automatically added to the load balancer.

If you anticipate frequent changes to the application, you may find it useful to put the application server and GIS server on separate tiers of the system. Likewise, if your application includes the ability to add or edit map features, you should put the required enterprise geodatabase instance on a separate tier.

These concepts and suggestions are not the only way to use Amazon EC2 to bring high capacity to your mapping application, but you may find that this model is a practical approach to maintaining fine-grained control over how and when your system scales.

↓ If the mapping application includes a feature editing capability, the data for those features should reside in an enterprise geodatabase.