

Resilient ArcIMS Architectures

John Cartwright
Ted Habermann
Ken Tanaka

Abstract

ArcIMS and geospatial databases form a critical part of the IT infrastructure at NGDC. However, these can be complicated systems with many potential points of failure. This paper will address the architecture, tools, and procedures that we have adopted to optimize the reliability and availability of internet mapping services.

Introduction

Internet map servers and geospatial databases form a critical part of the IT infrastructure at NOAA's National Geophysical Data Center (NGDC). These are complicated systems with many potential points of failure. The challenge for us has been to maximize the system reliability, and insure that, when failures do occur, they are either handled automatically, or at least that we are aware of them before our customers. We are using redundant architectures and automated monitoring to address this issue. Version control and remote administration also play an important role since the redundancy and the active on-going development efforts provide ample opportunity for service configuration files and viewers to get out of sync.

Current Architecture

The current ArcIMS architecture at NGDC was designed to maximize reliability and flexibility, primarily through redundancy. Performance is important of course, but up to this point it has not been our limiting factor. The “front-door” for all incoming HTTP Internet Mapping Service (IMS) requests is a single server (item 1 in Figure 1) running Apache web server and the Tomcat servlet container. Although this machine is a single point-of-failure for the entire system, it has been our experience that Apache and Tomcat on Linux are remarkably stable and this compromise presents a relatively low-risk. In the future we anticipate having dual servers at this point with a DNS-load balancer to allocate requests between them.

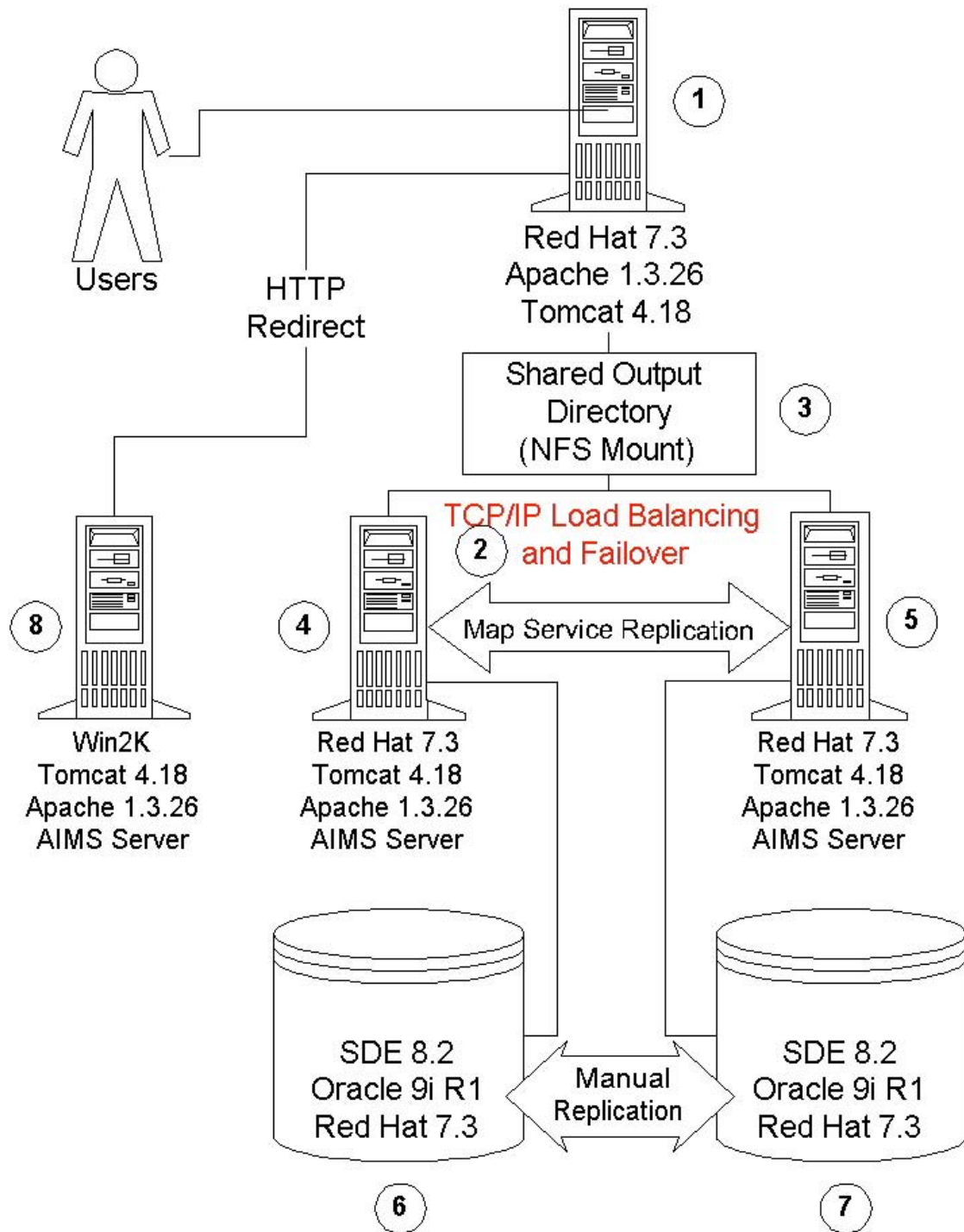
The ArcIMS Servlet Connector (item 2) runs on this machine and communicates with the Application Servers (items 4, 5) via TCP/IP. The Servlet Connector has an out-of-the-box capability that directs all requests to the primary machine (item 4) until a failure is detected. Then all subsequent requests go to the secondary machine (item 5). Once the Servlet Connector fails over, it will not automatically return to the primary machine after service has been restored. The servlet must be restarted to switch it back, and although this process could be automated, it happens rarely and is currently done manually.

The webserver (item 1) hosts the production HTML viewers and exports a directory (item 3) via NFS to the mapserver machines (items 4, 5). Map images and extract files are placed in this directory by the spatial server and are therefore available to the requesting client. The requesting client does not have direct access to the mapserver machines (items 4, 5).

The mapserver machines (items 4, 5) run both the ArcIMS Application Server and Spatial Server. All production mapservices are replicated on these machines and each of the mapservers has its own dedicated SDE server (items 6, 7 respectively). SDE layers are replicated between the two SDE instances. The process of replicating the mapservices and the SDE layers is only partially automated although we do have a few tools to assist in the verification. We anticipate further automating this process in the future.

The number of Spatial Server threads is adjusted on these two machines to accommodate their different processing capacities. Originally, we had a single instance of Application Server running on the webserver machine (item 1) and the Spatial Servers running on parallel machines (items 4, 5). This configuration is endorsed by ESRI, however this left the Application Server as a single point-of-failure, and at least with prior versions of ArcIMS, this did happen occasionally. One of the advantages of this configuration is that it facilitates system upgrades or the loading of large datasets. The Servlet Connector can be used to redirect all requests from one machine while it or its associated SDE server is in maintenance.

Finally, there is one standalone mapserver (item 8) that hosts mapservices that use a custom datasource (only available on Windows) or ArcMap Server. This machine runs Apache, Tomcat, ArcIMS Application Server, and Spatial Servers. Viewers that use these services are still hosted on the main webserver (1) but requests are redirected from there. The mapservices on this machine (item 8) are not replicated. Finally, there is a separate pair of servers (not shown in Figure 1) that host the development IMS and SDE servers.



Monitoring and Logging

Automated monitoring of the IMS is critical so that failures are detected and acted upon as soon as possible. We monitor on different levels for different types of problems. The most superficial is a plug-in for the opensource NetSaint monitoring framework (<http://www.netsaint.org>). This module periodically pings the ArcIMS application servers and integrates the results with the other network and server monitoring at NGDC.

We discovered that the most common IMS failures involve the connection between the Spatial Servers and the SDE servers. In this case the Application Server would usually continue to erroneously report that everything was all right even though map requests were failing. So this monitoring alone is not sufficient.

More detailed monitoring is accomplished through the use of a perl script, `aimsmonitor.pl`, which sends `<GET_IMAGE>` requests to each production mapservice on each server and parses the response. This approach has the advantage of most directly emulating the user request and, when successful, validates that all components of the system (webserver, servlet container, application server, spatial server, and SDE server) are all functioning. It has the side benefit of providing performance data on all the mapservices over time. This script is run by cron and gets its configuration information from a database table. If a number of mapservices on the same host fail to report successfully, the ArcIMS server on that host is automatically scheduled for restart. A second script then takes over and restarts the ArcIMS server (appserver, monitor, and tasker).

In addition to the status, performance, and restart logging done by the `aimsmonitor.pl` script, we use an opensource apache module, `mod_log_sql` (http://www.grubbybaby.com/mod_log_sql/), that logs to a database all map images delivered from the Spatial Server output directory. This allows easy reporting of the number of maps delivered for any given time period or mapservice. A variety of other web-based reports are available showing CPU load, IMS server statistics and status, webserver log summaries, SDE connections, etc.

MapServiceRegistry

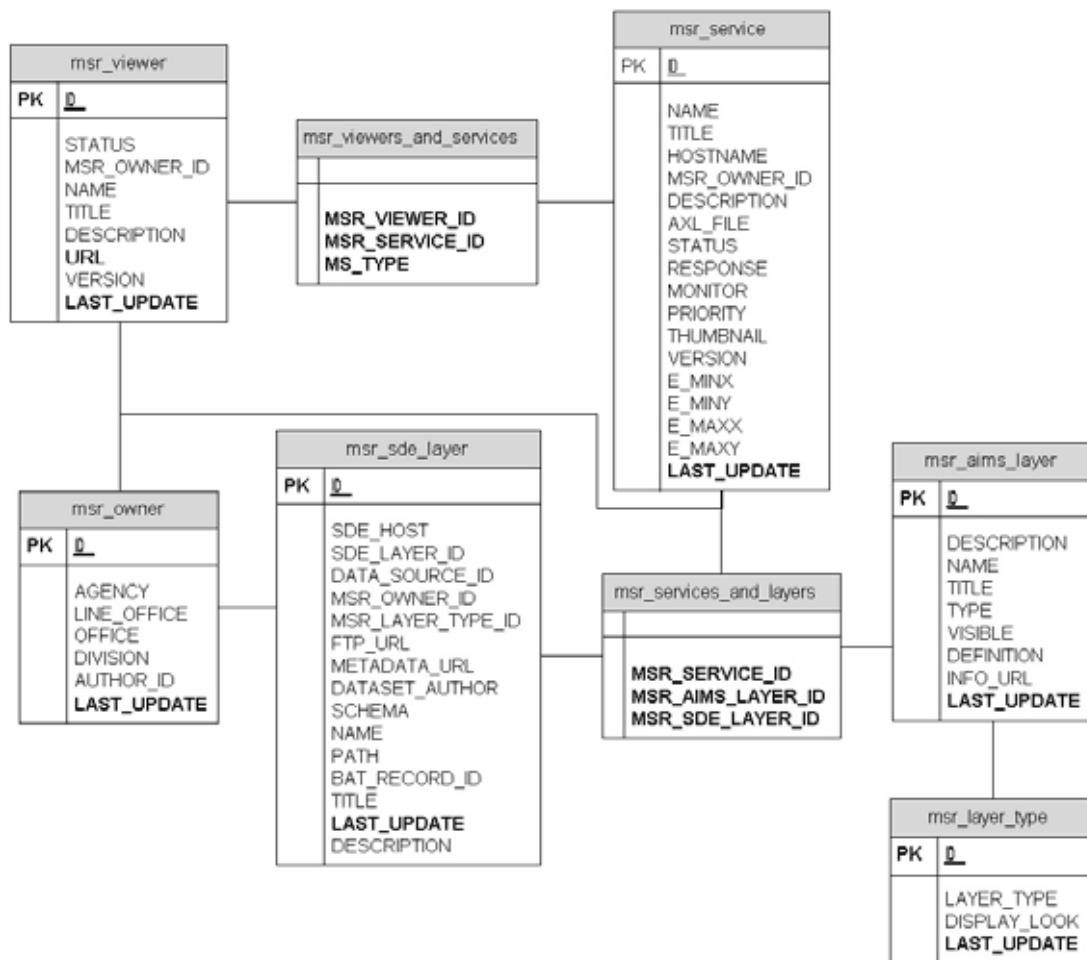
The MapServiceRegistry is a series of database tables (see Figure 2) that are central to the monitoring of and reporting on the various aspects of IMS at NGDC. These tables incorporate status and configuration information and describe the relationship between map viewers, mapservices, mapservice layers, and SDE layers/shapefiles. This allows us to answer a wide variety of questions like “what SDE layers are used by this mapservice?” or “what layers are visible by default in this mapservice?”

The database design is complicated by the fact that there are a number of many-to-many relationships between the entities. For example, a viewer may use one or more mapservices. Each of these mapservices contains one or more layers. Each of these layers uses a SDE layer or shapefile. A single SDE layer or shapefile however, can be used by multiple mapservices or even by multiple layers within a single mapservice. In our case, we also have the same mapservice potentially replicated on two servers. Parts of the MSR are populated automatically using the AXL file when a new service is created, and some fields (e.g. description, title) must be human-maintained. There is a web-based front end to assist the administrator in updating and querying the MSR.

The MSR_SERVICE table contains service metadata and is updated hourly by the aimsmonitor.pl script. The STATUS field contains a code indicating whether the service is currently up or down and the RESPONSE field holds the response time for the last <GET_IMAGE> request. It also contains the MONITOR and PRIORITY fields which are used by the aimsmonitor.pl script to determine whether to check a particular service and whether its failure is cause for alarm (or at least a server restart).

The index page listing ArcIMS viewers is periodically updated using the mapservice status stored in the database. In this way, a viewer using a mapservice whose status shows that it is down will not be listed on the index page and therefore minimize the chances of a user receiving an out-of-service error.

Map Service Registry



Remote Administration and Version Control

Because there are multiple IMS developers and administrators at NGDC, we use a version control system, CVS (<http://www.cvshome.org/docs/manual/>) . This allows us to track revisions to the AXL configuration files for the replicated mapservices. This is especially convenient when using ViewCVS, (<http://viewcvs.sourceforge.net/>) a web-

based front-end for CVS. HTML viewers are kept under source control as well as the SDE dbtune configuration files for each SDE server. In the future, we anticipate building a script that will periodically verify that each AXL file used by the mapservice is the current version as stored in CVS. The eventual goal is to have all files, whether HTML viewers or AXL configuration files “pass through” CVS as part of the process of being deployed into the operational environment.

For remote administration, we are using ESRI’s JSP-based esriadmin web application. We have extended it to allow editing of remote AXL files and are currently in the process of re-engineering it to use the Struts Framework and add additional AXL editing/validation capabilities as well as better integrate it with our MapServiceRegistry database.

Conclusion

Internet Map Servers and Geospatial databases are essential to our mission at NGDC. Like any complicated and loosely-connected system they can be susceptible to failures for a number of reasons, but the software itself continues to improve. Because of this improvement, and due to the architectures and monitoring that we have adopted, the reliability of our entire Internet Mapping System has improved dramatically over the last couple of years. Future challenges will likely center on version control and upkeep of the service metadata. Overall system performance may also become an issue as use of the sites increase, but scalability of the architecture should allow us to meet this need with the addition of hardware and software licenses.

John Cartwright
Associate Scientist
CIRES/NOAA-NGDC
325 Broadway E/GC1
Boulder, CO 80305-3328
303.497.6284
303.497.6513 (fax)
john.c.cartwright@noaa.gov