

Whitepaper: Monitoring the Interwoven Worksite DMS

By S. Bondy

Abstract

The document management system is crucial to the daily operation of the modern law firm. Yet frequently, the document management system is not monitored for health or signs of questionable user activity. This paper will outline some steps that can be taken to implement a monitoring system which can report database activities and statistics in a form accessible by any web browser. Optionally, e-mail alerts can also be configured.

Up on the twenty first floor, John Smith hangs up the phone. He has just accepted a position with a different law firm. After a quick check of his client list, he plugs a USB thumb drive into his PC, and exports over 500 documents from the firm's Interwoven Worksite document management system. Although these documents are the work product of the firm, he wants to take them with him to his new job, so he can try to persuade the clients to stay with him.

Down on the nineteenth floor, Jane Doe looks at her e-mail inbox. This weekend, the automated mailbox maintenance task will sweep her mailbox and clean out a large number of old messages. Some of these are important, and Jane needs to keep them. Instead of finding the messages she needs, Jane creates a folder in the document management system, and drops the entire contents of her inbox – all 10,000 messages – inside.

In the IT department, Sam gets a report from the helpdesk that the DMS is acting strangely. Checking on the database, Sam finds it has grown to the point that the server storage is almost filled. The DMS is on the verge of shutting down.

While these three scenarios may sound extreme, they are all representative of situations that have happened, in one form or another, to several law firms. But all these situations, while not necessarily avoidable, can be dealt with more effectively if key parameters of the document management system are closely monitored.

For example, what if within 5 minutes of John Smith exporting his documents, an IT staffer received an e-mail reporting unusual export activity on the DMS? The alerted staff member would then view a web page, showing a spike in activity for documents exported, and could further investigate the user responsible for the export.

What if the same kind of alert could be received for a sudden influx of new documents, such as Jane moving her entire, unfiltered inbox into the DMS? Or for a spike in database size? Fortunately, this is possible using freely available tools.

MRTG

Multi Router Traffic Grapher, MRTG, was first released in 1995. The author, Tobias Oetiker, wrote the software to track the performance of a link his company used to connect to the internet. Over time, he has improved the software greatly, and added many enhancements. It is now used widely, even by major ISPs, to chart the bandwidth usage of network links.

As originally intended, and most frequently used today, MRTG queries router hardware using Simple Network Management Protocol – SNMP. Using these queries, it can easily collect data on the bandwidth used by a network link. What is less well known is the ability of MRTG to monitor other values. This extension, using custom scripts, is the feature that is exploited in the development of the Interwoven monitoring system.

System Components

In addition to MRTG, the requirements for the monitoring system as implemented are straightforward: a Linux system with ODBC connectivity to the database, and some custom scripts.

While the monitoring system could have been developed on a Windows platform, using Linux affords us a few advantages. First, by using an open source operating system, the software investment to develop the system remains zero. This alone is a compelling justification, but in addition, the use of Linux enables us to develop the system on relatively low cost, desktop level hardware. In fact, in the course of developing the test system, a retired desktop PC with 128 MB of RAM proved more than adequate to the task.

For the scripts, perl was seen as an obvious choice. With some care, perl scripts are relatively portable, and could translate to a Windows environment if necessary. Also, the Comprehensive Perl Archive Network – CPAN – provides a variety of modules that greatly simplify the development of the scripts.

Development System

Configuring the development and testing system was straightforward. Linux was installed on a bare system. In this case, the Linux distribution used was Fedora, chosen for my familiarity with this particular distribution. The OS was installed with MRTG, perl, unixODBC, apache and development tools.

Unfortunately, one item missing from Linux is an ODBC driver which can connect to a MS SQL database. This proved to be the only significant stumbling block in building my test system. While Microsoft ODBC is a fairly proven and solid performer on Microsoft platforms, implementing a successful ODBC connection to a MS SQL Server from a Linux system proved a bit difficult until some research led to the discovery of a packaged ODBC driver compatible with Fedora. Once the driver was successfully implemented, what remained was to install the necessary perl modules, and begin development of my custom scripts.

The custom scripts that were developed are basically all variations of one model. In this model, a connection is established to the Interwoven database, a SQL query is executed, the results are collected, and data is output in the form expected by MRTG. The MRTG format actually expects 4 different items, which have a basis in MRTG's origins. According to the documentation, the values expected are bytes in, bytes out, up-time and target name. Because what MRTG actually graphs are the first 2 of these values, any script which queries the Interwoven database can query for and return 2 metrics, which will be plotted together.

In developing the scripts, the emphasis was on returning data that was meaningful relative to the system being monitored. The inference with this approach is that different scripts, collecting different values, may be required for different Interwoven databases. The test system developed graphs the following metrics:

Total number of documents/Total number of MIME documents
Documents printed within 5 minutes/Documents mailed within 5 minutes
Documents exported within 5 minutes/Documents deleted within 5 minutes
Documents opened within 5 minutes/Documents closed within 5 minutes
Documents created within 5 minutes/Versions created within 5 minutes
MS Word created within 5 minutes/WordPerfect created within 5 minutes
Database size/Database free space

The first pairing give us a big picture view of the database, and how fast it grows. It also shows us the relative growth of the number of e-mail messages versus the total, assuming the users are not changing the document type of e-mails to something other than the default.

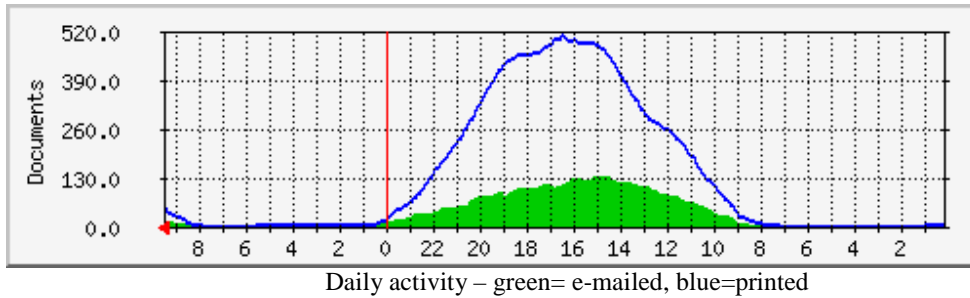
The second pairing is illustrative of the load that may be placed on print services, and e-mail services. Extreme spikes in either of these values may also indicate a security concern, if the spikes are caused by a user attempting to transport copies of firm work product off site.

The third pairing is another possible security metric, as it may reflect a user attempt to export documents onto removable media, or an attempt to purge large numbers of documents from the system. In some cases, the latter of these 2 metrics may also indicate a potential statutory violation, if the documents being deleted are subject to codified retention regulations, such as those specified in Sarbanes-Oxley, or those required by the Securities and Exchange Commission.

The remaining metrics are a mix of "interesting" values, and some that correspond directly to database growth and performance.

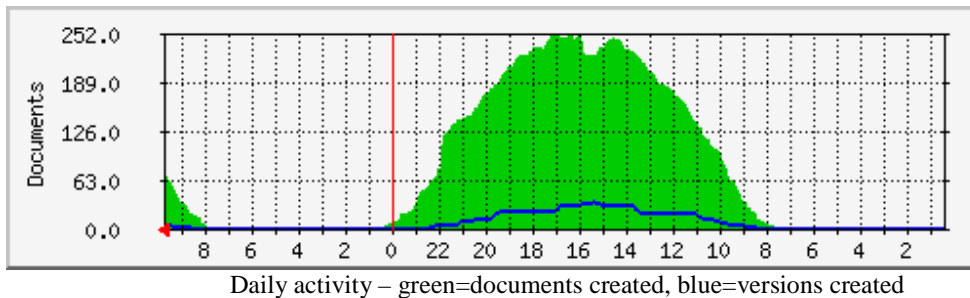
Results

After placing the system in production, and collecting data for a period of several weeks, I observed some interesting trends. Some of these were expected. For example, the daily graph of documents printed/mailed was unsurprising

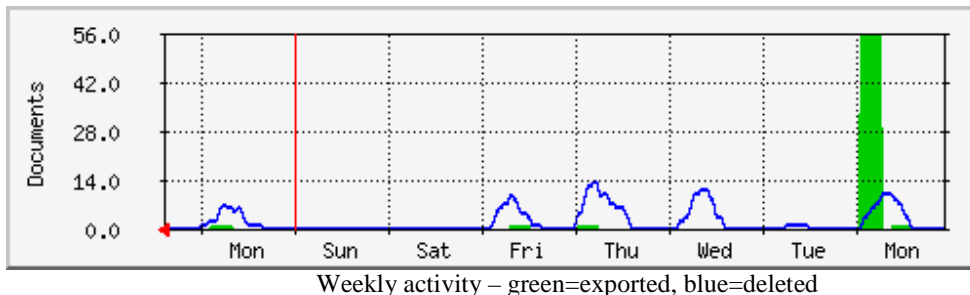


This graph basically reflects the activity in a normal workday. An increase in activity beginning coincident with the start of the business day, a rough plateau of activity between 3 and 6 PM, and a steady decline through the evening.

The daily graph of documents created showed a very similar, and again unsurprising, trend:



Once again, the graph reflects the trends of a normal workday, but it also provides us some idea of the rate that documents are being added to the system on a daily basis. More interesting results appeared with some other graphs, such as the weekly graph of documents exported and deleted:

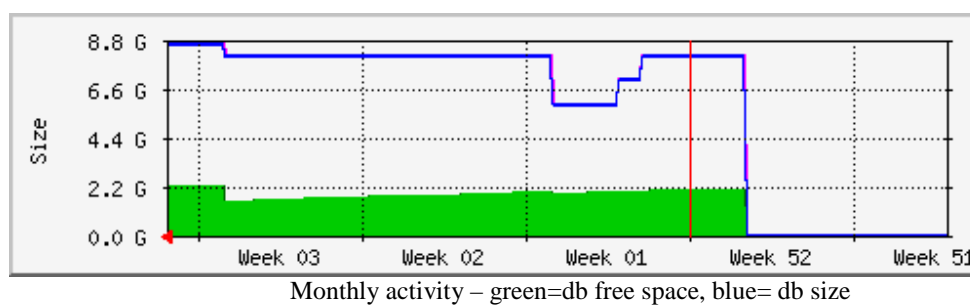


In this case, there is an obvious jump in documents exported late in the day on Monday. This type of spike might warrant an investigation, to determine the cause. In fact, in this

case, the spike was caused by a single user, who had tendered his resignation, exporting over 400 documents. A report of this activity was generated, and provided to the firm.

The number of documents in this case may be confusing, in relation to the scale on the left of the graph. This apparent discrepancy is due to the fact that the script queries the database every five minutes for activity that has occurred in a five minute interval. The flat topped spike thus indicates that for several samples, the number of documents per sample was at the top of the scale. Thus the total number of documents exported is the aggregate of all the samples during the spike.

One last example is a monthly graph of the database size and free space:



Here I was surprised to see that the database size was approaching 9 GB. Further investigation showed that SQL was configured to automatically grow the database by 10% when needed. While this may be appropriate with small databases, 10% of a large database represents substantial growth – in this case the next growth interval would have been approximately 800MB. The reduction shown was the result of scheduling database maintenance to compact the database. However, the growth configuration did not change, and once compacted, the database again jumped by a significant amount. This verified the monitoring systems ability to track changes in raw database size. As soon as possible, the database was then reconfigured to grow by a fixed amount when needed.

Enhancements to the System

Although I have not yet done so, the system is capable of more. Specifically, with a bit more work alerting could be added to any of the values that are being monitored. With such an addition, the system administrator could be notified by e-mail of thresholds being crossed in any of these counters. This could avert any one of several potential problems, including the export example above, the database size exceeding its optimum value, or the Interwoven repository exceeding the maximum recommended size.

Conclusions

Network monitoring in general is a challenging task for often overworked IT staff. Any tool that can ease the burden is worthy of consideration, especially if the tool has a small economic footprint. The Interwoven monitor system I have devised certainly fills that requirement, with its ability to be implemented with no-cost software and minimal

hardware. And with additional effort, the system could be modified to work with other enterprise systems which rely on Microsoft SQL as their database.