1 Monitoring and Tuning Performance

Introduction

This module is focused on the methods available to appraise system performance and the configuration parameters which can be tuned to optimize performance.

Objectives

By the end of this lesson, you will be able to:

- ▲ Identify the different methods available to monitor system performance
- ▲ View the different log files where Auspex-specific information is reported, and distinguish which files contain what information
- ▲ Optimize network, filesystem, and disk configuration

Performance Overview

Error and warning messages are the most basic indicator of performance

The ASMT Performance Monitor Tool and ax_perfmon offer the ability to monitor system performance in real-time

Performance tuning is generally a matter of properly configuring Auspex devices

Performance Overview

Performance Overview

This module covers the methods available to monitor system performance as well as some actions which can be taken to remedy shortcomings in performance. Some of the tools for performance monitoring have been covered in previous sections, specifically the Event Policy Manager and Performance Monitor Tool components in ASMT, and will not be covered in detail here. Please refer back to the ASMT module for detailed information.

The most basic indicator of performance comes in the form of error and warning messages found in the error logs. These error logs will be covered, as well as the configuration file used to control which messages will appear in what files, syslog.conf.

Performance monitoring with real-time measurement tools will then be discussed briefly, as the associated tools have either already been covered (ASMT) or should be familar to you (ax_perfmon).

Finally, performance tuning methods will be discussed. Most of this material will be a review of configuration options discussed in prior sections. This module will reiterate this information with the goal of optimizing the performance of a given hardware setup.

Error Logging

Error and warning messages are sent through the syslog facility to the proper error log or notification mechanism (console message, email, etc.)

/var/adm/messages contains all error level messages and messages generated by the kernel and system daemons

/var/log/auspex-messages contains only messages from Auspex components

/etc/syslog.conf specifies where the different types and severities of error messages are sent

syslog.conf has only been modified to send Auspex messages to /var/log/auspex-messages

Error Logging

Error Logging

The syslog facility provides a common mechanism for the operating system and programs to report warnings and errors. As distributed by Auspex, the configuration file (syslog.conf, discussed later) is set to deliver the bulk of significant messages to /var/adm/messages and Auspex-specific messages to /var/log/auspex-messages.

/var/adm/messages

/var/adm/messages contains all error level messages (including those generated by Auspex components), as well as messages generated by the kernel and system daemons. Note that this is the same as in Solaris, with the addition of Auspex messages.

/var/log/auspex-messages

/var/log/auspex-messages includes only messages from Auspex components. This provides a means of “filtering out” standard UNIX messages when attempting to locate a possible problem with Auspex hardware or software, but note that a UNIX problem reported in /var/adm/messages may manifest itself to a client as an “Auspex problem”, as would be the case where the mount or lock daemon could not get processor time due to some other runaway process.

/etc/syslog.conf

Where messages are reported is determined by the file /etc/syslog.conf. As distributed, Auspex configuration has only been appended to the bottom of the file (to send Auspex-related messages to /var/log/auspex-messages). If you wish to change which messages are reported to whom, please consult the syslog.conf(4) and syslogd(1) manpages.

Performance Monitoring

The ASMT Performance Tool can render both table and graphic formats of system performance data points, but cannot store this data for later viewing

ax_perfmon

ax_perfmon is a curses based performance monitoring tool

ax_perfmon is capable of capturing data for later replay

The screens in ax_perfmon have changed to reflect new hardware and software architecture

Performance Monitoring

Performance Monitoring

There are two system-wide performance monitoring utilities, ax_perfmon and the ASMT Performance Tool.

ax_perfmon is a curses-based performance monitoring tool. It is functionally identical to the ax_perfmon of the existing NetServer line, and therefor will not be covered in extensive detail. ax_perfmon offers the ability to capture data for later replay, which is ideal in situations where a general performance problem is perceived at the client end, but the cause on the server is not readily identifiable. Data can be collected and sent to Auspex support for detailed analysis.

The ASMT Performance Tool is a web-based performance monitoring tool, as described in a previous module. While the ASMT Performance Tool can render both table and graphical formats of performance information, note that at this time it is not capable of storing data for later viewing. For more information on the Performance Tool, please see the ASMT module.

ax_perfmon

The ax_perfmon utility has been ported from the existing NetServer product line. As with the previous version of ax_perfmon, there is a summary screen which gives a brief view of all the system components, and then there are separate screens which cover different aspects of system performance. This module will only high-light the changes in the new version of ax_perfmon.

The most noticable changes are those which have been made to accomodate the new hardware architecture of the M2000, that is the FSP and NP components rather than an NP, FP, and SP. The FSP screen now can scroll through the three possible Mylex controllers, and each screen has a section for each of the three channels on the Mylex.

As the software running on the NP and FSP has changed, there are now screens cover buffer and paging statistics.

Performance Tuning

Performance tuning is the process of identifying and removing performance bottlenecks

The three main bottlenecks for the M2000 are the network, disk access, and the filesystem

Note that each of these are presented in screens of ax_perfmon, which makes it an ideal tool to narrow down the search for performance problems

Network problems

Network problems are usually configuration issues between the Auspex and the attached network equipment

It is highly recommended that autonegotiation not be used for fast ethernet connections

Network Tuning

Performance Tuning

Obtaining optimal performance from any system requires the identification and removal of performance bottlenecks. For the purposes of our discussion, the impediments to NFS performance for the M2000 fall into three main categories, which roughly parallel the logical components of the Auspex Functional Multiprocessing (FMP) architecture:

- ▲ Network problems (network processor)
- ▲ Disk access problems (FSP/Mylex)
- ▲ Filesystem problems (FSP/LFS)

Each of these three parts of the performance puzzle are presented in an ax_perfmon screen, which makes it an excellent starting point in narrowing down the search for a bottleneck.

Network Problems

Problems involving the network are generally configuration issues between the Auspex and the network to which it is attached. No tuning parameters exist beyond general configuration issues such as link speed and duplex mode(fast ethernet) and off-net MTU. In both cases, failure to set these correctly generally results in near-loss of connectivity. While the primary intent of this module is to highlight more subtle changes which result in significant gains (i.e. performance tuning vs. fixing clearly incorrect configurations), it is worth noting the following: it is highly recommended that autonegotiation **not** be used; instead you should set your interface to the desired speed and mode in /usr/AXbase/etc/iftab and also hardwire your switch to match this.

Disk Access Problems

Bottlenecks occur in the disk access area when filesystems are laid out in sub-optimal fashions. The key to preventing bottlenecks is to reduce contention by striping filesystems accross multiple disks. While this is nothing new, the M2000 now allows filesystems to span multiple Mylex

Performance Tuning

Disk Access Problems

These problems are usually a result of poor layout, so care should be taken when planning how disks will be assigned to RAID arrays

Reduce contention by striping across channels

Now it is possible to effectively stripe across Mylex controllers by creating virtual partitions containing RAID array slices from different Mylex controllers

Specify write back caching when creating RAID arrays to ensure intelligent cache usage

Disk and Filesystem Tuning

controllers, by defining a virtual partition which consists of RAID arrays located on different controllers.

The write buffer/cache is also specified differently in the M2000. While it cannot be turned off, a write back or write through cache must be selected. Write through caching simply flushes items to disk as soon as possible, while write back will flush in a more intelligent manner, optimizing disk accesses. Also note that write back/write through caching is specified when RAID arrays are created (as opposed to when filesystems are mounted on the existing NetServer).

Filesystem Problems

The HTFS filesystem used by the M2000 presents a number of tunable filesystem parameters, which are detailed in the mount_lfs(1) manpage. The ones that are most relevant to this module are nochkpt and fastsync.

With checkpointing turned on, filesystem that are perceived as idle by the FSP will be completely flushed. Turning off checkpointing may improve filesystem performance in bursty environments. When data tends to come in separated bursts the situation may arise where the FSP begins to flush data just as a burst of writes arrives, resulting in poorer response times to this latest batch of writes.

Fastsync creates a larger sync log (a metadata cache which exists on disk), 2 MB vs. the standard 32 KB. There is a correspondingly increased allocation of memory for metadata, 256 KB vs. 32 KB. This is quite advantageous for filesystems that see a large number of metadata operations (as in a build environment), as they are able to reduce disk reads for metadata. However, filesystems which see more bulk reads and writes (geophysical data installations, for instance) do not benefit from the increased memory set aside for the metadata. Indeed they might suffer, as this metadata memory is drawn from the same reserve (memory on the ECHaP) as the regular user data cache. In short, careful consideration is warranted when tuning filesystems with with the fastsync parameter.

Performance Tuning

Filesystem Problems

The HTFS filesystem presents probably the most tunable component of the M2000

Most tunable parameters are specified when mounting the filesystem, and are detailed in the mount_lfs(1) manpage

Checkpointing flushes writes to disk when a filesystem is perceived as idle, which may cause problems if data tends to come in bursts

Fastsync creates a larger sync log (a metadata cache on disk), and a larger portion of the FSP memory is set aside for the metadata cache

This larger metadata cache will increase performance for environments seeing lots of attribute operations (build environments), but those that are doing more straight-forward reads and writes of big files are better served by having memory allocated to the user data cache

Filesystem Tuning