Advanced SUSE Linux Enterprise Server Administration (Course 3038)

Chapter 8

Perform a Health Check and Performance Tuning

Objectives

- Find Performance Bottlenecks
- Reduce System and Memory Load
- Optimize the Storage System
- Tune the Network Performance

Find Performance Bottlenecks

- Questions that can help you to find the performance bottleneck
  - What kind of server is affected?
  - What are the exact symptoms of a problem?
  - Does the problem occur at specific times of the day or the week?
  - When and how did the problem start?
  - Who is experiencing the problems?
  - Can the problem be reproduced?

Find Performance Bottlenecks (continued)

- Objectives
  - Analyze Processes and Processor Utilization
  - Analyze Memory Utilization and Performance
  - Analyze Storage Performance
  - Analyze Network Utilization and Performance

Analyze Processes and Processor Utilization

- You should look at the processor utilization
- Measure processor utilization using the system load
  - Use commands such as top or uptime
- Linux is a multiprocessing operating system
  - CPU can handle one process at a time
  - Kernel puts the running processes into a queue
  - Load value
    - Average number of waiting processes in the process queue in a specific amount of time
- A process does not always require CPU time
  - CPU waits for I/O processes

Analyze Processes and Processor Utilization (continued)

Table 8-1

Program	Description
top	Displays a sorted list of applications and the three values for the average load values in the last 1, 5, and 15 minutes. When you find that your system has a high load value, top can also be very helpful to find out which application is actually producing it.
uptime	uptime can also be used to display the system load in the last 1, 5, and 15 minutes.
mpstat	On multiprocessor systems, mpstat can be used to display the utilization of each installed processor.
KDE System Guard	KDE System Guard displays a graphical representation of the system load.

Analyze Memory Utilization and Performance

- Applications have to be loaded into memory
  - Before they can be executed by the CPU
- Memory is controlled by the Memory Management system of the Linux kernel
- Types of memory
  - Physical memory
  - Swap memory
- Command free
  - Displays utilization of physical and swap memory

Analyze Memory Utilization and Performance (continued)

- Command free output example

```
             total       used       free     shared    buffers   cached
Mem:        516204     502080      14124          0      29356   154920
-/+ buffers/cache:     317804     198400
Swap:      1036152     143320     892832
```

- Performance of the whole system is affected
  - When a lot of swap space has to be used
- Use top command to find programs that use a lot of memory
- Command vmstat
  - Displays the activity of swap memory

Analyze Memory Utilization and Performance (continued)

- Command vmstat output example

```
procs  --------memory---------- -swap- --io-- -system- ----cpu----
 r  b  swpd  free  buff  cache  si  so  bi bo   in  cs us sy id wa
 0  0     4  6728 34464 244744   0   0 447 42 1216 384 15  3 74  7
 0  0     4  6728 34464 244744   0   0   0  0 1186 222  1  1 98  0
 0  0     4  6760 34464 244744   0   0   0  0 1282 299  3  0 97  0
 0  0     4  6696 34532 244744   0   0   0 68 1139 147  1  1 97  1
 0  0     4  6696 34532 244744   0   0   0  0 1105 123  0  0 98  0
 0  0     4  6696 34532 244744   0   0   0  0 1117 131  0  0 98  0
```

Analyze Memory Utilization and Performance (continued)

Table 8-2

Program	Description
free	Displays the current utilization of the physical and swap memory.
vmstat	Monitors the activity of swap memory and can also be used to display other system parameters.
KDE System Guard	Offers the capability to display memory usage. Choose the signal plotter visualization to follow the memory usage over a period of time.

Analyze Storage Performance

- Can be an issue
  - On systems that face heavy hard disk utilization
- Discard problems with a too-high system load
  - Or a lack of physical memory
- System where performance problems are caused by the disk subsystem usually shows
  - Relatively low network and CPU utilization
  - High activity of the installed disks
    - That is not caused by memory paging or swapping

Analyze Storage Performance (continued)

- Command vmstat
  - Displays the activity of the disk subsystem
- Example: almost no disk operations

```
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id  wa
 0  0      4   6728  34464 244744    0    0   447    42 1216   384 15  3 74   7
 0  0      4   6728  34464 244744    0    0     0     0 1186   222  1  1 98   0
 0  0      4   6760  34464 244744    0    0     0     0 1282   299  3  0 97   0
 0  0      4   6696  34532 244744    0    0     0    68 1139   147  1  1 97   1
 0  0      4   6696  34532 244744    0    0     0     0 1105   123  0  0 100  0
 0  0      4   6696  34532 244744    0    0     0     0 1117   131  0  0 100  0
```

Analyze Storage Performance (continued)

- Example: high utilization of the disk subsystem

```
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
 1  2     52   5680   6100 221688    0    0     0 36160 1273  1655 42 58  0  0
 0  3    304   6896   1232 225672    0  256     4 22160 1586  1127 31 40  0 28
 1  2    304   5936   1252 226540    0    0     0 28400 1487   460 15 23  0 62
 1  0    304   7792   1276 224404    0    0     0 43328 1342   408 20 29  0 51
 1  2    304   6256   1624 224648    0    0     0 88260 1205   439 24 42  0 35
 0  2    476   6648   1672 224112    0  172     4 45452 1149  8015 29 54  0 17
 0  2    476   7672   1720 223184    0    0     8 36940 1168  8310 23 44  0 33
```

- Command iostat
  - Determines the average time a program has to wait for data from the disk
  - Syntax
    - iostat -x 1 /dev/hda

Analyze Storage Performance (continued)

- Example: low I/O load

```
avg-cpu:  %user   %nice    %sys %iowait   %idle
           8.08    0.04    1.73    1.70   88.45

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
hda          3.18  17.90  3.37  1.32  146.73  153.78    73.36    76.89    64.11     0.25   53.50   4.57   2.14

avg-cpu:  %user   %nice    %sys %iowait   %idle
           4.90    0.00    0.98    0.00   94.12

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
hda          0.00   0.00  0.00  0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00

avg-cpu:  %user   %nice    %sys %iowait   %idle
           5.05    0.00    0.00    0.00   94.95

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
hda          0.00   0.00  0.00  0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00   0.00
```

Analyze Storage Performance (continued)

- Example: high I/O load

```
avg-cpu:  %user   %nice    %sys %iowait   %idle
          26,00    0,00   45,00   29,00    0,00

Device:  rrqm/s wrqm/s    r/s   w/s  rsec/s  wsec/s   rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
hda        0,00 9198,00  4,00 39,00   32,00 73872,00  16,00 36936,00  1718,70   103,83 1430,33  23,28 100,10

avg-cpu:  %user   %nice    %sys %iowait   %idle
          20,79    0,00   39,60   39,60    0,00

Device:  rrqm/s wrqm/s    r/s   w/s   rsec/s  wsec/s   rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
hda        0,00 9105,94  0,00 44,55     0,00 73140,59   0,00 36570,30  1641,60    99,97 2441,89  22,24  99,11

avg-cpu:  %user   %nice    %sys %iowait   %idle
          26,26    0,00   45,45   28,28    0,00

Device:  rrqm/s wrqm/s    r/s   w/s   rsec/s  wsec/s   rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
hda        0,00 10313,13 0,00 41,41     0,00 82828,28   0,00 41414,14  2000,00    93,90 2529,10  24,41 101,11

avg-cpu:  %user   %nice    %sys %iowait   %idle
          24,00    0,00   48,00   28,00    0,00

Device:  rrqm/s wrqm/s    r/s   w/s   rsec/s  wsec/s   rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
hda        0,00 9293,00  0,00 41,00     0,00 74640,00   0,00 37320,00  1820,49    92,70 2447,00  24,41 100,10
```

Analyze Storage Performance (continued)

Table 8-3

Command	Description
vmstat	Monitors the amount of data that is read from or written to disk.
iostat	Displays how long I/O requests from applications take.

Analyze Network Utilization and Performance

- Network connection can be a performance bottleneck
- Monitor network with KDE System Guard
- Types of sensors
  - Receiver
  - Transmitter
- Network services can interfere with performance
  - DNS
  - Proxy
  - NFS

Analyze Network Utilization and Performance (continued)

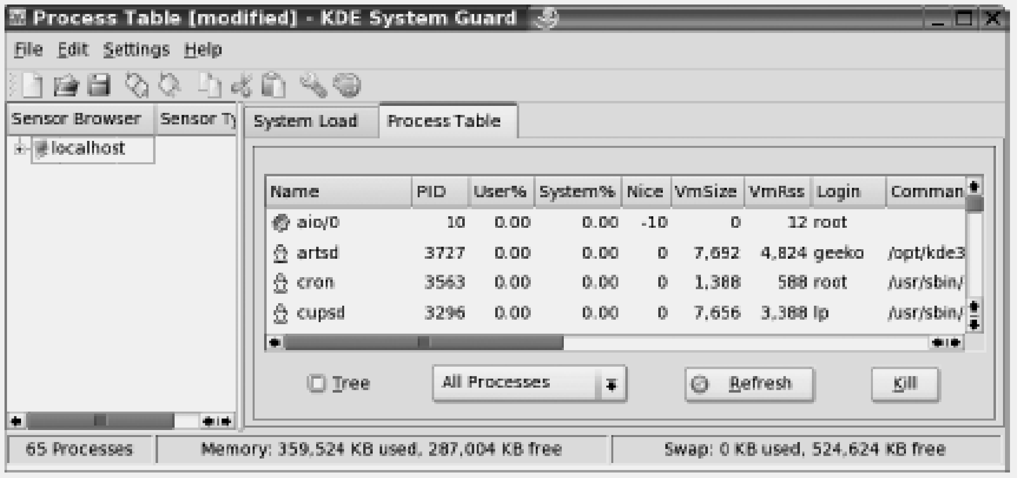

Figure 8-1

Analyze Network Utilization and Performance (continued)

Table 8-4

Sensor	Description
Data/Packets	The amount of data or packets sent or received by the interface. If performance problems occur during a high network load, the network connection or type might be too slow for the purpose of the server.
Collisions	This sensor is only available for the transmitter. Collisions usually occur more frequently when too many hosts share the same Ethernet domain (such as hosts that are connected with a hub and instead of a switch). Too many collisions can have a negative impact on the overall network performance.
Dropped Packets	This sensor displays the number of packets that are either dropped when they are received by the host or by other network components like routers on their way to the destination. Too many dropped packets can have a bad influence on the network performance. The following are some reasons for dropped packets: • Network components are running at a different speed. For example, the server runs at 100 Mbps, but the router at only 10 Mbps. • The network or system load of a server is too high to handle all received network packets properly. • A network component runs with a misconfigured packet filter that drops network packets.
Errors	An error occurs when a packet is transmitted but the content of the packet is corrupted. This can be caused by a bad physical connection or faulty network adapters.

Analyze Network Utilization and Performance (continued)

Table 8-5

Program	Description
KDE System Guard	Displays network utilization and different kinds of transmission errors.
Traffic-vis	Analyzes network connections to specific hosts. You need to install the package traffic-vis in order to use this tool.
ip	Displays the status of an interface as well as transmission errors.

Exercise 8-1 Analyze System Performance

- In this exercise, you will do the following:
  - Part I: Analyze Processor Utilization
  - Part II: Analyze Memory Utilization
  - Part III: Analyze Hard Disk Utilization
  - Part IV: Analyze Network Utilization

Reduce System and Memory Load

- Objectives
  - Analyze CPU Intensive Applications
  - Run Only Required Software
  - Keep Your Software Up To Date
  - Optimize Swap Partitions
  - Change Hardware Components

Analyze CPU Intensive Applications

- High system and memory load
  - Often caused by single application
- top utility
  - Used to find out which process uses the most resources on your system
- Sometimes a process uses a lot of system resources
  - Because of a faulty implementation
  - Determine this by restarting the process

Run Only Required Software

- Run a server system without X
  - Saves memory and CPU utilization
  - Switch to runlevel 3 manually
    - init 3
  - Set the default runlevel to 3 in file /etc/initab
    - id:3:initdefault:
- Reduce the number of daemon processes
  - Get an overview of the current service configuration
    - chkconfig –l
  - Remove a service from the init process
    - chkconfig apache2 off

Run Only Required Software (continued)

- Reduce the number of daemon processes (continued)
  - Re-enable a service
    - chkconfig apache2 3
  - Stop a running instance of apache2
    - rcapache2 stop

Keep Your Software Up to Date

- Reasons
  - Security issues caused by outdated software
  - Up to date software can improve performance
- Implementation errors
  - Can lead to a high utilization of system resources

Optimize Swap Partitions

- On a system with a lot of swapping
  - Add more main memory to enhance performance
- Make sure you have enough available swap space
  - Old rule: you should have double the size of the physical memory as swap space
- Key to speeding up the swap space
  - Spread it over several disks
- Every swap partition has an entry in the file /etc/fstab
  - Priority 1 means kernel can use partitions in parallel
  - Drives should run at the same speed

Change Hardware Components

- Upgrade the CPU
  - If your system shows a high system load
    - But all other parameters look normal
  - Consider the following before upgrading the CPU
    - Are there significantly faster CPUs available?
    - Are the rest of the system components fast enough for the new CPU?
    - Is the system going to be replaced in the near future?
    - Are other, faster systems available in your organization that could be used instead of the current system?

Change Hardware Components (continued)

- Upgrade the memory
  - Usually means installing more physical memory
  - How much additional memory you should install
    - Look at the amount of swap space used by the system
      - When the performance problems occur
  - Compare the cost of a memory upgrade
    - With the cost of installing a new system
  - Additional physical memory means more swap space
    - More than 1 GB of swap space does not increase performance significantly

Exercise 8-2 Reduce Resource Utilization

- In this exercise, you will practice reducing resource utilization

Optimize the Storage System

- Objectives
  - Configure IDE Drives with hdparm
  - Tune Kernel Parameters
  - Tune File System Access
  - Change Hardware Components

Configure IDE Drives with hdparm

- Tool hdparm
  - Tunes some settings of IDE hard drives
  - Syntax
    - hdparm -i /dev/had
- Direct Memory Access (DMA)
  - Data from a disk can be written directly to the main memory of a system without CPU utilization
- Check the current status of the DMA
  - hdparm -d /dev/had
- Enable DMA
  - hdparm -d 1 /dev/hda

Configure IDE Drives with hdparm (continued)

Table 8-6

Parameter	Description
-c 1	Enables 32-bit transfers of disk data over the PCI bus.
-u1	A setting of 1 permits the driver to unmask other interrupts during processing of a disk interrupt, which greatly improves Linux's responsiveness and eliminates serial port overrun errors.
-X *value*	Configures the drive to use a specific transfer mode.
-A 1	Enables read-ahead, which increases performance when dealing with large, sequential file operations.

Configure IDE Drives with hdparm (continued)

- Measure the transfer performance of a hard disk
  - hdparm -t /dev/had
- All changes that are made with hdparm
  - Are active only until the next reboot
- Make changes permanent by adding them to /etc/init.d/boot.local

Tune Kernel Parameters

- Tune the IO scheduler
  - Collects requests from the processes and hands them over to the hardware drivers
  - Configure scheduler parameter in file
    - /sys/block/*device*/queue/iosched/quantum
    - Use echo 6 > /sys/block/hda/queue/iosched/quantum
  - Tradeoff between data throughput and latency
    - Lower value = Shorter latency but lower data throughput
    - Higher value = Longer latency but higher data throughput

Tune Kernel Parameters (continued)

- Change the read-ahead parameter
  - Read-ahead basically means
    - More data from a file is read than requested by an application
  - Set the read-ahead parameter in the file
    - /sys/block/*device*/queue/read_ahead_kb
    - Use echo 256 > /sys/block/device/queue/read_ahead_kb
  - Larger values can lead to a better overall throughput
    - With the drawback of a higher latency

Tune Kernel Parameters (continued)

- Change the swappiness parameter
  - Affects both the memory and the I/O performance
  - Determines when a system starts to swap out data to the disk
  - Can be set in the file /proc/sys/vm/swappiness
    - Use echo 40 > /proc/sys/vm/swappiness

Tune File System Access

- Disable atime update
  - For every file Linux stores the following information:
    - When the file was created (ctime)
    - When the file was modified the last time (mtime)
    - When the file was accessed the last time (atime)
  - Kernel needs to update the atime attribute
    - Every time a file is accessed
  - If the atime attribute is not important to you
    - You can mount a data partition with the noatime option

Tune File System Access (continued)

- Implement File System Dependent Tuning Options
  - Mount a Reiser File System with the notail option
    - Syntax

```
/dev/hda2       /data   reiserfs    notail          0 0
```

  - Configure the journaling mode of Ext3
    - Modes
      - data=journal
      - data=ordered
      - data=writeback
    - Syntax

```
/dev/hda2       /data   ext3    data=writeback    0 0
```

Change Hardware Components

- From a performance perspective
  - A true SCSI hardware RAID system might be the best choice
- Compare the costs and the estimated advantages of an upgrade
  - With the purchase of a new system
- Hardware upgrade always has the risk of creating a new performance bottleneck

Exercise 8-3 Tune an IDE Hard Drive with hdparm

- In this exercise, you will tune your IDE hard drive.

Tune the Network Performance

- Objectives
  - Change Kernel Network Parameters
  - Change Your Network Environment

Change Kernel Network Parameters

- Parameters can be set with the sysctl command
  - You have to be the root user
  - The most important command line parameter is -w
- You can also access the kernel parameters from the proc file system
  - Which is mounted under /proc

Change Kernel Network Parameters (continued)

Table 8-7

sysctl command	Effect
sysctl -w net.ipv4.tcp_tw_reuse=1 sysctl -w net.ipv4.tcp_tw_recycle=1	When a TCP connection has been closed, the corresponding socket stays in the TIME-WAIT status for a while. Setting these two parameters enables the reuse of these sockets for new connections. On a system with many TCP connections, this can reduce the number of open connections and the utilization of system resources.

Change Kernel Network Parameters (continued)

Table 8-7 (continued)

sysctl command	Effect
sysctl -w net.ipv4.tcp_keepalive_time=900	TCP connections are usually kept alive for a specific amount of time. After this time period, a system probes to see if the connection partner is still reachable. If not, the connection is closed and the used resources are freed. The default time for SLES 9 is 1800 seconds. By reducing this time, you can reduce the number of opened but unused connections.

Change Your Network Environment

- Suggestions for improving network performance
  - Monitor all other system components
  - Limit the collision domain
  - Check cable quality
  - Check both sides of a connection
  - Change network adapters
  - Upgrade to a faster network type

Summary

- Regular monitoring of your SLES system
  - Key to identifying bottlenecks
- Monitor the system load and CPU utilization
- Processes that use more physical memory than they need (called a memory leak)
  - Lead to reduced performance
- Disk subsystem may be a bottleneck
- Malfunctioning, slow, or misconfigured network interfaces can cause performance bottlenecks

Summary (continued)

- DNS, Proxy, and NFS servers on your network
  - Can indirectly lead to reduced network performance
- Common solutions to most performance problems
  - Restarting CPU-intensive processes
  - Running SLES without X
  - Reducing the number of running daemon processes
  - Obtaining software updates
  - Using multiple swap partitions
  - Upgrading the CPU
  - Increasing the swap
  - Upgrading the system memory

Summary (continued)

- Many disk-related performance problems
  - Can be solved by changing a kernel parameter
- Can disable atime updates for mounted file systems
  - And configure ext3 journaling modes
- If software-based solutions do not lead to improved disk performance
  - You will need to upgrade your disk hardware
- Many network-related problems
  - May be solved by changing a kernel parameter in the /proc directory