CMPS 261 Server Management - Module 4: Tuning and Configuration

Mark Voortman, Ph.D.

Course Modules

- *Module 0: Course Design*
- Module 1: Introduction to Servers and Server Operating Systems
- Module 2: Getting Started with FreeBSD Server
- Module 3: Software Maintenance
- **Module 4: Tuning and Configuration**
- Module 5: Storage Management
- Module 6: Networking
- Module 7: Shell Scripting
- Module 8: Building a WordPress Server

Module 4

- **Tuning and Configuration**
- Part A
  - Perform configuration and tuning analysis and adjustments
    - Computer perspective
    - Enterprise perspective
    - Immediate problem solving
    - Long term monitoring
  - Start and stop services
  - Set up scheduled tasks
- Part B
  - Understand system logging and how to use logging for investigation
  - Understand the boot process

Objectives

- Upon successful completion of this module, you should be able to:
  - Understand the need for configuration and tuning
  - Understand what services are and how to control them
  - Schedule tasks
  - Use the system logging facilities
  - Understand the boot process

PART A

Server Performance

- Need to assure that servers are performing adequately for required tasks
  - Is a server under stress from too much work?
    - What is nature of stress? Memory? CPU? Disk I/O?
    - What is the cause of the stress?
    - What will solve the problem?
  - Long term, what is capacity of server to handle growth?

Investigation Versus Monitoring

- Investigation
  - When a problem is observed, and we must find out the cause
  - Also termed troubleshooting
- Monitoring
  - Proactive observations of key performance metrics
- Tools we use for each are different

Investigation/Troubleshooting

- No one single defined script to follow
- What you do is dictated by circumstances
- Alerted to issue in various ways
  - Internal users report speed or access problems
    - Can’t print, save files, or check email
  - Failures in applications
  - External users report access problems
    - Can’t get access to web site

Investigation/Troubleshooting

- Problem could be network
- Problem could be in storage system
- Problem could be on different server
  - Database server issue could be exposed via an application server
- Problem could be database related
  - Performance problems caused by corrupt database indexes

Investigation/Troubleshooting Commands

- Our review will be restricted to those commands available in FreeBSD to check on performance.
- We will first review the commands, then conduct a demonstration.
- We recommend that you open the manual pages web site as we discuss each command.
  - https://www.freebsd.org/cgi/man.cgi

Investigation/Troubleshooting Commands

- `vmstat`
  - Report virtual memory statistics
  - Other information also displayed
    - Processes
    - Memory
    - Page
    - Disks
    - Faults
    - CPU

Investigation/Troubleshooting Commands

- `uptime`
  - Shows how long system has been running
  - Reports number of users logged in
  - System load value at 1, 5 and 15 minute periods
  - Remember two shell commands that are useful here:
    - `who`
    - `w`

Investigation/Troubleshooting Commands

- `who -b`
  - Reports when system was last booted

Investigation/Troubleshooting Commands

- `top`
  - Display and update information on top (CPU consuming) processes

Investigation/Troubleshooting Commands

- ps
  - Display information about active processes
    - By default, for the user running command

Investigation/Troubleshooting Commands

- `swapinfo`
  - Gives condition of swap space

Investigation/Troubleshooting Commands

- `iostat`
  - Report I/O statistics

Investigation/Troubleshooting Commands

- `systat`
  - Display system statistics of various kinds

Investigating/Troubleshooting Summary

- We have seen commands that let us look at conditions on a single computer
  - Current conditions, not history
- Many more utilities and scripts can be found to provide additional information
- Now let’s turn our attention to longer term monitoring

Investigation Versus **Monitoring**

- Investigation
  - Looking at conditions on one server at a point in time
- Monitoring
  - Longer term observation, across the enterprise
  - Tools we use for each are different

Roll Your Own?

- Unix has a wealth of commands and utilities
- People can, and do, cobble together monitoring capabilities
- Example: Using the nc command

The nc Command

- nc is a Swiss Army knife TCP/UDP utility
- Could be used to send monitoring data from one server to a central repository
- Data could be processed from there
- Try it out
  - Set up the client/server model described in the man page

Reality Check

- While Unix/Linux are popular for servers, they are not exclusive
  - Most organizations have numerous servers, of many flavors
  - All should be monitored

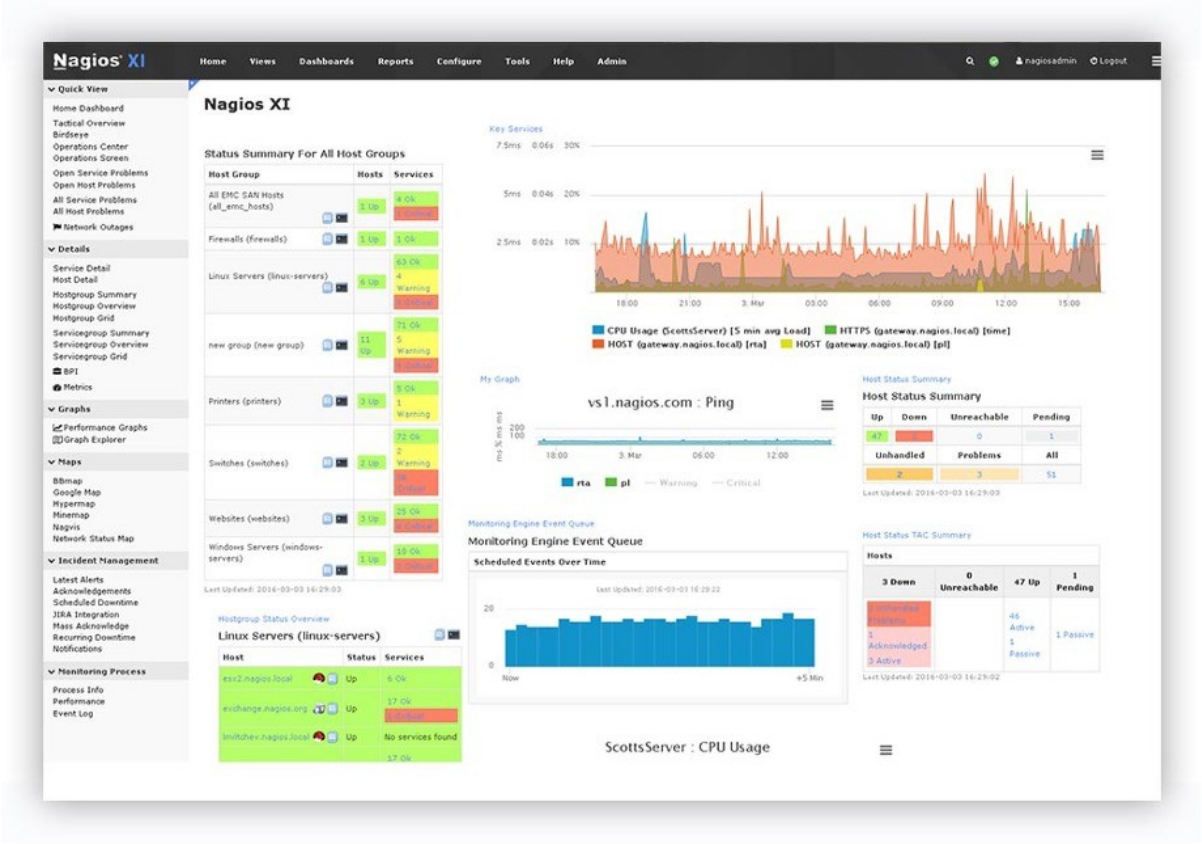
The Monitoring Tool Solution

- Server monitoring is a well-served software category
  - Commercial
  - Open source
  - Web-based
- Constant monitoring, with alerts generated if problems occur
- Rich graphical interfaces to portray information
  - “A picture is worth a thousand words”

SolarWinds (Commercial)

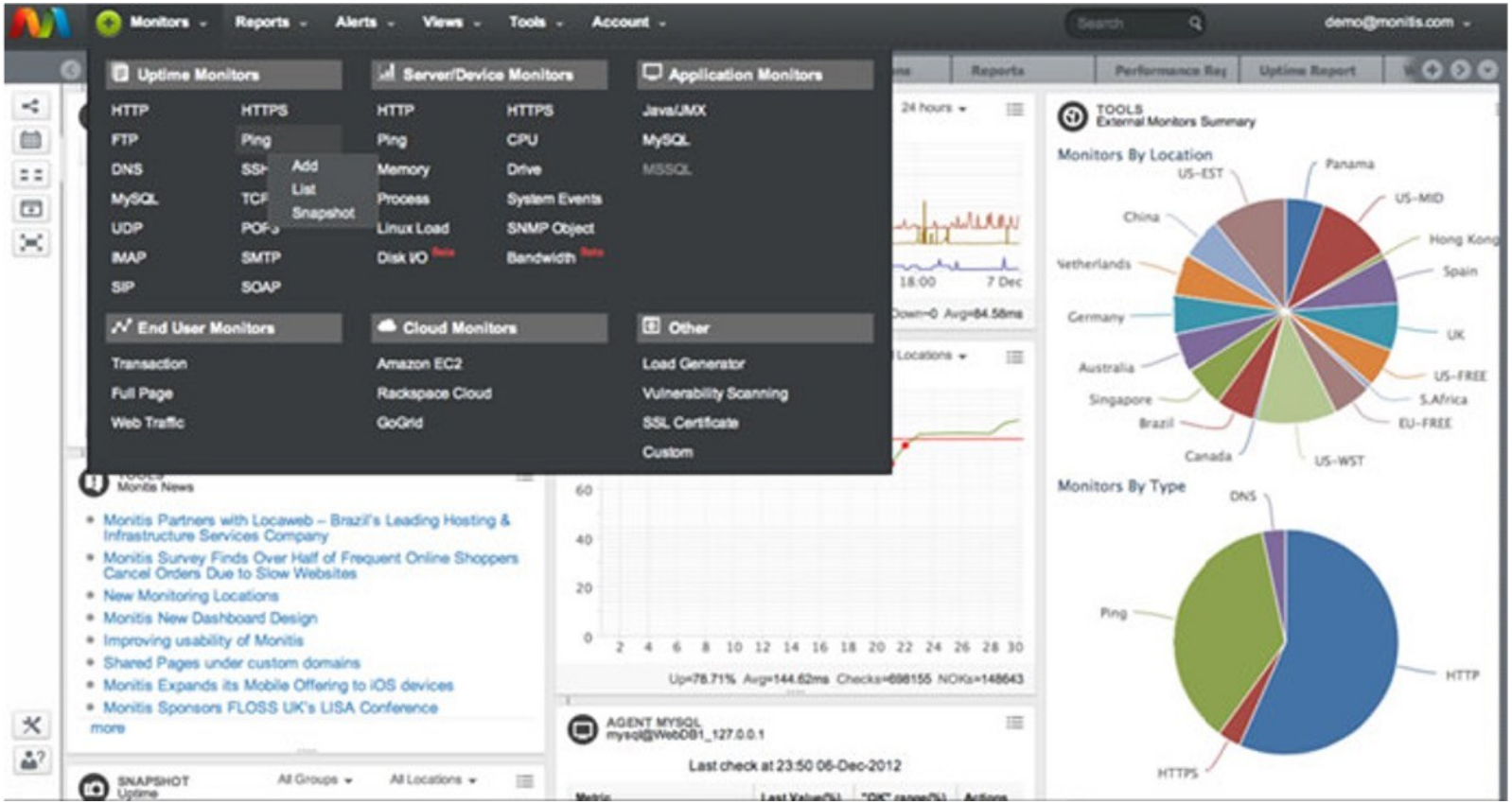
http://www.solarwinds.com/server-application-monitor?CMP=OTC-tad-dns-dns_sermon-sam-PP

Nagios (Commercial)



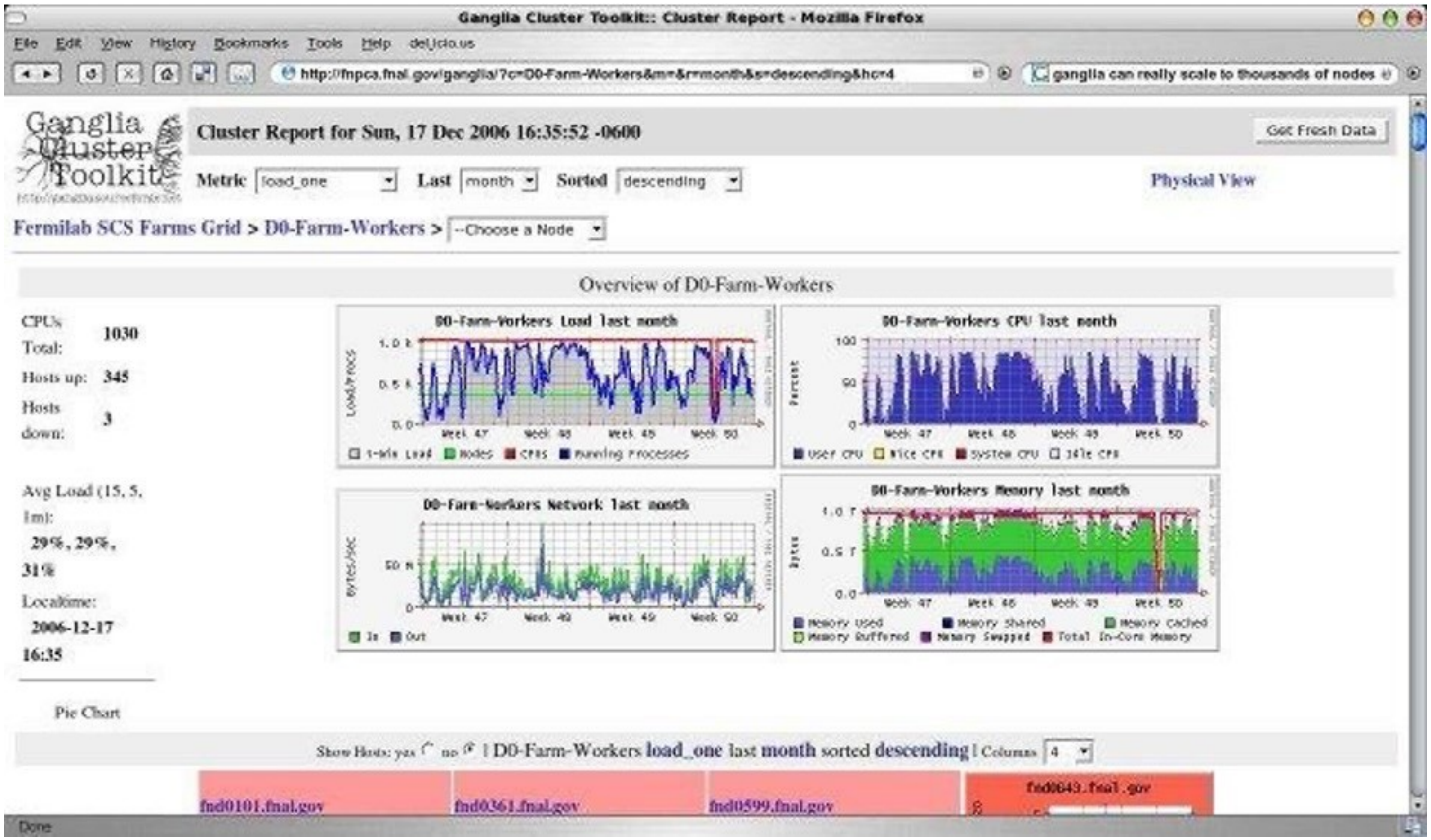
https://www.nagios.com/products/nagios-xi/

Monitor.US (Free/Commercial)



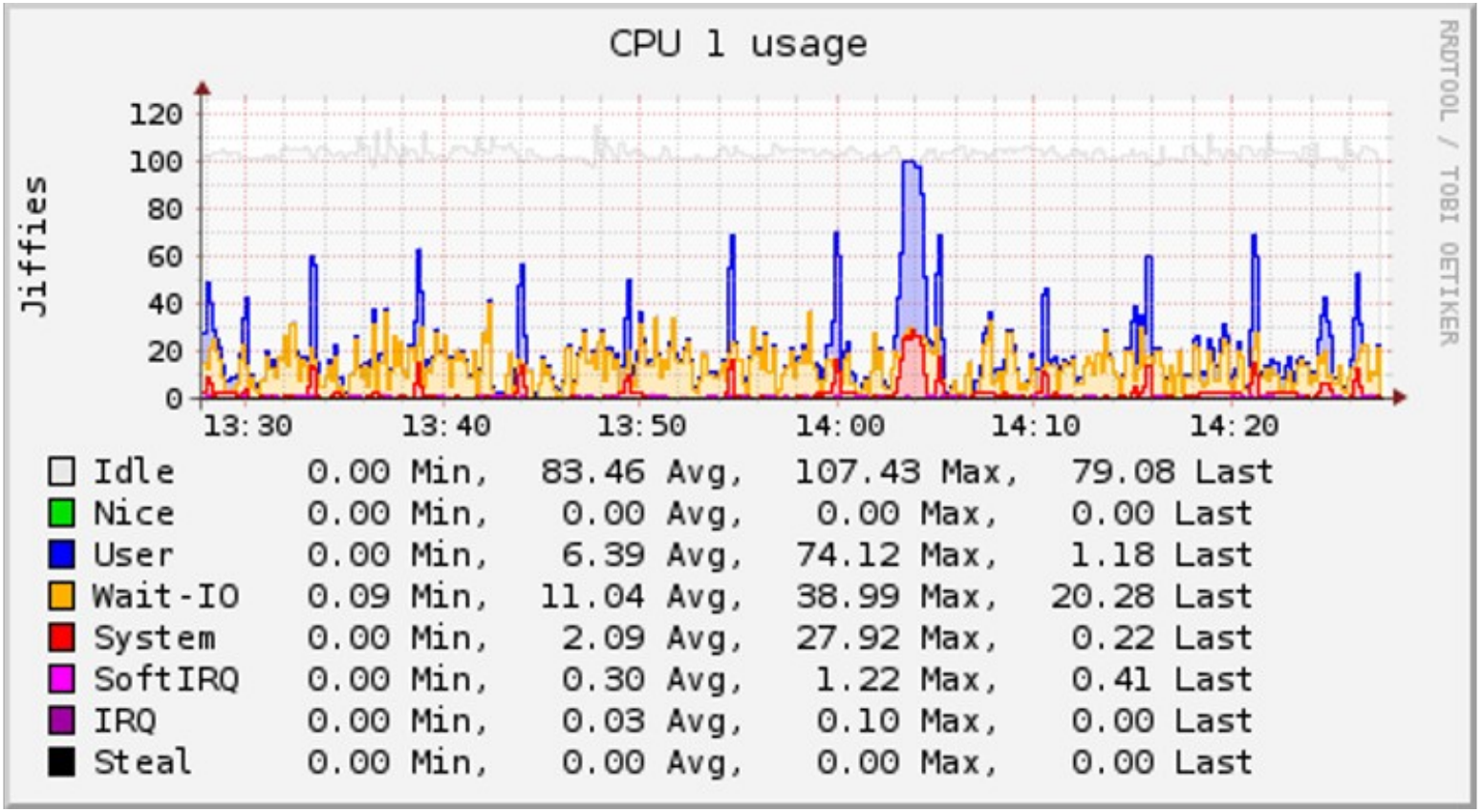
http://www.monitis.com/server-monitoring

Ganglia (Open Source)



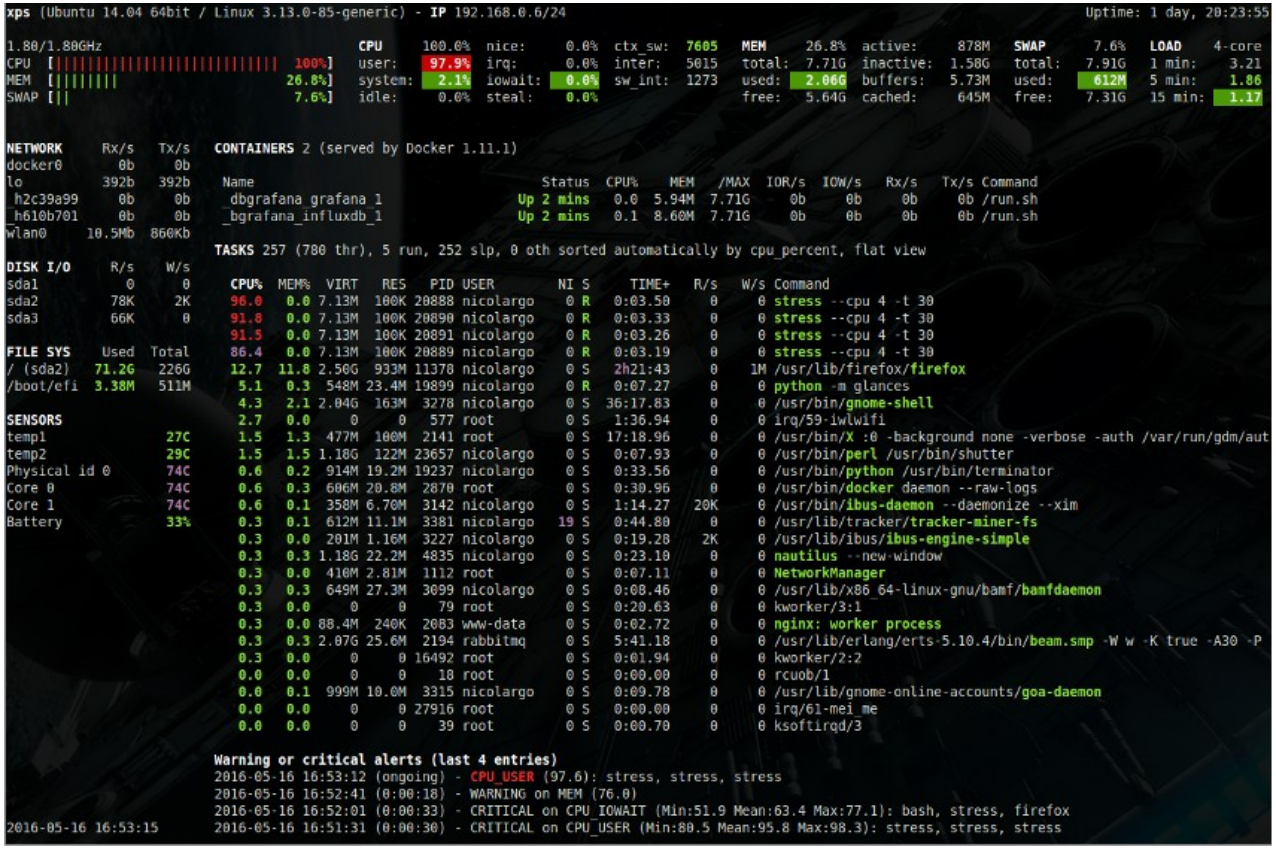
http://ganglia.info/

CollectD (Open Source)



http://collectd.org/

Glances (Open Source)



http://glances.readthedocs.io/en/latest/index.html

Glances

- Install steps:
  - `pkg install py38-glances`
  - `pkg install py38-bottle`
- To run Glances:
  - `glances`
- To run and allow remote web access:
  - `glances –w –B {IP address of server}`
  - `glances –w –B 192.168.1.74`
    - Observe assigned port number
    - Use port number when browsing to site

Summary

- A rich set of software solutions are available for enterprise server monitoring
- Value of the comprehensive information gathered makes this a compelling option

Services

- Services are background applications that run without a user interface
- Provide vital functionality
  - Internal to the server operations
  - External access to server functionality
- Examples
  - Running apache web server
  - Accessing your system remotely via secure shell

Services

- A service is a specially designed application (or script) that silently performs its job
- Example:
  - Apache monitors TCP/IP traffic coming in to ports 80 and 443 (typically)
    - Apache responds to requests by sending out web pages, graphics, etc.
  - Services tend to silently monitor, waiting for requests
  - Services may need to start at boot time

Managing Services

- System maintenance and trouble resolution may require interacting with services
- Services can crash or run amok
- `service` command available to interact with services

List Enabled Services

- `service –e` will display enabled services
- Try it out
  - `service –e`

Interacting With Services

- service command has four main options
  - start – to start the service
  - stop – to stop the service
  - restart – to stop and restart
  - status – to display status
- Try it out
  - `service devd status`

Interacting With Services

- Enabled service list does not show running services
- Try it out
  - `service sshd stop` (be careful not to lock yourself out)
  - `service sshd status`
  - `service -e`

Monitoring Services

- monit is a tool that can be used to monitor services
  - Other aspects of a system also
- `pkg install monit`
  - Will require configuration before running
  - See https://mmonit.com/monit/

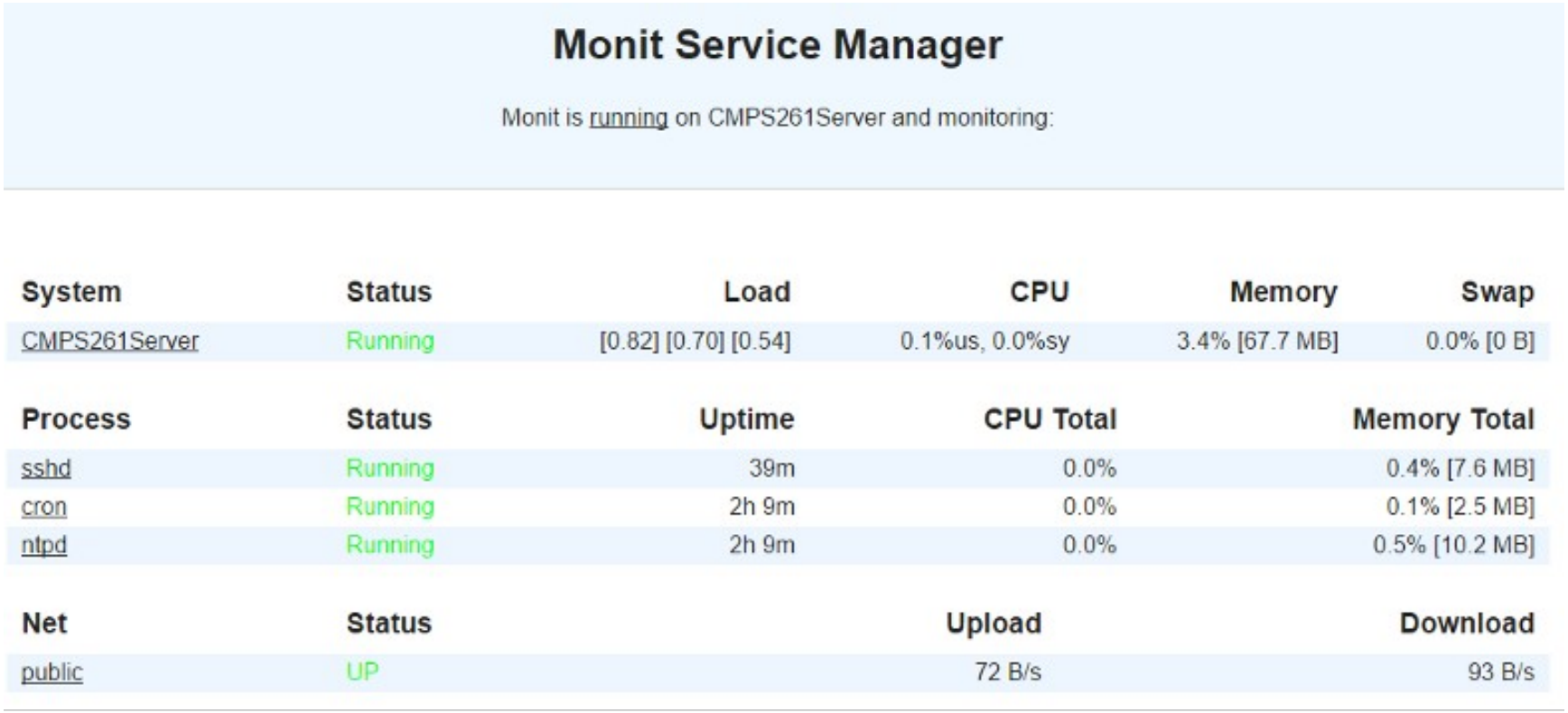
Monit

```
root@CMPS261Server:/usr/home/admin1 # monit summary
Monit 5.20.0 uptime: 1h 42m
```

Service Name	Status	Type
CMPS261Server	Running	System
sshd	Running	Process
cron	Running	Process
ntpd	Running	Process
public	UP	Network

monit can be run from command line

Monit



Monit Service Manager

Monit is running on CMPS261Server and monitoring:

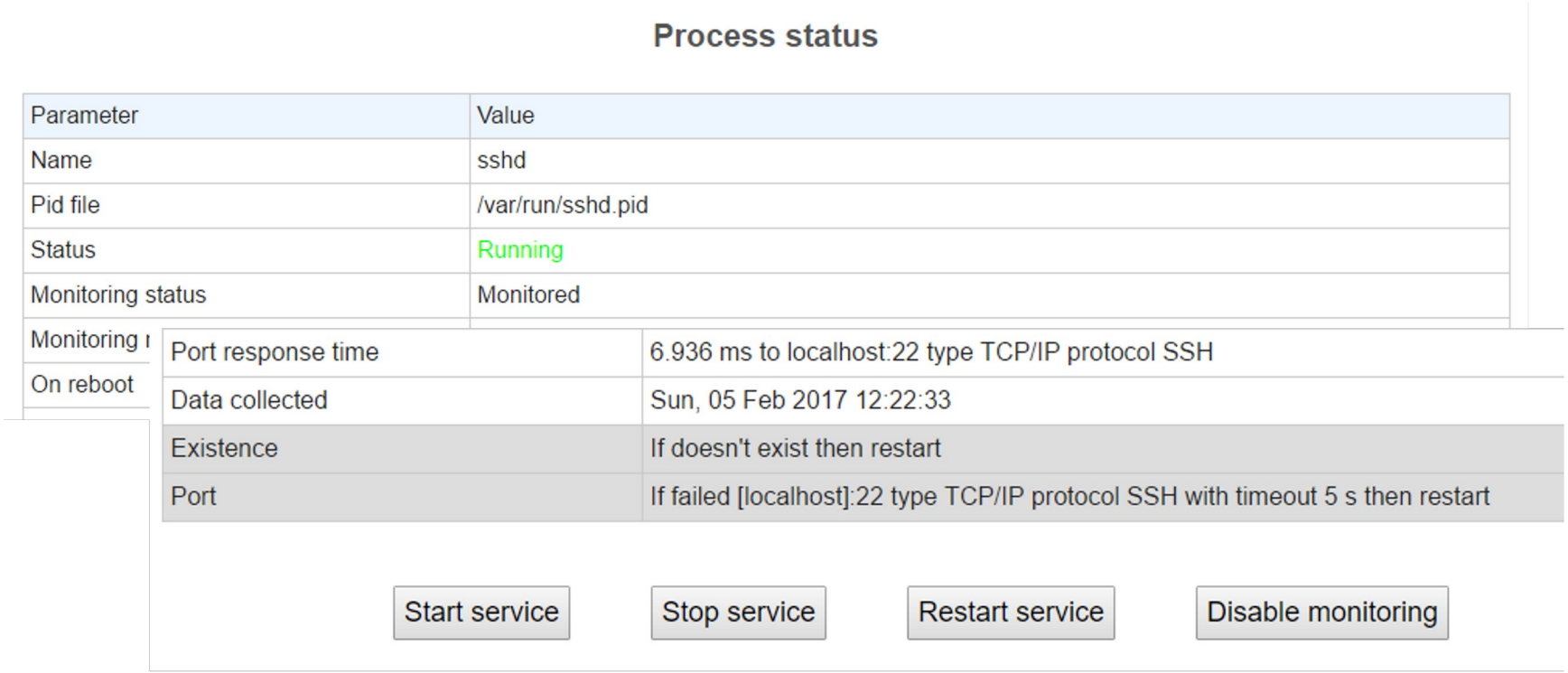
System	Status	Load	CPU	Memory	Swap
CMPS261Server	Running	[0.82] [0.70] [0.54]	0.1%us, 0.0%sy	3.4% [67.7 MB]	0.0% [0 B]

Process	Status	Uptime	CPU Total	Memory Total
sshd	Running	39m	0.0%	0.4% [7.6 MB]
cron	Running	2h 9m	0.0%	0.1% [2.5 MB]
ntpd	Running	2h 9m	0.0%	0.5% [10.2 MB]

Net	Status	Upload	Download
public	UP	72 B/s	93 B/s

monit can also be run remotely via browser

Monit



Services can be remotely managed

Summary

- Service applications provide critical functionality in the server environment
- Administrators will need to know how to interact with services
- Will also need to know how to set up new services
  - Discussed as we move forward

cron

- cron daemon is standard tool for running scheduled tasks
  - Starts when system boots
  - Runs while system is up
- cron reads configuration files
  - Contains lists of commands to run
  - When to run them
  - Commands are executed by sh
    - Virtually anything you can do by hand, can be done on schedule

cron service

- Check status of cron service:
  - service cron status
  - This service can only be checked by root
    - The `cron.pid` file is protected, which is why

cron Configuration Files

- cron controlled by one or two configuration files
  - System configuration file is `/etc/crontab`
  - User configuration file
    - Each user can have one, or no, configuration file
    - Location of user config files is `/var/cron/tabs`
    - Supplements system cron

cron Configuration File Contents

- Each line has run time parameters
  - `*` matches everything
  - Single integer matches exactly
  - Two integers separated by dash is range
  - Comma separated list of integers or ranges
  - `*/integer` is that dimension, every interval

cron Configuration File Contents

```
# /etc/crontab - root's crontab for FreeBSD
#
# $FreeBSD$
#
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
#
#minute hour    mday    month   wday    who     command
#
*/5     *       *       *       *       root    /usr/libexec/atrun
```

Cron overview

cron Configuration File Contents

```
#minute hour    mday    month   wday    who  command
20      1       *       *       *       root find /tmp -atime +3 -type f -exec rm -f {} ';'
```

Example task

User crons

- crontab command used to manage user crons
- Can be executed by root, or by a user (for their cron)
- `crontab -e`: Creates or edits a user cron
- `crontab -l`: Lists contents of a user cron
- `crontab -r`: Removes a user cron
- If running from root, always use –u to specify user

Controlling cron Access

- Two files control who can edit their crons:
  - `/var/cron/allow`: List of users allowed to use crontab
  - `/var/cron/deny`: List of users prohibited from using crontab
  - If any name appears in allow, then all allowed names must be included, even root
  - Entries are simply user names, one per line
- Where crons are stored:
  - /var/cron/tabs

cron Example

- Let’s add a cron to write the date and time to a file
- Look at contents of system cron

Summary

- Scheduling operations are a powerful way to have a system do the work for you
- Virtually any command can be executed
- Each user can have their own cron
  - Supplements, not replaces, main system cron

PART B

Syslog and Log Files

- Application and service log files are a prime source of key information about system performance
- Log files are text files
- Log management tools automatically rotate, compress, and remove log files
- Logging can be per machine, or centralized

Log File Locations

- Log files can be anywhere on the system
  - Decision made by application author
- Often, the location will be /var/log
- syslogd daemon is the standard logging facility
  - Applications can do their own logging thing
  - Look at `/etc/syslog.conf` for logging configuration

Syslog Configuration

- `/etc/syslog.conf` contains logging configuration
- Can be a guide to which log file pertains to a feature

`security.*`	`/var/log/security`
`auth.info;authpriv.info`	`/var/log/auth.log`
`mail.info`	`/var/log/maillog`
`lpr.info`	`/var/log/lpd-errs`
`ftp.info`	`/var/log/xferlog`
`cron.*`	`/var/log/cron`

Log File Location

- Finding log files made more difficult by non-standard naming
  - Example names (in /var/log):
  - auth.log
  - bsdinstall_log
  - cron
  - dmesg.today
  - lpd-errs
  - maillog
  - sendmail.st
  - xferlog

Log File Rotation

- Log files are often appended to
- This can result in log files that grow excessively large and contain obsolete information
- A cron process runs periodically that rotates log files and deletes ones no longer needed
- See `/etc/newsyslog.conf` for list
- Logs can be rotated based on size, point in time, or both

Centralized Logging

- FreeBSD allows writing logging information to a central server
- A FreeBSD server would be set up to perform this duty
- It's `syslog.conf` would contain the lists of clients sending it logging info
- Each sending server would then be configured to point to the logging server

Varying Levels of Access

- Some logs readable by all
- All logs readable by super user

Working With Log Files

- Since log files are (mostly) text files, you can handle them as such
  - `head`
  - `tail`
  - `grep`
  - `cat`
- Log files can get large
- Example:
  - `grep error /var/log/messages`

Working With Log Files

- Tool named colorize will display logs in color
  - `pkg install colorize`

Colorize Example Output

```
$ grep error /var/log/messages | colorize
Feb  5 10:20:23 CMPS261Server kernel: module_register_init: MOD_LOAD (vesa, 0xffffffff8101c970, 0) error 19
Feb  5 10:20:23 CMPS261Server sshd[609]: error: Bind to port 22 on :: failed: Address already in use.
Feb  5 10:20:23 CMPS261Server sshd[609]: error: Bind to port 22 on 0.0.0.0 failed: Address already in use.
Feb  6 18:29:51 CMPS261Server kernel: module_register_init: MOD_LOAD (vesa, 0xffffffff8101c970, 0) error 19
Feb  6 18:29:53 CMPS261Server sshd[609]: error: Bind to port 22 on :: failed: Address already in use.
Feb  6 18:29:53 CMPS261Server sshd[609]: error: Bind to port 22 on 0.0.0.0 failed: Address already in use.
```

Writing to the System Log

- logger command writes to messages log
- Useful in scripts
- `$ logger Check this out!!`
- `$ tail /var/log/messages`
- Last line should look like this:
  - `Feb  7 00:29:51 CMPS261Server admin1: Check this out!!`
- Try this yourself!

Summary

- Logs are one of the main diagnostic tools for the system administrator

System Start-Up

- Reference materials
  - FreeBSD handbook
  - init man page

System Start-Up

- The starting of a computer system is a well-orchestrated set of tasks
- Important to understand how it works
  - May be a need to intervene to fix problems
  - System might not boot up normally

The Boot Process

- The power on process executes code in the BIOS to find the startup code on the disk
- The master boot record or GUID partition table
- This executes boot0, which only knows to execute boot1
- boot1 knows only to find boot2 and execute it
- boot2 executes the loader process
  - Starts with `/boot/loader.rc`
  - Gets information from `/boot/defaults/loader.conf` and `/boot/loader.conf`
- This essentially loads the kernel and begins running it

System Initialization

- The kernel continues start-up by running the init process
- init process critical to successful operation
  - If init dies, system will reboot
- init runs `/etc/rc` which contains start-up commands
- `/etc/rc.conf` is a configuration file with some editable settings
  - DHCP versus fixed IP
  - Start up of services
  - Host name of computer

Intervening in the Start-Up

- There are numerous locations where edits can be made to control start-up
- Must tread very carefully here
- Unix is resilient, and will at least partially boot
- Can utilize recovery mode
  - Boot into single user mode
  - Stop at boot step to manually enter commands

Looking at Boot-Up Messages

- Use dmesg command to display messages
  - Message will indicate status of interactions with hardware devices

Shutting Down

- Shutdown is as ordered as start-up
- `/etc/rc.shutdown` is run
- TERM signals sent to all processes
  - If they don't shut down, then KILL signal is sent
- To immediately power off:
  - `shutdown -p now`
  - `poweroff`

Practice Project

- Install and configure Nextcloud in your jail
- Follow the Nextcloud tutorial
- Will need to apply all knowledge learned so far

THANK YOU!