

Monitoring: Paging and swapping System load

Tutor: Lưu Thanh Trà
Email: lttra@hoasen.edu.vn

Outline

- Paging and swapping
- Paging and swapping monitoring commands
- System load
- FTP automation

Paging and swapping

- Memory
 - RAM
 - Swap memory
- Swap memory
 - Volume
 - Configuration
 - Monitoring

Swapping monitoring commands

- Linux

- # free -m

	total	used	free	shared	buffers	cached
Mem:	52	51	1	0	1	20
-/+ buffers/cache:						
		30	22			
Swap:		211	9	202		

- Solaris

- # swap -s

total: 26788k bytes allocated + 7256k reserved
= 34044k used, 557044k available

■ AIX

- # lsps -a
- | Page Space | Physical Volume | Volume Group | Size | %Used | Active | Auto | Type |
|------------|-----------------|--------------|--------|-------|--------|------|------|
| paging00 | hdisk2 | rootvg | 1024MB | 11 | yes | yes | lv |
| hd6 | hdisk0 | rootvg | 1024MB | 9 | yes | yes | lv |
- # lsps -s
- Total Paging Space Percent Used
- 2048MB 10%

■ HP-UX

■ swapinfo -tm

	Mb	Mb	Mb	PCT	START/	Mb	
TYPE	AVAIL	USED	FREE	USED	LIMIT	RESERVE	PRI
dev	96	21	73	22%	928768	-	1
reserve	-	46	-46				
memory	15	5	10	33%			
total	111	72	37	65%	-	0	-

/dev/dsk/c0t6d0

Monitoring swapping - Linux

■ # ./linux_swap_mon.ksh

Swap Space Report for root

Sun Jun 9 13:01:06 EDT 2010

Total Amount of Swap Space: 211MB

Total KB of Swap Space Used: 16MB

Total KB of Swap Space Free: 195MB

Percent of Swap Space Used: 7.5800%

Percent of Swap Space Free: 92.4100%

- PERCENT_FREE=\$(bc <<EOF
- scale=4
- (\$SW_FREE / \$SW_TOTAL) * 100
- EOF
-)

-
1. ##### DEFINE VARIABLES HERE #####
 2. THISHOST=\$(hostname) # Host name of this machine
 3. **PC_LIMIT=65 # Upper limit of Swap space percentage**
 4. **# before notification**
 5. ##### INITIALIZE THE REPORT #####
 6. echo "\nSwap Space Report for \$THISHOST\n"
 7. date
 8. **free -m | grep -i swap | while read junk SW_TOTAL SW_USED SW_FREE**

```
9. do
10. # Use the bc utility in a here document to calculate the percentage
11. PERCENT_USED=$(bc <<EOF
12. scale=4
13. ($SW_USED / $SW_TOTAL) * 100
14. EOF
15. )
16. PERCENT_FREE=$(bc <<EOF
17. scale=4
18. ($SW_FREE / $SW_TOTAL) * 100
19. EOF
20. )
```

```
21. # Produce the rest of the paging space report:
22. echo "\nTotal Amount of Swap Space:\t${SW_TOTAL}MB"
23. echo "Total KB of Swap Space Used:\t${SW_USED}MB"
24. echo "Total KB of Swap Space Free:\t${SW_FREE}MB"
25. echo "\nPercent of Swap Space Used:\t${PERCENT_USED}%"
26. echo "\nPercent of Swap Space Free:\t${PERCENT_FREE}%"
27. # Grab the integer portion of the percent used to
28. # test for the over limit threshold
29. INT_PERCENT_USED=$(echo $PERCENT_USED | cut -d. -f1)
30. if (( PC_LIMIT <= INT_PERCENT_USED ))
31. then
32. # Swap space limit has exceeded the threshold, send
33. notification
34. tput smso # Turn on reverse video!
35. echo "\n\nWARNING: Paging Space has Exceeded the ${PC_LIMIT}!\n"
36. tput rmso # Turn off reverse video!
37. fi
```

System load

■ Uptime command

- measure of the average length of the run queue
1',5',15'

■ Example

- # uptime (<20 minutes)
 - 12:17pm **up 20 min, 4 users, load average: 2.29, 2.17, 1.51**
- # uptime (<1 hour)
 - 1:04pm **up 1:07, 4 users, load average: 1.74, 2.10, 2.09**
- # uptime (<14 days)
 - 4:40pm **up 12 days, 19:03, 4 users, load average: 1.52, 0.47, 0.16**

Determine the position of the uptime

- uptime
- # Find the correct field based on how long the system has been up.
- if \$(uptime | grep day | grep min >/dev/null)
- then
- **FIELD=11**
- elif \$(uptime | grep day | grep hr >/dev/null)
- then
- **FIELD=11**
- elif \$(uptime | grep day >/dev/null)
- then
- **FIELD=10**
- elif \$(uptime | grep min >/dev/null)

-
- then
 - FIELD=9
 - else # The machine has been up for 1 to 23 hours.
 - FIELD=8
 - fi
 - # Display the correct field.
 - echo "\nField is \$FIELD \n"
 - **LOAD=\$(uptime | sed s/,//g | awk '{print \$'\$FIELD'}')**

Another way

- function get_max
- {
- ((\$# == 0)) && return -1
- echo \$#
- }
- ##### MAIN #####
- MAX=\$(get_max \$(uptime)) # Get the total number of fields in uptime
- ((MAX == -1)) && echo "ERROR: Function Error...EXITING..." && exit 2
- TARGET_FIELD=\$((MAX - 2)) # Subtract 2 from the total
- **CPU_LOAD=\$(uptime | sed s/,//g | awk '{print '\$TARGET_FIELD'})**
- echo \$CPU_LOAD

System load command

- vmstat
- sar
- iostat

FTP automation


- [root:yogi]@/# cd /scripts/download
- [root:yogi]@/scripts/download# ftp wilma
- Connected to wilma.
- 220 wilma FTP server (SunOS 5.8) ready.
- Name (wilma:root): randy
- 331 Password required for randy.
- Password:
- 230 User randy logged in.
- ftp> cd /scripts
- 250 CWD command successful.
- ftp> get auto_ftp_xfer.ksh
- 200 PORT command successful.
- 150 ASCII data connection for auto_ftp_xfer.ksh (10.10.10.1,32787) (227 bytes).
- 226 ASCII Transfer complete.
- 246 bytes received in 0.0229 seconds (10.49 Kbytes/s)
- local: auto_ftp_xfer.ksh remote: auto_ftp_xfer.ksh
- ftp> bye
- 221 Goodbye.
- [root:yogi]@/scripts/download#

FTP script

- ftp -i -v -n wilma <<**END_FTP**
- user randy mypassword
- binary
- lcd /scripts/download
- cd /scripts
- get auto_ftp_xfer.ksh
- bye
- **END_FTP**

FTP commands

- get, put
- mget, mput
- lcd, cd
- nlist \$REMOTEDIR \$DIRLISTFILE

-
- RNODE="wilma"
 - USER="randy" 
 - UPASSWD="mypassword"
 - LOCALDIR="/scripts/download"
 - REMOTEDIR="/scripts"
 - DIRLISTFILE="\${LOCALDIR}/\${RNODE}.\${(basename
\${REMOVEDIR}).dirlist.out}"
 - cat /dev/null > \$DIRLISTFILE

 - ftp -i -v -n \$RNODE <<END_FTP
 - user \$USER \$UPASSWD
 - nlist \$REMOVEDIR \$DIRLISTFILE
 - bye
 - END_FTP
-