

SarCheck®

(A Solaris Example)

SarCheck(R): Automated Analysis of Solaris sar and ps data

NOTE: This software is scheduled to expire on 08/21 and has not yet been tied to your system's Host ID. To permanently activate SarCheck, please run `/opt/sarcheck/bin/analyze -o` and send the output to us so that we can generate an activation key for you.

This is an analysis of the data contained in the file `/tmp/rpt`. The data was collected on 06/17, from 08:20:01 to 15:40:00, from system 'drew'. There were 44 sar data records used to produce this analysis. Operating system is Solaris 2.7. One processor is configured. 64 megabytes of memory are present.

Data collected by the `ps -elf` command on 06/17 from 08:20:01 to 15:40:00, and stored in the file `/opt/sarcheck/ps/20040617`, will also be analyzed.

The default GRAPHDIR was changed with the `-gd` switch to `/tmp/test`.

Table of Contents

[Recommendations Section](#)
[Resource Analysis Section](#)
[Capacity Planning Section](#)
[Custom Settings Section](#)
[Summary of Statistics](#)

SUMMARY

When the data was collected, no CPU bottleneck could be detected. No significant memory bottleneck was seen. No significant I/O bottleneck was seen. A change has been recommended to at least one tunable parameter. Limits to future growth have been noted in the Capacity Planning section.

At least one possible memory leak has been detected. See the Resource Analysis Section for details.

Some of the defaults used by SarCheck's rules have been overridden using the `sarcheck_parms` file. See the Custom Settings section of the report for more information.

RECOMMENDATIONS SECTION

All recommendations contained in this report are based solely on the conditions which were present when the performance data was collected. It is possible that conditions which were not present at that time may cause some of these recommendations to result in worse performance. To minimize this risk, analyze data from several different days, implement only regularly occurring recommendations, and implement them one at a time.

Change the value of `maxpgio` from 60 to 65536. The reason for this significant change can be found in the Resource Analysis Section. This parameter can be changed by adding the following line to the `/etc/system` file: `'set maxpgio = 65536'`. NOTE: Don't forget to check `/etc/system` first to see if there's already a `set` command modifying this tunable parameter. If there is, modify that command instead of adding another one.

Change the value of `slowscan` from 100 to 500. This parameter can be changed by adding the following line to the `/etc/system` file: `'set slowscan = 500'`. An increase in the value of `slowscan` has been recommended due to the presence of significant scanning activity and recommendations made by Adrian Cockcroft on page 336 of the second edition of his Sun Performance and Tuning book. NOTE: Don't forget to check `/etc/system` first to see if there's already a `set` command modifying this tunable parameter. If there is, modify that command instead of adding another one.

More information on how to change tunable parameters is available in the System Administration Guide. We recommend making a copy of `/etc/system` before making changes, and understanding how to use `boot -a` in case your changes to `/etc/system` create an unbootable system.

RESOURCE ANALYSIS SECTION

Average CPU utilization was only 3.5 percent. This indicates that spare capacity exists within the CPU. If any performance problems were seen during the monitoring period, they were not caused by a lack of CPU power. User CPU as measured by the `%usr`

Information about the system being analyzed.

Possible memory leaks and runaway processes are identified by SarCheck, along with processes that are suspiciously large. You have full control over the thresholds used to identify these processes.

Changes to various tunable parameters are recommended here. More information about the reasons for modifying these parameters can be found in the Resource Analysis Section.

SarCheck will send you to other sources of information whenever it can. In this case, more information can be found in a popular book.

For HTML output, a table of contents will be printed enabling you to go to any section of the report.

The Summary Section can be used to quickly see what SarCheck has found.

All of SarCheck's recommendations are printed in one section, along with instructions to help you implement them.

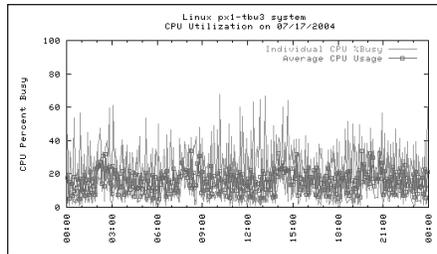
The Resource Analysis Section describes the utilization of the system's various resources, explains the reasons for various recommendations, and suggests tuning strategies where appropriate.

UNIX/Linux Performance Tuning Simplified

(A Linux Example)

Average CPU utilization was only 15.7 percent. This indicates that spare capacity exists within the CPU. If any performance problems were seen during the monitoring period, they were not caused by a lack of CPU power. CPU utilization peaked at 34.00 percent from 08:10:01 to 08:15:01. A CPU upgrade is not recommended because the current CPU had significant unused capacity.

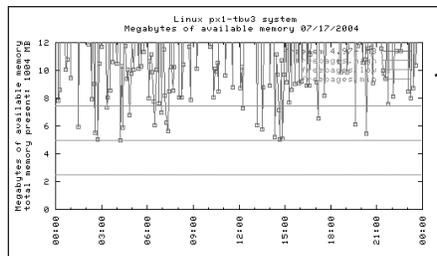
Individual CPU activity is highlighted so that you can see how well the load is balanced.



Minimum values will be reported.

The average amount of free memory was 5543.6 pages or 21.7 megabytes. The minimum amount of free memory was 1273 pages or 4.97 megabytes at 04:15:00.

Free memory can be seen in a graph so that 'bottoming out' can be easily seen.



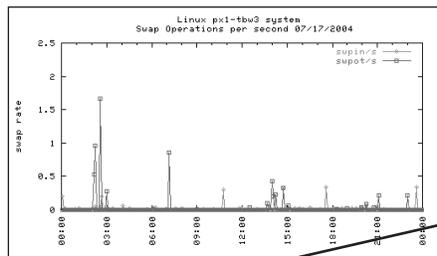
The above graph has been zoomed in to show the relationship between the size of the free list and the values of freepages parameters.

The freepages.min value was 638 pages or 2.5 megabytes. The freepages.low value was 1276 pages or 5.0 megabytes. The freepages.high value was 1914 pages or 7.5 megabytes. If the system's free list drops below freepages.high the kernel will start gently swapping. No significant memory bottleneck was seen. The number of pages of free memory occasionally dipped below the value of freepages.high but was never less than freepages.low.

For older kernels with freepages thresholds set in the /proc filesystem, these thresholds are mentioned and graphed with the free memory data.

Notable high values will be reported where appropriate.

Page ins peaked at 7138.59 per second from 10:59:59 to 11:05:00. An unusually high page in rate was detected. This may be normal for your environment, but it is still worth noting. The average page out rate was 459.715 per second. Page outs peaked at 1884.34 per second from 08:45:01 to 08:50:01. An unusually high page out rate was detected. This may be normal for your environment, but it is still worth noting.



Swap activity is part of the analysis in determining if the memory management system is operating efficiently.

Informative paragraphs on tunable kernel parameters are displayed and will vary based on the operating system.

The average swap in rate was 0.01 per second. Swap ins peaked at 0.34 per second from 23:30:03 to 23:35:01. The average swap out rate was 0.02 per second. Swap outs peaked at 1.67 per second from 02:30:00 to 02:35:01.

The kswapd parameter tries_base was set to 512. This controls the number of pages that kswapd will try to free each time it runs. The kswapd parameter tries_min was set to 32. This controls the number of times that kswapd tries to free a page of memory when it's called. The kswapd parameter swap_cluster was set to 8. This controls the number of pages that kswapd will try to write when it is called.



SarCheck®

UNIX/Linux
Performance
Tuning Simplified!

Here's what our customers are saying...

"I have found your software easy to install, and produces good information quickly from what can be a huge mass of data. My earlier analysis of an RS6000 system here (which had taken a week) was confirmed in minutes by the use of Sarcheck and gave me some additional pointers to improving the performance."

"I did find SarCheck valuable in identifying a programming bug in a third party software program. ps -elf identified a growing process that was consuming far too much memory as the day progressed. Further investigation found it to be a legitimate process that had a programmer bug and has since been resolved. I like the fact that this is a non intrusive program unlike others."

"With what I have seen so far, this is by far the best monitoring package I've seen to date."

"WOW! I can honestly say that I wish I had known about this product a long time ago. Our Accounts Dept. have been complaining for a long time that their system running a XXXXX database was grinding along and the developers were at a loss to cure the problems. On Saturday I installed SarCheck and ran off the first report based on only one days data. It immediately showed an I/O bottleneck, so on Monday I tuned the parameters as suggested and we found a distinct improvement. After running another SarCheck report later that day, we again made further changes which have resulted in reports that used to take 20mins to run, now being completed in 3mins."

P.S. Everyone in the Accounts Dept. is still smiling, even better, they now think I am 'the' genius."

Requirements

Most versions of Solaris SPARC, HP-UX, AIX and Linux x86

CPU, disk, and memory requirements are usually trivial.

"I wrote shell scripts to automate the report generation using SarCheck features like analyze. I must say that SarCheck actually helped me make a case to my organization that we need more powerful servers to achieve certain goals in systems performance. So thanks to all of you for keeping up the wonderful work on SarCheck."

"Here is a product which enables people who have very little knowledge to look sensibly at performance of their systems for a reasonable cost. If you employed me to look at the systems for you, the cost of this would far exceed the cost of this analysis tool."

Feel free to contact us for a list of customer references!

**APTITUNE
CORPORATION**

<http://www.sarcheck.com>

PO Box 1033, Plaistow, NH 03865

Voice +1-603-382-4200

Fax +1-603-382-4247

Visit our website for a
FREE Evaluation Copy at
www.sarcheck.com