[](https://rimuhosting.com/index.jsp)

[](https://rimuhosting.com/index.jsp)

[https://rimuhosting.com/images/icons/faenza/16/im-message-new.png](https://rimuhosting.com/chat.jsp) [Live chat](https://rimuhosting.com/chat.jsp)

* [Home](https://rimuhosting.com/index.jsp)
* [Order](https://rimuhosting.com/order/startorder.jsp)
* [Hosting](https://rimuhosting.com/vps/vpsindex.jsp)
* [Contact Us](https://rimuhosting.com/support/sindex.jsp)
* [Your account](https://rimuhosting.com/cp)
* [RimuHosting](https://rimuhosting.com/support/aboutrh.jsp)

[Home](https://rimuhosting.com/index.jsp) > [Support](https://rimuhosting.com/support/sindex.jsp) > [HOWTO List](https://rimuhosting.com/support/howtolist.jsp) > [Linux](https://rimuhosting.com/howto/linux.jsp) > Memory

**Working with Linux howtos**

* [Working with RPM](https://rimuhosting.com/howto/rpm.jsp)
* [Package Management Made Easy With Apt](https://rimuhosting.com/howto/apt.jsp)
* [Moving files around: Setting up FTP](https://rimuhosting.com/howto/ftp.jsp)
* [SCP: an FTP alternative](https://rimuhosting.com/howto/scp.jsp)
* [SFTP: Easy jailed accounts](https://rimuhosting.com/howto/sftp)
* [Backup FTP server: offsite backups that you control](https://rimuhosting.com/howto/backupspace.jsp)
* [Automated incremental backups: Script to automate backups using SSHFS and rdiff-backup](https://rimuhosting.com/howto/sshfs-rdiff-backup.jsp)
* [Backups with rSync](https://rimuhosting.com/howto/rsync.jsp)
* [Automated backups with RSync](https://rimuhosting.com/howto/rsyncbackup.jsp)
* [Load balancing and failover](https://rimuhosting.com/knowledgebase/rimuhosting/load-balancing-and-failover)
* [Accurate time with NTP](https://rimuhosting.com/howto/ntp.jsp)
* [Setting a UTC timezone](https://rimuhosting.com/howto/timezone.jsp)
* [Changing a hostname](https://rimuhosting.com/howto/hostname.jsp)
* [Application environment setup (using bash profiles)](https://rimuhosting.com/howto/profile.jsp)
* [Using find and xargs](https://rimuhosting.com/knowledgebase/linux/misc/using-find-and-xargs)
* [Setting up disk quotas](https://rimuhosting.com/knowledgebase/linux/misc/disk-quotas)
* [Working with Bash aliases](https://rimuhosting.com/knowledgebase/linux/misc/working-with-aliases)

**Troubleshooting Memory Usage**

Processing dying unexpectedly?  Want to know if you need more memory?

Check your /var/log/messages or /var/log/syslog log file.  If you see (on a 2.4.23 kernel):

Dec 11 10:21:43 www kernel: \_\_alloc\_pages: 0-order allocation failed (gfp=0x1d2/0)

Dec 11 10:21:44 www kernel: \_\_alloc\_pages: 0-order allocation failed (gfp=0x1f0/0)

Or (on a pre-2.4.23 kernel):

Dec 7 23:49:03 www kernel: Out of Memory: Killed process 31088 (java).

Dec 7 23:49:03 www kernel: Out of Memory: Killed process 31103 (java).

Or on a Xen-based VPS console:

swapper: page allocation failure. order:0, mode:0x20

[<c01303a4>] \_\_alloc\_pages+0x327/0x3e3

Then your programs need more memory than they can get.

Some good background on how linux uses RAM is available on [http://www.linuxatemyram.com](http://www.linuxatemyram.com/)

**Interpreting Free**

To see how much memory you are currently using, run free -m.  It will provide output like:

:~$ free -m

total used free shared buffers cached

Mem: 2008 1951 57 0 142 575

-/+ buffers/cache: 1234 774

Swap: 3812 35 3777

The top row 'used' (1951) value will almost always nearly match the top row total value (2008). Since Linux likes to use any spare memory to cache disk blocks.

The most important 'used' figure to look at is the buffers/cache row used value (1234). This is how much space your applications are currently using. For best performance, this number should be less than your total (2008) memory. To prevent out of memory errors, it needs to be less than the total memory (2008) and swap space (3812).

If you wish to quickly see how much memory is free look at the buffers/cache row free value (774). This is the total memory (2008) - the actual used (1234). (2008 - 1234 = 774)

Note that the kernel does need some RAM for caching, to help improve system performance of some subsystems, for example disk writes. If you are seeing some slowness, and RAM looks ok but tight, adding more RAM (or reducing the amount used by applications) is a good idea.

**free does not show all my memory?**

Due to some peculiarities of our older VPS environments, free may not accurately reflect the total allocated VPS memory, although it should be pretty close. To be absolutely sure sum the below two numbers, which should match the resource allocation from our control panel exactly.

$ grep DirectMap /proc/meminfo

DirectMap4k: 1024396 kB

DirectMap2M: 0 kB

**Interpreting ps**

If you want to see where all your memory is going, run ps aux.  That will show the percentage of memory each process is using.  You can use it to identify the top memory users (usually Apache, MySQL and Java processes).

For example in this output snippet:

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 854 0.5 39.2 239372 36208 pts/0 S 22:50 0:05

/usr/local/jdk/bin/java -Xms16m -Xmx64m -Djava.awt.headless=true -Djetty.home=/opt/jetty -cp /opt/jetty/ext/ant.jar:/opt/jetty/ext/jasper-compiler.jar:/opt/jetty/ext/jasper-runtime.jar:/opt/jetty/ext/jcert.jar:/opt/jetty/ext/jmxri.jar:/opt/jetty/ext/jmxtool

We can see that java is using up 39.2% of the available memory.

**Interpreting vmstat**

vmstat helps you to see, among other things, if your server is swapping.  Take a look at the following run of vmstat doing a one second refresh for two iterations.

# vmstat 1 2

procs memory swap io system cpu

r b w swpd free buff cache si so bi bo in cs us sy id

0 0 0 39132 2416 804 15668 4 3 9 6 104 13 0 0 100

0 0 0 39132 2416 804 15668 0 0 0 0 53 8 0 0 100

0 0 0 39132 2416 804 15668 0 0 0 0 54 6 0 0 100

The first row shows your server averages.  The si (swap in) and so (swap out) columns show if you have been swapping (i.e. needing to dip into 'virtual' memory) in order to run your server's applications.  The si/so numbers should be 0 (or close to it).  Numbers in the hundreds or thousands indicate your server is swapping heavily.  This consumes a lot of CPU and other server resources and you would get a very (!) significant benefit from adding more memory to your server.

Some other columns of interest: The r (runnable) b (blocked) and w (waiting) columns help see your server load.  Waiting processes are swapped out.  Blocked processes are typically waiting on I/O.  The runnable column is the number of processes trying to something.  These numbers combine to form the 'load' value on your server.  Typically you want the load value to be one or less per CPU in your server.

The bi (bytes in) and bo (bytes out) column show disk I/O (including swapping memory to/from disk) on your server.

The us (user), sy (system) and id (idle) show the amount of CPU your server is using.  The higher the idle value, the better.

**Resolving: High Java Memory Usage**

Java processes can often consume more memory than any other application running on a server.

Java processes can be passed a -Xmx option.  This controls the maximum Java memory heap size.  It is important to set a limit on the heap size, otherwise the heap will keep increasing until you get out of memory errors on your VPS (resulting in the Java process - or even some other, random, process - dying.

Usually the setting can be found in your /usr/local/jboss/bin/run.conf or /usr/local/tomcat/bin/setenv.sh config files.  And your RimuHosting default install should have a reasonable value in there already.

If you are running a custom Java application, check there is a -XmxNNm (where NN is a number of megabytes) option on the Java command line.

The optimal -Xmx setting value will depend on what you are running.  And how much memory is available on your server.

From experience we have found that Tomcat often runs well with an -Xmx between 48m and 64m.  JBoss will need a -Xmx of at least 96m to 128m.  You can set the value higher.  However, you should ensure that there is memory available on your server.

To determine how much memory you can spare for Java, try this: stop your Java process; run free -m; subtract the 'used' value from the "-/+ cache" row from the total memory allocated to your server and then subtract another 'just in case' margin of about 10% of your total server memory.  The number you come up with is a rough indicator of the largest -Xmx setting you can use on your server.

**Resolving: High Spam Assassin Memory Usage**

Are you running a Spam Assassin 'daemon'?  It can create multiple (typically 5) threads/processes and each of those threads can use a very large amount of memory.

SpamAssassin works very well with just one thread.  So you can reduce the 'children' setting and reclaim some memory on your server for other apps to run with.

for location in /etc/default/spamassassin /etc/sysconfig/spamassassin; do

if [ ! -e $location ]; then continue; fi

replace "SPAMDOPTIONS=\"-d -c -m5 -H" "SPAMDOPTIONS=\"-d -c -m1 -H" -- /etc/init.d/spamassassin

replace "\-m 10 " "-m 1 " -- $location

replace "\-m 5 " "-m 1 " -- $location

replace "\-m5 " "-m1 " -- $location

replace "max-children 5 " "max-children 1 " -- $location

done

Another thing to check with spamassassin is that any /etc/procmailrc entry only does one spamassassin check at a time.  Otherwise if you receive a batch of incoming email they will all be processed in parallel.  This could cause your server CPU usage to spike, slowing down your other apps, and it may cause your server to run out of memory.

To make procmailrc run only one email at a time through Spamassassin use a lockfile on your recipe line.  e.g. change the top line of:

:0fw:

# The following line tells Procmail to send messages to Spamassassin only if they are less thatn 256000 bytes. Most spam falls well below this size and a larger size could seriously affect performance.)

\* < 256000

| /usr/bin/spamc

To:

:0fw:/etc/mail/spamc.lock

# The following line tells Procmail to send messages to Spamassassin only if they are less thatn 256000 bytes. Most spam falls well below this size and a larger size could seriously affect performance.)

\* < 256000

| /usr/bin/spamc

**Resolving: High Apache Memory Usage**

Apache can be a big memory user.  Apache runs a number of 'servers' and shares incoming requests among them.  The memory used by each server grows, especially when the web page being returned by that server includes PHP or Perl that needs to load in new libraries.  It is common for each server process to use as much as 10% of a server's memory.

To reduce the number of servers, you can edit your httpd.conf file.  There are three settings to tweak: StartServers, MinSpareServers, and MaxSpareServers.  Each can be reduced to a value of 1 or 2 and your server will still respond promptly, even on quite busy sites.  Some distros have multiple versions of these settings depending on which process model Apache is using.  In this case, the 'prefork' values are the ones that would need to change.

To get a rough idea of how to set the MaxClients directive, it is best to find out how much memory the largest apache thread is using. Then stop apache, check the free memory and divide that amount by the size of the apache thread found earlier. The result will be a rough guideline that can be used to further tune (up/down) the MaxClients directive. The following script can be used to get a general idea of how to set MaxClients for a particular server:

#!/bin/bash

echo "This is intended as a guideline only!"

if [ -e /etc/debian\_version ]; then

APACHE="apache2"

elif [ -e /etc/redhat-release ]; then

APACHE="httpd"

fi

RSS=$(ps -aylC $APACHE |grep "$APACHE" |awk '{print $8'} |sort -n |tail -n 1)

RSS=$(expr $RSS / 1024)

echo "Stopping $APACHE to calculate free memory"

/etc/init.d/$APACHE stop &> /dev/null

MEM=$(free -m |head -n 2 |tail -n 1 |awk '{free=($4); print free}')

echo "Starting $APACHE again"

/etc/init.d/$APACHE start &> /dev/null

echo "MaxClients should be around" $(expr $MEM / $RSS)

From http://modperlbook.org/html/11-2-Setting-the-MaxRequestsPerChild-Directive.html:

*"Setting MaxRequestsPerChild to a non-zero limit solves some memory-leakage problems caused by sloppy programming practices and bugs, whereby a child process consumes a little more memory after each request. In such cases, and where the directive is left unbounded, after a certain number of requests the children will use up all the available memory and the server will die from memory starvation."*

**Resolving: High MySQL Memory Usage**

Our distros typically have MySQL preinstalled but not running.  Our pre-install uses a memory efficient /etc/my.cnf file.  If you install MySQL on a Debian server, edit the key\_buffer\_size setting in /etc/mysql/my.cnf.  A small value like 2M often works well. For an ultra-tiny setup add or change the follow entries to the mysqld section:

# if your are not using the innodb table manager, then just skip it to save some memory

#skip-innodb

innodb\_buffer\_pool\_size = 16k

key\_buffer\_size = 16k

myisam\_sort\_buffer\_size = 16k

query\_cache\_size = 1M

**My server seems slow!**

On servers that have been running for a while you may see a small amount of swap space usaage, even though there is plently of memory, And applications that are not used often may seem slow to respond initially, which can be frustrating. This is espcially common with larger applications, for example Liferay or similar java based apps.

To resolve that you need to tweak how likely it is for kernel to use swap space. This is controlled by the vm.swappiness sysctl parameter. Try the following and if that helps add the change to /etc/sysctl.conf to have the change stick across reboots The default value is 60.

echo 20 > /proc/sys/vm/swappiness

More information about the swappinesss value can bee seen at [kernel.org](https://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html)

**Troubleshooting Irregular Out Of Memory Errors**

Sometimes a server's regular memory usage is fine.  But it will intermittently run out of memory.  And when that happens you may lose trace of what caused the server to run out of memory.

In this case you can setup a script (see below) that will regularly log your server's memory usage.  And if there is a problem you can check the logs to see what was running.

wget http://proj.ri.mu/installmemmon.sh -q -O - | bash

We install this script by default on new servers.  And it is often very useful for diagnosing problems 'after the fact'.  e.g. after a system crash, or when someone asks why was my server slow at 10:23 on Tuesday.

The memmon script is invoked by cron every 10 minutes.  It logs the running processes; new messages from dmesg; changes to iptables rules; vmstat output (to see disk IO and cpu usage); and date/uptime to determine if/when a server is restarted and the current load (so you can review which periods had higher load).

**Just Add Memory**

A simple solution to resolving most out of memory problems is to add more memory.  If you'd like to increase the memory on your VPS, take a look at the [resource change tool](https://rimuhosting.com/cp/vps/resourcechange.jsp) or [send us a support ticket](https://rimuhosting.com/ticket/startticket.jsp) and let us know how much memory you need.

RimuHosting is a [RimuHosting Ltd](http://ri.mu/) business. [RimuHosting](http://ri.mu/) services include:

[https://rimuhosting.com/images/icons/redrata/white70/rh2-logo-white-70.png**Launchtime™**](http://launchtimevps.com/)VPS servers.

[https://rimuhosting.com/images/icons/redrata/white70/rimu-logo-white-70.png**RimuHosting™**](http://rimuhosting.com/)Dedicated servers.

[**Pingability™**](http://pingability.com/)Website monitoring service.

[**Bakop™**](http://bakop.com/)Remote FTP backup service.

[**Zonomi™**](http://zonomi.com/)DNS hosting service.

[**25 Mail St.™**](http://25mail.st/)Managed email hosting.

[Email the RimuHosting team](mailto:questions@rimuhosting.com?subject=RimuHosting%20enquiry)[About RimuHosting](http://rimuhosting.com/support/aboutrh.jsp)[Policies (SLA, terms and conditions, privacy)](https://rimuhosting.com/terms.jsp)[RimuHosting blog](http://blog.rimuhosting.com/)[RimuHosting news](http://ri.mu/news.jsp)

© RimuHosting Ltd 2002-2019. Rimuhosting, Launchtime and the Rimuhosting logo are RedRata™ trademarks.