Technical FAQs & FIXES

From Matchi Wiki

Contents

- 1 Database FAQs & FIXES
  - 1.1 Overrun of group_concat string length
  - 1.2 Extended Latin Characters are not correctly saved in the database
- 2 Server FAQ & FIXES
  - 2.1 BASH: Cannot allocate memory
    - 2.1.1 Retry running the process
  - 2.2 Reboot a Server

Database FAQs & FIXES

Overrun of group_concat string length

- **Symptom:** Broken JSON expressions that the website can't process, or unclosed HTML tags on pages that contain dynamic SQL-generated reports
- **Explanation:** This may happen in views that do a `group_concat` to create comma-delimited strings or a JSON string of row-data. By default, group_concat results are truncated to 1024 characters.
- **Affected areas:** Database views called `vwString_...`
- **Remedy**:
  - Edit the file `/etc/my.cnf` (as a sudo-er)
  - Add `group_concat_max_len = 10K`
  - Restart the mysql daemon: `sudo /etc/init.d/mysqld restart`

Extended Latin Characters are not correctly saved in the database

- **Symptom:** Interesting artefacts are stored and displayed in string-type columns instead of the intended characters
- **Explanation:**
  - This may happen because the string-type columns is not set to a character set that can display such characters. By default, we the use UTF-8 character set for all string data.
  - If the database is not set to UTF-8, then any new columns created in tables will assume the default character set of the database.
- **Affected areas:** String-type columns
- **Remedy**:
  - Change the database to UTF-8 encoding:

```
ALTER DATABASE [database] CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

This operation can be performed on an active database without affecting its performance.

-
  - Edit the file `/etc/my.cnf` (as a sudo-er) to ensure that any future databases created are of the correct character set and collation:

```
[client]
default-character-set=utf8

[mysql]
default-character-set=utf8

[mysqld]
init_connect='SET collation_connection = utf8_unicode_ci'
character-set-server = utf8
collation-server = utf8_unicode_ci
```

- - Restart the mysql daemon: `sudo /etc/init.d/mysqld restart`
  - Change the offending string-type fields in each table to UTF-8:

```
alter table [tablename] modify column [columnname] [stringtype] character set utf8 collate utf8_unicode_ci;
```

Server FAQ & FIXES

BASH: Cannot allocate memory

- **Symptoms:** You may encounter this error either on the console, a CRON report that has been emailed to you, or in the system log files:

When running a command:

```
-bash: fork: Cannot allocate memory
```

or in a CRON notification email:

```
/etc/cron.hourly/01restart_fsm_daemon_PROD.sh: fork: Cannot allocate memory
[2015/02/05 16:01:27][FATAL][01restart_fsm_daemon_PROD.sh][] [134] Environment specified in 01restart_fsm_daemon_PROD.sh i
```

- **Explanation:** The Linux Kernel was not able to allocate memory to run a process. This is often the case when a process repeatedly spawns itself and consumes a chunk of memory.
- **Affected Areas:** Only the immediate server is affected. The impact of failed processes should be well understood in order to recover any lost data.
- **Remedy:** Check the memory consumption. Use the `free -m` command. There appears to be sufficient memory:

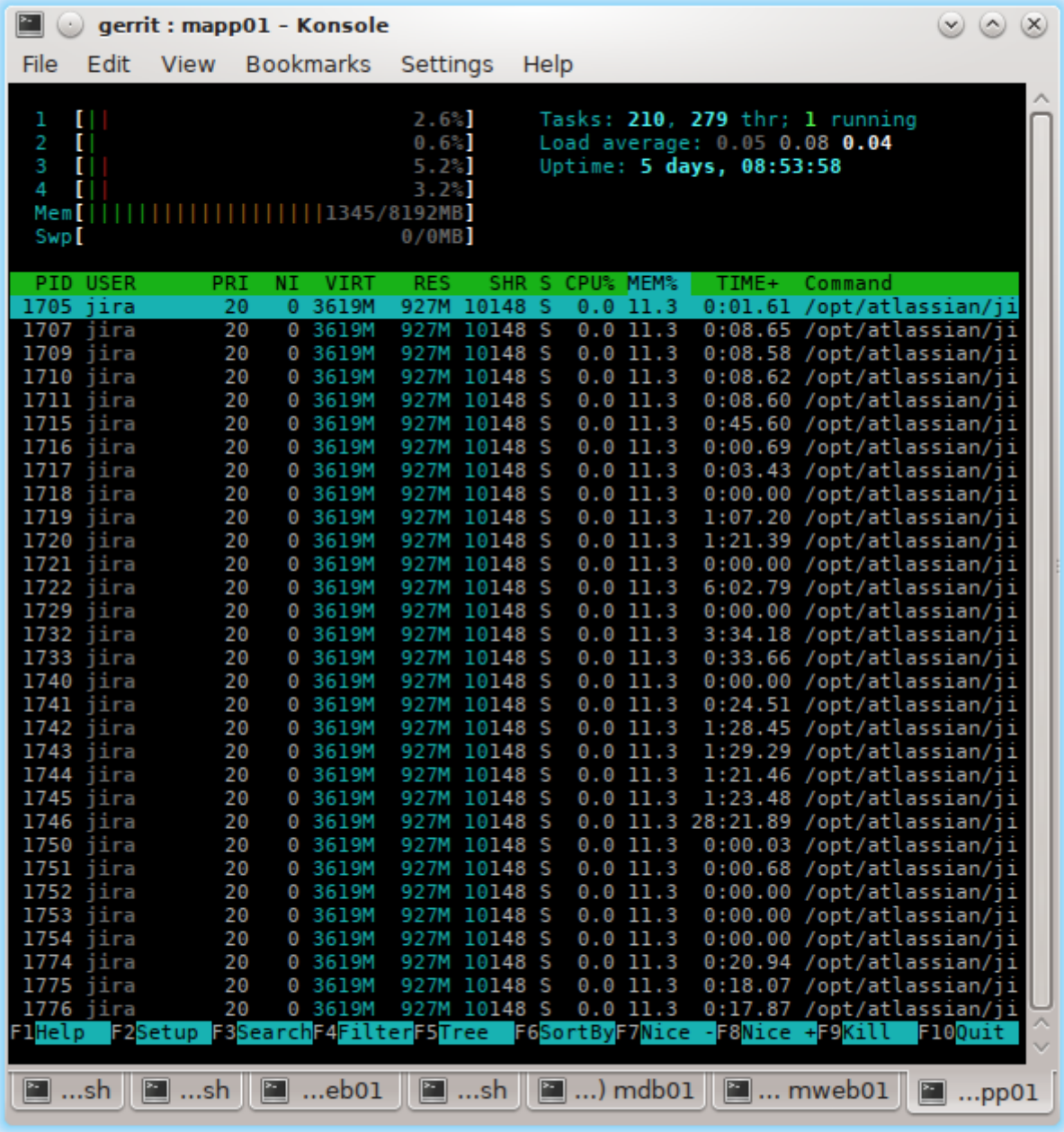
```
$ free -m
             total       used       free     shared    buffers     cached
Mem:          8192       5001       3190          4          0       3656
-/+ buffers/cache:       1344       6847
Swap:            0          0          0
```

Also look at the actual memory usage. This shows that there is still some memory left to go around:

```
$ sudo cat /proc/meminfo
MemTotal:        8388608 kB
MemFree:         3264852 kB
Cached:          3744196 kB
Buffers:               0 kB
Active:          2724464 kB
Inactive:        2351060 kB
```

```
Active(anon):     641136 kB
Inactive(anon):   690192 kB
Active(file):    2083328 kB
Inactive(file):  1660868 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:             0 kB
SwapFree:              0 kB
Dirty:               288 kB
Writeback:             0 kB
AnonPages:       1331328 kB
Shmem:              4868 kB
Slab:              48200 kB
SReclaimable:       2588 kB
SUnreclaim:        45612 kB
```

To see which process are consuming memory, run `htop` and order on the "MEM%" column.



In this case, it shows that the Java Virtual Machine may be to blame. The process that we know of that uses the JVM on this particular server is the JIRA applicaiton, so restart the process:

```
$ sudo /etc/init.d/jira stop
executing using dedicated user
                .....
          .... .NMMMD.  ...
        .8MMM.  $MMN,..~MMMO.
        .?MMM.          .MMM?.

     OMMMMZ.            .,NMMMN~
     .IMMMMMM. .NMMMN. .MMMMMN,
       ,MMMMMM$..3MD..ZMMMMMM.
        =NMMMMMM,. .,MMMMMMD.
         .MMMMMMMM8MMMMMMM,
           .ONMMMMMMMMMMZ.
             ,NMMMMMMM8.
            .:,.$MMMMMMM
          .IMMMM..NMMMMMD.
         .8MMMMM:  :NMMMMN.
         .MMMMMM.   .MMMMM~.
         .MMMMMN    .MMMMM?.

      Atlassian JIRA
      Version : 6.2.2

Detecting JVM PermGen support...
PermGen switch is supported. Setting to 384m

If you encounter issues starting or stopping JIRA, please see the Troubleshooting guide at http://confluence.atlassian.com


Server startup logs are located in /opt/atlassian/jira/logs/catalina.out
Using CATALINA_BASE:   /opt/atlassian/jira
Using CATALINA_HOME:   /opt/atlassian/jira
Using CATALINA_TMPDIR: /opt/atlassian/jira/temp
Using JRE_HOME:        /opt/atlassian/jira/jre/
Using CLASSPATH:       /opt/atlassian/jira/bin/bootstrap.jar:/opt/atlassian/jira/bin/tomcat-juli.jar
Using CATALINA_PID:    /opt/atlassian/jira/work/catalina.pid
```

Start the process again:

```
$ sudo /etc/init.d/jira start
...
```

The process should now be running OK again without unduly consuming memory. Until it looses the plot again.

- **Consider a further scenario:**

Running `htop` again shows that another process may also be causing memory allocation failures. In this case it is the Dropbox daemon:

Kill the offending process off using the `kill -9 [process Id]`-command. You can get the process Id by listing all the processes with the `ps -ef` and piping it through a filter of the name of the process that you are looking for, `| grep dropbox`. In this case, the process Id is `19031`. When the process is killed, it is a good idea to confirm that the process is actually terminated with a final `ps -ef | grep dropbox`, which should only return the process of the `grep`-command itself:

```
$ ps -ef | grep dropbox
madman   19031     1  0 Dec11 ?        00:11:35 /home/madman/.dropbox-dist/dropbox-lnx.x86_64-3.0.3/dropbox
madman   32569 15248  0 14:31 pts/1    00:00:00 grep dropbox
$ kill -9 19031
$ ps -ef | grep dropbox
madman     317 15248  0 14:31 pts/1    00:00:00 grep dropbox
```

Finally checking with `htop` again, there are not processes left that consume memory.

Note
You can also kill processes off on HTOP itself by selecting the process from the list and hitting the `F9` -key. This works for when there is only one process to kill, but not when there is a run-away spawning of multiple processes.

Retry running the process

First record the current number of memory allocation failures / page faults:

```
madman@s16972617 mapp01 ~ $ sudo cat /proc/user_beancounters | head -3
Version: 2.5
       uid  resource                     held              maxheld              barrier                limit
 73354475:  kmemsize                 42066418             43372544             50331648             58720256
```

In this example, it is 242.

Restart the offending process and see if the number of page faults grows. If it does, another process may be at fault.

Reboot a Server

You can reboot the server from the command line like this:

```
$ sudo shutdown -r now

Broadcast message from madman@s16972617.onlinehome-server.info
        (/dev/pts/1) at 14:41 ...

The system is going down for reboot NOW!
```

Since it is a virtual server, it should come up in about 30 seconds. Real ('tin') servers may take a little longer to come backup.

Note
Remember to use the `-r` option, which will cause the server to come up again. If you inadvertently stopped the server completely by using the `-h` option, then you can restart the server from the data centre's PLESK website.