Need to “calculate” optimum ulimit and fs.file-max values according to my own server needs

Need to "calculate" optimum ulimit and fs.file-max values according to my own server needs. Please do not conflict with "how to set those limits in various Linux distros" questions.

I am asking:

1. Is there any good guide to explain in detail, parameters used for ulimit? (> 2.6 series kernels)
2. Is there any good guide to show fs.file-max usage metrics?

Actually there are some old reference i could find on the net: http://www.faqs.org/docs/securing/chap6sec72.html "something reasonable like 256 for every 4M of RAM we have: i.e. for a machine with 128 MB of RAM, set it to 8192 - 128/4=32 32*256=8192"

Any up to date reference is appreciated.

linux-kernel ulimit sysctl

Forgot the mention that i need technical article. – Kunthar May 30 '11 at 20:43

2 Answers

For `fs.file-max` , I think in almost all cases you can just leave it alone. If you are running a very busy server of some kind and actually running out of file handles, then you can increase it -- but the value you need to increase it to will depend on exactly what kind of server you are running and what the load on it is. In general you would just need to increase it until you don't run out of file handles any more, or until you realize you need more memory or more systems to handle the load. The gain from "tuning" things by reducing file-max below the default is so minimal as to not be worth thinking about -- my phone works fine with an fs-max value of 83588.

By the way, the modern kernel already uses a rule of thumb to set file-max based on the amount of memory in the system; from `fs/file_table.c` in the 2.6 kernel:

```
/*
 * One file with associated inode and dcache is very roughly 1K.
 * Per default don't use more than 10% of our memory for files.
 */

n = (mempages * (PAGE_SIZE / 1024)) / 10;
files_stat.max_files = max_t(unsigned long, n, NR_FILE);
```

and `files_stat.max_files` is the setting of `fs.file-max` ; this ends up being about 100 for every 1MB of ram.

ulimits of course are about limiting resources allocated by users or processes. If you have multiple users or another situation like that, then you can decide how you want to divide up system resources and limit memory use, number of processes, etc. The definitive guide to the details of the limits you can set is the setrlimit man page (and the kernel source, of course).

To get a PAGE_SIZE value, use a command "getconf PAGE_SIZE", mostly it is 4096. – Gnought May 23 '15 at 18:08

Typically larger systems like Oracle or SAP recommend a very high limit in order to never be affected by it. I can only recommend to use this approach. The data structures will be allocated dynamically, so as long as you dont need them they dont use up memory. If you actually need them it will not help you to limit them, because if the limit is reached the application will normally crash.

> fs.file-max = 6815744 # this is roughly the default limit for a 70GB RAM system

The same is true for the user rlimits (nofile), you will use 65535.

Note that both recommendations are only good for dedicated servers with on critical application and trusted shell users. A multi-user interactive shell host must have a restrictive max setting.

Where do you get that ratio of 97 files per 1 MB? Also thanks for the answer :) – greggles Mar 9 '15 at 15:08

2 @greggles I actually dont remeber it was from some linux vendors sizing guide for oracle or sap. The actual formular is "use 10% of memory based on 1k/filehandle" as can be seen here: git.kernel.org/cgit/linux/kernel/git/stable/linux-stable.git/… – eckes Mar 9 '15 at 18:23

To be exact, it should be 65GB Ram, the file-max = 6815744 <-- be as (65*1024*1024*1024/1k)*10% – Gnought May 23 '15 at 18:07