

Предисловие: к сожалению, в моём случае процессы не убивались, система просто намертво зависала. Не знаю с чем это связано, пытался починить разными способами, менял настройки oom_killer, пытался медленней забивать память, и т.д. но ничего не помогло. Поэтому к каждому эксперименту есть описание графика, который я получил, и описание ожидаемого поведения, как мне кажется каким, должен быть график, если бы процессы аварийно завершались.

Параметры:

- Общий объем оперативной памяти: 15Gi
- Объем раздела подкачки: 4Gi
- Объем свободной физической памяти: 12Gi
- Объем свободного пространства в разделе подкачки: 4Gi
- Размер страницы виртуальной памяти: 4096

1 Первый эксперимент

1.1 Анализ с одним процессом

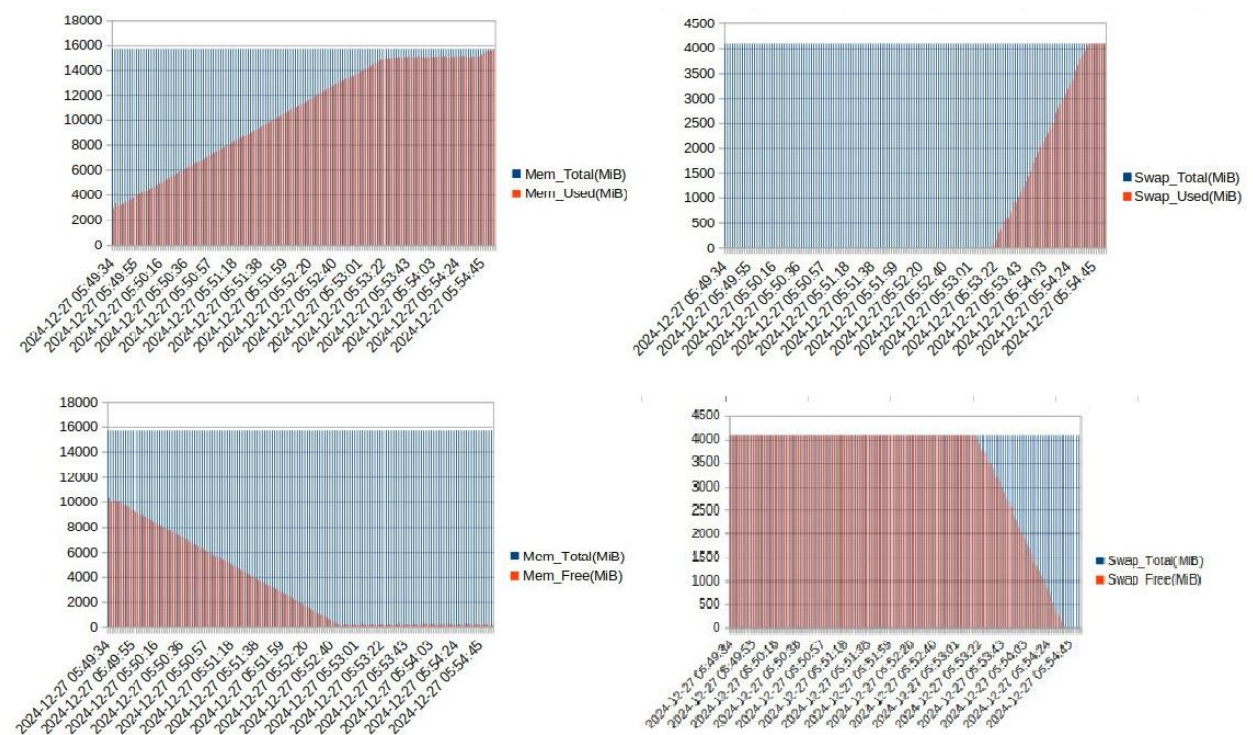


Рис.1 Использование памяти при запуске одного скрипта

1. Фактическое поведение памяти: на графике видно, что скрипт mem.bash постепенно увеличивает число потребляемой оперативной памяти.
2. Фактическое поведение подкачки: на графике видно, что, когда оперативной памяти не хватает, система начинает использовать раздел подкачки.
3. Ожидаемое поведение памяти: должно быть продолжение этого графика, в момент, когда закончилась оперативная память и память подкачки, должен быть резкий спад (как будто график обрубается), т.к. процесс аварийно завершается.
4. Ожидаемое поведение подкачки: рост потребления памяти подкачки начнётся позже, затем такой же резкий спад.
5. Итог: после исчерпания ресурсов процесс должен завершиться аварийно.

1.2 Анализ с двумя процессами

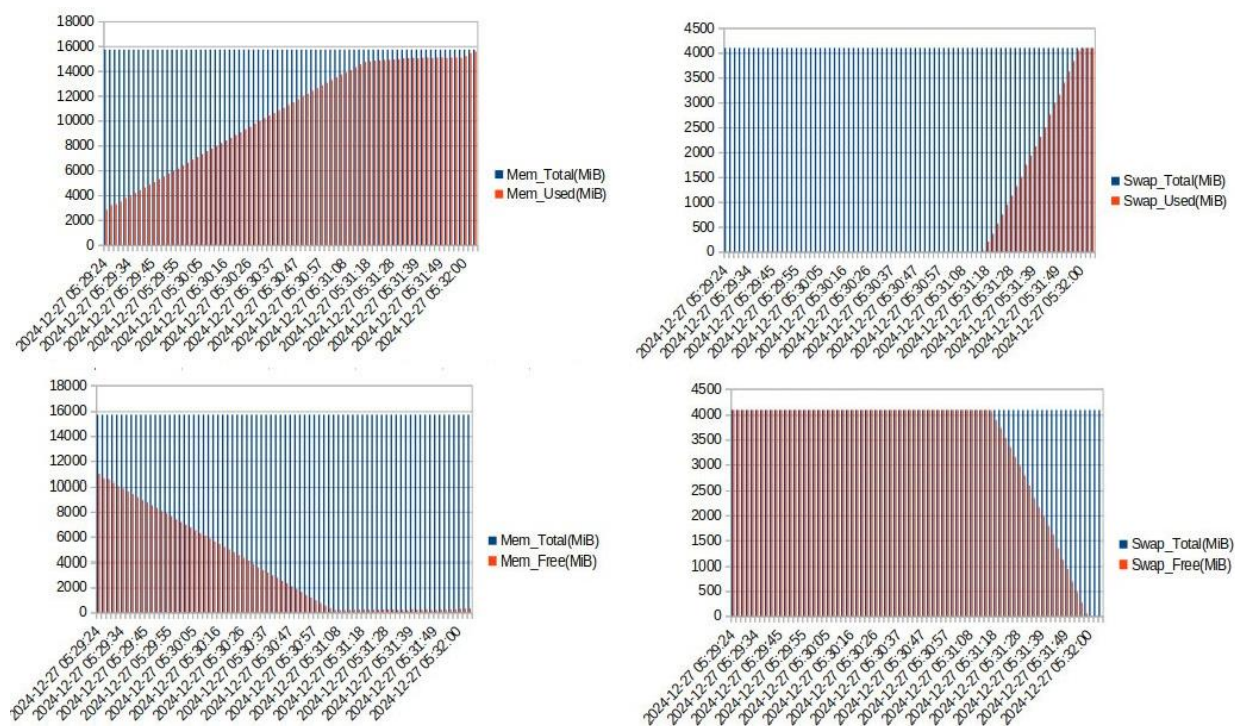


Рис.2 Использование памяти при запуске двух скриптов

1. Фактическое поведение памяти: на графике видно аналогичное первому эксперименту поведение. Заканчивается вся память.
2. Фактическое поведение подкачки: на графике видно аналогичное первому этапу поведение. Заканчивается вся подкачка.
3. Ожидаемое поведение памяти: через какое-то время один из процессов должен завершиться аварийно, произойдёт резкий спад потребляемой памяти. Затем память снова начнёт расти, т.к. второй процесс продолжает потреблять память. Затем резкий спад, т.к. второй процесс аварийно завершился.
4. Ожидаемое поведение подкачки: аналогичная использованию памяти картина в использование подкачки, единственное рост потребления будет начинаться позже.
5. Итог: после исчерпания ресурсов процессы должны завершиться аварийно, но произойдёт это в разное время.

Т.к. в случае первого эксперимента система зависла при 220 000 000, для следующего эксперимента возьмём $N = 22\,000\,000$

2 Второй эксперимент

2.1 Анализ при K = 10

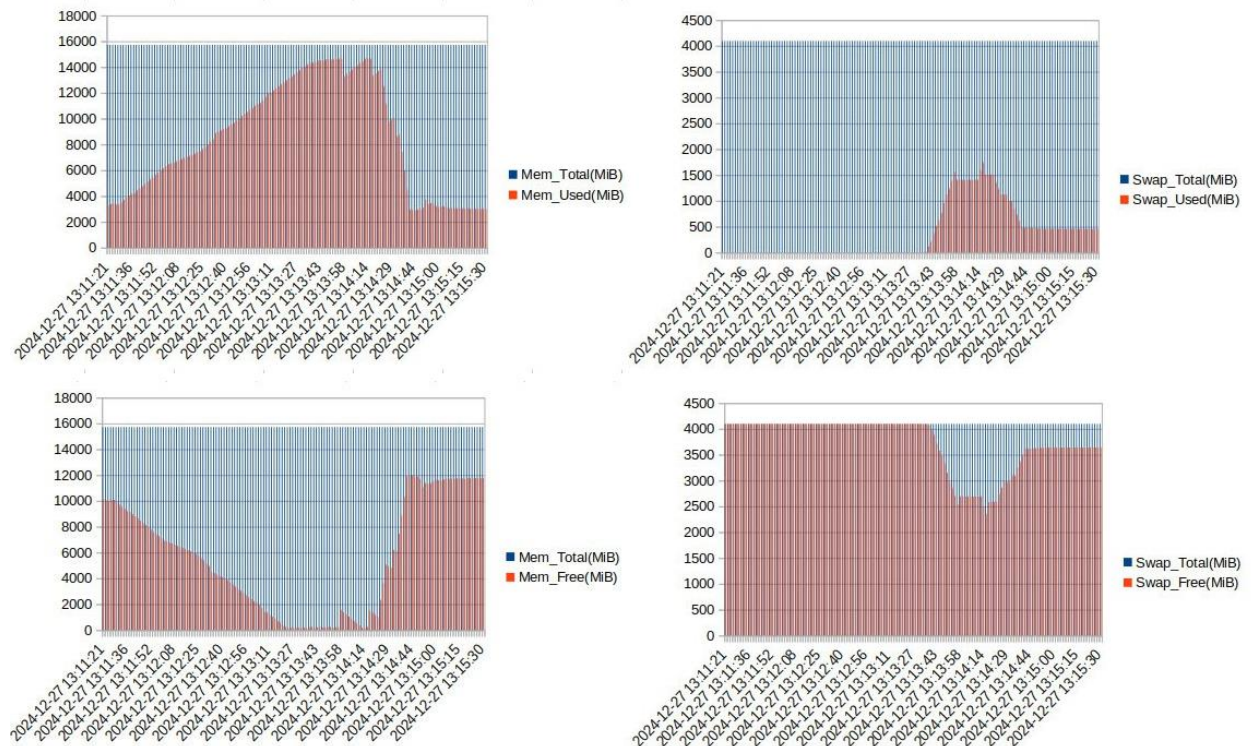


Рис.3 Использование памяти при K = 10

1. Фактическое поведение памяти: на первом графике видно, что при значении K = 10 все процессы завершились успешно. Использование памяти не превышает лимит.
2. Фактическое поведение подкачки: раздел подкачки используется в самый пик использования оперативной память, но также не превышает лимит.
3. Ожидаемое поведение памяти: тут всё хорошо, т.к. процессы выполнились. И не ожидалось, что они будут аварийно завершены.
4. Ожидаемое поведение подкачки: тут всё хорошо, т.к. процессы выполнились. И не ожидалось, что они будут аварийно завершены.
5. Итог: все 10 процессов завершились успешно.

2.2 Анализ при K = 30

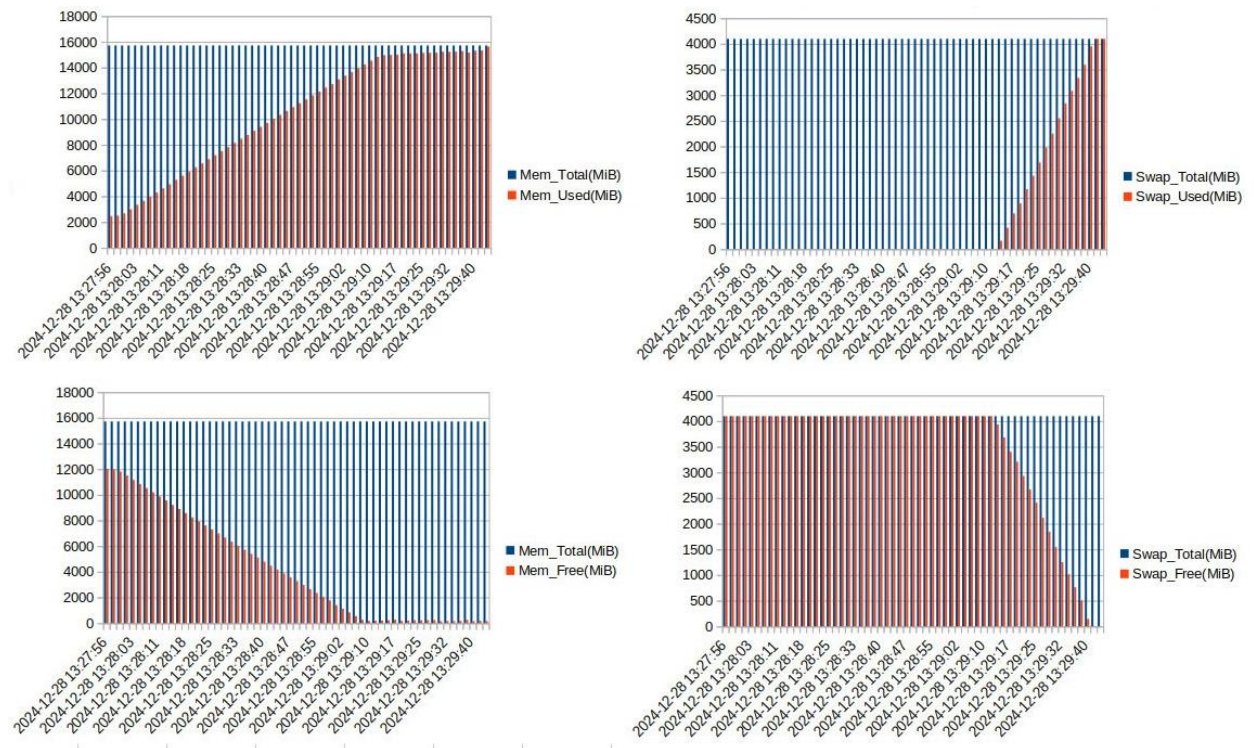


Рис.4 Использование памяти при K = 30

1. Фактическое поведение памяти: на графике видно, что все ресурсы были исчерпаны.
2. Фактическое поведение подкачки: на графике видно, что все ресурсы были исчерпаны.
3. Ожидаемое поведение памяти: т.к. запускается сразу 30 процессов мы должны увидеть похожую на второй этап первого эксперимента картину. Оперативная память должна достигать пика, затем заканчивается память подкачки и происходит аварийная остановка процесса, происходит резкий, но не большой спад, т.к. каждый из 30 процессов потребляет не так много памяти, но вместе они забивают её всю. Такой паттерн должен повториться 30 раз, т.к. 30 процессов. Каждый раз спад должен быть всё “глубже”, т.к. чем меньше процессов, тем больше отводится каждому отдельному процессу памяти. В самом конце должен быть резкий спад и потребление памяти перестаёт расти, т.к. все процессы убиты. График памяти и подкачки похожи сначала на рост, затем на “зубья пилы” и резкий спад.
4. Ожидаемое поведение подкачки: аналогично поведению потреблению памяти только рост начинается позже.
5. Итог: после исчерпания ресурсов процессы должны завершиться аварийно.

Максимальное значение N для K = 30, при котором все процессы выполняются, является = 8 300 000(Подобрал бинарным поиском)