NTT DATA
Global IT Innovator

Linux システムがハングアップしたら メモリ管理を疑え！？

2017年12月1日

NTTデータ先端技術株式会社
半田哲夫

Linux カーネルのメモリ管理サブシステムの5つの特徴

1. メモリ割り当て処理が先に進むことを保証しない
2. 問題が発生する可能性があっても現実に起こるまでは対処しない
3. 故意／悪意あるいはストレステストにより発生する問題には対処しない
4. 自力で真犯人を逮捕できない一般人は相手にしない
5. 他のカーネル開発者の関心を惹かない問題は解決されない

メモリ管理サブシステムに関わるようになったきっかけ

- git bisect 中に偶然見つけた CVE-2013-4312 および CVE-2016-2847
- これらの脆弱性の顛末、および、Linux 4.8 までの出来事については http://I-love.SAKURA.ne.jp/The_OOM_CTF.html というページに纏められています。
  - このページを全部説明するには4時間あっても足りません。
  - しかし、今日の話を理解する上で必要ですので、簡単に説明します。次ページ以降の内容は、その後の展開のごく一部です。

運命の The "too small to fail" memory-allocation rule 発覚から もうすぐ3年

- Linux 4.6 で導入された OOM reaper に起因したリグレッションの多くが修正されました。
- Linux 4.15 では、「 OOM killer が発動できる限りは OOM livelock 状態に陥らないことを証明できる」ことを目指しています。

運命の The "too small to fail" memory-allocation rule 発覚から もうすぐ3年（続き）

- 「 OOM killer を発動できないまま」静かにハングアップしてしまう原因となる依存関係の一部も修正されました。
  - https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/mm/vmscan.c?id=db73ee0d46379922324 4e96e7b7eea73b4a6ec31
  - https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/mm/page_alloc.c?id=e746bf730a76fe53b82c9e6b6da72d58e9ae3565
- その一方で、不安要素も残っています。今日は、それらの中の2つについて取り上げます。

不安要素1

- OOM killer は、__GFP_FS を含まないメモリ割り当て要求では発動しません。そのため、GFP_NOIO や GFP_NOFS なメモリ割り当て要求は、システムをハングアップさせる可能性があります。
  - 意図的な負荷を掛けることでハングアップさせることはできるのですが、現実の負荷で起こることを証明しないと、対処される見込みはありません。
    - http://lkml.kernel.org/r/201703031948.CHJ81278.VOHSFFFOOLJQMt@I-love.SAKURA.ne.jp
  - しかし、普通の利用者に対して、この問題によりハングアップしていることの証明を要求するのは酷なことです。

不安要素1（続き）

- 何のメッセージも出力することなくハングアップしてしまうのを避けるために、Linux 4.9 ではメモリの割り当て処理に10秒以上を要した場合にメッセージを「同期的に」出力するための warn_alloc() という呼び出しが追加されました。
  - しかし、この呼び出しは、同時多発的に呼ばれることによりシステムをハングアップさせることができることが確認されたため、Linux 4.15-rc1 で削除されました。
    - https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/mm/page_alloc.c?id=400e22499dd92613821374c8c6c88c7225359980

不安要素1（続き）

- システムをハングアップさせずに有用な情報を「非同期的に」出力するために、khungtaskd を拡張する watchdog を提案しています。
  - http://lkml.kernel.org/r/1495331504-12480-1-git-send-email-penguin-kernel@I-love.SAKURA.ne.jp
  - http://lkml.kernel.org/r/15108334448-19918-1-git-send-email-penguin-kernel@I-love.SAKURA.ne.jp
  - しかし、OOM livelock は最も関心を惹かない領域であり、賛同者が居なくて困っています。この議論に加わってくれる人を探しています。それが今日話をしている理由です。

不安要素2

- OOM reaper は mmu 環境専用です。nommu 環境で OOM livelock が発生したという報告例が存在しないことを理由に、nommu 環境向けの OOM livelock 回避機構は存在しません。
  - でも、それは普通の利用者が気づいていない(あるいは、デバッグする時間がない)だけという可能性は否定できません。
- そのため、nommu 環境で OOM livelock が起こるのかどうか検証してくれる人を探しています。

MMサブシステムはあなたの助けを必要としています。

- 現在の開発はTBクラスの物理メモリを搭載できる環境がターゲットになっています。
  - soft lockup 回避のために cond_resched() を挿入したり、起動時間短縮のためにメモリの初期化処理を並列化したりなど
- 少量の物理メモリしか搭載していない環境や、 nommu 環境のことは忘れ去られています。
  - OOM livelock の報告例が無いことを理由に nommu 環境向けのコードから MMF_OOM_SKIP をセットする処理を静かに削除したり、 SRCU を常に有効化（ CONFIG_SRCU オプションを削除）してコア処理を書き直すことが提案されたりなど
- 組み込み Linux ユーザの方々、MMの動向を注視して反応するのを忘れないでください。

NTT DATA

Global IT Innovator