

参考

https://techracho.bpsinc.jp/hachi8833/2017_12_28/50109

https://qiita.com/kamina_zzz/items/0540f663832e07ee1b10

http://www.rarul.com/mt/log/2018_09.html

<https://qiita.com/kaityo256/items/bf6563361c502bbf062e>

概要

Ruby の Sidekiq などのマルチスレッド処理でメモリの使用量が増え続けてしまうことがある。メモリリークしているわけではなく、Ruby のメモリ管理と malloc の仕様によりメモリの断片化が発生している。

メモリリークのようにメモリを消費し続けるわけではなく、対数的にある上限に収束するように増加する。

malloc はマルチスレッドでの排他処理の効率を上げるためにスレッド単位メモリアリーナを実装している。

以下引用

1. あるスレッドで malloc を呼び出します。このスレッドは、前回アクセスしたメモリアリーナ(他のアリーナが作成されていない場合はメインのアリーナ)のロック取得を試みます。
2. そのアリーナを利用できない場合は、次のメモリアリーナのロック取得を試みます(メモリアリーナが他にもある場合)
3. 利用できるアリーナがない場合は、アリーナを1つ作成してそれを使う。この新しいアリーナは、連結リストの最後のアリーナにリンクされる。

マルチスレッドかつ I/O 処理を実行する CRuby にスレッド単位メモリアリーナが使われ、新しいメモリアリーナが大量に作成されてしまう。

マルチスレッドでも I/O を行っていない場合は GVM(Global VM Lock) によって制御されるためメモリアリーナの奪い合いは発生しない。

解決策

方法 1 : メモリアリーナを削減する

MALLOC_ARENA_MAX 環境変数でメモリアリーナの最大数を設定できる。

メモリアリーナ数を調整することでメモリの使用量を減らすことができるが、排他処理が多くなるためパフォーマンスの低下に繋がる。

メモリの使用量とパフォーマンスのバランスを確認しながら設定する必要がある。

方法 2 : jemalloc を使う

アロケータを malloc から jemalloc に変更する。

jemalloc もスレッド単位アリーナを実装していますが、malloc で起きる断片化を回避する設計になっている。