jemalloc()

～ Firefox 3.0 爆速の理由 ～

山本和彦
(株)インターネットイニシアティブ
kazu@iij.ad.jp

今度のキツネは爆速だぜ！

http://jp.youtube.com/watch?v=Hlj6u2plNf4

Firefox 3 の爆速技術

- Phoenix
  - Mozilla から派生したブラウザ
  - 後の Firefox
  - AJAX などがない時代に作られた
  - 素朴なメモリー管理
- Firefox 2.x
  - AJAX 時代に素朴なメモリー管理を使い続けていた
  - たくさんのメモリーリーク
- Firefox 3.0
  - 徹底的なメモリーリークの排除
  - FreeBSD から `jemalloc()` を移植して、メモリーを管理
- Firefox 3.1
  - JIT コンパイル(Just In Time Compilation) の導入

jemalloc()

= phkmalloc() + lkmalloc()

さまざまな malloc()

- krmalloc()
  - B Kernighan, D Ritchie
- dlmalloc()
  - Doug Lea
- phkmalloc()
  - Poul-Henning Kamp
- lkmalloc()
  - P Larson, M Krishnan
- jemalloc()
  - Jason Evans
- その他多数

malloc(3) とメタデータ

- malloc(3) の仕様
  - void *malloc(size_t size);
  - void free(void *ptr);
- メタデータ
  - free() するデータのサイズ
    - free() にはサイズを指定しない
  - free() されたデータの管理
    - malloc() で再利用する

krmalloc()

- The C Programming Language
  - The Art of Computer Programming の
    帯域内自由リスト(in-band free list) に基づいた実装
- 他の malloc() に多大な影響を与えた
- データとメタデータが混在
  - 確保したサイズ以上のデータを書き込むとメタデータが壊れる
  - 誤って2回 free() すると何が起こるか分らない

dlmalloc()

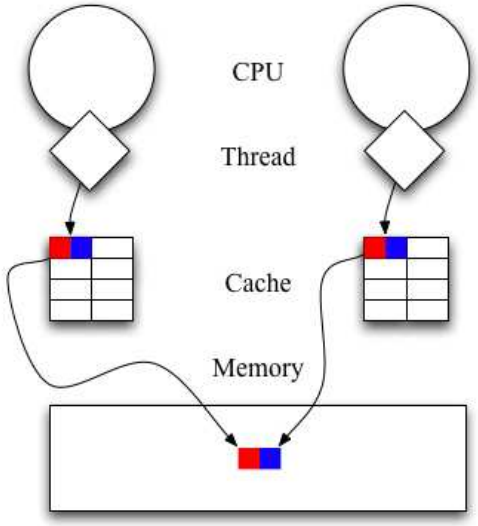
- GNU の libc と Linux に採用
- データとメタデータが混在
  - 確保したサイズ以上のデータを書き込むとメタデータが壊れる
  - 誤って2回 `free()` すると何が起こるか分らない
  - 註) バージョンによってデータ構造は変わってきている
- 参考文献
  - C/C++ セキュアコーディング

phkmalloc()

- FreeBSD 6 までの malloc()
- データとメタデータを分離
  - 確保したサイズ以上のデータを書き込んでも
    メタデータは壊れない
  - 誤って2回 free() すると、警告が出る
- ページを意識したデータ管理
  - 1ページは 4K
  - Large (> 2K)
    - 適切な数のページが丸ごと割り当てられる
  - Small (≦ 2K)
    - 2の累乗の大きさを持つ「囲い」として管理される
    - データは、囲いの大きさへ丸められる
    - 同じ大きさの囲いは、同じページにある
    - 補助データ(上記メタデータとは別)と囲いを、なるべく同じページ配置
- 参考文献
  - デーモン君のソース探検
  - Malloc(3) revisited
    - http://phk.freebsd.dk/pubs/malloc.pdf

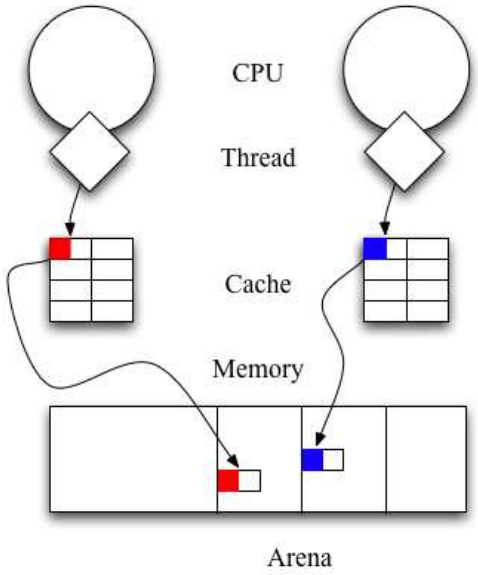
phkmalloc() の問題点

- マルチ CPU を考慮していない
  - False cache line sharing 問題が起こる



lkmalloc()

- スレッドごとに、別の領域「アリーナ」から
  メモリーを割り当てる

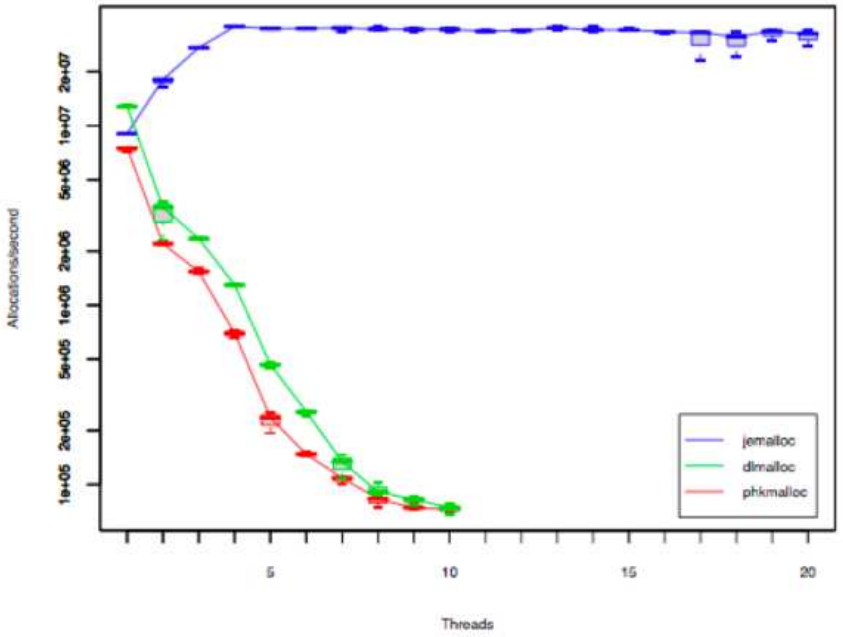


jemalloc()

- FreeBSD 7 からの malloc()
- phkmalloc() + lkmalloc()
- 「囲い」と「ページ」に加えて「塊」の層を導入
  - 塊の大きさは 2MB (512ページ)
  - Huge (> 1M)
    - 適切な数の塊を割り当てる
  - Large と Small のために、「塊」をアリーナに割り当てる
    - 各アリーナは、各スレッドに割り当てられる
  - Large (> 2K, ≦1M)
    - アリーナ内で、適切な数のページが丸ごと割り当てられる
  - Small (≦ 2K)
    - アリーナ内で、2の累乗の大きさを持つ「囲い」として管理される
- 参考文献
  - A Scalable Concurrent malloc(3) Implementation for FreeBSD
    - http://people.freebsd.org/~jasone/jemalloc/bsdcan2006/jemalloc.pdf
    - http://people.freebsd.org/~jasone/jemalloc/bsdcan2006/BSDcan2006_slides.pdf

malloc-test

- 4 CPU で実験



- 4 CPU まで性能が向上し、その後安定