[redacted]

Linux Kernel Slub Allocator

陈统
(chentong@s[redacted]com)

[redacted]

Contents

0. 承前（1）

- Slab Layer的作用及实现
  - 为kernel其他部分提供快速有效的 objects pool(cache)
    - 对 cpu 提供 hardware-cache-warned objects
    - 降低内存的页内碎片
  - 内核中提供有三种实现 SLOB/SLAB/SLUB Allocator
- 概念结构
  - pool(cache) 包含 slabs，而slab 又包含 objs
  - slab 通常由连续的几个页构成(from buddy system)
  - slab/objs 从 NUMA Node 中获取，又提供给运行于 cpu 之上代码的使用，所以 cache 设计时从是需要纠缠于 cpu 和 mem node

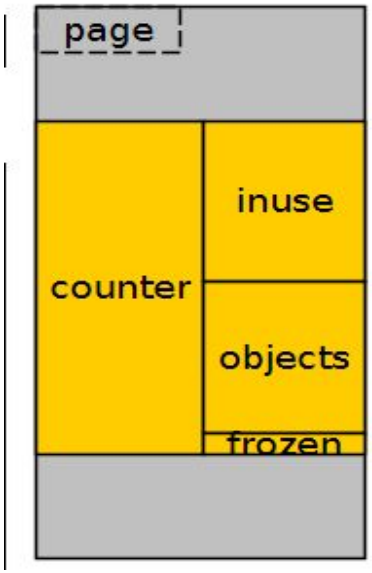
0. 承前（1）

- SLUB 较之于 SLAB
  - 通过去除 Array Cache(AC) 以降低实现的复杂性
  - 去除 slab 结构体，而降低 metadata 对内存的消耗；反之以在 struct page 中加塞字段来保存必须的元数据
  - 据称较slab, slub性能提升5%–10%,内存消耗减少50%，SP手机也采用

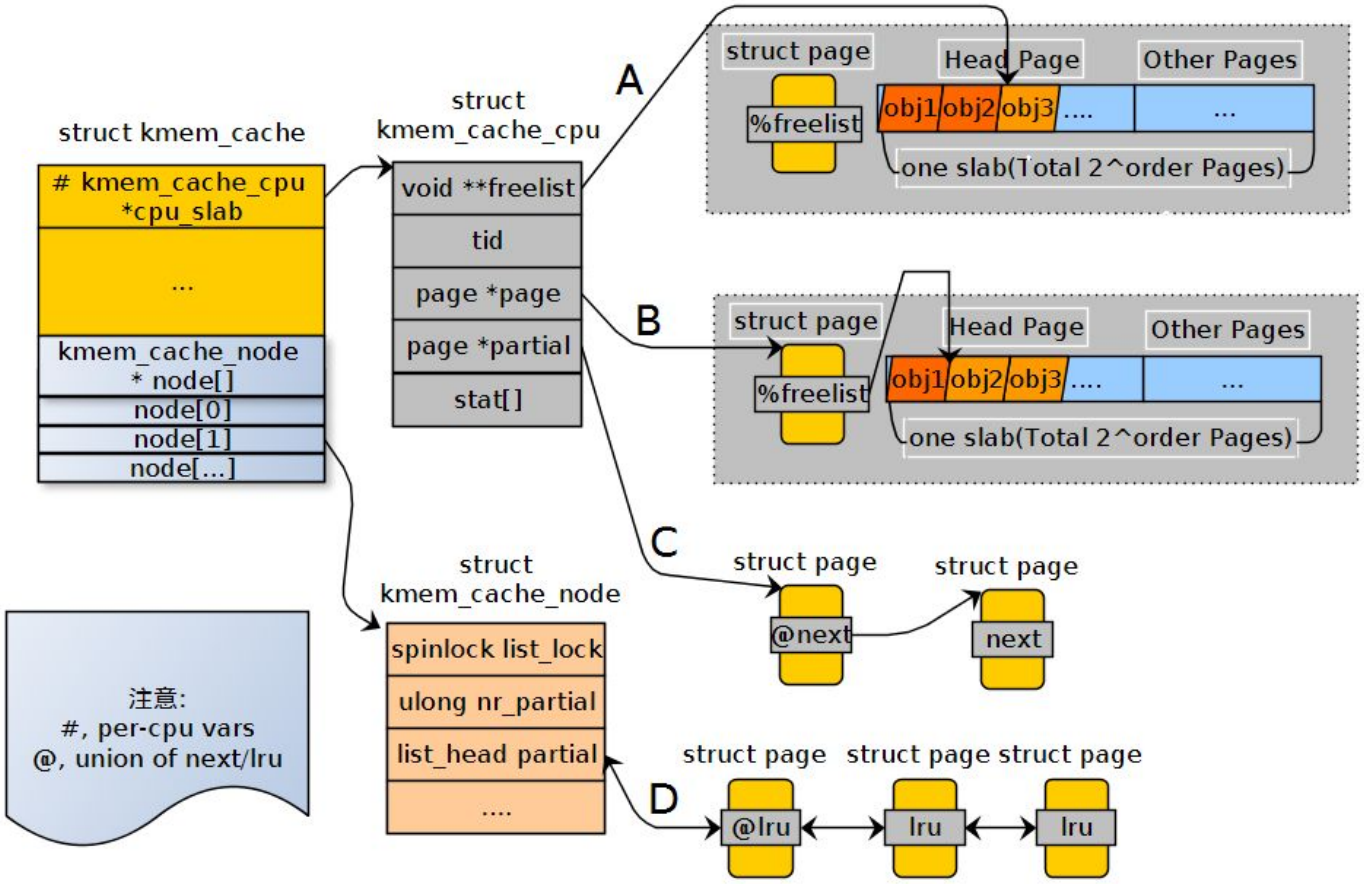
1. SLUB对 struct page 的加塞

- In the 2nd double word
  - freelist:
    - 指代下一个可用的 object
  - counter 与 inuse/objects/frozen三字段的 union

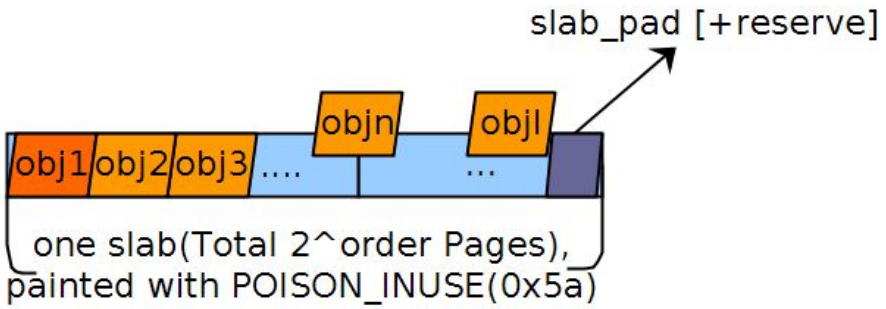
- In the 3st double word
  - next
    - 构成 partial slab list 用
    - 仅适用在 slab 的第一个 page

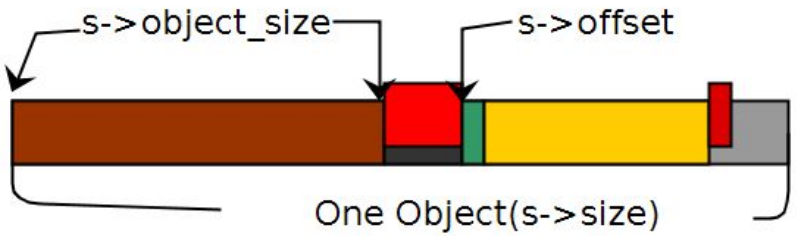


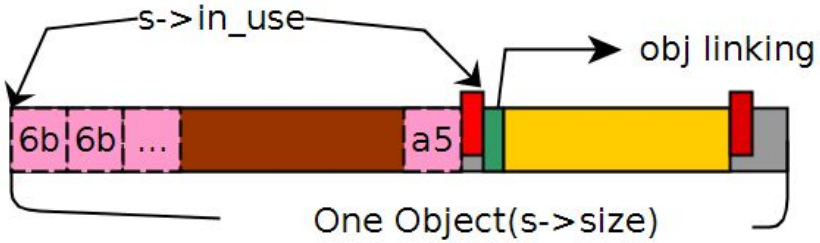
2. SLUB Allocator 的整体结构（1）

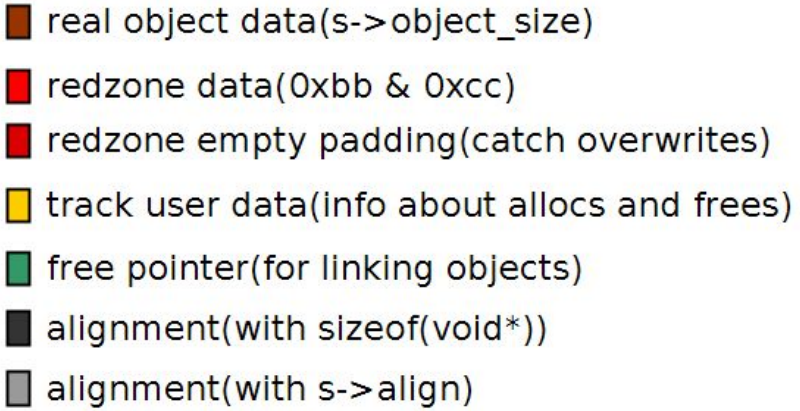


3. SLUB中 slab objects 的微观结构



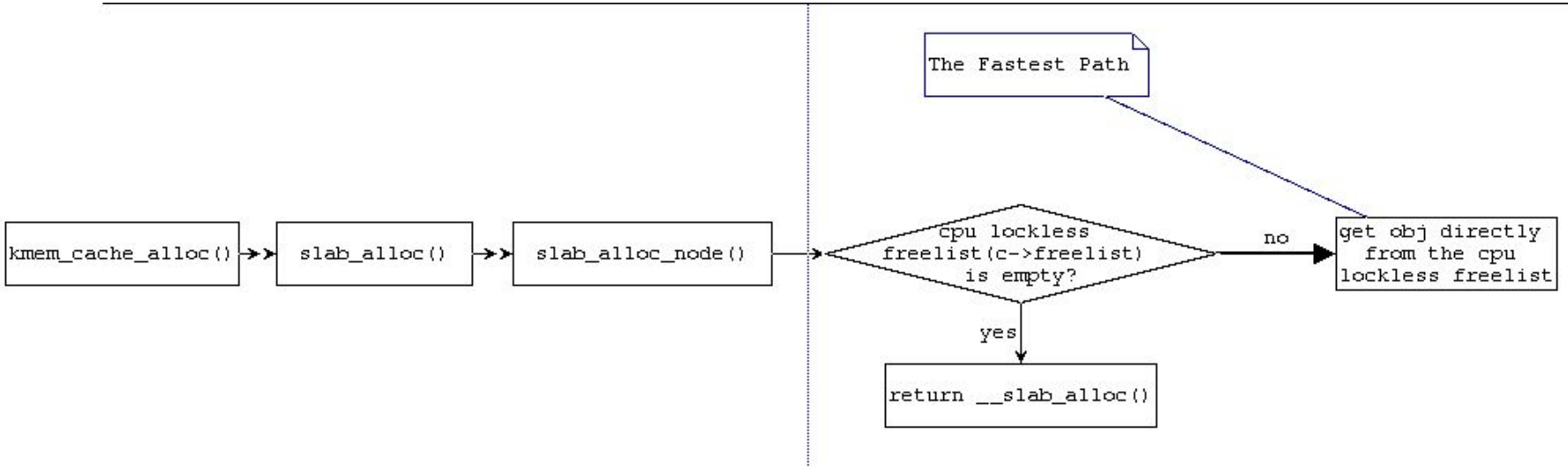






4. Objects 在 SLUB 中的分配 (1)

- 以 kmem_cache_alloc() 为入口



- 五种分配的情况
  - 直接从 cpu lockless list 中获取(最快路径)

    A != NULL, 直接摘取object

4. Objects 在 SLUB 中的分配 (2)

- 五种分配的情况(续1)
  - 因 A==NULL, 所以需从cpu regular list中获取(较快)
    - 返回 B 所指 slab 中第一个 obj
    - 将剩余 objs 挂接到 A(lockless list) 上
    - 置 B 为 NULL
  - 因 A&B==NULL, 所以需从 cpu partial slab lists 中取(适中)
    - 将cpu partial slab lists 中的首个 slab 挂接到B上
    - 设置 A为NULL后，继续按前一条路径分配

4. Objects 在 SLUB 中的分配 (3)

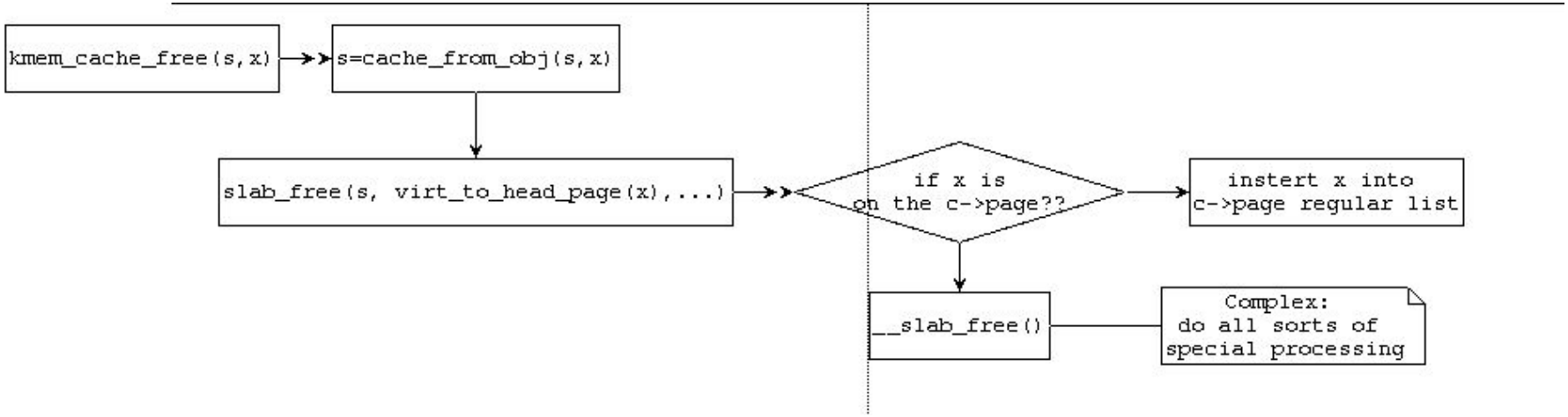
- 五种分配的情况(续2)
  - 因 A&B&C==NULL, 所以需从 node partial slab lists 中取(较慢)
    - 取第一个 slab 挂到 B 上
    - 从第二个 slab 开始取足够数量的 slab 挂到 C 上置 B 为 NULL
    - 返回 B 所指 slab 中的第一个可用 obj
    - 将 B 所指 slab deactivate 掉后，设置A/B均为NULL

4. Objects 在 SLUB 中的分配 (4)

- 五种分配的情况(续3)
  - 因 A&B&C&D==NULL, 所以需分配一个全新的 slab 后再取(最慢路径，因为还要初始化 slab)
    - 将新分配的 slab 挂到 B 上
    - 返回 B 所指 slab 中的第一个可用 obj
    - 将 B 所指 slab deactivate 掉后，设置A/B均为NULL

5. 返回 Objects 到 Slub 中 (1)

- 以 kmem_cache_free() 为入口
  - 直接返回到 c->page 中(最快路径)



5. 返回 Objects 到 Slub 中（2）

- 以 kmem_cache_free() 为入口(续1)
  - 使用 __slab_free() 处理其他各种复杂情况
    - 释放前如果对应 slab 为满，且该 slab 不在对应 node partial list中，则置入到 cpu partial list中
    - 释放后如果对应 slab 中变为空，则丢弃该 slab

Thanks !

陈统
(chentong@[redacted].com)