mbuf allocator

Robert N. M. Watson

16 May 2007

FreeBSD Developer Summit
BSDCan 2007
mbuf allocator

• Historically
  – Dedicated memory allocator for mbufs and clusters
  – Able to run in low level network contexts
  – Used a special mbuf memory map

• Today
  – Thin wrapper around UMA slab allocator
    • Per-CPU caches, complex cache behavior
    • Able to cache mbufs with attached clusters
  – Zones for several different cluster sizes
struct mbuf

All mbufs have basic header

mbufs may use internal or external storage (clusters, ...)

Packet header mbufs have additional packet headers, but reduced space for internal storage

struct m_hdr
m_hdr

struct pkthdr
MH_pkthdr

char
M_databuf
[MLEN]

struct m_ext
MH_ext

char
MH_databuf
[MHLEN]
mbuf chains and queues

- Packets stored as mbuf chains linked by m_next
  - Efficient append and prepend operations
- Packet queues link chains via m_nextpkt
  - Managed as part of ifqueue, sockbuf, etc.
mbuf external storage

- External storage
  - Clusters: 2k, 4k, 9k, 16k
  - Sendfile pages
  - User process pages
- Reference counted
  - Reference counts stored by UMA
  - Used during bridging, multiple socket delivery, etc
UMA – Universal Memory Allocator

• Slab allocator per Bonwick, et al.
  – Zones define fixed-size objects with init/ctor/dtor/fini
  – Slabs allocated using VM system
  – Objects cached partially or fully initialized
  – Per-CPU cache w/lock-free alloc/free

• Additional facilities
  – Object reference count storage as part of zone
  – “Secondary zones” allow variations on object types
  – Special support for malloc(9) large objects
UMA: Zones, Kegs, and Caches

struct uma_zone

uz_cpu[1]
uz_cpu[...]
uz_cpu[0]
uz_cpu[0]

uz_full_bucket
uz_free_bucket

struct uma_keg

uk_lock
uk_size
uk_part_slab
uk_free_slab
uk_full_slab
Key code paths to inspect

- mbuf.h, uma.h: definitions
- uma_core.c: UMA internals
  - uma_zalloc_arg(), uma_zfree_arg()
- kern_mbuf.c: mbuf allocator internals
  - mbuf_init(), mbuf_{ctor,dtor}._{clust,mbuf}(),
    mb_{zinit,zfini,ctor}_pack(), mb_reclaim()
- uipc_mbuf.c: mbuf allocator public interfaces
  - m_getm2(), m_freem(), m_extadd(), mb_free_ext(),
    mb_dupcl(), m_dup_pkthdr()