Malloc internal

Bruno Pujos

March 4, 2014

Plan

Malloc history

- Doug Lea's malloc
- Wolfram Gloger's malloc

This is not the fastest, most space-conserving, most portable, or most tunable malloc ever written. However it is among the fastest while also being among the most space-conserving, portable and tunable.

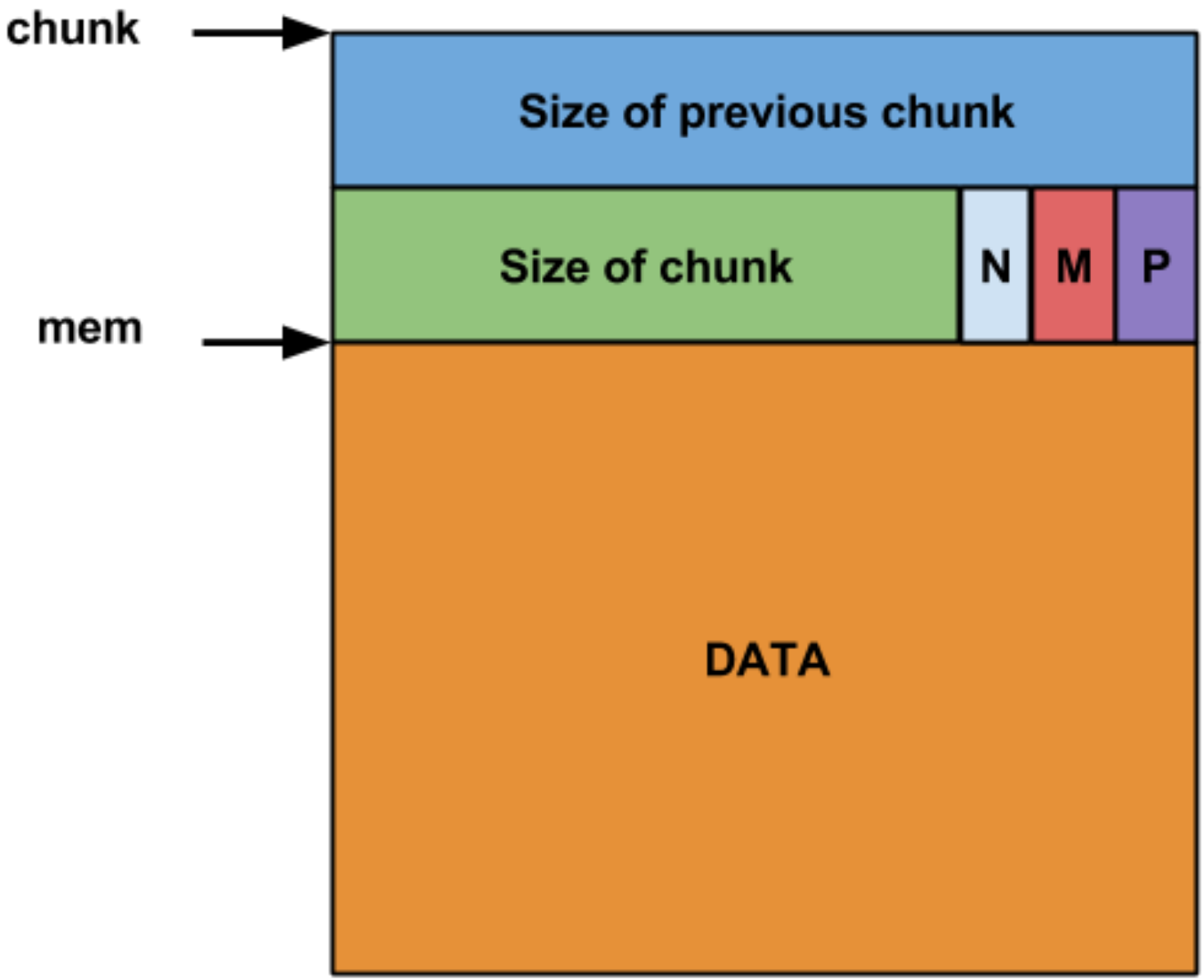
Consistent balance across these factors results in a good general-purpose allocator for malloc-intensive programs.

2 Malloc internal

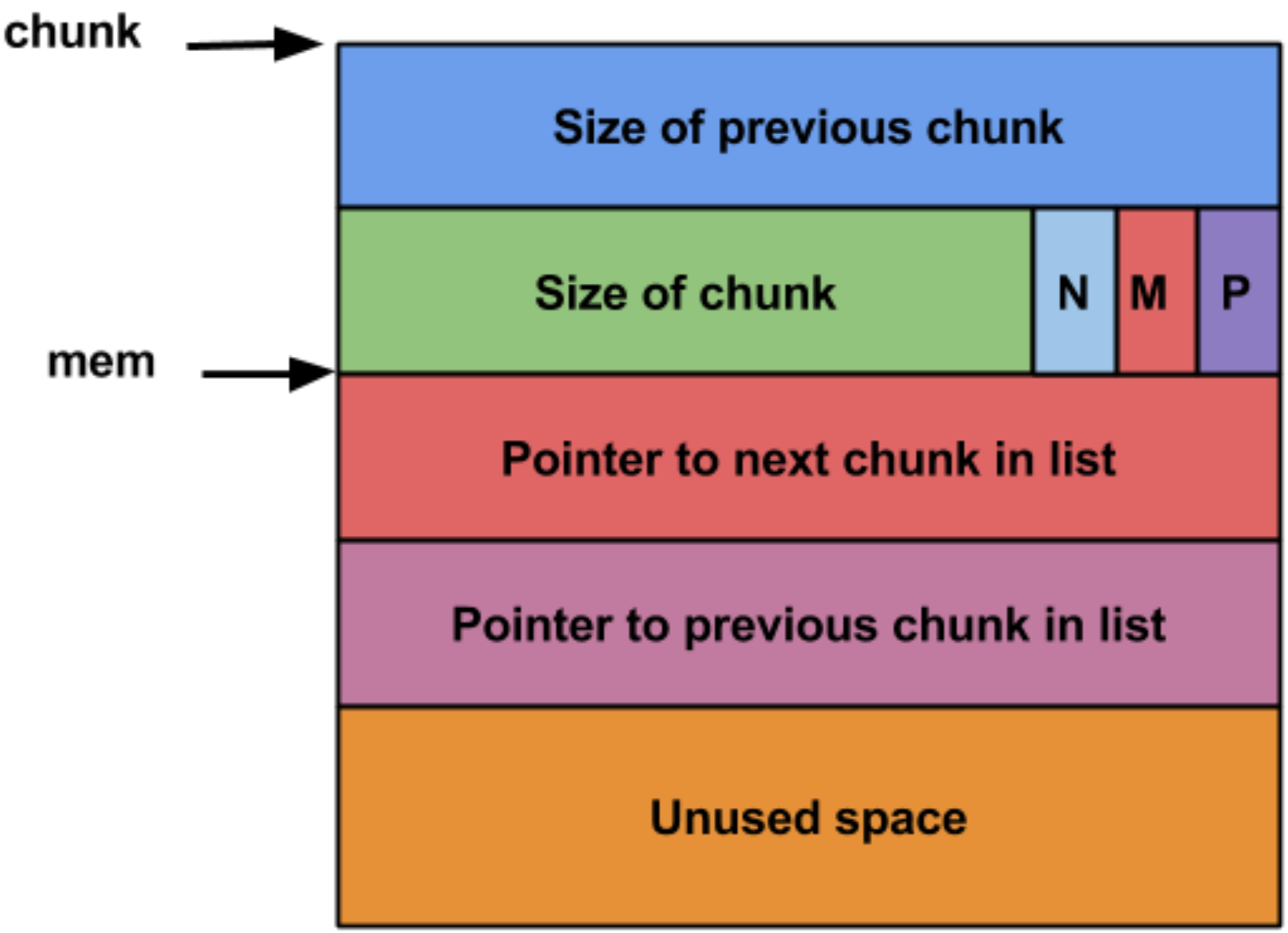
Malloc internal

- Two principales files : malloc.c, arena.c
- 4 differents type of request (size for 32bits):
  - small <= 64 bytes
  - medium > 64 bytes and < 512 bytes
  - large >= 512 bytes
  - very large >= 128 Kbytes
- Alignement of 2 * sizeof(size_t)
- Malloc use ”chunk” for keeping track of the data.

Allocated chunk



Free chunk


chunk
mem
Size of previous chunk
Size of chunk
N
M
P
Pointer to next chunk in list
Pointer to previous chunk in list
Unused space

Chunk struct


```
struct malloc_chunk {

  INTERNAL_SIZE_T       prev_size;
  INTERNAL_SIZE_T       size;

  struct malloc_chunk* fd;
  struct malloc_chunk* bk;

  struct malloc_chunk* fd_nextsize;
  struct malloc_chunk* bk_nextsize;
};
```

Malloc State

```
struct malloc_state
{
  mutex_t mutex;
  int flags;
  mfastbinptr fastbinsY[NFASTBINS];
  mchunkptr top;
  mchunkptr last_remainder;
  mchunkptr bins[NBINS * 2 - 2];
  unsigned int binmap[BINMAPSIZE];
  struct malloc_state *next;
  struct malloc_state *next_free;
  INTERNAL_SIZE_T system_mem;
  INTERNAL_SIZE_T max_system_mem;
};
```

Fastbin

- For small alloc
- Keep by size (looking directly for the good size)
- Single linked list
- LIFO system
- No consolidation

Common case

- Free chunk are regroup in bins
- Regroupment make by size :
  - 64 bins of size 8
  - 32 bins of size 64
  - 16 bins of size 512
  - 8 bins of size 4096
  - 4 bins of size 32768
  - 2 bins of size 262144
  - 1 bins of size ”what’s left”

Bins

- Chunks in bins are kept by size order (for best-fit)
- Chunks of the same size are linked before, and allocated from the end, LRU allocation order
- Binmap

Plan

3 Conclusion

Conclusions

- That's just the beginning. . .
- Possibility of doing stats (for optimization)
- -DMALLOC_DEBUG . . .
- Lot's of intersting paper on metadata corruptions :
  - "Advanced Doug Lea's malloc exploits"
  - "Once upon a free()"
  - "Vudo malloc tricks"
  - . . .

Link

- http://gee.cs.oswego.edu/dl/html/malloc.html
- http://malloc.de/en/index.html
- http://ftp.gnu.org/gnu/libc/