




Slub Memory Allocator -13- (slabinfo)

 2016-06-01 (<http://jake.dothome.co.kr/slub-slabinfo/>)  Moon Young-il
(<http://jake.dothome.co.kr/author/admin/>)  Linux Kernel (<http://jake.dothome.co.kr/category/linux/>)

<kernel v5.0>

/proc/slabinfo

```
$ cat /proc/slabinfo
slabinfo - version: 2.1
# name                <active_objs> <num_objs> <objsize> <objperslab> <pagesperslab> :
tunables <limit> <batchcount> <sharedfactor> : slabdata <active_slabs> <num_slabs>
<sharedavail>
nf_contrack_1      1035    1035    344    23    2 : tunables    0    0    0 : slabda
ta      45      45      0
ext4_groupinfo_4k  2112    2112    168    24    1 : tunables    0    0    0 : slabda
ta      88      88      0
ip6-frags          0        0    248    16    1 : tunables    0    0    0 : slabda
ta      0        0      0
ip6_dst_cache      126     126    384    21    2 : tunables    0    0    0 : slabda
ta      6        6      0
RAWv6             104     104   1216    26    8 : tunables    0    0    0 : slabda
ta      4        4      0
UDPLITEv6         0        0   1216    26    8 : tunables    0    0    0 : slabda
ta      0        0      0
UDIPv6            156     156   1216    26    8 : tunables    0    0    0 : slabda
ta      6        6      0
tw_sock_TCPv6      0        0    272    30    2 : tunables    0    0    0 : slabda
ta      0        0      0
request_sock_TCPv6 0        0    328    24    2 : tunables    0    0    0 : slabda
ta      0        0      0
TCPv6             56       56   2304    14    8 : tunables    0    0    0 : slabda
ta      4        4      0
cfq_io_cq         216     216    112    36    1 : tunables    0    0    0 : slabda
ta      6        6      0
bsg_cmd           0        0    312    26    2 : tunables    0    0    0 : slabda
ta      0        0      0
xfs_icr           0        0    144    28    1 : tunables    0    0    0 : slabda
ta      0        0      0
xfs_ili           0        0    152    26    1 : tunables    0    0    0 : slabda
ta      0        0      0
xfs_inode         0        0   1216    26    8 : tunables    0    0    0 : slabda
ta      0        0      0
xfs_efd_item      0        0    400    20    2 : tunables    0    0    0 : slabda
ta      0        0      0
xfs_log_item_desc 128     128     32   128    1 : tunables    0    0    0 : slabda
ta      1        1      0
xfs_da_state      0        0    480    17    2 : tunables    0    0    0 : slabda
ta      0        0      0
xfs_btree_cur     0        0    208    19    1 : tunables    0    0    0 : slabda
ta      0        0      0
xfs_log_ticket    0        0    184    22    1 : tunables    0    0    0 : slabda
ta      0        0      0
```

For each slab cache, the following information is printed:

- name
 - Slab Name
 - s->name
- <active_objs>

- The number of objects in use (even free objects in the per-cpu count towards the number of objects in use)
- `n->total_objects` of all nodes – (sum of `page->objects` – `page->inuse` of slab pages in the per node partial list)
- `<num_objs>`
 - Number of all objects on the allocated slab page (free + inuse)
 - Sum of `n->total_objects` of all nodes
- `<objsize>`
 - Slab object size (including meta information)
 - `s->size`
- `<objperslab>`
 - The number of objects that go into each slab page
 - Number of objects recorded in `s->oo`
- `<pagesperslab>`
 - The number of pages that go into each slab page
 - 2^{order} recorded in `s->oo`
- `<limit>`
 - Maximum number of objects to be cached – unused in the slub (always output as 0)
- `<batchcount>`
 - Number of objects that can be refilled at once – unused in a slub (always output as 0)
- `<sharedfactor>`
 - Not in use on slub (always output as 0)
- `<active_slabs>`
 - Number of slabs in use = equal to total number of slabs
 - Sum of `n->nr_slabs` of all nodes
- `<num_slabs>`
 - Total number of slabs = equal to the number of slabs in use
 - Sum of `n->nr_slabs` of all nodes
- `<sharedavail>`
 - Not in use on slub (always output as 0)

/sys/kernel/slub directory

Slab caches are managed in the `/sys/kernel/slub` directory and are categorized as follows:

- A directory with a non-merge slab cache name is created.
 - e.g. TCP Directory
- Mergeable slab cache
 - An auto-generated slab cache with a unique name pointed to by the alias cache, resulting in a directory starting with the string `:"`.
 - Example: `A-0001088` Directory
- Create a link file with the merged alias slab cache that points to the original cache.
 - e.g. `lrwxrwxrwx 1 root root 0 Nov 19 15:42 UDP -> :A-0001088`

The rules for mergable slab cache names are as follows:

- format
 - ":" character + [[d][a][F][A]-] + unique 7-digit number

The meaning of each option letter is as follows:

- d
 - DMA-enabled slap cache
 - Using SLAB_CACHE_DMA Flags
- D
 - DMA32 Enabled Slap Cache
 - Using SLAB_CAHE_DMA32 flags (added in kernel v5.1-rc3)
- a
 - Reclaimable Slap Cache
 - Using SLAB_RECLAIM_ACCOUNT Flags
- F
 - consistency checks: allowed slab cache
 - Using SLAB_CONSISTENCY_CHECKS Flags
- A
 - MEMCG Control Allowed Slap Cache
 - Using SLAB_ACCOUNT Flags
- t
 - Using SLAB_NOTRACK flags (removed in kernel v.4.15-rc1)
 - Note: kmemcheck: remove whats left of NOTRACK flags
(<https://github.com/torvalds/linux/commit/d8be75663cec0069b85f80191abd2682ce4a512f#diff-e3a2d857926322961b78602dfb5dfb7f>)

The merged slab caches as follows refer to the original cache directories that can be merged.

```
$ ls /sys/kernel/slab -la
...
lrwxrwxrwx    1 root root 0 Nov 19 15:32 PING -> :A-0000960
drwxr-xr-x    2 root root 0 Nov 19 15:32 RAW
drwxr-xr-x    2 root root 0 Nov 19 15:32 TCP
lrwxrwxrwx    1 root root 0 Nov 19 15:32 UDP -> :A-0001088
lrwxrwxrwx    1 root root 0 Nov 19 15:32 UDP-Lite -> :A-0001088
lrwxrwxrwx    1 root root 0 Nov 19 15:32 UNIX -> :A-0001024
lrwxrwxrwx    1 root root 0 Nov 19 15:32 aio_kiocb -> :0000192
...
```

The following shows the full slap cache.

```

$ ls /sys/kernel/slub
:0000024      :a-0000256      files_cache      nfs_inode_cache
:0000032      :a-0000360      filp             nfs_page
:0000040      PING            fs_cache         nfs_read_data
:0000048      RAW            fsnotify_mark    nfs_write_data
:0000056      TCP            fsnotify_mark_connector nsproxy
:0000064      UDP            hugetlbfs_inode_cache numa_policy
:0000080      UDP-Lite       iint_cache       p9_req_t
:0000088      UNIX          inet_peer_cache  pde_opener
:0000104      aio_kiocrb     inode_cache      pid
:0000128      anon_vma       inotify_inode_mark pid_namespace
:0000192      anon_vma_chain iommu_iova       pool_workqueue
:0000208      asd_sas_event  ip4-frags        posix_timers_cache
:0000216      audit_buffer   ip_dst_cache     proc_dir_entry
:0000240      audit_tree_mark ip_fib_alias     proc_inode_cache
:0000256      bdev_cache     ip_fib_trie      radix_tree_node
:0000320      bio-0          isp1760_qh       request_queue
:0000344      bio-1          isp1760_qtd      request_sock_TCP
:0000384      bio_integrity_payload isp1760_urb_listitem rpc_buffers
:0000448      biovec-128     jbd2_inode       rpc_inode_cache
:0000464      biovec-16     jbd2_journal_handle rpc_tasks
:0000512      biovec-64     jbd2_journal_head sas_task
:0000640      biovec-max     jbd2_revoke_record_s scsi_data_buffer
:0000704      blkdev_ioc     jbd2_revoke_table_s sd_ext_cdb
:0000768      buffer_head    jbd2_transaction_s seq_file
:0000896      configfs_dir_cache kernfs_node_cache sgpool-128
:0001024      cred_jar       key_jar           sgpool-16
:0001088      debug_objects_cache khugepaged_mm_slot sgpool-32
:0001984      dentry         kioctx            sgpool-64
:0002048      dio           kmalloc-128      sgpool-8
:0002112      dmaengine-unmap-128 kmalloc-1k       shared_policy_node
:0004096      dmaengine-unmap-16 kmalloc-256      shm_inode_cache
:A-0000032     dmaengine-unmap-2 kmalloc-2k       sighand_cache
:A-0000040     dmaengine-unmap-256 kmalloc-4k       signal_cache
:A-0000064     dnotify_mark    kmalloc-512      sigqueue
:A-0000072     dnotify_struct  kmalloc-8k       skbuff_ext_cache
:A-0000080     dquot           kmalloc-rcl-128  skbuff_fclone_cache
:A-0000128     eventpoll_epi   kmalloc-rcl-1k   skbuff_head_cache
:A-0000192     eventpoll_pwq   kmalloc-rcl-256  sock_inode_cache
:A-0000256     ext2_inode_cache kmalloc-rcl-2k   squashfs_inode_cache
:A-0000704     ext4_allocation_context kmalloc-rcl-4k   task_delay_info
:A-0000960     ext4_extent_status kmalloc-rcl-512  task_group
:A-0001024     ext4_free_data  kmalloc-rcl-8k   task_struct
:A-0001088     ext4_groupinfo_4k kmem_cache       taskstats
:A-0005120     ext4_inode_cache kmem_cache_node  tcp_bind_bucket
:a-0000016     ext4_io_end     ksm_mm_slot      tw_sock_TCP
:a-0000024     ext4_pending_reservation ksm_rmap_item    uid_cache
:a-0000032     ext4_prealloc_space ksm_stable_node  user_namespace
:a-0000040     ext4_system_zone mbcache          uts_namespace
:a-0000048     fanotify_event_info mm_struct        v9fs_inode_cache
:a-0000056     fanotify_perm_event_info mnt_cache        vm_area_struct
:a-0000064     fasync_cache    mqueue_inode_cache xfrm_dst_cache
:a-0000072     fat_cache       names_cache      xfrm_state
:a-0000104     fat_inode_cache net_namespace

```

```

:a-0000128  file_lock_cache      nfs_commit_data
:a-0000144  file_lock_ctx        nfs_direct_cache

```

Slap Cache Properties

```

$ ls /sys/kernel/slub/TCP
aliases      destroy_by_rcu  objects_partial  red_zone          slabs_cpu_p
artial
align        free_calls      objs_per_slab    remote_node_defrag_ratio  store_user
alloc_calls  hwcachalign    order            sanity_checks       total_objec
ts
cpu_partial  min_partial     partial          shrink             trace
cpu_slabs    object_size     poison           slab_size           usersize
ctor         objects         reclaim_account  slabs              validate

```

- aliases
 - Number of merged alias caches
- align
 - The align value to use for the slab object.
 - s->align
- alloc_calls
 - Use alloc user (owner) tracking to output the allocation history of the slab cache.
 - e.g. 1 0xffff000008b8a068 age=2609 pid=3481
- cache_dma
 - It shows whether it is a slap cache using DMA zones. (1=DMA Zone Use Slap Cache, 0=Normal Zone Use Slap Cache)
 - SLAB_CACHE_DMA Flag Usage
- cpu_partial
 - Maximum number of slab objects to be managed on the per-CPU
 - Depending on the size (S->size), the default value is 2, 6, 13, or 30.
 - If debug is enabled, per-cpu management is not enabled and 0 is specified.
- cpu_slabs
 - It sums up the number of slab pages that are being managed for the per-CPU. (c->page + c->partial page count)
 - For each node, N[nid]=<per-cpu number of slab pages> is additionally noted.
 - e.g. 21 N0=21
- ctor
 - Shows the name of the constructor function of the slab cache with the constructor.
 - e.g. init_once+0x0/0x78
- destroy_by_rcu
 - Shows whether or not you are using the slap object deletion technique using RCU.
 - Presence or absence of RCU method for quick deletion through lock-less access (1=enabled, 0=unused)
- free_calls
 - e.g. 2 <not-available> age=4295439025 pid=0

- `hwcache_align`
 - Shows whether or not there is alignment in the L1 hardware cache line. (1=used, 0=unused)
 - `SLAB_HWCACHE_ALIGN` Flag Usage
- `min_partial`
 - Shows the minimum number of slab pages to keep in a partial list for each node.
 - The default (`S->min_partial`) value uses a value in the range of 5~10 proportional to the size, which is applied to each node.
 - `/proc/sys/kernel/`
- `object_size`
 - Shows the size of the slab object, excluding metadata.
 - `s->object_size`
- `objects`
 - Shows the total number of slab objects in use. (Note: Free objects managed by the Per-CPU are also counted as in use.)
 - For each node, `N[nid]=<number of slab objects>` is additionally noted.
 - e.g. 1288 N0=1288
- `objects_partial`
 - Shows the number of slab objects in use in a partial list for each node.
 - For each node, `N[nid]=<number of slab objects>` is additionally noted.
 - e.g. 2 N0=2
- `objs_per_slab`
 - Shows the number of objects to be used on the slab page.
 - The number of objects to be included on the order page recorded in `s->oo`.
- `order`
 - This is the order value that will be used for slab page assignment. (`s->oo`)
 - This value was calculated appropriately according to the size of the slab cache when it was created.
 - If you are allocating a slab page in a situation where you are out of memory, you may want to allocate the slab page with a minimum order (`s->min`) value instead of the order above.
- `partial`
 - It shows the sum of the number of slab pages that are managed in the partial list of nodes.
 - Summing `n->nr_partial`
 - For each node, `N[nid]=<per-cpu slab object number>` is additionally noted.
 - e.g. 1 N0=1
- `poison`
 - `poison` to show if debug is enabled. (1=used, 0=unused)
 - `SLAB_POISON` Flag Enabled
 - `"slab_debug=FP,<SlapCache Name>"`
- `reclaim_account`
 - Shows whether it is a reclaimable slab cache. (1=reclaimable cache, 0=regular unreclaimable cache)
 - When you create slab caches that support shrinker, use the `SLAB_RECLAIM_ACCOUNT` flag to create a slab cache.

- red_zone
 - red-zone shows whether debug is enabled or not. (1=used, 0=unused)
 - SLAB_RED_ZONE Flag Usage
 - "slab_debug=FZ,<SlapCache Name>"
- remote_node_defrag_ratio
 - If the local node is running out of slab pages, the remote node will allow memory by the percentage specified by this value.
 - The default value is 100 and 0~100 is allowed, and 0 prevents the remote node's slab page from being used.
- sanity_checks
 - sanity check shows whether debug is enabled or not. (1=used, 0=unused)
 - SLAB_CONSISTENCY_CHECKS Flag Enabled
 - "slab_debug=F,<SlapCache Name>"
- shrink
 - Performs memory reclamation of the reclaimable slab cache.
 - e.g. echo 1 > /sys/kernel/slab/ext4_inode_cache/shrink
- slab_size
 - Shows the size of the slab object including the metadata.
 - s->size
- slabs
 - Shows the total number of slap pages.
 - For each node, N[nid]=<number of slap pages> is additionally noted.
 - e.g. 28 N0=28
- slabs_cpu_partial
 - The per-cpu partial list shows the number of free slab objects and slab pages that are being managed. (Excludes s->page)
 - For each cpu, C[cpu] = <cpu partial free slab object > (<cpu partial slab page >).
 - e.g. 28(28) C0=6(6) C1=3(3) C2=18(18) C3=1(1)
- store_user
 - Shows whether user tracking debug is enabled. (1=used, 0=unused)
 - SLAB_STORE_USER Flag Usage
 - "slab_debug=FU,<SlapCache Name>"
- total_objects
 - Shows the total number of slab objects.
 - For each node, N[nid]=<number of slab objects> is additionally noted.
 - e.g. 1288 N0=1288
- trace
 - Shows whether trace debug is enabled or not. (1=used, 0=unused)
 - SLAB_TRACE Flag Enabled
 - "slab_debug=T,<SlapCache Name>"
- usersize
 - Shows the user size to use for copy to/from user.
 - s->usersize
- validate

- Perform validation of the slab cache. (Force debug check)
- e.g. `echo 1 > /sys/kernel/slab/anon_vma/validate`

slabinfo utility

Build debugging tools

```
$ gcc -o slabinfo tools/vm/slabinfo.c
```

usage

```
$ sudo ./slabinfo -h
slabinfo 4/15/2011. (c) 2007 sgi/(c) 2011 Linux Foundation.
```

```
slabinfo [-ahnpvtsz] [-d debugopts] [slab-regexp]
-a|--aliases          Show aliases
-A|--activity          Most active slabs first
-d<options>|--debug=<options> Set/Clear Debug options
-D|--display-active    Switch line format to activity
-e|--empty             Show empty slabs
-f|--first-alias       Show first alias
-h|--help              Show usage information
-i|--inverted          Inverted list
-l|--slabs             Show slabs
-n|--numa              Show NUMA information
-o|--ops               Show kmem_cache_ops
-s|--shrink            Shrink slabs
-r|--report            Detailed report on single slabs
-S|--Size              Sort by size
-t|--tracking          Show alloc/free information
-T|--Totals            Show summary information
-v|--validate          Validate slabs
-z|--zero              Include empty slabs
-1|--1ref              Single reference
```

Valid debug options (FZPUT may be combined)

```
a / A          Switch on all debug options (=FZUP)
-              Switch off all debug options
f / F          Sanity Checks (SLAB_DEBUG_FREE)
z / Z          Redzoning
p / P          Poisoning
u / U          Tracking
t / T          Tracing
```

Slap Cache List

```
$ sudo ./slabinfo
```

Name	Objects	Objsize	Space	Slabs/Part/Cpu	O/S	0	%Fr	%Ef	Flg
:at-0000016	256	16	4.0K	0/0/1	256	0	0	100	*a
:at-0000032	3968	32	126.9K	22/0/9	128	0	0	100	*a
:at-0000040	408	40	16.3K	0/0/4	102	0	0	99	*a
:at-0000064	32128	64	2.0M	454/0/48	64	0	0	100	*a
:at-0000104	156	104	16.3K	0/0/4	39	0	0	99	*a
:t-0000024	680	24	16.3K	0/0/4	170	0	0	99	*
:t-0000032	9472	32	303.1K	12/0/62	128	0	0	100	*
:t-0000040	612	40	24.5K	0/0/6	102	0	0	99	*
:t-0000064	12483	64	802.8K	128/1/68	64	0	0	99	*
:t-0000088	2714	88	241.6K	15/0/44	46	0	0	98	*
:t-0000096	168	96	16.3K	0/0/4	42	0	0	98	*
:t-0000104	6552	104	688.1K	158/0/10	39	0	0	99	*
:t-0000128	2240	128	286.7K	15/0/55	32	0	0	100	*
:t-0000192	4305	192	839.6K	152/0/53	21	0	0	98	*
:t-0000256	192	256	49.1K	3/0/9	16	0	0	100	*
:t-0000320	954	320	335.8K	12/5/29	25	1	12	90	*A
:t-0000384	84	384	32.7K	0/0/4	21	1	0	98	*A
:t-0000512	720	512	368.6K	26/0/19	16	1	0	100	*
:t-0000640	50	640	32.7K	1/0/1	25	2	0	97	*A
:t-0000960	187	936	180.2K	1/0/10	17	2	0	97	*A
:t-0001024	176	1024	180.2K	4/0/7	16	2	0	100	*
:t-0002048	176	2048	360.4K	2/0/9	16	3	0	100	*
:t-0004032	153	4032	655.3K	5/2/15	8	3	10	94	*
:t-0004096	64	4096	262.1K	0/0/8	8	3	0	100	*
anon_vma	2124	104	241.6K	10/0/49	36	0	0	91	
bdev_cache	72	848	65.5K	0/0/4	18	2	0	93	Aa
biovec-128	84	1536	131.0K	0/0/4	21	3	0	98	A
biovec-256	10	3072	32.7K	0/0/1	10	3	0	93	A
biovec-64	84	768	65.5K	0/0/4	21	2	0	98	A
blkdev_queue	34	1824	65.5K	0/0/2	17	3	0	94	
blkdev_requests	204	232	49.1K	0/0/12	17	0	0	96	
dentry	20500	200	4.1M	1012/0/13	20	0	0	97	a
ext4_groupinfo_4k	253	172	45.0K	10/0/1	23	0	0	96	a
ext4_inode_cache	10686	1232	13.4M	400/0/11	26	3	0	97	a
fat_cache	170	20	4.0K	0/0/1	170	0	0	83	a
fat_inode_cache	60	776	49.1K	1/0/2	20	2	0	94	a
file_lock_cache	100	160	16.3K	0/0/4	25	0	0	97	
fscache_cookie_jar	32	124	4.0K	0/0/1	32	0	0	96	
ftrace_event_file	595	48	28.6K	6/0/1	85	0	0	99	
idr_layer_cache	270	1068	294.9K	5/0/4	30	3	0	97	
inode_cache	5589	584	3.3M	191/0/16	27	2	0	96	a
jbd2_journal_handle	292	56	16.3K	0/0/4	73	0	0	99	a
jbd2_transaction_s	189	176	36.8K	0/0/9	21	0	0	90	Aa
kmalloc-8192	24	8192	196.6K	1/0/5	4	3	0	100	
kmem_cache	128	116	16.3K	1/0/3	32	0	0	90	A
kmem_cache_node	128	68	16.3K	1/0/3	32	0	0	53	A
mm_struct	112	536	65.5K	0/0/4	28	2	0	91	A
mqueue_inode_cache	18	840	16.3K	0/0/1	18	2	0	92	A
nfs_commit_data	18	448	8.1K	0/0/1	18	1	0	98	A
posix_timers_cache	18	216	4.0K	0/0/1	18	0	0	94	
proc_inode_cache	546	616	344.0K	4/0/17	26	2	0	97	a
radix_tree_node	2106	304	663.5K	71/0/10	26	1	0	96	a

shmem_inode_cache	644	696	458.7K	20/0/8	23 2	0	97
sighand_cache	184	1372	262.1K	0/0/8	23 3	0	96 A
sigqueue	112	144	16.3K	0/0/4	28 0	0	98
sock_inode_cache	100	616	65.5K	0/0/4	25 2	0	93 Aa
taskstats	24	328	8.1K	0/0/1	24 1	0	96
TCP	68	1816	131.0K	0/0/4	17 3	0	94 A
UDP	64	960	65.5K	0/0/4	16 2	0	93 A

- Name
 - Slap People
- Objects
 - Number of in-use objects
- Objsize
 - Object size excluding metadata (S->obj_size)
- Space
 - The byte size used in the entire slab page (the values used for the units are 1024 units, not 1000 units)
- Slabs/Part/Cpu
 - Slabs
 - Total number of slap pages – CPU
 - = full number of slap pages + part
 - Part
 - Number of slab pages managed by partial lists per node
 - Cpu
 - Number of slab pages managed by slab pages per CPU (c->page + c->partial pages)
- O/S
 - Number of objects per slab page
 - s->objs_per_slab
- O
 - order value
- %Fr
 - Slappage percentage of a node partial list
- %Ef
 - Percentage of objects in use
- Flg
 - The flag values are as follows:
 - *- alias cache
 - d – dma
 - A – Align the L1 hardware cache (hwcachalign)
 - p – poison
 - a – Reclaimable Slab Cache
 - Z – red-zone
 - F – sanity check
 - U – Owner Tracking
 - T – Trace

Loss Sorting Order (-L)

\$ sudo ./slabinfo -L

Name	Objects	Objsize	Loss	Slabs/Part/Cpu	O/S	0	%Fr	%E
f Flg								
task_struct	92	3456	206.3K	11/9/5	9	3	56	6
0								
kmalloc-512	624	512	147.4K	52/31/5	16	1	54	6
8								
:A-0000192	2317	192	140.8K	112/58/31	21	0	40	7
5 *A								
kernfs_node_cache	13230	128	112.8K	432/0/9	30	0	0	9
3								
proc_inode_cache	1114	648	97.3K	96/18/4	12	1	18	8
8 a								
dentry	7417	192	70.9K	341/56/24	21	0	15	9
5 a								

Utilization Indication (-D)

```
$ sudo ./slabinfo -D
```

Name	Objects	Alloc	Free	%Fast	Fallb	0	CmpX	UL
:0000024	170	0	0	0	0	0	0	0
:0000040	102	0	0	0	0	0	0	0
:0000048	85	0	0	0	0	0	0	0
:0000056	73	0	0	0	0	0	0	0
:0000064	64	0	0	0	0	0	0	0
:0000080	51	0	0	0	0	0	0	0
:0000128	32	0	0	0	0	0	0	0
:0000192	21	0	0	0	0	0	0	0
:0000256	192	0	0	0	0	0	0	0
:0000448	36	0	0	0	0	1	0	0
:0000896	36	0	0	0	0	2	0	0
:0001024	16	0	0	0	0	2	0	0
:0002048	16	0	0	0	0	3	0	0
:0004096	40	0	0	0	0	3	0	0
:a-0000032	128	0	0	0	0	0	0	0
:a-0000048	85	0	0	0	0	0	0	0
:a-0000056	73	0	0	0	0	0	0	0
:A-0000064	1340	0	0	0	0	0	0	0
:A-0000072	168	0	0	0	0	0	0	0
:A-0000080	255	0	0	0	0	0	0	0
:a-0000104	1482	0	0	0	0	0	0	0
:A-0000128	320	0	0	0	0	0	0	0
:A-0000192	2317	0	0	0	0	0	0	0
:a-0000256	16	0	0	0	0	0	0	0
:A-0001088	130	0	0	0	0	2	0	0
anon_vma	654	0	0	0	0	0	0	0
bdev_cache	19	0	0	0	0	2	0	0
blkdev_ioc	39	0	0	0	0	0	0	0
configfs_dir_cache	46	0	0	0	0	0	0	0
...								

alias slab cache list (-a)

```
$ sudo ./slabinfo -a

:at-0000016  <- discard_entry f2fs_inode_entry jbd2_revoke_table_s inmem_page_entry
free_nid f2fs_ino_entry sit_entry_set
:at-0000024  <- nat_entry nat_entry_set
:at-0000032  <- ext4_extent_status jbd2_revoke_record_s
:at-0000040  <- ext4_free_data ext4_io_end
:at-0000064  <- mmcbk0p6 jbd2_journal_head buffer_head
:at-0000104  <- ext4_allocation_context ext4_prealloc_space
:t-0000024  <- ip_fib_alias dnotify_struct jbd2_inode nsproxy scsi_data_buffer
:t-0000032  <- ftrace_event_field fanotify_event_info dmaengine-unmap-2 secpath_ca
che anon_vma_chain sd_ext_cdb ip_fib_trie tcp_bind_bucket ext4_system_zone
:t-0000040  <- eventpoll_pwq page->ptl
:t-0000064  <- nfs_page pid kmalloc-64 kiocb uid_cache fasync_cache file_lock_ctx
cfq_io_cq
:t-0000088  <- flow_cache vm_area_struct
:t-0000096  <- dnotify_mark fsnotify_mark inotify_inode_mark
:t-0000104  <- task_delay_info kernfs_node_cache
:t-0000128  <- ip_mrt_cache blkdev_ioc sgpool-8 pid_namespace fs_cache inet_peer_c
ache kmalloc-128 ip_dst_cache eventpoll_epi cred_jar
:t-0000192  <- bio-0 key_jar skbuff_head_cache ip4-frags request_sock_TCP rpc_task
s kmalloc-192 biovec-16
:t-0000256  <- mnt_cache sgpool-16 pool_workqueue kmalloc-256 files_cache
:t-0000320  <- xfrm_dst_cache filp
:t-0000384  <- skbuff_fclone_cache dio
:t-0000512  <- sgpool-32 kmalloc-512
:t-0000640  <- nfs_write_data kioclx nfs_read_data
:t-0000960  <- RAW PING signal_cache
:t-0001024  <- UNIX kmalloc-1024 sgpool-64
:t-0002048  <- kmalloc-2048 sgpool-128 rpc_buffers
:t-0004032  <- task_struct net_namespace
:t-0004096  <- names_cache kmalloc-4096
```

Slap cache summary (-T)

```
$ sudo ./slabinfo -T
Slabcache Totals
-----
Slabcaches : 71      Aliases : 118->49  Active: 59
Memory used: 32.5M   # Loss  : 741.4K   MRatio: 2%
# Objects  : 125.2K  # PartObj: 9      ORatio: 0%

Per Cache      Average      Min      Max      Total
-----
#Objects       2.1K        10      32.1K    125.2K
#Slabs         58         1       1.0K     3.4K
#PartSlab      0           0        2         2
%PartSlab      0%          0%       10%       0%
PartObjs       0           0         9         9
% PartObj      0%          0%        5%       0%
Memory        550.9K      4.0K    13.4M    32.5M
Used          538.3K      3.4K    13.1M    31.7M
Loss          12.5K        0      302.4K   741.4K

Per Object     Average      Min      Max
-----
Memory        255         16      8.1K
User          253         16      8.1K
Loss           1           0       64
```

Slap Cache Details (-r)

```
$ slabinfo -r jake

Slabcache: jake           Aliases: 0 Order : 1 Objects: 2

Sizes (bytes)    Slabs           Debug           Memory
-----
Object :      30  Total :      1  Sanity Checks : On  Total:    8192
SlabObj:     392  Full  :      0  Redzoning      : On  Used :      60
SlabSiz:    8192  Partial:    1  Poisoning      : On  Loss :    8132
Loss  :     362  CpuSlab:    0  Tracking       : On  Lalign:    724
Align  :      56  Objects:   20  Tracing        : On  Lpadd:    352

jake has no kmem_cache operations

jake: Kernel object allocation
-----
    1 0xfffff000008b8a068 age=1174850 pid=3481
    1 0xfffff000008b8a078 age=1174837 pid=3481

jake: Kernel object freeing
-----
    2 <not-available> age=4296123146 pid=0

jake: No NUMA information available.
```

- Object

- Object size excluding metadata
- SlabObj
 - Object size including metadata
- SlabSiz
 - 1 srep page size
- Loss
 - Unusable space size on 1 slap page (remains)
- Align
 - align unit
- Total
 - Total number of slap pages
- Full
 - The number of slab pages that all objects are all in use
- Partial
 - Number of slap pages per node
- CpuSlab
 - Per-CPU Slap Page Count
- Objects
 - Number of objects per 1 slab page
- Sanity Checks
 - Sanity Check Debug Enabled?
- Redzoning
 - Enable Red-zone Debug
- Poisoning
 - Poison debug enabled?
- Tracking
 - Enable user tracking debug?
- Tracing
 - Trace Debug?
- Total
 - The memory size being used for the entire srep page
- Used
 - The size of memory occupied by the in-use slab object, excluding metadata.
- Loss
 - Toal – Used
- Lalign
 - (object size including metadata – object size excluding metadata) * Number of objects in use
 - (SlabObj – Object) * Number of objects in use
- Lpadd
 - The sum of all the remaining areas of the slab page.

slabtop utility

```
$ slabtop
Active / Total Objects (% used)      : 514911 / 563284 (91.4%)
Active / Total Slabs (% used)        : 30238 / 30238 (100.0%)
Active / Total Caches (% used)       : 89 / 121 (73.6%)
Active / Total Size (% used)         : 198611.59K / 205849.01K (96.5%)
Minimum / Average / Maximum Object  : 0.02K / 0.37K / 12.00K

  OBJS ACTIVE  USE OBJ SIZE  SLABS OBJ/SLAB  CACHE SIZE NAME
114156 110759   0%   0.19K   5436     21    21744K dentry
 86895  84904   0%   1.04K  13611     30    435552K ext4_inode_cache
 56589  45405   0%   0.10K   1451     39     5804K buffer_head
 46512  19299   0%   0.04K    456    102     1824K ext4_extent_status
 44832  43854   0%   0.12K   1401     32     5604K kmalloc-128
 41300  39940   0%   0.57K   1475     28    23600K radix_tree_node
 25664  25664 100%   0.06K    401     64     1604K anon_vma_chain
 23070  22810   0%   0.13K    769     30     3076K kernfs_node_cache
 18774  18489   0%   0.57K    783     28    12528K inode_cache
 14344  14234   0%   0.18K    652     22     2608K vm_area_struct
 13110  13110 100%   0.09K    285     46     1140K anon_vma
  8816   8696   0%   0.25K    551     16     2204K filp
  4752   4057   0%   0.64K    198     24     3168K proc_inode_cache
  4386   4386 100%   0.04K     43    102      172K pde_opener
  3825   3825 100%   0.05K     45     85      180K ftrace_event_field
  3616   3616 100%   0.12K    113     32      452K pid
  3549   3549 100%   0.19K    169     21      676K cred_jar
  3125   3125 100%   0.62K    125     25    20000K squashfs_inode_cache
  2896   2877   0%   0.50K    181     16    1448K kmalloc-512
  2496   2496 100%   0.06K     39     64      156K dmaengine-unmap-2
  2408   2408 100%   0.14K     86     28      344K ext4_groupinfo_4k
  2304   2292   0%   1.00K    144     16    2304K kmalloc-1024
  2192   2140   0%   0.25K    137     16      548K kmalloc-256
  2075   2075 100%   0.62K     83     25    1328K sock_inode_cache
...

```

consultation

- Slab Memory Allocator -1- (Structure) (<http://jake.dothome.co.kr/slub/>) | Qc
- Slab Memory Allocator -2- (Initialize Cache) (http://jake.dothome.co.kr/kmem_cache_init) | Qc
- Slub Memory Allocator -3- (Create Cache) (<http://jake.dothome.co.kr/slub-cache-create>) | Qc
- Slub Memory Allocator -4- (Calculate Order) (<http://jake.dothome.co.kr/slub-order>) | Qc
- Slub Memory Allocator -5- | (<http://jake.dothome.co.kr/slub-slub-alloc>) Qc
- Slub Memory Allocator -6- (Assign Object) (<http://jake.dothome.co.kr/slub-object-alloc>) | Qc
- Slub Memory Allocator -7- (Object Unlocked) (<http://jake.dothome.co.kr/slub-object-free>) | Qc
- Slub Memory Allocator -8- (Drain/Flash Cache) (<http://jake.dothome.co.kr/slub-drain-flush-cache>) | Qc
- Slub Memory Allocator -9- (Cache Shrink) (<http://jake.dothome.co.kr/slub-cache-shrink>) | Qc
- Slub Memory Allocator -10- | (<http://jake.dothome.co.kr/slub-slub-free>) Qc
- Slub Memory Allocator -11- (Clear Cache (<http://jake.dothome.co.kr/slub-cache-destroy>)) | Qc

- Slub Memory Allocator -12- (Debugging Slub) (<http://jake.dothome.co.kr/slub-debug>) | Qc
- Slub Memory Allocator -13- (slabinfo) (<http://jake.dothome.co.kr/slub-slabinfo>) | Sentence C – Current post

LEAVE A COMMENT

Your email will not be published. Required fields are marked with *

Comments

name *

email *

Website

WRITE A COMMENT

◀ Per-cpu -4- (atomic operations) (<http://jake.dothome.co.kr/per-cpu-atomic/>)

Slub Memory Allocator -4- (order calculation) ▶ (<http://jake.dothome.co.kr/slub-order/>)

Munc Blog (2015 ~ 2024)