

# Linuxcgroup

## Linux

BIOSKernelMemTotal free RAMLinux

- free
- /proc/meminfo

Linux

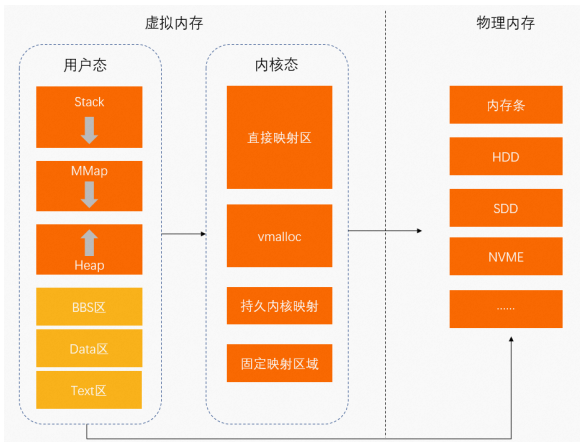
```
# = + +
total = used + free + buff/cache
```

Kernel

- PageCache

### Content Menu

- Linux
  - - PageCache
  - - - - top
        - ps
        - smem
      - cgroup
  - DockerK8s
    - docker stat
    - kubectl top pod
  -




- - - Stack
    - MMap(Memory Mapping Segment)
    - Heap
    - BBS
    - Data
    - Text
  - MMap
    - VMALLOC

13A2B4AB



top	VIRT(Virtual Set Size)		
	RES(Resident Set Size)	RSS	anno_rss + file_rss + shmem_rss
	SHR(Shared Memory)		file_rss + shmem_rss
	%MEM		RES / MemTotal


ps

 KB

```
[root@ecs-10-0-15-12 ~]# ps aux | more
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  251448 16808 ?        Ss   3月10 0:22 /usr/lib/systemd/systemd --switched-root --system --deserialize 17
root         2  0.0  0.0      0     0 ?        Ss   3月10 0:00 kthreadd/
root         3  0.0  0.0      0     0 ?        Ss   3月10 0:00 rcu_gp/
root         4  0.0  0.0      0     0 ?        Ss   3月10 0:00 rcu_par_gp/
root         6  0.0  0.0      0     0 ?        Ss   3月10 0:00 worker/0-ns-4blockd/
root         8  0.0  0.0      0     0 ?        Ss   3月10 0:00 ksm_perio_wq/
root         9  0.0  0.0      0     0 ?        Ss   3月10 14:05 ksoftirqd/0/
root        10  0.0  0.0      0     0 ?        Ss   3月10 12:10 rcu_sched/
```

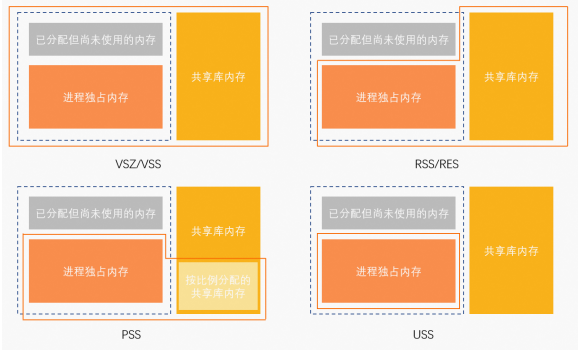
ps	VSZ(Virtual Set Size)		
	RSS(Resident Set Size)	RSS	anno_rss + file_rss + shmem_rss
	%MEM		RSS / MemTotal

smem



```
[root@ecs-10-0-15-12 ~]# smem -k
PID User      Command      Swap    USS    PSS    RSS
74147 root      /pause      0      40.0K  51.0K  684.0K
93523 root      /pause      0      40.0K  51.0K  684.0K
93595 root      /pause      0      40.0K  51.0K  684.0K
100294 root      /pause      0      40.0K  51.0K  684.0K
103999 root      /pause      0      40.0K  51.0K  684.0K
104634 root      /pause      0      40.0K  51.0K  684.0K
105066 root      /pause      0      40.0K  51.0K  684.0K
105180 root      /pause      0      40.0K  51.0K  684.0K
105468 root      /pause      0      40.0K  51.0K  684.0K
107013 root      /pause      0      40.0K  51.0K  684.0K
110834 root      /pause      0      40.0K  51.0K  684.0K
111448 root      /pause      0      40.0K  51.0K  684.0K
115291 root      /pause      0      40.0K  51.0K  684.0K
```

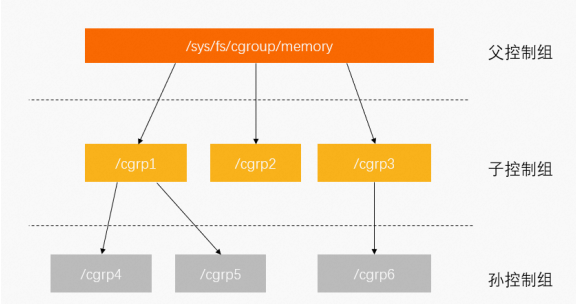
smem	USS(Unique Set Size)		anno_rss
	PSS(Proportional Set Size)		anno_rss + file_rss/m + shmem_rss/n
	RSS(Resident Set Size)	RSS	anno_rss + file_rss + shmem_rss





## cgroup

cgroupLinux  
cgroupcgroupMemory Control Group(memcg)



memory cgroup

```
cgroup.event_control      # eventfd
memory.usage_in_bytes     #
memory.limit_in_bytes     # /
memory.failcnt            #
memory.max_usage_in_bytes #
memory.soft_limit_in_bytes # /
memory.stat               # cgroup
memory.use_hierarchy      # /cgroupcgroup
memory.force_empty        # cgroup
memory.pressure_level     # cgroup.event_control
memory.swappiness         # swappiness
memory.move_charge_at_immigrate # cgroup
memory.oom_control        # /oom controls
memory.numa_stat          # numa
```

3

- memory.limit\_in\_bytescgroupk8sdockermemory limits
- memory.usage\_in\_bytescgroupmemory.statRSS+Cache
- memory.statcgroup

memory.stat	
cache	PageCache
rss	cgroupanno_rss
mapped_file	cgroupfile_rssshmem_rss
active_anon	LRUleast-recently-usedAnonymousSwap tmpfsshmembytes
inactive_anon	LRUAnonymousSwap tmpfsshmembytes
active_file	LRUFile-backedbytes
inactive_file	LRUFile-backedbytes
unevictable	bytes

total\_cgroupcgroup total\_rss cgroupcgroupRSS

cgroup

- `cgroupRSSanno_rssUSScgroupmapped_file+RSSRSS`
- `PageCachecgroup memcg PageCache`

		cgroup(memcg)
RSS	anon_rss + file_rss shmem_rss	anon_rss
mapped_file		file_rss + shmem_rss
cache		PageCache

## DockerK8s

DockerK8SLinux memcg

### docker stat

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIPS
87c1d1871d2d	amazing_shtern	0.37%	475.7MiB / 1.93GiB	23.96%	2.20MB / 860kB	4.99MB / 77MB	38

- `LIMITmemory.limit_in_bytes`
- `MEM USAGEmemory.usage_in_bytes - memory.stat[total_cache]`



docker stat

### kubectl top pod

kubectl topMetric-serverHeapsterCadvisiorworking\_setPodPauseMetrics-serverPod

```
func decodeMemory(target *resource.Quantity, memStats *stats.MemoryStats)
error {
    if memStats == nil || memStats.WorkingSetBytes == nil {
        return fmt.Errorf("missing memory usage metric")
    }

    *target = *uint64Quantity(*memStats.WorkingSetBytes, 0)
    target.Format = resource.BinarySI

    return nil
}
```

Cadvisiorworkingset

```

func setMemoryStats(s *cgroups.Stats, ret *info.ContainerStats) {
    ret.Memory.Usage = s.MemoryStats.Usage.Usage
    ret.Memory.MaxUsage = s.MemoryStats.Usage.MaxUsage
    ret.Memory.Failcnt = s.MemoryStats.Usage.Failcnt

    if s.MemoryStats.UseHierarchy {
        ret.Memory.Cache = s.MemoryStats.Stats["total_cache"]
        ret.Memory.RSS = s.MemoryStats.Stats["total_rss"]
        ret.Memory.Swap = s.MemoryStats.Stats["total_swap"]
        ret.Memory.MappedFile = s.MemoryStats.Stats["total_mapped_file"]
    } else {
        ret.Memory.Cache = s.MemoryStats.Stats["cache"]
        ret.Memory.RSS = s.MemoryStats.Stats["rss"]
        ret.Memory.Swap = s.MemoryStats.Stats["swap"]
        ret.Memory.MappedFile = s.MemoryStats.Stats["mapped_file"]
    }
    if v, ok := s.MemoryStats.Stats["pgfault"]; ok {
        ret.Memory.ContainerData.Pgfault = v
        ret.Memory.HierarchicalData.Pgfault = v
    }
    if v, ok := s.MemoryStats.Stats["pgmajfault"]; ok {
        ret.Memory.ContainerData.Pgmajfault = v
        ret.Memory.HierarchicalData.Pgmajfault = v
    }

    workingSet := ret.Memory.Usage
    if v, ok := s.MemoryStats.Stats["total_inactive_file"]; ok {
        if workingSet < v {
            workingSet = 0
        } else {
            workingSet -= v
        }
    }
    ret.Memory.WorkingSet = workingSet
}

```

kubectl top podMemory Usage = Memory WorkingSet = memory.usage\_in\_bytes - memory.stat[total\_inactive\_file]

		Memory Usage
docker stat	Docker	memory.usage_in_bytes - memory.stat[total_cache]
kubectl top pod	K8s	memory.usage_in_bytes - memory.stat[total_inactive_file]

toppsMemory Usagetopps

Memcg	rss + cacheactive cache + inactive cache
Docker	rss
K8s	rss + active cache

- <https://www.alibabacloud.com/help/zh/arms/application-monitoring/memory-metrics>
- <http://hustcat.github.io/memory-usage-in-process-and-cgroup>
- <https://www.51cto.com/article/692936.html>
- <https://itnext.io/from-rss-to-wss-navigating-the-depths-of-kubernetes-memory-metrics-4d7d77d8fdcb>

