

拉勾教育

— 互联网人实战大学 —

《Java 性能优化实战 21 讲》

李国 前京东、陌陌高级架构师

— 拉勾教育出品 —

03 | 深入剖析：哪些资源，容易成为瓶颈？

深入剖析：哪些资源，容易成为瓶颈？

拉勾教育

— 互联网人实战大学 —

第 02 课

介绍解决性能问题常用的一些切入点

本课时

从计算机资源层面讲解，哪些系统组件容易出现性能瓶颈

以及如何判断该系统组件是否达到了瓶颈



深入剖析：哪些资源，容易成为瓶颈？

拉勾教育

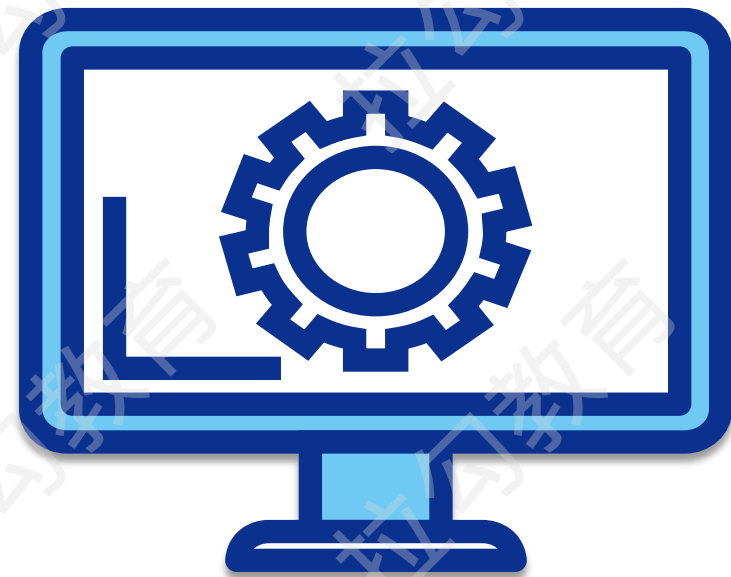
— 互联网人实战大学 —

计算机各个组件之间的速度很不均衡

当系统存在短板时，会对性能造成较大的负面影响

当 CPU 的负载特别高时，任务就会排队，不能及时执行

CPU、内存、I/O 这三个系统组件，又容易成为瓶颈



中央处理器 CPU

通过 top 命令，来观测 CPU 的性能

通过负载，评估 CPU 任务执行的排队情况

通过 vmstat，看 CPU 的繁忙程度

top 命令 — CPU 性能

```
top - 16:11:56 up 68 days, 6:16, 4 users, load average: 0.00, 0.01, 0.05
Tasks: 243 total, 1 running, 242 sleeping, 0 stopped, 0 zombie
%Cpu0  :  0.3 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1  :  0.3 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 7994076 total, 183720 free, 754500 used, 7055856 buff/cache
KiB Swap: 4063228 total, 4063228 free, 0 used. 6144012 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
98606	root	20	0	2696552	217896	51776	S	1.3	2.7	310:06.98	prometheus
10248	root	20	0	115868	16800	5992	S	0.7	0.2	68:15.49	blackbox_export
19696	root	20	0	1786988	41476	29088	S	0.3	0.5	82:28.54	promtail-linux-
1	root	20	0	125524	4148	2516	S	0.0	0.1	2:21.64	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.56	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:38.73	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	rt	0	0	0	0	S	0.0	0.0	0:01.53	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	7:57.85	rcu_sched
10	root	rt	0	0	0	0	S	0.0	0.0	0:18.88	watchdog/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:13.42	watchdog/1
12	root	rt	0	0	0	0	S	0.0	0.0	0:01.49	migration/1

top 命令 — CPU 性能

拉勾教育

— 互联网人实战大学 —

- **us** ——用户态所占用的 CPU 百分比，即引用程序所耗费的 CPU
- **sy** ——内核态所占用的 CPU 百分比，需要配合 vmstat 命令，查看上下文切换是否频繁
- **ni** ——高优先级应用所占用的 CPU 百分比
- **wa** ——等待 I/O 设备所占用的 CPU 百分比，过高输入输出设备可能存在明显的瓶颈

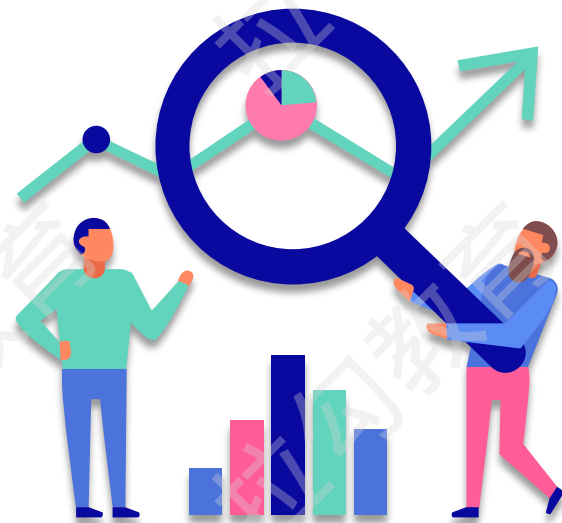


top 命令 — CPU 性能

拉勾教育

— 互联网人实战大学 —

- **hi** ——硬中断所占用的 CPU 百分比
- **si** ——软中断所占用的 CPU 百分比
- **st** ——测量的是虚拟机等待宿主机 CPU 的时间占比
- **id** ——空闲 CPU 百分比



负载 —— CPU 任务排队情况

拉勾教育

— 互联网人实战大学 —

负载 (load) ——评估 CPU 任务执行的排队情况

使用 **uptime 命令** 查看负载情况

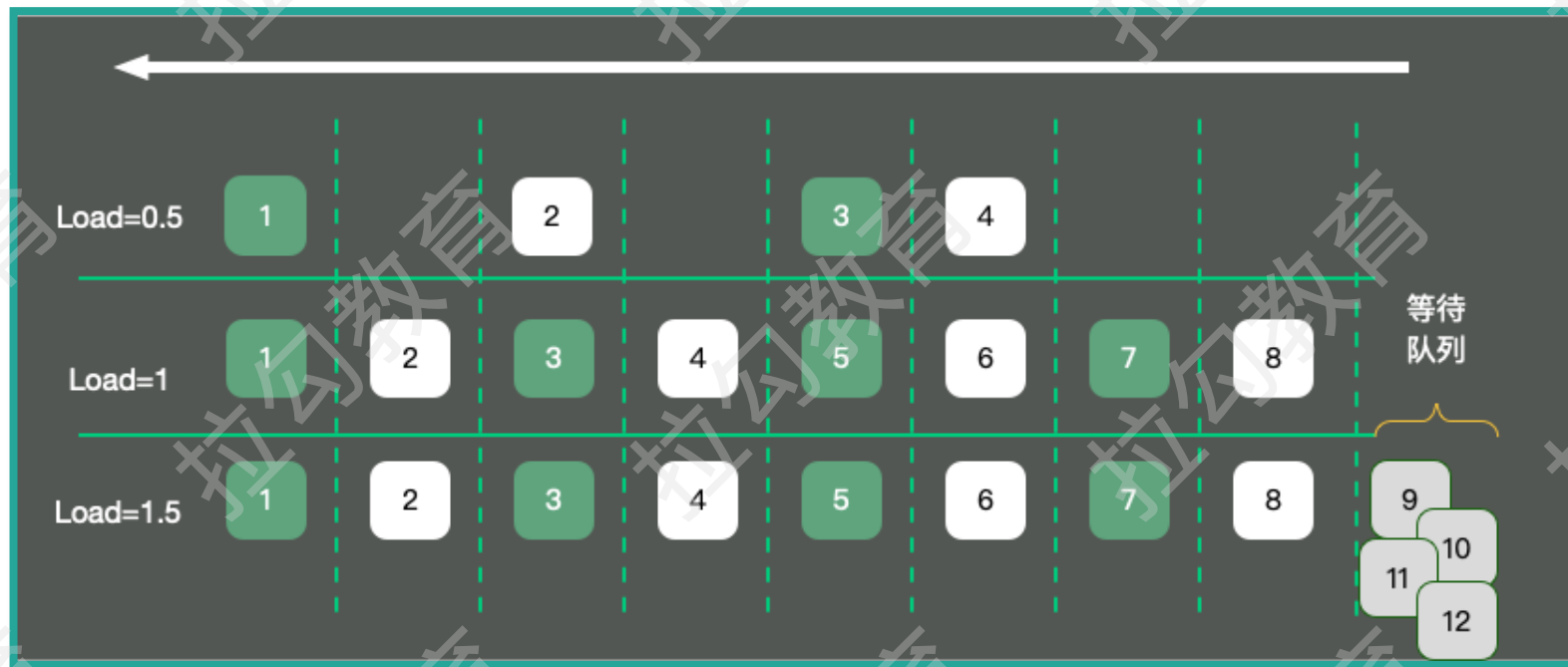
load 的效果分别显示了最近 1min、5min、15min 的数值



负载 —— CPU 任务排队情况

拉勾教育

— 互联网人实战大学 —

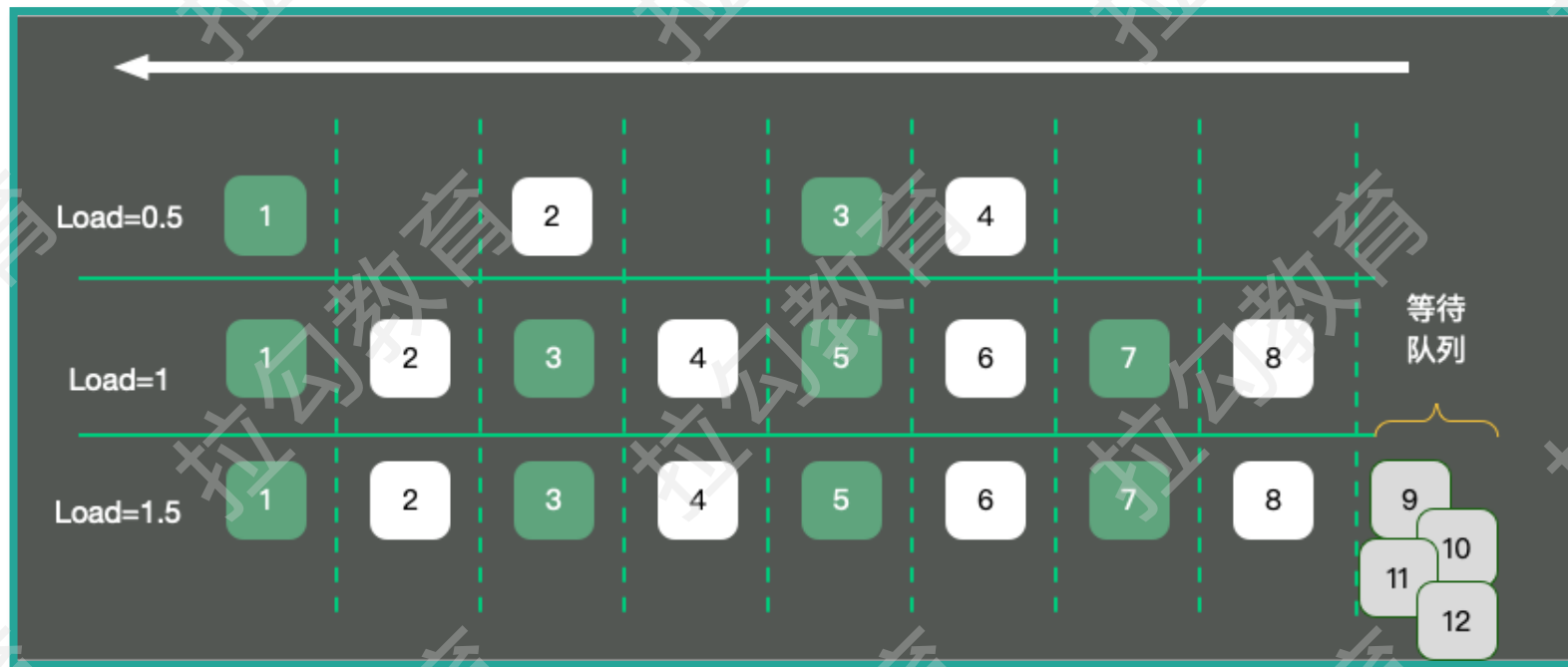


马路上的车只有 4 辆，车辆畅通无阻，load 大约是 0.5

负载 —— CPU 任务排队情况

拉勾教育

— 互联网人实战大学 —

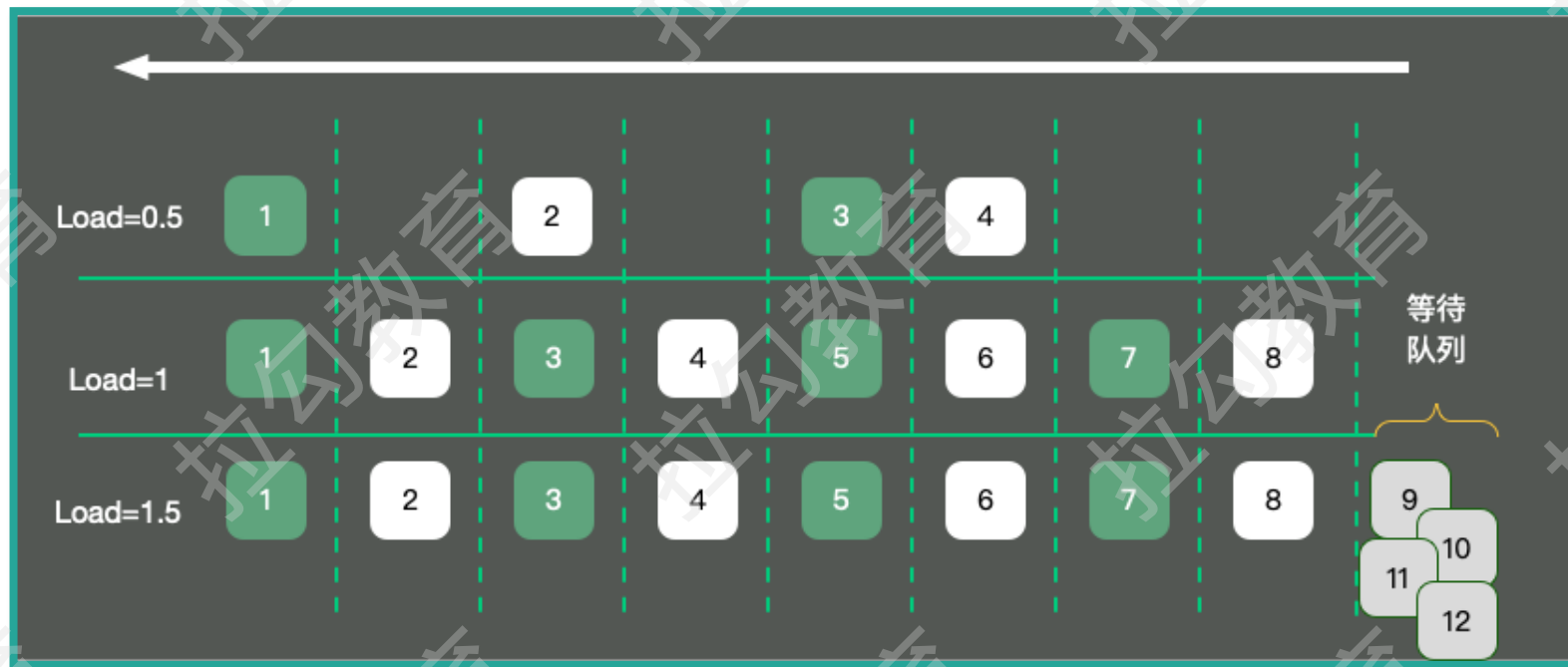


马路上的车有 8 辆，能首尾相接安全通过，此时 load 大约为 1

负载 —— CPU 任务排队情况

拉勾教育

— 互联网人实战大学 —



马路上的车有 12 辆，除了在马路上的 8 辆车，有 4 辆在马路外面排队，此时 load 大约为 1.5

load 为 1 代表的是什么 ?

在单核的硬件上，load 的值达到 1

认为系统负载已经到了极限



负载 —— CPU 任务排队情况

拉勾教育

— 互联网人实战大学 —

与 CPU 的个数有关

- **单核**的负载达到 1，总 load 的值约为 1
- **双核**的每核负载都达到 1，总 load 约为 2
- **四核**的每核负载都达到 1，总 load 约为 4



vmstat —— CPU 繁忙程度

拉勾教育

— 互联网人实战大学 —

```
1 0 0 131388 250844 979312 0 0 0 40 3317 2150 51 3 47 0 0
2 0 0 131636 250844 979312 0 0 0 0 3462 2270 53 3 45 0 0
3 0 0 131388 250844 979320 0 0 0 0 3275 1909 51 2 47 0 0
2 0 0 131512 250844 979324 0 0 0 0 4380 2894 74 5 22 0 0
4 0 0 131496 250848 979328 0 0 0 48 4244 3579 69 9 22 0 0
0 0 0 131388 250848 979328 0 0 0 8 2798 1687 53 2 46 0 0
3 0 0 131512 250848 979328 0 0 0 0 2693 1632 46 3 51 0 0
1 0 0 131636 250848 979336 0 0 0 0 2773 1628 63 2 36 0 0
3 0 0 131636 250848 979336 0 0 0 0 3447 2123 65 2 33 0 0
0 0 0 131636 250852 979336 0 0 0 0 2951 2350 41 3 57 0 0
^C
[xjj]vmstat 1
procs -----memory----- --swap-- --io-- --system-- --cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
2 0 0 130604 250864 979476 0 0 0 9 0 0 7 2 91 0 0
```

运行队列

阻塞队列

上下文

切换

和Top一样

vmstat —— CPU 繁忙程度

拉勾教育

— 互联网人实战大学 —

- **b** (Uninterruptible Sleep)——等待 I/O，可能是读盘或者写盘动作比较多
- **si/so**——显示了交换分区的一些使用情况
- **cs** ——每秒钟上下文切换(Context Switch)的数量

如果上下文切换过于频繁，需要考虑是否是进程或者线程数开的过多



vmstat —— CPU 繁忙程度

拉勾教育

— 互联网人实战大学 —

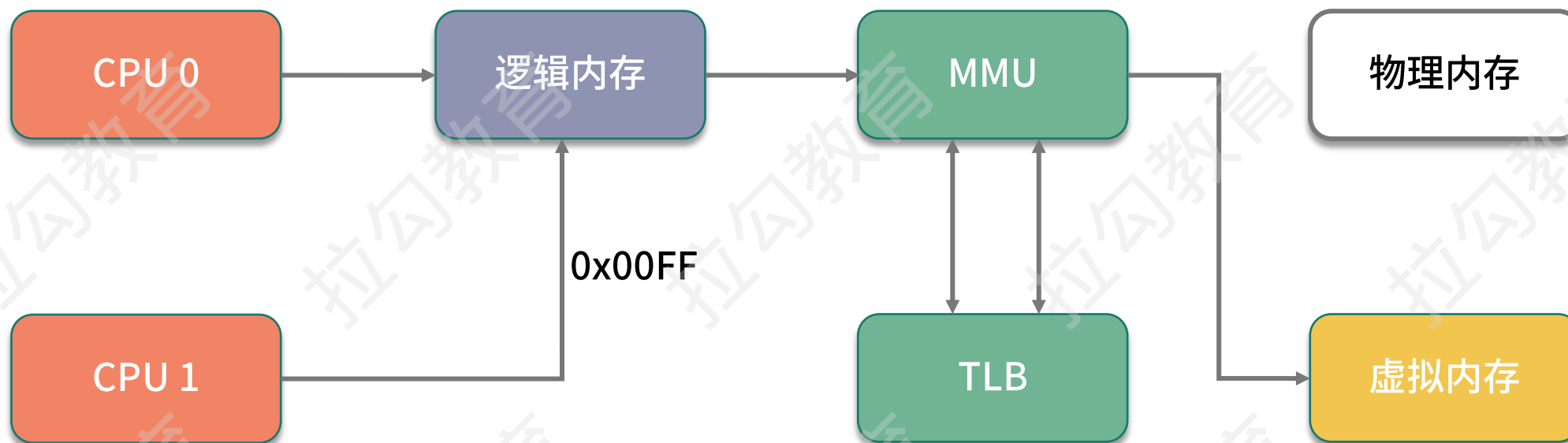
每个进程上下文切换的具体数量，通过查看内存映射文件获取

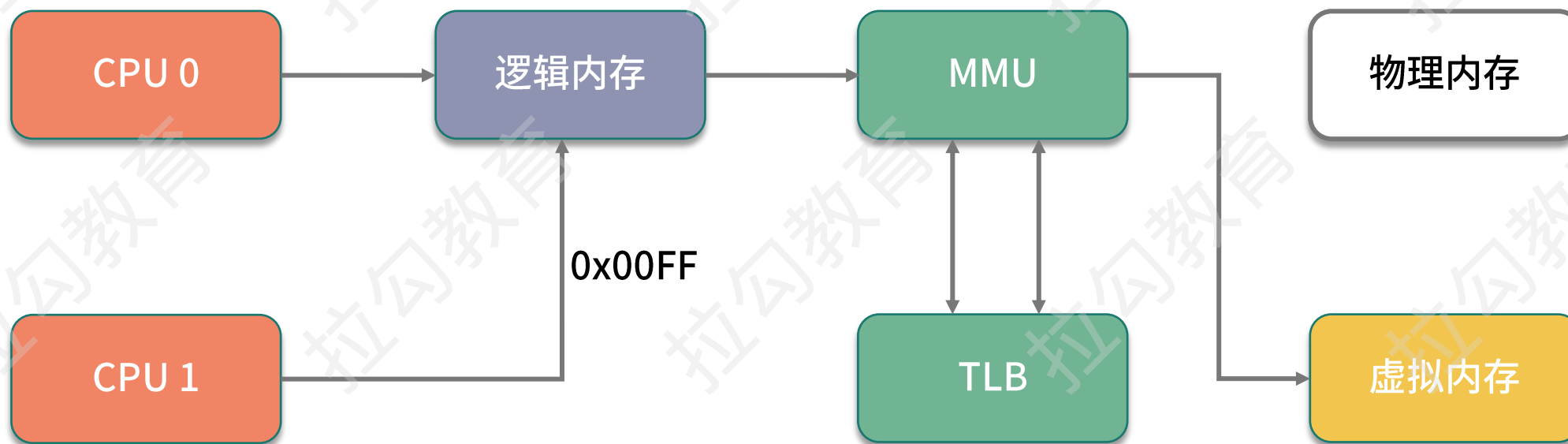
```
[root@localhost ~]# cat /proc/2788/status
```

```
...
```

```
voluntary_ctxt_switches: 93950
```

```
nonvoluntary_ctxt_switches: 171204
```





比如物理内存是 4GB，分配了 8GB 的 SWAP 分区，那么应用可用的总内存就是 12GB

top 命令

拉勾教育

— 互联网人实战大学 —

```
Tasks: 112 total, 1 running, 111 sleeping, 0 stopped, 0 zombie
Cpu(s): 62.9%us, 4.3%sy, 0.0%ni, 31.9%id, 0.0%wa, 0.0%hi, 0.8%si, 0.0%st
Mem: 4056412k total, 3929412k used, 127000k free, 261924k buffers
Swap: 0k total, 0k used, 0k free, 964092k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
27642	root	20	0	3714m	23g	7140	S	124.8	59.8	5736:02	java
20558	telegraf	20	0	475m	28m	3940	S	7.3	0.7	1554:41	telegraf
1452	root	20	0	2451m	70m	3192	S	2.3	1.8	1640:47	java
11047	dxhsu	20	0	15020	1336	1008	R	0.7	0.0	0:00.07	top
24924	root	20	0	129m	9908	5404	S	0.7	0.2	522:20.80	AliYunDun

VIRT —— 虚拟内存

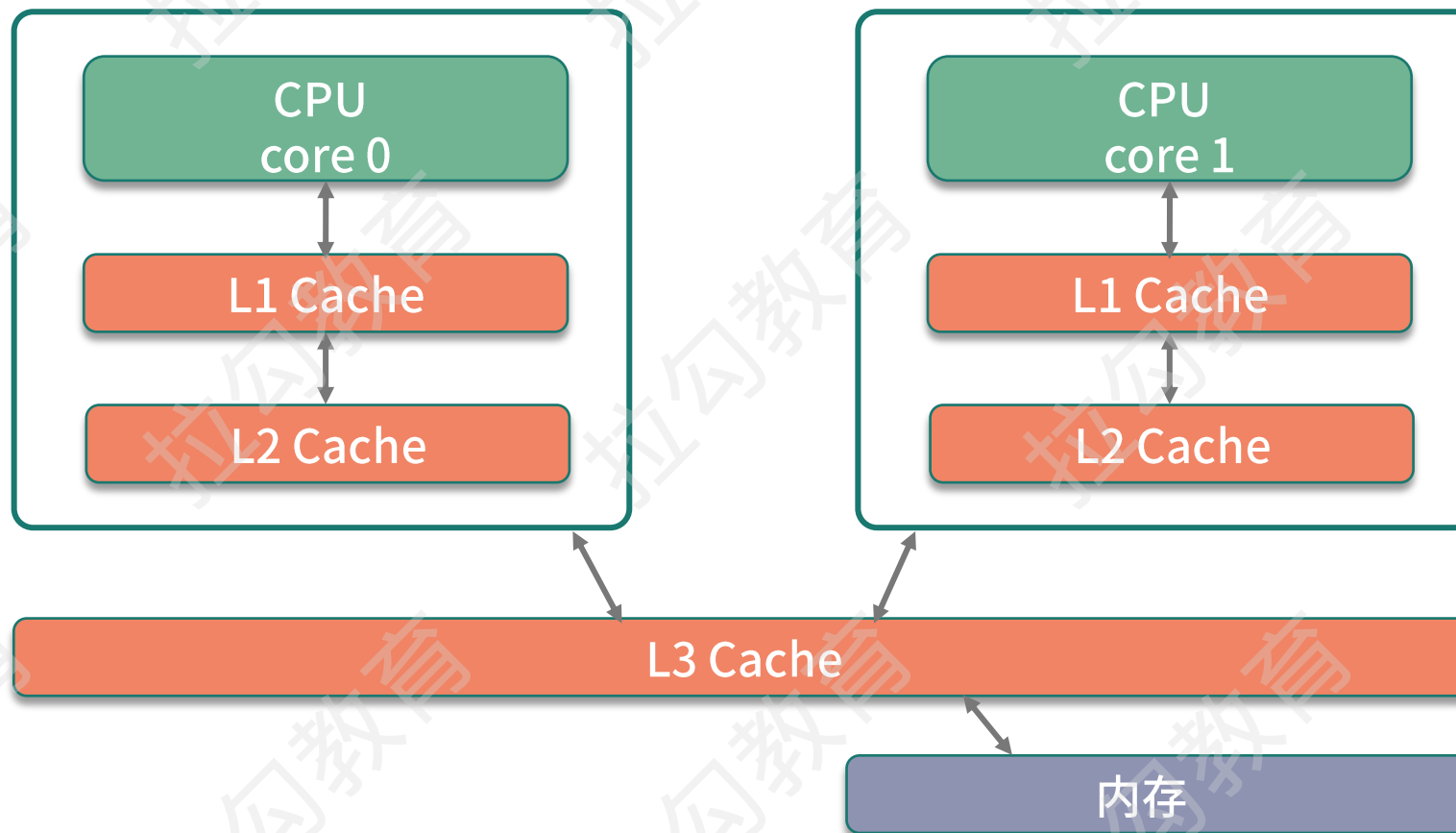
RES —— 进程实际占用的内存

SHR —— 共享内存，比如可以复用的一些 so 文件等

CPU 缓存

拉勾教育

— 互联网人实战大学 —



伪共享(False Sharing)——在这些高速缓存中，以缓存行为单位进行存储

当多线程修改一些变量的值时，如果这些变量都在同一个缓存行里，会造成频繁刷新



通过以下的命令查看CPU0的缓存行大小，这个值一般是 64

```
cat /sys/devices/system/cpu/cpu0/cache/index0/coherency_line_size  
cat /sys/devices/system/cpu/cpu0/cache/index1/coherency_line_size  
cat /sys/devices/system/cpu/cpu0/cache/index2/coherency_line_size  
cat /sys/devices/system/cpu/cpu0/cache/index3/coherency_line_size
```

通过 cpuinfo 得到的结果

```
# cat /proc/cpuinfo | grep cache  
cache size: 20480 KB  
cache_alignment: 64  
cache size: 20480 KB  
cache_alignment: 64  
cache size: 20480 KB  
cache_alignment: 64  
cache size: 20480 KB  
cache_alignment: 64
```


JDK8 以上的版本，通过开启参数 `-XX:-RestrictContended`

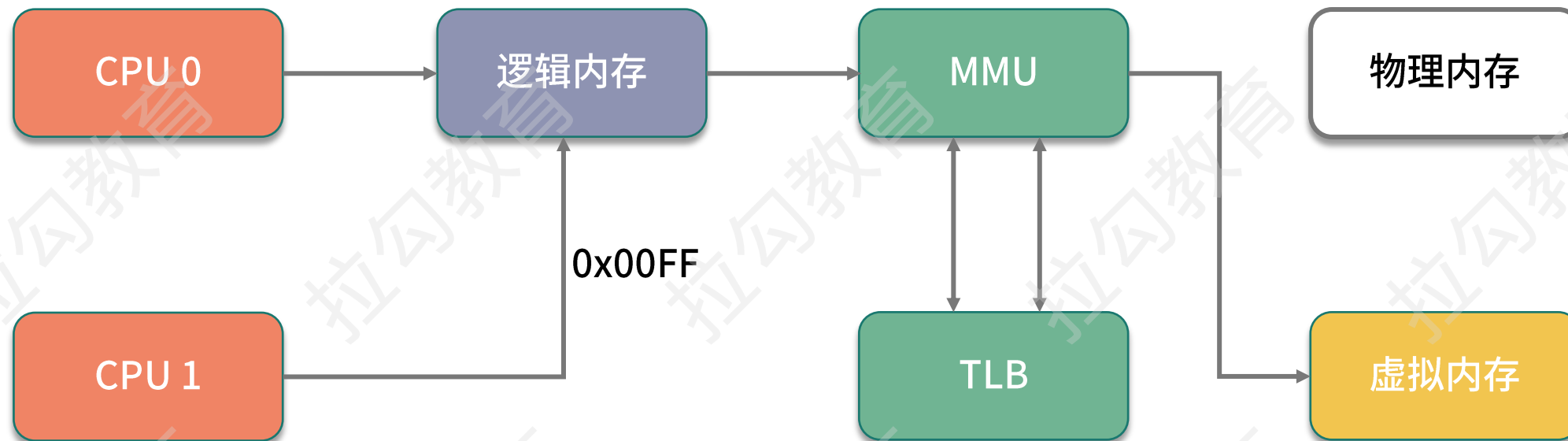
使用注解 `@sun.misc.Contended` 进行补齐，来避免伪共享的问题

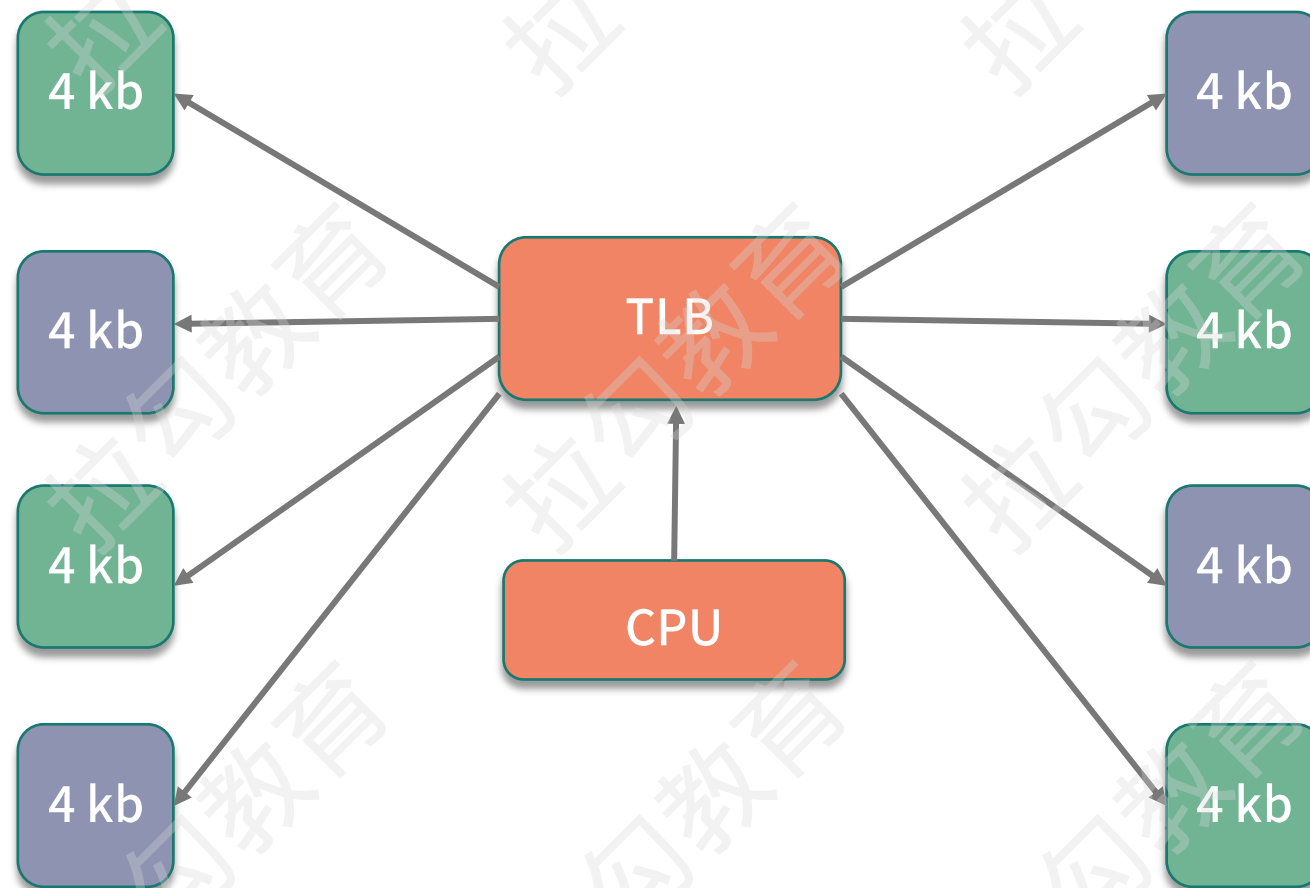


HugePage

拉勾教育

— 互联网人实战大学 —

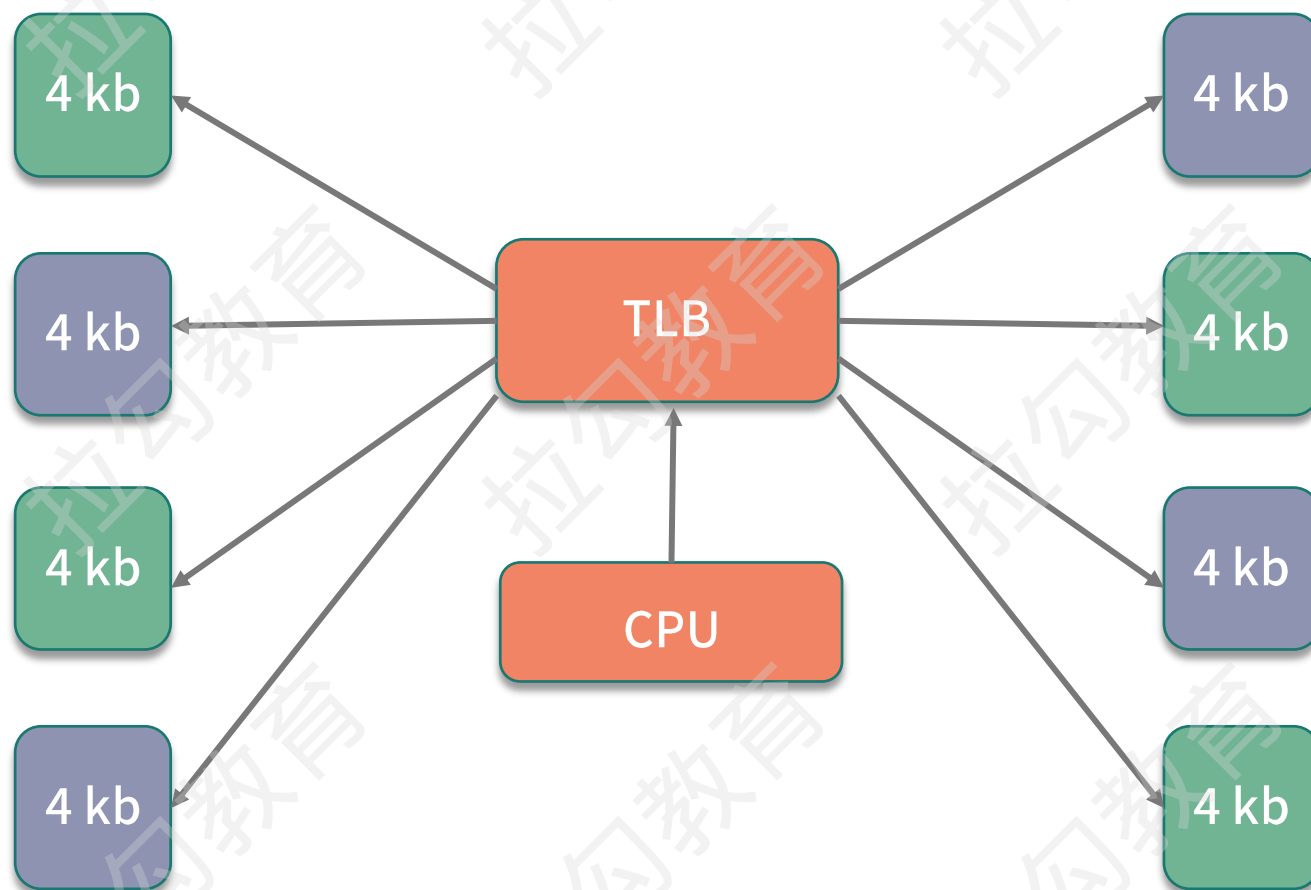




HugePage

拉勾教育

— 互联网人实战大学 —



Huge Page ——将页增大

HugePage 有一些副作用，比如竞争加剧

在一些大内存的机器上，开启后在一定程度上会增加性能



预先加载

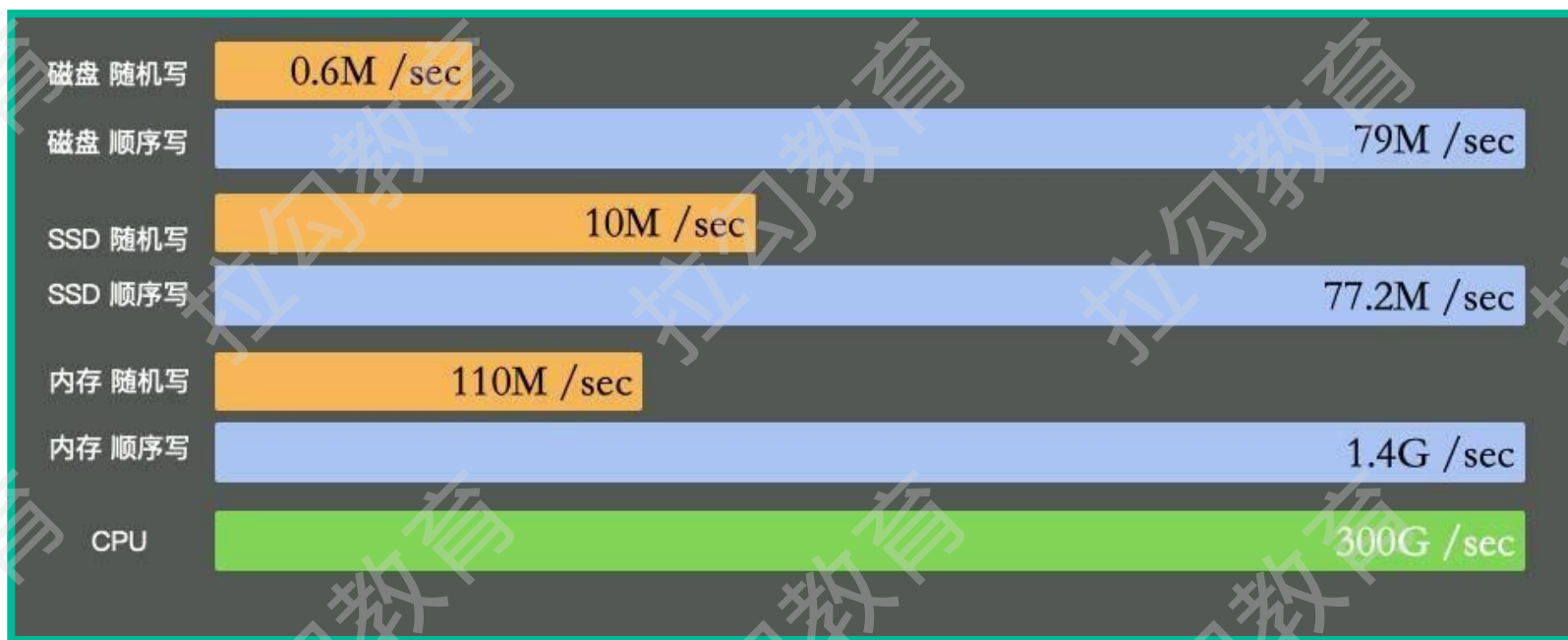
拉勾教育

— 互联网人实战大学 —

一些程序的默认行为会对性能有所影响，比如 JVM 的 `-XX:+AlwaysPreTouch` 参数
默认情况下，JVM 虽然配置了 `Xmx`、`Xms` 等参数，指定堆的初始化大小和最大大小
如果加上 **AlwaysPreTouch**，JVM 会在启动的时候，把所有的内存预先分配



I/O 设备指的不仅仅是硬盘，还包括外围的所有设备



缓冲区是解决速度差异的唯一工具

在极端情况下，比如断电时，这些缓冲区容易丢



iostat

拉勾教育

— 互联网人实战大学 —

```
top - 14:46:37 up 81 days, 3:52, 1 user, load average: 1.24, 0.97, 0.70
Tasks: 104 total, 1 running, 103 sleeping, 0 stopped, 0 zombie
%Cpu(s): 11.1 us, 2.8 sy, 0.0 ni, 64.8 id, 21.2 wa, 0.0 hi, 0.1 si, 0.0 st
KiB Mem : 8010196 total, 444552 free, 6178968 used, 1386676 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 1507016 avail Mem

PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
procs -----memory----- --swap-- --io-- --system-- --cpu-----
r  b   swpd   free   buff  cache   si   so    bi   bo   in  cs us sy id wa st
1  1       0 147812 6104 1675196    0    0    769 1466    1    1 1 92  6  0
0  1       0 142256 6292 1682868    0    0   268 58204 2681 2240 4 1 64 31  0
1  0       0 120844 6476 1701664    0    0   292 62536 7045 5374 20 3 66 11  0
1  0       0 139088 6172 1686072    0    0  2640 54144 3552 2976 16 1 68 15  0
1  0       0 158072 5840 1665288    0    0  1472 61860 5055 5182 23 1 64 11  0

12:00:01 AM      tps      rtps      wtps    bread/s    bwrtn/s
12:10:01 AM      82.80      13.21      69.59      940.04      4473.94
12:20:01 AM     217.15      19.93     197.22     1542.62     12582.43
12:30:01 AM     183.76      12.39     171.37     541.85      8118.10
12:40:01 AM     175.89      12.70     163.19     555.90      7315.44
12:50:01 AM     261.63      13.78     247.85     730.03     17758.78
01:00:01 AM     182.76      12.23     170.53     624.14      7751.02
01:10:01 AM     702.41     312.72     389.69    30118.17    51173.57
01:20:01 AM     861.42     267.68     593.74    38735.56    73936.36
01:30:01 AM     519.44      34.81     284.64    4432.57    29382.19
01:40:01 AM     236.06      30.21     205.86    3443.44    20267.84
01:50:02 AM     184.69      24.55     160.14    2978.61    18533.55
02:00:01 AM     173.79       8.08     165.71     987.01     7771.88
02:10:01 AM      72.37     10.99      61.39    1855.60     4757.25
```

iostat

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00   0.25   0.00    0.00   99.75

Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    kB/s   avgrq-sz  avgqu-sz   await  r_await  w_await  svctm  %util
sda                0.00    0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-0               0.00    0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-1               0.00    0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-2               0.00    0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00   0.00   0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00   0.00   0.00    0.00   99.50

Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    kB/s   avgrq-sz  avgqu-sz   await  r_await  w_await  svctm  %util
sda                0.00    0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-0               0.00    0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-1               0.00    0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-2               0.00    0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00   0.00   0.00
```

iostat

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.25    0.00    0.00   99.75

Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await  w_await  svctm  %util
sda                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00  0.00    0.00    0.00  0.00
dm-0                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00  0.00    0.00    0.00  0.00
dm-1                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00  0.00    0.00    0.00  0.00
dm-2                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00  0.00    0.00    0.00  0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00    0.00    0.00    0.00   99.50

Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await  w_await  svctm  %util
sda                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00  0.00    0.00    0.00  0.00
dm-0                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00  0.00    0.00    0.00  0.00
dm-1                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00  0.00    0.00    0.00  0.00
dm-2                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00  0.00    0.00    0.00  0.00
```


iostat

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00    0.25    0.00    0.00   99.75

Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    kB/s   avgrq-sz  avgqu-sz   await  r_await  w_await  svctm  %util
sda                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00    0.00   0.00
dm-0               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00    0.00   0.00
dm-1               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00    0.00   0.00
dm-2               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00    0.00   0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00    0.00    0.00    0.00   99.50

Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    kB/s   avgrq-sz  avgqu-sz   await  r_await  w_await  svctm  %util
sda                0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00    0.00   0.00
dm-0               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00    0.00   0.00
dm-1               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00    0.00   0.00
dm-2               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00   0.00    0.00    0.00   0.00
```

iostat

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00   0.25   0.00    0.00   99.75

Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  r_await  w_await  svctm  %util
sda                0.00    0.00    0.00    0.00     0.00     0.00   0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-0                0.00    0.00    0.00    0.00     0.00     0.00   0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-1                0.00    0.00    0.00    0.00     0.00     0.00   0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-2                0.00    0.00    0.00    0.00     0.00     0.00   0.00     0.00    0.00   0.00    0.00   0.00   0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00   0.00   0.00    0.00   99.50

Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  r_await  w_await  svctm  %util
sda                0.00    0.00    0.00    0.00     0.00     0.00   0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-0                0.00    0.00    0.00    0.00     0.00     0.00   0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-1                0.00    0.00    0.00    0.00     0.00     0.00   0.00     0.00    0.00   0.00    0.00   0.00   0.00
dm-2                0.00    0.00    0.00    0.00     0.00     0.00   0.00     0.00    0.00   0.00    0.00   0.00   0.00
```

iostat

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00   0.25   0.00    0.00   99.75

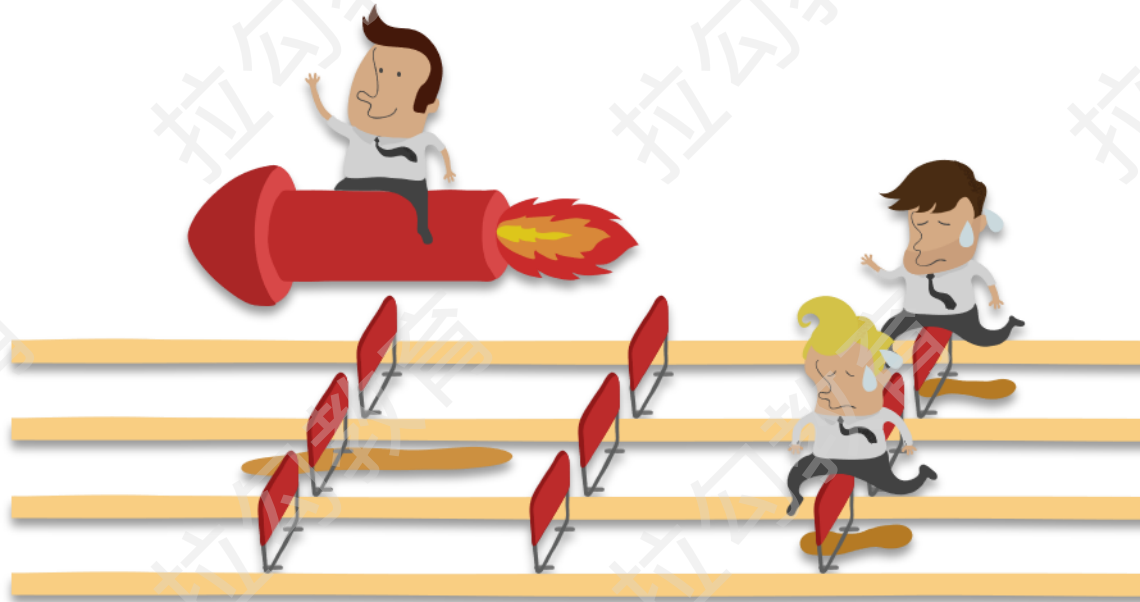
Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
sda                 0.00     0.00     0.00    0.00     0.00     0.00     0.00     0.00     0.00  0.00    0.00    0.00   0.00
dm-0                 0.00     0.00     0.00    0.00     0.00     0.00     0.00     0.00     0.00  0.00    0.00    0.00   0.00
dm-1                 0.00     0.00     0.00    0.00     0.00     0.00     0.00     0.00     0.00  0.00    0.00    0.00   0.00
dm-2                 0.00     0.00     0.00    0.00     0.00     0.00     0.00     0.00     0.00  0.00    0.00    0.00   0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00   0.00   0.00    0.00   99.50

Device:            rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
sda                 0.00     0.00     0.00    0.00     0.00     0.00     0.00     0.00     0.00  0.00    0.00    0.00   0.00
dm-0                 0.00     0.00     0.00    0.00     0.00     0.00     0.00     0.00     0.00  0.00    0.00    0.00   0.00
dm-1                 0.00     0.00     0.00    0.00     0.00     0.00     0.00     0.00     0.00  0.00    0.00    0.00   0.00
dm-2                 0.00     0.00     0.00    0.00     0.00     0.00     0.00     0.00     0.00  0.00    0.00    0.00   0.00
```

零拷贝是一种非常重要的性能优化手段

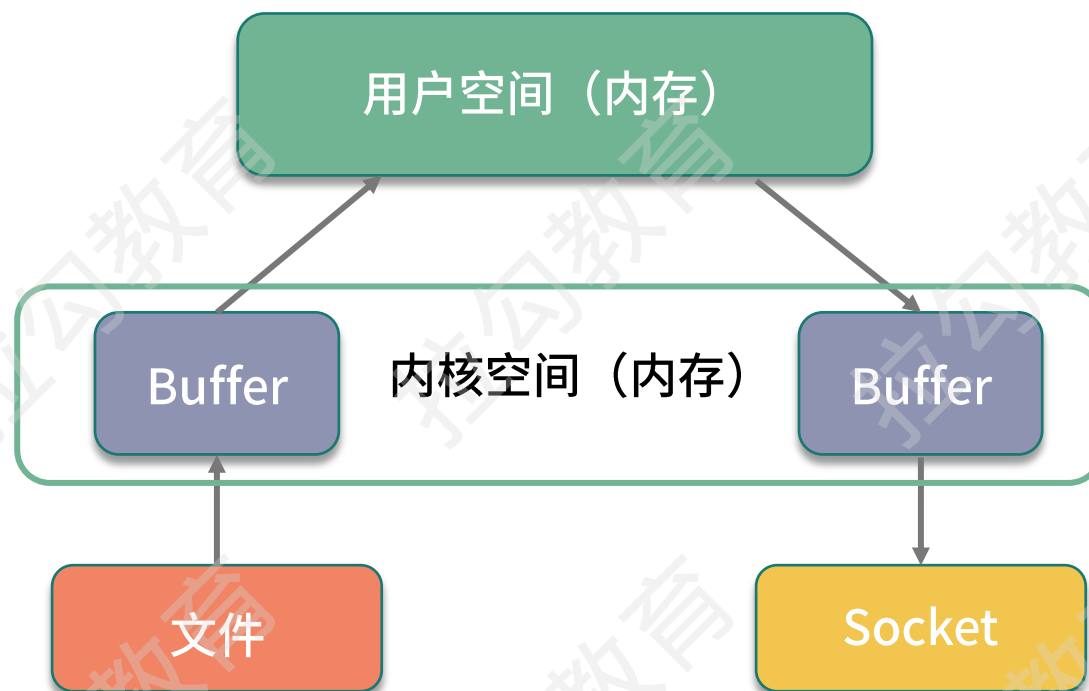
比如常见的 Kafka、Nginx 等，就使用了这种技术



没有采取零拷贝手段

拉勾教育

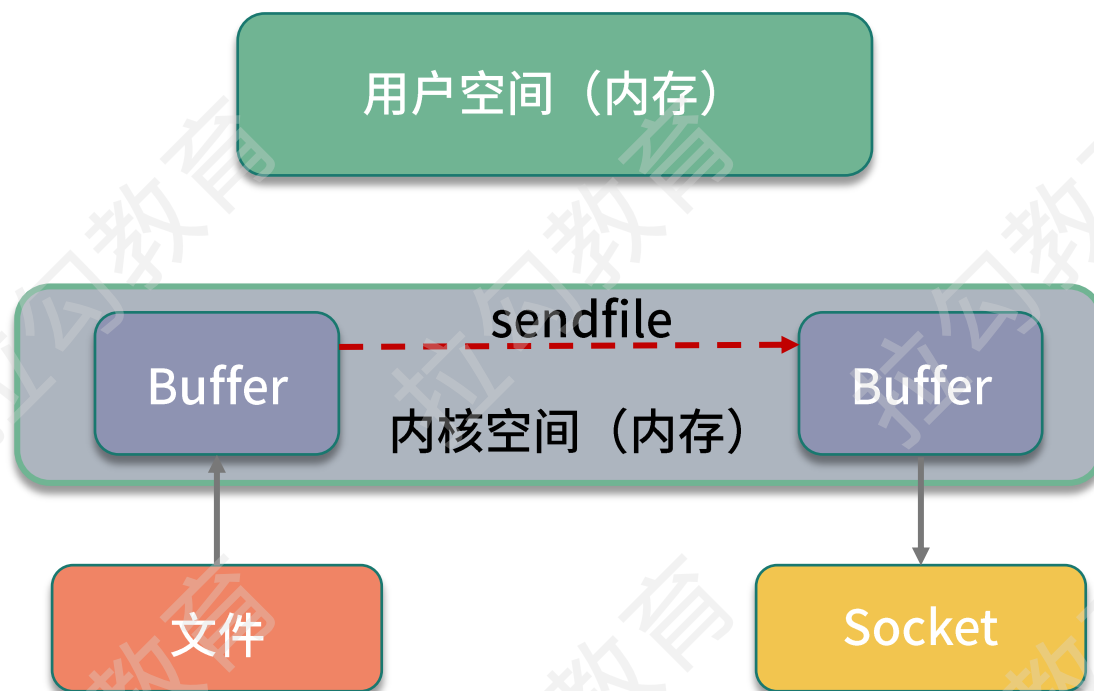
— 互联网人实战大学 —



采取了零拷贝手段

拉勾教育

— 互联网人实战大学 —



小结

拉勾教育

— 互联网人实战大学 —

本课时学习了计算机中对性能影响最大的三个组件：**CPU、内存、I/O**

并深入了解了观测它们性能的一些命令



磁盘的速度这么慢

为什么 Kafka 操作磁盘，吞吐量还能那么高？



Next: 04 | 《工具实践：如何获取代码性能数据？》

拉勾教育

— 互联网人实战大学 —



关注拉勾「教育公众号」
获取更多课程信息