

CPU 和内存监测

vmstat 命令的 VM 模式

vmstat 可以监测给定时间间隔的服务器的状态值，包括 CPU 的使用率，内存的使用，虚拟内存的交换情况，IO 读写情况。

主要从/proc/meminfo，/proc/stat 和/proc/*/stat 中获取数据

常用手段 vmstat [采样的时间间隔秒数] [采样的次数]，举例如下：

```
[root@dtbase-master-2 /root]#vmstat 5 10

procs  -----memory-----  ---swap--  -----io----  --system-
-  -----cpu-----

 r  b   swpd   free   buff  cache      si   so   bi   bo    in    cs us sy
id wa st

 1  0     0 4126684 314132 58308212    0    0    0  637     0    0  2  1
97  0  0

 1  0     0 4256016 314132 58308420    0    0    0 42898 19349 35174  3  2
95  0  0

 4  0     0 4010396 314132 58474112    0    0    0 68072 19846 37088  4  2
93  0  0

 1  0     0 4180556 314132 58308800    0    0    0 67730 19228 34979  4  2
94  0  0

 1  0     0 4158408 314140 58313740    0    0    1 36034 19217 39319  4  2
94  0  0

 2  0     0 4185816 314144 58313892    0    0    0 43252 19071 33302  3  2
95  0  0

 2  0     0 4237076 314144 58314180    0    0    0 43059 18850 36224  3  2
95  0  0

 1  0     0 4174432 314144 58314432    0    0    0 29495 18353 35612  3  2
95  0  0
```

```
2 0 0 4168024 314144 58318984 0 0 0 49652 19954 38394 3 2
94 0 0
```

```
0 0 0 4241088 314144 58319304 0 0 0 36272 17563 32637 2 1
97 0 0
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14

上述命令表示每 5 秒采样一次，采样 10 次，因此输出 10 条记录，如果不指定采样次数的话，就一直采集直到用户手动停止采集。

上述显示中每一列的含义介绍如下：

*procs 部分

*r 表示运行队列，表示 Runnable 状态的进程数，表示等待 CPU 的进程数量，越高意味着 CPU 越忙。

*b 表示阻塞的进程，表示 uninterruptible sleep 状态的进程数，通常可以认为表示等待 IO 的进程数量。

*memory 部分

*swpd 表示已使用的交换空间大小，如果大于 0，表示内存不足了。

*free 表示空闲的物理内存大小，单位是 KB，上面表示还剩 4G 物理内存空闲

*buff 表示要输出到块设备但还缓存在内存中的数据

*cache 表示从块设备读入到内存中，缓存在内存中的数据，不包括 tmpfs 消耗的内存大小

*swap 部分

*si 表示每秒有多少块从磁盘换入内存，如果这个值大于 0，表示物理内存耗尽了

*so 表示每秒有多少块从内存换入磁盘，如果这个值大于 0，表示物理内存耗

尽了

* io 部分

* bi 表示从块设备每秒收到的块数量，单位为 blocks/s

* bo 表示每秒发到块设备的块数量，单位为 blocks/s

* system 部分

* in 表示每秒 CPU 的中断次数，包括时间中断

* cs 表示每秒上下文切换次数，越小越好

* cpu 部分

* us 表示用户 cpu 时间，就是非内核态运行的 CPU 时间，单位是百分比

* sy 表示系统 cpu 时间，就是内核代码运行时间，单位是百分比

* id 表示空闲 CPU 时间，单位是百分比

* wa 表示等待 IO 的 CPU 时间，单位是百分比

* st 虚拟机占用的时间，单位是百分比

上述内存、交换区、IO 统计的单位是块数，通常在 Linux 中一个块是 1K 字节。

free 命令

参数和用法

free 命令用于查看系统整体的内存使用情况，它的参数如下：

* -b/-k/-m/-g，调整显示单位，分别表示以字节/千字节/兆字节/吉字节为单位显示，默认是-k

* -t，增加一行显示内存总量，包括总内存量、总已用内存量和总空闲内存量

* -o，不显示 buffer/cache 调整行，就是-/+ buffers/cache 行

* -l，增加两行显示低端和高端内存统计

* -s/-c，表示采样时间和采样次数

free 命令的信息从/proc/meminfo 中获取，meminfo 函数可以解析 meminfo 的信息

输出分析

free 命令输出举例如下：

```
#free -l -t
```

	total	used	free	shared	buffers	cached
Mem:	99174024	95247820	3926204	0	314920	58382632

```

Low:      99174024  95247820  3926204

High:      0        0        0

-/+ buffers/cache:  36550268  62623756

Swap:      2097144      0    2097144

Total:    101271168  95247820  6023348

```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

上述各行含义分别是：

- * **Mem**：这行是表示从 OS 角度看系统内存的使用情况。
- * **total** 表示物理内存总量
- * **used** 表示
- * **free**
- * **shared** 表示几个进程共享的内存，目前已经废弃，总是 0。
- * **buffers**，表示被 OS buffer 的内存，也就是缓冲要输出到块设备的数据
- * **cached**，表示被 OS cache 的内存，是 OS 从磁盘读出缓存到内存中的数据
- * **Low**：表示从 OS 角度看的低端内存使用情况
- * **High**：表示从 OS 角度看的高端内存使用情况，上面因为是 64 位系统，因此高端内存部分为 0。
- * **-/+ buffers/cache**：
- * **used**，表示从一个应用程序的角度看，系统有多少内存被用掉了。
- * **free**，表示从一个应用程序的角度看，系统还要多少剩余内存可用。
- * **Swap**：表示交换区信息。
- * **Total**：表示总量。

一般为了提高性能，操作系统总是会将磁盘的数据预先读到内存，而要写入磁盘的数据则先缓存在内存中，因此会看到 **free** 列比较小，而 **buffers/cache** 列比较大。

top 命令

`top` 命令用来实时监控系统的 CPU、内存等负载状态，可以按照进程实时输出各进程占用的资源状态。

参数和用法

启动参数如下：

- * `-p` 指定进程号，仅监控某个进程
- * `-c` 显示整个命令行，而不仅仅是命令名

交互命令如下：

- * `h`，输出帮助
- * `k`，杀死进程，需要根据提示输入 PID
- * `i`，忽略闲置和僵尸进程，就是忽略掉不占用 CPU 的进程
- * `f`，选择显示的列，是交互设置的，新版本还可以设置按照哪一列排序，老版本支持的字段较少。
- * `q`，退出程序

输出分析

```
top - 16:44:59 up 3 days, 30 min,  1 user,  load average: 0.00, 0.01, 0.05
```

```
Tasks:  1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
```

```
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.3 st
```

```
KiB Mem:  1883984 total,  1797296 used,    86688 free,   169388 buffers
```

```
KiB Swap: 2097148 total,        0 used,  2097148 free.  230428 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
15597	root	20	0	2809456	721308	9592	s	0.0	38.3	28:37.43	java

- 1
- 2
- 3
- 4
- 5
- 6

- 7
- 8

- 首行显示系统负载和系统运行时间，当前登录用户数
- Tasks 行显示当前系统进程按状态的统计数
- %Cpus 显示当前系统的 CPU 消耗百分比
- Kib Mem 显示内存使用量
- Kib Swap 显示交换空间使用了量

ps 命令

用于显示进程信息，有两种参数格式，一种是标准格式 `ps -opt`，一种是 BSD 格式 `ps opts`

典型用法

- 查看所有进程
 - `ps aux / ps ax`
 - `ps -eF / ps -ef / ps -ely / ps -e`
- 显示线程信息
 - `ps -elf`
 - `ps axms`
- 树形结构显示
 - `ps -ejH`
 - `ps axjf`
- 格式化输出列，格式化输出支持的选项，具体可以查看 `man ps` 的 STANDARD FORMAT SPECIFIERS 部分
 - `ps -eo [column1], [column2]`
 - `ps axo [column1], [column2]`
- 按用户输出
 - `ps U [username]`

- 排序支持，使用`-sort` 选项，`-sort spec`，这里的 `spec` 格式为`[+|-]key[,+|-]key`，默认是+排序，表示升序
 - `ps -e -sort=uid,-pid`，表示按照 `uid` 升序，`pid` 降序排序输出

输出分析

- USER 用户名
- UID 用户 ID (User ID)
- PID 进程 ID (Process ID)
- PPID 父进程的进程 ID (Parent Process id)
- SID 会话 ID (Session id)
- %CPU 进程的 `cpu` 占用率
- %MEM 进程的内存占用率
- VSZ 进程所使用的虚存的大小 (Virtual Size)
- RSS 进程使用的驻留集大小或者是实际内存的大小，Kbytes 字节。
- TTY 与进程关联的终端 (`tty`)
- STAT 进程的状态：进程状态使用字符表示的 (STAT 的状态码)
 - R 运行 Runnable (on run queue)
 - S 睡眠 Sleeping
 - I 空闲 Idle
 - Z 僵死 Zombie (a defunct process) 进程已终止，但进程描述符存在，直到父进程调用 `wait4()` 系统调用后释放。
 - D 不可中断 Uninterruptible sleep (usually IO) 收到信号不唤醒和不可运行，进程必须等待直到有中断发生。
 - T 终止 Terminate，进程收到 `SIGSTOP`, `SIGSTP`, `SIGTIN`, `SIGTOU` 信号后停止运行运行。
 - P 等待交换页
 - W 无驻留页 has no resident pages，没有足够的内存分页可分配。
 - X 死掉的进程
 - < 高优先级进程 高优先序的进程
 - N 低优先级进程 低优先序的进程
 - L 内存锁页 Lock 有记忆体分页分配并缩在记忆体内
 - s 进程的领导者 (在它之下有子进程)；
 - l 多进程的 (使用 `CLONE_THREAD`，类似 `NPTL pthreads`)
 -
 - 位于后台的进程组

- **START** 进程启动时间和日期
- **TIME** 进程使用的总 **cpu** 时间
- **COMMAND** 正在执行的命令行命令
- **NI** 优先级(Nice)
- **PRI** 进程优先级编号(Priority)
- **WCHAN** 进程正在睡眠的内核函数名称；该函数的名称是从`/root/system.map`文件中获得的。
- **FLAGS** 与进程相关的数字标识

IO 监测

iostat

参数和用法

iostat 命令的参数介绍如下

- * **-c**, 显示 **CPU** 利用率，默认启用，单独指定后默认不输出设备利用率
- * **-d**, 显示设备利用率，默认启用，单独指定后默认不输出 **cpu** 利用率
- * **-h**, 配合**-n**选项使用，使 **NFS** 系统的报告更 **human readable**
- * **-k/-m**, 调整输出单位，千字节每秒或者兆字节每秒，默认单位是块每秒
- * **-N**, 显示磁盘阵列信息
- * **-n**, 显示 **NFS** 网络文件系统的 **IO** 报告
- * **-p**, 可指定分区查看，格式为**-p [{device [,...]} | ALL]**
- * **-t**, 输出时打印时间
- * **-x**, 输出扩展信息，就是输出更详细的信息，一般都使用该参数。
- * **-z**, 忽略没有活动的设备的输出

用法如下

iostat [参数] [采样时间秒数] [采样次数]

iostat -c 2 每隔 2 秒显示 1 次 **CPU** 统计信息

iostat -d 2 每隔 2 秒显示 1 次设备统计信息

iostat -d -x 2 10 每隔 2 秒显示 1 次设备扩展统计信息，共显示 10 次。

iostat -d sda sdb -x 2 10 每隔 2 秒显示 1 次设备 **sda** 和 **sdb** 的扩展统计信息，共显示 10 次。

输出分析

```
#iostat -x 1 6
```


Linux 2.6.32-220.23.2.ali878.el6.x86_64 (dtbase-master-2)
07/08/2016 _x86_64_ (32 CPU)

avg-cpu: %user %nice %system %iowait %steal %idle

 2.03 0.00 1.14 0.01 0.00 96.82

Device:	rrqm/s	wrqm/s	r/s	w/s	rsec/s	wsec/s	avgrq-sz
avgqu-sz	await	svctm	%util				
sda	0.01	3964.69	0.17	591.78	7.41	41068.15	69.39
0.14	0.24	0.03	1.75				

#iostat -d 5 1

Linux 2.6.32-220.23.2.ali878.el6.x86_64 (dtbase-master-2)
07/08/2016 _x86_64_ (32 CPU)

Device:	tps	Blk_read/s	Blk_wrtn/s	Blk_read	Blk_wrtn
sda	593.11	7.39	41159.65	5836572	32509188906

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14

上述输出中各部分信息描述如下：

- * **avg-cpu** 部分，对应-c 输出的内容
- * **%user**，表示用户 CPU 时间百分比
- * **%nice**，表示用户态 CPU 的 nice 操作消耗的 CPU 时间百分比
- * **%system**，表示内核 CPU 时间百分比
- * **%iowait**，表示 CPU 花在 io 等待上的时间百分比
- * **%steal**，表示虚机占用的 CPU 时间百分比
- * **%idle**，表示 CPU 空闲时间百分比
- * **device** 部分，对应-d 输出的内容
- * **rrqm/s**，表示每秒这个设备有多少读取请求被 Merge 了，当系统调用读取数据时，VFS 将请求发给各个 FS，各个 FS 会根据请求是否是读取相同的 block 来合并读请求。
- * **wrqm/s**，表示每秒这个设备有多少写请求被 Merge 了。
- * **r/s**，表示每秒发送到设备的读取请求数
- * **w/s**，表示每秒发送到设备的写入请求数
- * **rsec/s**，表示每秒读取的扇区数量
- * **wsec/s**，表示每秒写入的扇区数量
- * **avgrq-sz**，表示发送到设备的请求的平均大小，单位是扇区
- * **avgqu-sz**，表示发送到设备的请求的平均队列长度
- * **await**，表示 IO 请求的平均执行时间，包括发送请求和执行的时间，但是为毫秒
- * **svctm**，表示发送到设备的 IO 请求的平均执行时间，单位是毫秒，这里仅包含执行时间，不包含发送请求的时间
- * **%util**，表示设备带宽利用率，实际是 IO 请求发送到设备期间占用的 CPU 时间的百分比。
- * **tps**，表示该设备每秒的传输次数，一次传输是一次 IO，多个逻辑 IO 可能被合并为一次物理 IO，一次 IO 的大小未知。
- * **blk_read/s**，表示每秒读入块的数量
- * **blk_wrtn/s**，表示每秒写入块的数量
- * **blk_read**，表示系统启动以来读入块的总数
- * **blk_wrtn**，表示系统启动以来写入块的总数

vmstat 的 disk 和 partition 模式

disk 模式

vmstat -d [采样时间] [采样次数]，列出所有块设备的读写情况，每个块设备一行。

```
[root@iz2371imwx3z ~]# vmstat -d 5 2

disk-----reads-----writes-----
IO-----

      total merged sectors      ms  total merged sectors      ms    cur
sec

xvda 2143955 51246 98066178 79024080 787667 454849 30875448 26476350
0 2937

xvda 2143955 51246 98066178 79024080 787668 454849 30875456 26476352
0 2937

• 1
• 2
• 3
• 4
• 5
```

上述命令中各个字段的含义是：

* **reads** 部分

* **total**，表示系统启动到现在完成读的总次数

* **merged**，表示系统启动到现在 **group** 读的次数，**group read** 是反映为一次 IO

* **sectors**，表示系统启动到现在读了多少个扇区

* **ms**，表示系统启动到现在读消耗的毫秒数

* **writes** 部分

* **total**，表示系统启动到现在完成写的总次数

* **merged**，表示系统启动到现在 **group** 写的次数，**group write** 反映为一次 IO

* **sectors**，表示系统启动到现在写了多少个扇区

* **ms**，表示系统启动到现在写消耗的毫秒数

* **IO** 部分

* **cur**：正在处理的 IO 数

* **sec**，表示系统启动到现在 IO 消耗的时间

partition 模式

vmstat -p 设备分区 [采样时间] [采样次数]，列出指定分区的读写情况

```
[root@iz2371imwx3z ~]# vmstat -p /dev/xvda1 5 2
```

xvda1	reads	read sectors	writes	requested writes
	2143776	98064746	788481	30893256
	2143776	98064746	788482	30893288

- 1
- 2
- 3
- 4

上述命令中各个字段的含义如下：

- * **reads**，表示系统启动到现在此分区上总的读次数
- * **read sectors**，表示系统启动到现在此分区上读的扇区总数
- * **writes**，表示系统启动到现在此分区上总的写次数
- * **requested writes**，表示系统启动到现在对此分区的写请求总数

网络监测

netstat

netstat 命令用来显示网络连接状态信息，这个命令官方已经废弃，可以用 **ss** 代替

参数和用法

- **-r**，显示路由表信息，现在由 **ip route** 命令替代。
- **-g**，显示组播信息，现在由 **ip maddr** 命令替代。
- **-i**，显示接口信息，包括接口的 **MTU**、二层包统计，接口标签信息。现在由 **ip -s link** 替代
- **-s**，显示各协议的总的统计信息
- **-v**，显示详细信息
- **-W**，宽显示，不会截断 **ip** 地址
- **-n**，以数字形式显示地址端口用户，而不是以 **DNS**、协议名、用户名显示
- **-protocol=family**，指定协议族，包括 **inet**, **inet6**, **unix**, **ipx**, **ax25**, **netrom**, **econet**, **ddp**，可以简写为 **-4**, **-6** **-x**
- **-c**，每秒连续输出
- **-e**，显示扩展信息，目前主要是增加了 **user** 和 **inode** 列显示
- **-o**，增加一列显示定时器信息

- -p, 增加一列显示该 socket 所属的进程号和进程信息
- -l, 仅显示监听端口, 监听端口默认是不输出的
- -a, 显示监听和非监听的端口
- -r -F, 输出 FIB 的路由信息, 内核路由表
- -r -C, 输出路由缓存的路由信息, 内核路由缓存
- delay 整数参数, 指定每个多少秒输出一次

典型用法:

netstat -anp

输出分析

```
[root@iZ2371imwx3Z ~]# netstat --protocol=inet -a -W -e -o -v -p Active
Internet connections (servers and established)Proto Recv-Q Send-Q Local
Address          Foreign Address    State      User       Inode
PID/Program name  Timer

tcp        0      0 0.0.0.0:https      0.0.0.0:*          LISTEN
root       315002    15597/java      off (0.00/0/0)

tcp        0      0 localhost:mxid     0.0.0.0:*          LISTEN
root       315068    15597/java      off (0.00/0/0)

tcp        0      0 0.0.0.0:8009      0.0.0.0:*          LISTEN
root       314999    15597/java      off (0.00/0/0)

tcp        0      0 0.0.0.0:http      0.0.0.0:*          LISTEN
root       314998    15597/java      off (0.00/0/0)

tcp        0      0 0.0.0.0:ssh       0.0.0.0:*          LISTEN
root       13009     780/sshd        off (0.00/0/0)

tcp        1      0 iZ2371imwx3Z:http  iZ2371imwx3Z:55825
CLOSE_WAIT root      906801    15597/java      off (0.00/0/0)

tcp        0      0 iZ2371imwx3Z:45708 69.172.201.153:http
ESTABLISHED root      734431    15597/java      off (0.00/0/0)

tcp        0      0 iZ2371imwx3Z:45613 69.172.201.153:http
ESTABLISHED root      882000    15597/java      off (0.00/0/0)
```

```
tcp      1      0 iZ2371imwx3Z:http      iZ2371imwx3Z:42164
CLOSE_WAIT root      890885    15597/java      off (0.00/0/0)
```

```
#netstat -x -e -o -pActive UNIX domain sockets (w/o servers)Proto RefCnt
```

Flags	Type	State	I-Node	PID/Program name	Path
unix 2	[]	DGRAM		23365 9251/java	@alimonitor9251
unix 2	[]	DGRAM		15468 2820/java	@alimonitor2820
unix 2	[]	DGRAM		26408 9670/java	@alimonitor9670
unix 2	[]	DGRAM		9514 1108/udev	@/org/kernel/udev/udev
unix 3	[]	STREAM	CONNECTED	62870049 19964/syslog-ng	/dev/log
unix 3	[]	STREAM	CONNECTED	62870048 17807/pickup	

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19

- 20
- 21
- 22
- Inet 和 Inet6 域 Socket 的输出
 - Proto, 当前 socket 的协议类型, tcp、udp、udpl、raw
 - Recv-Q
 - ESTABLISHED 状态表示用户程序还没有从这个 socket 读取的字节数
 - LISTENING 状态表示当前的 syn backlog
 - Send-Q
 - ESTABLISHED 状态表示对端没有 ACK 的字节数
 - LISTENING 状态表示 syn backlog 的最大 size
 - Local Address: 本地地址和端口
 - Foreign Address: 对端地址和端口
 - State, 表示 socket 状态, 一般仅在 TCP 下才有效, 对应 TCP 的各个状态
 - User, 表示这个 socket 的属主
 - PID/Program name, 表示这个 socket 所属进程的 PID 和进程名
- UNIX 域 socket 的输出
 - Proto, 当前协议类型, unix
 - RefCnt, 当前 socket 关联的进程数
 - Flags, socket 选项
 - Type, socket 类型, 包括数据报、流类型、RAW 类型、RDM、SEQPACKET 类型、SOCK_PACKET
 - State, Free 未分配, CONNECTING 表示连接建立, 空表示未连接
 - Path, 表示该 socket 路径

SS

ss 用于替代 netstat 命令, 用于输出 socket 的统计信息, 输出类似 netstat, 可以显示更详细的 TCP 状态信息。

参数和用法

大部分参数和 `netstat` 一样，下面描述几个特有的参数

- * `-i`，输出 TCP 内部的一些信息，包括各个连接的水位、选项的值。
- * `-m`，输出 socket 的一些内存使用信息
- * `-s`，按协议输出统计信息
- * `-D filename`，dump socket 原始信息到文件，这里是二进制信息。
- * `-t/-u/-4/-6/-x/-w/-0`，指定协议的快捷选项，分别表示 tcp、udp、ip、ipv6、unix、raw、packet

ifconfig

显示接口信息，目前 `ip` 命令可以实现其所有功能，这个命令正逐步被废弃。

参数和用法

- `-a`，显示所有接口，包括 down 的接口，默认 down 接口不显示
- `-s`，简要显示，类似 `netstat -i`
- `up/down` 启用/禁用接口
- `arp/-arp` 在指定接口上启用/禁用 ARP
- `promisc/-promisc` 在指定接口上启用/禁用混杂模式
- `allmulti/-allmulti`，在指定接口上启用/禁用组播
- `mtu N`，设置 MTU
- `dstaddr addr`，点对点链路配置远端地址
- `netmask addr`，设置网络掩码
- `add addr/prefixlen`，添加地址
- `del addr/prefixlen`，删除地址
- `media type`，设置接口媒体类型
- `broadcast addr`，设置或者取消接口广播地址
- `pointopoint [addr]`，设置接口为点对点模式，`addr` 表示远端的协议层地址
- `hw class address`，设置接口硬件地址，`class` 包括 ether、ax25、ARCnet、netrom

典型用法

`ifconfig [接口名] 选项`

`ifconfig eth0 add 192.168.1.1/24`

输出分析


```
[root@iZ2371imwx3Z ~]# ifconfig eth0:
flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

    inet 10.252.100.80 netmask 255.255.248.0 broadcast
10.252.103.255

    ether 00:16:3e:00:85:43 txqueuelen 1000 (Ethernet)

    RX packets 27674 bytes 5689074 (5.4 MiB)

    RX errors 0 dropped 0 overruns 0 frame 0

    TX packets 33051 bytes 4136017 (3.9 MiB)

    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

ip

ip 是显示和管理网络设备、路由、邻接表、策略路由、隧道的命令，功能强大。

参数和用法

链路管理

ip link

地址管理

ip addr

路由管理

ip route

ARP 和邻接表管理

ip neigh

ip ntable

路由策略管理

ip rule

TCP Metric 管理

ip tcp_metrics

ip 隧道管理

ip tunnel