Investigação de Load alto

> As médias de load no Linux são uma medida de quantos processos estavam utilizando a CPU (ou aguardando para usá-la) durante um determinado período de tempo, normalmente 1 minuto, 5 minutos ou 15 minutos.
>
> O load é sempre uma média ao longo do tempo, pois o número de processos usando a CPU, ou esperando por ela, é volátil. Pode haver 10 processos em um segundo e 0 no próximo, por isso a média fornece uma visão significativa da carga de trabalho da CPU.

Segue abaixo um exemplo de médias de Load:

```
load average over the last 1 minute: 1.05
load average over the last 5 minutes: 0.70
load average over the last 15 minutes: 5.09
```

Esses números dizem que a média de load foi de 1,05 no último minuto, 0,70 nos últimos 5 minutos e 5,09 nos últimos 15 minutos, respectivamente.

O que esses números dizem depende do sistema. Uma média de load de 5 em um sistema com um único núcleo de CPU significaria que havia no máximo 1 processo em execução e 4 esperando em média, ou seja, esse sistema estaria em overload. Uma média de load de 5 em um sistema com 8 núcleos de CPU significa que há no máximo 5 processos em execução em núcleos diferentes, com 3 núcleos ociosos. Esse sistema não estaria em overload.

Identificando um gargalo

De um modo geral, existem duas causas principais de overload:

- Existem muitos processos em execução.
- Os processos estão aguardando I/O.

1. Nesse caso, há mais processos em execução do que a CPU pode suportar, com isso ela não pode concluir a execução dos processos rápido o suficiente para evitar que vários processos precisem esperar pelo tempo da CPU.

2. A segunda causa, quando um processo faz uma solicitação para um dispositivo de I/O, aguarda a conclusão da solicitação de I/O antes de retomar a execução. Há um grande número de dispositivos de I/O (monitor, alto-falantes, impressora, teclado, mouse, disco rígido, placa de rede) mas, no contexto de investigações de load nos concentramos no disco rígido. As solicitações de I/O não estão sendo concluídas com rapidez suficiente e a CPU normalmente gasta uma quantidade substancial de tempo ocioso, apesar do load alto.

Determinar o que está causando a carga é fundamental para qualquer investigação, e uma das ferramentas mais eficazes para fazer isso é o comando `sar`.

O pacote sysstat contém várias ferramentas úteis para investigar o desempenho e o uso em um sistema, entre eles o sar. O sar formata os dados coletados pelo sysstat em intervalos regulares sobre várias partes do sistema. Como existem muitos dados coletados para serem efetivamente visualizados de uma só vez, o `sar` tem opções para especificar quais partes desses dados serão exibidas. Abaixo iremos detalhar algumas das opções mais úteis para nossa demanda:

Estatísticas de load

A opção `-q` do `sar` mostrará dados sobre processos na fila e load. Aqui está o trecho relevante do manual:

```
-q     Report queue length and load averages. The following values are displayed:

       runq-sz
              Run queue length (number of tasks waiting for run time).

       plist-sz
              Number of tasks in the task list.

       ldavg-1
              System load average for the last minute.  The load average is calculated as the average number
of runnable or running tasks (R state), and the number of tasks in  unin
              terruptible sleep (D state) over the specified interval.

       ldavg-5
              System load average for the past 5 minutes.

       ldavg-15
              System load average for the past 15 minutes.

       blocked
              Number of tasks currently blocked, waiting for I/O to complete.
```

- runq-sz: Tamanho da fila de execução (número de tarefas aguardando para execução).
- plist-sz: O número de tarefas na lista de processos.

Esta opção fornecerá uma visão geral do load ao longo do tempo. Quando o servidor está sobrecarregado, as médias de load aumentam assim como os valores em `runq-sz` e `plist-sz`. Os valores na coluna `blocked` geralmente aumentam significativamente apenas se o I/O for um fator, mas não é um indicador confiável de que o I/O é a causa da sobrecarga.

Deve-se tomar bastante cuidado na verificação desses dados. As médias de load em 10 não indicam overload se o servidor tiver 20 núcleos de CPU. 10 processos bloqueados quando há 100 processos na fila é diferente de 10 processos bloqueados com 1000 processos e nenhum indica realmente quanto tempo os processos aguardam o I/O.

> A opção `-q` é útil para estabelecer se ocorreu uma sobrecarga e em que período ocorreu essa sobrecarga, mas são necessárias outras opções para entender melhor porque a sobrecarga ocorreu.

Estatísticas de I/O

Existem várias opções que exibem estatísticas para vários dispositivos de I/O, mas vamos detalhar a opção `-d`. A opção `-d` exibe estatísticas gerais para os discos rígidos. Segue o trecho da man page

```
-d     Report activity for each block device.  When data are displayed, the device specification dev m-n is
generally used ( DEV column).  m is the major number of the device  and  n its  minor  number. Device  names
may also be pretty-printed if option -p is used or persistent device names can be printed if option -j is
used (see below).  Note that disk activity depends on sadc options "-S DISK" and "-S XDISK" to be collected.
The following values are displayed:

       tps
              Indicate the number of transfers per second that were issued to the device.  Multiple logical
requests can be combined into a single I/O request to the device. A transfer is of indeterminate size.

       rd_sec/s
              Number of sectors read from the device. The size of a sector is 512 bytes.

       wr_sec/s
              Number of sectors written to the device. The size of a sector is 512 bytes.

       avgrq-sz
              The average size (in sectors) of the requests that were issued to the device.

       avgqu-sz
              The average queue length of the requests that were issued to the device.

       await
              The  average time (in milliseconds) for I/O requests issued to the device to be served. This
includes the time spent by the requests in queue and the time spent servicing them.

       svctm
              The average service time (in milliseconds) for I/O requests that were issued to the device.
Warning! Do not trust this field any more. This field will be removed  in  a future sysstat version.

       %util
              Percentage of elapsed time during which I/O requests were issued to the device (bandwidth
utilization for the device). Device saturation occurs when this value is close to 100%.
```

O `sar` mostrará estatísticas para cada dispositivo. Se o servidor tiver várias unidades de disco, ele produzirá uma linha para cada dispositivo a cada intervalo. Por exemplo:

```
12:00:04 AM       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svctm     %util
12:01:01 AM       sdb      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:01:01 AM       sda    531.05  27682.44  29893.98    108.42      1.65      3.12      0.14      7.43
12:02:02 AM       sdb      1.74      0.00     53.68     30.79      0.00      0.63      0.08      0.01
12:02:02 AM       sda    125.99    318.82   2419.47     21.73      0.40      3.16      0.09      1.12
12:03:02 AM       sdb      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:03:02 AM       sda    122.10    384.06   2367.85     22.54      0.40      3.28      0.08      1.00
12:04:01 AM       sdb      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
12:04:01 AM       sda    133.57    319.38   2937.14     24.38      0.45      3.34      0.09      1.20
```

Esta saída mostra que existem duas unidades `sda` e `sdb`. É importante prestar atenção ao dispositivo e quais dados estão nesse dispositivo. Uma unidade de backup saturada causará problemas diferentes de uma unidade primária saturada.

Use a opção `-p` em conjunto com a `-d` opção para que os nomes dos dispositivos sejam exibidos da melhor forma.

O modo como cada um desses campos se relaciona ao load é muito menos claro, mas para fins de determinação da causa de overload, os campos a `wait` e `%util` são os mais significativos. Aqui estão as descrições desses campos novamente para dar ênfase.

```
await
       The  average time (in milliseconds) for I/O requests issued to the device to be served. This includes
the time spent by the requests in queue and the time spent servicing them.

%util
       Percentage of elapsed time during which I/O requests were issued to the device (bandwidth utilization
for the device). Device saturation occurs when this value is close to 100%.
```

`%util` é, em teoria, o mais direto dos dois. Se `%util` estiver próximo ou igual a 100%, o dispositivo está saturado. No entanto, se o dispositivo for uma matriz RAID, `%util` torna-se uma medida muito pior de saturação. Ainda é uma medida razoável da atividade geral de I/O e valores próximos a 100% ainda são um indicador de que pode haver um gargalo de I/O, mas não é tão fácil de identificar quanto em unidades únicas.

O campo `await` geralmente será a maneira mais confiável de identificar gargalos de I/O. Se os valores no campo tiverem mais 3 dígitos, haverá algum tipo de gargalo de I/O. Qualquer coisa maior que 50 é provavelmente um problema, mas valores menores como esse tornam dificultam a identificação. Geralmente, você terá que procurar outros sinais, como valores grandes no campo `blocked`, ou valores `%util` próximos a 100% em conjunto com `await` 50 +/- 20.

Estatísticas de memória

Embora o uso da memória não esteja relacionado a overload, as estatísticas da memória devem ser verificadas quando existirem indicações de saturação para a unidade primária. Especificamente, o sistema deve ser verificado quanto a sinais de thrashing. Thrashing é o termo para o que acontece quando um sistema quase esgotou sua memória e começa a mover rapidamente as páginas da memória da memória física para o disco e vice-versa. O thrashing aumenta a utilização do disco e geralmente leva a sobrecargas.

A opção geral para estatísticas de memória com o `sar` é a opção `-r`. Segue trecho do manual:

```
-r     Report memory utilization statistics.  The following values are displayed:

       kbmemfree
              Amount of free memory available in kilobytes.

       kbmemused
              Amount of used memory in kilobytes. This does not take into account memory used by the kernel
itself.

       %memused
              Percentage of used memory.

       kbbuffers
              Amount of memory used as buffers by the kernel in kilobytes.

       kbcached
              Amount of memory used to cache data by the kernel in kilobytes.

       kbcommit
              Amount of memory in kilobytes needed for current workload. This is an estimate of how much RAM
/swap is needed to guarantee that there never is out of memory.

       %commit
              Percentage of memory needed for current workload in relation to the total amount of memory
(RAM+swap). This number may be greater than 100% because the kernel usually overcommits memory.

       kbactive
              Amount of active memory in kilobytes (memory that has been used more recently and usually not
reclaimed unless absolutely necessary).

       kbinact
              Amount of inactive memory in kilobytes (memory which has been less recently used. It is more
eligible to be reclaimed for other purposes).

       kbdirty
              Amount of memory in kilobytes waiting to get written back to the disk.
```

Esses campos não podem ser visualizados de forma isolada. Os campos de uso, `kbmemfree`, `kbmemused`, e `%memused`, mostram a quantidade de memória utilizada/disponível, mas não indica necessariamente que há um problema. A memória usada pelos buffers e cache, os campos `kbbuffers` e `kbcached`, respectivamente, podem ser descartados da memória para liberar memória para aplicações. De um modo geral, o uso de memória será próximo de 100% nos sistemas que estão fazendo uso efetivo de seu hardware.

O campo `%commit` pode ser um forte indicador de um problema, mas não há um limite claro. Uma boa regra geral é que qualquer valor acima de 200% é um problema, mas alguns sistemas podem ter problemas em 150% e às vezes valores próximos a 100%. Lembre-se sempre de analisar esse valor de acordo com o hardware do servidor.

Nenhum desses campos é um forte indicador de thrashing. O thrashing refere-se a páginas sendo movidas para dentro e para fora do disco, dentro e fora da swap. A opção `-S` exibe estatísticas sobre a utilização da swap. Segue trecho do manual:

```
-S     Report swap space utilization statistics.  The following values are displayed:

       kbswpfree
              Amount of free swap space in kilobytes.

       kbswpused
              Amount of used swap space in kilobytes.

       %swpused
              Percentage of used swap space.

       kbswpcad
              Amount  of  cached  swap  memory  in kilobytes.  This is memory that once was swapped out, is
swapped back in but still also is in the swap area (if memory is needed it doesn't need to be swapped out
again because it is already in the swap area. This saves I/O).

       %swpcad
              Percentage of cached swap memory in relation to the amount of used swap space.
```

Nenhum desses campos também é um forte indicador de trashing. Isso é incluído aqui para enfatizar que **nenhum desses campos, inclusive `%swpused` , é um indicador de thrashing**. Se o sistema estiver com **thrashing**, `%swpused` estará próximo de 100%, mas também estará próximo de 100% com frequência em sistemas sob condições perfeitamente normais. Se `%swpused` não estiver perto de 100%, é seguro dizer que o sistema não está com problemas. Se o servidor não possuir swap, ele não pode ter **thrashing.**

A opção `-B` exibe estatísticas de paginação (memória) e será uma boa opção se o sistema estiver com **thrashing**. Segue manual:

```
-B     Report paging statistics.  The following values are displayed:

       pgpgin/s
              Total number of kilobytes the system paged in from disk per second.

       pgpgout/s
              Total number of kilobytes the system paged out to disk per second.

       fault/s
              Number of page faults (major + minor) made by the system per second.  This is not a count of
page faults that generate I/O, because some page  faults  can  be  resolved without I/O.

       majflt/s
              Number of major faults the system has made per second, those which have required loading a
memory page from disk.

       pgfree/s
              Number of pages placed on the free list by the system per second.

       pgscank/s
              Number of pages scanned by the kswapd daemon per second.

       pgscand/s
              Number of pages scanned directly per second.

       pgsteal/s
              Number of pages the system has reclaimed from cache (pagecache and swapcache) per second to
satisfy its memory demands.

       %vmeff
              Calculated  as  pgsteal / pgscan, this is a metric of the efficiency of page reclaim. If it is
near 100% then almost every page coming off the tail of the inactive list is being reaped. If it gets too
low (e.g. less than 30%) then the virtual memory is having some difficulty.  This field is displayed as
zero  if  no  pages  have  been scanned during the interval of time.
```

Vários desses campos podem ser usados para indicar **thrashing**, mas o campo `%vmeff` geralmente é o único no qual você precisa prestar atenção. Este campo é uma medida da eficiência geral da memória virtual. Os valores aqui são idealmente exatamente 0,00 (significando que a memória virtual não foi acessada) ou quase 100,00 (a memória virtual foi acessada, mas não está causando problemas importantes ao sistema). Se o sistema estiver com trashing, `%vmeff` deve ser 30% ou inferior. Valores entre 30% e 70% são um tanto ambíguos, e esse campo deve ser analisado junto com outros campos para ajudar a determinar se o trashing está realmente ocorrendo.

Se você acredita que o sistema estava sobrecarregado como resultado do trashing e se estabilizou rapidamente, verifique em `/var/log/messages` se há registros de Out of Memory (OOM) Killer. Eles são uma forte indicação de que o sistema estava ficando sem memória e que o kernel precisou interromper um processo para liberar memória para evitar uma falha.

> É comum ver instâncias do killer do OOM nos logs quando o servidor está executando o CloudLinux, mesmo quando o sistema não está com alto uso de memória.

Resumo de verificações no sar:

Variáveis	Load	Disco	RAM
Opções	*-q*	*-d -p*	*-r (ram)*
O que verificar (1)	runq-sz e plist-sz altos	%util estiver próximo ou igual a 100%, o disco está saturado	%commit com valor acima de 100% pode indicar sobrecarga
O que verificar (2)	----------	await com mais de 3 dígitos, haverá algum tipo de gargalo de I/O	%commit com valor acima de 200% é certeza de sobrecarga