

Compreenda a utilização da alta utilização da CPU relatada pelo vManage para o vEdge 5000/1000/100B e as Plataformas da nuvem do vEdge

Índice

[Introdução](#)

[Compreenda a utilização da alta utilização da CPU que é relatada no vEdge 5000/1000/100B e nas Plataformas da nuvem do vEdge](#)

[Explicação](#)

[Uso da alta utilização da CPU com FP-um processo](#)

[Conclusão](#)

Introdução

Este original descreve porque você pôde ver o uso da alta utilização da CPU relatado no vManage para o vEdge 5000/1000/100B e nas Plataformas da nuvem do vEdge apesar do desempenho das Plataformas que são normais sem a alta utilização da CPU relatada como vista na **parte superior**.

Compreenda a utilização da alta utilização da CPU que é relatada no vEdge 5000/1000/100B e nas Plataformas da nuvem do vEdge

Com o 17.2.x e as liberações mais atrasadas, um CPU mais alto e o consumo de memória para Plataformas do vEdge e da nuvem do vEdge podem ser observados. Isto é observado no painel do vManage para um dispositivo dado. Em alguns casos, isto igualmente conduz a um número aumentado de alertas e de avisos no vManage.

Explicação

A razão para o uso relatado da alta utilização da CPU quando o dispositivo executa normalmente com o normal, o ponto baixo, ou a nenhuma carga é devido a uma mudança na fórmula usada a fim calcular o uso. Com as 17.2 liberações, a utilização CPU é computada com base na **média de carga do status de sistema da mostra** no vEdge.

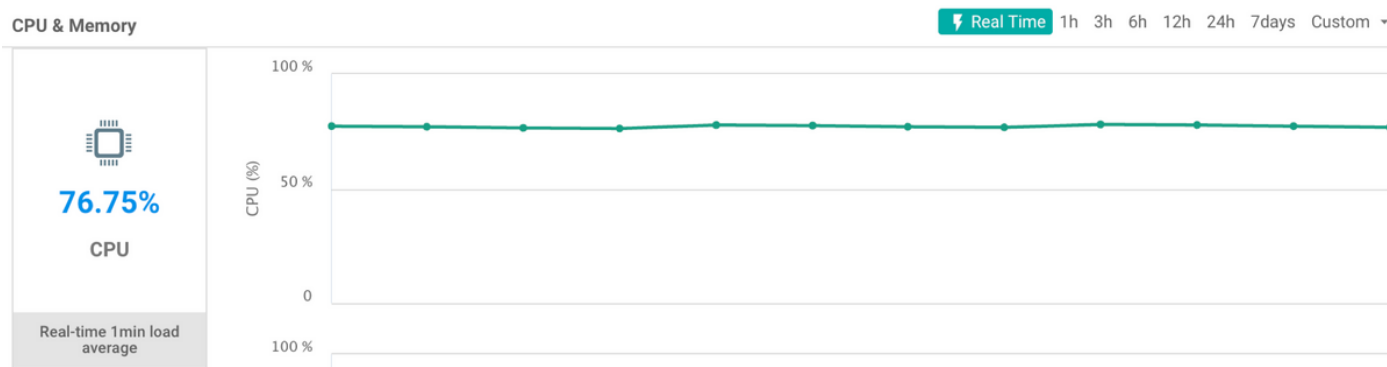
o vManage mostra a utilização CPU do tempo real para um dispositivo. Puxa o **1 minuto [min1_avg]** e **5 [min5_avg] minuto da média médio** baseado em dados históricos. **A média de carga**, por definição, inclui as várias coisas e não apenas ciclos de CPU que contribuem ao cálculo da utilização. Por exemplo, o tempo de espera IO, processa durante o tempo, e outros valores são considerados quando você apresenta este valor para a plataforma. Neste caso, você ignora os valores mostrados para os estados CPU e valores CPU no **comando top do vShell**.

Está aqui um exemplo em como a utilização CPU, que é realmente a **1 média de carga minuto**, obtém calculada e mostrada no painel do vManage:

Quando você verifica a carga de um vEdge CLI, este pode ser visto:

```
vEdge# show system status | include Load
Load average:          1 minute: 3.10, 5 minutes: 3.06, 15 minutes: 3.05
Load average:          1 minute: 3.12, 5 minutes: 3.07, 15 minutes: 3.06
Load average:          1 minute: 3.13, 5 minutes: 3.08, 15 minutes: 3.07
Load average: 1 minute: 3.10, 5 minutes: 3.07, 15 minutes: 3.05
```

Neste caso, a utilização CPU é computada com base na **média de carga/# dos núcleos (vCPUs)**. Para este exemplo, o nó tem 4 núcleos. A média de carga está convertida então por um fator de 100 antes que você se divida pelo número de núcleos. Quando você calcula a média da média de carga de todos os núcleos e a multiplica por 100, você chega em um valor de ~310. Tome este valor e a partilha por 4 rendimentos, uma leitura CPU de 77.5% CPU, que alinhasse com o valor considerado no gráfico em tempo real no vManage capturado em torno do tempo a saída CLI foi recolhida e segundo as indicações da imagem.



A fim considerar as médias de carga e o número de núcleos CPU no sistema, a saída da **parte superior** pode ser consultada do vShell no dispositivo.

No exemplo abaixo, o vEdge contém 4 vCPUs. O primeiro núcleo (**Cpu0**) é usado para o **controle** (visto completamente a utilização mais baixa do usuário) quando os 3 núcleos permanecendo forem usados para **dados**:

```
top - 01:14:57 up 1 day, 3:15, 1 user, load average: 3.06, 3.06, 3.08
Tasks: 219 total, 5 running, 214 sleeping, 0 stopped, 0 zombie
Cpu0  :  1.7%us,  4.0%sy,  0.0%ni, 94.3%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu1  : 56.0%us, 44.0%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu2  : 54.2%us, 45.8%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu3  : 59.3%us, 40.7%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   7382664k total, 2835232k used, 4547432k free, 130520k buffers
Swap:   0k total,    0k used,    0k free, 587880k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
978	root	20	0	3392m	664m	127m	R	100	9.2	1635:21	fp-um-2
692	root	20	0	3392m	664m	127m	R	100	9.2	1635:18	fp-um-1
979	root	20	0	3392m	664m	127m	R	100	9.2	1634:51	fp-um-3
694	root	20	0	1908m	204m	131m	S	1	2.8	15:29.95	ftmd
496	root	20	0	759m	72m	3764	S	0	1.0	1:31.50	confd

A fim obter o número de CPU do vEdge CLI, este comando pode ser usado:

```
vEdge# show system status | display xml | include total_cpu
```

```
<total_cpu_count>4</total_cpu_count>
```

Um outro exemplo do cálculo para o valor mostrado no vManage no vEdge 1000 é fornecido aqui. Depois que você emite a **parte superior** do vShell, I está interessado a fim indicar a carga para todos os núcleos:

```
top - 18:19:49 up 19 days, 1:37, 1 user, load average: 0.55, 0.71, 0.73
```

Desde que um vEdge 1000 tem somente um núcleo CPU disponível, a carga relatada aqui é 55% (0.55×100).

Uso da alta utilização da CPU com FP-um processo

Você pode igualmente às vezes observar da **parte superior** que FP-um o processo executa altamente e mostra até 100% CPU. Isto é esperado nos núcleos CPU que são usados para o processamento plano dos dados.

Do comando top provido mais cedo, 3 núcleos operam-se em 100% CPU e 1 núcleo mostra a utilização normal:

```
top - 01:14:57 up 1 day, 3:15, 1 user, load average: 3.06, 3.06, 3.08
Tasks: 219 total, 5 running, 214 sleeping, 0 stopped, 0 zombie
Cpu0  :  1.7%us,  4.0%sy,  0.0%ni, 94.3%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu1  : 56.0%us, 44.0%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu2  : 54.2%us, 45.8%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu3  : 59.3%us, 40.7%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   7382664k total, 2835232k used, 4547432k free, 130520k buffers
Swap:      0k total,      0k used,      0k free, 587880k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
978	root	20	0	3392m	664m	127m	R	100	9.2	1635:21	fp-um-2
692	root	20	0	3392m	664m	127m	R	100	9.2	1635:18	fp-um-1
979	root	20	0	3392m	664m	127m	R	100	9.2	1634:51	fp-um-3

Este primeiro núcleo (Cpu0) é usado para o **controle** e os três núcleos restantes usados para **dados**. Como você pode ver na lista de processo, FP-um o processo usa aqueles recursos.

FP-um é um processo que use um direcionador votação-MODE, assim que significa que senta e vota a porta subjacente para pacotes constantemente de modo que possa processar todo o quadro assim que for recebido. Este processo segura a transmissão e é equivalente à transmissão do caminho rápido no vEdge 1000, no vEdge 2000, e no vEdge 100. Esta arquitetura votação-MODE é usada por Intel para o processamento do pacote eficiente baseado na estrutura DPDK. Porque o encaminhamento de pacote é executado em um laço apertado, o CPU permanece ou perto em 100% em todas as vezes. Embora isto seja feito, nenhuma latência é introduzida com estes CPU porque este é comportamento esperado.

A informações de fundo na votação DPDK pode ele encontrou [aqui](#).

A nuvem do vEdge e as Plataformas do vEdge 5000 usam a mesma arquitetura da transmissão e exibem o mesmo comportamento a este respeito. Está aqui um exemplo de um vEdge 5000 puxado da saída **superior**. Tem 28 núcleos, de que 2 (Cpu0 e Cpu1) são usados para o **controle** (como o vEdge 2000) e 26 são usados para **dados**.

top - 02:18:30 up 1 day, 7:33, 1 user, load average: 26.24, 26.28, 26.31

Tasks: 382 total, 27 running, 355 sleeping, 0 stopped, 0 zombie

Cpu0 : 0.7%us, 1.3%sy, 0.0%ni, 98.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1 : 0.7%us, 1.3%sy, 0.0%ni, 98.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2 : 79.4%us, 20.6%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3 : 73.4%us, 26.6%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu4 : 73.4%us, 26.6%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu5 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu6 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu7 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu8 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu9 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu10 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu11 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu12 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu13 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu14 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu15 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu16 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu17 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu18 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu19 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu20 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu21 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu22 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu23 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu24 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu25 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu26 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu27 :100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 32659508k total, 10877980k used, 21781528k free, 214788k buffers

Swap: 0k total, 0k used, 0k free, 1039104k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2028	root	20	0	12.1g	668m	124m	R	100	2.1	1897:21	fp-um-3
2029	root	20	0	12.1g	668m	124m	R	100	2.1	1897:22	fp-um-4
2030	root	20	0	12.1g	668m	124m	R	100	2.1	1897:12	fp-um-5
2031	root	20	0	12.1g	668m	124m	R	100	2.1	1897:22	fp-um-6
2032	root	20	0	12.1g	668m	124m	R	100	2.1	1897:22	fp-um-7
2034	root	20	0	12.1g	668m	124m	R	100	2.1	1897:22	fp-um-9
2035	root	20	0	12.1g	668m	124m	R	100	2.1	1897:21	fp-um-10
2038	root	20	0	12.1g	668m	124m	R	100	2.1	1897:21	fp-um-13
2040	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-15
2041	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-16
2043	root	20	0	12.1g	668m	124m	R	100	2.1	1897:22	fp-um-18
2045	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-20
2052	root	20	0	12.1g	668m	124m	R	100	2.1	1897:18	fp-um-27
2033	root	20	0	12.1g	668m	124m	R	100	2.1	1897:21	fp-um-8
2036	root	20	0	12.1g	668m	124m	R	100	2.1	1897:21	fp-um-11
2037	root	20	0	12.1g	668m	124m	R	100	2.1	1897:21	fp-um-12
2039	root	20	0	12.1g	668m	124m	R	100	2.1	1897:09	fp-um-14
2042	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-17
2044	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-19
2046	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-21
2047	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-22
2048	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-23
2049	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-24
2050	root	20	0	12.1g	668m	124m	R	100	2.1	1897:22	fp-um-25
2051	root	20	0	12.1g	668m	124m	R	100	2.1	1897:23	fp-um-26
1419	root	20	0	116m	5732	2280	S	0	0.0	0:02.00	chmgrd
1323	root	20	0	753m	70m	3764	S	0	0.2	1:51.20	confd
1432	root	20	0	1683m	172m	134m	S	0	0.5	0:58.91	fpmd

Aqui, a média de carga é sempre alta porque 26 da corrida de 28 processadores em 100% devido

FP-um ao processo.

Conclusão

O USO de CPU relatado no vManage para as liberações 17.2.x antes de 17.2.7 não é o USO de CPU real mas é calculado pelo contrário com base na média de carga. Isto pode conduzir à confusão em compreender o valor relatado e conduzir aos alarmes falsos relativos à alta utilização da CPU quando a plataforma se operar normalmente com normal, o ponto baixo, ou nenhuns tráfego/carga de rede reais.

Este comportamento é mudado/alterado com os 17.2.7 e os 18.2 libera tais que a leitura CPU pode agora ser exata baseada na leitura do `cpu_user` da **parte superior**.

A edição é mencionada igualmente nos [17.2 Release Note](#).