Verständnis der von vManage für vEdge 500/2000/1000/100B und vEdge Cloud-Plattformen gemeldeten hohen CPU-Auslastung

Inhalt

Einführung

In diesem Dokument wird erläutert, warum in vManage für vEdge 500/2000/1000/1000/100B- und vEdge-Cloud-Plattformen möglicherweise eine hohe CPU-Auslastung festgestellt wird, obwohl die Leistung der Plattformen normal ist und keine hohe CPU **als oben** angezeigt wird.

Verständnis der hohen CPU-Auslastung, die auf den vEdge 500/2000/1000/100B- und vEdge Cloud-Plattformen gemeldet wird

Bei den Versionen 17.2.x und höher kann eine höhere CPU- und Speicherbelegung für vEdge- und vEdge-Cloud-Plattformen beobachtet werden. Dies wird auf dem vManage Dashboard für ein bestimmtes Gerät festgestellt. In einigen Fällen führt dies auch zu einer erhöhten Anzahl von Warnmeldungen und Warnungen in vManage.

Erläuterung

Der Grund für die gemeldete hohe CPU-Auslastung, wenn das Gerät normal, niedrig oder ohne Last arbeitet, liegt in einer Änderung der Formel zur Berechnung der Auslastung. Bei den Versionen 17.2 wird die CPU-Auslastung basierend auf dem **Lastdurchschnitt** aus dem **Systemstatus** im vEdge berechnet.

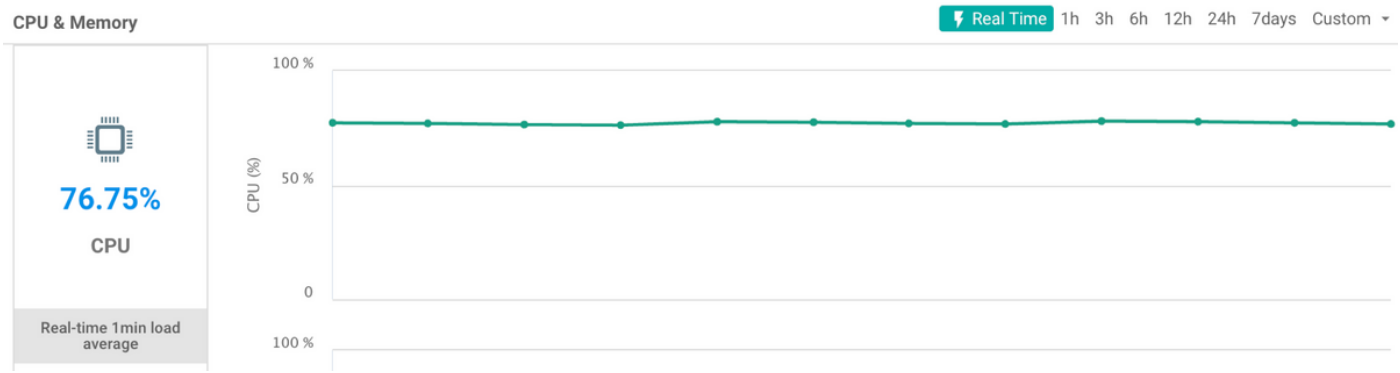
vManage zeigt die CPU-Nutzung für ein Gerät in Echtzeit an. Der **1-minütige Durchschnitt [min1_avg]** und der **5-minütige Durchschnitt [min5_avg]** basieren auf historischen Daten. **Load Average** umfasst definitionsgemäß verschiedene Aspekte und nicht nur CPU-Zyklen, die zur Nutzungsberechnung beitragen. Beispielsweise werden die E/A-Wartezeit, die Prozessauslaufzeit und andere Werte berücksichtigt, wenn Sie diesen Wert für die Plattform angeben. In diesem Fall ignorieren Sie die Werte, die für die CPU-Zustände und CPU-Werte im **oberen** Befehl von vShell angezeigt werden.

Im folgenden Beispiel wird die CPU-Auslastung, also der **1-minütige Lastdurchschnitt**, berechnet und im vManage-Dashboard angezeigt:

Wenn Sie die Last von einer vEdge-CLI überprüfen, wird Folgendes angezeigt:

```
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
```

In diesem Fall wird die CPU-Auslastung basierend auf **Load-Average/# of Cores (vCPUs)** berechnet. In diesem Beispiel hat der Knoten vier Kerne. Der Lastmittelwert wird dann um den Faktor 100 konvertiert, bevor Sie ihn durch die Anzahl der Kerne dividieren. Wenn Sie den Durchschnitt für die Last aller Kerne multiplizieren und mit 100 multiplizieren, ergibt sich ein Wert von ~310. Dividieren Sie diesen Wert durch vier Felder, eine CPU-Messung von 77,5 %, die mit dem Wert übereinstimmt, der in der Echtzeit-Grafik in vManage zu sehen ist. Diese Werte wurden um den Zeitpunkt erfasst, an dem die CLI-Ausgabe erfasst wurde, wie im Bild gezeigt.



Um die Lastdurchschnitte und die Anzahl der CPU-Kerne im System anzuzeigen, kann die Ausgabe von **top** von vShell auf dem Gerät abgerufen werden.

Im folgenden Beispiel enthält der vEdge 4 vCPUs. Der erste Kern (**Cpu0**) wird für die **Steuerung** verwendet (durch die geringere Benutzerauslastung sichtbar), während die restlichen 3 Kerne für **Daten** verwendet werden:

```
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
```

Um die Anzahl der CPUs aus der vEdge-CLI abzurufen, kann der folgende Befehl verwendet werden:

```
vEdge# show system status | display xml | include total_cpu
    <total_cpu_count>4</total_cpu_count>
```

Ein weiteres Beispiel für die Berechnung des in vManage auf dem vEdge 1000 angezeigten Werts

finden Sie hier. Nachdem Sie **Top** von vShell ausgegeben haben, möchte **I** die Last für alle Kerne anzeigen:

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

Da ein vEdge 1000 nur über einen CPU-Core verfügt, wird hier eine Last von 55 % (0,55 x 100) gemeldet.

Hohe CPU-Auslastung mit fp-um-Prozess

Sie können auch manchmal von **oben** bemerken, dass der **fp-um-**Prozess hoch läuft und bis zu 100 % CPU anzeigt. Dies ist bei den CPU-Kernen zu erwarten, die für die Datenebenenverarbeitung verwendet werden.

Über den **oben** genannten Befehl werden drei Kerne mit 100 % CPU betrieben, und ein Kern weist eine normale Auslastung auf:

```
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
...
```

Dieser erste Kern (Cpu0) wird für **Control** verwendet, und die drei verbleibenden Kerne werden für **Data** verwendet. Wie Sie in der Prozessliste sehen können, verwendet der **fp-um-**Prozess diese Ressourcen.

fp-um ist ein Prozess, der einen Pollenmodus-Treiber verwendet. Das bedeutet, dass der zugrunde liegende Port ständig für Pakete sitzt und abfragt, sodass er jeden Frame verarbeiten kann, sobald er empfangen wird. Dieser Prozess verarbeitet die Weiterleitung und entspricht der Fast-Path-Weiterleitung im vEdge 1000, vEdge 2000 und vEdge 100. Diese Polll-Mode-Architektur wird von Intel für eine effiziente Paketverarbeitung auf der Basis des Data Plane Development Kit (DPDK)-Frameworks verwendet. Da die Paketweiterleitung in einer engen Schleife implementiert wird, bleibt die CPU jederzeit bei oder nahe bei 100 %. Obwohl dies der Fall ist, wird durch diese CPUs keine Latenz eingeführt, da dieses Verhalten erwartet wird.

Hintergrundinformationen zum DPDK-Polling finden Sie hier.

Die vEdge Cloud- und vEdge 500-Plattformen verwenden dieselbe Weiterleitungsarchitektur und weisen in dieser Hinsicht dasselbe Verhalten auf. Hier ein Beispiel von einem vEdge 500, der von der **oberen** Ausgabe heruntergeladen wurde. Er verfügt über 28 Kerne, von denen 2 (CPU0 und CPU1) für die **Steuerung** verwendet werden (wie der vEdge 2000) und 26 für **Daten** verwendet werden.

```
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
```

Hier ist der Lastdurchschnitt immer hoch, da 26 der 28 Prozessoren aufgrund des **fp-um-**

Prozesses mit 100 % ausgeführt werden.

Schlussfolgerung

Die gemeldete CPU-Auslastung in vManage für 17.2.x-Versionen vor 17.2.7 ist nicht die tatsächliche CPU-Auslastung, sondern basiert auf dem Load Average. Dies kann zu Verwirrung beim Verständnis des gemeldeten Werts führen und zu Fehlalarmen im Zusammenhang mit hoher CPU führen, während die Plattform normal mit normalem, niedrigem oder nicht tatsächlichem Datenverkehr bzw. Netzwerkauslastung arbeitet.

Dieses Verhalten wird mit den Versionen 17.2.7 und 18.2 so geändert/geändert, dass die CPU-Lektüre jetzt exakt sein kann, basierend auf der **oben** abgebildeten CPU_user-Lektüre.

Das Problem wird auch in den Versionshinweisen zu Version 17.2 erwähnt.