

# Quantum a virtualisé l'exemple distribué par noyau de paquet (QvPC-DI) : supervision des performances de l'unité de processeur d'iftask et de réseau (NPU)

## Contenu

[Introduction](#)

[Composants utilisés](#)

[Architecture d'iftask](#)

[Surveillance de la représentation d'iftask](#)

[Bulkstats](#)

## Introduction

Ce document décrit comment surveiller la représentation de l'iftask/NPU sur des QvPC-DI.

### [Composants utilisés](#)

Les informations dans ce document sont basées sur des QvPC-DI.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

## Architecture d'iftask

l'iftask est un processus en QvPC-DI. Il active la fonctionnalité du kit de développement de plan de données (DPDK) sur la carte virtuelle de fonction de service (SF) et la carte virtuelle de fonction de contrôle (CF) pour les ports de réseau de DI et les ports de service. DPDK est plus de façon efficace de manipuler l'entrée/sortie dans les environnements virtualisés.

Les pilotes de périphérique des contrôleurs d'interface de réseau à hautes performances (NIC) sont maintenant déplacés à l'userspace, cela évite les Commutateurs chers de contexte (userspace/kernelspace).

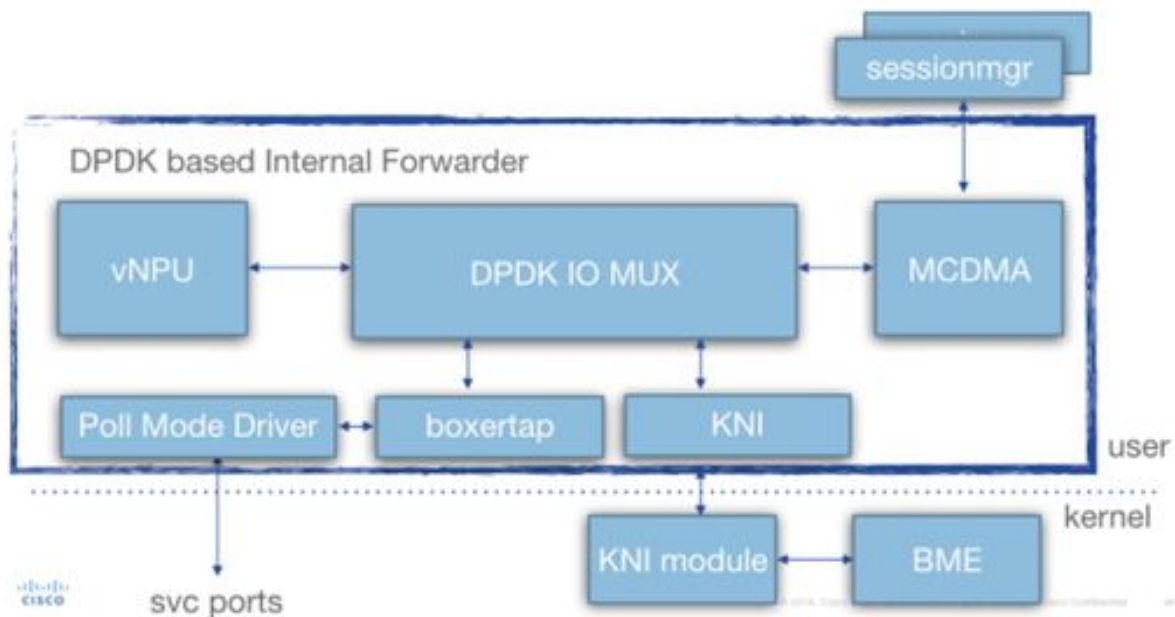
Les gestionnaires exécutés en mode sans coupure dans l'userspace, et les thread ont l'accès direct aux files d'attente HW/aux mémoires tampons de sonnerie dans des ces pilotes de carte NIC.

La documentation au sujet de l'architecture est disponible à :

Entretient ultra l'introduction de plate-forme (USP) à partir [ultra du guide d'administration système de plate-forme de passerelle](#).

## [Disponibilité pour des différentes versions.](#)

L'architecture en profondeur d'iftask (pour SF) est vue dans ce diagramme :



Pendant le déploiement, le certain montant d'unités centrales virtuelles (vCPU) sont statiquement alloués au processus d'iftask. Ceci réduit la quantité de noyaux pour des applications d'userspace (sessmgr etc.), mais il améliorent considérablement la représentation de l'E/S.

Cette allocation est faite par l'intermédiaire du paramètre ci-dessous dans ce **modèle param.cfg** qui est associé avec chaque SF/CF pendant le déploiement :

- IFTASK\_CORES (% de noyaux disponibles à assigner avec l'iftask)
- (IFTASK\_CRYPTOCORES - non discuté ici)
- (IFTASK\_MCDMA\_CORES - non discuté ici)

1. Sur un SF, le processus d'iftask intérieurement distribuera ses noyaux assignés dans : Votez les vCPU des gestionnaires de mode (PMD) (faisant l'activité tx/rx/vnpuVCPU MCDMA, faisant le transfert des paquets à partir de l'iftask au sessmgr et au dos
2. Sur un CF, aucun vCPU MCDMA n'est exigé, puisque SF n'accueillent pas des processus de sessmgr.

La commande « **iftask de matériel de nuage d'exposition** » fournit plus de détails au sujet de ceci sur votre déploiement QVPC-DI :

```
[local]UGP# show cloud hardware iftask
```

```
Card 1:
```

```
Total number of cores on VM:      8
Number of cores for PMD only:      0
Number of cores for VNPU only:     0
Number of cores for PMD and VNPU:  2    <-- CF: 2 out of 8 cores are assigned to iftask
PMD/VNPU
Number of cores for MCDMA:         0    <-- CF: no cores allocated to MCDMA as there is no
sessmgr process on CF
Number of cores for Crypto:        0
```

```

Hugepage size:          2048 kB
Total hugepages:       3670016 kB
NPUSHM hugepages:     0 kB
CPU flags: avx sse sse2 ssse3 sse4_1 sse4_2
Poll CPU's: 1 2
KNI reschedule interval: 5 us
...
Card 3:
  Total number of cores on VM:      8
  Number of cores for PMD only:     0
  Number of cores for VNPU only:    0
  Number of cores for PMD and VNPU: 2    <-- SF: 2 out of 8 core are assigned to iftask
PMD/VNPU
Number of cores for MCDMA: 1 <-- SF: 1 out of 8 cores is assigned to iftak MCDMA
Number of cores for Crypto: 0
Hugepage size: 2048 kB
Total hugepages: 4718592 kB
NPUSHM hugepages: 0 kB
CPU flags: avx sse sse2 ssse3 sse4_1 sse4_2
Poll CPU's: 1 2 3
KNI reschedule interval: 5 us

```

La commande « **configuration de nuage d'exposition** » fournira plus de détails sur les paramètres utilisés :

```

[local]UGP# show cloud configuration
Card 1:
  Config Disk Params:
  -----
CARDSLOT=1
CPUID=0
CARDTYPE=0x40010100
DI_INTERFACE=BOND:TYPE:ixgbevf-1,TYPE:ixgbevf-2
DI_INTERFACE_VLANID=2111
VNFM_INTERFACE=MAC:fa:16:3e:23:aa:e9
VNFM_PROXY_ADDRS=172.16.180.3,172.16.180.5,172.16.180.6
MGMT_INTERFACE=MAC:fa:16:3e:87:23:9b
VNFM_IPV4_ENABLE=true
VNFM_IPV4_DHCP_ENABLE=true

  Local Params:
  -----
CARDSLOT=1
CARDTYPE=0x40010100
CPUID=0
...
Card 3:
  Config Disk Params:
  -----
CARDSLOT=3
CPUID=0
CARDTYPE=0x42030100
DI_INTERFACE=BOND:TYPE:ixgbevf-1,TYPE:ixgbevf-2
SERVICE1_INTERFACE=BOND:TYPE:ixgbevf-3,TYPE:ixgbevf-4
SERVICE2_INTERFACE=BOND:TYPE:ixgbevf-5,TYPE:ixgbevf-6
DI_INTERFACE_VLANID=2111
VNFM_INTERFACE=MAC:fa:16:3e:29:c6:b7
IFTASK_CORES=30
VNFM_IPV4_ENABLE=true
VNFM_IPV4_DHCP_ENABLE=true

```

Local Params:

-----  
CARDSLOT=3  
CARDTYPE=0x42010100  
CPUID=0

## Surveillance de la représentation d'iftask

Le processus d'iftask peut être surveillé de plusieurs manières.

Consolidez la liste de commandes show :

```
show subscribers data-rate
show npumgr dinet utilization pps
show npumgr dinet utilization pps
show cloud monitor di-network summary
show cloud hardware iftask
show cloud configuration
show iftask stats summary
show port utilization table
show npu utilization table
show npumgr utilization information
show processes cpu
```

Les informations CPU de **#show de commande bavardes** ne fourniront pas des informations sur les noyaux d'iftask. Ils seront toujours répertoriés à l'utilisation de 100%.

Dans l'exemple ci-dessous, le noyau 1,2,3 sont associés avec l'iftask, et sont répertoriés à l'utilisation de 100%, ceci est prévu.

```
Card 3, CPU 0:
  Status                : Standby, Kernel Running, Tasks Running
  Load Average          : 3.12, 3.12, 3.13 (3.95 max)
  Total Memory          : 16384M
  Kernel Uptime         : 4D 21H 56M
  Last Reading:
    CPU Usage All       : 1.9% user, 0.3% sys, 0.0% io, 0.0% irq, 97.8% idle
    Core 0              : 5.8% user, 0.2% sys, 0.0% io, 0.0% irq, 94.0% idle
    Core 1              : Not Averaged (Poll CPU)
    Core 2              : Not Averaged (Poll CPU)
    Core 3              : Not Averaged (Poll CPU)
    Core 4              : 2.2% user, 0.2% sys, 0.0% io, 0.0% irq, 97.6% idle
    Core 5              : 0.8% user, 0.5% sys, 0.0% io, 0.0% irq, 98.7% idle
    Core 6              : 0.4% user, 0.5% sys, 0.0% io, 0.0% irq, 99.1% idle
    Core 7              : 0.1% user, 0.3% sys, 0.0% io, 0.0% irq, 99.6% idle
  Poll CPUs            : 3 (1, 2, 3)
    Core 1              : 100.0% user, 0.0% sys, 0.0% io, 0.0% irq, 0.0% idle
    Core 2              : 100.0% user, 0.0% sys, 0.0% io, 0.0% irq, 0.0% idle
    Core 3              : 100.0% user, 0.0% sys, 0.0% io, 0.0% irq, 0.0% idle
  Processes / Tasks    : 143 processes / 16 tasks
  Network mcdmaN       : 0.002 kpps rx, 0.001 mbps rx, 0.002 kpps tx, 0.001 mbps tx
  File Usage           : 1504 open files, 1627405 available
  Memory Usage         : 7687M 46.9% used
  Memory Details:
    Static              : 330M kernel, 144M image
    System              : 10M tmp, 0M buffers, 54M kcache, 79M cache
    Process/Task        : 6963M (120M small, 684M huge, 6158M other)
```

```

Other          : 104M shared data
Free           : 8696M free
Usable        : 5810M usable (8696M free, 0M reclaimable, 2885M reserved by tasks)

```

La table d'utilisation de npu de #show de commande donnera un bon résumé sur l'utilisation de chaque noyau associé avec le processus d'iftask (sur chaque carte).

Remarque: Important voici identifier si quelques noyaux sont uniformément plus élevés dans l'utilisation que d'autres noyaux.

```
[local]UGP# show npu utilization table
```

```

-----iftask-----
 lcore      now    5min   15min
-----
01/0/1      0%     0%     0%
01/0/2      0%     0%     0%
02/0/1      0%     0%     0%
02/0/2      2%     1%     0%
03/0/1      0%     0%     0%
03/0/2      0%     0%     0%
03/0/3      0%     0%     0%
04/0/1      0%     0%     0%
04/0/2      0%     0%     0%
04/0/3      0%     0%     0%
05/0/1      0%     0%     0%
05/0/2      0%     0%     0%
05/0/3      0%     0%     0%

```

Les informations d'utilisation de npumgr de #show de commande ([commande masquée](#))

Cette commande fournit plus d'informations sur chaque noyau d'iftask, et ce qui consomme la CPU sur ces noyaux.

Remarque: Les noyaux PMD ont leur CPU consommée sur PortRX, PortTX, KNI, chiffrement.

Remarque: Les noyaux MCDMA ont leur CPU consommée par MCDMA.

Les noyaux PMD et MCDMA ont même le chargement.

Si ce n'est pas le cas, certains accordant pourraient être exigés (allouer plus/moins de MDMA creuse par exemple).

```

***** show npumgr utilization information 3/0/0 *****
 5-Sec Avg: lcore01| lcore02| lcore03| lcore04| lcore05| lcore06| lcore07| lcore08| lcore09|
lcore10| lcore11| lcore12|
  Idle:      41%|    47%|    73%|    62%|    48%|    49%|    69%|    71%|    49%|
64%|    48%|    69%|
  PortRX:    32%|    36%|    0%|    0%|    0%|    0%|    31%|    29%|    0%|
0%|    0%|    31%|
  PortTX:    16%|    18%|    0%|    0%|    0%|    0%|    0%|    0%|    0%|
0%|    0%|
  KniRX:     3%|    0%|    0%|    0%|    0%|    0%|    0%|    0%|    0%|
0%|    0%|
  McdmaRX:   0%|    0%|    11%|    15%|    22%|    21%|    0%|    0%|    20%|

```

14%	20%	0%								
	Mcdma:	0%	0%	1%	2%	3%	2%	0%	0%	3%
3%	3%	0%								
	McdmaFlush:	0%	0%	15%	21%	27%	27%	0%	0%	27%
18%	28%	0%								
	Cipher:	7%	0%	0%	0%	0%	0%	0%	0%	0%
0%	0%	0%								
rx kbits/sec:	173914	257249	248543	224081	319973	299437	450992	380112	391400	
325099	355809	399515								
rx frames/sec:	30557	36549	37465	32560	46914	43938	70711	54818	58482	
47462	53477	58793								
tx kbits/sec:	169641	251958	253930	228760	326714	305750	440773	372187	399806	
331914	363491	391002								
tx frames/sec:	30551	36524	37465	32560	46914	43938	70681	54785	58483	
47462	53477	58749								

5-Min Avg: lcore01   lcore02   lcore03   lcore04   lcore05   lcore06   lcore07   lcore08   lcore09   lcore10   lcore11   lcore12										
	Idle:	18%	36%	60%	62%	46%	45%	65%	62%	44%
53%	39%	65%								
	PortRX:	29%	45%	0%	0%	0%	0%	35%	38%	0%
0%	0%	35%								
	PortTX:	17%	20%	0%	0%	0%	0%	0%	0%	0%
0%	0%	0%								
	KniRX:	3%	0%	0%	0%	0%	0%	0%	0%	0%
0%	0%	0%								
	Kni:	1%	0%	0%	0%	0%	0%	0%	0%	0%
0%	0%	0%								
	McdmaRX:	0%	0%	17%	17%	23%	24%	0%	0%	24%
19%	26%	0%								
	Mcdma:	0%	0%	2%	2%	3%	3%	0%	0%	4%
3%	4%	0%								
	McdmaFlush:	0%	0%	21%	20%	28%	28%	0%	0%	28%
24%	30%	0%								
	Cipher:	32%	0%	0%	0%	0%	0%	0%	0%	0%
0%	0%	0%								
rx kbits/sec:	217296	297699	261605	268546	389380	397101	479195	528945	438931	
360583	454262	468575								
rx frames/sec:	39194	42022	40122	40086	58217	59507	80931	76094	67224	
54592	68565	67013								
tx kbits/sec:	211773	291616	267373	274308	397747	405655	467493	517944	448590	
368412	464116	458868								
tx frames/sec:	39182	41998	40122	40086	58217	59507	80895	76058	67224	
54592	68565	66973								

15-Min Avg: lcore01   lcore02   lcore03   lcore04   lcore05   lcore06   lcore07   lcore08   lcore09   lcore10   lcore11   lcore12										
	Idle:	22%	37%	60%	61%	45%	46%	64%	64%	44%
49%	41%	66%								
	PortRX:	33%	43%	0%	0%	0%	0%	36%	36%	0%
0%	0%	34%								
	PortTX:	18%	20%	0%	0%	0%	0%	0%	0%	0%
0%	0%	0%								
	KniRX:	4%	0%	0%	0%	0%	0%	0%	0%	0%
0%	0%	0%								
	McdmaRX:	0%	0%	16%	16%	24%	24%	0%	0%	24%
22%	25%	0%								
	Mcdma:	0%	0%	2%	2%	3%	3%	0%	0%	4%
4%	4%	0%								
	McdmaFlush:	0%	0%	21%	20%	28%	28%	0%	0%	28%
26%	30%	0%								
	Cipher:	23%	0%	0%	0%	0%	0%	0%	0%	0%
0%	0%	0%								
rx kbits/sec:	225682	297062	267631	272770	419821	405286	489633	511100	476834	

```

401810| 467134| 466549|
rx frames/sec: 39670| 42772| 40892| 40834| 62740| 61170| 83540| 76519| 72158|
60242| 70707| 67961|
tx kbits/sec: 220089| 290875| 273510| 278639| 428840| 414080| 477557| 500037| 487201|
410421| 477298| 456711|
tx frames/sec: 39657| 42748| 40892| 40834| 62740| 61170| 83504| 76484| 72158|
60242| 70707| 67925|

```

@ tick 896633 (+ve-skew-cnt=123633, -ve-skew-cnt=0), failed samples 0

Commande l'utilisation PPS de dinet de npumgr de #show, les bbps d'utilisation de dinet de npumgr de #show et la table d'utilisation de port de #show

Ils fournissent des informations sur le chargement sur les ports de DI, et les ports de services.

La performance réelle dépend de NIC's/CPU et d'affectation de temps processeur À l'iftask.

```
[local]UGP# show npumgr dinet utilization pps
```

```

----- Average DINet Port Utilization (in kpps) -----
Port   Type
      Current      5min      15min
      Rx      Tx      Rx      Tx      Rx      Tx
-----
1/0   Virtual Ethernet      0      0      0      0      0      0
2/0   Virtual Ethernet      0      0      0      0      0      0
3/0   Virtual Ethernet      0      0      0      0      0      0
4/0   Virtual Ethernet      0      0      0      0      0      0
5/0   Virtual Ethernet      0      0      0      0      0      0

```

```
[local]UGP# show npumgr dinet utilization bps
```

```

----- Average DINet Port Utilization (in mbps) -----
Port   Type
      Current      5min      15min
      Rx      Tx      Rx      Tx      Rx      Tx
-----
1/0   Virtual Ethernet      1      1      1      1      1      1
2/0   Virtual Ethernet      1      0      1      0      1      0
3/0   Virtual Ethernet      0      0      0      0      0      0
4/0   Virtual Ethernet      0      0      0      0      0      0
5/0   Virtual Ethernet      0      0      0      0      0      0

```

```
[local]UGP# show port utilization table
```

```

----- Average Port Utilization (in mbps) -----
Port   Type
      Current      5min      15min
      Rx      Tx      Rx      Tx      Rx      Tx
-----
1/1   Virtual Ethernet      0      0      0      0      0      0
2/1   Virtual Ethernet      0      0      0      0      0      0
3/10  Virtual Ethernet      0      0      0      0      0      0
3/11  Virtual Ethernet      0      0      0      0      0      0
4/10  Virtual Ethernet      0      0      0      0      0      0
4/11  Virtual Ethernet      0      0      0      0      0      0
5/10  Virtual Ethernet      0      0      0      0      0      0
5/11  Virtual Ethernet      0      0      0      0      0      0

```

Résumé de Di-réseau de moniteur de nuage de #show de commande

Cette commande surveille les santés du réseau de DI. Les cartes envoient des pulsations entre eux, et la perte est surveillée. Dans un système sain, aucune perte n'est signalée.

```
[local]UGP# show cloud monitor di-network summary
```

Card 3 Heartbeat Results:

```

ToCard  Health  5MinLoss  60MinLoss
  1      Good    0.00%    0.00%
  2      Good    0.00%    0.00%

```

```

4      Good      0.00%      0.00%
5      Good      0.00%      0.00%
Card 4 Heartbeat Results:
ToCard  Health   5MinLoss   60MinLoss
1      Good    0.00%     0.00%
2      Good    0.00%     0.00%
3      Good    0.00%     0.00%
5      Good    0.00%     0.00%

```

```

Card 5 Heartbeat Results:
ToCard  Health   5MinLoss   60MinLoss
1      Good    0.00%     0.00%
2      Good    0.00%     0.00%
3      Good    0.00%     0.00%
4      Good    0.00%     0.00%

```

### Stats d'iftask de #show de commande récapitulatifs

Avec des chargements plus élevés NPU, il pourrait être possible que le trafic soit obtenir abandonné.

Pour évaluer ceci, la sortie **récapitulative de stats d'iftask de #show** de commande peut être prise.

Remarque: Les ÉCARTS peuvent être différents de zéro.

Remarque: tous autres compteurs ne devraient pas incrémenter.

```

[local]UGP# show cloud monitor di-network summary
Card 3 Heartbeat Results:
ToCard  Health   5MinLoss   60MinLoss
1      Good    0.00%     0.00%
2      Good    0.00%     0.00%
4      Good    0.00%     0.00%
5      Good    0.00%     0.00%
Card 4 Heartbeat Results:
ToCard  Health   5MinLoss   60MinLoss
1      Good    0.00%     0.00%
2      Good    0.00%     0.00%
3      Good    0.00%     0.00%
5      Good    0.00%     0.00%
Card 5 Heartbeat Results:
ToCard  Health   5MinLoss   60MinLoss
1      Good    0.00%     0.00%
2      Good    0.00%     0.00%
3      Good    0.00%     0.00%
4      Good    0.00%     0.00%

```

## Bulkstats

Le schéma de Bulkstat est élaboré pour QPVC-DI relié à la performance à iftask/dinet. C'est utile pour surveiller le dinet, les ports de service, et l'utilisation de npu d'un point de vue de représentation/chargement :

```

[local]UGP# show cloud monitor di-network summary
Card 3 Heartbeat Results:
ToCard  Health   5MinLoss   60MinLoss
1      Good    0.00%     0.00%

```



2	Good	0.00%	0.00%
4	Good	0.00%	0.00%
5	Good	0.00%	0.00%

Card 4 Heartbeat Results:

ToCard	Health	5MinLoss	60MinLoss
1	Good	0.00%	0.00%
2	Good	0.00%	0.00%
3	Good	0.00%	0.00%
5	Good	0.00%	0.00%

Card 5 Heartbeat Results:

ToCard	Health	5MinLoss	60MinLoss
1	Good	0.00%	0.00%
2	Good	0.00%	0.00%
3	Good	0.00%	0.00%
4	Good	0.00%	0.00%