



Utilisation de GPU pour l'inspection et le traitement de trafic réseau

Hackfest Decade
Québec
3 novembre 2018

2018-11-02 KMS - Martin Dubois, ing. 1



La présentation

Je vais vous présenter un projet personnel

- Le **problème** à l'origine de l'idée
- L'**idée**
- Le **comment**
- Une des **limitations** rencontré et le **contournement**
- Les **résultats**
- Ce qui reste à faire

Et une petite récapitulation et une idée des applications possibles pour terminer

2018-11-02 KMS - Martin Dubois, ing. 2

Avant de débiter « Qui suis-je? »



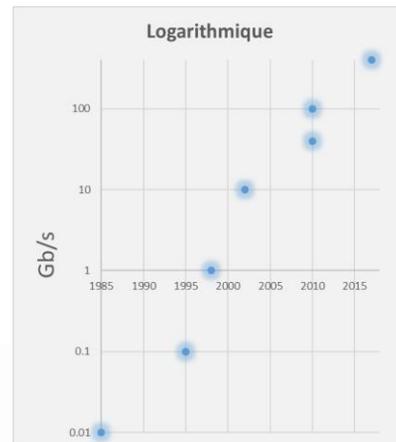
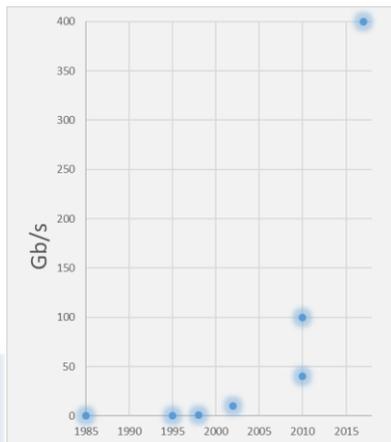
- Je ne suis pas un **expert en sécurité**
 - J'ai quand même un grand intérêt pour la sécurité
- Je me spécialise dans **l'intégration du logiciel et de l'électronique** et donc, dans le développement de **pilotes de périphériques**
- Je fais de la **consultation** et de la **formation** dans ce domaine

2018-11-02

KMS - Martin Dubois, ing.

3

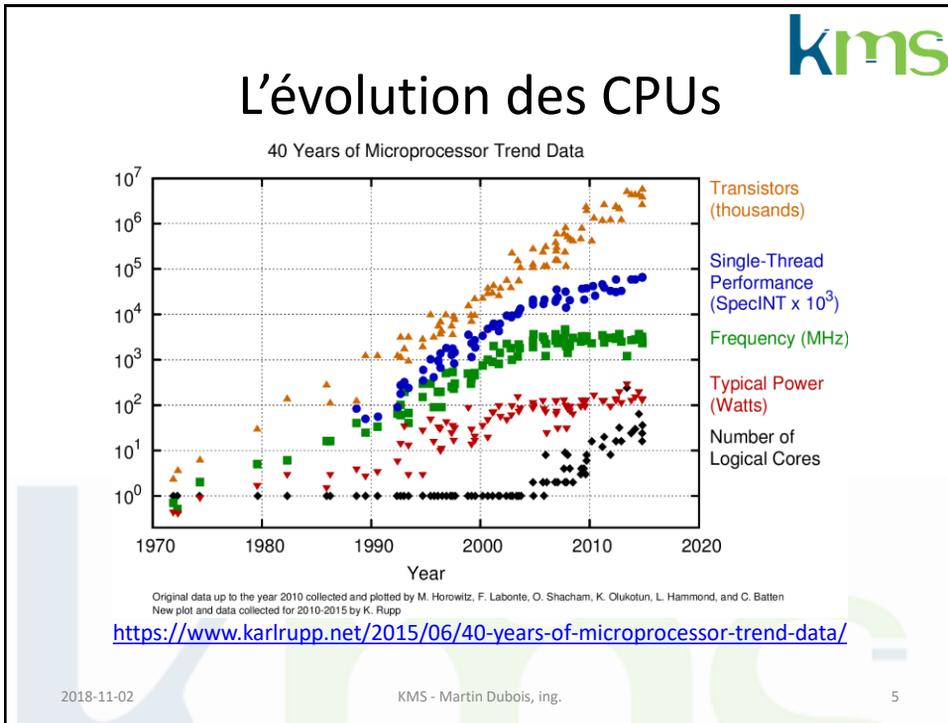
L'évolution d'Ethernet



2018-11-02

KMS - Martin Dubois, ing.

4



kms

Utilisation de GPU

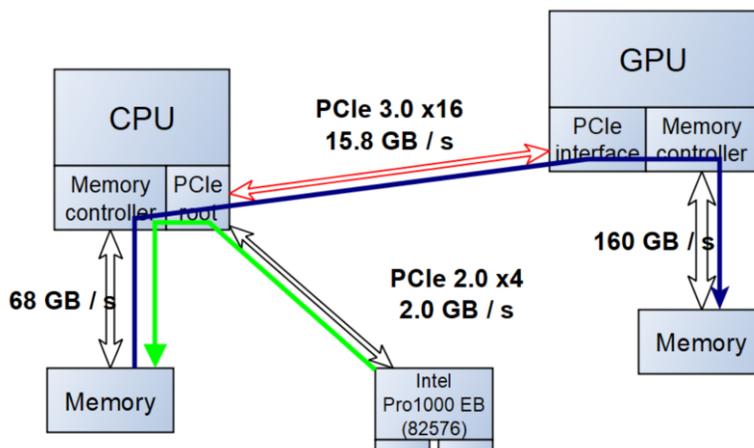
- L'utilisation d'un GPU semble tout appropriée
 - Le GPU offre une **grande puissance de calcul**
 - Le GPU offre une **grande rapidité d'accès à la mémoire**
 - L'inspection de paquets de données est **très facilement parallélisable**

2018-11-02 KMS - Martin Dubois, ing. 6

Le problème

- Les approches existantes utilisent
 - Le pilote de carte réseau fourni par le fabricant ou avec le système d'exploitation
 - La pile réseau du système d'exploitation
- Les données sont placées dans la **mémoire principale de l'ordinateur**

Double copie

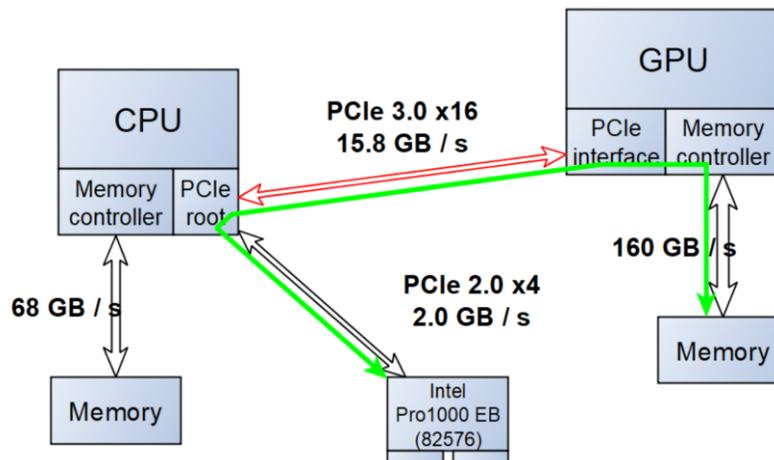


Double copie

- Le cout de cette **double copie** est important. Il faut
 - Attendre qu'un certain nombre de paquets soit reçu
 - Programmer le transfert des données
 - Attendre la fin du transfert
 - Lancer le traitement des données

Tout ça rend le traitement des paquets par le GPU beaucoup moins intéressant

Mon idée



Mon idée

- Pour cela, il faut
 - **Un pilote de carte réseau** qui permet de diriger les données vers la carte graphique
 - **Une carte graphique qui accepte de recevoir des données** d'un autre périphérique

DirectGMA vs GPUDirect



kms

Radeon™ Pro WX 5100

- Cout approximatif
475 \$ Can / 340 \$ US
- 8 GB DDR5 (256 bits)
– **160 GB / s**
- 28 Compute Units
– 64 Stream Processors / Compute Unit
- 1792 Stream Processor
- PCIe 3.0 x16 (**15.8 GB / s**)
- 75 W



A photograph of the Radeon Pro WX 5100 graphics card. It is a blue PCB with a single fan in the center. The text 'RADEON PRO WX 5100' is printed on the top left of the PCB, and a large blue 'R' logo is on the fan. The AMD logo is visible at the bottom of the PCB.

2018-11-02 KMS - Martin Dubois, ing. 13

kms

Radeon™ Pro WX 9100

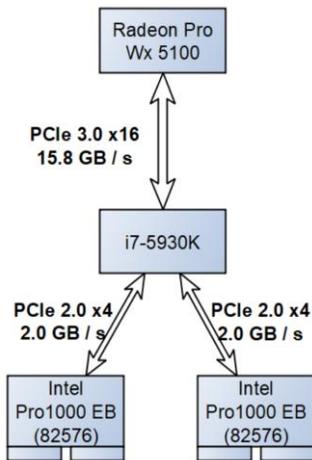
- Cout approximatif
2350 \$ Can / 1700 \$ US
- 16 GB HBM2 (2048 bits)
– **484 GB / s**
- 64 Compute Units
- 4096 Stream Processors
- PCIe 3.0 x16 (**15.8 GB / s**)
- 230 W



A photograph of the Radeon Pro WX 9100 graphics card. It is a blue PCB with a single fan on the right side. The text 'RADEON PRO WX 9100' is printed on the top left of the PCB, and a large blue 'R' logo is on the fan. The AMD logo is visible at the bottom of the PCB.

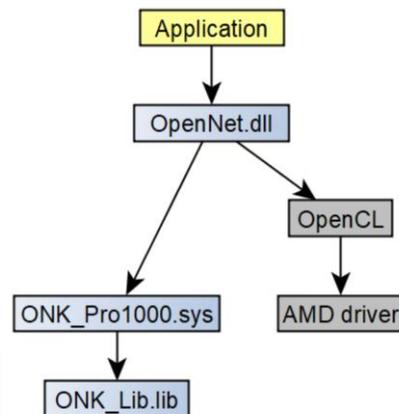
2018-11-02 KMS - Martin Dubois, ing. 14

Le système



Le logiciel

- C / C++
- OpenNet.dll
 - Environ 7300 lignes
- ONK_Pro1000.sys
 - Environ 2300 lignes
- ONK_Lib.lib
 - Environ 2200 lignes



Brevet en instance

Le code OpenCL

```
#include <OpenNetK/Kernel.h>

OPEN_NET_KERNEL_DECLARE
{
    OPEN_NET_KERNEL_BEGIN

        __global unsigned char * lData = lBase + lPacketInfo->mOffset_byte;

        for ( unsigned int i = 0; i < lPacketInfo->mSize_byte; i ++ )
        {
            // Ajoutez le code de traitement ici !
        }

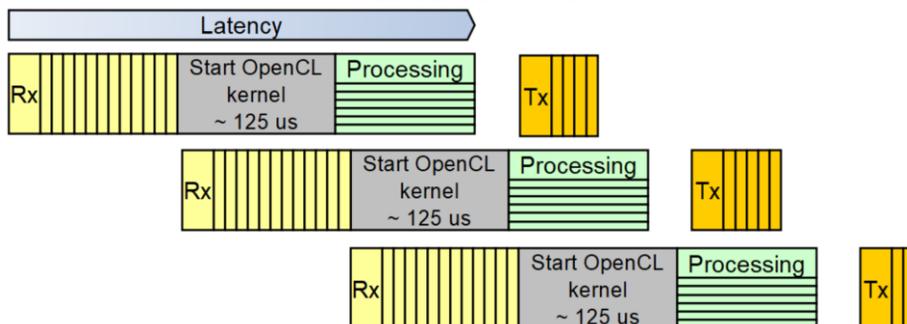
        lPacketInfo->mSendTo = OPEN_NET_PACKET_PROCESSED | (1 << 0);

    OPEN_NET_KERNEL_END
}

```

Indice de la carte
réseau qui doit
re-transmettre le
paquet.

La latence



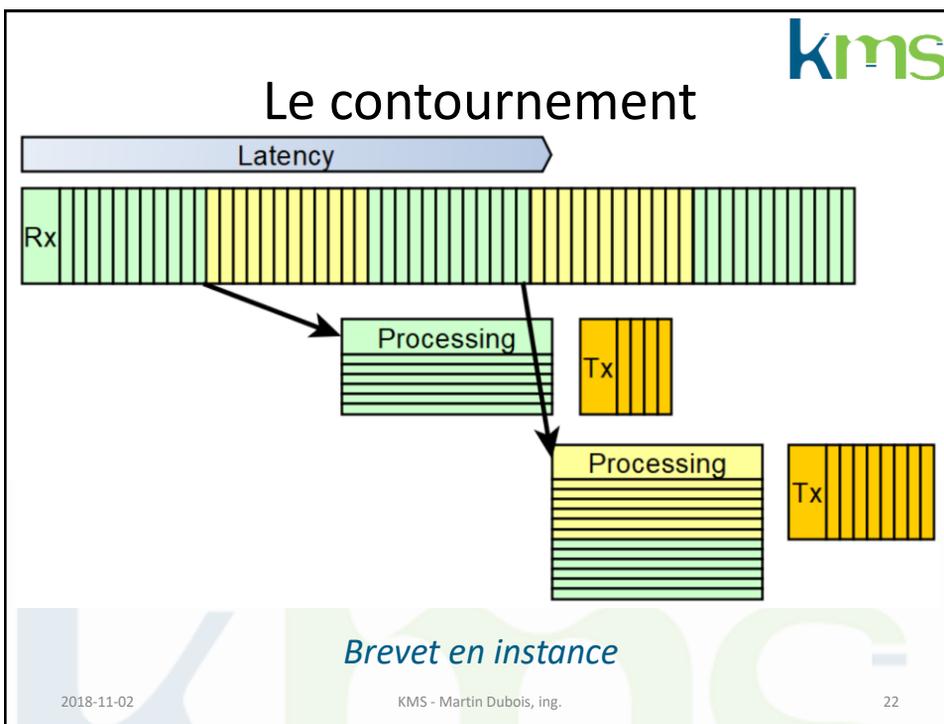
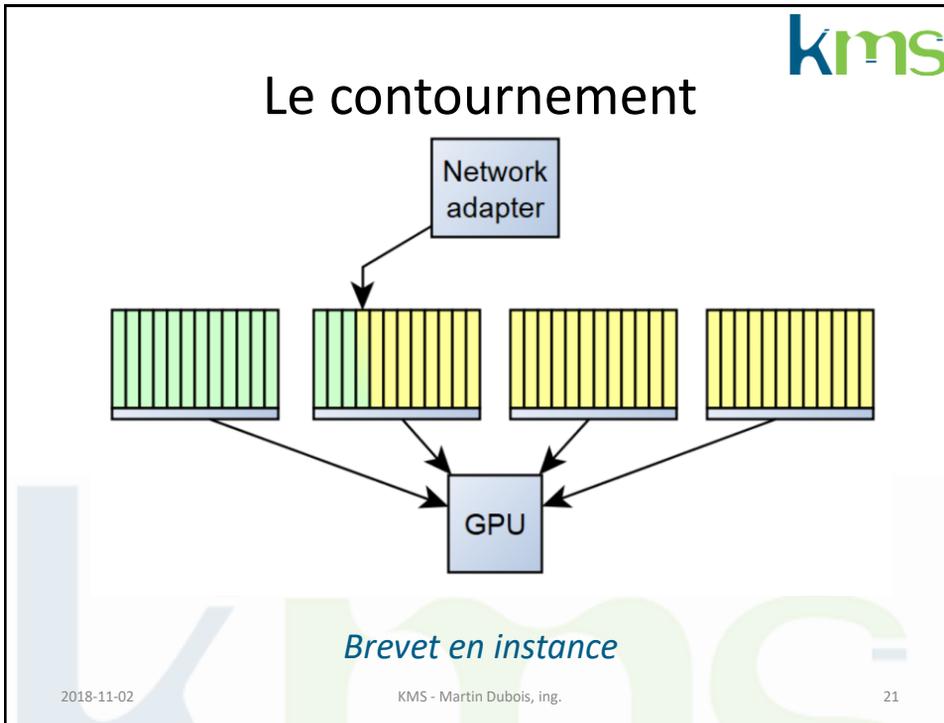
Brevet en instance

Limitation

- Si recevoir **64 paquets** prend moins de **~ 125 us**, ça ne fonctionne plus
- La limite est d'environ **512 000 paquets/s**
- Pour un lien **très rapide** ou qui **transmet principalement de petits paquets**, c'est limitatif
 - Cela augmente la latence
 - Cela réduit la quantité de données qu'il est possible de traiter par seconde

Plus de 64 paquets par tâche

- C'est possible
- Mais cela augmente la latence et l'augmente dans tous les cas
- Assez rapidement, doubler le nombre de paquets revient à doubler la latence



Résultats

Réception et traitement



- 1 Gb/s
 - 9000 octets/paquet
 - **119.0 Mio/s**
 - 64 octets/paquet
 - **91.6 Mio/s**
 - **1.41 million de paquets/s**



2018-11-02 KMS - Martin Dubois, ing. 23

Résultats

Réception et traitement



- 2 liens 1 Gb/s
 - 9000 octets/paquet
 - **238.0 Mio/s**
 - 128 octets/paquet
 - **205.4 Mio/s**
 - **1.63 million de paquets/s**
 - 64 octets/paquet
 - 101.9 Mio/s
 - 1.57 million de paquets/s



2018-11-02 KMS - Martin Dubois, ing. 24

Résultats



Réception, traitement et etransmission

- 1 Gb/s en réception - 1 Gb/s en transmission
 - 9000 octets/paquet
 - **119.0 Mio/s** (238.0 Mio/s au total)
 - 200 octets/paquet
 - **108.3 Mio/s** (216.6 Mio/s au total)
 - 64 octets/paquet
 - 48.9 Mio/s (97.8 Mio/s au total)
 - 754 000 paquets/s (1.57 million paquets/s au total)

TESTED



Estimation – Réception et traitement

10 Gb/s

- 9000 octets/paquet
 - **1.16 Gio/s**
- 768 octets/paquet
 - 1.13 Gio.s
 - 1.58 million de paquets/s

40 Gb/s

- 9000 octets/paquet
 - **4.64 Gio/s**
- 3072 octets/paquet
 - 4.62 Gio/s
 - 1.61 million de paquets/s

Démonstration

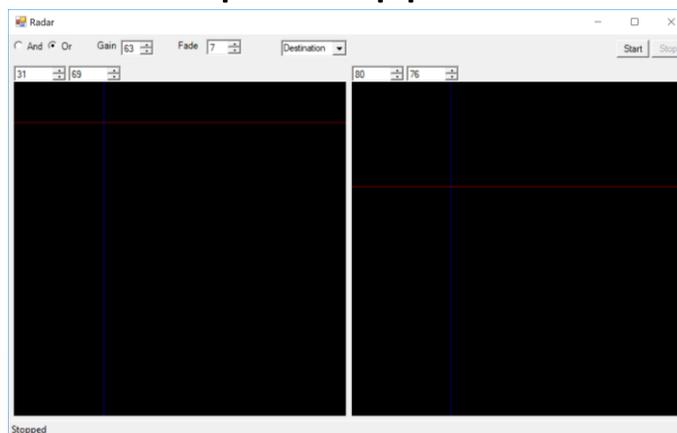


2018-11-02

KMS - Martin Dubois, ing.

27

Exemple d'application



- Le code de l'application est disponible à <http://www.kms-quebec.com/opennet.htm>

2018-11-02

KMS - Martin Dubois, ing.

28

Il reste encore des heures de plaisir



- Ce que j'ai présenté et une **preuve de concept**
- Il reste
 - De petites parties à **compléter**
 - Des **optimisations** à faire
 - Des **tests** à effectuer – autres cartes graphiques
 - D'autre **carte réseau** à supporter – 10 Gb/s
 - Créer d'autres **exemples**
 - **Développer des applications réelles...**

2018-11-02

KMS - Martin Dubois, ing.

29

En conclusion - Les possibilités



- **Analyser** le trafic réseau, à la **vitesse du câble**, avec toute la flexibilité qu'offre un langage de programmation
- **Modifier** les paquets, à la **vitesse du câble**, sans ajouter de délai
- **Décider de retransmettre, ou non**, les paquets analysés, et possiblement modifiés, **vers le système d'exploitation et/ou via une ou des cartes réseau spécifiques**

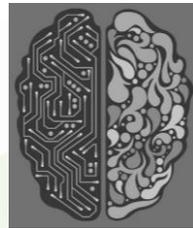
2018-11-02

KMS - Martin Dubois, ing.

30

Les applications en sécurité

- Analyser le trafic réseau
 - **Détecter des attaques ou des intrusions**
 - Rechercher des données dans **le contenu** des paquets
- **Classifier** des paquets et les **iguiller vers les outils d'analyse** appropriés
- **Pare-feu**
- Effectuer des **attaques!**
- Utiliser des **réseaux de neurones** (Intelligence artificielle) pour analyser le trafic réseau
- ...



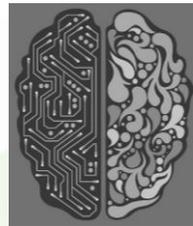
2018-11-02

KMS - Martin Dubois, ing.

31

Les autres applications

- **Balanceur** de charges
- **Routeur**
- Réception d'**images ou de vidéo** directement vers la carte graphique (GigE Vision, RTSP...)
- Transmission d'**images ou de vidéo** directement de la carte graphique
- Réception de données directement vers la carte graphique pour le **traitement** par un **algorithme** ou un **réseau de neurones** (Intelligence artificielle)
- ...



2018-11-02

KMS - Martin Dubois, ing.

32



Des questions ?

Martin Dubois, ing.

mdubois@kms-quebec.com

www.kms-quebec.com