Technologies et exemples d'architectures

Infrastructure Datacenters

Stéphane Mathieu

17 avril 2018

PUB !

- La liste des stages au CEA `http://www.cea.fr/emploi/Pages/stages/offres-stage.aspx`
- !TODO !

Bibliographie

Besta, M. and Hoefler, T. (2014). Slim fly : A cost effective low-diameter network topology. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '14, pages 348–359, Piscataway, NJ, USA. IEEE Press.

Dally, W. and Towles, B. (2003). Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Examen et notations

- Un QCM portant sur les cours et les TP : IB, SDN, VXLAN
- Participation en cours
- Les supports de cours seront distribués

Plan du cours

Technologies réseau

- Rappels sur les enjeux
- Les différentes technologies

Infiniband

- Qu'est-ce que la technologie Infiniband ?
- Le modèle Infiniband
- La comparaison des modèles
- La gestion de la qualité de services
- Subnet Manager
- Quelques commandes utiles
- Quelques éléments matériels

Plan du cours

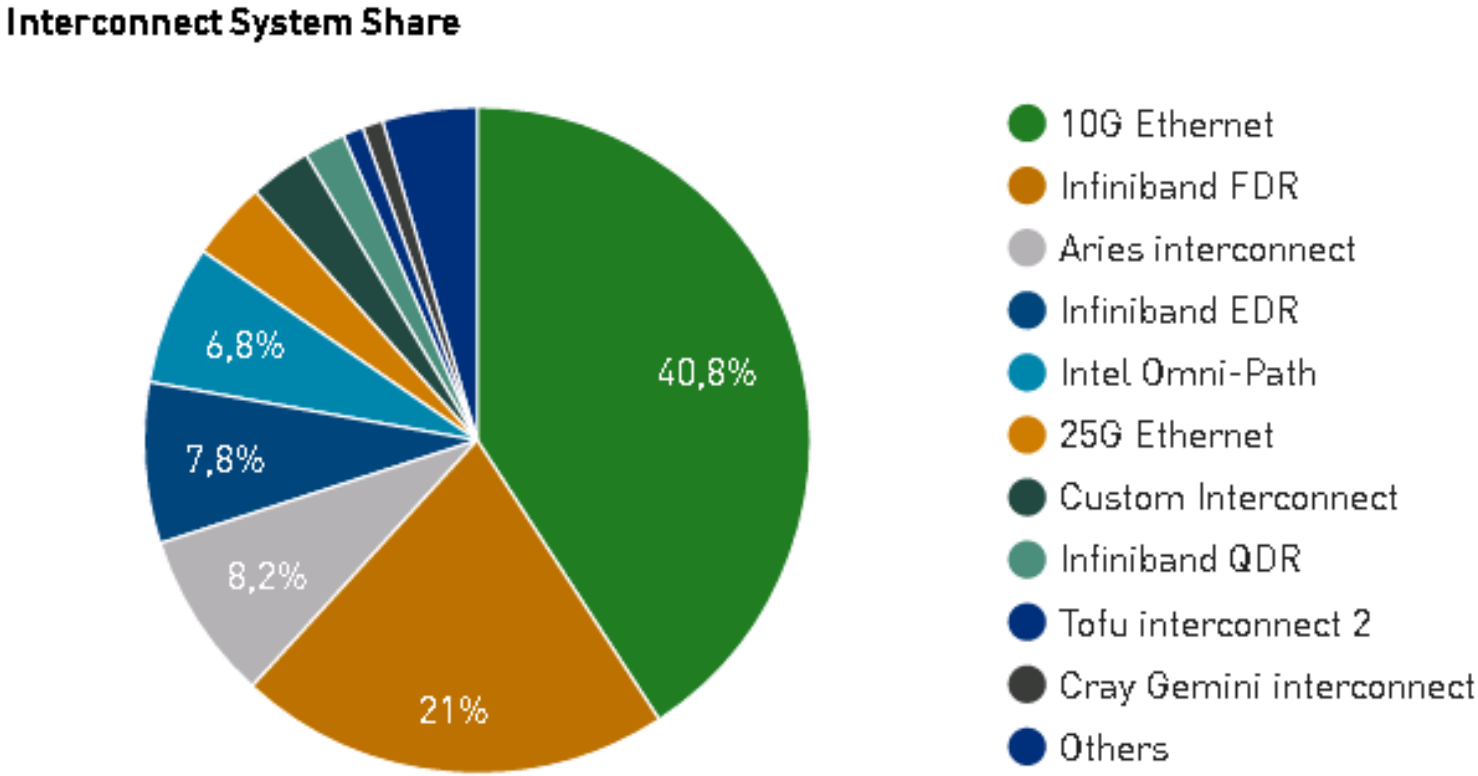
D'où vient la performance d'un calculateur ?

- Fréquence *brute* du processeur (*ie* : nombre d'opérations par seconde)
- Mémoire : taille de la mémoire, des caches ainsi que le temps d'accès ; les caches L1 et L2 sont d'accès plus rapides que la RAM
- Latence : temps d'acheminement d'un paquet au travers du réseau
- Réseau d'interconnexion :
  - Débit : volume de données acheminées aux terminaux du réseau
  - Topologie du réseau : le placement physique des éléments du réseau
  - Routage : les chemins empruntés (ou possibles) d'un paquet pour aller d'un nœud source au nœud destination
  - Placement des *jobs* : comment sont répartis les *jobs* au sein du cluster

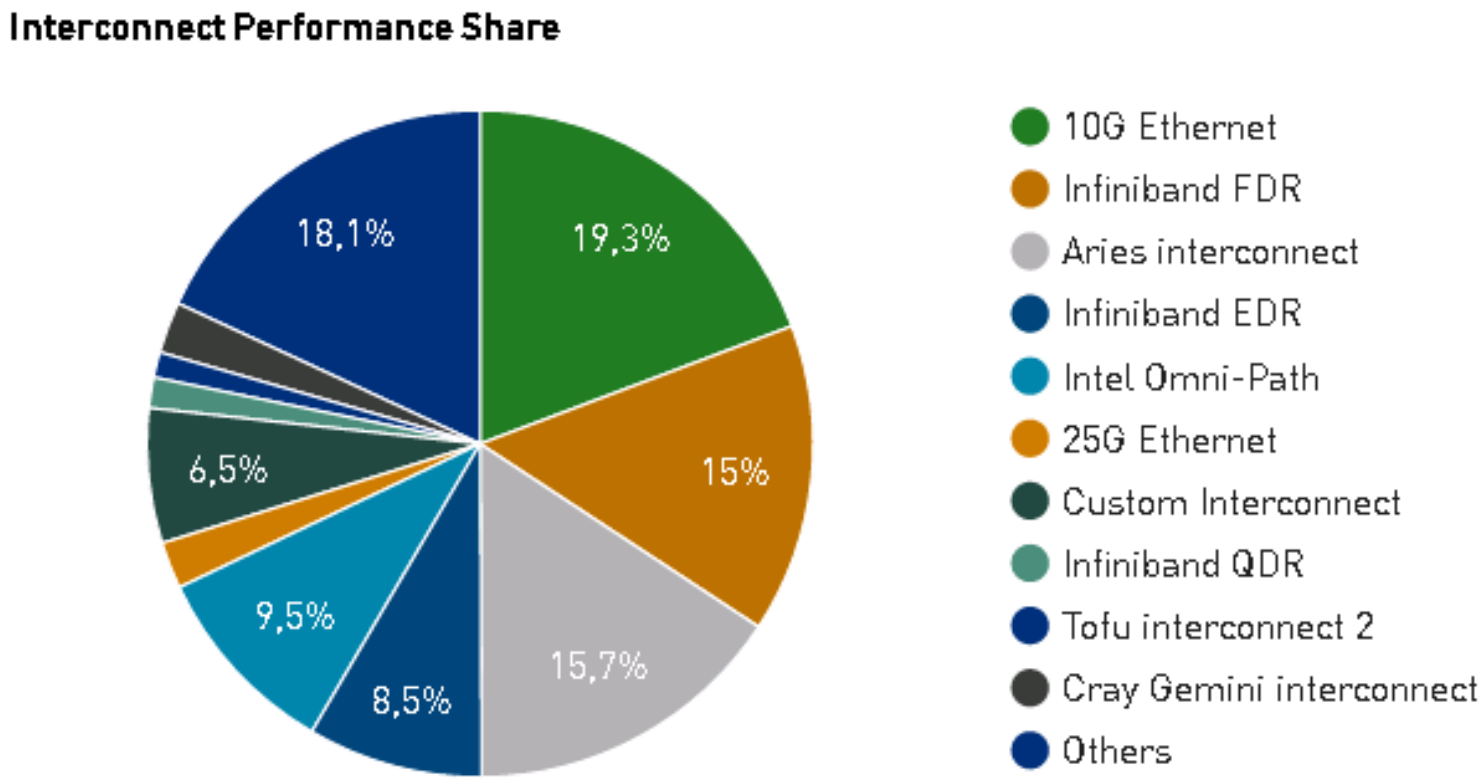
Les technologies

- Technologies historiques : Ethernet, GigaEthernet et 10-Giga Ethernet
- Infiniband : Technologie développée par Mellanox, basée sur les *verbs*
- BXI : Technologie développée par Atos/Bull, basée sur *portals*
- OPA : Technologie développée par Intel, Intel Fabric suite

Répartition dans le top500



Répartition dans le top500



Répartition dans le top500

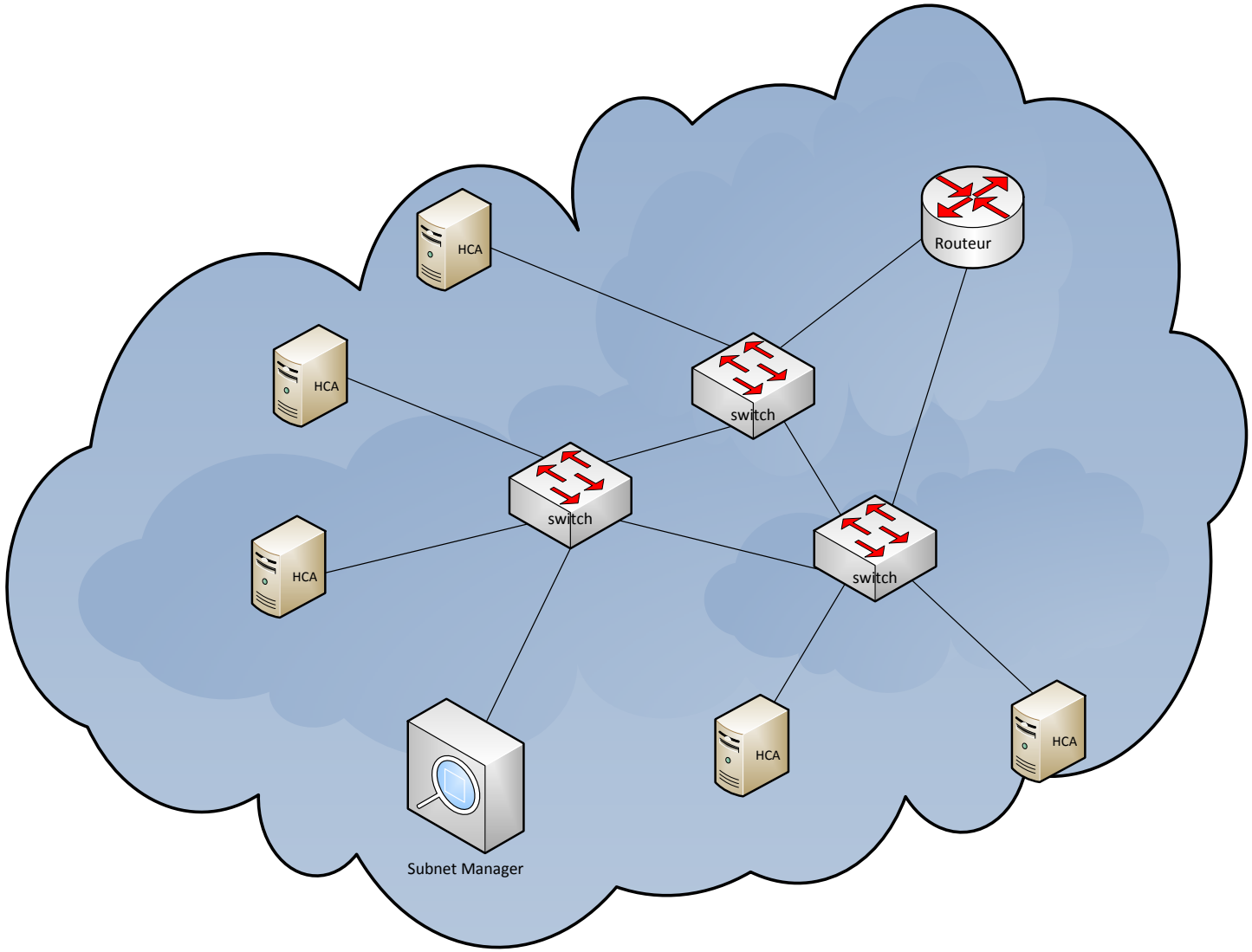
Interconnect	Count	System Share (%)	Rmax (GFlops)	Rpeak (GFlops)	Cores
10G Ethernet	204	40,8	162,743,886	330,318,127	9,093,016
Infiniband FDR	105	21	126,682,715	175,918,542	6,065,214
Aries interconnect	41	8,2	132,552,766	210,194,986	5,229,448
Infiniband EDR	39	7,8	71,809,664	106,145,823	23,658,670
Intel Omni-Path	34	6,8	80,169,289	131,841,742	2,660,588
25G Ethernet	19	3,8	19,174,400	36,294,400	439,616
Custom Interconnect	15	3	55,208,949	63,753,374	4,716,512
Infiniband QDR	10	2	12,519,011	19,719,038	790,816
Tofu interconnect 2	5	1	10,422,200	11,530,310	354,384
Cray Gemini interconnect	5	1	21,030,100	31,700,246	962,552
Infiniband	4	0,8	3,360,605	5,200,401	162,084
40G Ethernet	3	0,6	1,997,700	2,928,094	49,360
TH Express-2	2	0,4	35,934,090	57,976,934	3,294,720
Infiniband EDR/FDR	2	0,4	1,996,336	3,596,544	74,160
Proprietary	2	0,4	3,337,700	6,043,751	239,616
Bull BXI 1.2	2	0,4	5,915,109	10,942,669	244,256
Sunway	1	0,2	93,014,594	125,435,904	10,649,600
Gigabit Ethernet	1	0,2	557,340	1,028,352	49,440
Intel TrueScale Infiniband	1	0,2	596,010	681,574	32,768
100G Ethernet	1	0,2	613,200	920,000	25,000
Dell EMC H-Series (Omni-Path)	1	0,2	614,500	1,081,000	17,020
56G Infiniband FDR	1	0,2	1,013,721	1,372,134	32,984
Tofu interconnect	1	0,2	1,043,000	1,135,411	76,800
Infiniband FDR14	1	0,2	2,813,620	3,578,266	86,016

Plan du cours

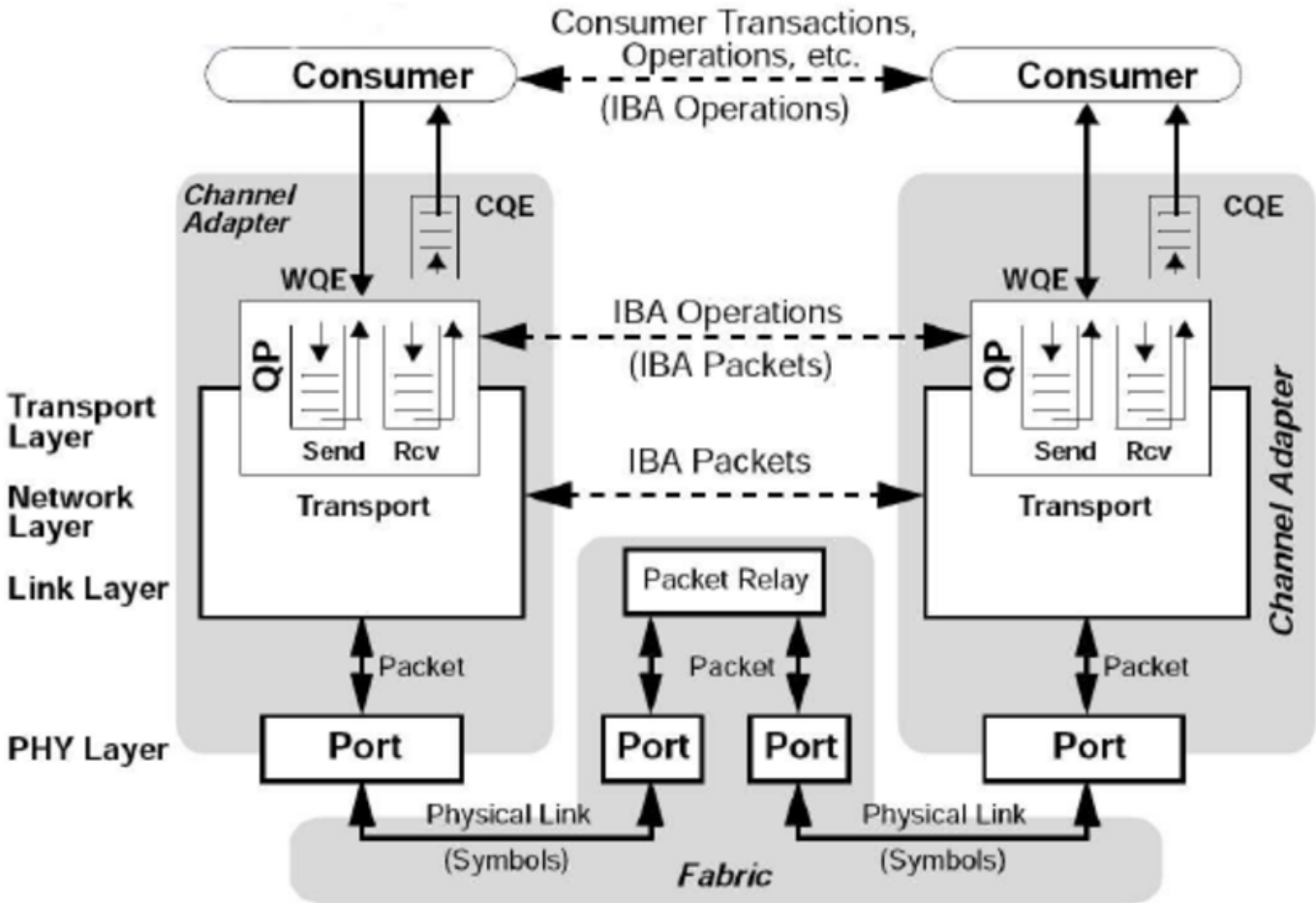
Qu'est-ce que InfiniBand ?

- InfiniBand Trade Association (IBTA), au début des années 2000
- Définit les spécifications depuis la couche matériel jusqu'à la couche applicatif
- Faible latence
- Importante bande passante
- Qualité de Service (QoS)
- Transport fiable (sans perte)
- CPU Offload
  - Protocole de transport basé sur le matériel
  - Kernel bypass
  - Remote Direct Memory Access

Réseau IB


HCA
HCA
HCA
HCA
HCA
Routeur
switch
switch
switch
Subnet Manager

Stack IB


Consumer Transactions,
Operations, etc.
(IBA Operations)
Consumer
Consumer
Channel Adapter
CQE
WQE
QP
Send
Rcv
Transport
IBA Operations
(IBA Packets)
IBA Packets
Transport Layer
Network Layer
Link Layer
PHY Layer
Packet
Packet Relay
Port
Physical Link
(Symbols)
Fabric
Channel Adapter

Définitions

- Work Queue : Permet à un consommateur de définir un ensemble d'instructions qui seront exécutées par le HCA
- Queue Pair : Entité adressable constituée de deux éléments : Send Queue et Receive Queue
- Send Queue : Buffer d'envoie
- Receive Queue : Buffer de réception
- Completion Queue : Buffer de complétion des requêtes

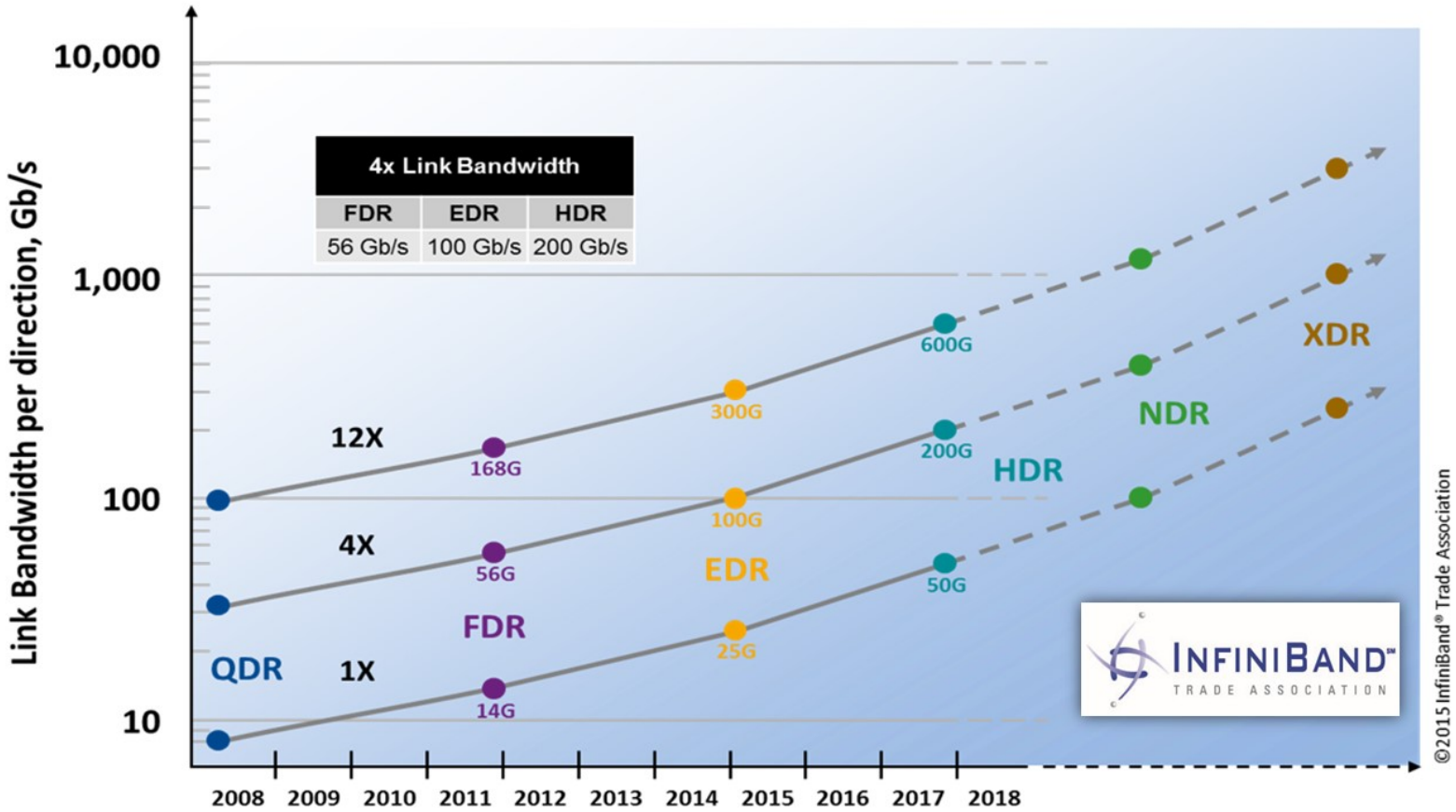
Adressage

- Adressage logique (16 bits au sein d'un même sous-réseau) donné par le *subnet manager*
  - 0x0001 - 0xBFFE : unicast
  - 0xC000 - 0xFFFE : multicast
  - 0x0000, 0xFFFF : réservés
- Le routage entre différents réseaux IB se fait avec un Global IDentifier(GID), de 128 bits, basés sur le modèle d'IPv6
- Un GUID de 64 bits pour les éléments du réseau : HCA, Switch, router et est utilisé pour la définition des LID

Comparaison des technologies

Largeur	SDR	DDR	QDR	FDR10	FDR	EDR	HDR
1x	2.5	5	10	10	14	25	50
2x	5	10	20	20	28	50	100
4x	10	20	40	40	56	100	200
12x	30	60	120	120	168	300	600

La bande passante InfiniBand



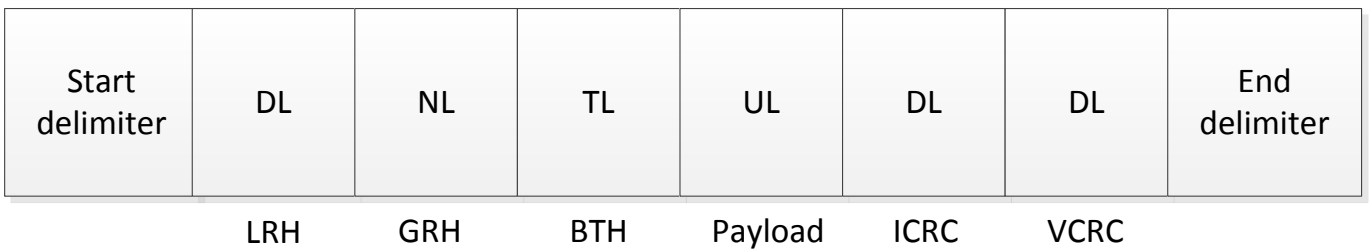
http://www.infinibandta.org/content/pages.php?pg=technology_overview

Infiniband

Le modèle IB se décompose en 5 couches distinctes :

- Physical Layer
- (Data) Link Layer
- Network Layer
- Transport Layer
- Upper Layer

Packet InfiniBand



Infiniband

- Physical Layer
  - Établir le lien physique
  - Garantir l'intégrité des données (End-to-End reliability) : Bit Error Rate
  - Avoir un réseau LossLess
  - Surveiller l'état des liens
  - Informer le Link Layer de l'état du lien
- Différents encodages :
  - 8/10b
  - 64/66b

Infiniband

- Link Layer
  - Réaliser l'adressage
  - Gérer le contrôle des flux (Flow Cnntrol)
  - Niveau du switching
  - Possède un Global Uniq IDentifier (GUID) 64 bits (basé sur le modèle IPv6)
  - Possède un Local IDentifier (LID) (attribué par le Subnet Manager)
  - Paramètre le Local Routing Header (LRH) : routage au sein d'un même sous-réseau
- Le LRH permet de définir les éléments de la communication
  - Service Lane : gestion de la QoS (`control flow`)
  - Destination LID : LID de la machine destinatrice

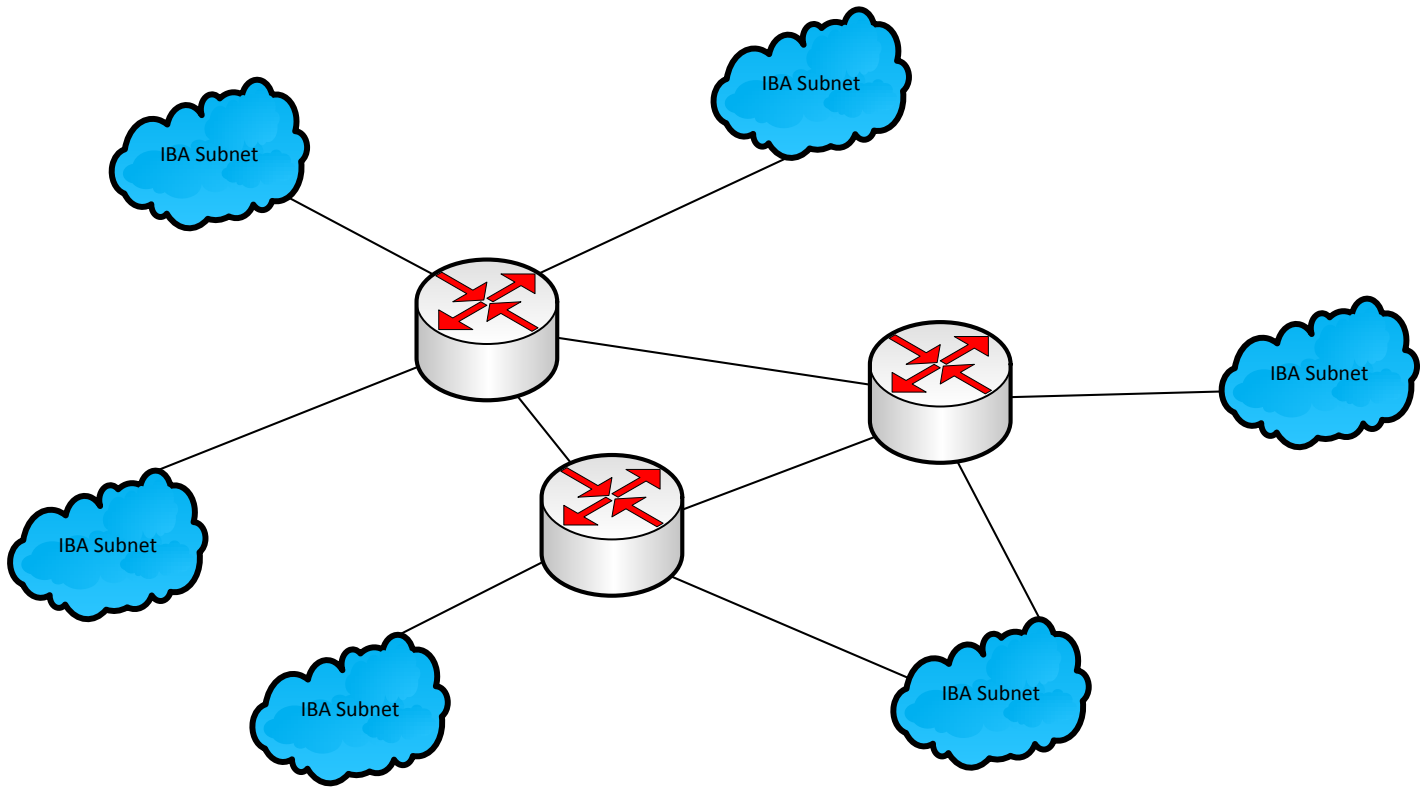
Infiniband

- Network Layer
  - Paramètre le Global Routing Header (GRH) : routage entre plusieurs sous-réseaux

Infiniband

bits bytes	31-24	23-16	15-8	7-0
0-3	IPVer / TClass / FlowLabel			
4-7	PayLen		NxtHdr	HopLmt
8-11	SGID[127-96]			
12-15	SGID[95-64]			
16-19	SGID[63-32]			
20-23	SGID[31-0]			
24-27	DGID[127-96]			
28-31	DGID[95-64]			
32-35	DGID[63-32]			
36-39	DGID[31-0]			

Placement des routeurs sur une interconnexion IB



Infiniband

- Transport Layer
  - Définit/régule le comportement des Queues Pairs (QP)
  - Segmentation des packets : MTU par défaut de 4096 bits.
  - Définit la classe de la connexion :
    - orienté connexion VS datagramme
    - fiable VS non-fiable

Comparaison des modèles de connexion

Attribute		RC and XRC	Reliable Datagram	Unreliable Datagram	Unreliable Connection	Raw Datagram (both IPv6 & ethertype)
Scalability (M processes on N Processor nodes communicating with all processes on all nodes)		**RC: M^2*N** QPs required on each processor node, per CA **XRC: M * N** QPs required on each processor node, per CA.	**M** QPs required on each processor node, per CA.	**M** QPs required on each processor node, per CA.	**M^2*N** QPs required on each processor node, per CA.	Minimum **1** QP required on each end node, per CA.
Reliability	Corrupt data detected	Yes				
	Data delivery guarantee	Data delivered exactly once		No guarantees		
	Data order guaranteed	Yes, per connection	Yes, packets from any one source QP are ordered to multiple destination QPs.	No	Unordered and duplicate packets are detected.	No
	Data loss detected	Yes		No	Yes	No
	Error recovery	**Reliable**. Errors are detected at both the requestor and the responder. The requestor can transparently recover from errors (retransmission, alternate path, etc.) without any involvement of the client application. QP processing is halted only if the destination is inoperable or all fabric paths between the channel adapters have failed.		**Unreliable**. Packets with some types of errors may not be delivered. Neither source nor destination QPs are informed of dropped packets.	**Unreliable**. Packets with errors, including sequence errors, are detected and may be logged by the responder. The requestor is not informed.	**Unreliable**. Packets with errors are not delivered. The requestor and responder are not informed of dropped packets.

Comparaison des modèles de connexion

RDMA and ATOMIC Operations	Yes	Yes	No	Yes: RDMA WRITEs No: RDMA READs & ATOMICs	No
Bind Memory Window	Yes	Yes	No	Yes	No
IBA Unreliable Multicast Support	No	No	Yes	No	No
Raw Multicast	No	No	No	No	Yes
Remote Invalidation	Yes	No	No	No	No
Shared Receive Queue	Yes	No	Yes	No	No

Comparaison des modèles de connexion

Attribute	RC and XRC	Reliable Datagram	Unreliable Datagram	Unreliable Connection	Raw Datagram (both IPv6 & ethertype)
Message Size	Transport supports a message size of zero to 2^{31} bytes. Implementations may support a smaller maximum message size. Actual maximum message size to be used may be negotiated by upper (application) layers. A message may consist of multiple packets.		**Single PMTU packet** datagrams - 0 to 4096 bytes.	Transport supports a message size of zero to 2^{31} bytes. Implementations may support a smaller maximum message size. Actual maximum message size to be used may be negotiated by upper (application) layers. A message may consist of multiple packets.	**Single PMTU packet** datagrams - 0 to 4096 bytes.
Connection Oriented?	**Connected**. The client connects the local QP to one and only one remote QP. No other traffic flows over these QPs.	**Connectionless**. Appears connectionless to the client - uses one or more End-to-End contexts per CA to provide reliability service.	**Connectionless**. No prior connection is needed for communication.	**Connected**. The client connects the local QP to one and only one remote QP. No other traffic flows over these QPs.	**Connectionless**. No prior connection is needed for communication.

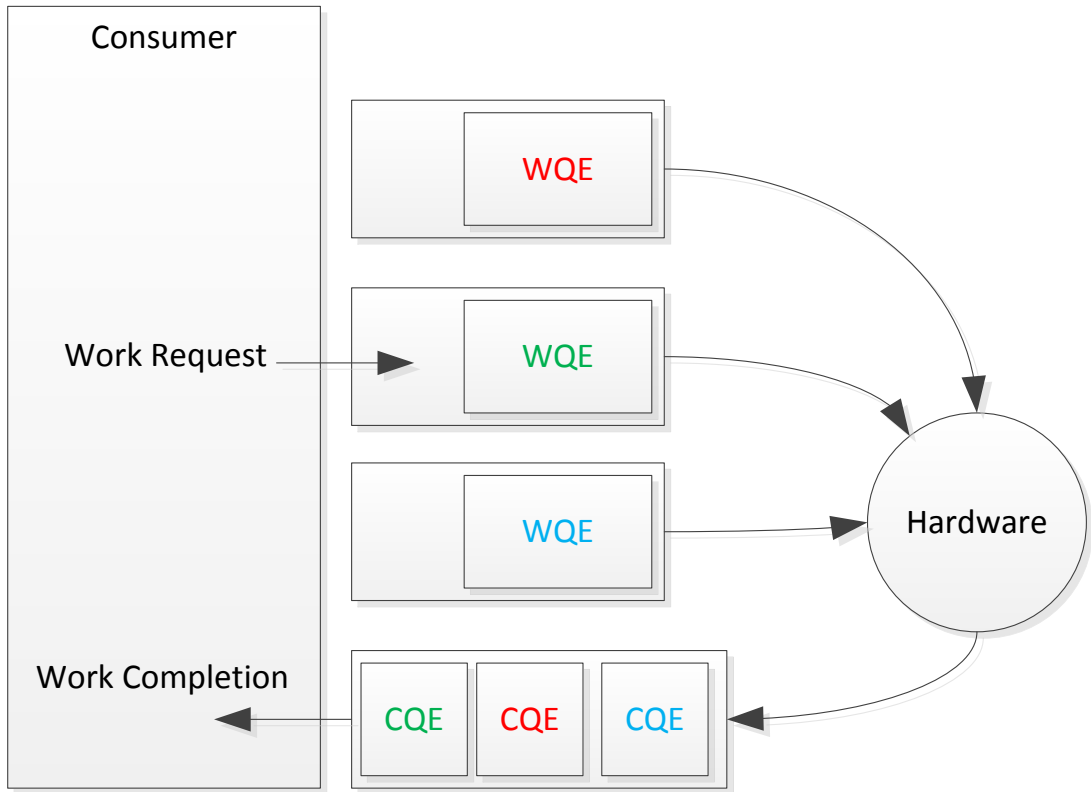
Infiniband

- Modification de la Maximum Transmission Unit
  - Grande MTU :
    - 4096 bits
    - Moins d'overhead,
    - Plus grande bande passante
    - Stockage, Virtual Protocol Interconnect
    - HPC
  - Petite MTU
    - 256 bits
    - Moins d'attente
    - Baisse de la latence globale
    - Consommation moindre en CPU
- La définition de la MTU a donc une place importante dans un réseau d'interconnexion

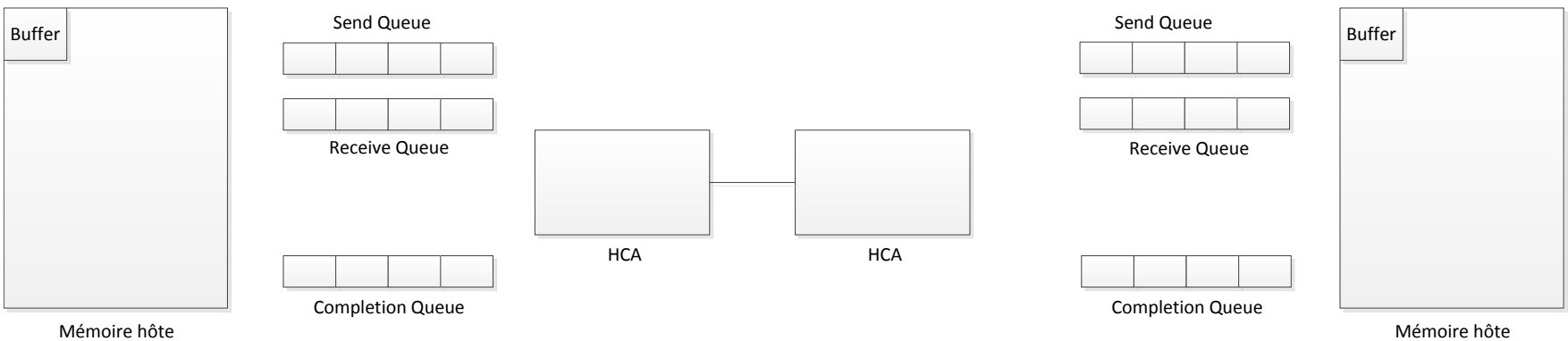
Qu'est-ce que les Queue Pair ?

- Canal de communication entre deux applications distantes
- Work Queue (WQE)
  - Send Queue
  - Receive Queue
- Completion Queue (CQE)
- Cet exemple concerne une utilisation standard du protocole infiniband : envoie d'un message entre deux HCA

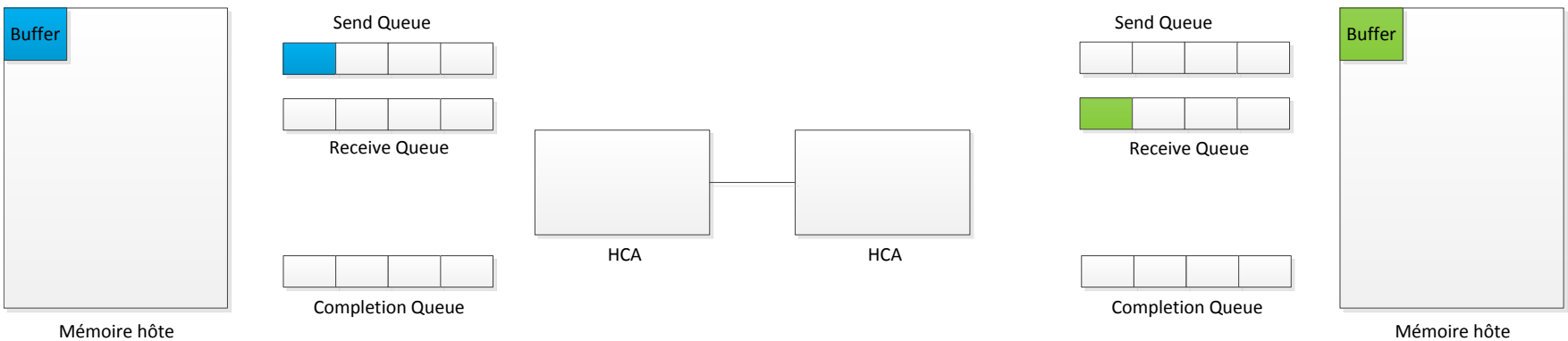
Queuing completion model



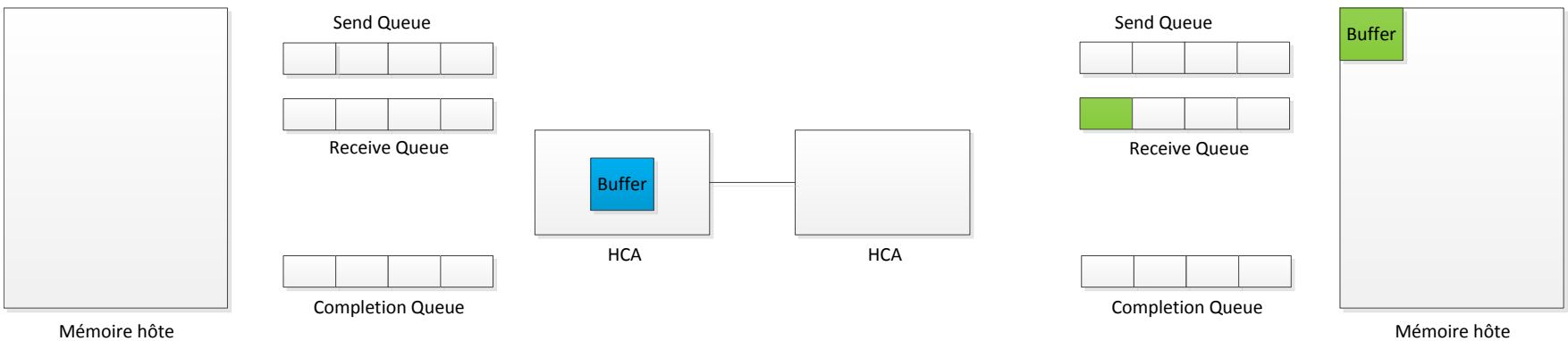
Queue Pair


Buffer
Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

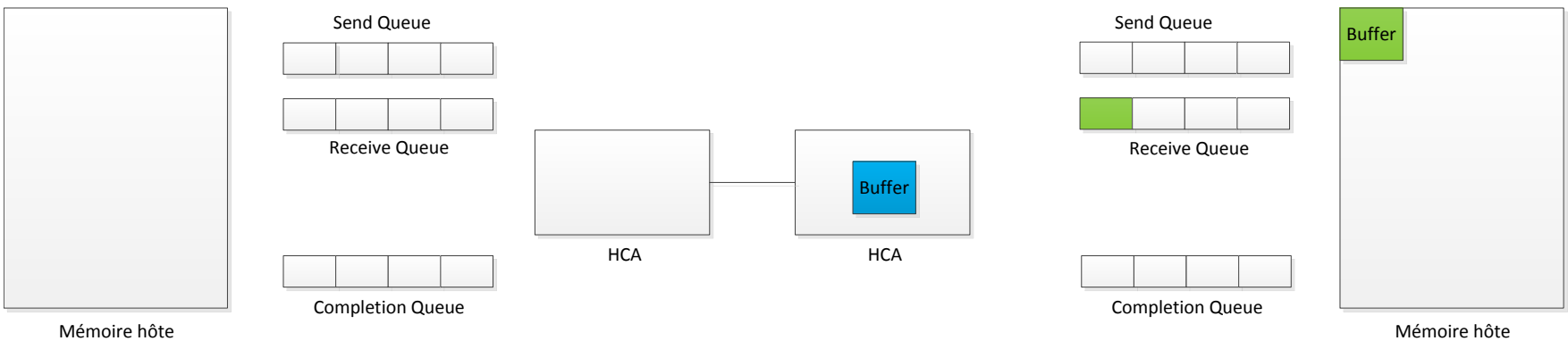
Queue Pair


Buffer
Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

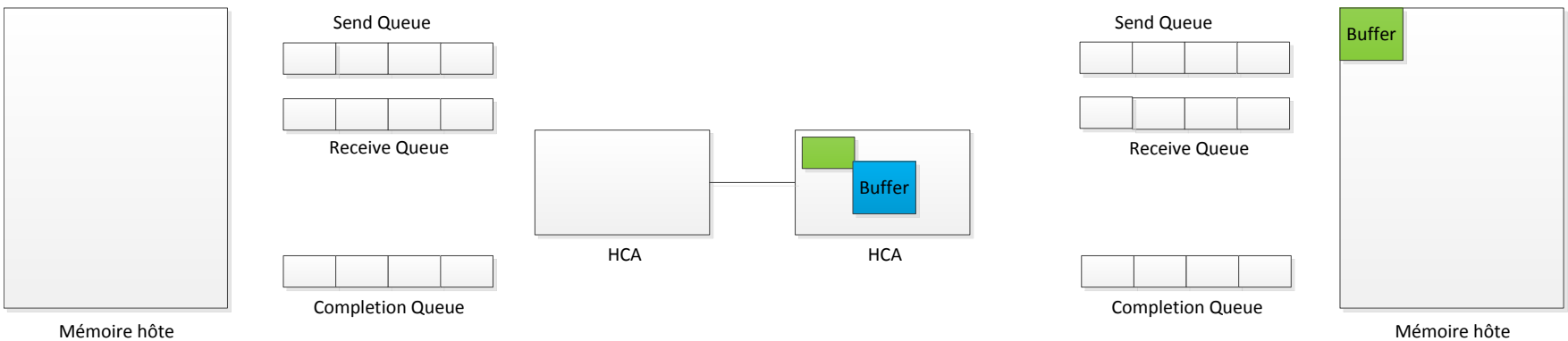
Queue Pair


Send Queue
Receive Queue
Completion Queue
Mémoire hôte
Buffer
HCA
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

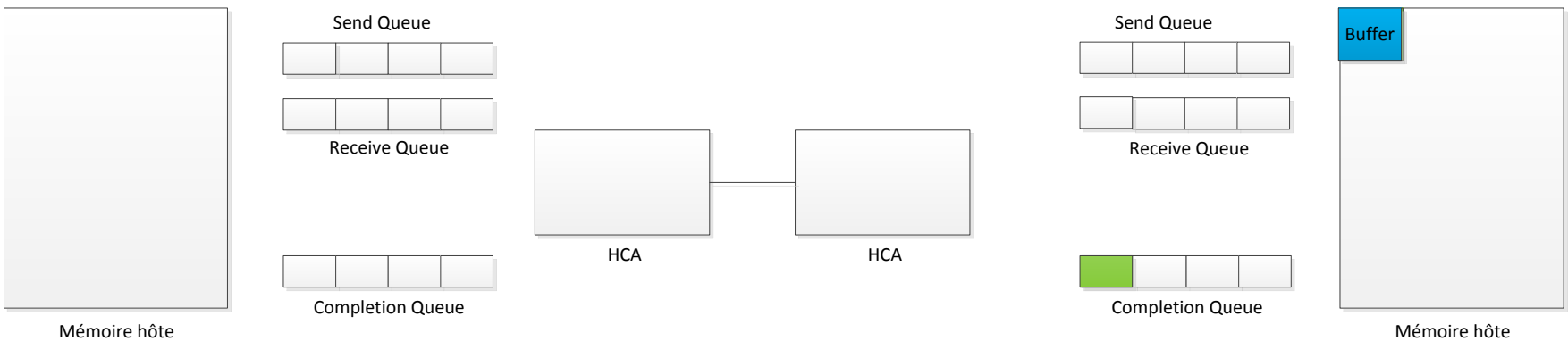
Queue Pair


Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
Buffer
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

Queue Pair


Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
Buffer
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

Queue Pair


Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

RDMA

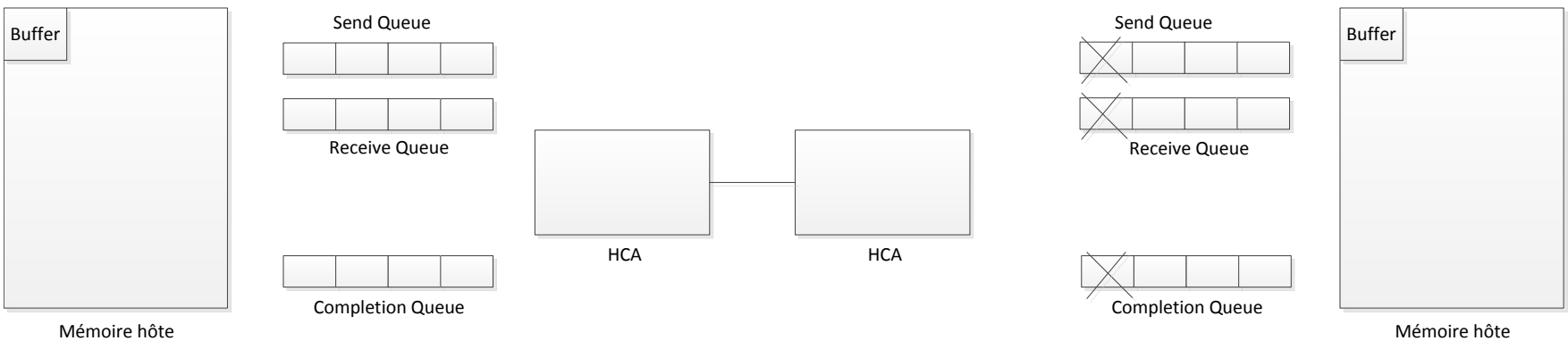
Remote Direct Memory Access

- Zero Copy
- Kernel Bypass
- Asynchronous operations
- En plus de la vitesse de lien fournie par IB, il est possible de le combiner à des technologies de bypass du noyau pour gagner encore en latence.
- RDMA définit 3 types d'opération spécifique
- L'exemple qui suit est un RDMA-WRITE

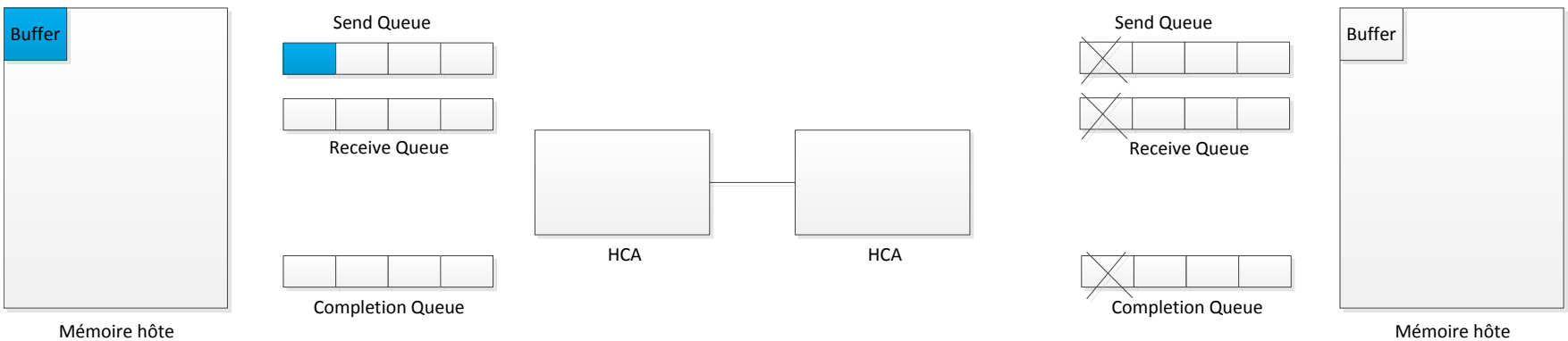
RDMA

- RDMA-READ
  - Lit la mémoire du nœud distant
- RDMA-WRITE
  - Ecrit dans la mémoire distante
- ATOMIC
  - Modification de la cellule mémoire directement dans la mémoire distante

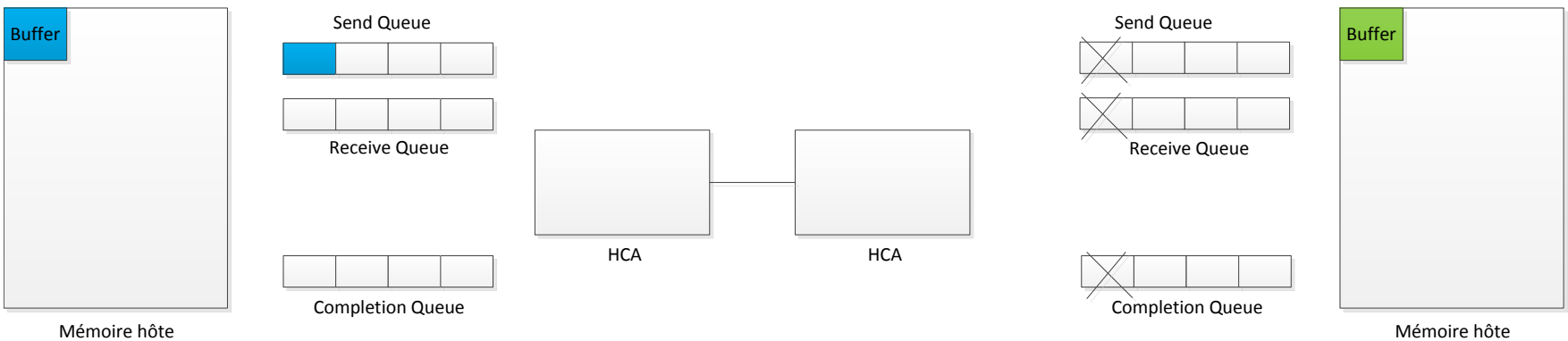
RDMA WRITE


Buffer
Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

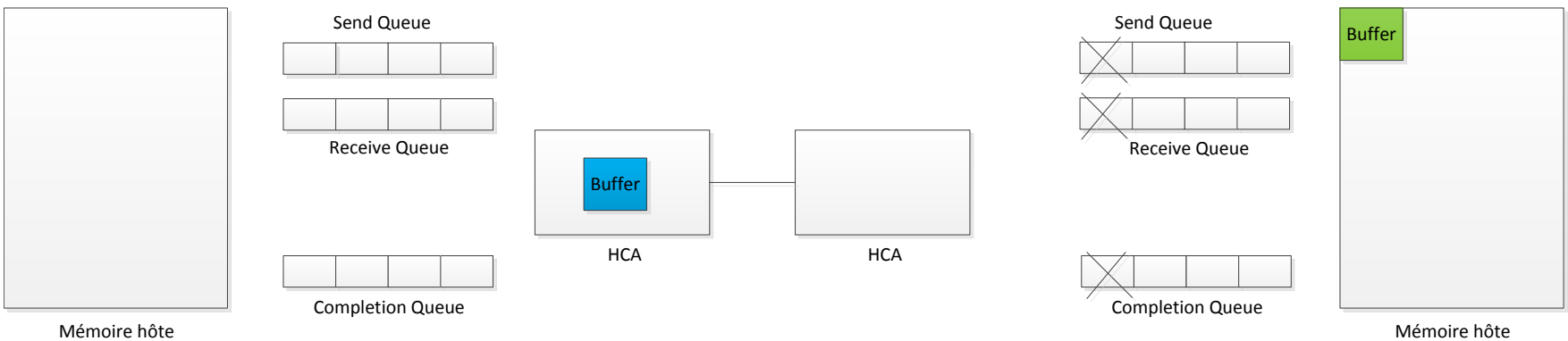
RDMA WRITE


Buffer
Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

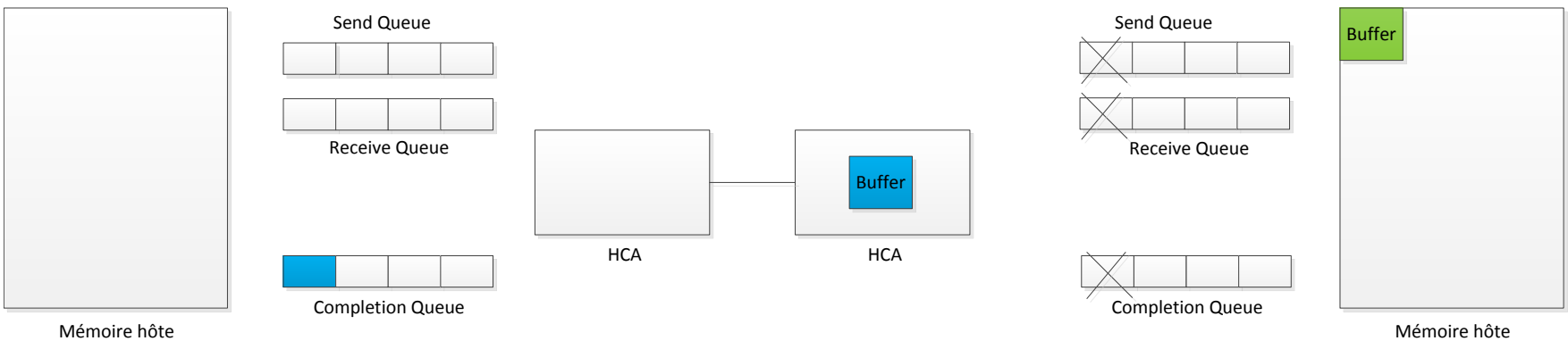
RDMA WRITE


Buffer
Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

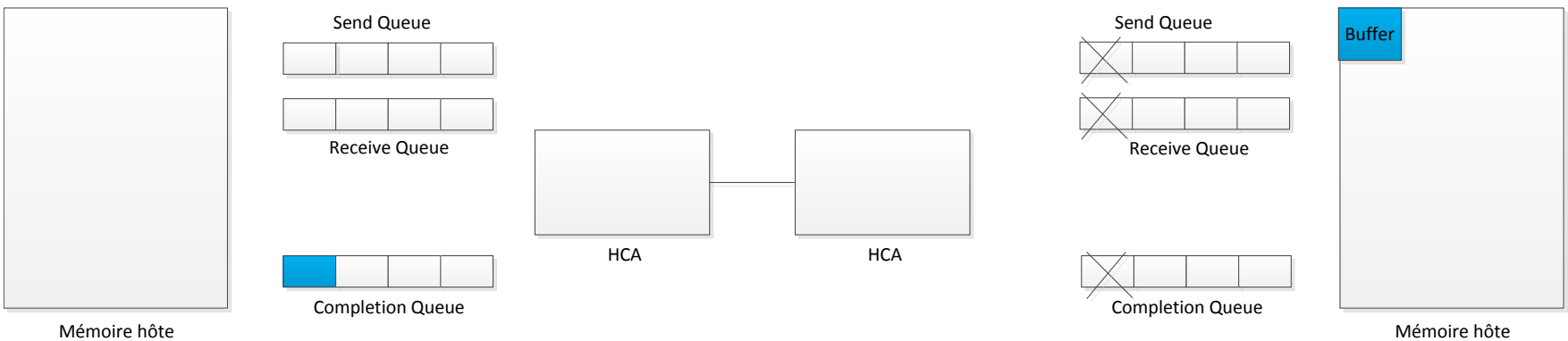
RDMA WRITE


Send Queue
Receive Queue
Completion Queue
Mémoire hôte
Buffer
HCA
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

RDMA WRITE


Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
Buffer
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

RDMA WRITE


Send Queue
Receive Queue
Completion Queue
Mémoire hôte
HCA
HCA
Send Queue
Receive Queue
Completion Queue
Buffer
Mémoire hôte

Infiniband

- Upper Layer
  - Niveau des API de communication
  - Support des protocoles de types TCP/IP
  - Management de la fabric
  - Support natif de RDMAoIB
  - Software Transport : verbs

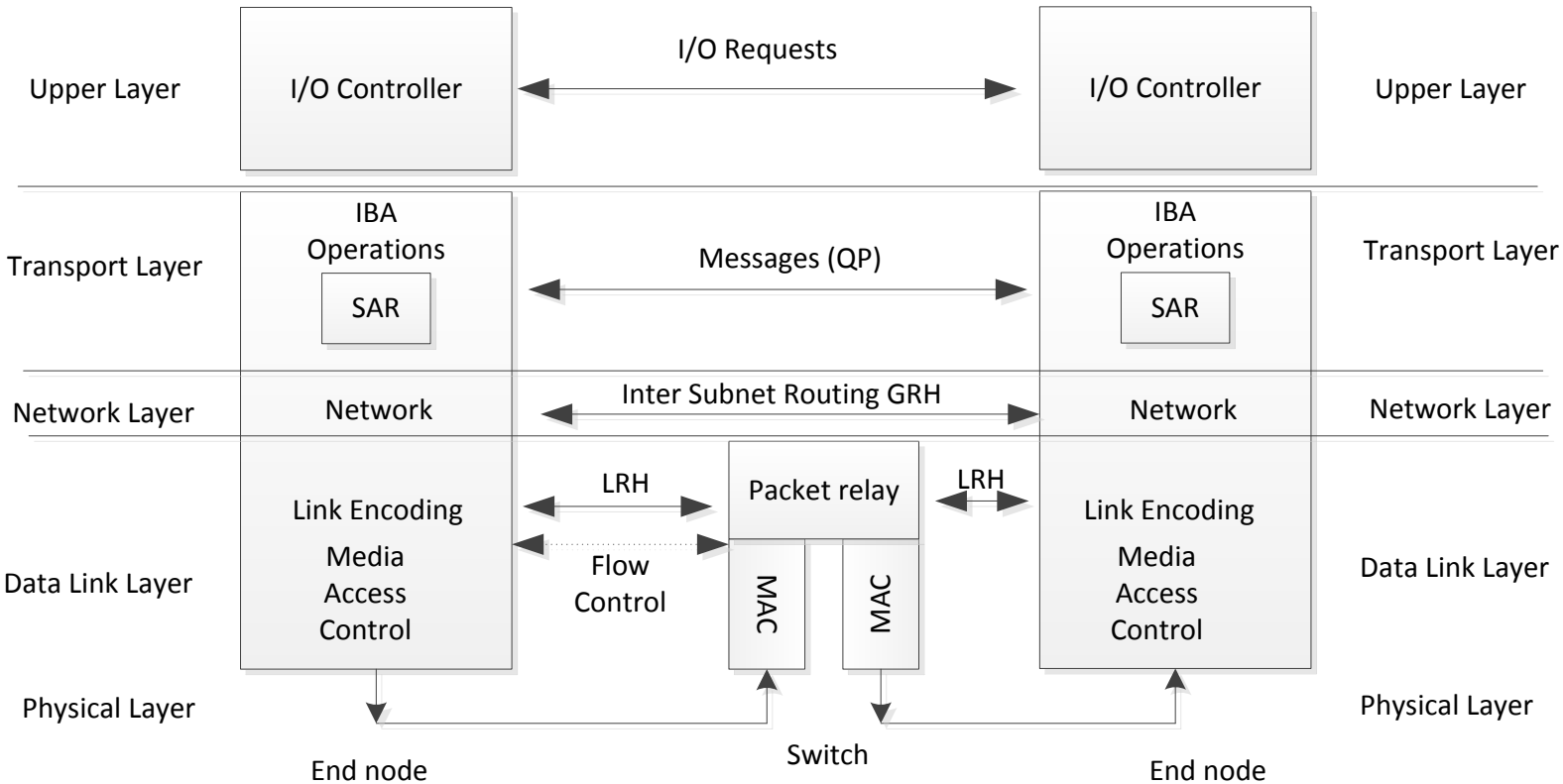
Infiniband

- Verbs
  - Interface entre la carte (HCA) et la fabric IB
  - Ce n'est pas une API
  - Interface de communication

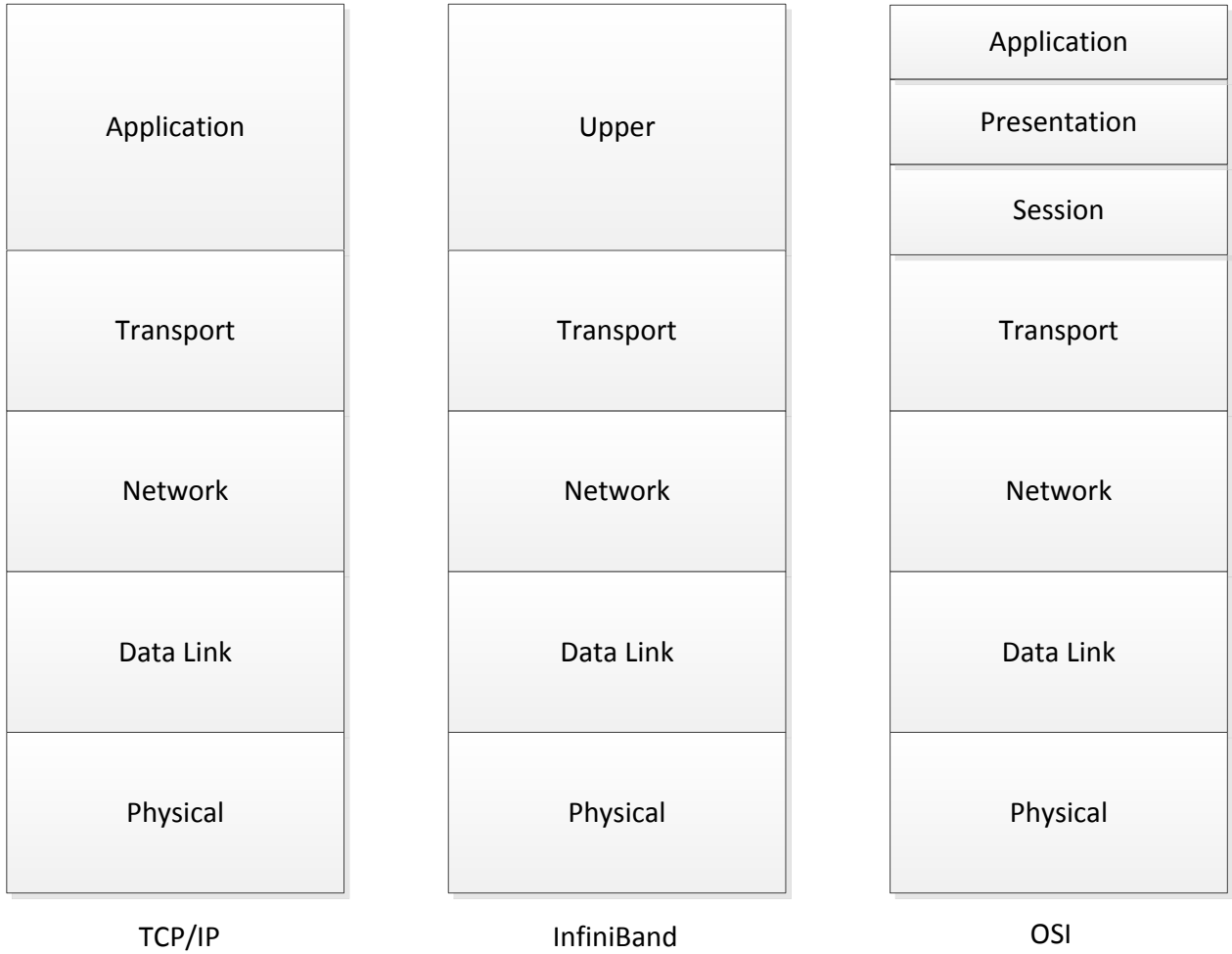
Infiniband

- Protocoles de la couche Upper Layer
  - MPI : Message Passing Interface, langage de programmation pour réaliser des calculs en parallèle
  - IPoIB : support du protocole IP transporté par une liaison IB
  - Socket Direct Protocol (SDP) : interface de programmation permettant de faire du RDMA
  - NFS RDMA
  - iSCSI : protocole de stockage basé sur IB

InfiniBand globale



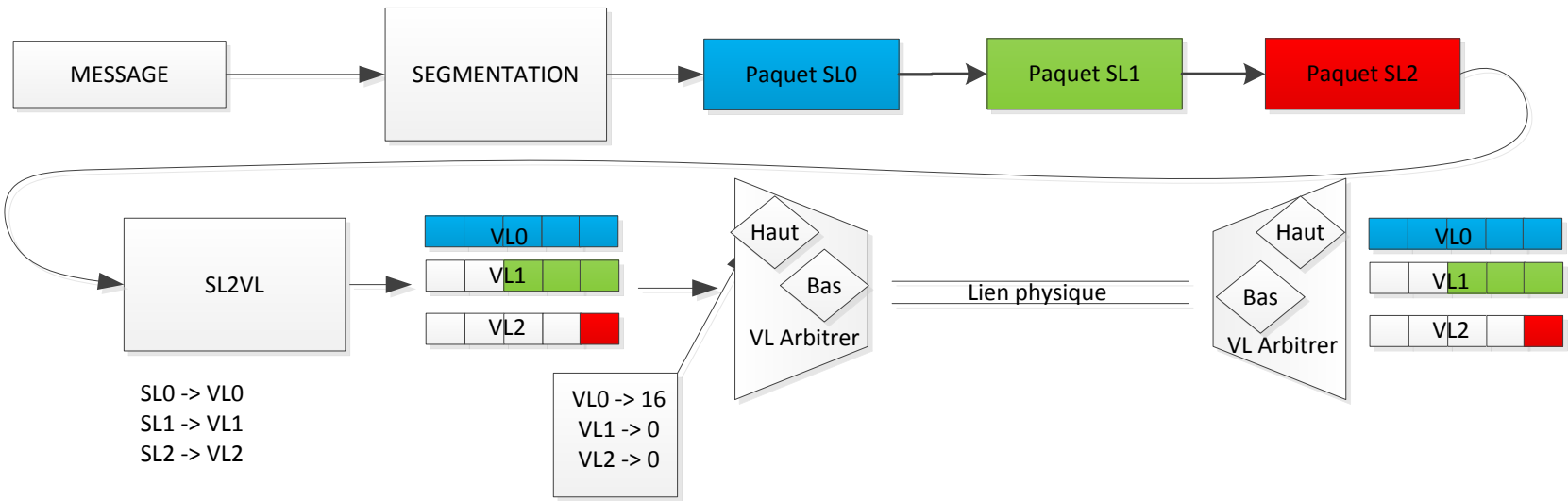
La comparaison des modèles



La QoS

- Deux concepts fondamentaux :
  - Service Lane : permet de caractériser le trafic : de 0 à 15
  - Virtual Lane : permet d'allouer des ressources au sein du réseau, allant de 0 à 15. Le VL 15 est réservé pour le management, les autres pour la données (1 obligatoire, 8 typique)
- Chaque service lane est mappée (=associée) dans une virtual lane (SL2VL)
- Plusieurs SL peuvent aller dans le même VL
- Chaque VL dispose de son contrôle de flux indépendant
  - Haut
  - Bas
- Les congestions sont contenues dans une VL

Ordonnancement des SL et VL



VL Arbitrer

- Chaque VL arbitrer dispose d'une pondération allant de 0 à 255.
  - 0 : désactivé
  - 255 : pas de décompte
  - 1-254 : utilisable pour gérer les flux
- High VL Arbitrer
  - Les flux sont strictement prioritaires par rapport au Low
  - Décompte des messages sur un compteur
- Low VL Arbitrer
  - Si on dispose d'un crédit et que la taille d'un paquet est comprise entre (64 et 4096 o) alors le paquet passe
  - 1 VL prendra l'intégralité de la bande passante

Routage

- Comme vu dans le précédent module, le subnet manager apporte la partie routage et fournit par conséquent les LID aux HCA/Switch
- OpenSM est un Subnet Manager
- MinHop
  - Pas de contrainte topologique
  - Étalement des routes sur les liens parallèles
  - Chemin le plus court
- UpDown
  - Contrainte de routage : une route monte puis descend
  - Étalement des routes sur les liens parallèles
  - Chemin le plus court
- Ftree
  - Contrainte topologique de Fat Tree
  - Étalement des routes sur des liens parallèles
  - Chemin le plus court

Partitions

- Partitions
  - Analogie avec les VLAN
  - Cloisonnement des ressources par fonction
  - PKeys sur 32 bits : 1 bit de port fort définissant le type de membership, 31 bits pour l'ID
  - Partitions par défaut : 0xFFFF full membership, 0x7FFF limited membership
- Membership
  - Définit les communications dans cette partitions
  - Full Membership : accès à toutes les ressources de la partition
  - Limited : accès restreint aux ressources "full membership"

partitions.conf

```
part1=0x0020, ipoib, mtu=5, sl=0 : ALL_SWITCHES=full, SELF=full;
part1=0x0020, ipoib, mtu=5, sl=0 : GUID_1=full,
        GUID_2=full, GUID_3=full;

part2=0x0010, ipoib, mtu=5, sl=0 : ALL_SWITCHES=full, SELF=full;
part2=0x0010, ipoib, mtu=5, sl=0 : GUID_3=full;
part2=0x0010, ipoib, mtu=5, sl=0 : GUiD_4=full,
                GUID_2=full, GUiD_5=full;
```

qos-policy.conf

```
qos-levels
     qos-level
          name: DEFAULT
          sl: 0
     end-qos-level
     qos-level
          name: QOS1
          sl: 1
          rate-limit: 16
          mtu-limit: 5
     end-qos-level
     qos-level
          name: QOS2
          sl: 11
          rate-limit: 16
          mtu-limit: 5
     end-qos-level
end-qos-levels
```

qos-policy.conf

```
port-groups
     port-group
          name: PORT_G1
          port-guid: GUIDs
     end-port-group
     port-group
          name: PORT_G2
          port-guid: GUIDs
     end-port-group
end-port-groups
qos-match-rules
     qos-match-rule
          source: PORT_G1
          destination: PORT_G2
          pkey: pkzy1
          qos-level-name: QOS1
     end-qos-match-rule
     qos-match-rule
          source: PORT_G3
          destination: PORT_G1
          pkey: peky2
          qos-level-name: QOS2
     end-qos-match-rule
end-qos-match-rules
```

smpquery : interrogation des LID

```
smpquery
Usage: smpquery [options] <op> <dest dr_path|lid|guid> [op params]
Supported ops (and aliases, case insensitive):
  NodeInfo (NI) <addr>
  NodeDesc (ND) <addr>
  PortInfo (PI) <addr> [<portnum>]
  PortInfoExtended (PIE) <addr> [<portnum>]
  SwitchInfo (SI) <addr>
  PKeyTable (PKeys) <addr> [<portnum>]
  SL2VLTable (SL2VL) <addr> [<portnum>]
  VLArbitration (VLArb) <addr> [<portnum>]
  GUIDInfo (GI) <addr>
  MlnxExtPortInfo (MEPI) <addr> [<portnum>]
```

smpquery : interrogation des LID

```
smpquery nodeinfo 73
# Node info: Lid 73
BaseVers:.....1
ClassVers:.....1
NodeType:.....Channel Adapter
NumPorts:.....1
SystemGuid:.....S_GUID
Guid:.....GUID
PortGuid:.....P_GUID
PartCap:.....1 2 8
DevId:.....0 x1013
Revision:.....0 x00000000
LocalPort:.....1
VendorId:.....0 x00000
```

smpquery : récupération des pkeys sur un LID

```
smpquery pkeys 73
    0: 0xffff 0x8020 0x8030 0x8010 0x8011 0x8012 0x8021 0x8022
    8: 0x8031 0x8070 0x8071 0x8072 0x8073 0x0000 0x0000 0x0000
   16: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   24: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   32: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   40: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   48: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   56: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   64: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   72: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   80: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   88: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
   96: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
  104: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
  112: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
  120: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
128 pkeys capacity for this port
```

ibstat : information sur la carte IB

```
ibstat
CA 'mlx5_0'
        CA type: MT4115
        Number of ports: 1
        Firmware version: 12.21.2010
        Hardware version: 0
        Node GUID:  N_GUID
        System image GUID: S_GUID
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 100
                Base lid: 73
                LMC: 0
                SM lid: 73
                Capability mask: 0x2651e84a
                Port GUID: P_GUID
                Link layer: InfiniBand
```

ibportstate : information sur les capacités du port

```
ibportstate 73 1
CA/RT PortInfo:
# Port info: Lid 73 port 1
LinkState:.....Active
PhysLinkState:.....LinkUp
Lid:.....73
SMLid:.....73
LMC:.....0
LinkWidthSupported:.....1X or 4X
LinkWidthEnabled:.....1X or 4X
LinkWidthActive:.....4X
LinkSpeedSupported:.....2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedEnabled:.....2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedActive:.....10.0 Gbps
LinkSpeedExtSupported:.....14.0625 Gbps or 25.78125 Gbps
LinkSpeedExtEnabled:.....14.0625 Gbps or 25.78125 Gbps
LinkSpeedExtActive:.....25.78125 Gbps
Mkey:.....<not displayed>
MkeyLeasePeriod:.....0
ProtectBits:.....0
# MLNX ext Port info: Lid 73 port 1
StateChangeEnable:.....0x00
LinkSpeedSupported:.....0x01
LinkSpeedEnabled:.....0x01
LinkSpeedActive:.....0x00
```

ibtracert : fournit le chemin entre 2 nœuds

```
ibtracert 73 75
From ca {GUID} portnum 1 lid 73-73 "toto1"
[1] -> switch port {GUID}[35] lid 68-68 "toto2"
[6] -> switch port {GUID}[6] lid 63-63 "toto3"
[2] -> switch port {GUID}[5] lid 62-62 "toto4"
[35] -> ca port {GUID}[1] lid 75-75 "toto5"
To ca {GUID} portnum 1 lid 75-75 "toto5"
```

ibping : test la connectivité entre 2 nœuds

Listing 1 – Sur le serveur

```
ibping -S
```

Listing 2 – Sur le client

```
ibping LID
```

Quelques autres commandes

- iblinkinfo : détermine l'état des liens de la fabric
- ibnetdiscover, ibnodes, ibswitches

saquery : information sur le nœud

```
saquery --help
Usage: saquery [options] [query-name] [<name> | <lid> | <guid>]
Supported query names (and aliases):
  ClassPortInfo (CPI)
  NodeRecord (NR) [lid]
  PortInfoRecord (PIR) [[lid]/[port]/[options]]
  SL2VLTableRecord (SL2VL) [[lid]/[in_port]/[out_port]]
  PKeyTableRecord (PKTR) [[lid]/[port]/[block]]
  VLArbitrationTableRecord (VLAR) [[lid]/[port]/[block]]
  InformInfoRecord (IIR) [subscriber_gid]
  LinkRecord (LR) [[from_lid]/[from_port]] [[to_lid]/[to_port]]
  ServiceRecord (SR)
  PathRecord (PR)
  MCMemberRecord (MCMR)
  LFTRecord (LFTR) [[lid]/[block]]
  MFTRecord (MFTR) [[mlid]/[position]/[block]]
  GUIDInfoRecord (GIR) [[lid]/[block]]
  SwitchInfoRecord (SWIR) [lid]
  SMInfoRecord (SMIR) [lid]
```

saquery

```
MCMember Record dump:
                MGID . . . . . ff12:601b:ffff::1:ff1e:cfb6
                PortGid . . . . . ::
                qkey . . . . . 0 xb1b
                mlid . . . . . 0 xc0ab
                mtu . . . . . 0 x85
                TClass . . . . . 0 x0
                pkey . . . . . 0 xffff
                rate . . . . . 0 x83
                pkt_life . . . . . 0 x80
                SL . . . . . 0 x0
                FlowLabel . . . . . 0 x0
                HopLimit . . . . . 0 x0
                Scope . . . . . 0 x0
                JoinState . . . . . 0 x0
                ProxyJoin . . . . . 0 x0
```

Modules nécessaires

```
ib_ipoib # ip over ib
ib_ucm                 18489  0
ib_ipoib              132849  0
ib_cm                  47000  3 rdma_cm,ib_ucm,ib_ipoib
ib_uverbs              75206  2 ib_ucm,rdma_ucm
ib_umad                22330  6
mlx4_ib               195822  0
ib_sa                  33950  5 rdma_cm,ib_cm,mlx4_ib,rdma_ucm,ib_ipoib
ib_mad                 56678  4 ib_cm,ib_sa,mlx4_ib,ib_umad
mlx4_core             346830  2 mlx4_en,mlx4_ib
libcrc32c              12644  1 xfs
mlx5_ib               215909  0
ib_core               151094  12 rdma_cm,ib_cm,ib_sa,iw_cm,mlx4_ib,mlx
ib_addr                19143  3 rdma_cm,ib_core,rdma_ucm
ib_netlink             14070  3 rdma_cm,iw_cm,ib_addr
mlx5_core             532312  1 mlx5_ib
mlx_compat             16639  18 rdma_cm,ib_cm,ib_sa,iw_cm,mlx4_en,mlx
```

Infiniband cable Coopeer

Fibre Mellanox

Transceiver optique

Cable splitter

Gateway IB/Ethernet

Switch EDR 648 ports

Switch FDR 648 ports