ORACLE®

Virtual Machine Migration with SR-IOV Supported InfiniBand

Wei Lin Guay, Bjørn Dag Johnsen, Chien-Hua Yen, Richard Frank
and Sven-Arne Reinemo (Simula Research Laboratory)
March. 2012

Table of Content

- Introduction
- Migrating an InfiniBand (VF) vs Ethernet (VF).
- The challenges of *hot* migration with active QP.
  - Detaching a VF if an active QP exists.
  - Manage the location dependent resources.
  - Handle the outstanding operations.
  - Re-establish the connection with peer QP.
- *Bottom-up* approach – based on hot plug mechanism.
- *Top-down* approach using Reliable Datagram Socket (RDS)
- Discussions.

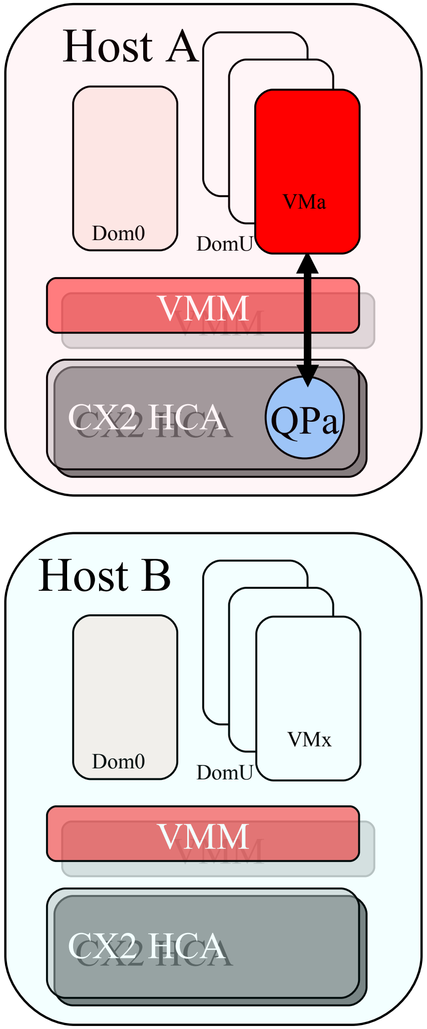
Introduction

- Live migration is a powerful feature provided by virtualization.
  - Ease cluster management.
  - Load balancing.
  - Fault tolerance.
- Lot of works have been demonstrated with *pass-through* device and SR-IOV (mainly *Ethernet*)? How about with InfiniBand (IB) SR-IOV?
- Each guest VM can obtain near to native IB performance but it is not compatible with live migration.

Migrating an IB VF vs an Ethernet VF?

- No Linux bonding device available yet for IB native network except bundles with IPoIB.
- Not only need to maintain the hardware state (SR-IOV) but keep track of the QP state.
- The addressing – LID is assigned by SM and all VF that shared the same physical port is sharing the same LID. (IB shared port implementation)

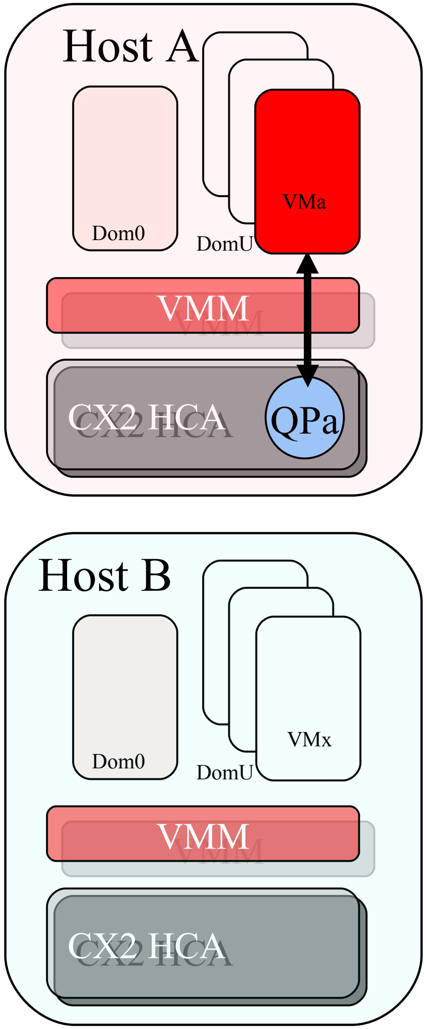
The First Problem



- Migrate VMa from Host A to Host B when QPa is an active QP.
- Problem 1a: VM migration is not allowed if a VF is attached to the VM.

From the perspective of migrated node...

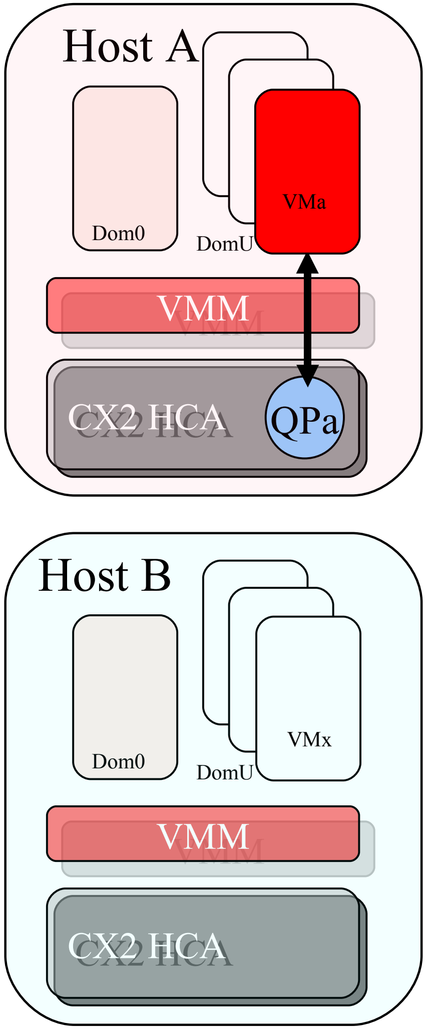
The First Problem (cont)



- A workaround for Problem 1a, hot-plug/unplug the VF from VMa.
- Problem 1b: Failed to detach its VF if an active QP exists (QPa).
- Problem 1(a+b): How to detach a VF in order to migrate a VM if an active QP exists.

From the perspective of migrated node...

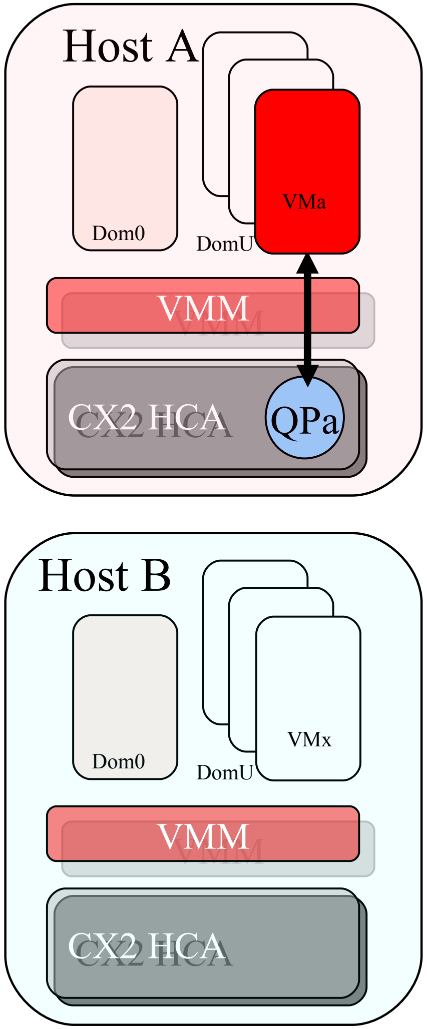
The First Problem (cont)



- Detach a VF from a VM if an active QP exists.
  - Most application is calling IB user verbs.
  - All ib_uverbs contexts need to be released before ib_uverbs module can be unloaded.
  - If a QP is created by an app, detaching a VF returns an error because the xen hot-plug time out before uverbs’ wait_for_completion().

From the perspective of migrated node...

The First Problem (cont)

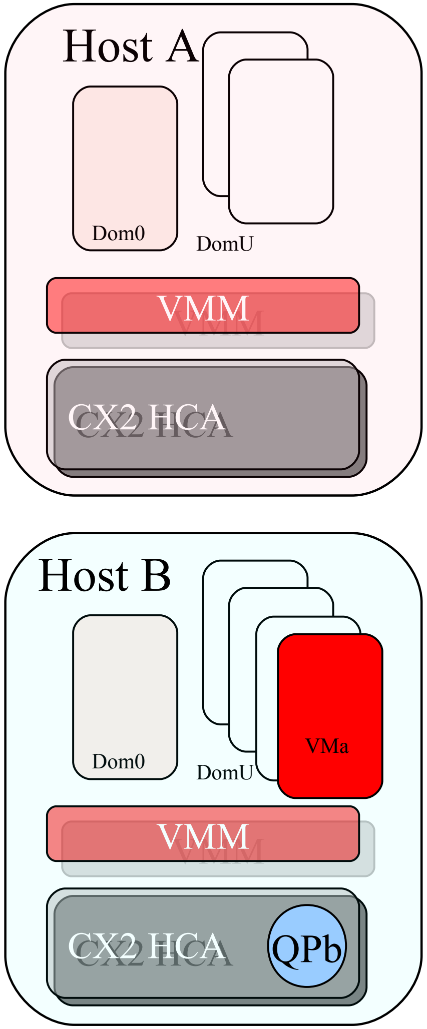


- Detach a VF from a VM if an active QP exists
  - When a QP is created, the PID of the user app. is registered in the kernel.
  - Before ib_uverbs is removed, the kernel *signals an event to the user space mlx4_ib.
  - The user space mlx4_ib releases the completion event.
  - Uverbs’ wait_for_completion() is executed successfully.
  - VF is detached from VMa successfully.
  - The code stays in a loop until a new VF is reattached to VMa.

* A potential security hole when kernel is calling a function in the user space.

From the perspective of migrated node...

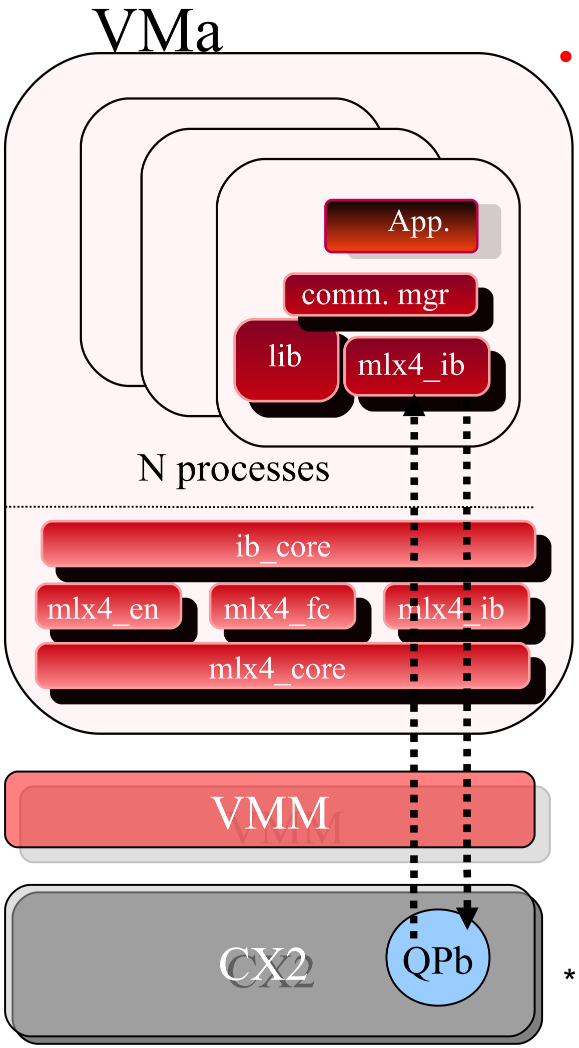
The Second Problem



- After VMa is migrated from Host A to Host B, a new VF is attached to VMa.
- User application continues with the invalid opaque handle that is pointed to QPa which no longer exists in Host B.
- Problem 2: How can user application continues with QPa’ (create a new QP - QPb) after the migration?

From the perspective of migrated node...

The Second Problem (cont)

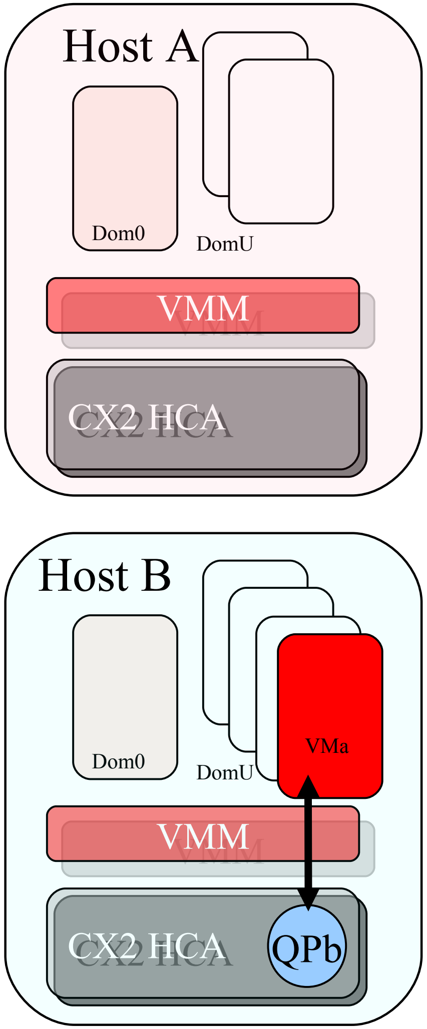


- Handle the location dependent resources.
  - Recreate *PD, MR, CQ, and QP that is based on the QPa's qpn.
  - The VM migration logic is part of the user space mlx4_ib device driver.
  - Why we implement the logic on the user space mlx4_ib device driver?
    - The time critical operations such as post send, poll cq do not involve kernel space.
    - Host B might have the same qpn with Qpa. We can avoid the conflict in qpn if the translation table is only visible by per user process.

* This includes a mechanism to replace the rkey and lkey in the posted WRE.

From the perspective of migrated node...

The Third Problem

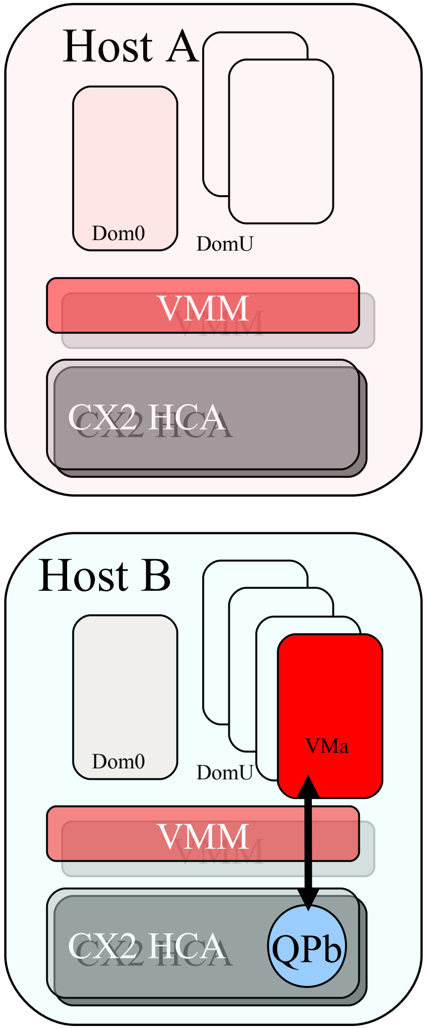


- User application resumes the send operation with QPb on host B.
- Before migration, there are some outstanding send operations. *How to retrieve and continue those operations in Host B?
- Problem 3: How to handle the outstanding send operations after migration?

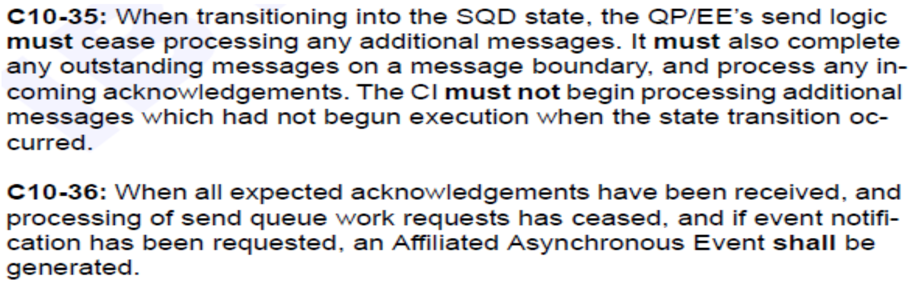
* Assume that there is no recovery mechanism in the ULP.

From the perspective of migrated node...

The Third Problem (cont)



- Maintain the connection state information.
  - *Simulate SQD-like QP state in SW.
  - Use a deterministic state for migration.
  - All the outstanding send operations are completed (received the CQ) before detach the VF from Host A.
  - QPb does not require to handle the remaining (outstanding) send operations in the QPa’s sender queue.



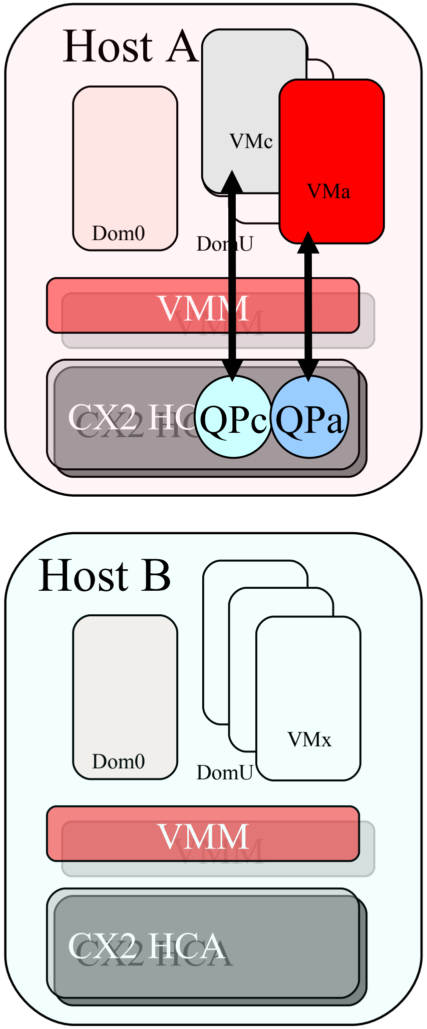
C10-35: When transitioning into the SQD state, the QP/EE’s send logic **must** cease processing any additional messages. It **must** also complete any outstanding messages on a message boundary, and process any incoming acknowledgements. The CI **must not** begin processing additional messages which had not begun execution when the state transition occurred.

C10-36: When all expected acknowledgements have been received, and processing of send queue work requests has ceased, and if event notification has been requested, an Affiliated Asynchronous Event **shall** be generated.

*SQD is not supported by MLX CX2 yet. Another difference is that our implementation always make sure that all WREs within SQ are sent.

From the perspective of migrated node...

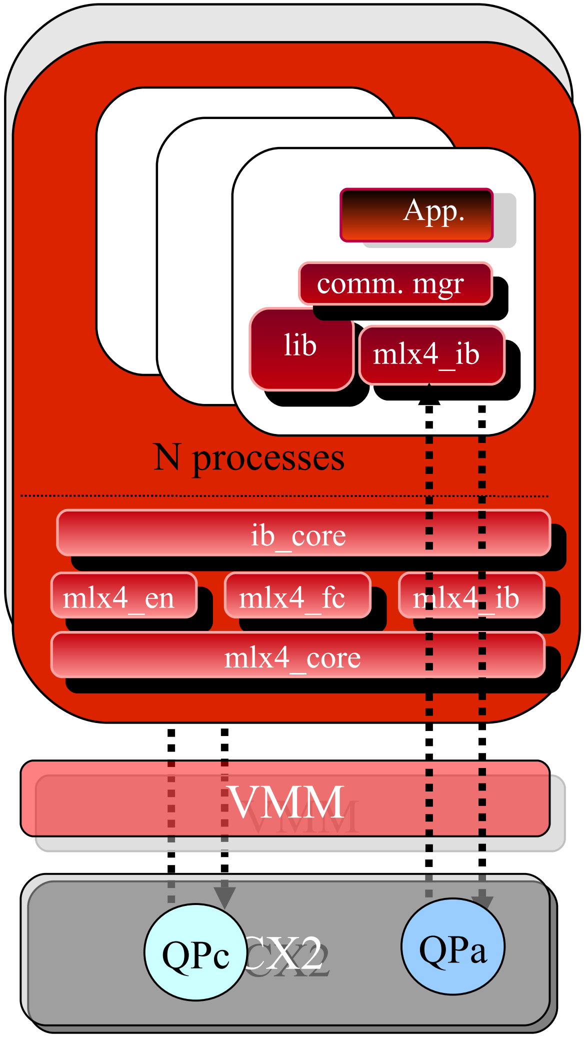
The Forth Problem



- The original QPa (VMa) is communicating with QPc (VMc).

From the perspective of the connected QPs of the migrated node...

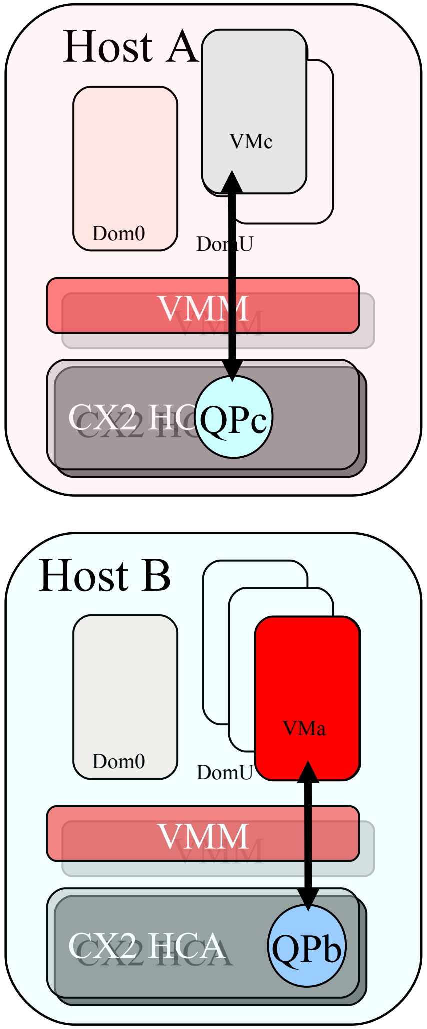
The Forth Problem (cont)



- The original QPa (VMa) is communicating with QPc (VMc).
- After VMa is migrated to host B, QPb is created to continue the remaining operations.
- However, QPc is not being notifed that QPa is replaced by QPb in Host B.
- Problem 4: How to maintain the connection with the remote QP after the migration?

From the perspective of the connected QPs of the migrated node...

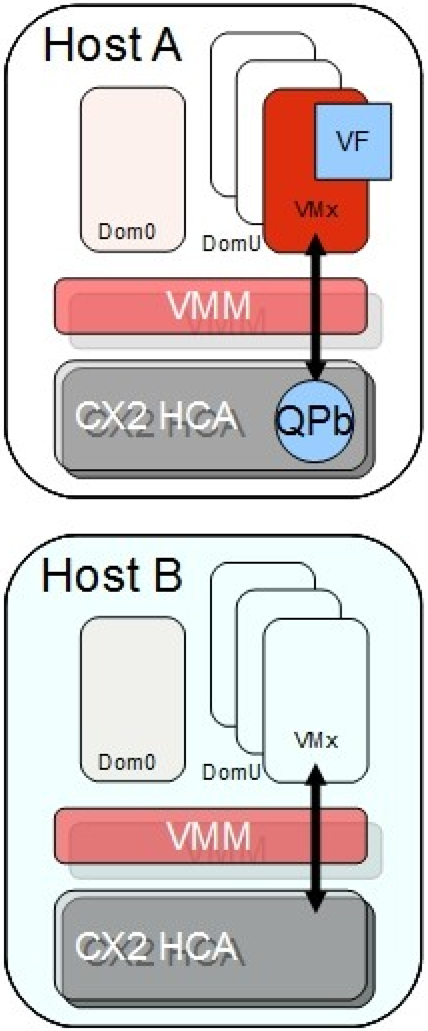
The Forth Problem (cont)



- Re-establish the remote QP connection.
  - Assumption: all app. must use RDMA_CM to establish the connection.
  - Before VMa releases the uverb, it notifies VMc using the RC connection.
  - After VMa destroys the CM, do not issue DREQ to the remote QP.
  - Keep track of the sideband communcation (socket destaddr) in VMa.
  - After VMa has attached a new VF on host B, notify VMc and create a new cm_id to establish a connection with VMc.
  - RDMA_CM is responsible to exchange psn, qpn between QPc and QPb.

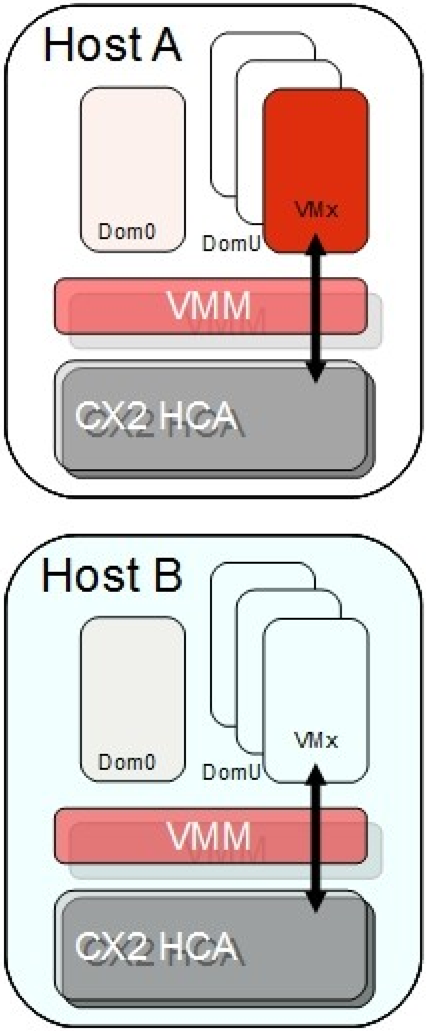
From the perspective of the connected QPs of the migrated node...

Bottom-up approach



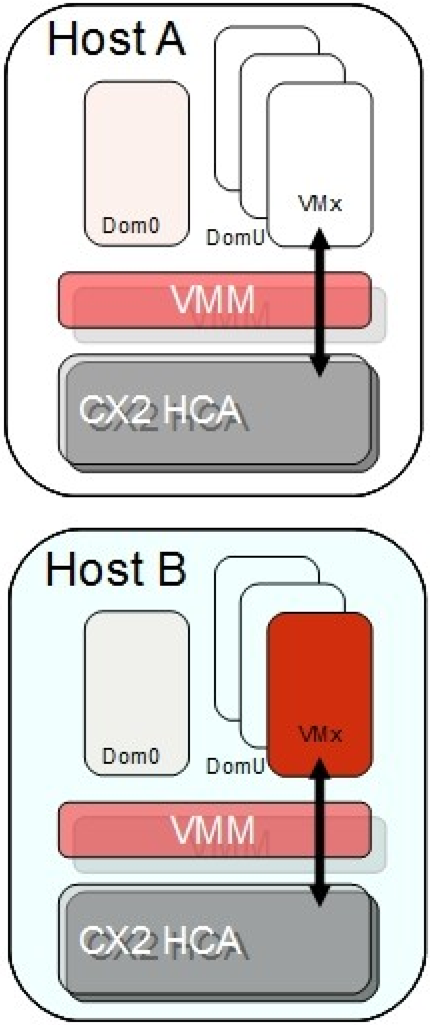
- A three-stage migration process is performed to migrate a VM using IPoIB.

Bottom-up approach (cont)



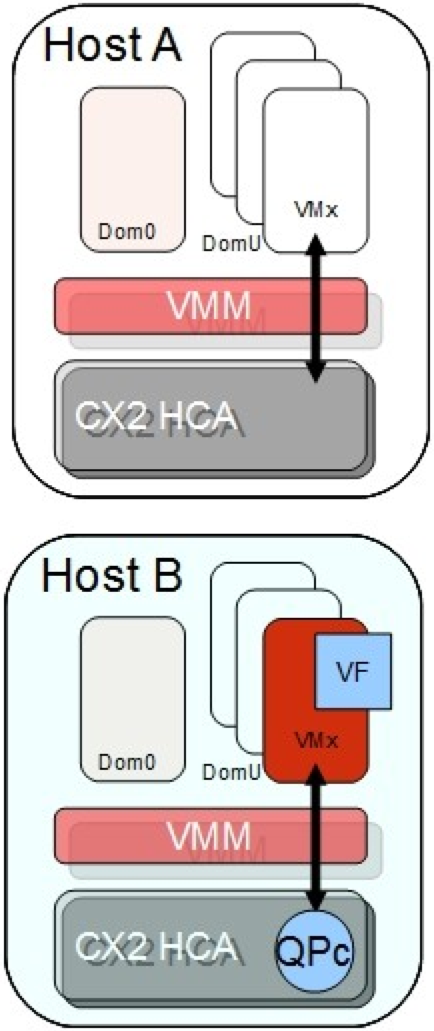
- A three-stage migration process is performed to migrate a VM using IPoIB.
  - Stage 1: Detach a PCI-bypass virtual function.

Bottom-up approach (cont)



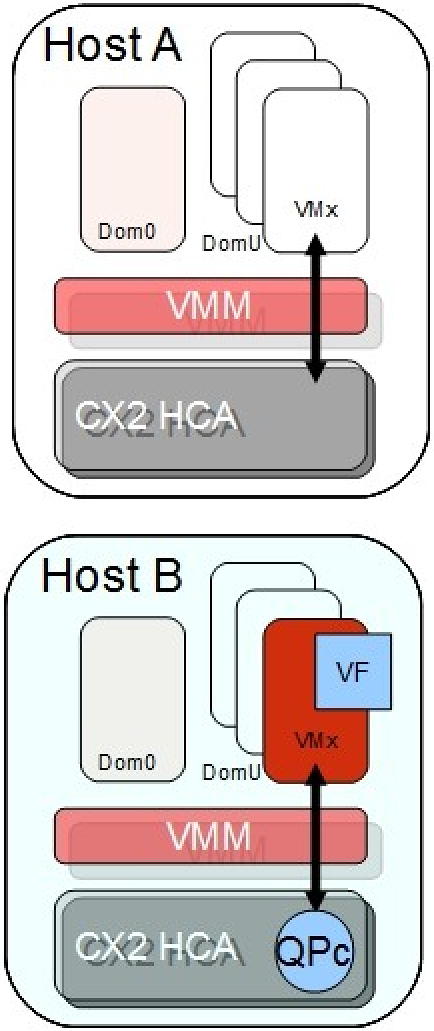
- A three-stage migration process is performed to migrate a VM using IPoIB.
  - Stage 1: Detach a PCI-bypass virtual function.
  - Stage 2: Migrate the VM.

Bottom-up approach (cont)



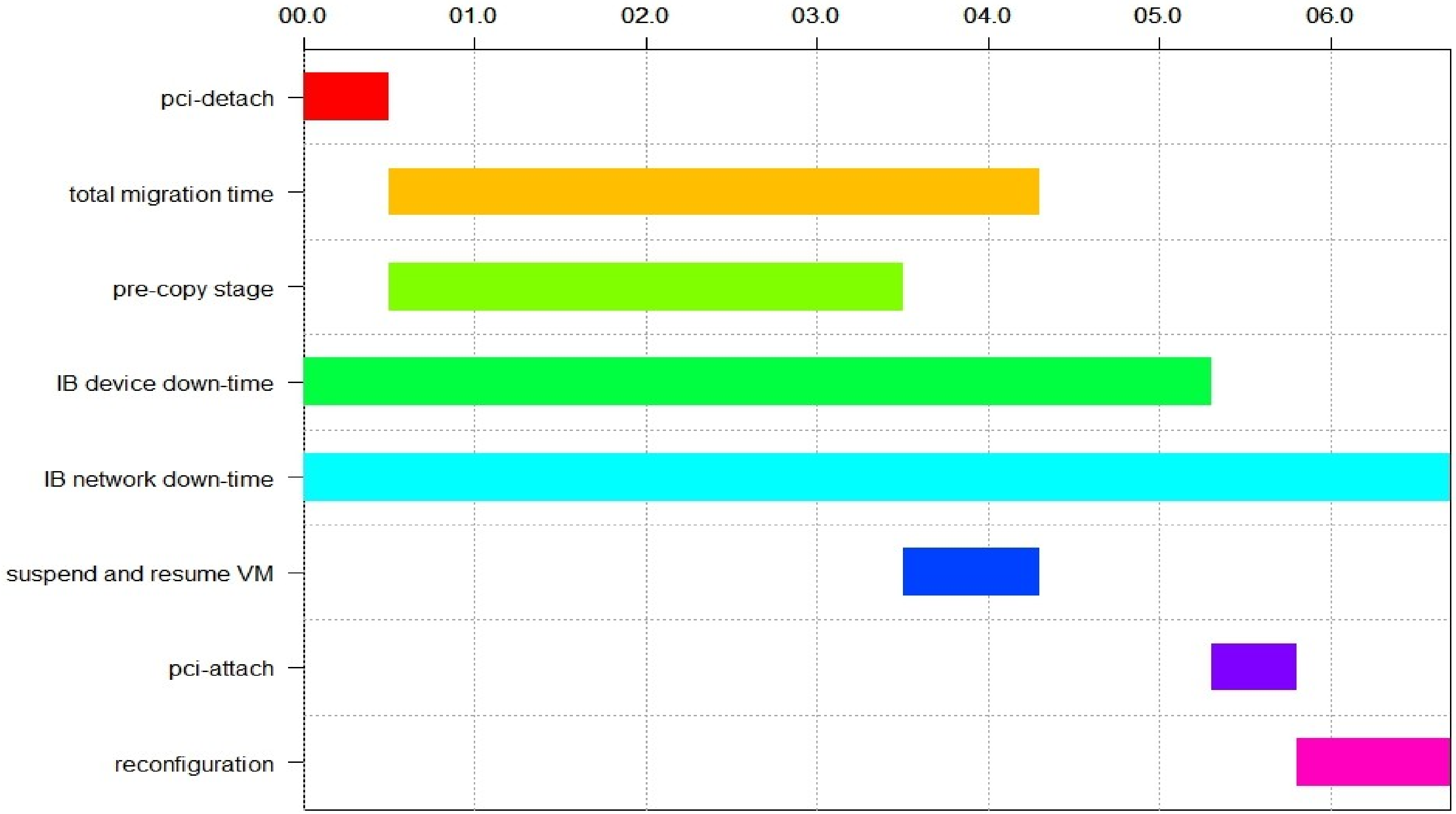
- A three-stage migration process is performed to migrate a VM using IPoIB.
  - Stage 1: Detach a PCI-bypass virtual function.
  - Stage 2: Migrate the VM.
  - Stage 3: Attach a new PCI-bypass virtual function.

Bottom-up approach (cont)

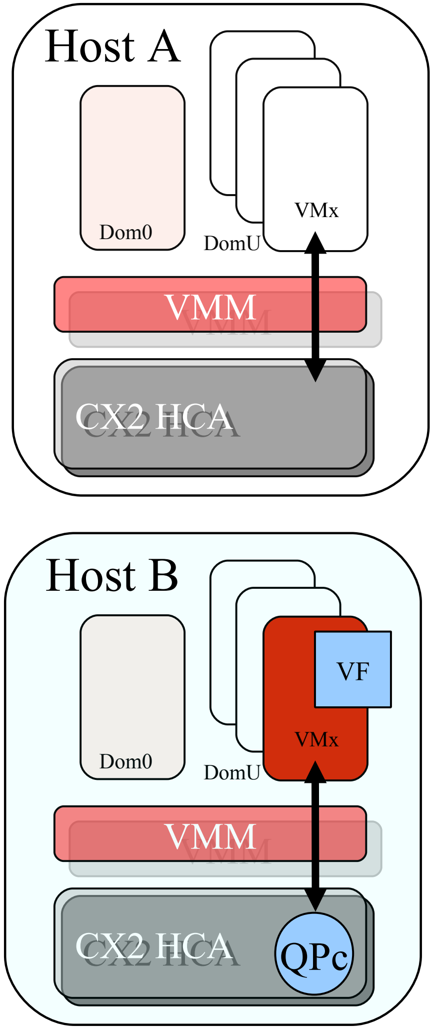


- A three-stage migration process is performed to migrate a VM using IPoIB.
  - Stage 1: Detach a PCI-bypass virtual function.
  - Stage 2: Migrate the VM.
  - Stage 3: Attach a new PCI-bypass virtual function.
- A three-stage process is time-consuming.
- It requires user to detach/attach a VF manually.
- Problem: How to reduce IB device down-time?

Bottom-up approach (cont)

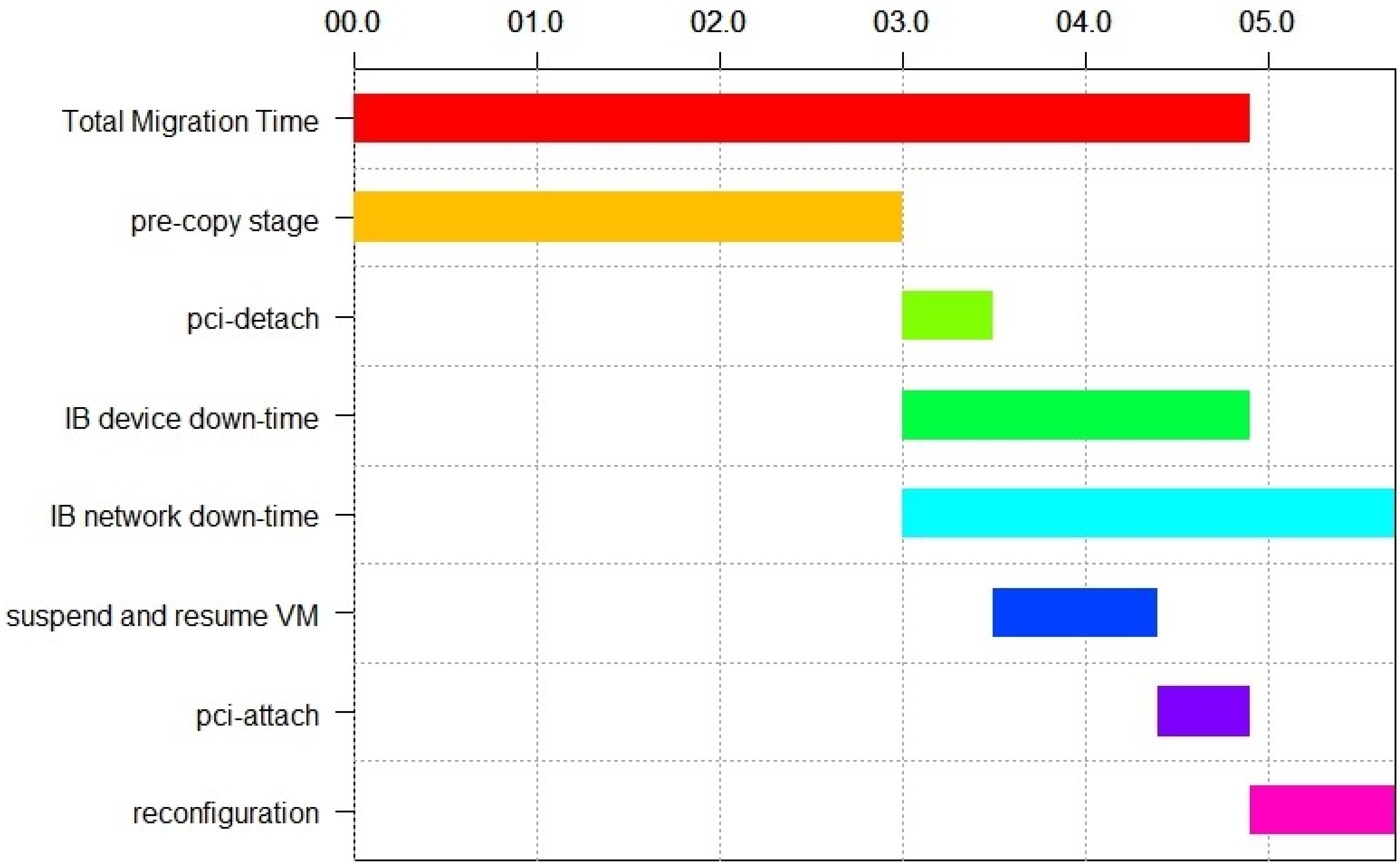


Bottom-up approach (cont)

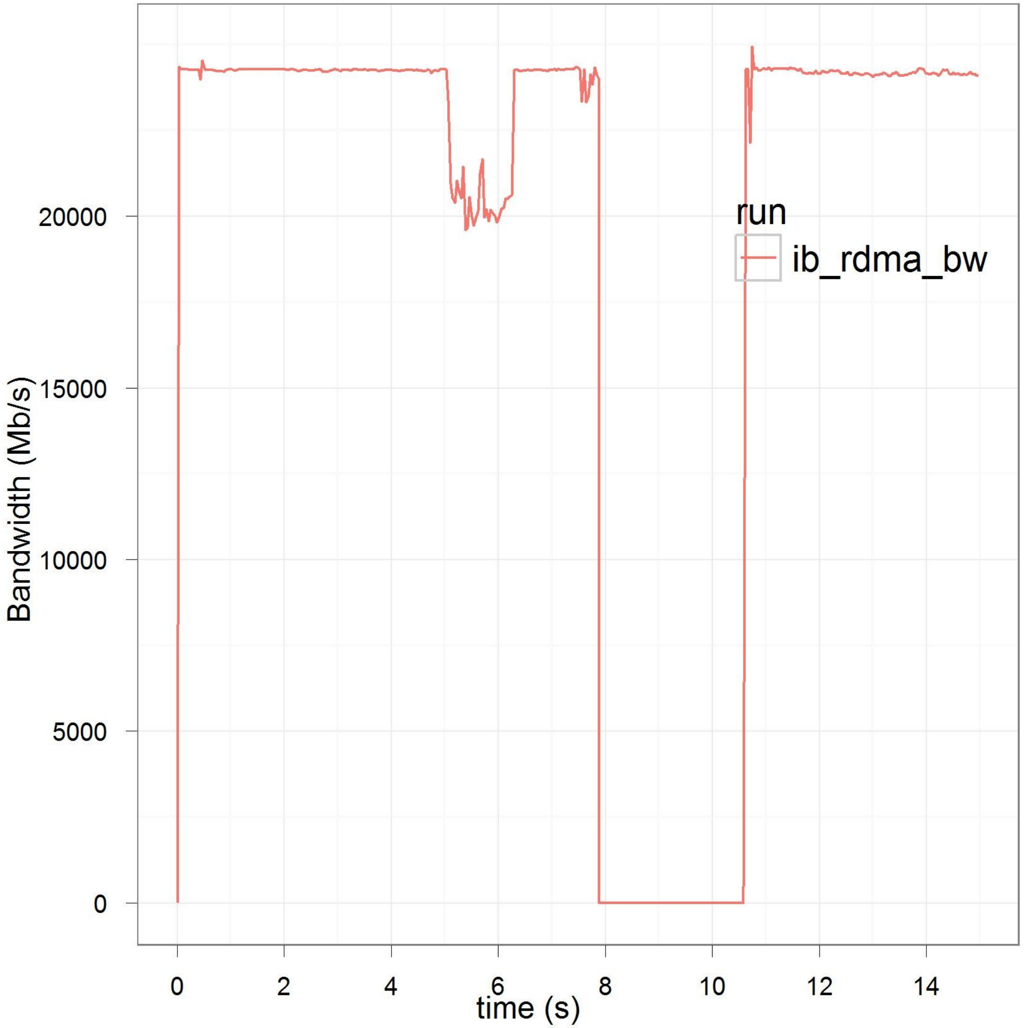


- Reduce IB device down-time during migration.
  - ’workaround’ the xen migration script to allow migration with a PCI-bypass virtual function.
  - Migrating node: detach the VF before the VM is suspended.
  - Dest. node: do not initialise the PCI bypass function during the early restoration.
  - Dest. node: attach a new VF at the final stage of the restoration.

Bottom-up approach (cont)



Bottom-up approach (cont)

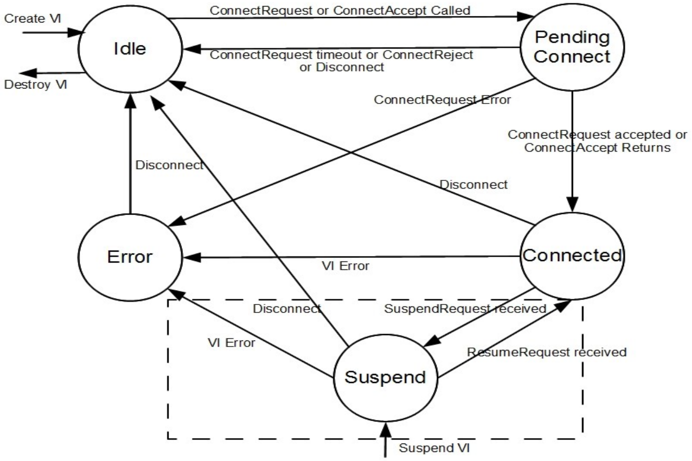


Top-down approach

- Reliable Datagram Socket (RDS) is a reliable-socket protocol with low-overhead, low latency and high-bandwidth.
- RDS is tolerance to fault including the *device removal*.
- The migration model remains the same where it is still based on hot plugging mechanism.
- TODO – show the result with rds-stress w/ bcopy + live migration

Discussion

- It is not acceptable to have the service downtime of 2.7s. How can the service downtime to be further improved?
- A *vSwitch* model is a better architecture that can integrate SR-IOV with IBA/VIA?
    - A better model from networking and routing perspective.
    - A better model from Virtual Machine perspective. (service downtime)
- A new state diagram to define *Suspend*?



Question?