Datacenter Fabric Workshop
Windows IB

Windows Core SW
Kernel Mode Future

Fab Tillier
SilverStorm Technologies
ftillier@silverstorm.com

August 22, 2005

Agenda

- General Direction
- Verbs at DISPATCH_LEVEL
- HCA device resolution for UM
- Client Reference Counting
- HCA Future

General Direction

- Better integration into Windows
- Leverage key design elements from Linux OpenIB stack where it makes sense
  - CQ Polling WC array vs. linked list
  - Public structures for objects instead of opaque handles
- Partition Support
- InfiniBand 1.2 Compliance

Agenda

- General Direction
- Verbs at DISPATCH_LEVEL
- HCA device resolution for UM
- Client Reference Counting
- HCA Future

Verbs at DISPATCH_LEVEL

Problem:

- Currently, most verbs block
  - Requires IRQL < DISPATCH_LEVEL
- Miniport entrypoints are invoked at DISPATCH_LEVEL
  - ULPs must find ways to get into a thread context capable of performing verb calls
- Local MAD pre-empted by I/O completion processing
  - Non-responsive node from the SM perspective

Verbs at DISPATCH_LEVEL

Solution:

- Allow all operations to be performed at DISPATCH_LEVEL
- Client chooses async vs. sync processing
- Use I/O Completion routines for async completion notifications
  - IRP based
- Support both IOCTL and direct call paths

Resource Creation - IOCTL

- Used by user-mode
- Can be used by kernel clients
- Object handles returned in pIrp->IoStatus.Information
- IOCTL input and output buffer definitions public
- Initiated via IoCallDriver
  - Typically sent to PDO of client's devnode

Resource Creation - Direct

- Available only to kernel clients
- Object handles returned in pIrp->IoStatus.Information

```
NTSTATUS IbCreateCq(
      IN    IB_CA *pCa,
      IN    IB_CQ_CREATE *pCqCreate,
      IN    IRP *pIrp );
```

Resource Destruction - IOCTL

- Used by user-mode
- Can be used by kernel clients
- IOCTL input and output buffer definitions public
- Initiated via IoCallDriver
  - Typically sent to PDO of client's devnode

Resource Destruction - Direct

- Available only to kernel clients

```
NTSTATUS IbDestroyCq(
      IN   IB_CQ *pCq,
      IN   IRP *pIrp );
```

Agenda

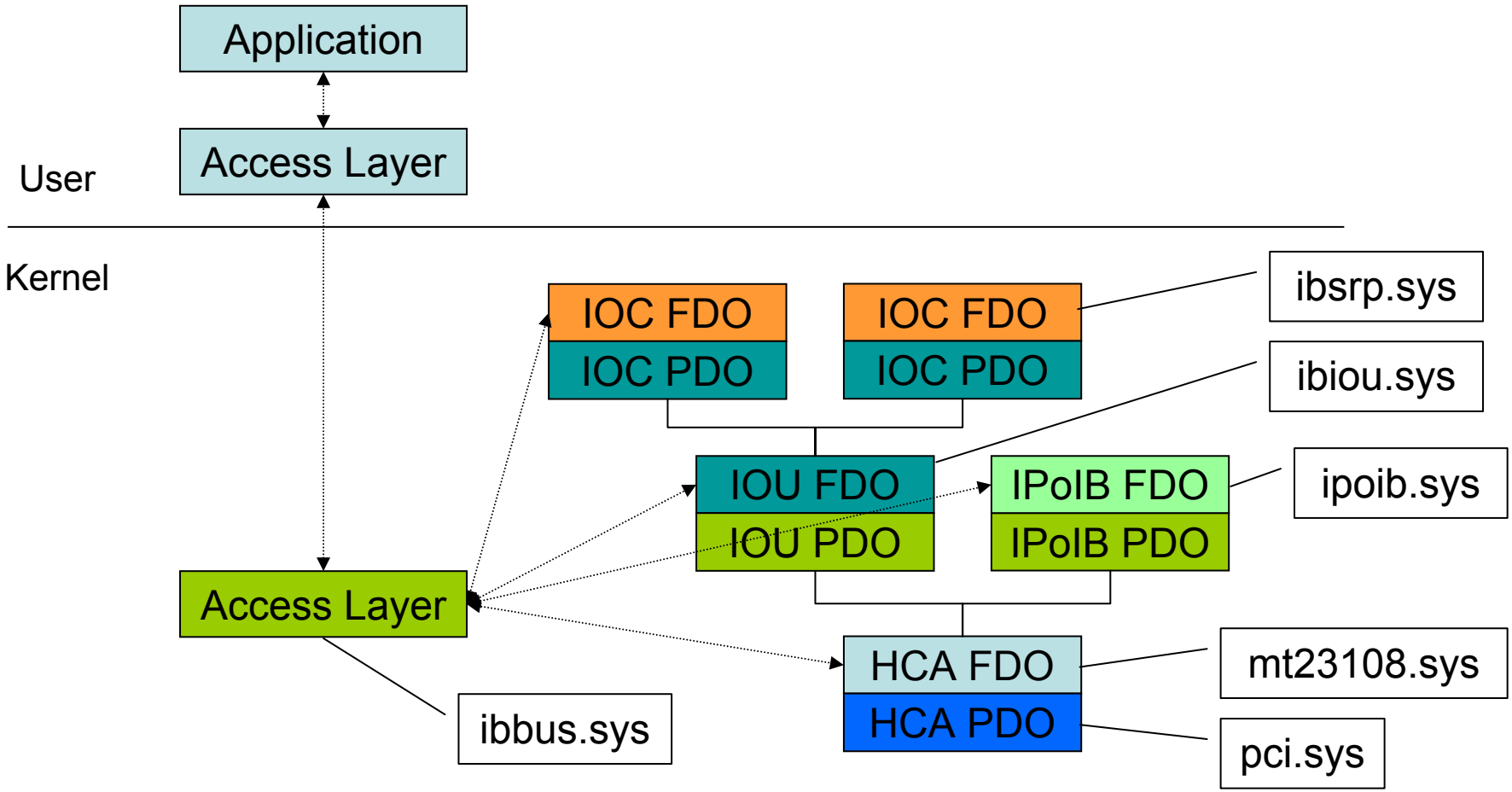
- General Direction
- Verbs at DISPATCH_LEVEL
- HCA device resolution for UM
- Client Reference Counting
- HCA Future

HCA Resolution

Problem:

- Single file object exposed to UM
  - \Device\ibal
- No relationship between file object and target HCA
  - Device usage not reflected by file usage
  - Affects PnP Manager
  - Requires custom PnP notification mechanism

HCA Resolution



HCA Resolution

Solution:

- Use Reparse Points!
- Applications open simple file name:
  - \Device\iba\<CA GUID>
- Access Layer redirects to proper target
- All verb calls performed on CA's file

Agenda

- General Direction
- Verbs at DISPATCH_LEVEL
- HCA device resolution for UM
- Client Reference Counting
- HCA Future

Client Reference Counting

Problem:

- Device drivers can unload without first cleaning up
- Can cause system crash if a callback is invoked into an unloaded module

```
ib_api_status_t
ib_open_al(
     OUT  ib_al_handle_t* const ph_al );
```

Client Reference Counting

Solution:

- Take as input either a device object or driver object on which to take a reference
  - ObReferenceObject
- Release the reference when it is safe to do so
  - ObDereferenceObject

```
NTSTATUS
IbOpenAl(
        IN      DEVICE_OBJECT* pDevObj,
        OUT     IB_AL** const pAl );
```

Agenda

- General Direction
- Verbs at DISPATCH_LEVEL
- HCA device resolution for UM
- Client Reference Counting
- HCA Future

HCA Future

- Mellanox memfree technology support
- IRP-based verbs
  - Use Irp->RequestorMode to determine UM vs. KM calls

Resources

- OpenIB WiKi
  - https://openib.org/tiki/tiki-index.php?page=OpenIB+Windows
- Openib-windows mailing list
  - http://openib.org/mailman/listinfo/openib-windows
- Sign up to contribute
  - http://windows.openib.org/openib/contribute.aspx

Q & A