BKM for OFED – Installation, configuration, scale-out configuration, testing, error codes

Test from lowest level up as follow:

1. **Fabric**
2. **Verbs**
3. **Rdma_cm**
4. **uDAPL**
5. **Intel MPI**

Installation:

OFED 1.5 or newer recommended:

http://www.openfabrics.org/downloads/OFED

Latest uDAPL v2 package – (used by OFED and Distributors)

http://www.openfabrics.org/downloads/dapl/dapl-2.0.30.tar.gz

uDAPL documentation

http://www.openfabrics.org/downloads/dapl/documentation

OFED wiki

http://wiki.openfabrics.org/wiki/index.php/Main_Page

Firmware (Mellanox):

http://www.mellanox.com/supportdownloader/

Example query: mstflint -nofs -d /proc/bus/pci/04/00.0 query
Example burn: mstflint -nofs -d /proc/bus/pci/04/00.0 -i /root/firmware/fw-2xxx.bin bu
Example build image: mlxburn -fw fw-25408-debug.mlx -conf 448262-B21.ini -wrimage ./fw-HP-mlx4-rel-2_4_938.bin

Basic configuration:

1. Bump up memlock settings (be careful here):

/etc/security/limits.conf

```
*          hard    memlock       2000000
*          soft    memlock       2000000
```

Note: This is a per-process limit. Administrators and applications need to be aware of the possibility "**Out of Memory**" when over subscribing pinned memory as you scale up on many cores. The rdma verbs simply uses this limit for checking during registration and doesn't limit based on system wide memory usage.

2. Check udev rules

/etc/udev/rules.d/90-ib.rules

set umad permissions to 0666 if you want users to run ibdiags

KERNEL="umad*", NAME="infiniband/%k", MODE="0666"
KERNEL="issm*", NAME="infiniband/%k", MODE="0666"
KERNEL="ucm*", NAME="infiniband/%k", MODE="0666"
KERNEL="uverbs*", NAME="infiniband/%k", MODE="0666"
KERNEL="uat", NAME="infiniband/%k", MODE="0666"
KERNEL="ucma", NAME="infiniband/%k", MODE="0666"
KERNEL="rdma_cm", NAME="infiniband/%k", MODE="0666"

3. IPoIB configuration (connected mode or non-connected, mtu settings for cm):

/etc/infiniband/openib.conf

SET_IPOIB_CM=yes **(large clusters should** SET_IPOIB_CM=no**)**
NOTE: default MTU=65520 causes slow scp performance, set MTU=32768

For statistics:
cat /sys/class/net/ib0/statistics/*

For mode:
cat /sys/class/net/ib0/mode

4. IPoIB address configuration and testing:

Sample ifcfg-ib0 with static ip addresses

[cst-1]$ cat /etc/sysconfig/network-scripts/ifcfg-ib0
DEVICE=ib0
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.0.50
NETMASK=255.255.255.0

create a pingall.sh and ping every IPoIB interface on the cluster

pingall.sh (example for a 10 node cluster)
ping -c 1 192.168.0.50
ping -c 1 192.168.0.51
ping -c 1 192.168.0.52
ping -c 1 192.168.0.53
ping -c 1 192.168.0.54
ping -c 1 192.168.0.55
ping -c 1 192.168.0.56
ping -c 1 192.168.0.57
ping -c 1 192.168.0.58
ping -c 1 192.168.0.59

run from everynode (will warm up the ARP cache with IPoIB entries)

pdsh –a pingall.sh

5. Multi IB port configuration, IPoIB arp reply issues

When two interfaces running one interface may reply to an ARP directed to the other interface on the system. The following configuration will cause the interfaces to ignore ARP requests if not specifically for their IP address.

Add the following lines to /etc/sysctl.conf

```
net.ipv4.conf.all.arp_ignore=1
net.ipv4.conf.ib0.arp_ignore=1
net.ipv4.conf.ib1.arp_ignore=1
```

or use sysctl:

```
sysctl -w net.ipv4.conf.all.arp_ignore=1
sysctl -w net.ipv4.conf.ib0.arp_ignore=1
sysctl -w net.ipv4.conf.ib1.arp_ignore=1
```

Scale-out configuration (64+ nodes):

1. Configure IPoIB to start in non-connected mode (UD):

/etc/infiniband/openib.conf

SET_IPOIB_CM=no

2. Configure and increase open files (3 fd's for each dapl evd, 2 per open):

cat /proc/sys/fs/file-nr
ulimit –f 100000

3. Monitor IPoIB errors when scaling out:

Run "netstat –iib0" to see if there is a pattern of ERR or DRP packets at the IPoIB driver level.

Kernel Interface table

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR
ib0	2044	0	163592	0	0	0	117	0	2	0

Run check before and after your job, look for ERR or DRP:

```
pdsh –a netstat –iib0 > before
run job
pdsh –a netstat –iib0 > after
```

4. Increase send/recv queue sizes via IPoIB driver:

IPoIB supports increasing sizes of send and recv queues through a module parameter. The parameter value can be controlled at load time or runtime. At load time this can be done by inserting the following line in

/etc/modprobe.conf:

```
options ib_ipoib send_queue_size=128
options ib_ipoib recv_queue_size=128
```

Verify ib0 hwaddr, mtu, queue size with "/sbin/ifconfig ib0"

5. Increase ARP timeout for IPoIB. arp default timeout is set to 30 seconds;

```
[root@hn ~]# sysctl net.ipv4.neigh.ib0.base_reachable_time
net.ipv4.neigh.ib0.base_reachable_time = 30
```

We should go for larger timeout as follow:

```
[root@hn ~]# sysctl -w net.ipv4.neigh.ib0.base_reachable_time=14400
net.ipv4.neigh.ib0.base_reachable_time = 14400
```

6. OFED uDAPL provider recommendations (/etc/dat.conf) for Intel MPI:

Note: There are 3 providers, with different connection managers, for all devices. The following recommendations are for larger clusters, assuming a mlx4 device.

/etc/dat.conf device entries for mlx4 port 1. SCM, UCM, and CMA types:

ofa-v2-mlx4_0-1 u2.0 nonthreadsafe default libdaploscm.so.2 dapl.2.0 "mlx4_0 1" ""
ofa-v2-mlx4_0-1u u2.0 nonthreadsafe default libdaploucm.so.2 dapl.2.0 "mlx4_0 1" ""
ofa-v2-ib0 u2.0 nonthreadsafe default libdaplofa.so.2 dapl.2.0 "ib0 0" ""

======

6.1 **SCM** - uses sockets to exchange QP information.
IPoIB, ARP, and SA queries NOT required.

Pros: Each rank has own instance of socket cm. Doesn't require path-record lookup.
Cons: Socket resources grow with scale-out, serialization of connections with kernel based tcp sockets, competing for MPI socket resources/port space for job startup and possibly other TCP application. Sockets remain in TIMEWAIT state for minutes after closure. Requires ARP for name resolution. Doesn't support iWARP devices.

export I_MPI_DEVICE=rdssm:**ofa-v2-mlx4_0-1**

6.2 **UCM** - use's IB UD QP to exchange QP info.
Sockets, ARP, IPoIB, SA queries NOT required.

Pros: Each rank has own instance of CM, resources fixed per rank regardless of scale-out,
no serialization of user or kernel resources establishing connections,
simple 3-way msg handsake,
CM messages fit in inline data for lowest message latency,
no address resolution nor path resolution required.
Cons: brand new provider with limited testing, a little tougher to debug.
Doesn't support iWARP.

export I_MPI_DEVICE=rdssm:**ofa-v2-mlx4_0-1u**

6.3 **CMA** - uses OFA rdma_cm to setup QP's.
IPoIB, ARP, and SA queries required.

Pros: OFA rdma_cm has the most testing across many applications.
Supports both iWARP and IB.

Cons: serialization of connections with kernel based CM service,
Requires ARP for name resolution and SA for path record queries for IB fabrics.
Supports only 48 bytes of private data, less then DAT minimum of 64 bytes.

export I_MPI_DEVICE=rdssm:**ofa-v2-ib0**

OFED IB fabric and SM testing

make sure permissions of /dev/infiniband/umad0 is set for users or run as root.

crw-rw-rw- 1 root root 231, 0 Dec 1 16:16 umad0

run **ibnetdiscover** to create a topology file (for ibchecknet)

```
clear error counters to start

ibdiagnet –pc

run a query node description test to load fabric

ibdiagnet –pc –c 1000

ibcheckerrors
ibchecknet

NOTE: Results from ibchecknet indicate potential bad cable:

      #warn: Link configured as 1X
      Port check lid 10 port 8:

      #warn: Link configured as 1X
      Port check lid 32 port 1:
```

NOTE: false positive with forwarded ARP back to src port on some switches
#warn: counter RcvSwRelayErrors = 3971

```
LINK error definitions
ibIfPortSymbolErrs
      "Total number of symbol errors detected on one or virtual
       more lanes."
ibIfPortLinkErrRecovery
      "Total number of times the Port Training state machine has
      successfully completed the link error recovery process."
ibIfPortLinkDowned
      "Total number of times the Port Training state machine
       has failed the link error recovery process and downed
       the link."
ibIfPortStatLocalPhyErrs
      "Total number of packets received on the port that contain
       local physical errors (ICRC, VCRC, FCCRC, and
       all physical errors that cause entry into the BAD
       PACKET or BAD PACKET DISCARD states of the
       packet receiver state machine)."
ibIfPortStatMalPktErrs
      "Total number of packets received on the port that contain
       malformed packet errors
       - data packets: LVer, length, VL
       - link packets: operand, length, VL"
ibIfPortStatRcvRemPhyErrs
      "Total number of packets marked with the EBP delimiter
      received on the port."

 ibIfPortStatRcvConstrErrs
```

```
       "Total number of packets received on the port that are
        discarded for the following reasons:
        - FilterRawInbound is true and packet is raw
        - PartitionEnforcementInbound is true and packet fails
        partition key check, IP version check, or transport
        header version check."
ibIfPortStatInactDiscards
         "Total number of outbound packets discarded by the port
         because it is in the inactive state."
ibIfPortStatNeighMTUDiscards
       "Total number of outbound packets discarded by the
        port because packet length exceeded the neighbor MTU."
ibIfPortStatSwLifetimeDiscards
         "Total number of outbound packets discarded by the port
         because the switch lifetime limit was exceeded.
        Applies to switches only."
ibIfPortStatHOQLifetimeDiscards
       "Total number of outbound packets discarded by the
        port because the switch HOQ lifetime was exceeded.
        Applies to switches only."
ibIfPortStatLinkIntergrityErrs
       "The number of times that the frequency of packets
containing local physical errors exceeded local_phy_errors."
ibIfPortStatExcBufOverrunErrs
       "The number of times that overrun_errors consecutive
        flow control update periods occurred with at least one
        overrun error in each period."
ibIfPortStatVL15Dropped OBJECT-TYPE
       "Number of incoming VL15 packets dropped due to
        resource limitations on port selected by the port
        selected (due to lack of buffers)."
```

Run **perfquery** on each node to check local counters:

```
# Port counters: Lid 1 port 1
PortSelect:......................1
CounterSelect:...................0x0000
SymbolErrors:....................0
LinkRecovers:....................0
LinkDowned:......................0
RcvErrors:.......................0
RcvRemotePhysErrors:.............0
RcvSwRelayErrors:................0
XmtDiscards:.....................1
XmtConstraintErrors:.............0
RcvConstraintErrors:.............0
LinkIntegrityErrors:.............0
ExcBufOverrunErrors:.............0
VL15Dropped:.....................2
XmtData:........................4294967295
RcvData:........................4294967295
XmtPkts:........................98790210
RcvPkts:........................91843962
```

PORT error definitions

```
"SymbolErrors", "No action is required except if counter is
increasing along with LinkRecovers",
"LinkRecovers", "If this is increasing along with SymbolErrors
this may indicate a bad link, run ibswportwatch.pl on this port",
"LinkDowned", "Number of times the port has gone down (Usually
for valid reasons)",
"RcvErrors", "This is a bad link, if the link is internal to a
288 try setting SDR, otherwise check the cable",
"RcvRemotePhysErrors", "This indicates a problem ELSEWHERE in the
fabric."
"XmtDiscards", "This is a symptom of congestion and may require
tweeking either HOQ or switch lifetime values",
"XmtConstraintErrors", "This is a result of bad partitioning,
check partition configuration.",
"RcvConstraintErrors", "This is a result of bad partitioning,
check partition configuration.",
"LinkIntegrityErrors", "May indicate a bad link, run
ibswportwatch.pl on this port",
"ExcBufOverrunErrors", "This is a flow control state machine
error and can be caused by packets with physical errors",
"VL15Dropped", "check with ibswportwatch.pl, if increasing in
SMALL incriments, OK",
"RcvSwRelayErrors", "This counter can increase due to a valid
network event"
```

Mellanox Connectx (MLX4) adapter provides diag_counters

```
Clear counters:
echo 1 > /sys/class/infiniband/mlx4_0/diag_counters/clear_diag

Read counters:

for i in /sys/class/infiniband/mlx4_0/diag_counters/*; do echo -n
$i:;cat $i;done

/sys/class/infiniband/mlx4_0/diag_counters/clear_diag:This file
is write only
/sys/class/infiniband/mlx4_0/diag_counters/num_baddb:0
/sys/class/infiniband/mlx4_0/diag_counters/num_cqovf:0
/sys/class/infiniband/mlx4_0/diag_counters/num_eqovf:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_lae:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_leeoe:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_lle:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_lpe:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_lqpoe:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_mce:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_oos:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_rae:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_rire:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_rnr:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_rsync:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_ucsdprd:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_udsdprd:0
/sys/class/infiniband/mlx4_0/diag_counters/rq_num_wrfe:126
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_bre:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_ieecne:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_ieecse:0
```

```
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_leeoe:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_lle:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_lpe:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_lqpoe:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_mwbe:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_oos:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_rabrte:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_rae:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_rire:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_rnr:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_roe:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_rree:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_rsync:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_tree:0
/sys/class/infiniband/mlx4_0/diag_counters/sq_num_wrfe:0
```

diag_counter definitions:

RESPONDER	
rq_num_lae	Responder - number of local access errors
rq_num_leeoe	????
rq_num_lle	Responder - number of local length errors
rq_num_lpe	Responder - number of local protection errors
rq_num_lqpoe	Responder - number local QP operation error
rq_num_mce	Responder - number of bad multicast packets received
rq_num_oos	Responder - number of out of sequence requests received
rq_num_rae	Responder - number of remote access errors. R_Key Violation Responder detected an R_Key violation while executing an RDMA request. NAK may or may not be sent.
rq_num_rire	Responder - number of remote invalid request errors. NAK may or may not be sent. 1. QP Async Affiliated Error: Unsupported or Reserved OpCode (RC only): Inbound request OpCode was either reserved, or was for a function not supported by this QP. (E.g. RDMA or ATOMIC on QP not set up for this). 2. Misaligned ATOMIC: VA does not point to an aligned address on an atomic operation. 3. Too many RDMA READ or ATOMIC Requests: There were more requests received and not ACKed than allowed. 4. Out of Sequence OpCode, current packet is "First" or "Only": The Responder detected an error in the sequence of OpCodes; a missing "Last" packet 5. Out of Sequence OpCode, current packet is not "First" or "Only": The Responder detected an error in the sequence of OpCodes; a missing "First" packet 6. Local Length Error: Inbound "Send" request message exceeded the responder.s available buffer space. 7. Length error: RDMA WRITE request message contained too much or too little pay-load data compared to the DMA length advertised in the first or only packet. 8. Length error: Payload length was not consistent with the opcode: a: 0 byte <= "only" <= PMTU bytes b: ("first" or "middle") == PMTU bytes c: 1byte <= "last" <= PMTU bytes 9. Length error: Inbound message exceeded the size supported by the CA port.
rq_num_rnr	Responder - the number of RNR Naks sent
rq_num_rsync	????
rq_num_ucsdprd	The number of UC packets silently discarded on the receive queue due to lack of receive descriptor.
rq_num_udsdprd	The number of UD packets silently discarded on the receive queue due to lack of receive descriptor.
rq_num_wrfe	Responder - number of CQEs with error. Incremented each time a CQE with error is generated

REQUESTER	
sq_num_bre	Requester - number of bad response errors
sq_num_ieecne	????
sq_num_ieecse	????
sq_num_leeoe	????
sq_num_lle	Requester - number of local length errors
sq_num_lpe	Requester - number of local protection errors
sq_num_lqpoe	Requester - number local QP operation error
sq_num_mwbe	Requester - number of Memory Window bind errors
sq_num_oos	Requester - number of out of sequence Naks received
sq_num_rabrte	Requester - number of remote aborted errors
sq_num_rae	Requester - number of remote access errors NAK-Remote Access Error on: R_Key Violation: Responder detected an invalid R_Key while executing an RDMA Request.
sq_num_rire	Requester - number of remote invalid request errors. NAK-Invalid Request on: 1. Unsupported OpCode: Responder detected an unsupported OpCode. 2. Unexpected OpCode: Responder detected an error in the sequence of OpCodes, such as a missing "Last" packet. Note: there is no PSN error, thus this does not indicate a dropped packet.
sq_num_rnr	Requester - the number of RNR Naks received
sq_num_roe	Requester - number of remote operation errors NAK-Remote Operation Error on: Remote Operation Error: Responder encountered an error, (local to the responder), which prevented it from completing request.
sq_num_rree	Requester - number of RNR nak retries exceeded errors
sq_num_rsync	????
sq_num_tree	Requester - number of transport retries exceeded errors
sq_num_wrfe	Requester - number of CQEs with error. Incremented each time a CQE with error is generated
General	
num_baddb	Number of bad DoorBells
num_cqovf	Number of CQ overflows
num_eqovf	Number of EQ overflows

TEST SM Subnet Administrator:

run **osmtest** to create a inventory file and run a path record query test against the SM:
osmtest –f c
osmtest –f f –s3

sample result:

-I- Start time is : 1210010427:537245 [sec:usec]
-I- End time is : 1210010430:561821 [sec:usec]
-I- Querying 10108 Path Record queries CA to CA (rmpp)
took 0003:024576 [sec:usec]
-I- Queries to Record Ratio is 10108 records, 10108 queries : 1.00
-I- End time is : 1210010440:733957 [sec:usec]
-I- Querying 20216 Path Record queries CA to CA (rmpp)
took 0013:196712 [sec:usec]
-I- Queries to Record Ratio is 20216 records, 20216 queries : 1.00
-I- End time is : 1210010440:733993 [sec:usec]
-I- Querying 20216 Path Record queries (rmpp) took 0013:196748 [sec:usec]
OSMTEST: TEST "Stress SA" PASS

OFED verbs testing – ibv_rc_pingpong

Run **ibv_rc_pingpong** verbs test

Run ibv_rc_pingpong from one system to the other:

rping example from 2 systems (you can use eth addresses):

Start server:

```
[node-12]# ibv_rc_pingpong -s 395856
  local address:  LID 0x0001, QPN 0x080426, PSN 0xd4b107
  remote address: LID 0x0002, QPN 0x080426, PSN 0x035f2e
791712000 bytes in 0.55 seconds = 11479.58 Mbit/sec
1000 iters in 0.55 seconds = 551.74 usec/iter
```

Start client:

```
[node-13]# ibv_rc_pingpong -s 395856 node-12
  local address:  LID 0x0002, QPN 0x080426, PSN 0x035f2e
  remote address: LID 0x0001, QPN 0x080426, PSN 0xd4b107
791712000 bytes in 0.55 seconds = 11482.93 Mbit/sec
1000 iters in 0.55 seconds = 551.58 usec/iter
```

OFED rdma_cm testing – rping and ib_rdma_bw -c

rping example from 2 systems (must use IPoIB addresses):

Start server on one node:

[cst-1]# rping -s -v -C 10

Start client on another node and set –a to server IPoIB address

[cst-2]# rping -c -v -C 10 –a cst-1-ib0

You will see something like this:

ping data: rdma-ping-ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefgh
ping data: rdma-ping-1: BCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefgh
ping data: rdma-ping-2: CDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghi
ping data: rdma-ping-3: DEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghij
ping data: rdma-ping-4: EFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijkl
ping data: rdma-ping-5: FGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklm
ping data: rdma-ping-6: GHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmn
ping data: rdma-ping-7: HIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmno
ping data: rdma-ping-8: IJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnop
ping data: rdma-ping-9: JKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnop

OFED uDAPL testing – dtest and dapltest

1. dtest example from 2 systems using CMA provider
Note: must use IPoIB addresses:

Start server:

```
[cst-1]# dtest -s
29689 Running as server - ofa-v2-ib0
29689 Server waiting for connect request..
…
```

Start client on other node:

```
[cst-2]$ dtest -h cst-1-ib0
29501 Running as client - ofa-v2-ib0
29501 Server Name: cst-51-ib0
29501 Server Net Address: 192.168.0.51
29501 Waiting for connect response

29501 CONNECTED!
…

29501: DAPL Test Complete.
```

2. dapltest example from 2 systems (must use IPoIB addresses):

Start server:

```
[cst-1]# dapltest -T S
Dapltest: Service Point Ready - ofa-v2-ib0
```

Start client on other node:

```
[cst-2] dapltest -T T -V -t 2 -w 4 -i 100 -s cst-51-ib0 \
      client RW 8192 1 server RW 8192 4 \
      client RR 8192 1 server RR 8192 1 \
      client SR 8192 4 server SR 8192 2 \
      client SR 8192 3 -f server SR 8192 1 -f
```

Should see something like the following when complete:

```
Server Name: cst-51-ib0
Server Net Address: 192.168.0.51
DT_cs_Client: Starting Test...
----- Stats ---- : 2 threads, 4 EPs
Total WQE        :   22797.92 WQE/Sec
Total Time       :       0.38 sec
Total Send       :      49.15 MB -     127.33 MB/Sec
Total Recv       :      26.21 MB -      67.91 MB/Sec
Total RDMA Read  :      13.10 MB -      33.95 MB/Sec
Total RDMA Write :      32.76 MB -      84.89 MB/Sec
DT_cs_Client: ========== End of Work -- Client Exiting
```

3. How to install and build uDAPL DEBUG libraries from source

Note:

For non-debug builds (default library install) debug settings apply as follow:
DAPL_DBG_DEST=0x1 (stdout)
DAPL_DBG_TYPE=0x1 (error)
You can increase TYPE error/warnings only with non-debug libraries.
You can reset DEST to any option including syslog.

For example (OFED 1.2) on a Redhat server using the source:

cd OFED-1.2/SRPMS
rpm –i ofa_user-1.2-0.src.rpm
cd /usr/src/redhat/SOURCES/
tar zxf ofa_user-1.2.tgz
cd ofa_user-1.2/src/userspace/dapl/
./configure --enable-debug && make
cd dapl/udapl/.libs

libdaplcma.so.1.0.2 is the debug version. Save the free version on your cluster, copy this library out, and rerun your application.

sample with OFED target options when using "make install"

x86_64

```
./configure --enable-debug --prefix /usr --sysconf=/etc \
--libdir /usr/lib64 LDFLAGS=-L/usr/lib64 \
CPPFLAGS="-I/usr/include"
```

ia32

```
./configure --enable-debug --prefix /usr --sysconf=/etc \
--libdir /usr/lib LDFLAGS=-L/usr/lib \
CPPFLAGS="-I/usr/include"
```

4. Setting up debug output using a debug uDAPL library.

```
/*
 * Debug level switches
 *
 * Use these bits to enable various tracing/debug options.
 * Each bit represents debugging in a particular subsystem or
 * area of the code.
 *
 * ERR bit should always be on unless someone disables it for a
 * reason: The ERR flag is used sparingly and will print useful
 * information if it fires.
 */
typedef enum
{
    DAPL_DBG_TYPE_ERR           = 0x0001,
    DAPL_DBG_TYPE_WARN          = 0x0002,
    DAPL_DBG_TYPE_EVD           = 0x0004,
    DAPL_DBG_TYPE_CM            = 0x0008,
    DAPL_DBG_TYPE_EP            = 0x0010,
    DAPL_DBG_TYPE_UTIL          = 0x0020,
    DAPL_DBG_TYPE_CALLBACK      = 0x0040,
    DAPL_DBG_TYPE_DTO_COMP_ERR  = 0x0080,
    DAPL_DBG_TYPE_API           = 0x0100,
    DAPL_DBG_TYPE_RTN           = 0x0200,
    DAPL_DBG_TYPE_EXCEPTION     = 0x0400,
    DAPL_DBG_TYPE_SRQ           = 0x0800,
    DAPL_DBG_TYPE_CNTR          = 0x1000

} DAPL_DBG_TYPE;

/* debug logs output to stdout or syslog */
typedef enum
{
    DAPL_DBG_DEST_STDOUT        = 0x0001,
    DAPL_DBG_DEST_SYSLOG        = 0x0002,

} DAPL_DBG_DEST;
```

Example environment settings (output, dbg level):

For default settings (stdout)
export DAPL_DBG_DEST=0x1

For default settings (errors)
export DAPL_DBG_TYPE=0x1

5. Other useful DAPL environment variables:

To pickup your own private dat.conf, in lieu of /etc/dat.conf
export DAT_OVERRIDE=/your_own_directory/your_dat.conf

To increase inline rdma write/send inline data threshold (default=64 for iwarp, 200 for IB)
export DAPL_MAX_INLINE=256

OFED and Intel MPI testing

1. Intel MPI debug settings if needed:

1 = warnings if device unusable, 2 = positively confirm device used,
>2 extra levels
export I_MPI_DEBUG=10

If enabled and the rdma/rdssm device fails, the library will
failover to the next device listed in dat.conf
If disabled and the rdma/rdssm device open fails, the library
terminates the MPI job
export I_MPI_FAILOVER=disable

If set to enable and an attempt to initialize specified
fabric fails, the library falls back upon the shared memory
and/or socket fabrics. The exact combination depends on
number of processes started per node. This device ensures
that the job will run but it may not provide the highest
possible performance for the given cluster configuration.
If the I_MPI_FALLBACK_DEVICE is set to disable and an attempt
to initialize specified fabric fails, the library terminates MPI job.
export I_MPI_FALLBACK_DEVICE=disable

to reduce memory footprint (OFED 1.2 and greater)
export I_MPI_RDMA_RECV_QUEUE_SIZE=8

2. uDAPL environment setting for scale-out (>256 cores):

NOTE: If SA caching is NOT enabled set the following (otherwise stick with the defaults).

CM Address resolution timer in milliseconds, default is 4000
export DAPL_CM_ARP_TIMEOUT_MS=2000

CM Address resolution retry count, default is 15
export DAPL_CM_ARP_RETRY_COUNT=20

CM route/path record lookup timeout in milliseconds, default is 4000
For 1024 cores with a dedicated SM can process ~14,000/sec
Connections = (n * (n-1)/2), for 256x4 set to 50000
export DAPL_CM_ROUTE_TIMEOUT_MS=50000

CM route/path record lookup retries, default is 15
export DAPL_CM_ROUTE_RETRY_COUNT=10

defines queue size of DAPL connection event dispatcher.
If this variable is set, the minimum value between
size and the value obtained from the provider is
used as the size of the event queue
-- set to (2*np)+32
export I_MPI_CONN_EVD_QLEN=2080

3. Other useful options for DAT configuration and library:

export DAT_OVERRIDE=/your_own_directory/your_dat.conf
export I_MPI_DAT_LIBRARY=/your_own_library_path/your_own_libdat.so

4. iWARP device configuration

Adjust defaults for smaller resources versus IB devices as follow:

export RDMA_DEFAULT_MAX_WQE=400
export RDMA_READ_RESERVE=100
export I_MPI_RDMA_RECV_QUEUE_SIZE=10
export I_MPI_USE_DYNAMIC_CONNECTIONS=0
export DAPL_MAX_INLINE=64

4.1 Chelsio

```
1) Pull the t3fw-6.0.0.bin firmware file from service.chelsio.com.  Put
this in /lib/firmware on each node.  This will flash the new required
firmware after you installed ofed-1.3.1 and reboot.

When you load the iw_cxgb3 module, you need to set a new option named
peer2peer to 1.  This can be done by adding the following to
/etc/modprobe.conf:

options iw_cxgb3 peer2peer=1

All systems in the cluster need to set this option.  Do this just
before installing ofed-1.3.1 which will install the new iw_cxgb3 module
that uses the option.

2) Install ofed-1.3.1 making sure to include the cxgb3 support, and
then reboot the cluster. If all goes well, when you bring up the
chelsio ethX interface, you should see that is now is running 6.0
firmware.  Use ethtool -i ethX to verify this.  Also you should see the
peer2peer option in /sys/module/iw_cxgb3/parameters/*.

3) Configure /etc/dat.conf with an entry similar to this (netdev=eth2):

chelsio u1.2 nonthreadsafe default libdaplcma.so.1 dapl.1.2 "eth2 0" ""

4) The env vars I set for the user running Intel MPI includes:

export RSH=ssh
export DAPL_MAX_INLINE=64
export I_MPI_DEVICE=rdssm:chelsio
export MPIEXEC_TIMEOUT=180
export MPI_BIT_MODE=64

5) the mpd.hosts file should include the ip addresses associated with
the chelsio interfaces.
```

4.1 NetEffects

Intel MPI - DTO error codes:

DTO error operation codes via cookie =
0x00000 Write unsig
0x10000 Write sig
0x20000 Rndv sig
0x30000 Read sig
0x40000 Read sig
0x50000 Send unsig
0x60000 Send sig
0x70000 Recv sig

^^^^ = rank (4 digits)

Large Scale-out settings: uDAPL 1.2.8 and Intel MPI 3.1+

```
8 core nodes, mlx4, port 1:

I_MPI_DEVICE=rdssm:OpenIB-mlx4_0-1

DAPL_ACK_RETRY=7
DAPL_ACK_TIMER=20
DAPL_RNR_RETRY=7
DAPL_RNR_TIMER=28

I_MPI_ADJUST_ALLGATHER=3
I_MPI_CACHE_BYPASS_THRESHOLDS=16384,16384,16384,16384,16384,16384
I_MPI_FAIR_READ_SPIN_COUNT=100000000
I_MPI_FALLBACK_DEVICE=disable
I_MPI_JOB_STARTUP_TIMEOUT=10000
I_MPI_PIN_PROCESSOR_LIST=0,2,4,6,1,3,5,7
I_MPI_RDMA_MAX_MSG_SIZE=1048576
I_MPI_RDMA_READ_MAX_NUM=2
I_MPI_RDMA_RECV_QUEUE_SIZE=0
I_MPI_RDMA_REQUEST_QUEUE_SIZE=80
I_MPI_RDMA_RNDV_BUFFER_ALIGNMENT=256
I_MPI_RDMA_RNDV_WRITE=on
I_MPI_RDMA_SCALABLE_PROGRESS=1
I_MPI_RDMA_TRANSLATION_CACHE=1
I_MPI_RDMA_TRANSLATION_CACHE_MAX_MEMORY_SIZE=131072
```

uDAPL status and error codes

```
typedef enum dat_event_number
{
    DAT_DTO_COMPLETION_EVENT                    = 0x00001,
    DAT_RMR_BIND_COMPLETION_EVENT               = 0x01001,
    DAT_CONNECTION_REQUEST_EVENT                = 0x02001,
    DAT_CONNECTION_EVENT_ESTABLISHED            = 0x04001,
    DAT_CONNECTION_EVENT_PEER_REJECTED          = 0x04002,
    DAT_CONNECTION_EVENT_NON_PEER_REJECTED      = 0x04003,
    DAT_CONNECTION_EVENT_ACCEPT_COMPLETION_ERROR = 0x04004,
    DAT_CONNECTION_EVENT_DISCONNECTED           = 0x04005,
    DAT_CONNECTION_EVENT_BROKEN                 = 0x04006,
    DAT_CONNECTION_EVENT_TIMED_OUT              = 0x04007,
    DAT_CONNECTION_EVENT_UNREACHABLE            = 0x04008,
    DAT_ASYNC_ERROR_EVD_OVERFLOW                = 0x08001,
    DAT_ASYNC_ERROR_IA_CATASTROPHIC             = 0x08002,
    DAT_ASYNC_ERROR_EP_BROKEN                   = 0x08003,
    DAT_ASYNC_ERROR_TIMED_OUT                   = 0x08004,
    DAT_ASYNC_ERROR_PROVIDER_INTERNAL_ERROR     = 0x08005,
    DAT_HA_DOWN_TO_1                            = 0x08101,
    DAT_HA_UP_TO_MULTI_PATH                     = 0x08102,
    DAT_SOFTWARE_EVENT                          = 0x10001

} DAT_EVENT_NUMBER;

typedef enum dat_dto_completion_status
{
    DAT_DTO_SUCCESS                 = 0,
    DAT_DTO_ERR_FLUSHED             = 1,
    DAT_DTO_ERR_LOCAL_LENGTH        = 2,
    DAT_DTO_ERR_LOCAL_EP            = 3,
    DAT_DTO_ERR_LOCAL_PROTECTION    = 4,
    DAT_DTO_ERR_BAD_RESPONSE        = 5,
    DAT_DTO_ERR_REMOTE_ACCESS       = 6,
    DAT_DTO_ERR_REMOTE_RESPONDER    = 7,
    DAT_DTO_ERR_TRANSPORT           = 8,
    DAT_DTO_ERR_RECEIVER_NOT_READY  = 9,
    DAT_DTO_ERR_PARTIAL_PACKET      = 10,
    DAT_RMR_OPERATION_FAILED        = 11,
} DAT_DTO_COMPLETION_STATUS;
```

```
typedef enum dat_return_type
{
    DAT_SUCCESS                                 = 0x00000000,
    /* operation aborted because IA was closed or EVD destroyedvv */
    DAT_ABORT                                   = 0x00010000,
    /* specified Connection Qualifier was in use.                 */
    DAT_CONN_QUAL_IN_USE                        = 0x00020000,
    /* operation failed due to resource limitations.              */
    DAT_INSUFFICIENT_RESOURCES                  = 0x00030000,
    /* Provider internal error                                    */
    DAT_INTERNAL_ERROR                          = 0x00040000,
    /* One of the DAT handles was invalid.                        */
    DAT_INVALID_HANDLE                          = 0x00050000,
    /* One of the parameters was invalid.                         */
    DAT_INVALID_PARAMETER                       = 0x00060000,
    /* One of the parameters was invalid. There are Event Streams
     * associated with the Event Dispatcher feeding it.           */
    DAT_INVALID_STATE                           = 0x00070000,
    /* The size of the receiving buffer is too small for sending  *
     * buffer data.  The size of the local buffer is too small for*
     * the data of the remote buffer.                             */
    DAT_LENGTH_ERROR                            = 0x00080000,
    /* The requested Model was not supported by the Provider.     */
    DAT_MODEL_NOT_SUPPORTED                     = 0x00090000,
    /* The specified IA name was not found in the list of         *
     * registered Providers.                                      */
    DAT_PROVIDER_NOT_FOUND                      = 0x000A0000,
    /* Protection violation for local or remote memory access     *
     * Protection Zone mismatch between an LMR of one of the      *
     * local_iov segments and the local Endpoint.                 */
    DAT_PRIVILEGES_VIOLATION                    = 0x000B0000,
    /* Privileges violation for local or remote memory access. One*
     * of the LMRs used in local_iov was either invalid or did not*
     * have local read privileges.                                */
    DAT_PROTECTION_VIOLATION                    = 0x000C0000,
    /* The operation timed out without a notification.            */
    DAT_QUEUE_EMPTY                             = 0x000D0000,
    /* The Event Dispatcher queue is full.                        */
    DAT_QUEUE_FULL                              = 0x000E0000,
    /* The operation timed out. uDAPL ONLY                        */
    DAT_TIMEOUT_EXPIRED                         = 0x000F0000,
    /* The provider name was already registered                   */
    DAT_PROVIDER_ALREADY_REGISTERED             = 0x00100000,
    /* The provider is "in-use" and cannot be closed at this time */
    DAT_PROVIDER_IN_USE                         = 0x00110000,
    /* The requested remote address is not valid or not reachable */
    DAT_INVALID_ADDRESS                         = 0x00120000,
    /* [Unix only] dat_evd_wait, dat_cno_wait interrupted         */
    DAT_INTERRUPTED_CALL                        = 0x00130000,
    /* No Connection Qualifiers are available                     */
    DAT_CONN_QUAL_UNAVAILABLE                   = 0x00140000,
    /* specified IP Port was in use. */
    DAT_PORT_IN_USE                             = 0x00160000,
    /* specified COMM not supported. */
    DAT_COMM_NOT_SUPPORTED                      = 0x00170000,
    /* Provider does not support the operation yet */
    DAT_NOT_IMPLEMENTED                         = 0x3FFF0000
} DAT_RETURN_TYPE;
```

OFA IB status and error codes

IB verbs work completion status:

```
enum ib_wc_status {
    IB_WC_SUCCESS,
    IB_WC_LOC_LEN_ERR,
    IB_WC_LOC_QP_OP_ERR,
    IB_WC_LOC_EEC_OP_ERR,
    IB_WC_LOC_PROT_ERR,
    IB_WC_WR_FLUSH_ERR,
    IB_WC_MW_BIND_ERR,
    IB_WC_BAD_RESP_ERR,
    IB_WC_LOC_ACCESS_ERR,
    IB_WC_REM_INV_REQ_ERR,
    IB_WC_REM_ACCESS_ERR,
    IB_WC_REM_OP_ERR,
    IB_WC_RETRY_EXC_ERR,
    IB_WC_RNR_RETRY_EXC_ERR,
    IB_WC_LOC_RDD_VIOL_ERR,
    IB_WC_REM_INV_RD_REQ_ERR,
    IB_WC_REM_ABORT_ERR,
    IB_WC_INV_EECN_ERR,
    IB_WC_INV_EEC_STATE_ERR,
    IB_WC_FATAL_ERR,
    IB_WC_RESP_TIMEOUT_ERR,
    IB_WC_GENERAL_ERR
};
```

IB verbs async event type:

```
enum ib_event_type {
    IB_EVENT_CQ_ERR,
    IB_EVENT_QP_FATAL,
    IB_EVENT_QP_REQ_ERR,
    IB_EVENT_QP_ACCESS_ERR,
    IB_EVENT_COMM_EST,
    IB_EVENT_SQ_DRAINED,
    IB_EVENT_PATH_MIG,
    IB_EVENT_PATH_MIG_ERR,
    IB_EVENT_DEVICE_FATAL,
    IB_EVENT_PORT_ACTIVE,
    IB_EVENT_PORT_ERR,
    IB_EVENT_LID_CHANGE,
    IB_EVENT_PKEY_CHANGE,
    IB_EVENT_SM_CHANGE,
    IB_EVENT_SRQ_ERR,
    IB_EVENT_SRQ_LIMIT_REACHED,
    IB_EVENT_QP_LAST_WQE_REACHED,
    IB_EVENT_CLIENT_REREGISTER
};
```

Vendor error codes: (cqe->vendor_err): see HCA vendor for definitions.

RDMA_CMA event type:

```
enum rdma_cm_event_type {
      RDMA_CM_EVENT_ADDR_RESOLVED,
      RDMA_CM_EVENT_ADDR_ERROR,
      RDMA_CM_EVENT_ROUTE_RESOLVED,
      RDMA_CM_EVENT_ROUTE_ERROR,
      RDMA_CM_EVENT_CONNECT_REQUEST,
      RDMA_CM_EVENT_CONNECT_RESPONSE,
      RDMA_CM_EVENT_CONNECT_ERROR,
      RDMA_CM_EVENT_UNREACHABLE,
      RDMA_CM_EVENT_REJECTED,
      RDMA_CM_EVENT_ESTABLISHED,
      RDMA_CM_EVENT_DISCONNECTED,
      RDMA_CM_EVENT_DEVICE_REMOVAL,
      RDMA_CM_EVENT_MULTICAST_JOIN,
      RDMA_CM_EVENT_MULTICAST_ERROR
};
```

RDMA_CM status == IB_CM reject reason

```
enum ib_cm_rej_reason {
      IB_CM_REJ_NO_QP                         = 1,
      IB_CM_REJ_NO_EEC                        = 2,
      IB_CM_REJ_NO_RESOURCES                  = 3,
      IB_CM_REJ_TIMEOUT                       = 4,
      IB_CM_REJ_UNSUPPORTED                   = 5,
      IB_CM_REJ_INVALID_COMM_ID               = 6,
      IB_CM_REJ_INVALID_COMM_INSTANCE         = 7,
      IB_CM_REJ_INVALID_SERVICE_ID            = 8,
      IB_CM_REJ_INVALID_TRANSPORT_TYPE        = 9,
      IB_CM_REJ_STALE_CONN                    = 10,
      IB_CM_REJ_RDC_NOT_EXIST                 = 11,
      IB_CM_REJ_INVALID_GID                   = 12,
      IB_CM_REJ_INVALID_LID                   = 13,
      IB_CM_REJ_INVALID_SL                    = 14,
      IB_CM_REJ_INVALID_TRAFFIC_CLASS         = 15,
      IB_CM_REJ_INVALID_HOP_LIMIT             = 16,
      IB_CM_REJ_INVALID_PACKET_RATE           = 17,
      IB_CM_REJ_INVALID_ALT_GID               = 18,
      IB_CM_REJ_INVALID_ALT_LID               = 19,
      IB_CM_REJ_INVALID_ALT_SL                = 20,
      IB_CM_REJ_INVALID_ALT_TRAFFIC_CLASS     = 21,
      IB_CM_REJ_INVALID_ALT_HOP_LIMIT         = 22,
      IB_CM_REJ_INVALID_ALT_PACKET_RATE       = 23,
      IB_CM_REJ_PORT_CM_REDIRECT              = 24,
      IB_CM_REJ_PORT_REDIRECT                 = 25,
      IB_CM_REJ_INVALID_MTU                   = 26,
      IB_CM_REJ_INSUFFICIENT_RESP_RESOURCES   = 27,
      IB_CM_REJ_CONSUMER_DEFINED              = 28,
      IB_CM_REJ_INVALID_RNR_RETRY             = 29,
      IB_CM_REJ_DUPLICATE_LOCAL_COMM_ID       = 30,
      IB_CM_REJ_INVALID_CLASS_VERSION         = 31,
      IB_CM_REJ_INVALID_FLOW_LABEL            = 32,
      IB_CM_REJ_INVALID_ALT_FLOW_LABEL        = 33
   }
```

Additional diagnostic codes – mlx4 only

mlx4 specific diag counters: (print example follows)

for i in /sys/class/infiniband/mlx4_0/diag_counters/*; do echo -n $i:;cat $i;done

Counter name	Description
clear_diag	echo 1 > (clear all counters)
RESPONDER	
rq_num_lae	Responder - number of local access errors
rq_num_leeoe	????
rq_num_lle	Responder - number of local length errors
rq_num_lpe	Responder - number of local protection errors
rq_num_lqpoe	Responder - number local QP operation error
rq_num_mce	Responder - number of bad multicast packets received
rq_num_oos	Responder - number of out of sequence requests received
rq_num_rae	Responder - number of remote access errors. R_Key Violation Responder detected an R_Key violation while executing an RDMA request. NAK may or may not be sent.
rq_num_rire	Responder - number of remote invalid request errors. NAK may or may not be sent. 1. QP Async Affiliated Error: Unsupported or Reserved OpCode (RC only): Inbound request OpCode was either reserved, or was for a function not supported by this QP. (E.g. RDMA or ATOMIC on QP not set up for this). 2. Misaligned ATOMIC: VA does not point to an aligned address on an atomic operation. 3. Too many RDMA READ or ATOMIC Requests: There were more requests received and not ACKed than allowed. 4. Out of Sequence OpCode, current packet is "First" or "Only": The Responder detected an error in the sequence of OpCodes; a missing "Last" packet 5. Out of Sequence OpCode, current packet is not "First" or "Only": The Responder detected an error in the sequence of OpCodes; a missing "First" packet 6. Local Length Error: Inbound "Send" request message exceeded the responder.s available buffer space. 7. Length error: RDMA WRITE request message contained too much or too little pay-load data compared to the DMA length advertised in the first or only packet. 8. Length error: Payload length was not consistent with the opcode: a: 0 byte <= "only" <= PMTU bytes b: ("first" or "middle") == PMTU bytes c: 1byte <= "last" <= PMTU bytes 9. Length error: Inbound message exceeded the size supported by the CA port.
rq_num_rnr	Responder - the number of RNR Naks sent
rq_num_rsync	????
rq_num_ucsdprd	The number of UC packets silently discarded on the receive queue due to lack of receive descriptor.
rq_num_udsdprd	The number of UD packets silently discarded on the receive queue due to lack of receive descriptor.
rq_num_wrfe	Responder - number of CQEs with error. Incremented each time a CQE with error is generated
REQUESTER	
sq_num_bre	Requester - number of bad response errors
sq_num_ieecne	????
sq_num_ieecse	????

sq_num_leeoe	????
sq_num_lle	Requester - number of local length errors
sq_num_lpe	Requester - number of local protection errors
sq_num_lqpoe	Requester - number local QP operation error
sq_num_mwbe	Requester - number of Memory Window bind errors
sq_num_oos	Requester - number of out of sequence Naks received
sq_num_rabrte	Requester - number of remote aborted errors
sq_num_rae	Requester - number of remote access errors
	NAK-Remote Access Error on: R_Key Violation: Responder detected an invalid R_Key while executing an RDMA Request.
sq_num_rire	Requester - number of remote invalid request errors. NAK-Invalid Request on: 1. Unsupported OpCode: Responder detected an unsupported OpCode. 2. Unexpected OpCode: Responder detected an error in the sequence of OpCodes, such as a missing "Last" packet. Note: there is no PSN error, thus this does not indicate a dropped packet.
sq_num_rnr	Requester - the number of RNR Naks received
sq_num_roe	Requester - number of remote operation errors NAK-Remote Operation Error on: Remote Operation Error: Responder encountered an error, (local to the responder), which prevented it from completing request.
sq_num_rree	Requester - number of RNR nak retries exceeded errors
sq_num_rsync	????
sq_num_tree	Requester - number of transport retries exceeded errors
sq_num_wrfe	Requester - number of CQEs with error. Incremented each time a CQE with error is generated
General	
num_baddb	Number of bad DoorBells
num_cqovf	Number of CQ overflows
num_eqovf	Number of EQ overflows

not provided via linux diag_counters

rq_num_roe	NAK-Remote Operation Error on: 1. Malformed WQE: Responder detected a malformed Receive Queue WQE while processing the packet. 2. Remote Operation Error: Responder encountered an error, (local to the responder), which prevented it from completing the request.