

```
#
# This repository is a Transparency and Reproducibility Initiative data
# artifact for the publication: "An Evaluation of Ethernet Performance
# for
# Scientific Workloads", by Joseph P. Kenny, Jeremiah J. Wilke,
# Craig D. Ulmer, Gavin M. Baker, Samuel Knight and Jerrold A. Friesen
#
```

```
#
# Testbed Hardware/Software Details
#
CPU                2x Intel E5-2683 v4 (32 total cores)
OS/Kernel          CentOS Linux release 7.7.1908
                   Linux 3.10.0-1062.9.1.el7.x86_64
NIC                Mellanox ConnectX-5 MT27800
                   hw_ver: 0x0, board_id: MT_0000000011
NIC Drivers/Firmware MLNX_OFED_LINUX-4.6-1.0.1.1, fw_ver: 16.24.1000
Switch OS          Mellanox Onyx, version 3.8.2004 (3.7.1134 for
SN2700)
```

```
#
# Benchmark Details
#
Sample launch scripts are provided in benchmarks directory.
Compiled with gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39)
MPI apps used Open MPI version 4.0.4.
+ benchmark launch scripts specify transports used
+ https://www.open-mpi.org/
MPI point-to-point tests
+ osu_bw and osu_latency from mvapich2 distribution
+ https://mvapich.cse.ohio-state.edu/
Incast benchmark script
+ listing provided in Kenny, Joseph P., and Craig D. Ulmer. "RoCE:
Promising
  Technology for Ethernet as a High Performance Networking Fabric."
  No. SAND2019-13444. Sandia National Lab.(SNL-CA),
  Livermore, CA (United States), 2019.
Halo3D motif
+ ember/mpi/halo3d-26/
+ https://github.com/sstsimulator/ember
FFT proxy
+ chatterbug/subcom3d-a2a
+ https://github.com/hpcgroup/chatterbug
HPL benchmark
+ https://www.netlib.org/benchmark/hpl/
+ CCFLAGS = -O3 -w -z noexecstack -z relro -z now -Wall
+ intel-mkl-2017.2.174
```

```
# Switch Configurations
```

Sample switch configurations are found in the switch-configs directory.
Ports connecting core to leaf switches are members of a port-channel.
"sample-config-ethernet.txt" shows basic ethernet configuration.
"sample-config-pfc.txt" shows commands that enable PFC.
"sample-config-ets.txt" shows commands that enable ETS.

host254
host266
host235
host277
host226
host260
host275
host244
host207
host281
host268
host203
host262
host214
host247
host253
host288
host250
host231
host271
host278
host200
host221
host196
host251
host199
host215
host274
host283
host213
host195
host256
host261
host212
host198
host284
host225
host220
host219
host233
host273
host282
host259
host228
host242
host240
host229
host248

```
#!/bin/bash
```

```
MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
```

```
EXE_DIR=$HOME/testbed-openmpi
```

```
$MPIRUN --hostfile hosts-halo-random.txt --mca pml ucx --mca btl  
^tcp,openib -x UCX_NET_DEVICES=mlx5_0:1 -x UCX_IB_GID_INDEX=5 -np 48 -N 1  
$EXE_DIR/halo3d-26/halo3d-26 -pex 4 -pey 4 -pez 3 -nx 100 -ny 100 -nz 100  
-vars 8 -sleep 216000 -iterations 30000 -print 1 >& stdout.app1 &
```

```
sleep 10
```

```
$MPIRUN --hostfile hosts-subcom-random.txt --mca pml ucx --mca btl  
^tcp,openib -x UCX_NET_DEVICES=mlx5_0:1 -x UCX_IB_GID_INDEX=5 -np 48 -N 1  
$EXE_DIR/subcom-a2a/subcom-a2a -pex 4 -pey 4 -pez 3 -nx 10000 -ny 10000 -  
nz 10000 -iterations 16000 -print 1 >& stdout.app2 &
```

```
wait
```

```
wait
```

```
#!/bin/bash
```

```
MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
```

```
EXE_DIR=$HOME/testbed-openmpi
```

```
$MPIRUN --hostfile hosts-halo-random.txt --mca btl_tcp_if_include  
eth2.100 --mca pml ^ucx --mca btl ^openib,uct -np 48 -N 1  
$EXE_DIR/halo3d-26/halo3d-26 -pex 4 -pey 4 -pez 3 -nx 100 -ny 100 -nz 100  
-vars 8 -sleep 216000 -iterations 30000 -print 1 >& stdout.app1 &
```

```
sleep 10
```

```
$MPIRUN --hostfile hosts-subcom-random.txt --mca btl_tcp_if_include  
eth2.100 --mca pml ^ucx --mca btl ^openib,uct -np 48 -N 1  
$EXE_DIR/subcom-a2a/subcom-a2a -pex 4 -pey 4 -pez 3 -nx 10000 -ny 10000 -  
nz 10000 -iterations 16000 -print 1 >& stdout.app2 &
```

```
wait
```

```
wait
```

```
host255
```

```
host243
```

```
host211
```

```
host204
```

```
host286
```

```
host249
```

```
host258
```

```
host279
```

```
host245
```

```
host217
```

```
host285
```

```
host272
```

```
host205
```

```
host201
```

```
host270
```

```
host234
```

```
host241
host287
host218
host239
host263
host237
host206
host227
host265
host202
host246
host238
host208
host216
host236
host257
host197
host264
host210
host223
host267
host224
host209
host230
host276
host193
host232
host222
host280
host194
host252
host269
#!/bin/bash
```

```
size="8"
#size="8192"
#size="81920"
#size="819200"
#size="8388608"
```

```
python traffic.py --test iperf3 --sources host250 --destination host249 -
-max_streams 10 --client-options "-t 5 -l $size"
mv traffic.out s$size-src1.out
python traffic.py --test iperf3 --sources host[250,251] --destination
host249 --max_streams 10 --client-options "-t 5 -l $size"
mv traffic.out s$size-src2.out
python traffic.py --test iperf3 --sources host[250,251,252] --destination
host249 --max_streams 10 --client-options "-t 5 -l $size"
mv traffic.out s$size-src3.out
python traffic.py --test iperf3 --sources host[250,251,252,253] --
destination host249 --max_streams 10 --client-options "-t 5 -l $size"
mv traffic.out s$size-src4.out
#!/bin/bash
```

```

size="8"
#size="8192"
#size="81920"
#size="819200"
#size="8388608"

python ib-traffic.py --test ib_write_bw --sources host250 --destination
host249 --max_streams 10 --options "-D 5 -s $size"
mv test-ib.out s$size-src1.out
python ib-traffic.py --test ib_write_bw --sources host[250,251] --
destination host249 --max_streams 10 --options "-D 5 -s $size"
mv test-ib.out s$size-src2.out
python ib-traffic.py --test ib_write_bw --sources host[250,251,252] --
destination host249 --max_streams 10 --options "-D 5 -s $size"
mv test-ib.out s$size-src3.out
python ib-traffic.py --test ib_write_bw --sources host[250,251,252,253] -
-destination host249 --max_streams 10 --options "-D 5 -s $size"
mv test-ib.out s$size-src4.out
#
# host id ordering
#

```

The incast test was run on an early version of the testbed with 32 nodes connected directly to the core switch. All incast tests are between nodes connected to this same core switch. There is no even/odd split of host ids as in other tests.

```

$HOME/install/openmpi-4.0.4-testbed/bin/mpirun --hostfile hosts.txt --mca
btl_tcp_if_include eth2.100 --mca pml ^ucx --mca btl ^openib,uct -np 3072
-N 32 ./xhpl

```

HPLinpack benchmark input file

Innovative Computing Laboratory, University of Tennessee

```

HPL.out          output file name (if any)
6                device out (6=stdout,7=stderr,file)
1 #4             # of problems sizes (N)
491520 #32 * 96 * 10 * 8
1                # of NBs
256 #1 2 3 4     NBs
0                PMAP process mapping (0=Row-,1=Column-major)
1 #3             # of process grids (P x Q)
32 #16 #2 1 4    Ps
96 #20 #2 4 1    Qs
16.0            threshold
1                # of panel fact
0                PFACTs (0=left, 1=Crout, 2=Right)
1                # of recursive stopping criterium
4                NBMINs (>= 1)
1                # of panels in recursion
2                NDIVs
1                # of recursive panel fact.
0                RFACTs (0=left, 1=Crout, 2=Right)
1                # of broadcast
0                BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1                # of lookahead depth
0                DEPTHS (>=0)

```

```
2          SWAP (0=bin-exch,1=long,2=mix)
64         swapping threshold
0          L1 in (0=transposed,1=no-transposed) form
0          U  in (0=transposed,1=no-transposed) form
1          Equilibration (0=no,1=yes)
8          memory alignment in double (> 0)
host193
host194
host195
host196
host197
host198
host199
host200
host201
host202
host203
host204
host205
host206
host207
host208
host209
host210
host211
host212
host213
host214
host215
host216
host217
host218
host219
host220
host221
host222
host223
host224
host225
host226
host227
host228
host229
host230
host231
host232
host233
host234
host235
host236
host237
host238
host239
host240
```

```
host241
host242
host243
host244
host245
host246
host247
host248
host249
host250
host251
host252
host253
host254
host255
host256
host257
host258
host259
host260
host261
host262
host263
host264
host265
host266
host267
host268
host269
host270
host271
host272
host273
host274
host275
host276
host277
host278
host279
host280
host281
host282
host283
host284
host285
host286
host287
host288
$HOME/install/openmpi-4.0.4-testbed/bin/mpirun --hostfile hosts.txt --mca
pml ucx --mca btl ^tcp,openib -x UCX_NET_DEVICES=mlx5_0:1 -x
UCX_IB_GID_INDEX=5 -np 3072 -N 32 ./xhpl
#!/bin/bash
```

```
MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
```

```
EXE_DIR=$HOME/testbed-openmpi
```

```
$MPIRUN --hostfile hosts-halo-random.txt --mca pml ucx --mca btl  
^tcp,openib -x UCX_NET_DEVICES=mlx5_0:1 -x UCX_IB_GID_INDEX=5 -np 48 -N 1  
$EXE_DIR/halo3d-26/halo3d-26 -pex 4 -pey 4 -pez 3 -nx 100 -ny 100 -nz 100  
-vars 8 -sleep 216000 -iterations 16000 -print 1 >& stdout.app1  
CXX=$HOME/install/openmpi-4.0.4-testbed/bin/mpicxx  
CXXFLAGS=-O3 -std=c++11 -fPIC  
CPPFLAGS=-I../util  
LIBS=
```

```
halo3d-26: halo3d-26.cc
```

```
$(CXX) $(CPPFLAGS) $(CXXFLAGS) -o $@ $< $(LIBS)
```

```
clean:
```

```
rm halo3d-26
```

```
host254  
host266  
host235  
host277  
host226  
host260  
host275  
host244  
host207  
host281  
host268  
host203  
host262  
host214  
host247  
host253  
host288  
host250  
host231  
host271  
host278  
host200  
host221  
host196  
host251  
host199  
host215  
host274  
host283  
host213  
host195  
host256  
host261  
host212  
host198  
host284  
host225  
host220
```

```

host219
host233
host273
host282
host259
host228
host242
host240
host229
host248
MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
EXE_DIR=$HOME/testbed-openmpi

$MPIRUN --hostfile ../../hosts-halo-random.txt --mca btl_tcp_if_include
eth2.100 --mca pml ^ucx --mca btl ^openib,uct -np 48 -N 1
$EXE_DIR/halo3d-26/halo3d-26 -pex 4 -pey 4 -pez 3 -nx 100 -ny 100 -nz 100
-vars 8 -sleep 216000 -iterations 16000 -print 1 >& stdout.appl
#!/bin/bash

MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
EXE_DIR=$HOME/testbed-openmpi/osu

$MPIRUN --hostfile ./hosts-intra.txt --mca pml ucx --mca btl ^tcp,openib
-x UCX_NET_DEVICES=mlx5_0:1 -x UCX_IB_GID_INDEX=5 -np 2 -N 1
$EXE_DIR/osu_bw >& stdout.bw
#!/bin/bash

MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
EXE_DIR=$HOME/testbed-openmpi/osu

$MPIRUN --hostfile ./hosts-intra.txt --mca pml ucx --mca btl ^tcp,openib
-x UCX_NET_DEVICES=mlx5_0:1 -x UCX_IB_GID_INDEX=5 -np 2 -N 1
$EXE_DIR/osu_latency >& stdout.latency
#!/bin/bash

MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
EXE_DIR=$HOME/testbed-openmpi/osu

$MPIRUN --mca btl_tcp_if_include eth2.100 --mca pml ^ucx --mca btl
^openib,uct --hostfile ./hosts-intra.txt -np 2 -N 1 $EXE_DIR/osu_latency
>& stdout.latency
$HOME/install/openmpi-4.0.4-testbed/bin/mpicc -I ./ osu_pt2pt.c osu_bw.c
-o osu_bw
$HOME/install/openmpi-4.0.4-testbed/bin/mpicc -I ./ osu_pt2pt.c
osu_latency.c -o osu_latency
host193
host195
#!/bin/bash

MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
EXE_DIR=$HOME/testbed-openmpi/osu

```

```
$MPIRUN --mca btl_tcp_if_include eth2.100 --mca pml ^ucx --mca btl
^openib,uct --hostfile ./hosts-intra.txt -np 2 -N 1 $EXE_DIR/osu_bw >&
stdout.bw
#!/bin/bash
```

```
MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
EXE_DIR=$HOME/testbed-openmpi
```

```
$MPIRUN --hostfile hosts-subcom-random.txt --mca pml ucx --mca btl
^tcp,openib -x UCX_NET_DEVICES=mlx5_0:1 -x UCX_IB_GID_INDEX=5 -np 48 -N 1
$EXE_DIR/subcom-a2a/subcom-a2a -pex 4 -pey 4 -pez 3 -nx 10000 -ny 10000 -
nz 10000 -iterations 16000 -print 1 >& stdout.app2
CXX=$HOME/install/openmpi-4.0.4-testbed/bin/mpicxx
CXXFLAGS= -O3 -std=c++11 -fPIC
CPPFLAGS= -I../util
```

```
all: subcom-a2a
```

```
subcom-a2a : subcom-a2a.cc
            $(CXX) $(CXXFLAGS) $(CPPFLAGS) -o $@ $<
```

```
clean:
```

```
            rm -f subcom-a2a$(SUFFIX)
#!/bin/bash
```

```
MPIRUN=$HOME/install/openmpi-4.0.4-testbed/bin/mpirun
EXE_DIR=$HOME/testbed-openmpi
```

```
$MPIRUN --hostfile hosts-subcom-random.txt --mca btl_tcp_if_include
eth2.100 --mca pml ^ucx --mca btl ^openib,uct -np 48 -N 1
$EXE_DIR/subcom-a2a/subcom-a2a -pex 4 -pey 4 -pez 3 -nx 10000 -ny 10000 -
nz 10000 -iterations 16000 -print 1 >& stdout.app2
```

```
host255
host243
host211
host204
host286
host249
host258
host279
host245
host217
host285
host272
host205
host201
host270
host234
host241
host287
host218
host239
host263
host237
```

```
host206
host227
host265
host202
host246
host238
host208
host216
host236
host257
host197
host264
host210
host223
host267
host224
host209
host230
host276
host193
host232
host222
host280
host194
host252
host269
#
# host id ordering
#
```

Testbed host ids run sequentially from host193 to host288. Due to physical layout of the testbed racks, even and odd blocks of 12 host ids are connected to each leaf switch.

For example, one switch has the following block of 12 odd ids:

```
host193
host195
host197
host199
host201
host203
host205
host207
host209
host211
host213
host215
```

Another switch has the following block of 12 even ids:

```
host194
host196
host198
```

```
host200
host202
host204
host206
host208
host210
host212
host214
host216
##
## DCBX ETS configuration
##
interface ethernet 1/2 traffic-class 0 dcb ets wrr 50
interface ethernet 1/2 traffic-class 1 dcb ets wrr 50
interface ethernet 1/2 traffic-class 2 dcb ets wrr 0
interface ethernet 1/2 traffic-class 3 dcb ets wrr 0
interface ethernet 1/2 traffic-class 4 dcb ets wrr 0
interface ethernet 1/2 traffic-class 5 dcb ets wrr 0
interface ethernet 1/2 traffic-class 6 dcb ets wrr 0
interface ethernet 1/2 traffic-class 7 dcb ets wrr 0
interface ethernet 1/4 traffic-class 0 dcb ets wrr 50
interface ethernet 1/4 traffic-class 1 dcb ets wrr 50
interface ethernet 1/4 traffic-class 2 dcb ets wrr 0
interface ethernet 1/4 traffic-class 3 dcb ets wrr 0
interface ethernet 1/4 traffic-class 4 dcb ets wrr 0
interface ethernet 1/4 traffic-class 5 dcb ets wrr 0
interface ethernet 1/4 traffic-class 6 dcb ets wrr 0
interface ethernet 1/4 traffic-class 7 dcb ets wrr 0
interface ethernet 1/6 traffic-class 0 dcb ets wrr 50
interface ethernet 1/6 traffic-class 1 dcb ets wrr 50
interface ethernet 1/6 traffic-class 2 dcb ets wrr 0
interface ethernet 1/6 traffic-class 3 dcb ets wrr 0
interface ethernet 1/6 traffic-class 4 dcb ets wrr 0
interface ethernet 1/6 traffic-class 5 dcb ets wrr 0
interface ethernet 1/6 traffic-class 6 dcb ets wrr 0
interface ethernet 1/6 traffic-class 7 dcb ets wrr 0
interface ethernet 1/8-1/16 traffic-class 0 dcb ets wrr 50
interface ethernet 1/8-1/16 traffic-class 1 dcb ets wrr 50
interface ethernet 1/8-1/16 traffic-class 2 dcb ets wrr 0
interface ethernet 1/8-1/16 traffic-class 3 dcb ets wrr 0
interface ethernet 1/8-1/16 traffic-class 4 dcb ets wrr 0
interface ethernet 1/8-1/16 traffic-class 5 dcb ets wrr 0
interface ethernet 1/8-1/16 traffic-class 6 dcb ets wrr 0
interface ethernet 1/8-1/16 traffic-class 7 dcb ets wrr 0
interface port-channel 1 traffic-class 0 dcb ets wrr 50
interface port-channel 1 traffic-class 1 dcb ets wrr 50
interface port-channel 1 traffic-class 2 dcb ets wrr 0
interface port-channel 1 traffic-class 3 dcb ets wrr 0
interface port-channel 1 traffic-class 4 dcb ets wrr 0
interface port-channel 1 traffic-class 5 dcb ets wrr 0
interface port-channel 1 traffic-class 6 dcb ets wrr 0
interface port-channel 1 traffic-class 7 dcb ets wrr 0
##
## DCBX PFC configuration
```

##

```
dcb priority-flow-control enable force
dcb priority-flow-control priority 0 enable
dcb priority-flow-control priority 1 enable
dcb priority-flow-control priority 2 enable
dcb priority-flow-control priority 3 enable
dcb priority-flow-control priority 4 enable
dcb priority-flow-control priority 5 enable
dcb priority-flow-control priority 6 enable
dcb priority-flow-control priority 7 enable
interface ethernet 1/2 dcb priority-flow-control mode on force
interface ethernet 1/4 dcb priority-flow-control mode on force
interface ethernet 1/6 dcb priority-flow-control mode on force
interface ethernet 1/8-1/16 dcb priority-flow-control mode on force
interface port-channel 1-8 dcb priority-flow-control mode on force
```

##

Interface Ethernet configuration

##

```
interface port-channel 1-8
interface ethernet 1/1-1/16 mtu 4500 force
interface port-channel 1 mtu 4500 force
interface ethernet 1/1 channel-group 1 mode on
interface ethernet 1/2 switchport mode trunk
interface ethernet 1/3 channel-group 1 mode on
interface ethernet 1/4 switchport mode trunk
interface ethernet 1/5 channel-group 1 mode on
interface ethernet 1/6 switchport mode trunk
interface ethernet 1/7 channel-group 1 mode on
interface ethernet 1/8-1/16 switchport mode trunk
interface port-channel 1 switchport mode trunk
interface port-channel 1 description test-interface
```