Installation

===============================================================================

-After cloning the repository a perftest directory should appear in your current

directory

-Cloning example :

git clone <URL>, In our situation its --> git clone <https://github.com/linux-rdma/perftest.git>

-After cloning, Follow this commands:

-cd perftest/

yum install pciutils-devel

-./autogen.sh

-./configure Note:If you want to install in a specific directory use the optional flag --prefix=<Directory path> , e.g: ./configure --prefix=<Directory path>

-make

-make install

-All of the tests will appear in the perftest directory and in the install directory.

===============================================================================

3. Notes on Testing Methodology

===============================================================================

- The benchmarks use the CPU cycle counter to get time stamps without context

switch. Some CPU architectures (e.g., Intel's 80486 or older PPC) do not

have such capability.

- The latency benchmarks measure round-trip time but report half of that as one-way

latency. This means that the results may not be accurate for asymmetrical configurations.

- On all unidirectional bandwidth benchmarks, the client measures the bandwidth.

On bidirectional bandwidth benchmarks, each side measures the bandwidth of

the traffic it initiates, and at the end of the measurement period, the server

reports the result to the client, who combines them together.

- Latency tests report minimum, median and maximum latency results.

The median latency is typically less sensitive to high latency variations,

compared to average latency measurement.

Typically, the first value measured is the maximum value, due to warmup effects.

- Long sampling periods have very limited impact on measurement accuracy.

The default value of 1000 iterations is pretty good.

Note that the program keeps data structures with memory footprint proportional

to the number of iterations. Setting a very high number of iteration may

have negative impact on the measured performance which are not related to

the devices under test.

If a high number of iterations is strictly necessary, it is recommended to

use the -N flag (No Peak).

- Bandwidth benchmarks may be run for a number of iterations, or for a fixed duration.

Use the -D flag to instruct the test to run for the specified number of seconds.

The --run\_infinitely flag instructs the program to run until interrupted by

the user, and print the measured bandwidth every 5 seconds.

- The "-H" option in latency benchmarks dumps a histogram of the results.

See xgraph, ygraph, r-base (<http://www.r-project.org/>), PSPP, or other

statistical analysis programs.

\*\*\* IMPORTANT NOTE:

When running the benchmarks over an Infiniband fabric,

a Subnet Manager must run on the switch or on one of the

nodes in your fabric, prior to starting the benchmarks.

Architectures tested: i686, x86\_64, ia64