



# CHAPTER 6

## uDAPL

---

This chapter describes uDAPL and includes the following sections:

- [Introduction, page 6-1](#)
- [uDAPL Test Performance, page 6-1](#)
- [Compiling uDAPL Programs, page 6-4](#)



Note

---

See the “[Root and Non-root Conventions in Examples](#)” section on [page ix](#) for details about the significance of prompts used in the examples in this chapter.

---

## Introduction

uDAPL defines a single set of user-level APIs for all RDMA-capable transports. uDAPL also defines a transport-independent and platform-standard set of APIs that takes advantage of RDMA capabilities such as those present in IB. To obtain uDAPL, install the drivers. No additional configuration is required to use uDAPL.

For additional details about uDAPL, go to the following URL:

<http://www.datcollaborative.org>

## uDAPL Test Performance

This section describes the uDAPL test performance. The utility to test uDAPL performance is included with the RPMs after the host drivers are installed.

The uDAPL test utility is located in the following directory:

`/usr/local/topspin/bin/`

The uDAPL test must be run on a server and a client host.

## uDAPL Throughput Test Performance

The Throughput test measures RDMA WRITE throughput using uDAPL. To perform a uDAPL Throughput test performance, perform the following steps:

**Step 1** Start the Throughput test on the server host. The syntax for the server is as follows:

```
/usr/local/topspin/bin/thru_server.x device_name RDMA_size iterations batch_size
```

The following example shows how to start the Throughput test on the server host:

```
host1$ /usr/local/topspin/bin/thru_server.x ib0 262144 500 100
RDMA throughput server started on ib0
```

- **ib0** is the name of the device.
- **262144** is the size in bytes of the RDMA WRITE.
- **500** is the number of RDMA operations to perform for the test.
- **100** is the number of RDMA operations to perform before waiting for completions.

The server starts and then waits for the client to start.

**Step 2** Start the Throughput test on the client. The syntax for the client is as follows:

```
/usr/local/topspin/bin/thru_client.x device_name server_IP_address RDMA_size
```

The following example shows how to start the Throughput test on the client:

```
host2$ /usr/local/topspin/bin/thru_client.x ib0 192.168.0.1 262144
Server Name: 192.168.0.1
Server Net Address: 192.168.0.1
dat_rmr_bind completed!
sending_rmr_context = 1b3b78 target_address = 95e3a000 segment_length = 40000
```

- **ib0** is the name of the device.
- **192.168.0.1** is the IPoIB address of the server host.
- **262144** is the size in bytes of the RDMA WRITE.

**Step 3** View the Throughput test results from the server.

The following example shows the Throughput test results:

```
Created an EP with ep_handle = 0x2a95f8a300
queried max_recv_dtos = 256
queried max_request_dtos = 1024
Accept issued...
Received an event on ep_handle = 0x2a95f8a300
Context = 29a
Connected!
received rmr_context = 1b3b78 target_address = 95e3a000 segment_length = 40000
Sent 7759.462 Mb in 1.0 seconds throughput = 7741.811 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.583 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.499 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.753 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.885 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.800 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.769 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.769 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.707 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7741.703 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.260 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.283 Mb/sec
Sent 7759.462 Mb in 1.0 seconds throughput = 7742.483 Mb/sec
total secs 13 throughput 7742 Mb/sec
Received an event on ep_handle = 0x2a95f8a300
Context = 29a
```

The notable performance result in the example is Throughput as 7.7 gigabits per second.

## uDAPL Latency Test Performance

The uDAPL Latency test measures half of the round-trip latency for uDAPL sends. To perform a uDAPL Latency test performance, perform the following steps:

**Step 1** Start the Latency test on the server host. The syntax for the server is as follows:

```
/usr/local/topspin/bin/lat_server.x device_name iterations msg_size 0:poll/1:event
```

The following example shows how to start the Latency test on the server host:

```
host1$ /usr/local/topspin/bin/lat_server.x ib0 200000 1 0
latency server started on ib0
```

- **ib0** is the name of the device.
- **200000** is the number of RDMA writes to perform for the test.
- **1** is the size in bytes of the RDMA WRITE.
- **0** is a flag specifying whether polling or event should be used. 0 signifies polling, and 1 signifies events.

**Step 2** Start the Latency test on the client.

The syntax for the client is as follows:

```
/usr/local/topspin/bin/lat_client.x device_name server_name iterations msg_size 0:poll/1:event
```

The following example shows how to start the Latency test on the client:

```
host2$ /usr/local/topspin/bin/lat_client.x ib0 192.168.0.1 200000 1 0
```

- **ib0** is the name of the device.
- **192.168.0.1** is the IPoIB address of the server host.
- **200000** is the number of RDMA's to perform for the test.
- **1** is the size in bytes of the RDMA WRITE.
- **0** is a flag specifying whether polling or event should be used. 0 signifies polling, and 1 signifies events.

### Step 3 View the Latency results.

The following example is a display of the Latency test results:

```
Server Name: 192.168.0.1
Server Net Address: 192.168.0.1
      Connection Event: Received the correct event
Latency:      6.5 us
Latency:      6.5 us
Latency:      6.5 us
Latency:      6.5 us
Latency:      6.5 us
Latency:      6.5 us
Latency:      6.5 us
Latency:      6.5 us
Latency:      6.5 us
Latency:      6.5 us
Average latency:      6.5 us
      Connection Event: Received the correct event
closing IA...
Exiting program...
```

The notable performance value in the example above is Latency result that is 6.5 microseconds.

---

## Compiling uDAPL Programs

This section provides information on how to compile uDAPL programs. Compiling uDAPL applications from source code requires use of the uDAPL header files and libraries included with the drivers.

Sample makefiles and C coder are in `/usr/local/topspin/examples/dapl`.