UDT as an Alternative Transport Protocol for GridFTP

Raj Kettimuthu
kettimut@mcs.anl.gov
Argonne National Laboratory
The University of Chicago
Outline

- GridFTP
- GridFTP Architecture
- Globus XIO
- UDT
- GridFTP/UDT integration
- Experimental results
GridFTP

- A secure, robust, fast, efficient, standards based, widely accepted data transfer protocol
- We also supply a reference implementation:
  - Server
  - Client tools (globus-url-copy)
  - Development Libraries
- Multiple independent implementations can interoperate
  - University of Virginia and Fermi Lab have home grown servers that work with ours.
- Lots of people have developed clients independent of the Globus Project.
GridFTP

- Two channel protocol like FTP
- **Control Channel**
  - Communication link (TCP) over which commands and responses flow
  - Low bandwidth; encrypted and integrity protected by default
- **Data Channel**
  - Communication link(s) over which the actual data of interest flows
  - High Bandwidth; authenticated by default; encryption and integrity protection optional
GridFTP offers a powerful feature called striped transfers (cluster-to-cluster transfers)
GridFTP Architecture
GridFTP Architecture

GridFTP Data Transfer Pipeline
Globus XIO
Grid Communication

- Geographically Distributed Resources
- Varying Networks Characteristics
  - LAN, WAN, LFN, Dedicated, Shared, QOS
- Varying Network Protocols
  - HTTP, UDT, TCP, RBUDP, etc.
  - Researching making newer and faster
- Varying Conditions
  - Congested/Idle
Varying Environments

Application

LFN

Dedicated

Infiniband

Shared

TCP
Globus XIO

- Framework to compose different protocols
- Provides a unified interface open/close/read/write
- Driver interface to hook 3rd party protocol libraries
Varying Networks

Application

Globus XIO

UDT

Infiniband

TCP Driver

LFN

Dedicated

Shared
Stack

- An arrangement of drivers
- Transport
  - Exactly one per stack
  - Must be on the bottom
- Transform
  - Zero or many per stack
Stack

- An arrangement of drivers
- Transport
  - Exactly one per stack
  - Must be on the bottom
- Transform
  - Zero or many per stack
Stack

- An arrangement of drivers
- Transport
  - Exactly one per stack
  - Must be on the bottom
- Transform
  - Zero or many per stack
UDT
UDT

- **UDT: UDP based Data Transfer**
  - Application level transport protocol, over UDP with reliability, congestion, and flow control
  - Implementation: Open source C++ library
- **Rate based congestion control (Rate Control)**
  - RC tunes the packet sending period.
  - RC is triggered periodically.
- **Window based flow control (Flow Control)**
  - FC limits the number of unacknowledged packets.
  - FC is triggered on each received ACK.
UDT

- AIMD: Increase parameter is related to link capacity and current sending rate; Decrease factor is 1/9, but not decrease for all loss events.
- Link capacity is probed by packet pair, which is sampled UDT data packets.
  - Every 16th data packet and its successor packet are sent back to back to form a packet pair.
  - The receiver uses a median filter on the interval between the arrival times of each packet pair to estimate link capacity.
UDT

![Graph showing throughput over time for different destinations with varying Round Trip Times (RTTs).](image)
GridFTP/UDT Integration
Wrapblock Driver Development

- Easy way to write XIO Drivers
  - Create from third party libraries.
- Blocking API
  - Thread pooling/event callbacks to morph async to sync
  - Recommend threaded builds
- UDT driver developed using the wrapblock feature
Interface functions

- A set of function signatures
  - open/close/read/write implemented by driver
  - cntl() functions for driver specific hooks
  - Wrapped into a structure and registered with Globus XIO

- Calls to these functions are made expecting specific behaviours
  - Ex: the read() interface function should produce some data, and the write() interface function should consume data, etc
Example Interface Functions

```c
static
globus_result_t
globus_l_xio_udt_ref_read(
    void * driver_specific_handle,
    const globus_xio_iovec_t * iovec,
    int iovec_count,
    globus_size_t * nbytes)
{
    globus_result_t result;
    xio_l_udt_ref_handle_t * handle;

    handle = (xio_l_udt_ref_handle_t *) driver_specific_handle;

    *nbytes = (globus_size_t) UDT::recv(
        handle->sock, (char *)iovec[0].iov_base,
        iovec[0].iov_len, 0);

    /* need to figure out eof */
    if(*nbytes <= 0)
    {
        result = GlobusXIOUdtError("UDT::recv failed");
        goto error;
    }

    return GLOBUS_SUCCESS;
error:
    return result;
}

static
globus_result_t
globus_l_xio_udt_ref_write(
    void * driver_specific_handle,
    const globus_xio_iovec_t * iovec,
    int iovec_count,
    globus_size_t * nbytes)
{
    globus_result_t result;
    xio_l_udt_ref_handle_t * handle;

    handle = (xio_l_udt_ref_handle_t *) driver_specific_handle;

    *nbytes = (globus_size_t) UDT::send(
        handle->sock, (char*)iovec[0].iov_base,
        iovec[0].iov_len, 0);

    if(*nbytes < 0)
    {
        result = GlobusXIOUdtError("UDT::send failed");
        goto error;
    }

    return GLOBUS_SUCCESS;
error:
    return result;
}
```
## Throughput achieved using various transport mechanisms

<table>
<thead>
<tr>
<th>Transport Mechanism</th>
<th>Testbed</th>
<th>ANL - NZ</th>
<th>ANL - ISI</th>
<th>BMI - Japan</th>
<th>Japan – ORNL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iperf – 1 stream</td>
<td></td>
<td>19.7</td>
<td>74.5</td>
<td>59</td>
<td>110</td>
</tr>
<tr>
<td>Iperf – 8 streams</td>
<td></td>
<td>40.3</td>
<td>117.0</td>
<td>115.3</td>
<td>592.4</td>
</tr>
<tr>
<td>scp – 1 stream</td>
<td></td>
<td>2.96</td>
<td>9.6</td>
<td>3.13</td>
<td>3.14</td>
</tr>
<tr>
<td>bbcp mem – 1 stream</td>
<td></td>
<td>-</td>
<td>35.3</td>
<td>5.84</td>
<td>118.2</td>
</tr>
<tr>
<td>bbcp mem – 8 streams</td>
<td></td>
<td>-</td>
<td>59.2</td>
<td>11.7</td>
<td>467.25</td>
</tr>
<tr>
<td>bbcp disk – 1 stream</td>
<td></td>
<td>-</td>
<td>35.15</td>
<td>5.85</td>
<td>112.6</td>
</tr>
<tr>
<td>bbcp disk – 8 streams</td>
<td></td>
<td>-</td>
<td>54.8</td>
<td>11.7</td>
<td>451.3</td>
</tr>
<tr>
<td>GridFTP mem TCP – 1 stream</td>
<td></td>
<td>16.4</td>
<td>63.8</td>
<td>79.6</td>
<td>123.3</td>
</tr>
<tr>
<td>GridFTP mem TCP – 8 streams</td>
<td></td>
<td>40.2</td>
<td>112.6</td>
<td>222</td>
<td>586.5</td>
</tr>
<tr>
<td>GridFTP disk TCP – 1 stream</td>
<td></td>
<td>16.3</td>
<td>59.6</td>
<td>73.6</td>
<td>113.5</td>
</tr>
<tr>
<td>GridFTP disk TCP – 8 streams</td>
<td></td>
<td>37.4</td>
<td>102.4</td>
<td>201.6</td>
<td>574.6</td>
</tr>
<tr>
<td>GridFTP mem UDT</td>
<td></td>
<td>189.7</td>
<td>426.6</td>
<td>238</td>
<td>382.5</td>
</tr>
<tr>
<td>GridFTP disk UDT</td>
<td></td>
<td>187.9</td>
<td>418.3</td>
<td>220.5</td>
<td>380.6</td>
</tr>
<tr>
<td>UDT mem</td>
<td></td>
<td>202.3</td>
<td>432.5</td>
<td>246.3</td>
<td>397.2</td>
</tr>
<tr>
<td>UDT disk</td>
<td></td>
<td>174.2</td>
<td>398.0</td>
<td>211.6</td>
<td>374.5</td>
</tr>
</tbody>
</table>
## Impact of concurrent flows

<table>
<thead>
<tr>
<th>Japan-ORNL testbed</th>
<th>BMI-Japan testbed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nonconcurrent flows</strong></td>
<td><strong>Nonconcurrent flows</strong></td>
</tr>
<tr>
<td>GridFTP-TCP</td>
<td>GridFTP-TCP</td>
</tr>
<tr>
<td>GridFTP-UDT</td>
<td>GridFTP-UDT</td>
</tr>
<tr>
<td><strong>2 Concurrent TCP flows</strong></td>
<td><strong>2 Concurrent TCP flows</strong></td>
</tr>
<tr>
<td>GridFTP-TCP</td>
<td>GridFTP-TCP</td>
</tr>
<tr>
<td>GridFTP-TCP</td>
<td>GridFTP-TCP</td>
</tr>
<tr>
<td><strong>2 Concurrent UDT flows</strong></td>
<td><strong>2 Concurrent UDT flows</strong></td>
</tr>
<tr>
<td>GridFTP-UDT</td>
<td>GridFTP-UDT</td>
</tr>
<tr>
<td>GridFTP-UDT</td>
<td>GridFTP-UDT</td>
</tr>
<tr>
<td><strong>2 Concurrent flows (1 TCP and 1 UDT)</strong></td>
<td><strong>2 Concurrent flows (1 TCP and 1 UDT)</strong></td>
</tr>
<tr>
<td>GridFTP-TCP</td>
<td>GridFTP-TCP</td>
</tr>
<tr>
<td>GridFTP-UDT</td>
<td>GridFTP-UDT</td>
</tr>
<tr>
<td><strong>3 Concurrent flows (2 TCP and 1 UDT)</strong></td>
<td><strong>3 Concurrent flows (2 TCP and 1 UDT)</strong></td>
</tr>
<tr>
<td>GridFTP-TCP</td>
<td>GridFTP-TCP</td>
</tr>
<tr>
<td>GridFTP-TCP</td>
<td>GridFTP-TCP</td>
</tr>
<tr>
<td>GridFTP-UDT</td>
<td>GridFTP-UDT</td>
</tr>
</tbody>
</table>
Resource Utilization of UDT vs TCP

- The performance of TCP and UDT comparable on TeraGrid network between ANL and ORNL
  - Both TCP and UDT achieved a throughput around 700 Mbit/s on this testbed.
- The CPU utilization for TCP transfers was in the range of 30–50%, whereas for UDT transfers it was around 80%.
- The memory consumption was around 0.2% for TCP and 1% for UDT.
Questions