A Multipath Controller for Accelerating GridFTP Transfer over SDN

Che Huang, Chawanat Nakasan, Kohei Ichikawa, Hajimu Iida

Nara Institute of Science and Technology, Japan
Background

- The amount of data produced by scientific researches has been rapidly increasing
  - High-speed data transfer service is needed as a platform service
Problem

- High-speed transfer mechanism of GridFTP
  - GridFTP supports parallel data transfer scheme by using multiple TCP streams to mitigate network congestion
  - Alternate paths are not used efficiently
  - GridFTP only uses a single Shortest path according to the default IP routing policy

There is still significant room for improvement in data transfer performance by traffic engineering technique using multiple paths
SDN (Software-Defined Network)

- OpenFlow
  - Software technology dynamically controls routing and packet processing on a per-flow basis
Purpose of research

- Traffic engineering using the SDN technology
- Distributes the parallel TCP streams of GridFTP into multiple network paths

Realizes high-speed data transfer by aggregating available bandwidth
Approach

- Parallel GridFTP transfer using SDN

Diagram:
- GridFTP client
- GridFTP server
- Multipath OpenFlow controller
- Route Request
- Route Control
- OpenFlow Switches
How GridFTP works

1. GridFTP client opens a control channel for FTP

2. The default GridFTP implementation uses a built-in TCP XIO driver and opens multiple TCP streams according to the specified number of threads

- Globus XIO is an I/O plug-in framework implemented in the Globus Toolkit

```
.globus-url-copy -p 3 data gsiftp://HostB/data
```
Design of proposed system

- Secures available multiple paths before transfer starts
- Assigns one of the secured paths each time a TCP stream is opened

- Implements a XIO driver to communicate with the OpenFlow controller and establishes TCP streams accordingly
- Minimizes the modifications to the GridFTP client itself

Implemented as XML-RPC
- Independent of a specific implementation
- Enables any applications to use the proposed controller
Implementation of the proposed system

- **AssignMultipath**: Route search and Allocation
- **MakePath**: Deployment of flow entry
- **assign_mpath**: Client for the route search request
- **XIO driver** for the proposed system
Implementation of the proposed system

- AssignMultipath: Route search and Allocation
  MakePath: Deployment of flow entry

- assign_mpath: Client for the route search request

- XIO driver for the proposed system
Implementation of the proposed system

- **AssignMultipath**: Route search and Allocation
  - **MakePath**: Deployment of flow entry
  - **assign_mpath**: Client for the route search request
  - **XIO** driver for the proposed system

*User*
- assign_mpath

*Host A*
- GridFTP Client
- XIO

*Host B*
- GridFTP server

*OpenFlow Switches*

- Calculates the available paths with the breadth-first search algorithm
- Returns the number of available paths with an assignment ID
- Secures the specified number of available paths sorted by path length

*Multipath OpenFlow Controller*
Implementation of the proposed system

- AssignMultipath: Route search and Allocation
  MakePath: Deployment of flow entry
- assign_mpath: Client for the route search request
- XIO driver for the proposed system

Starts GridFTP client with the assignment ID

Route deployment request

The proposed XIO driver requests the OpenFlow controller with the ID to deploy flow entries accordingly each time a TCP stream opens
Implementation of the proposed system

- **AssignMultipath**: Route search and Allocation
- **MakePath**: Deployment of flow entry
- **assign_mpath**: Client for the route search request

**XIO** driver for the proposed system

- Acquires the set of paths assigned for this client, and pick a route one by one in a round-robin manner from the path set
- Installs flow entries accordingly so that the requested TCP stream goes through the selected path

**User**

**assign_mpath**

**GridFTP Client**

**XIO**

**Host A**

**GridFTP server**

**Host B**

**OpenFlow Switches**
EVALUATION

1. Experiments on Virtual Environment
   • To measure the exact performance of proposed system

2. Experiments on Real Global-scale Environment
   • To demonstrate the practicality of proposed system
Six virtual machines were installed with Open vSwitch on different physical machine.
Host A and Host B were used as GridFTP client and server respectively.
Various topologies can be prepared by changing the combination of the GRE links.
The bandwidth of each link was limited to 100Mbps.
Network Topologies

### Topology A
- A simple topology where each path is completely independent
  - sw1-sw2-sw6
  - sw1-sw3-sw6
  - sw1-sw4-sw6
  - sw1-sw5-sw6
- Evaluated using two and four parallel TCP streams

### Topology B
- A more realistic topology that includes some overlapped part
  - sw1-sw5-sw6
  - sw1-sw2-sw3-sw6
  - sw1-sw2-sw4-sw6
- Evaluated using three parallel TCP streams
Results of Topology A with two parallel TCP streams

<table>
<thead>
<tr>
<th></th>
<th>Proposed system</th>
<th>Conventional method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer time</td>
<td>47.486s</td>
<td>92.108s</td>
</tr>
</tbody>
</table>

Diagram showing network topology and bandwidth usage over time.
Results of Topology A
with four parallel TCP streams

<table>
<thead>
<tr>
<th></th>
<th>Proposed system</th>
<th>Conventional method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer time</td>
<td>23.404s</td>
<td>90.533s</td>
</tr>
</tbody>
</table>
Results of Topology B

<table>
<thead>
<tr>
<th></th>
<th>Proposed system</th>
<th>Conventional method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer time</td>
<td>46.603s</td>
<td>90.480s</td>
</tr>
</tbody>
</table>
Real Global-scale Experimental Environment (PRAGMA-ENT)

Path 1
Path 2
Path 3
Path 4

GRE link; OFS: OpenFlow Switch; OVS: Open vSwitch

GridFTP client
 GridFTP server

NAIST(JAPAN)
OU(JAPAN)
Tokyo
OFS(NEC PF5240)
OFS(Pica8 P-3290)
OVS

Osaka
NICT(JAPAN)
OFS(NEC PF5240)
OFS(Pica8 P-3290)
OFS(Pica8 P-3290)
OVS

UCSD(USA)
OU(JAPAN)
UF(USA)
Result 1: Average speed of data transfer between proposed system and conventional method for increasing the number of parallel TCP streams one by four.
Result 2: Used bandwidth of each path by using 24 parallel TCP streams of proposed system (Each path used 6 parallel TCP streams)
Conclusion

- Proposed a multipath controller providing multiple paths for the parallel transfer of GridFTP by using SDN technology based on OpenFlow
- Performed experiments to demonstrate the effectiveness of our proposed system
  - In a virtual environment
  - In a real global-scale environment
Future Work

- Implements algorithms to search optimal routes considering bandwidth and latency of each path
- Considers a method that calculates the optimal number of parallel TCP streams for each path