UDT Based Application Performance for High Speed Data Transfer

Joshua Miller [jsmiller@uchicago.edu]
The University of Chicago

Overview

Problem:
Over high performance wide area networks, TCP has been shown to reach a data transfer bottleneck before UDP. UDT is a proven and reliable UDP based application level protocol, but is not user friendly.

Purpose:
- Improve performance of UDT based rsync wrapper (UDR)
- Create a new application to eliminate possible overhead (UDTCAT)
- Create an encryption package to match the transfer rate

UDT

UDT is proven for transporting large data sets over high performance networks and has powered applications that have won the Supercomputing Bandwidth Challenge in 2006, 2008 and 2009.

Benefits:
- Disk-disk transfer rate is 9 Gbps over 10 Gbps WAN
- Forms the underlying technology for products already on the market
- High performance with a single flow

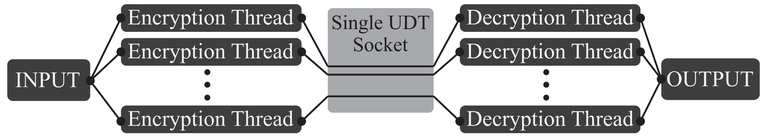
UDR

UDR (developed by the Laboratory for Advanced Computing):
- One dependency, OpenSSL encryption
- Simple prefix wrapper to rsync
- Compatible with all rsync options
- Open source

UDR uses rsync and scp as they are, there is no need to patch or install a new version.

UDT Applications

Threaded Encryption:
An encryption package was needed to match the throughput of the UDT tools.
- Created a system to align parallel OpenSSL EVP ciphers



UDR Contributions:
- Re-implemented UDR for use with scp
- Introduced threaded encryption
- Attempts to buffer data to optimal block size did not increase throughput

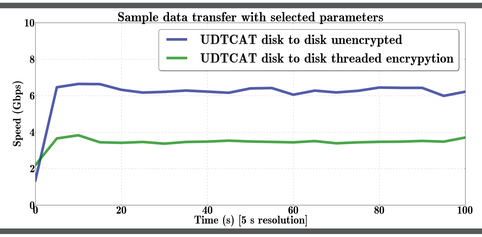
```
local$ udr [rsync/scp command]
```

UDTCAT:
- Originally based on netcat functionality
- Includes **multi-threaded** encryption and **authentication** using the threaded encryption system
- Useful for piping large data streams securely over UDT
- Any application that binds to a TCP port can be securely piped over UDT in combination with a two way pipe and netcat
- Open source

```
client$  uc host port [< infile]
server$  uc -l port   [> outfile]
```

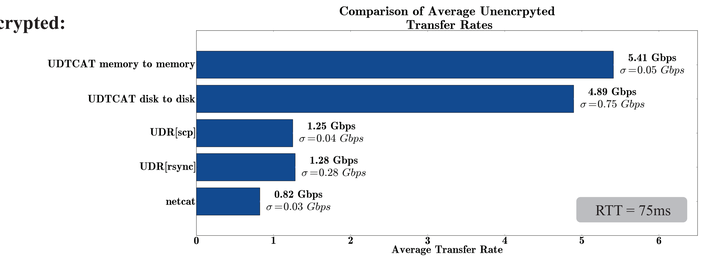
UDTCAT works just like a secure netcat

Example Transfers

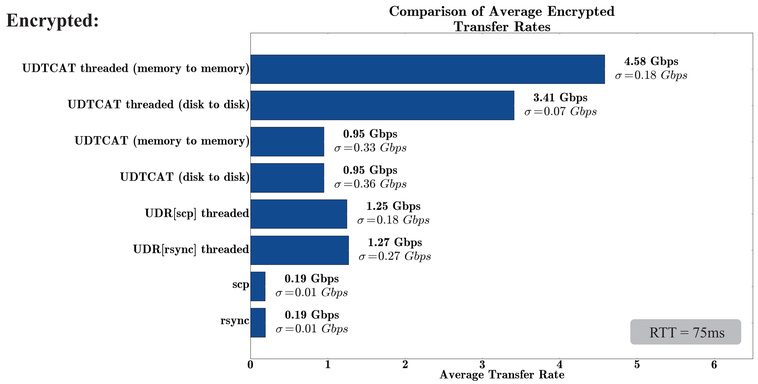


Performance Results

Unencrypted:



Encrypted:



This data was collected on a network with approximately 75ms RTT and .1% packet loss. The data on threaded encryption was taken with 16 encryption threads using the Intel Xeon CPU E5-2650 @ 2.00 GHz . Disk to disk transfers were measured using a 50 GB file on a Seagate 400GB SSD . All chart entries are the average of 3 nonconsecutive tests.

Looking Forward

Future plans in this project include:
- Adding option to patch rsync to remove buffer size bottleneck
- Continued development of UDTCAT for easier file transfer and more robust piping options

Special thanks to Allison Heath, Matt Greenway, and the rest of Robert Grossman's team.
This material is based upon work supported by the National Science Foundation under Grants No. CISE-1127316 and OISE-1129076.

The University of Chicago · OSDC Open Science Data Cloud · NSF · OCC Open Cloud Consortium · LAC Laboratory for Advanced Computing www.labcomputing.org