

A Reliable and Fast Data Transfer for Grid Systems Using a Dynamic Firewall Configuration

Thomas Oistrez

Research Centre Juelich
Juelich Supercomputing Centre

August 21, 2008

Overview

- 1 existing solutions
 - UNICORE ByteIO
 - GridFTP
 - problems
- 2 a new solution
 - requirements
 - UDP hole punching
 - UDT
 - UNICORE integration
- 3 results
 - performance
 - security

UNICORE ByteIO

- how it works
 - data is send via Webservice calls
 - default UNICORE protocol stack is used
- performance
 - complex protocols are used (SOAP / WS / HTTPS)
 - the gateway is a bottleneck
- security
 - no direct connections between hosts
 - no open server ports needed
 - all data is encrypted

GridFTP

- how it works
 - encrypted control connection
 - several data connections in parallel
 - file is split in pieces
- performance
 - very high
 - to many data connections lead to control overhead
- security
 - statically opened ports needed on every server
 - these ports are not used by GridFTP for the most time

why we need a new file transfer

- ByteIO is too slow for many applications (max. 400 KBit/s)
- GridFTP is fast, but it needs an unsecure network configuration
- a better solution should be
 - easy to install/configure
 - easy to use
 - fast
 - secure

requirements

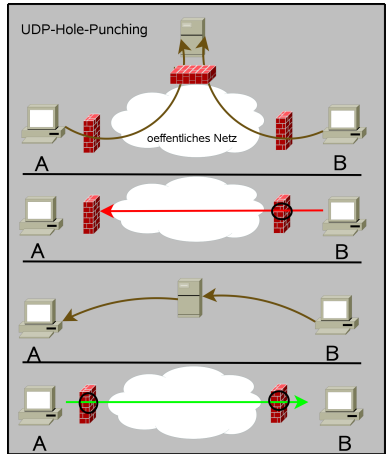
- It can be integrated smoothly into an existing administrative security framework
- It can be used in open source and in commercial solutions
- Configuration happens automatically
- Configuration is always up to date
- Connections are allowed for a limited period of time

preconditions

- outgoing TCP connections are allowed
- outgoing UDP packets are allowed (outgoing virtual UDP Connections)
- one central gateway is needed to exchange control data

hole punching

- central gateway service
- TCP control connection to the gateway
- information exchange through the gateway
- UDP Hole Punching
- data exchange with UDP



UDP based Data Transfer (UDT)

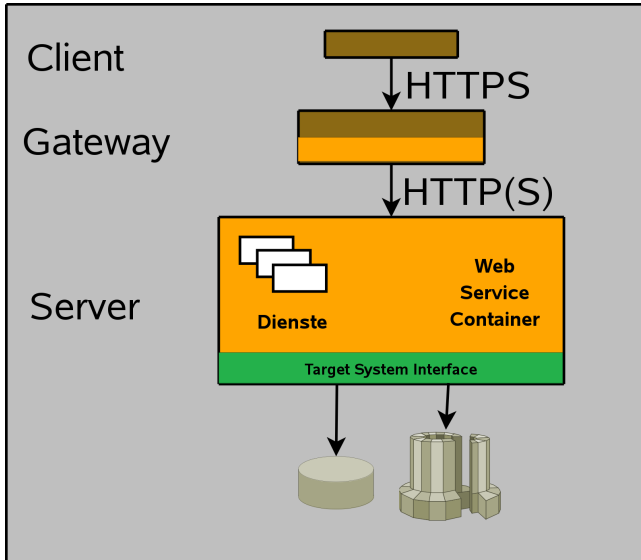
features

- open source c++ code
- connection oriented
- sequence numbers
- acknowledgments

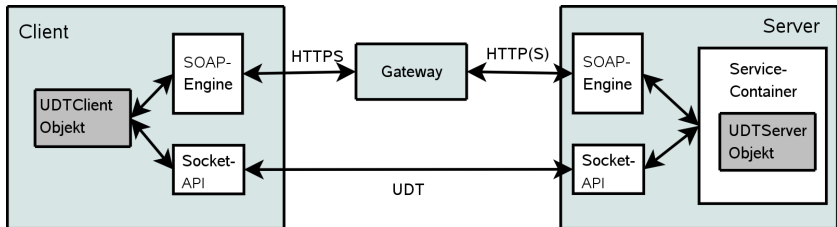
performance

- flexible congestion control
- as fair as necessary
- as fast as possible

UNICORE architecture



implementation of the new file transfer



implementation

- control messages are send as Webservice calls
- data is send through a UDT connection

performance comparison (1 GBit/s net)

ByteIO

- performance is limited by the protocol stack
- maximum bandwidth is about 400 KBit/s

GridFTP

- performance depends on the number of connections used in parallel
- best performance with 4 connections: 300 MBit/s
- more connections have too much controll overhead

performance comparison (1 GBit/s net)

UDT

- performance depends on the congestion control algorithm
- with the default congestion control, up to 930 MBit/s were possible
- fairness was very bad with the default algorithm

security comparison

ByteIO

- no direct connections between hosts
- all traffic is encrypted
- highest level of security

GridFTP

- control messages are encrypted
- many statically opened ports
- this leads to security holes

security comparison

UDP Hole Punching

- no incoming connections allowed
- outgoing connections are allowed in most environments
- only really needed connections are allowed for a short period of time
- control messages are encrypted

existing solutions
a new solution
results

performance
security

Questions?

Questions?