# CSC 7700: Scientific Computing
# Module B: Networks and Data
# Lecture 2: Network Applications, Special Subjects

*Dr. Andrei Hutanu*

# *"Application" layer*

- Second part of this module (lectures 3-5) will cover increasingly complex applications of networks
- Today will introduce bulk file transfer and videoconferencing
  - Directly on top the transport layer
  - But can be quite complex on their own
  - Will talk about GridFTP file transfer mechanism and its tuning options

# *Grids and grid security*

- Brief introduction
  - A grid is a collection of resources used for a common goal
  - A form of distributed computing (module by Dr. Jha)
  - Many grid applications involve using multiple machines
    - To run a complex application workflow where each machine works on a sub-problem
    - Or even to do a file transfer (at least two machines involved)
    - There needs to be a mechanism for the user to authenticate (log-in) and for the processes they run on these machines to “recognize” and trust each other

# *Public/private key*

- Data Encryption
  - Can do with symmetric keys (same key to encode/decode) – but need to securely transmit the key to the receiver and trust the receiver with it
  - Public/private key
    - Public key is used to Encrypt the message, private is used to decrypt only and is kept safely
    - So you know if the receiver can decipher the message, he must have the correct private key
  - But how do you know that public key you received is authentic?

# *Certificates*

- Trusted third-party that signs your public key
- Delegation
  - So you can use the public key to secure communication and identify to a particular service
  - But what if you want that service to act on your behalf
    - For example you want it to transfer a file to a third site
    - The service should be able to authenticate
    - Solution: a new limited-term public-private key pair is generated (proxy), signed with private key and shared with trusted services to authenticate on your behalf

# Practice

- MyProxy
  - Manages your private key and certificate
- Used in TeraGrid (log-in to any TG machine first)
  - myproxy-logon -l <portal TeraGrid username>
  - Use TeraGrid portal password
  - You will get a proxy on that machine that you can use for whatever operation
  - Integrated with portal, that's how ssh log-in works from the TeraGrid portal
- More info: https://www.teragrid.org/web/user-support/auth_proxy

# GridFTP motivation

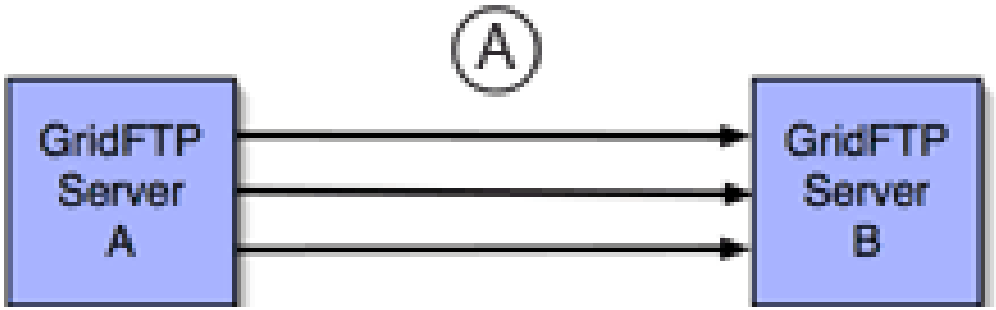
- How do you transfer files?
- Transfer large data files (tera->petabytes). Bulk data transfer
- Efficiency
- Protocol to enable transfers between high-performance data servers
- Starting from the FTP protocol
  - Wide understood and implemented
  - Architecture allows for extensions

# What is GridFTP?

- Standard file transfer protocol defined by the Open Grid Forum (OGF)
  - www.ggf.org/documents/GWD-R/GFD-R.020.pdf
  - www.ogf.org/documents/GFD.21.pdf
  - www.ogf.org/documents/GFD.47.pdf
- Implemented as part of the Globus Toolkit
- Installed on TeraGrid machines
  - http://www.globus.org/toolkit/docs/4.0/data/gridftp/
  - http://info.teragrid.org/restdemo/html/tg/services/gridftp?sort=Endpoint

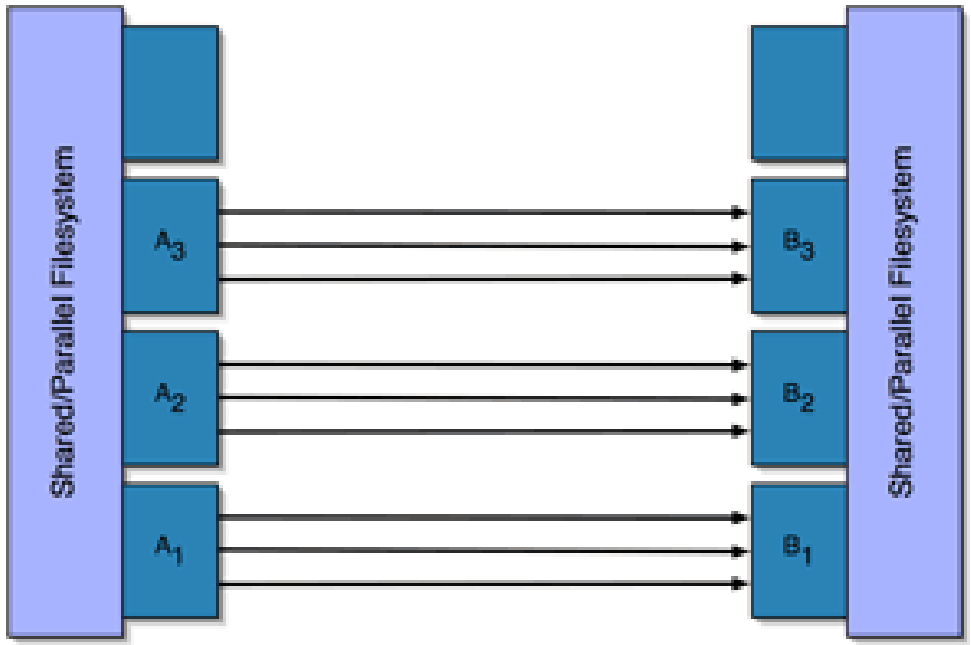
# GridFTP features

- Parallelism
  - Multiple parallel data streams can be used to transfer data between two machines
  - Talked about this as a method to optimize data transfer



# GridFTP features

- Striping
  - Multiple hosts can participate in a single data/file transfer (or multiple network interfaces on a single machine)

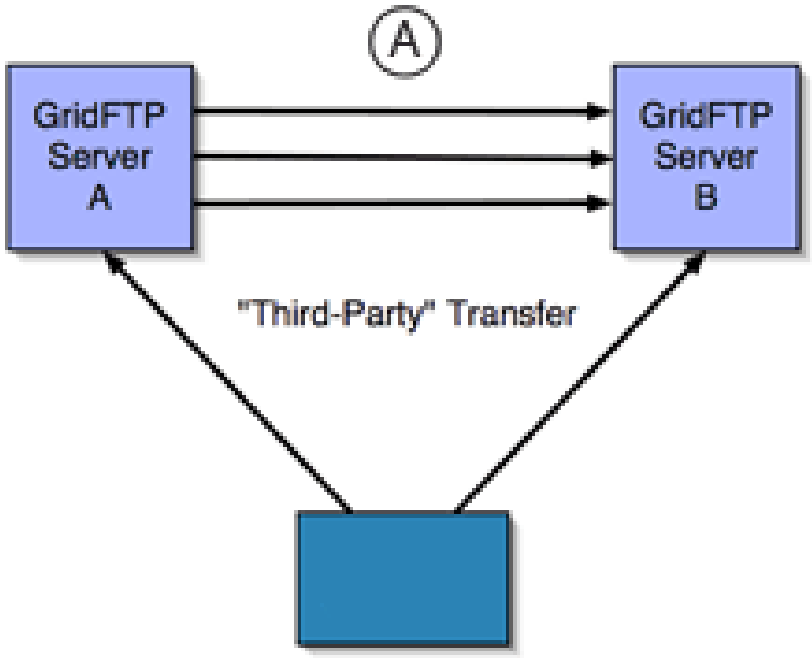


# *GridFTP features*

- Restartable data transfers
  - If something goes wrong, and the restart markers were saved, transfer can be started from where it was interrupted
    - To recover from transient network or server failures
- Partial file transfer
  - To copy only parts of the file
  - Support for server-side data transform operations: for data selection operations that are more complex than (offset, length) or other simple server-side operations

# GridFTP features

- Third-party transfer




- Support for other transport protocols (UDT), multicast – newer versions

# GridFTP usage

- There's a development library
  - Can implement your own GridFTP client
- globus-url-copy
  - Command-line utility to use GridFTP
  - Need to have a grid proxy first (get it through myproxy on TeraGrid – see earlier discussion)
  - globus-url-copy <source> <destination>
  - Example:
    - log-in to Abe
    - execute pwd to print working directory

# GridFTP usage

- pwd
  - /u/ac/hutanu (will be different from you)
  - Create a file called test
  - [hutanu@honest1 ~]$ ls -la test
  - -rw-r-----  1 hutanu out 702092 Aug 20 17:08 test
- Run globus-url-copy with URL of Abe gridftp server + filename as source, URL of Steele gridftp server destination
- URL with all GridFTP servers in TeraGrid
  http://info.teragrid.org/restdemo/html/tg/services/gridftp?sort=Endpoint

# GridFTP example

- [hutanu@honest1 ~]$ globus-url-copy -vb gsiftp://gridftp-abe.ncsa.teragrid.org:2811/u/ac/hutanu/test gsiftp://tg-steele.purdue.teragrid.org:2811/dev/null

  Source: gsiftp://gridftp-abe.ncsa.teragrid.org:2811/u/ac/hutanu/

  Dest: gsiftp://tg-steele.purdue.teragrid.org:2811/dev/

  test -> null

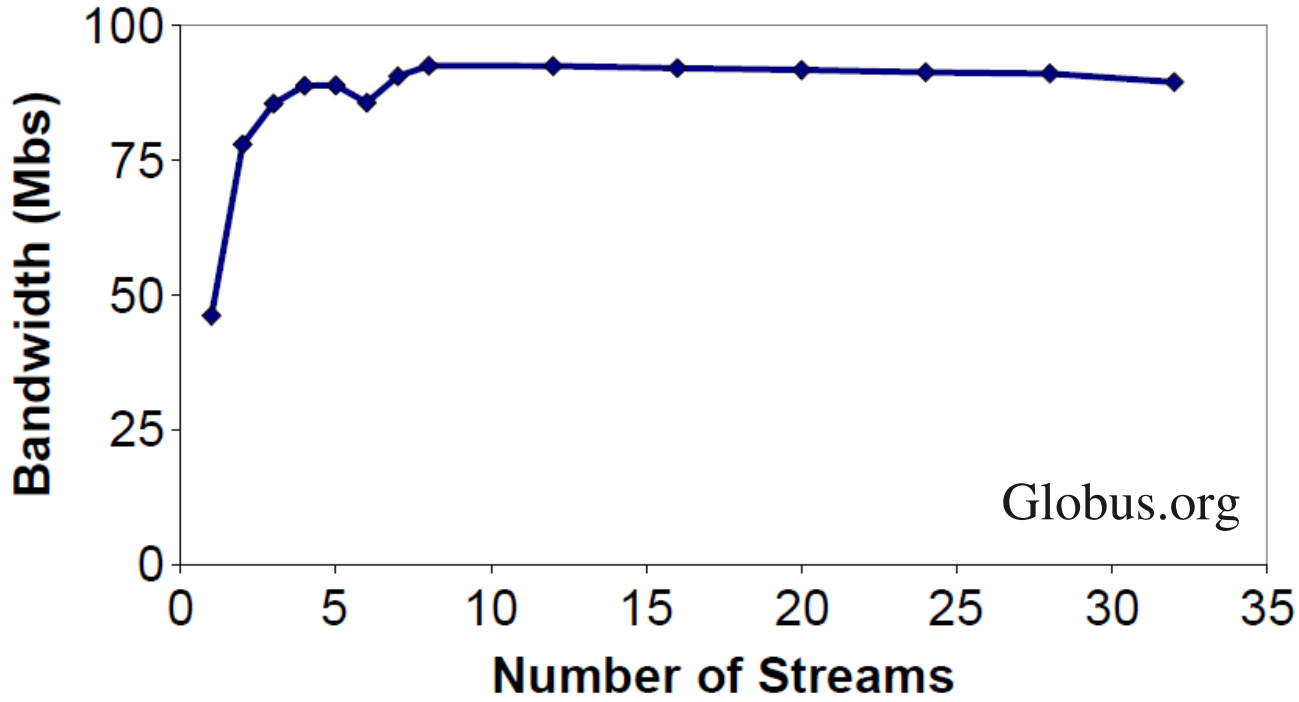
  702092 bytes 3.35 MB/sec avg 3.35 MB/sec inst
- -vb makes it print some information (including performance)
- /dev/null destination is special file .. anything that is “written” to that location is discarded

# GridFTP example

- Also have a special source location: /dev/zero – anything that you read from there will be a zero. (no filesystem access .. this is from memory)
  - But it is of infinite size .. so copy will never end
  - Can use it as a source, but you need to use partial file transfer to limit the amount of data that is transferred:
    - Use -len parameter: -len 100000000
    - globus-url-copy -vb -len 100000000 gsiftp://gridftp-qb.loni-lsu.teragrid.org:2811/dev/zero gsiftp://tg-gridftp.lonestar.tacc.teragrid.org:2811/dev/null
    - Changed machines .. doing third party transfer

# GridFTP optimization

- Parallel streams
  - -p <number of streams>
- How many streams? Somewhere between 2 and 10 perhaps .. (over 15 will be too many)



# Papers on parallel TCP streams

- "The end-to-end performance effects of parallel TCP sockets on a lossy wide-area network" (one of the earlier papers on the subject)
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1015527&tag=1
- "Dynamically Tuning Level of Parallelism in Wide Area Data Transfers" (automatic selection of parallelism)
  http://www.cct.lsu.edu/~kosar/papers/dadc_2008.pdf
- See previous lecture note on parallel streams
- TeraGrid network has a lot of capacity, and congestion is not a problem, so it is ok to use parallelism

# Other flags

- Can use striping where enabled
  - Shown on the servers list
  - -stripe on the command line
- TCP buffer size (default is auto-tuned).
  - -tcp-bs <size>. Might degrade your performance though
- You can also do server->file or file->server transfer (no third party). Use file:// as prefix for source or destination (followed by local file system path)
- globus-url-copy -help
- http://www.globus.org/toolkit/docs/4.0/data/gridftp/rn01re01.html

# Assignment (part 2)

- Use GridFTP with globus-url-copy to evaluate the performance of GridFTP in TeraGrid
- Same machines as used for iperf, now with gsiftp:// URL's pointing to the GridFTP server installations
- Use /dev/zero as source and /dev/null as destination for memory to memory transfer only (no file system access). Don't forget -len
  - See effect of data size (-len parameter), parallelism (-p), striping (-stripe). Use -vb to get the performance report.
- Optional: use real files, for example using MSS (which doesn't have a /dev/zero) and document the performance difference compared to mem-to-mem

# File transfer (miscellaneous)

- bbftp
  - parallel stream file transfer
- Reliable file transfer service/globus.org service
  - Third party web service
  - You give it the file transfer command
  - Uses GridFTP and stores restart markers, attempting multiple restarts
- LHC data grid
  - http://lcg.web.cern.ch/lcg/
  - Nice graphs with up-to-date data transfer information

# *Videoconferencing*

- Very common application
- Quite different from bulk data transfer
  - Bulk file transfer wants reliable transfer, and cares about how long it takes
- Videoconferencing parameters
  - Delay (latency) – too much delay will make it feel non-interactive
  - So reliable data transport can be detrimental, because it adds delay! (Major difference to bulk data transfer)
  - Packet delay variation and packet ordering (so everything is in sequence)
    - Some buffering used

# Quality

- Resolution (number of pixels)
- Frame rate
- How about compression?
  - Adds delay, and possibly artifacts
  - Drastically reduces the bandwidth required
- Done experiments in the past with uncompressed HD videoconferencing (following slides)
  - http://www.sitola.cz/papers/728231.pdf
- Audio quality! (most important actually, sound is not optional)

# *Multicast*

- Usually for audio/video transmission to multiple receivers
- Distribution tree
  - Sender does not send a copy of data for each receiver (and overloads his network connection)
  - Data is distributed in the network
- IP Multicast – done by the routers in the network
  - Very slow/no adoption in commercial internet
- Overlay multicast
  - Computers attached to nodes in the network perform it

# *HD Classroom*

- Early experiment with high-speed networks in a production environment
- Classroom teaching, strong quality requirements
- “Introduction to HPC”: Professor Thomas Sterling, spring 2007
- 5 participants (LSU, Masaryk University, University of Arkansas, Louisiana Tech, MCNC)
- Uncompressed high-definition video (Ultragrid)
  - Bandwidth: 1.5 Gbps each stream (full HD – 1920x1080 pixels, 30fps)

# HD Classroom

- Czech Republic view (left), Arkansas (right)
- Many applications of videoconferencing
  - Medical consultation, live performance, observation (ocean floor) etc ..

# *AccessGrid. Cinegrid*

- AccessGrid (see www.accessgrid.org)
  - Community effort to build high-quality videoconferencing infrastructure. Used for remote teaching, conferences.
  - Based on open-source tools (VIC/RAT)
  - (Two nodes at LSU, two rooms in this building)
- CineGrid (see www.cinegrid.org)
  - Research and development of high-quality collaborative tools for manipulating digital media
  - 4K (4096x2160 pixels) ~ 4 x full HD resolution
  - Movie industry, High-speed networks

# *Circuit-Based Services*

- Regular (shared), or packet-switched networks provide no guarantee
  - Makes it difficult to plan .. it may work it may not
- New paradigm: User-controlled circuit networks
  - Allocate your own network circuit, guaranteed
  - Information (about circuits) is stored in network
  - Production services: Internet2 ION, ESnet Science Data Network
  - Implementations: UCLP, OSCARS, etc

# *Internet2 ION*

- A reservation of bandwidth across the backbone
  - Not a dedicated physical circuit
- Recording (up to minute 1:20)
  - http://www.youtube.com/watch?v=jg5HQ1wc9_A
  - Can see the old, shared route printed with traceroute
  - Reservation is made
  - New path is used (less hops)
- http://www.internet2.edu/ion/

# *Lightpath switching*

- Optical fiber can transport a number of wavelengths or colors of laser light – lambdas or lightpaths
- Each lambda can now transport 10Gbit/s
- Can have permanent/static lambdas
  - Provides a dedicated channel between two locations (reliable, secure)
- Or dynamic lambdas
  - Temporarily turned on/switched for a particular user or application. Actually implemented using mirrors.
  - Time-sharing
- Lightpaths can support protocols other than IP!

# *Wide-area InfiniBand*

- If you have access to a dedicated (even for a short time) physical point-to-point connection you can use your own protocol!
  - No need for IP (no routing)
- InfiniBand – protocol design for high performance data transmission inside a cluster (Intro to HPC course)
- Paper: “Wide-Area Performance Profiling of 10GigE and InfiniBand Technologies”
  - http://ft.ornl.gov/pubs-archive/2008-sc-ib-wan.pdf

# Network scheduling

- One option for implementing the circuit service is “on-demand”
  - Try to see if a circuit is available, then you can reserve it
- However, that's still somewhat limiting. Want to plan in advance
  - Have a videoconference/class Tuesdays at noon – need guarantee!
- Scheduling
  - Stored “in the network”
  - You can reserve the network circuit in advance

# *Co-allocation/Distributed Scientific Apps*

- Prerequisite: distributed application (can execute parts at various locations); network connections set-up to the actual interesting resources – networks are as useful as the resources they connect
- Futuristic (but realistic, working demos) scenario:
  - Find available resources schedule (graphics, compute, data, network)
  - At time of interest
  - Select the optimal resources for application
  - co-allocate resources (atomically)
  - Execute application

# GLIF and GOLE

- Setting up network circuits in a single “administrative domain” - single virtual entity/network is fairly well understood
- Want to create circuits across multiple networks (and they should be connected)
- GLIF community (many international network providers) -> GOLE project
- Automated circuit exchange points (between networks)
  - Can allocate the exchange point together with the networks they connect -> make a circuit from Europe to Japan, through the US

# Content Distribution Networks

- Akamai, Limelight, Amazon
  - ~ 30% of all internet traffic now, increasing
- A way to get around the fact that the Internet backbone is congested
  - All users cannot access a single centralized server
  - Replicate the objects to strategically placed servers around the world
  - Bring your “web” closer to you, so you can access it faster
  - User accesses the closest replica
- http://www.akamai.com/html/technology/nocc.html

# *Network "neutrality"*

- Network is not neutral
  - It would be if we would have: Infinite supply, Limited demand. But we have: Limited supply and infinite demand
  - Networks have to choose what packets to drop
- Right now, each packet is equal
  - If a user X sends video at 1Gbps and Y sends e-mail at 1kbps each of X's packets is equal to each Y's packets. So a 1Gbps network is allocated to 99.9999% to the first user, and 0.0001% to the second one
  - Flow Rate Fairness: dismantling a religion
    - http://bobbriscoe.net/projects/2020comms/refb/fair_ccr.pdf

# *Alternatives*

- Prioritize packets somehow
- But how?
  - Some ISP's want to give priority to their packets. That's not fair either!
- Based on how much you pay?
  - Works well for other limited resources, but some don't like it – creates “inequality”
- Underlying issue is real – networks become more congested and there's no incentive to upgrade
- Discussion mostly political, we need to look into the technical issues! (if interested, consult wiki for links)

# IPv6

- Activated on campus on August 26 (last Thursday)
  - You probably have now both IPv4 and V6 (http://test-ipv6.com/)
- Benefits
  - Larger address space (4.2 billion for IPv4, 340 trillion trillion trillion for IPv6 - 670 quadrillion addresses per square mm)
  - Improved security features
  - Jumbogram
    - 4GB packets – however dependent on both the underlying link layer support and the transport protocols on top (not existing yet)