the globus alliance
www.globus.org

GridCopy: Moving Data Fast on the Grid

Raj Kettimuthu, Bill Allcock, Lee Liming, JP Navarro and Ian Foster

Argonne National Laboratory, Argonne, IL

What is GridCopy or GCP?

- GridFTP has been commonly used as a data transfer protocol in the Grid
- Provides a SCP-style GridFTP client interface to users
- Takes care of tuning required to get optimal performance for data transfers
- Provide extensible configuration options for site administrators to optimize data movement

GridFTP

- Extends standard FTP protocol to provide a lot of important features
  - Striped data transfer (cluster to cluster)
  - Partial file transfer
  - Reliable and restartable data transfer
  - Data channel caching
  - Supports Grid Security Infrastructure
  - Setting of TCP buffer sizes

Globus GridFTP

- Modular design
  - XIO architecture makes it easy to switch transport protocols (TCP/UDT)
  - Data Storage Interface (DSI) makes it easy to access different storage systems (HPSS, SRB)
- Client side optimizations needed to get maximum performance
  - Users are either unaware or find it difficult to do it

Optimizations

- TCP is the default transport protocol used by GridFTP
- It is critical to use optimal socket buffer sizes to get maximum throughput
  - Bandwidth-delay product
- Sometimes it is necessary to use multiple TCP streams
  - Difficult to predict the optimum number of streams

GCP

- Globus-url-copy, RFT, uberftp are some of the well-known GridFTP clients
  - Users have to do these optimizations - which is not an easy task for many
- GCP is a wrapper over globus-url-copy and RFT
  - It calculates the optimal TCP buffer size and optimal number of TCP streams for users
  - Accepts SCP-style source and destination specifications

GCP

- GCP uses a configuration file to translate the user request into a potentially complicated data movement request
- Site administrators fill this configuration file using their knowledge of the local system
  - GridFTP servers may be running on hosts that can access source and/or destination files faster
  - TeraGrid uses such translations to optimize data transfer
  - Looking at sophisticated options to share the translation information among the sites

TCP buffer size

- Optimal buffer size = 2*bandwidth*delay
- GCP uses King to calculate the delay
  - From any node on the Internet, measure latency between arbitrary hosts on Internet
  - No additional infrastructure needed on end hosts
  - Estimate latency between the domain name servers
  - Claim ~75% of DNS servers support recursive queries from any host
  - Assume name servers are located close to their hosts

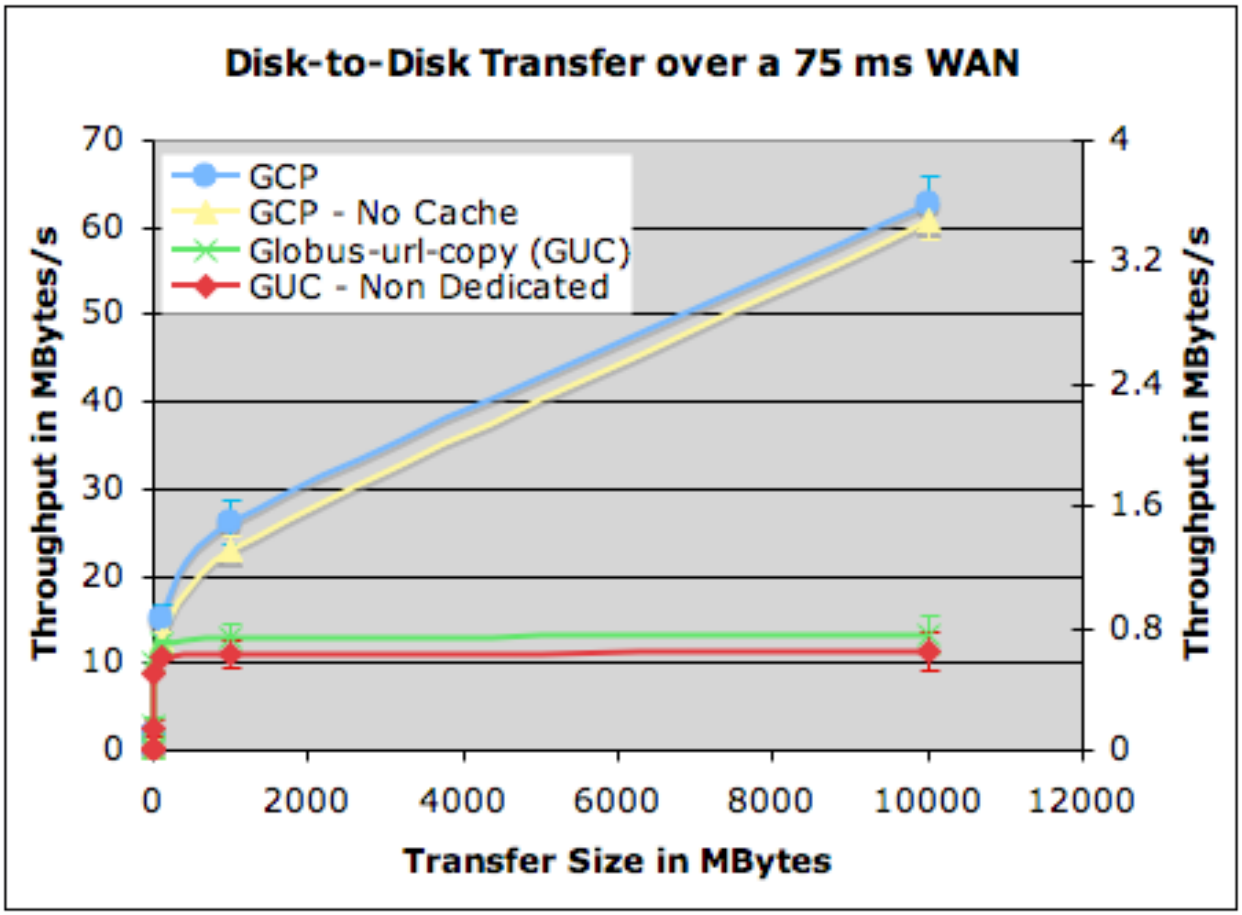
TCP buffer size

- For bandwidth, it uses total capacity (static value but configurable) of the link
  - Calculating the current available bandwidth is tricky - also it may not give desired result
  - Default value used is 1Gbit/s
- Option to cache the calculated value
- Multiple TCP streams
  - (2 * BDP)/max(1, streams/loss_factor)
- Loss_factor accommodates for congestion hit streams

Number of streams

- Default is 4
  - Based on the past experience, increasing the number of streams above 4 does not fetch much
- Try to arrive at a optimal number of streams for subsequent transfers between endpoints
  - Decreasing the number of streams and comparing the achieved throughput with the prior value (there is a timeout period)

Experimental results



Small files

XFER SIZE	GCP (B/s)	GCP-NC (B/s)	GUC (B/s)
1KB	283	200	304
10KB	2833	2427	3391
100KB	29596	26929	24489
1M	295819	259634	228358

Future work

- Moving translation rules across the sites have scalability issues
  - Plan to provide more scalable solutions using MDS or PubSub models
- Evaluation of parallel streams prediction heuristics