the globus alliance
www.globus.org

Parallel TCP

Bill Allcock
Argonne National Laboratory

Definitions

- Logical Transfer
  - The transfer of interest to the initiator, i.e. move file foo from server A to server B.
- Network Endpoint
  - In general something that has an IP address. A Network Interface Card (NIC).
- Parallel Transfer
  - Use of multiple TCP streams between a given pair of network endpoints during a logical transfer
- Striped Transfer
  - Use of multiple pairs of network endpoints during a logical transfer.

What's wrong with TCP?

- You probably wouldn't be here if you didn't know that.
- TCP was designed for Telnet / Web like applications.
- It was designed when T1 was a fast network, big memory was 2MB, not 2 GB, and a big file transfer was 100MB, not 100GB or even Terabytes.

AIMD and BWDP

- The primary problems are:
  - Additive Increase Multiplicative Decrease (AIMD) congestion control algorithm of TCP
  - Requirement of having a buffer equal to the Bandwidth Delay Product (BWDP)
  - The interaction between those two.
  - We use parallel and striped transfers to work around these problems.

AIMD

- To the first order this algorithm:
  - Exponentially increases the congestion window (CWND) until it gets a congestion event
  - Cuts the CWND in half
  - Linearly increases the CWND until it reaches a congestion event.
  - This assumes that congestion is the limiting factor
  - Note that CWND size is equivalent to Max BW

BWDP

- Use a tank as an analogy
- I can keep putting water in until it is full.
- Then, I can only put in one gallon for each gallon removed.
- You can calculate the volume of the tank by taking the cross sectional area times the height
- Think of the BW as the area and the RTT as the length of the network pipe.

Recovery Time

$$\text{Recovery Time} = \frac{\text{Bytes to Recover}}{\text{Rate of Recovery}}$$

$$\text{Bytes to Recover} \propto \frac{1}{2}\ BW * RTT\ (BWDP)$$

$$\text{Rate of Recovery} \propto \frac{MTU}{RTT}$$

$$\text{Recovery Time} \propto \frac{\frac{1}{2}\ BW * RTT}{\frac{MTU}{RTT}} \Rightarrow \frac{RTT^2 * BW}{MTU}$$

Recovery Time for a Single Congestion Event

- T1 (1.544 Mbs) with 50ms RTT ≅ 10 KB
  - Recovery Time (1500 MTU): 0.16 Sec
- GigE with 50ms RTT ≅ 6250 KB
  - Recovery Time (1500 MTU): 104 Seconds
- GigE to Amsterdam (100ms) ≅ 1250 KB
  - Recovery Time (1500 MTU): 416 Seconds
- GigE to CERN (160ms) ≅ 2000 KB
  - Recovery Time (1500 MTU): 1066 Sec (17.8 min)

How does Parallel TCP Help?

- We are basically cheating…. I mean we are taking advantage of loopholes in the system
- Reduces the severity of a congestion event
- Buffers are divided across streams so faster recovery
- Probably get more than your fair share in the router

Reduced Severity from Congestion Events

- Don’t put all your eggs in one basket
- Normal TCP your BW Reduction is 50%
  - 1000 Mbs * 50% = 500 Mbs Reduction
- In Parallel TCP BW Reduction is:
  - Total BW / N Streams * 50%
  - 1000 / 4 * 50% = 125 Mbs Reduction
- Note we are assuming only one stream receives a congestion event

Faster Recovery from Congestion Events

- Optimum TCP Buffer Size is now BWDP / N where N is number of Streams
- Since Buffers are reduced in size by a factor of 1/N so is the recovery time.
- This can also help work around host limitations. If the maximum buffer size is too small for max bandwidth, you can get multiple smaller buffers.

More than your Fair Share

- This part is inferred, but we have no data with which to back it up.
- Routers apply fair sharing algorithms to the streams being processed.
- Since your logical transfer now has N streams, it is getting N times the service it otherwise normally would.
- I am told there are routers that can detect parallel streams and will maintain your fair share, though I have not run into one yet.

What about Striping?

- Typically used in a cluster with a shared file system, but it can be a multi-homed host
- All the advantages of Parallel TCP
- Also get parallelism of CPUs, Disk subsystems, buses, NICs, etc..
- You can, in certain circumstances, also get parallelism of network paths
- This is a much more complicated implementation and beyond the scope of what we are primarily discussing here.

Nothing comes for free…

- As noted earlier, we are cheating.
- Congestion Control is there for a reason
- Buffer limitations may or may not be there for a reason
- Other Netizens may austracize you.

Congestion Control

- Congestion Control is in place for a reason.
- If every TCP application started using parallel TCP, overall performance would decrease and there would be the risk of congestive network collapse.
- Note that in the face of no congestion parallel streams does not help
- In the face of heavy congestion, it can perform worse.

Buffer Limitations

- More often than not, the system limitations are there because that is way it came out of the box.
- It requires root privilege to change them.
- However, sometimes, they are there because of real resource limitations of the host and you risk crashing the host by over-extending its resources.

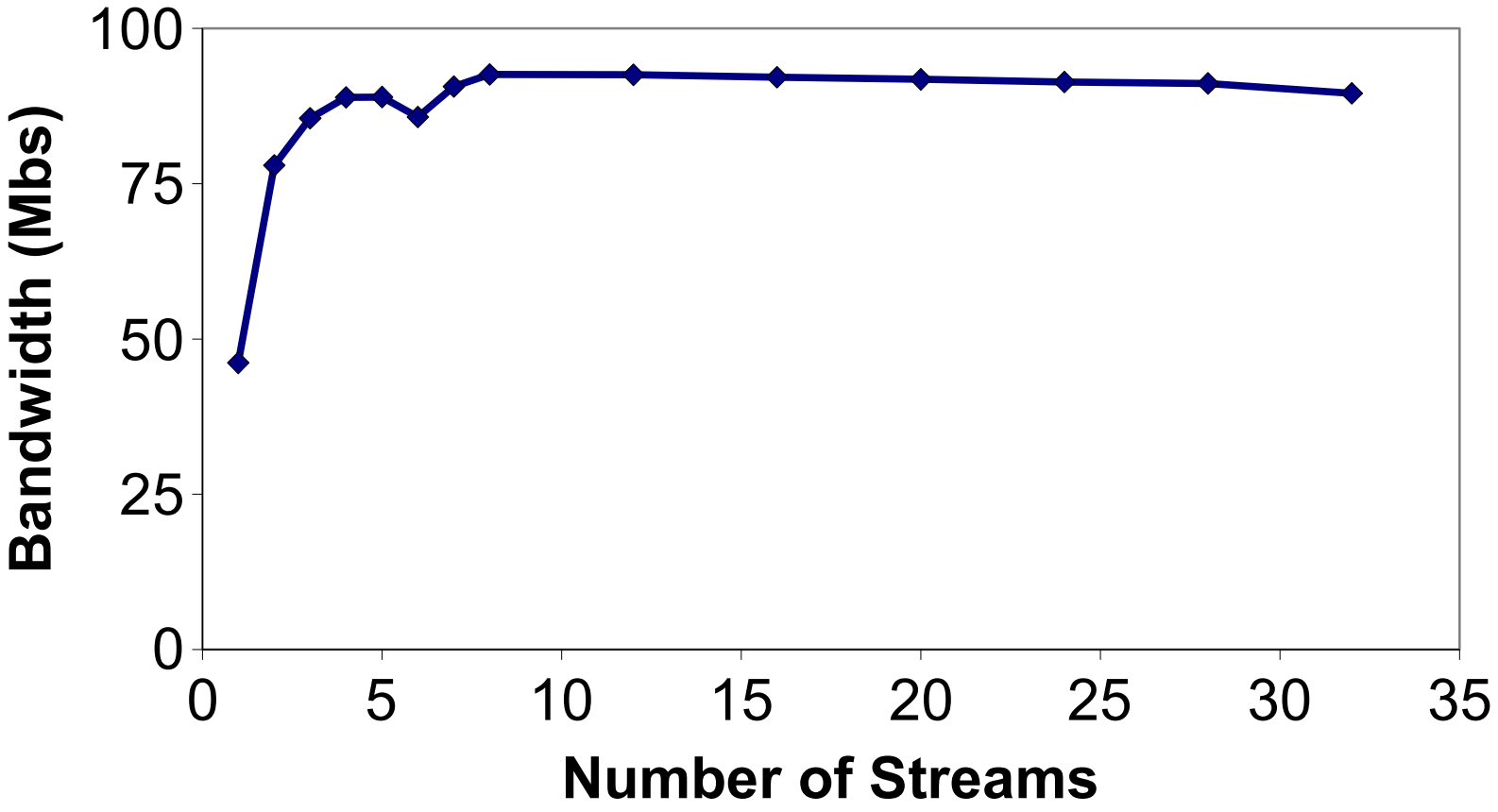
Cheat enough, but not too much

- If your use of parallel TCP causes too many problems you could find yourself in trouble.
  - Admins get cranky when you crash their machines
  - Other users get cranky if you are hurting overall network performance.
- Be a good Netizen

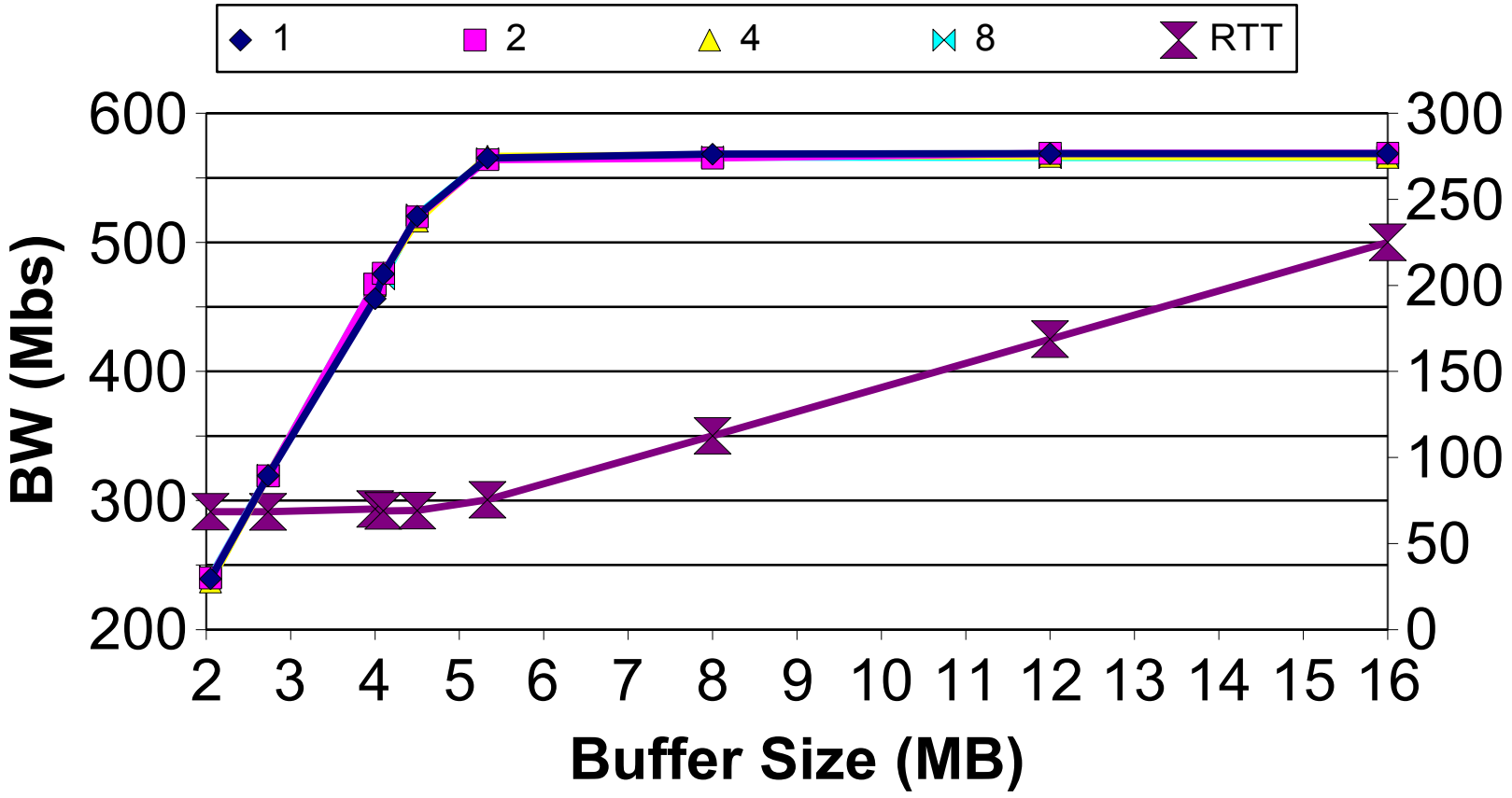
When should you use Parallel TCP?

- Engineered, private, semi private, or very over provisioned networks are good places to use parallel TCP.
- Bulk data transport. It makes no sense at all to use parallel TCP for most interactive apps.
- QOS: If you are guaranteed the bandwidth, use it
- Community Agreement: You are given permission to hog the network.
- Lambda Switched Networks: You have your own circuit, go nuts.

Affect of Parallel Streams
ANL to ISI (n=5)



Affect of TCP Buffer Size (iperf)



William (Bill) E. Allcock

Bill Allcock is the technology coordinator and evangelist for GridFTP within the Globus Alliance. Bill has a BS in Computer Science and an MS in Paper Science. In his 15 years work experience he has been involved in a wide array of areas including computer networking, distributed systems, embedded systems, data acquisition, process engineering, control system tuning, and colloidal chemistry.

Bill's current research focus involves applications requiring access to large (Terabyte and Petabyte sized) data sets, so called DataGrid problems. He is also heavily involved in the Global Grid Forum. Bill has presented several previous tutorials on the Globus Toolkit(R), primarily on GridFTP use and development libraries. He also has lead tutorials on Introduction to Grids, as well as IO and security in the Globus Toolkit.