



# A Comparison of GSIFTP and RFIO on a WAN

---

**Rajesh Kalmady** / Bhabha Atomic Research Centre  
[rajesh@magnum.barc.ernet.in](mailto:rajesh@magnum.barc.ernet.in)

**Brian L. Tierney** / Lawrence Berkeley Natl Lab  
[bltierney@lbl.gov](mailto:bltierney@lbl.gov)

(Presented by: **Harry Renshall** / CERN)



# Goals / Overview



- This talk will cover:
  - How to tune TCP performance
  - How parallel transfers also help
  - GSIFTP performance results
  - GSIFTP and RFIO results compared
  - Performance monitoring with “Netlogger”
  - Linux 2.4 performance issues



# TCP Performance Tuning Issues



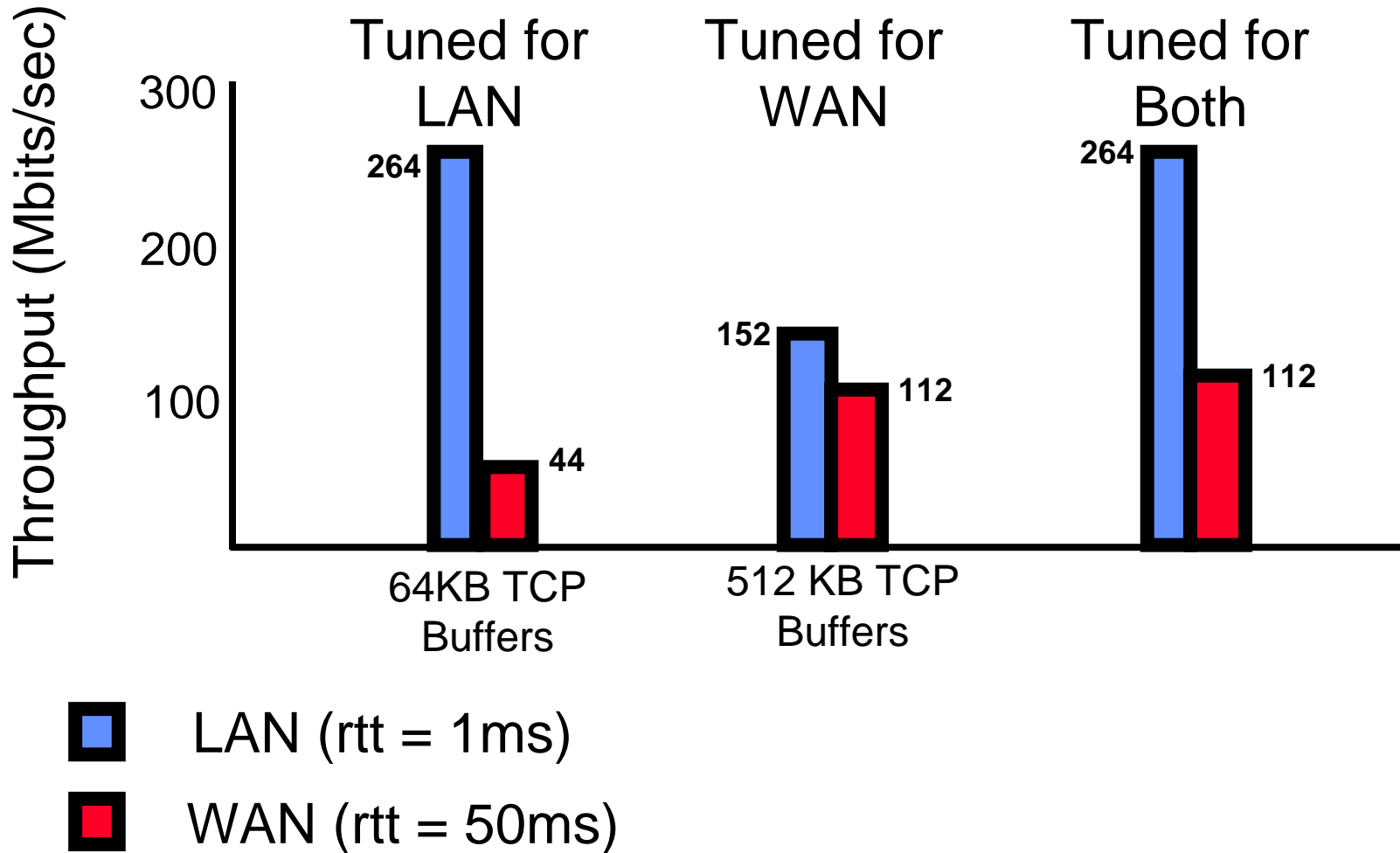
- **Getting good TCP performance over high-latency high-bandwidth networks is hard!**
- **Need to keep the pipe full, and the size of the pipe is directly related to the network latency:**
  - Example: from LBNL to ANL, there is (was) an OC12 network, and the one-way latency is 25ms
    - Bandwidth = 67 MB/sec (OC12 - ATM / IP headers = 539 Mb/s)
  - Need 67 Mbytes \* .025 sec = 1.7 MB of data “in flight” to fill the pipe
- **Must use optimal buffering and TCP window sizes**
- **Can also use parallel transfers to fill window**



# Setting the TCP buffer sizes



- **It is critical to use the optimal TCP send and receive socket buffer sizes for the link you are using.**
  - if too small, the TCP window will never fully open up
  - if too large, the sender can overrun the receiver, and the TCP window will shut down
- **Default TCP buffer sizes are way too small for this type of network**
  - default TCP send/receive buffers are typically 24 or 32 KB
  - with 24 KB buffers, can get **only 2.2%** of the available bandwidth!
- **(Parallel transfers can help to compensate this)**





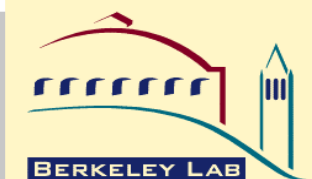
# Buffer Size Example



- ping time = 50 ms
- slowest network segment = 10 Mbytes/sec (e.g.: the end-to-end network consists of all 100 BT ethernet and OC3 (155 Mbps))
- TCP buffers should be:
  - $.05 \text{ sec} * 10 \text{ MB/sec} = 500 \text{ KBytes}$ .
- Remember: default buffer size is usually only 24KB, and default maximum buffer size is only 64-256KB !



# Advantage of Parallel Transfers



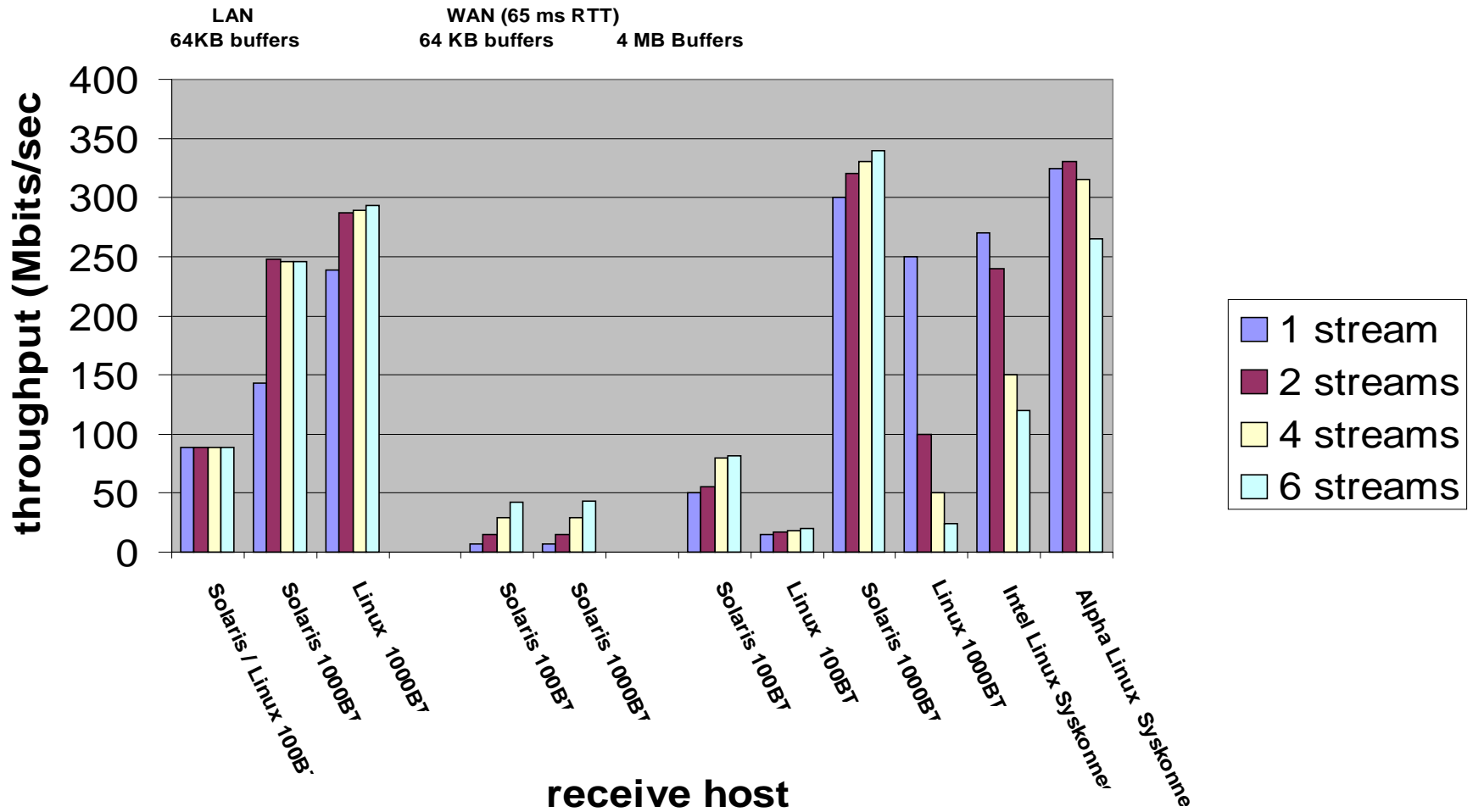
graph from Davide Salomoni, SLAC



# TCP WAN Performance: Host Issues



## Network Performance





# What is GSIFTP ?



- GSIFTP is a **high-performance, secure file transfer protocol**, forming part of the **GLOBUS Toolkit**.
- GSIFTP uses the GSI (Grid Security Infrastructure), for authentication. It is a **subset of the GRIDFTP** protocol.
- GSIFTP is currently being enhanced with a variety of protocol features appropriate for Grid applications, to allow ubiquitous, **high-performance access to data**, as well as remaining compatible with the standard FTP protocol.
- GSIFTP **includes TCP buffer and socket tuning**, and a major feature of GSIFTP is **parallel data transfer**.



# What is RFIO ?



- **RFIO (Remote File I/O)** is the CERN-developed **data access protocol** developed for SHIFT and used in the CERN Advanced Storage Manager (CASTOR).
- RFIO provides a **remote version of most standard POSIX calls** like **open, read, write, lseek** and **close**;
- RFIO is a user level implementation over TCP, **including buffer and socket tuning**.
- The control and data streams are separated. Data throughput is optimised by overlapping network and disk I/O: a circular buffer and two threads are used for each connection.
- **Multiple parallel streams are not yet implemented:** this should be done for use on high speed WANs.



# GridFTP / RFIO Test Setup



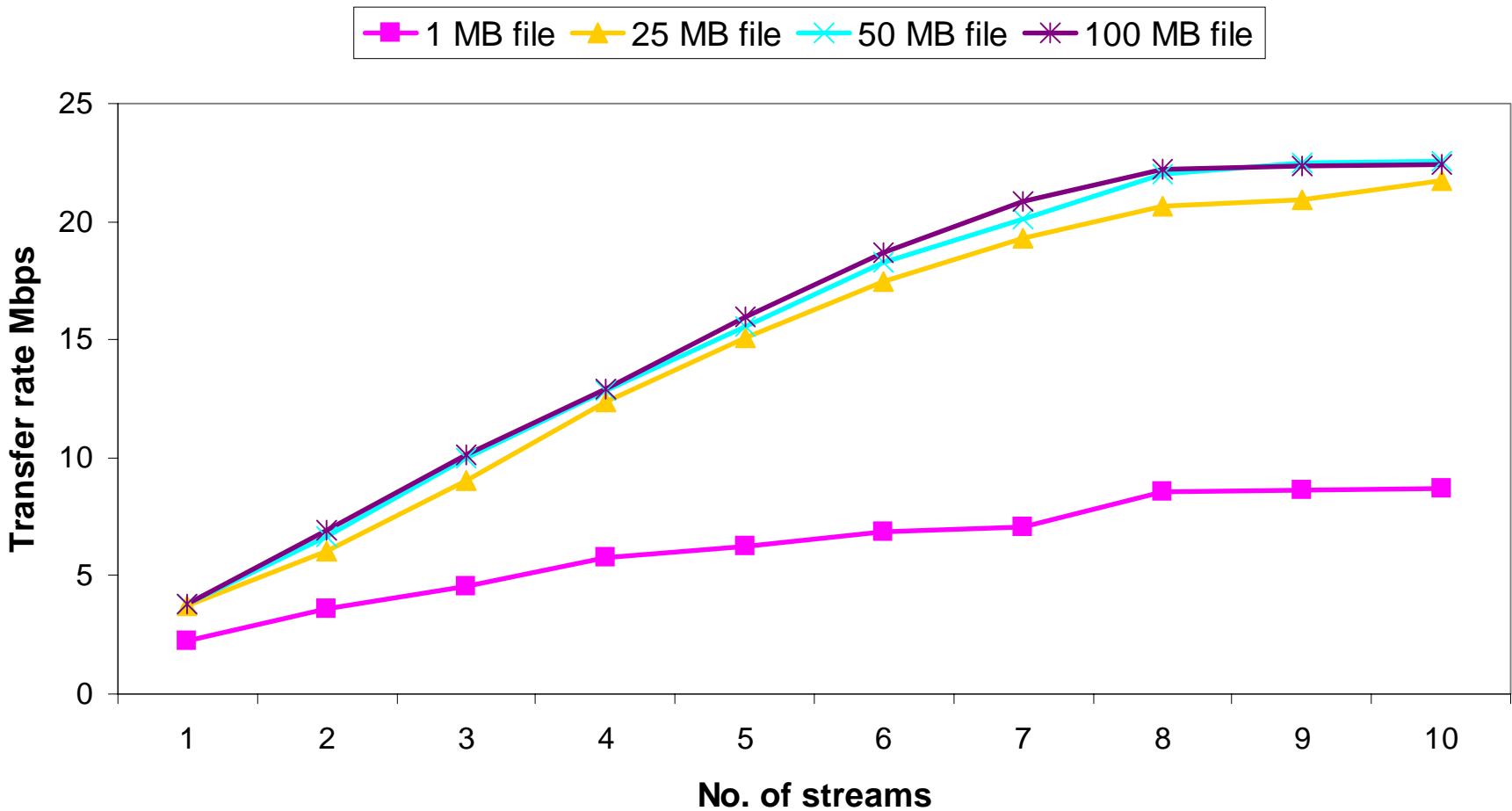
- **Test Environment:**
  - CERN to ANL
  - RTT = 125 ms
  - bottleneck link: 45 Mbits/sec
  - end hosts: Linux/Pentium/100BT
    - Note: these tests done just before CERN/Chicago link was upgraded to OC-3, still waiting for CERN LAN upgrade...
- **“Netlogger”** tracing/monitoring calls inserted into client and server modules.
  - See <http://www-didc.lbl.gov/NetLogger/>



# GSIFTP Get: Default TCP Buffers



Gsiftp Get Default TCP buffers

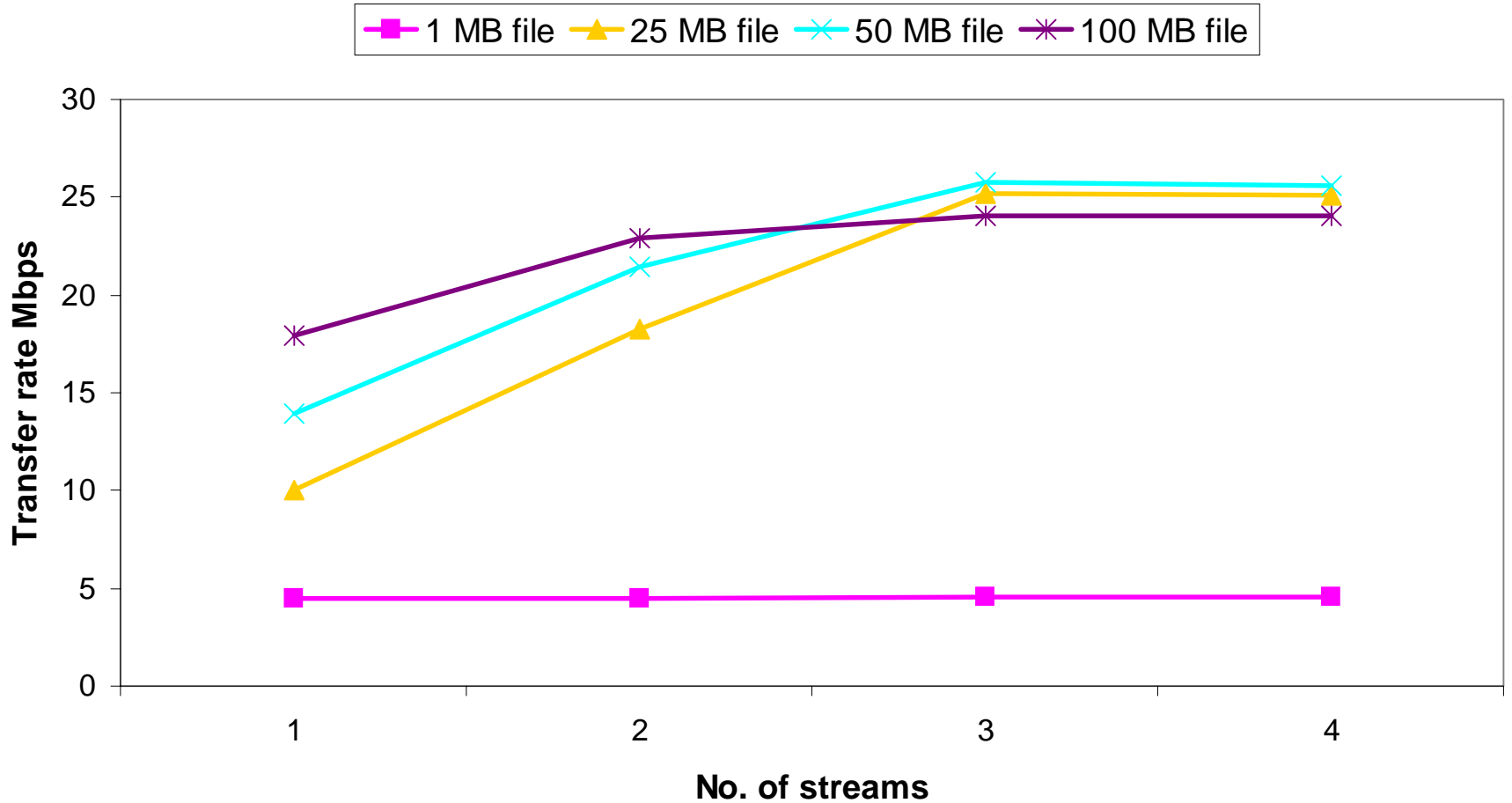




# GSIFTP Get: Tuned TCP Buffers (1 MB size)



Gsiftp Get tuned TCP buffers



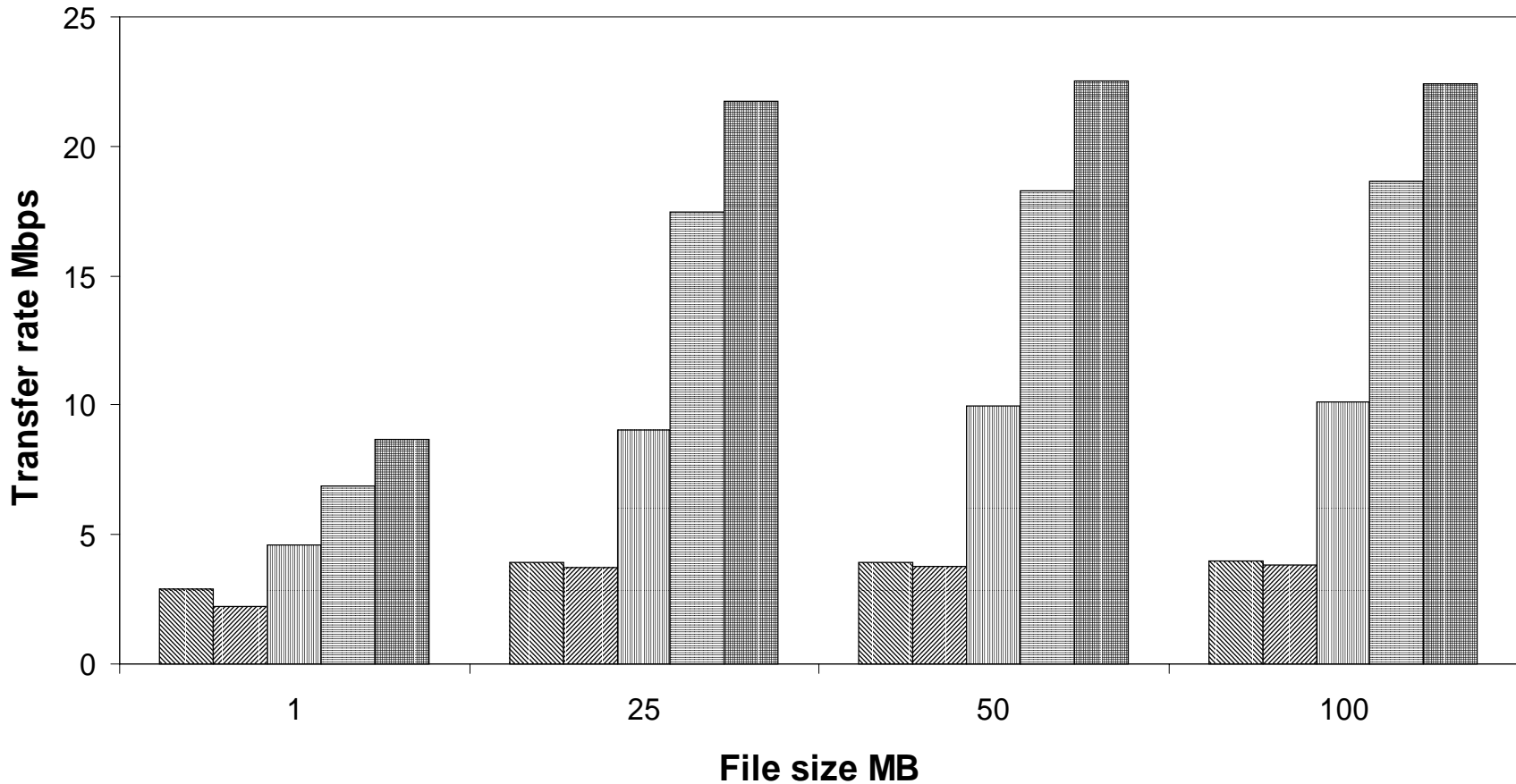


# Comparison of RFIO Get and GSIFTP Get (Default Buffers)



## RFIO Vs GSIFTP Get Default TCP Buffers

RFIO Gsiftp 1 stream Gsiftp 3 streams Gsiftp 6 streams Gsiftp 10 streams



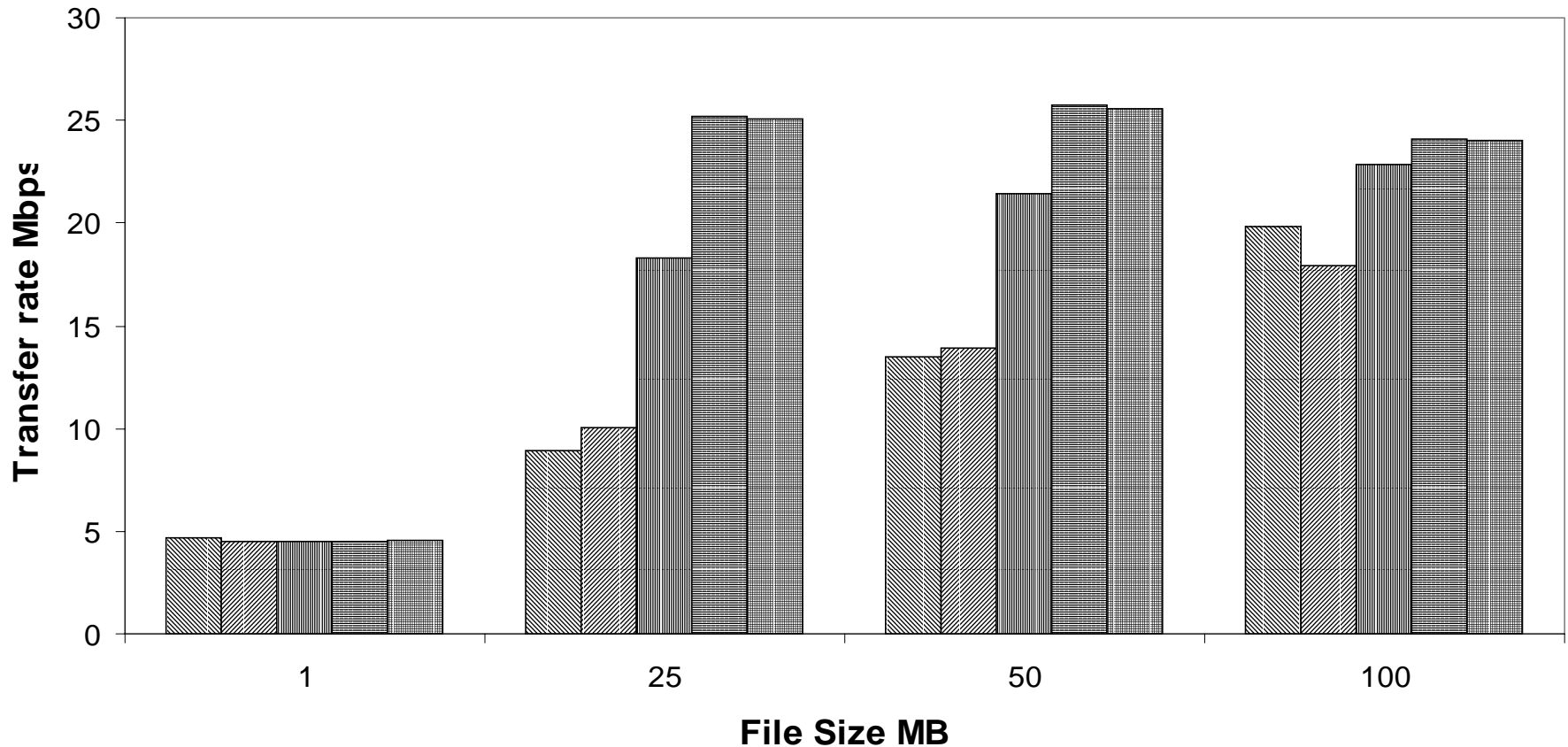


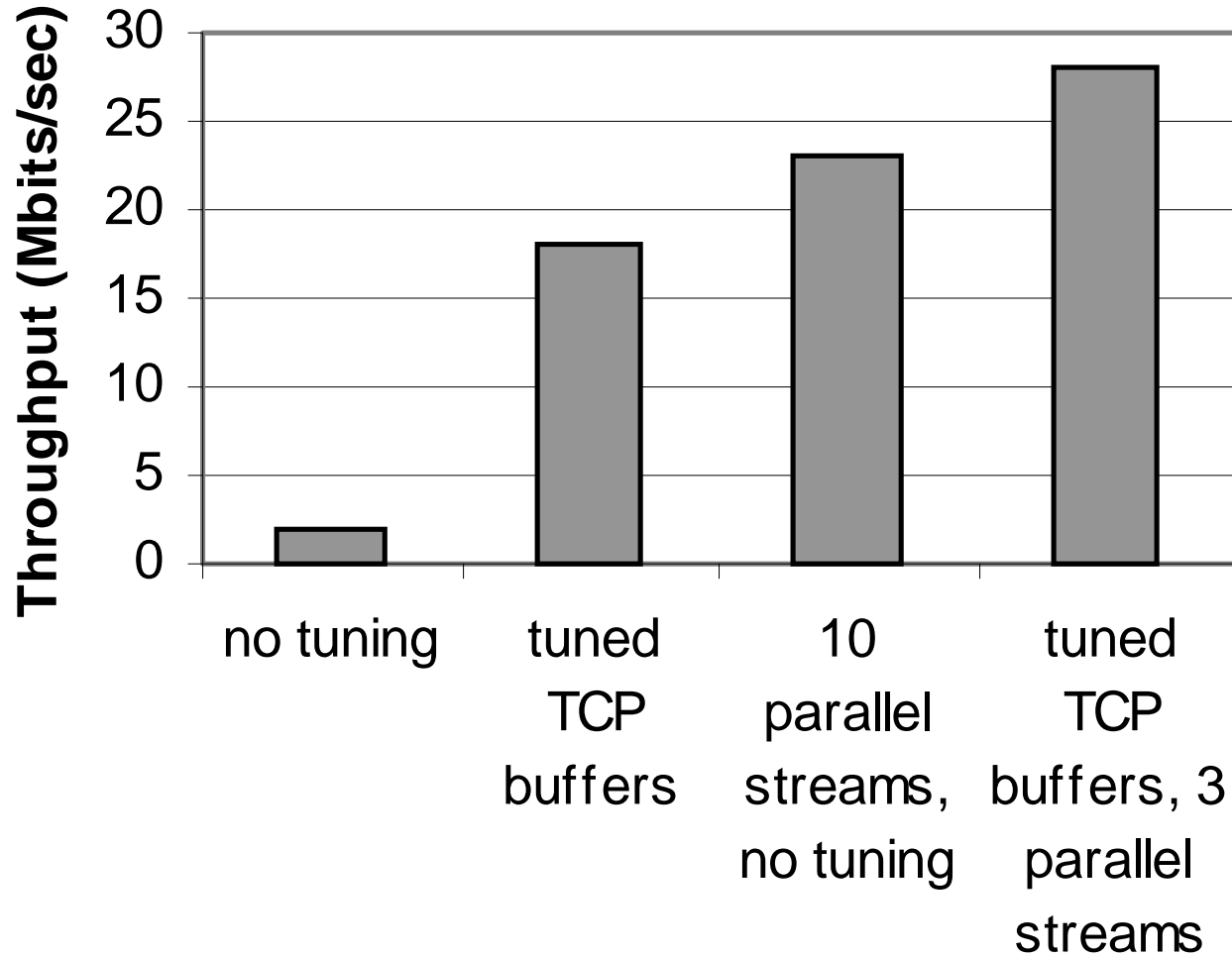
# Comparison of RFIO Get and GSIFTP Get (Tuned 1 MB Buffers)

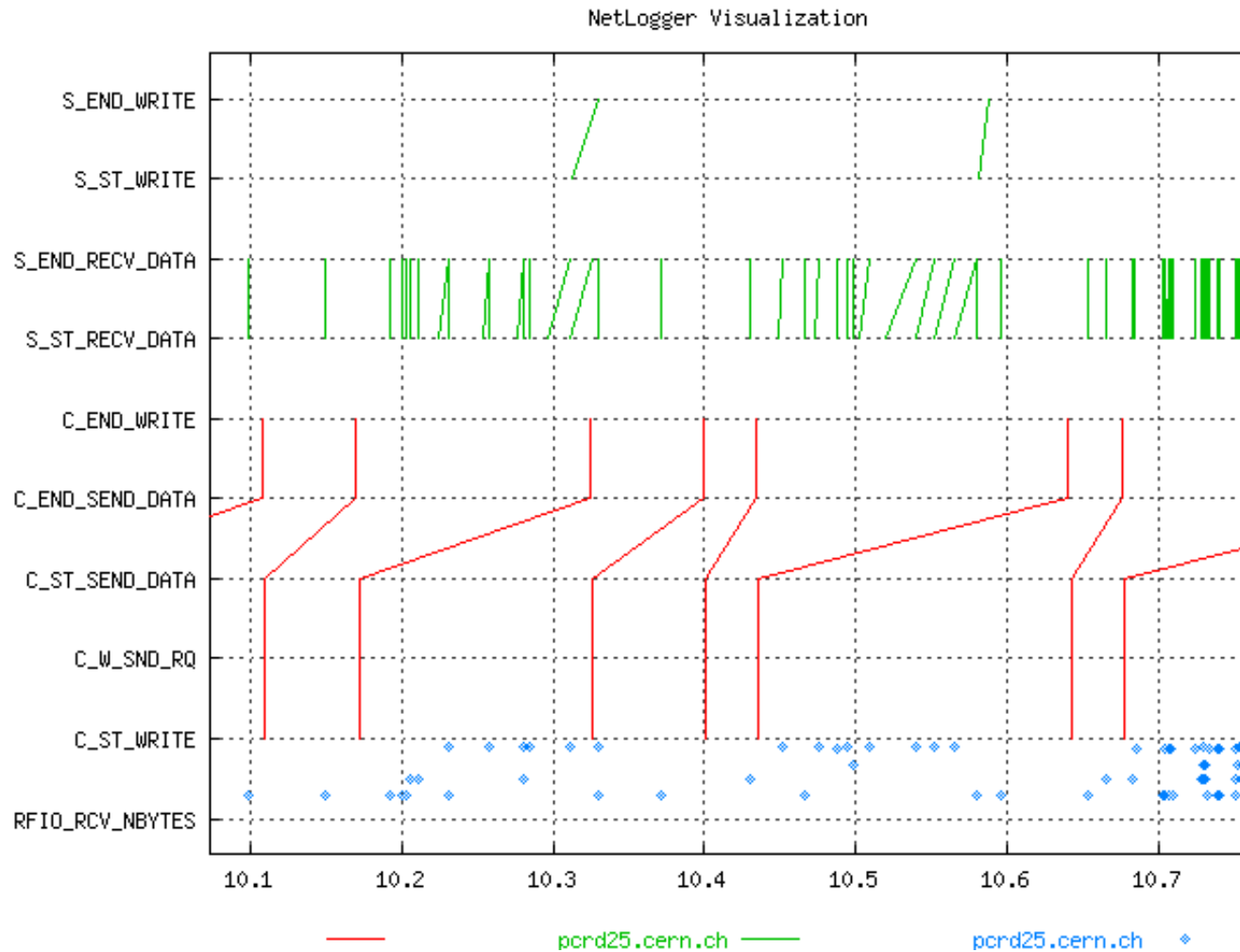


## RFIO Vs Gsiftp Tuned TCP Buffers

RFIO Gsiftp 1 stream Gsiftp 2 streams Gsiftp 3 streams Gsiftp 4 streams









# Linux TCP Bug found!



- **Using Netlogger, we discovered a Linux TCP bug (plus two GSIFTP bugs):**
  - e.g.: During CERN-LBL testing:
    - Linux to Solaris: throughput = 0.5 Mbps
    - Linux to Linux, Solaris to Solaris, etc. throughput = 25 Mbps (50 times slower!)
    - Due to an error computing RTO (Retransmit Timeout)
- Reported to Linux developers and bug fixed in kernels 2.2.18 upward and 2.4.0-test12 upward (Redhat 7.1).
- For more information, see:
  - <http://www-didc.lbl.gov/Linux-tcp-bug.html>



# Linux 2.4 Autotuning



- New feature that has been introduced in Linux kernel 2.4
  - TCP buffer sizes starts at 16 KB.
  - As the data transfer takes place, the buffer size is continuously readjusted
- The optimum buffer size for CERN – LBL link is about 1MB
  - throughput = 28 Mbps.
- The Linux 2.4 autotuning algorithm typically set the buffer size to about 200-400 KB (re-run this test!)
  - throughput = 22 Mbps
  - this is double the performance of the 2.2 kernel using 64 KB buffers
- Info on configuring Linux 2.4 autotuning is at:
  - <http://www-didc.lbl.gov/tcp-wan.html>



# Conclusions



- **Proper TCP buffer size setting is the single most important factor in achieving good performance.**
- **2-3 parallel streams will often gain an additional 25% performance over a single tuned stream.**
- **It is sometimes possible to get the same throughput as tuned buffers using untuned TCP buffers with enough parallel streams.**
- **Data transfer with tuned buffers is highly sensitive to variations in ambient network traffic.**