Science Data Movement: Deployment of a Capability

Joint Techs
Salt Lake City, UT
January 31, 2010

Eli Dart, Network Engineer
Energy Sciences Network (ESnet)
Lawrence Berkeley National Laboratory

ESnet
Energy Sciences Network

Supporting Advanced Scientific Computing Research • Basic Energy Sciences • Biological and Environmental Research • Fusion Energy Sciences • High Energy Physics • Nuclear Physics

BERKELEY LAB

U.S. DEPARTMENT OF ENERGY
Office of Science

Overview

- Bulk data movement – a common task
- Pieces of the puzzle
  - Network architecture
  - Dedicated hosts
  - Software tools
- Test, measurement and troubleshooting

Bulk Data Movement

- Common task at all data scales
- Driven by collaboration, distributed resources
  - Computing centers
  - Facilities
  - Major instruments (e.g. LHC)
- Fundamental to the conduct of science (scientific productivity follows data locality)
- Data sets of 200GB to 5TB are now common
- Often a difficult task for various reasons

Time to Copy 1 Terabyte

- 10 Mbps network : 300 hrs (12.5 days)
- 100 Mbps network : 30 hrs
- 1 Gbps network : 3 hrs (are your disks fast enough?)
- 10 Gbps network : 20 minutes (need *really* fast disks and filesystem)
- These figures assume some headroom left for other users

- Compare these speeds to:
  - USB 2.0 portable disk
    - 60 MB/sec (480 Mbps) peak
    - 20 MB/sec (160 Mbps) reported on line
    - 5-10 MB/sec reported by colleagues
    - 15-40 hours to load 1 Terabyte

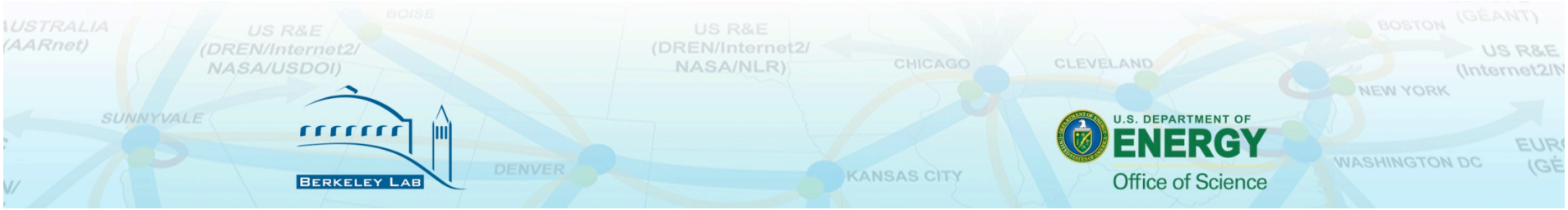
Data Throughput – Transfer Times

Bandwidth Requirements to move Y Bytes of data in Time X

Bits per Second Requirements

	1H	8H	24H	7Days	30Days
10PB	25,020.0 Gbps	3,127.5 Gbps	1,042.5 Gbps	148.9 Gbps	34.7 Gbps
1PB	2,502.0 Gbps	312.7 Gbps	104.2 Gbps	14.9 Gbps	3.5 Gbps
100TB	244.3 Gbps	30.5 Gbps	10.2 Gbps	1.5 Gbps	339.4 Mbps
10TB	24.4 Gbps	3.1 Gbps	1.0 Gbps	145.4 Mbps	33.9 Mbps
1TB	2.4 Gbps	305.4 Mbps	101.8 Mbps	14.5 Mbps	3.4 Mbps
100GB	238.6 Mbps	29.8 Mbps	9.9 Mbps	1.4 Mbps	331.4 Kbps
10GB	23.9 Mbps	3.0 Mbps	994.2 Kbps	142.0 Kbps	33.1 Kbps
1GB	2.4 Mbps	298.3 Kbps	99.4 Kbps	14.2 Kbps	3.3 Kbps
100MB	233.0 Kbps	29.1 Kbps	9.7 Kbps	1.4 Kbps	0.3 Kbps

This table available at http://fasterdata.es.net



Overview

- Bulk data movement – a common task
- Pieces of the puzzle
  - ➢ **Network architecture**
  - – Dedicated hosts
  - – Software tools
- Test, measurement and troubleshooting

Network Architecture

- Most LANs are not purpose-built for science traffic – they carry many types of traffic
  - Desktop machines, laptops, wireless
  - VOIP
  - HVAC control systems
  - Financial systems, HR
  - *Some science data coming from someplace*
- Bulk data transfer traffic is typically very different than enterprise traffic

Architecture – Enterprise Networks

- Business continuity
  - Risk management
    - Personally Identifiable Information (PII)
    - Financial information
    - Embarrassment due to security incidents
  - Relatively low bandwidth (100s of Mbps) unless there are a lot of users
- Unsophisticated user base from a computer security perspective
  - Lots of desktop boxes
  - Laptops, visitors (hosts that visit other networks)
- Need network-level policy controls to mitigate risk
  - Firewalls
  - Management of file sharing traffic (e.g. BitTorrent)

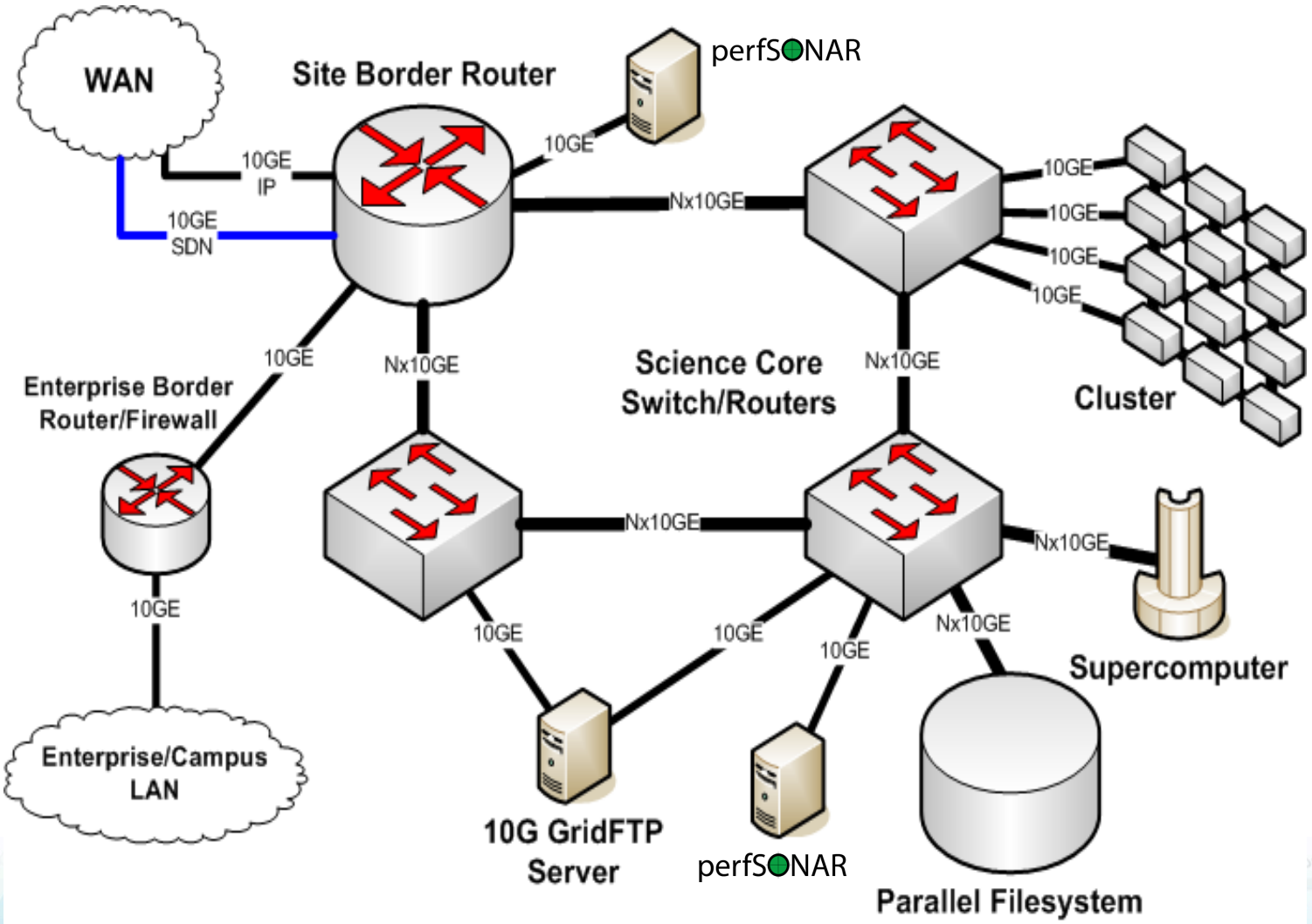
Architecture – Science Networks

- High bandwidth Requirement (10s of Gbps)
  - Not just in connection speed, but in delivered performance to computational, visualization and storage resources
  - Different tool set and traffic profile
    - This isn't for desktop boxes
    - Built for special-purpose hosts, e.g. data movers
- Relatively sophisticated users
- Sensitive to perturbations caused by security devices
  - Numerous cases of firewalls causing problems
  - Often difficult to diagnose
  - Router filters can often provide equivalent security without the performance impact

Science vs. Enterprise Networks

- Science and Enterprise network requirements are in conflict
- One possible remedy: build a science network for the science and attach the enterprise network to the science network
  - Put the Enterprise security perimeter at the edge of the enterprise network, not at the site border
  - Science resources are not burdened by Enterprise firewall configuration

Separate Enterprise and Science Networks



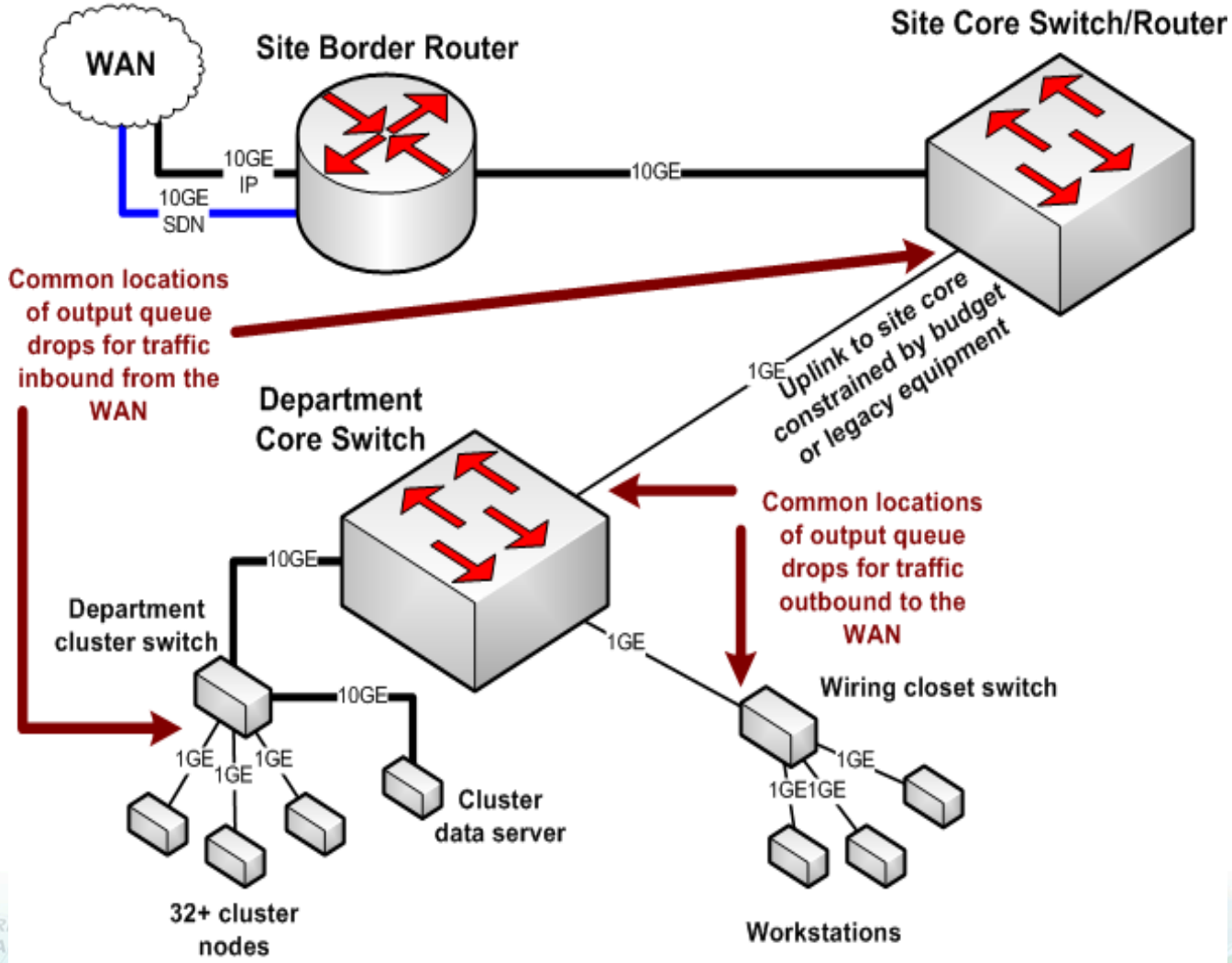
Network Pitfalls – Soft Failures

- "Soft Failures" are network problems that don't result in total loss of connectivity
  - The network (or a particular router or link) is up, but does not perform well
  - Problem often goes unnoticed until someone tries to use the WAN for high throughput
- Soft failure examples
  - Process switching ("punting")
  - Dirty fiber
  - Failing optics
  - Misconfigured buffers/queues
  - Routing table overflow in Cisco devices (causes punting)

Router and Switch Configuration

- Buffer/queue management
  - TCP traffic is bursty
  - Coincident bursts destined for a common egress port cause momentary oversubscription of the output queue
  - Traffic entering via a 10G interface and leaving via a 1G interface can cause oversubscription of output queue
  - Momentary oversubscription of output queues causes packet loss
  - Default configuration of many devices is inadequate
- Example: Cisco commands
  - ‘sho int sum’ – check for output queue drops
  - ‘hold-queue 4096 out’ – change from default 40-packet output queue depth
- See http://fasterdata.es.net/

Output Queue Oversubscription



Router and Switch Issues

- Need to have adequate buffering to handle TCP bursts
  - Long fat pipes mean large TCP windows which means large wire-speed bursts
  - Changes in speed (e.g. 10Gbps to 1Gbps)
- Many devices do not have sufficient output queue resources to handle bulk data flows
  - Need to understand device capabilities
  - When purchasing new equipment, require adequate buffering
- Many defaults assume a different traffic profile (e.g. millions of web browsers)
  - Drop traffic at first sign of oversubscription
  - Makes TCP back off because of packet loss
  - Protects flows such as VOIP and videoconferencing (remember the enterprise traffic?)
  - Non-default configuration is necessary

Network Architecture Summary

- Build a network to support bulk data transfers with data transfer in mind
  - Don't just plug it in anyplace
  - Avoid traversing Enterprise infrastructure if you can
  - Connect your bulk transfer resources as close to the border router as you can
- Configure routers and switches for adequate buffering
  - Watch drop counters (e.g. sho int sum or sho int queue)
  - Watch error counters
- If you have to, collocate your data server near the border router
  - On-site transfers to your server in another building will usually be high performance due to low latency
  - WAN transfers bypass the Enterprise infrastructure
  - Might be better for network administrators as well (no need to upgrade all the switches in the path to accommodate science traffic)

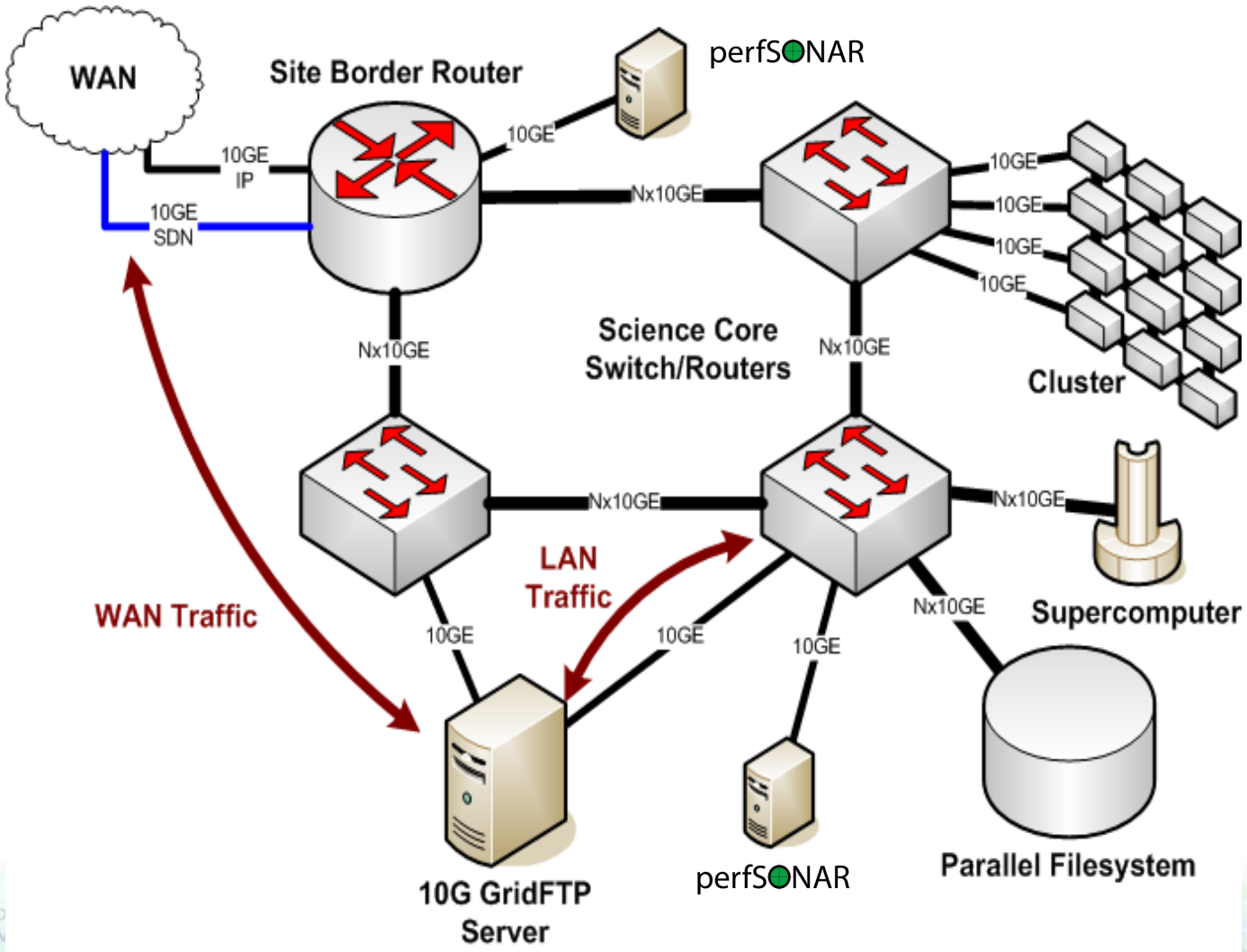
Overview

- Bulk data movement – a common task
- Pieces of the puzzle
  - Network architecture
  - ➢ **Dedicated hosts**
  - Software tools
- Test, measurement and troubleshooting

Data Transfer Nodes

- Reasons for dedicated hosts
  - One thing to test and tune
  - One place for large WAN flows to go (it's easier to give one host a special configuration than to do this for all workstations)
  - One set of firewall exceptions
- If you can, use different network connections for LAN and WAN flows
  - LAN flows can easily saturate network interfaces, especially 1Gbps interfaces
  - LAN flows recover quickly, unlike WAN flows

Internal / External Traffic Separation



Host Tuning – TCP

- TCP tuning commonly refers to the proper configuration of buffers that correspond to TCP windowing
- Historically TCP tuning parameters were host-global, with exceptions configured per-socket by applications
  - Applications had to understand the network in detail, and know how far away clients were
  - Some applications did this – most did not
- Solution: auto-tune TCP connections within pre-configured limits

Socket Buffer Autotuning

- To solve the buffer tuning problem, based on work at LANL and PSC, Linux OS added TCP Buffer autotuning
  - Sender-side TCP buffer autotuning introduced in Linux 2.4
  - Receiver-side autotuning added in Linux 2.6
- Most OS's now include TCP autotuning
  - TCP send buffer starts at 64 KB
  - As the data transfer takes place, the buffer size is continuously re-adjusted up max autotune size
- Current OS Autotuning default maximum buffers
  - Linux 2.6: 256K to 4MB, depending on distribution
  - FreeBSD 7: 256K
  - Windows Vista: 16M
  - Mac OSX 10.5: 8M
- Some defaults are still wrong!

Autotuning Settings (For 16MB Max)

- Linux 2.6

```
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
# autotuning min, default, and max number of bytes to use
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
```

- FreeBSD 7.0

```
net.inet.tcp.sendbuf_auto=1
net.inet.tcp.recvbuf_auto=1
net.inet.tcp.sendbuf_max=16777216
net.inet.tcp.recvbuf_max=16777216
```

- Windows Vista

```
netsh interface tcp set global autotuninglevel=normal
```

  - max buffer fixed at 16MB

- OSX 10.5 ("Self-Tuning TCP")

```
kern.ipc.maxsockbuf=16777216
```

- For more info, see: http://fasterdata.es.net/TCP-Tuning/

Congestion Control

- TCP senses network congestion by detecting packet loss
- Historically (TCP Reno) TCP used AIMD (Additive Increase, Mutiplicative Decrease) for window sizing in response to loss
- After loss, window opens back up very slowly – causes very poor performance
- Newer algorithms, available in Linux, offer higher performance than Reno
  - Cubic (now the default in several Linux distributions)
  - HTCP (Hamilton)
  - Others (see http://fasterdata.es.net/TCP-tuning/linux.html)

Congestion Algorithms In Several Distributions

Linux Distribution	Available Algorithms	Default Algorithm
Centos 5.2	bic	bic
Centos 5.3	bic	bic
Centos 5.4	bic	bic
Debian 5.0.3	bic	bic
Debian Unstable (future 6?)	cubic, reno	cubic
Fedora 10	cubic, reno	cubic
Fedora 11	cubic, reno	cubic
Fedora 12	cubic, reno	cubic
Redhat 5.4	bic	bic
Ubuntu 8.0	reno	reno
Ubuntu 8.10	cubic, reno	cubic
Ubuntu 9.04	cubic, reno	cubic
Ubuntu 9.10	cubic, reno	cubic

Dedicated Host Setup – Summary

- Use a newer OS that supports TCP buffer autotuning and congestion recovery
- Increase the maximum TCP autotuning buffer size
- Use a modern congestion control algorithm

Overview

- Bulk data movement – a common task
- Pieces of the puzzle
  - Network architecture
  - Dedicated hosts
  - ➢ **Software tools**
- Test, measurement and troubleshooting

Data Transfer Tools

- Parallelism is key
  - It is much easier to achieve a given performance level with four parallel connections than one connection
  - Several tools offer parallel transfers
- Latency interaction is critical
  - Wide area data transfers have much higher latency than LAN transfers
  - Many tools and protocols assume a LAN
  - Examples: SCP/SFTP, HPSS mover protocol

Sample Data Transfer Results: LBNL to BNL

- Using the right tool is very important
  - SCP/SFTP: 10 Mbps
    - standard Unix file copy tools
    - fixed 1 MB TCP window in OpenSSH
      - only 64 KB in OpenSSH versions < 4.7
  - FTP: 400-500 Mbps
    - assumes TCP buffer autotuning
  - Parallel stream FTP: 800-900 Mbps

GridFTP

- GridFTP from ANL has everything needed to fill the network pipe
  - Buffer Tuning
  - Parallel Streams
- Supports multiple authentication options
  - anonymous
  - ssh (available in starting with Globus Toolkit version 4.2)
  - X509
- Ability to define a range of data ports
  - helpful to get through firewalls
- Sample Use:
  - `globus-url-copy -p 4 sshftp://data.lbl.gov/home/mydata/myfile file://home/mydir/myfile`
- Available from: http://www.globus.org/toolkit/downloads/

newer GridFTP Features

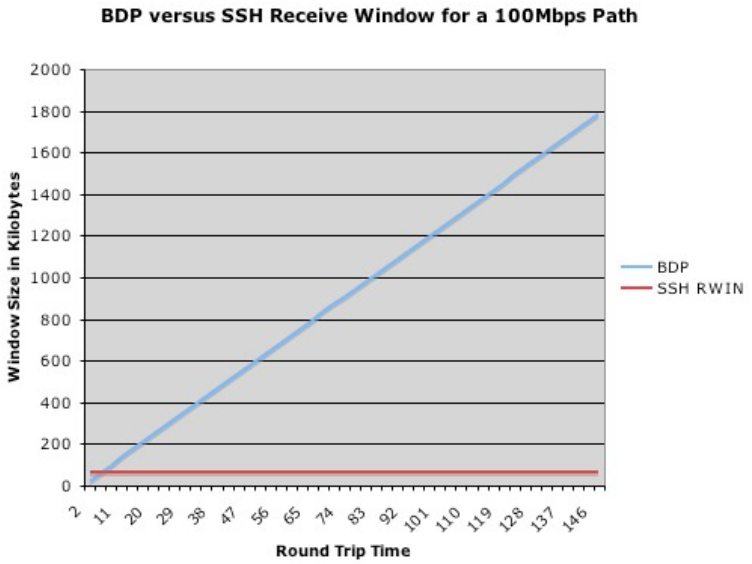
- ssh authentication option
  - Not all users need or want to deal with X.509 certificates
  - Solution: Use SSH for Control Channel
    - Data channel remains as is, so performance is the same
  - see http://fasterdata.es.net/gridftp.html for a quick start guide
- Optimizations for small files
  - Concurrency option (-cc)
    - establishes multiple control channel connections and transfer multiple files simultaneously.
  - Pipelining option:
    - Client sends next request before the current completes
  - Cached Data channel connections
    - Reuse established data channels (Mode E)
    - No additional TCP or GSI connect overhead
- Support for UDT protocol

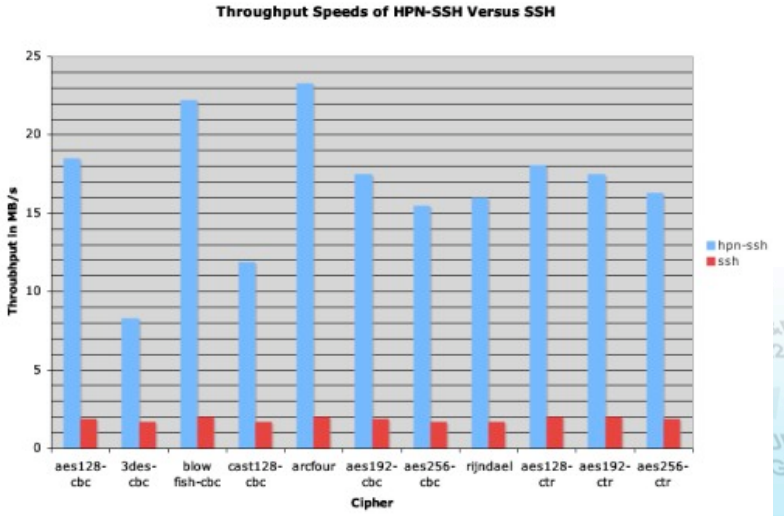
Why Not Use SCP or SFTP?

- Pros:
  - Most scientific systems are accessed via OpenSSH
  - SCP/SFTP are therefore installed by default
  - Modern CPUs encrypt and decrypt well enough for small to medium scale transfers
  - Credentials for system access and credentials for data transfer are the same
- Cons:
  - The protocol used by SCP/SFTP has a fundamental flaw that limits WAN performance
  - CPU speed doesn't matter – latency matters
  - Fixed-size buffers reduce performance as latency increases
  - It doesn't matter how easy it is to use SCP and SFTP – they simply do not perform
- Verdict: ***Do Not Use Without Performance Patches***

A Fix For SSH

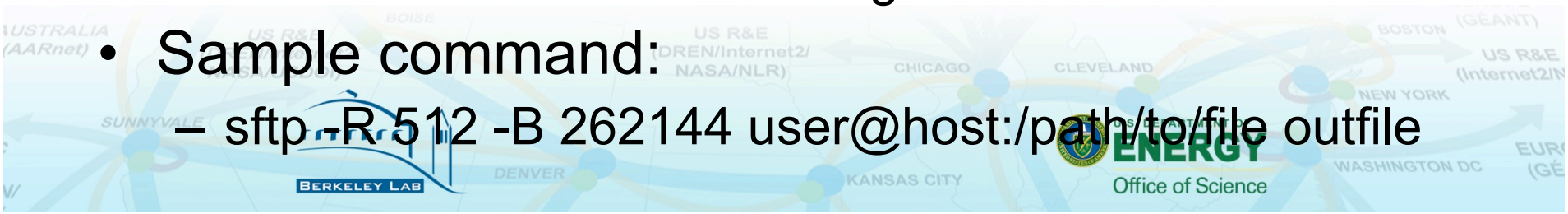
- PSC has a patch set that fixes problems with SSH
- http://www.psc.edu/networking/projects/hpn-ssh/
- Significant performance increase
- Advantage – this helps rsync too





SFTP

- Uses same code as SCP, so don't use SFTP for WAN transfers unless you have installed the HPN patch from PSC
- But even with the patch, SFTP has yet another flow control mechanism
  - By default, SFTP limits the total number of outstanding messages to 16 32KB messages.
  - Since each datagram is a distinct message you end up with a 512KB outstanding data limit.
  - You can increase both the number of outstanding messages ('-R') and the size of the message ('-B') from the command line though.
- Sample command:
  - sftp -R 512 -B 262144 user@host:/path/to/file outfile



Other Tools

- bbcp: http://www.slac.stanford.edu/~abh/bbcp/
  - supports parallel transfers and socket tuning
  - bbcp -P 4 -v -w 2M myfile remotehost:filename
- Tools page on fasterdata.es.net
  - More tools
  - Quick start guides
  - Additional information
  - http://fasterdata.es.net/tools.html

Overview

- Bulk data movement – a common task
- Pieces of the puzzle
  - Network architecture
  - Dedicated hosts
  - Software tools

➢ **Test, measurement and troubleshooting**

Test And Measurement – Why?

- **You can't fix what you can't find**
- **You can't find what you can't see**

- Typical scenario:
  1. Attempt to transfer data to some remote site
  2. Transfer fails or is slow
  3. Try to transfer locally – works fine
  4. Conclude that "the network is broken"
  5. Give up
- The problem could just as easily be the local switch, failing optics in somebody's box in another state, or a duplex mismatch at the far end – how do you know?

Test And Measurement Advantages

- It is typically difficult to get statistics and diagnostic output from every device in the path
  - Many administrative domains
  - Difficulty finding people, etc
- Test tools allow a person to locate the source of the problem (or at least make significant progress) without involving a large number of other people
- Many networks are now instrumented with perfSONAR, providing a common test and measurement framework

perfSONAR

- An open web-services based framework for collecting, managing and sharing network measurements
- A description of a common set of test and measurement tools
- A set of known-good measurement points (hosts) at strategic locations
  - Major network hubs
  - Site borders
  - Near major resources
- perfSONAR is being deployed across the science community

perfSONAR Deployments

- Internet2
- ESnet
- Argonne National Lab
- Brookhaven National Lab
- Fermilab
- National Energy Research Scientific Computing Center (NERSC)
- Pacific Northwest National Lab
- University of Michigan, Ann Arbor
- Indiana University
- Boston University
- University of Texas Arlington
- Oklahoma University, Norman
- Michigan Information Technology Center
- William & Mary
- University of Wisconsin Madison
- Southern Methodist University, Dallas
- University of Texas Austin
- Vanderbilt University
- APAN
- GLORIAD
- JGN2PLUS
- KISTI, Korea
- Monash University, Melbourne
- NCHC, HsinChu, Taiwan
- Simon Fraser University
- GEANT
- GARR
- HUNGARNET
- PIONEER
- SWITCH
- CCIN2P3
- CERN
- CNAF
- DE-KIT
- NIKHEF/SARA
- PIC
- RAL
- TRIUMF

Level 1 perfSONAR Deployment

- Several ways to deploy perfSONAR
- Level 1 provides simple throughput testing via bwctl/iperf
- Others provide increased levels of capability
- Just a simple throughput test host can be an amazingly powerful tool
  - It's not alone – there are many many other perfSONAR instances with bwctl servers
  - Diversity of test points allows for testing to different locations to narrow down the source of the problem
- http://fasterdata.es.net/ps_level1_howto.html

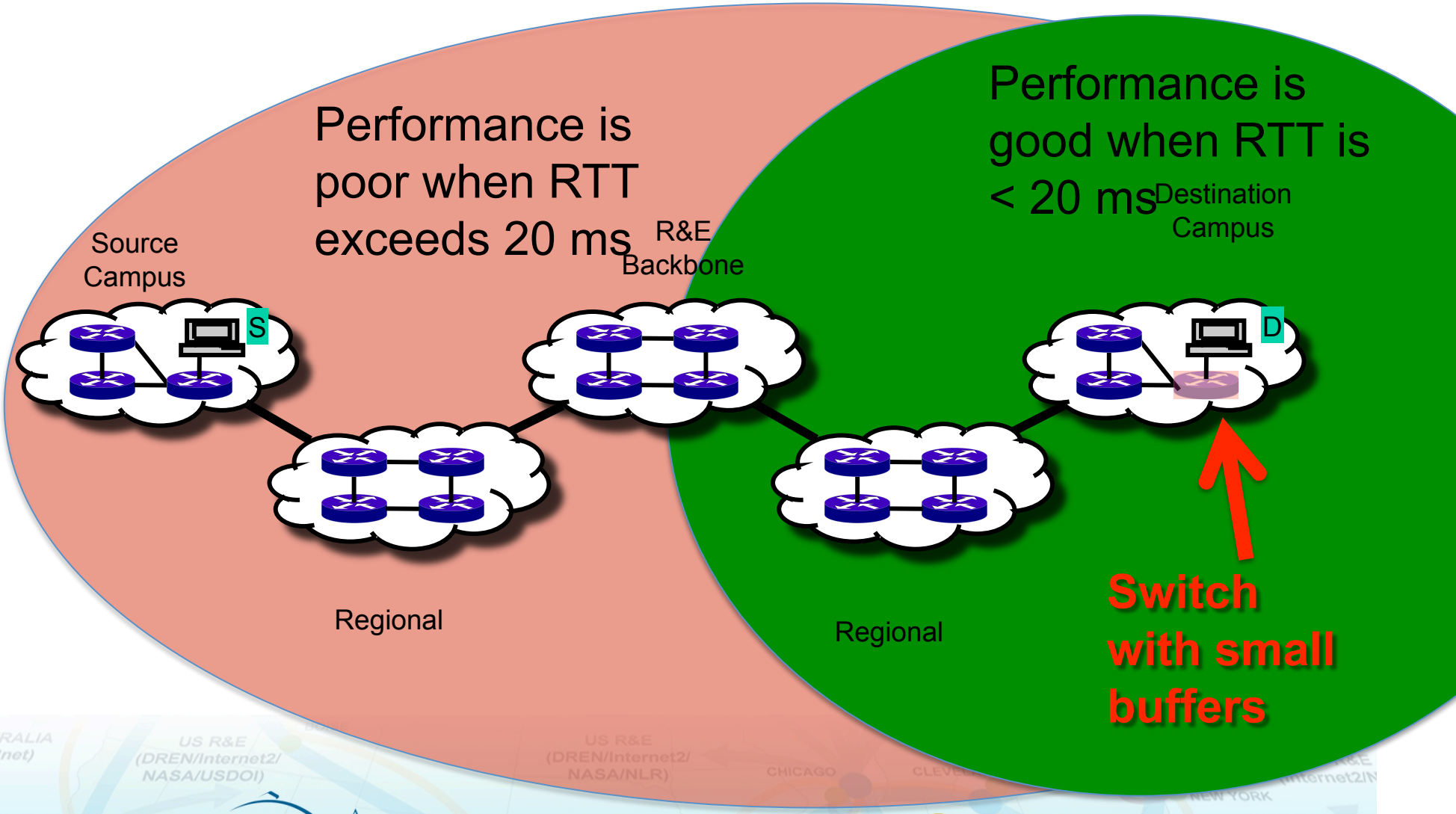
Test Methodology

- Base assumptions:
  - Hosts are set up properly (modern congestion control algorithm, TCP autotuning enabled and given sufficient headroom, no duplex mismatches, etc)
  - Using reasonable data transfer software (e.g. GridFTP)
  - Performance is still slow
  - There is a Level 1 perfSONAR host available
- Progressive testing along the path

Progressive Testing

- Test from local perfSONAR host to nearest perfSONAR host using bwctl (see http://www.perfsonar.net/deploy.html for list of servers)
- Then test to the next nearest, and so on
- If performance degrades significantly and consistently as you move the test endpoint further away, loss might well be local
- If everything is fine till you get to a particular location, and everything is bad after that, it is likely that something is wrong near that location
- However, small switch/router buffers can cause tricky problems (in this case, path diversity is key)

Short Distance Tests Aren't Adequate



How Useful Is perfSONAR?

- We have used perfSONAR extensively to test and measure our own network and to help solve problems with sites
- Every perfSONAR deployment has found problems
  - The utility of perfSONAR cannot be overstated
  - If you haven't verified your network tests clean, it is almost certain there are problems

Transfers Between DOE Supercomputer Centers

- Users were having problems moving data between supercomputer centers
  - One user was: “waiting more than an entire workday for a 33 GB input file”
- perfSONAR Measurement tools were installed
  - Regularly scheduled measurements were started
- Numerous issues were identified & corrected
- Dedicated data transfer nodes were built
  - Tuned for wide area data transfers
  - Typical disk-to-disk speeds currently 1.4Gbps to 2Gbps
  - http://www.nersc.gov/nusers/systems/DTN/transferspeeds.html

Importance of Regular Testing

- You can't wait for users to report problems and then fix them (soft failures can go unreported for years!)
- Things just break sometimes
  - Failing optics
  - Somebody messed around in a patch panel and kinked a fiber
  - Hardware goes bad
- Problems that get fixed have a way of coming back
  - System defaults come back after hardware/software upgrades
  - New employees may not know why the previous employee set things up a certain way and back out fixes
- perfSONAR makes is easy to collect, archive, and alert on throughput information
- If you know something went wrong on a particular day in the recent past, it is much easier to ask what changed and find the issue

Deploying perfSONAR In Under 30 Minutes

- There are two easy ways to deploy a perfSONAR host
- Level 1 perfSONAR install
  - Build a Linux machine as you normally would (configure TCP properly!)
  - Go through the Level 1 HOWTO
  - http://fasterdata.es.net/ps_level1_howto.html
  - Simple, fewer features, runs on your standard Linux build
- Use the PS Performance Toolkit boot CD
  - Use command line tool to configure if you like a CLI
  - Use Web GUI to configure if you like a GUI
  - http://psps.perfsonar.net/toolkit/
  - More features, runs from CD

Putting It All Together

1. TCP tuning is critical, but is now easy
   - Four lines in /etc/sysctl.conf to give autotuning room to help you
   - Make sure you're not stuck with TCP Reno
2. Build one host for WAN data transfers, make sure it's right
   - GridFTP, BBCP, HPN-SSH
   - Make sure TCP parameters are configured
3. Deploy at least a Level 1 perfSONAR host
   - Test and measurement are critical
   - Even if you're fabulously lucky and don't need it today, you'll need it tomorrow
4. Plug your hosts into the right place in the network
   - Move it to another building if you have to
   - The less Enterprise infrastructure you traverse, the better
   - Avoid all the cheap switches you can – shallow queues drop packets
   - Make sure your network devices are using the queues they have

Thanks!

- Questions?