DMOVER: Parallel Data Migration for Mainstream Users

Nathan T.B. Stone

Pittsburgh Supercomputing Center

DMOVER Origins

"Why am I here?"

– Adm. James Stockdale

"It's the *users*, stupid."

– almost Bill Clinton

- One particular user needed to migrate TBytes of data between PSC and SDSC with regularity, else not compute at PSC.

Mainstream User: Defined

mān'strēm yo͞o'zər (n.)

HPC researcher with a low tolerance for infrastructural instability or complexity.
Such users are perceived to be dependent upon resources and utilities delivering exceptional performance exactly as advertised every time.

Is that hard?

- At the time the DMOVER project was launched (late 2004) large data transfers were still:
  - Not running in parallel
  - Getting 10s of MB/sec, “on a good day”
    - Users occasionally reported <10 MB/sec
  - This was not well suited to inter-site migration of large datasets

Typical Storage Case

Recent survey of HSM utilization at PSC

- 93% of data stored in the HSM are transferred in "sessions" of 10 or more files
- For these sessions:
  - Average file size = 93 MB
  - Average number of files = 378
- Therefore:
  - *Typical* users need solutions for large file count but modest file sizes

GridFTP Client Survey

- globus-url-copy
  - The early standard (functionality only)
  - Good for single files, but command lines are “long”
- “Striped GridFTP” (later feature of GUC)
  - Good for large (GBs) files, but still singles
- uberFTP
  - Great user interface
    - Editor’s Choice: for browsing and small file count
  - Supports a parallel mode
    - But all streams terminate at the client’s host ☹

GridFTP Client Survey (cont.)

- Reliable File Transfer (RFT)
  - "What we've all been waiting for" (Apr05)
  - Good for large file count
  - Parallel
  - Asynchronous
  - Reliable (automagically retries failures)
  - Editor's Choice: for performance and function

  …

  As long as it's between two GridFTP servers
  - But this is not the case for all sites
  - And, with the ever-expanding ETF, it is likely to get harder, not easier

0111010001100101011100100110000101110011011000110110000101101100011001010

What if…

- Your site could not run a GridFTP server (daemon) on your file server (host)?

➢ You'd need a different approach…

- That happened to be our situation at PSC
  - The GridFTP server relies upon the Globus Toolkit
    …but the Globus Toolkit (v4) does not build on Tru64

- So we cannot present our LeMieux:/scratch parallel file system to the ETF via GridFTP

…Make Lemonade

Faced with the fact that:

- There is no GridFTP server that can run on the LeMieux file servers (yet)
- Users store their large datasets in LeMieux /scratch (a PFS)
- Users want to migrate data from this location to other ETF sites

We choose to:

- Run local distributed clients!

The DMOVER Strategy

- Use the batch system to acquire local nodes for parallel transfer clients
- Use a process manager to farm out parallel streams
- Use Qsockets for optimal transfer bandwidth

Qsockets

- PSC-specific network optimization
- An intercept library to divert TCP socket operations through:
  - An RPC library, for setup/tear-down & ioctl
  - QSNet, for send/recv
- Works with legacy binaries, client & server
- Acts as a client-side library, communicating with a "Qserver" process on the Application GateWay (AGW) nodes
  - Qserver acts as the ultimate client (or server) relaying the data back to the application via QSNet

Application GateWays (AGWs)

- Multi-home servers connecting LeMieux internal compute nodes to the ETF network
- Impedance-matching
  - 1 QSNet (250+ MB/sec DMA)
  - 2 GigE (110+ MB/sec)
  - Two virtual servers ("Qservers") to each node
- Co-scheduled via PBS with compute jobs

Qsockets Efficiency

- At SC'04 we competed in the Bandwidth Challenge
  - Transfer as much data as possible via a scientific application
- Our application: Writing remote checkpoints from a running application

Results:

- 32 writers over 32 AGW nodes achieved an *average* of 31.1 Gbps (97% network BW)

DMOVER

Three Portable Scripts…

01110100011001010111001001100001011100110110001101100001011011000110010101

Script 1. Batch Script (Bash)

```
#PBS -l rmsnodes=4:4
#PBS -l agw_nodes=4

# root of the file(s)/directory(s) to transfer        (a convenience)
export SrcDirRoot=$SCRATCH/mydata/

# path to the target sources, relative to SrcDirRoot (wildcards allowed)
export SrcRelPath="*.dat"

# destination host name                                (one or more, round-robin)
export DestHost=tg-c001.sdsc.teragrid.org,
tg-c002.sdsc.teragrid.org,tg-c003.sdsc.teragrid.org,
tg-c004.sdsc.teragrid.org

# root of the file(s)/directory(s) at the other side (dest path)
export DestDirRoot=/gpfs/ux123456/mydata/

# run the process manager
/scratcha1/dmover/dmover_process_manager.pl "$SrcDirRoot" "$SrcRelPath"
   "$DestHost" "$DestDirRoot" "$RMS_NODES"
```

Script 2. Process Manager (Perl)

```
for ($i=0; $i<=$#file; $i++){
  # pick host IDs, unless we just got them from wait()
  if ($i<$nStreams){
    $shostID = $i % $ENV{'RMS_NODES'};
    $dhostID = $i % ($#host+1);
    $dest=$host[$dhostID];
  }

  # command to launch the transfer agent
  $cmd = "prun -N 1 -n 1 -B `offset2base $shostID`
$DMOVERHOME/dmover_transfer.sh $SrcDirRoot
$file[$i] $dest $DestDirRoot $shostID"

  $child = fork();
  if ($child){
    $cid{$child}[0] = $shostID;
    $cid{$child}[1] = $dhostID;
  }
  if (!$child){
    $ret = system($cmd);
  }

  # keep the number of streams constant
  if ($nStreams<=$i+1){
    $pid = wait;
    # re-use whichever source host just finished...
    $shostID = $cid{$pid}[0];
    # re-use whichever remote host just finished...
    $dhostID = $cid{$pid}[1];
    delete($cid{$pid});
  }
}

while (-1 != wait){
  sleep(1);
}
```

Script 3. Transfer Agent (Bash)

```
export X509_USER_PROXY=$HOME/.proxy
export GLOBUS_LOCATION=/usr/local/globus/globus-2.4.3
export GLOBUS_HOSTNAME=`/bin/hostname -s`.psc.edu
. $GLOBUS_LOCATION/etc/globus-user-env.sh
# set up Qsockets
. $DMOVERHOME/agw_setup.sh $5

SrcDirRoot=$1
SrcRelPath=$2
DestHost=$3
DestDirRoot=$4

args="-tcp-bs 8388608"
cmd="$GLOBUS_LOCATION/bin/globus-url-copy $args file:///$SrcDirRoot/$SrcRelPath
      gsiftp://$DestHost/$DestDirRoot/$SrcRelPath"
echo `/bin/hostname -s` : $cmd

time agw_run $cmd
```

User Documentation (already!)

- Check out:
  - http://teragrid.psc.edu/lemieux/jobs.html#dmover

- And the user said?

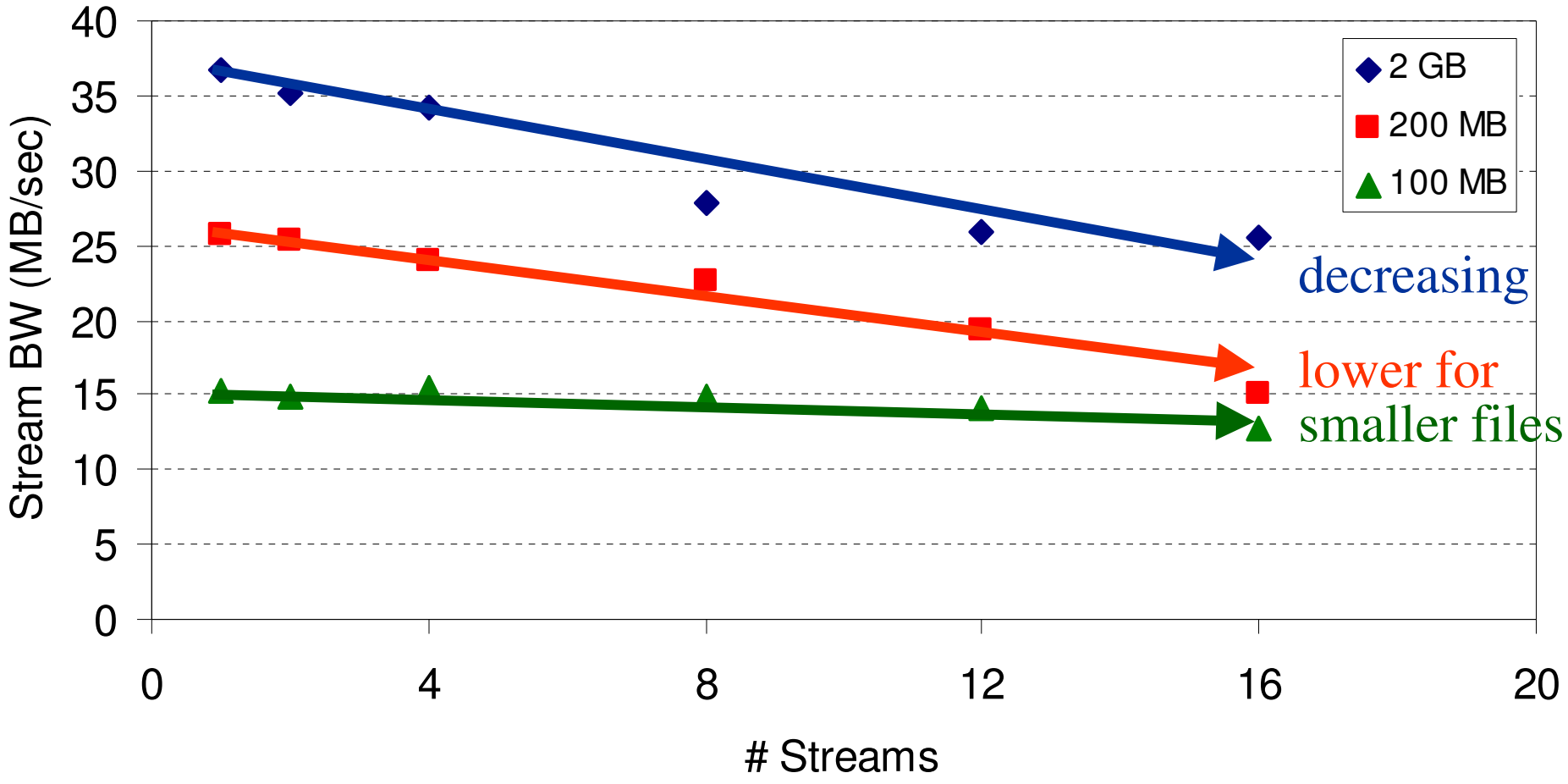
"I moved a directory containing **516 files** from /scratch1 on Lemieux to /gpfs on our TeraGrid IA-64 system. The total size was **134GBytes** and the transfer took around 10 minutes, or roughly **200 MByte/sec**. Very nice!
Thank you so much for getting me past the globus roadblock."

And so on, to TBytes…

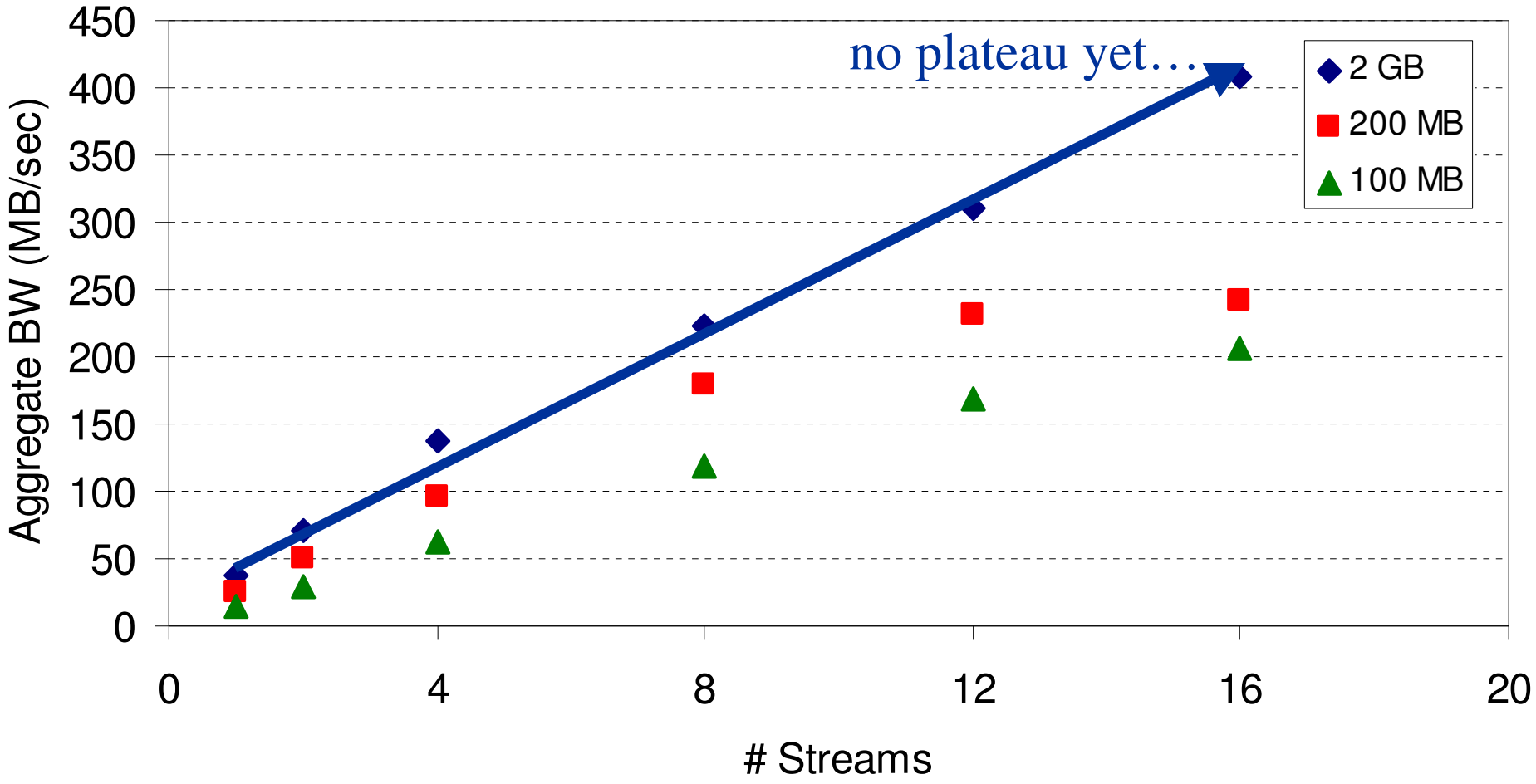
Performance and Portability

"Promises may make friends,
but 'tis *performances* that keep them."
– German proverb

Per-Stream BW



Aggregate BW



Scalability

- Per-stream BW: decreases with increasing stream count
  - Indicates poor scaling of the underlying PFS
    (NB: <12 streams is always one per host)
- Aggregate BW: no clear plateau yet, so more streams would still save wall-time (in case of emergency)
- Smaller files (even 100's of MB) suffer from per-session overhead
  - No cure in sight for this…
- Ultimately limited by the PFS performance
  - And where most of the hard work goes…

Portability

- All HPC sites have schedulers
- These scripts could run anywhere
- Although:
  - Qsockets lines are PSC-specific
    - But this feature is not likely to be needed elsewhere

0111010001100101011100100110000101110011011000110110000101101100011001010

Questions?

Nathan Stone

<nstone@psc.edu>

http://www.psc.edu/~nstone/

PSC Advanced Systems Group

http://www.psc.edu/advanced_systems/

Whitepapers for ongoing work at PSC

http://www.psc.edu/publications/tech_reports/