

Transferring large files to and from HPCx: a comparison of (parallel) ‘gridftp’ and ‘scp’

Chris Johnson¹ and Stephen Booth¹,

¹*EPCC, The University of Edinburgh, James Clerk Maxwell Building,
Mayfield Road, Edinburgh, EH9 3JZ, UK*

April 6, 2006

Abstract

With many data files being produced on HPCx with sizes in the Gbyte range, the question of how best to transfer these files to and from HPCx to remote machines is an important one. In particular we focus here on the transfer of large files between HPCx and a machine in Manchester, giving results which will be interesting for users wanting to transfer data from the CSAR service to the HPCx service. Here we give a short description of the findings of some experiments performed to test the rate of transfer of binary files of up to 2 Gbyte in size. We compare the use of ‘scp’ and ‘gridftp’ and find that for larger files, `gridftp` appears faster than `scp`, especially when the parallel data connection option is enabled (where available). Optimum values for the number of parallel data connections are explored. It was noted that in some case behaviour varied significantly depending on the direction of transfer.

1 Introduction

This short report highlights the findings of experiments in the transfer of binary files to and from HPCx [Ltd02]. The files tested were of various sizes up to 2 Gbyte and consisted of binary floating point numbers, which were randomly generated to suppress the compressibility of the files produced. Tests were first performed using the default options for Unix secure copy (`scp`) and globus `gridftp` (`globus-url-copy`). Various options for enabling parallel data connections using `gridftp` were then tested. Timings were repeated to identify anomalies which may have been due to high network usage. Timings for transfers which failed, or took exceptionally long, were discarded. The speed of copying files within local file systems was tested by timing local copying of large files.

The machines considered in this report consisted of HPCx itself, where the transfers were performed to and from the `/work` disk; a machine in Manchester provided as part of the National Grid Service (NGS) [JIS05] and a typical machine in at EPCC in Edinburgh, where the machine's local disk was used. The motivation for testing the connection between Manchester and HPCx, located nearby in Daresbury, was that it will be useful for users wishing to transfer files between the CSAR [con98] service at Manchester and the HPCx service. For comparison, we used a local machine at EPCC, located some distance from HPCx both physically and in terms of the topology of the network.

In order to use `gridftp` to transfer files, a user is required to have a valid globus certificate. More information about obtaining a valid certificate from the UK e-Science CA (Certificate Authority) can be found here [GOS05].

2 Commands used

2.1 The 'gridftp' facility (globus-url-copy)

The 'gridftp' facility is invoked via the command `globus-url-copy` [All03]. An example of a copy command is given here:

```
globus-url-copy -vb -p 8 \  
gsiftp://login.hpcx.ac.uk//hpcx/home/z001/z001/chrisj/work/bin2G \  
file:///home/ngs0330/bin2G
```

In this case the command is being run on the command line of a machine located in Manchester. The URL prefix `gsiftp://` refers to the remote machine (HPCx) and the URL prefix `file://` refers to the local destination to which the file will be copied. The flag `-vb` enables verbose output and the flag `-p 8` enables 8 parallel data streams to speed up the data transfer.

2.2 The 'scp' facility

The `scp` facility does not have many options available and was used in the following way:

```
scp \  
chrisj@login.hpcx.ac.uk://hpcx/home/z001/z001/chrisj/work/bin2G \  
/home/ngs0330/bin2G
```

Again, this command was run on the machine in Manchester with HPCx as the remote source.

2.3 Method of testing

Commands of the above form were run through the Unix `time` facility to determine the transfer rate of files. Files of various sizes were transferred backwards and

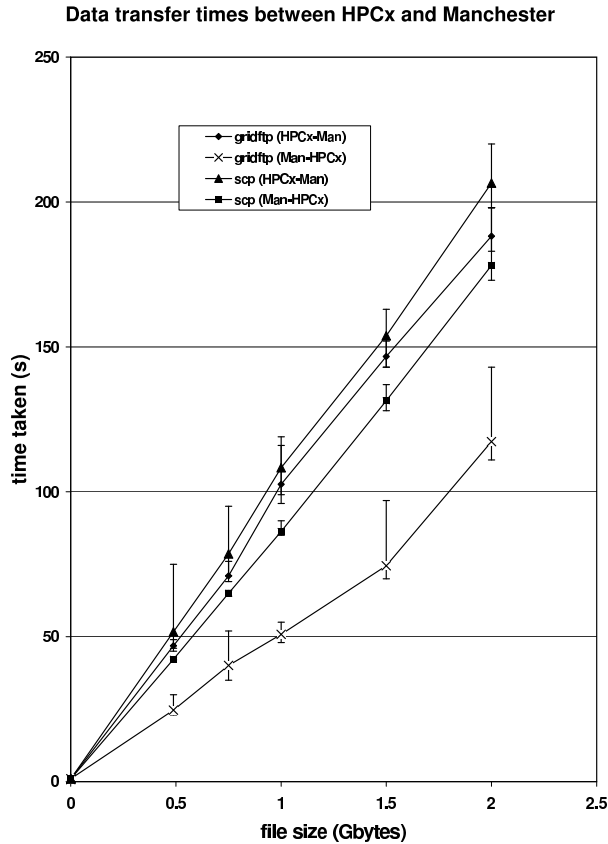


Figure 1: Time taken to transfer files between HPCx and Manchester.

forwards between machines. Parallel options were also investigated for `gridftp` when transferring files between Manchester and HPCx. The parallel options did not appear to work between EPCC and HPCx.

3 Results

3.1 Transfer between HPCx and Manchester: `gridftp` and `scp` comparison

Figure 1 shows the times taken to transfer files of various sizes between a machine in Manchester (`grid-data.man.ac.uk`) and HPCx (`login.hpcx.ac.uk`). Table 1 shows all of these results as a bandwidth (excluding the transfer of files below 0.5 Gbyte). It can be seen that, when considering either direction of transfer, `gridftp` offers the better rate of transfer.

The “error-bars” actually show the range of values achieved for 10 executions of

Table 1: Bandwidths measured when transferring files of 1-2 Gbyte between the three machines considered in this report.

Source location	Destination	transfer method	No. parallel data connections	Transfer Bandwidth (Mbytes/s)
Edinburgh	HPCx	gridftp	1	2
Edinburgh	HPCx	scp	1	2
HPCx	Edinburgh	gridftp	1	3
HPCx	Edinburgh	scp	1	2
HPCx	Manchester	gridftp	1	11
Manchester	HPCx	gridftp	1	20
HPCx	Manchester	scp	1	10
Manchester	HPCx	scp	1	12
HPCx	Manchester	gridftp	1	10
HPCx	Manchester	gridftp	2	22
HPCx	Manchester	gridftp	4	35
HPCx	Manchester	gridftp	6	47
HPCx	Manchester	gridftp	8	37
HPCx	Manchester	gridftp	16	37
Manchester	HPCx	gridftp	1	19
Manchester	HPCx	gridftp	2	22
Manchester	HPCx	gridftp	4	18
Manchester	HPCx	gridftp	6	26
Manchester	HPCx	gridftp	8	13
Manchester	HPCx	gridftp	16	11

the copying command.

One should take into account the speed at which files are copied to and from disk within a local system. Figure 6 shows the time taken to perform a local copy of files of various sizes on the three machines considered in this report.

3.2 Transfer between HPCx and Manchester: parallel gridftp

Figures 2 and 3 show timings of data transfer using `gridftp` with the parallel option enabled. This option was only tested for large files (1-2 Gbyte). Using 6 parallel data connections appears to give the best performance here.

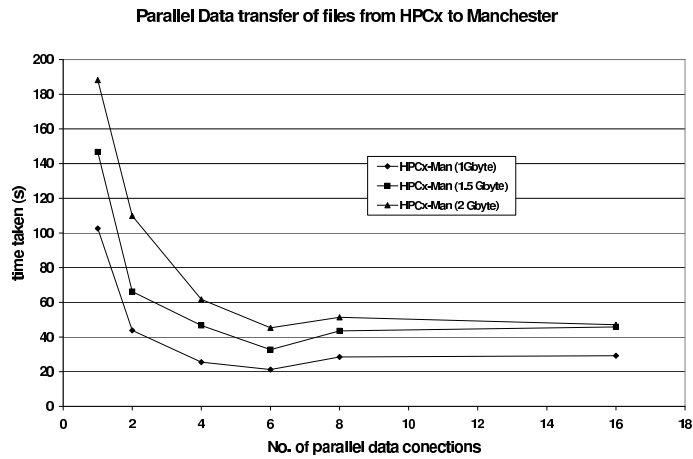


Figure 2: Time taken to transfer files between from HPCx to Manchester using parallel data connections.

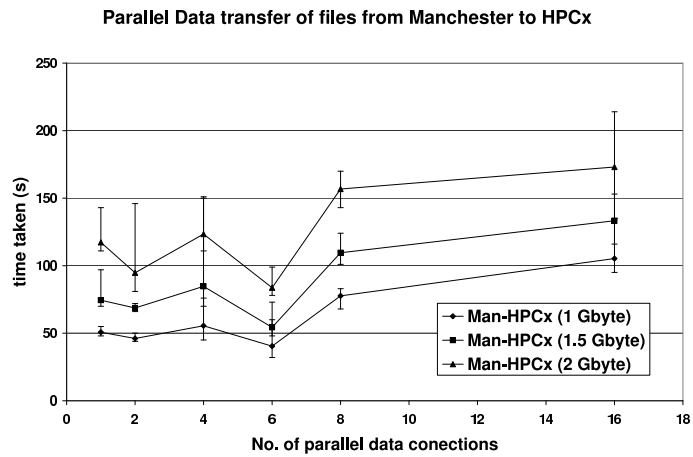


Figure 3: Time taken to transfer files between from Manchester to HPCx using parallel data connections.

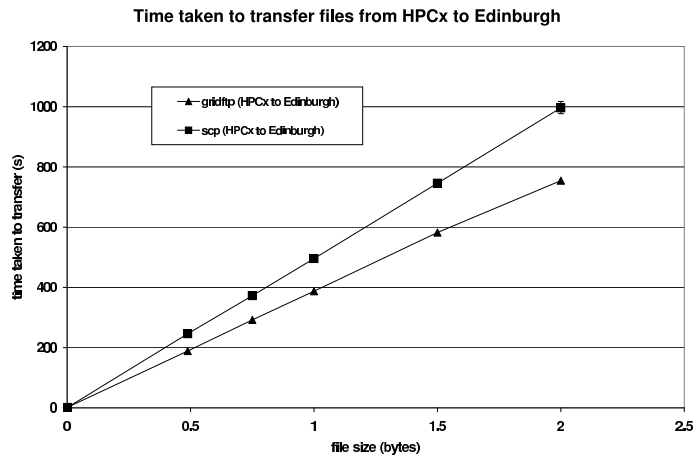


Figure 4: Time taken to transfer files between from HPCx to Edinburgh.

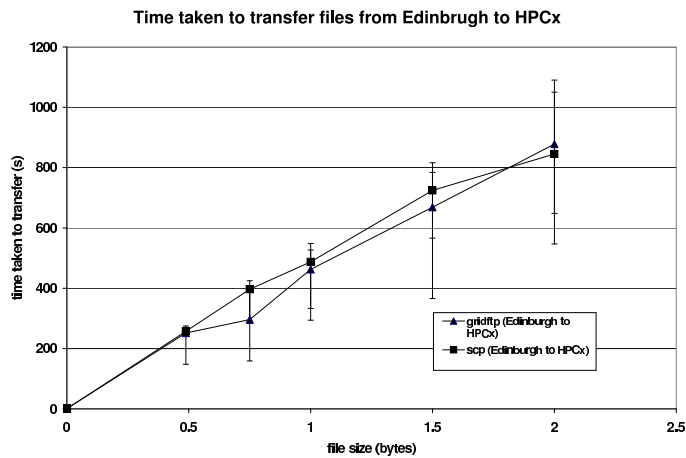


Figure 5: Time taken to transfer files from Edinburgh to HPCx.

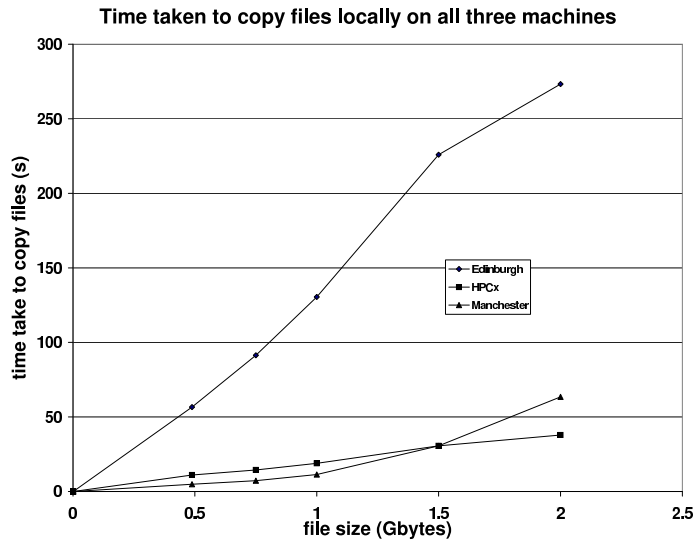


Figure 6: Time taken to perform a local copy of files on each of the three machines considered in this report.

3.3 Transfer between HPCx and Edinburgh

The rate of transfer between HPCx and Edinburgh differs significantly depending on the direction of transfer. Figure 4 shows the file transfer times with HPCx as the source and a machine at EPCC as the destination. Figure 5 shows the results for the reverse direction. In both cases the script which initiated the copying commands was run on the machine at EPCC. Table 1 shows the results as a bandwidth.

Again, the “error-bars” actually show the range of values achieved for 10 executions of the same copying command ¹.

It is clear that data transfer in the direction HPCx-Edinburgh shows a stable transfer rate and `gridftp` is faster for all file sizes above 1 Mbyte.

Analysing the transfer rate in the other direction is not as straightforward. As can be seen from the wildly varying timing results there is no clear winner here and there is not much performance gain in using one method rather than the other.

Some investigation using the `traceroute` facility along with a study of JANET [taU94], the UK’s education and research network, reveals that the route taken by files when being transferred is dependent on the direction of transfer. This is one possibility for the cause of this direction-dependent behaviour.

Again one should consider the speed of local file copying shown in figure 6.

¹If the copy failed for some reason the result considered as void, hence in two cases there were only 9 executions recorded, most were repeated 10 times. This did not appear to affect the results significantly

4 Conclusion

The speed and bandwidth of file transfer between HPCx and two other machines within the UK was measured when using two methods of file transfer, namely `scp` and `gridftp`. In general `gridftp` appears to provide an increase in bandwidth over `scp` for files in the Gbyte range, and gives even better results when run in parallel mode where this option is available. In the case of transferring files between Manchester and HPCx, the optimal number of parallel data connections was found to be 6. Our results show a more stable bandwidth when transferring data from HPCx to a remote machine. The reasons for this were not fully investigated. The copying of files locally on the machine at EPCC was relatively slow and this may have had an effect on the apparent transfer rate when copying files to and from Edinburgh. Again, this was not fully investigated.

References

- [All03] Globus Alliance. Website, 2003. <http://www.globus.org/>.
- [con98] CfS consortium. Website, 1998. <http://www.csar.cfs.ac.uk/>.
- [GOS05] GOSC. Website, 2005. <http://www.grid-support.ac.uk/>.
- [JIS05] JISC/GOSC. Website, 2005. <http://www.ngs.ac.uk/>.
- [Ltd02] UoE HPCx Ltd. The hpcx website. Website, 2002. <http://www.hpcx.ac.uk/>.
- [taU94] JNT Association trading as UKERNA. Website, 1994. <http://www.ja.net/>.