

Using BBCP

Overview

bbc_p is a point-to-point network file copy application developed at SLAC. It was initially used by the [BaBar Collaboration](#) to transfer massive amounts of data between numerous sites of the Collaboration. The protocol has been specifically designed for WAN. In particular, it allows sending data in multiple simultaneous streams.

This document gives only some tips that are intended to help readers to begin using the tool to transfer LCLS data. To find the complete documentation on the tool please visit the official site: <http://www.slac.stanford.edu/~abh/bbcp/>.

Installation

bbc_p is easy to install. Installation basically involves placing the bbc_p executable in your path on all the systems you want to use it on. All standard methods of authentication can be used: passwords and certificates. Keep in mind, though, that we're only going to support the password-based authentication based on SLAC UNIX ([Kerberos](#)) accounts.

Versions of bbc_p are available for most major flavors of UNIX, including Linux and Solaris.

- Note that bbc_p is not available for Windows. Should this be a problem consider other methods of data exportation which are presented in the [parent](#) document.

Since bbc_p is a peer-to-peer application no permanently running server process is required - you just invoke the bbc_p executable on a source machine and in response another bbc_p executable is started on the target machine. A more sophisticated scenario involves using the three-party scheme: the **source** and **target** machines do not need to be the same (**third**) machine that you initiate the file transfer from. Please, consult the official bbc_p documentation for further detail on that subject. At the mean time, all examples presented in the current document will be based on a simple two-party mode.

You can either download a [pre-build](#) version of bbc_p or build it yourself from [source](#) (or clone the git repo: **git clone** <http://www.slac.stanford.edu/~abh/bbcp/bbcp.git>). For more information check the download section in the [bbc_p documentation](#).

Make sure the bbc_p executable is available in the executable search path at both sides.

bbc_p on psexport

bbc_p is installed in `/usr/bin/bbcp` on the psexport machines.

Usage

First tests

The first test would be to transfer something on the same host. In the following example, you'll be transferring data from the special device (generating zeroes) into the null sink:

```
% bbcp -P 2 localhost:/dev/zero /dev/null
user@localhost's password:
bbcp: Creating /dev/null/zero
bbcp: At 090413 17:37:56 copy 0% complete; 811858.2 KB/s
bbcp: At 090413 17:37:58 copy 0% complete; 819277.7 KB/s
bbcp: At 090413 17:38:00 copy 0% complete; 827783.2 KB/s
..
```

The "-P" option would tell the tool to print the statistics every 2 seconds. The tool may ask a user for his/her password on the "local" machine because it will be making SSH login (similar to 'ssh user@localhost') and launching the bbc_p executable on the server side.

Transferring data in multiple streams

By default bbc_p will use 4 simultaneous streams to send data over the network. You can increase (or decrease) that number by using the "-s" option and specified the desired number of streams:

```

% bbcp -P 2 -s 32 localhost:/dev/zero /dev/null
user@localhost's password:
bbcp: Creating /dev/null/zero
bbcp: At 090413 18:12:04 copy 0% complete; 113401.7 KB/s
bbcp: At 090413 18:12:06 copy 0% complete; 113695.8 KB/s
bbcp: At 090413 18:12:08 copy 0% complete; 113981.2 KB/s
..
% bbcp -P 2 -s 1 localhost:/dev/zero /dev/null
user@localhost's password:
bbcp: Creating /dev/null/zero
bbcp: At 090413 18:13:10 copy 0% complete; 99670.7 KB/s
bbcp: At 090413 18:13:12 copy 0% complete; 100668.6 KB/s
bbcp: At 090413 18:13:14 copy 0% complete; 101580.4 KB/s
bbcp: At 090413 18:13:16 copy 0% complete; 101393.9 KB/s
..

```

The optimal number of streams depends on many factors. For instance, there is no improvement to have more than 1 stream when transferring data over LAN. And that is exactly the situation which is seen in the last example. For WAN it often helps to increase the number of streams above the default number of 4. In our experience we could get the best (in a specific setup of transferring data between SLAC and IN2P3/Lyon/France) with 32 streams. In that case the transfer speed was nearing 38 MBytes/sec.

Increasing the TCP window size

Another parameter worth to consider is so called [TCP window](#) size. By default `bbcp` would set it to 64 KBytes. The corresponding option for `bbcp` is called `"-w"`. In the following example, the desired window size is set to 2 MBytes:

```

% bbcp -P 2 -w 2M localhost:/dev/zero /dev/null
user@localhost's password:
bbcp: Creating /dev/null/zero
bbcp: At 090413 18:12:04 copy 0% complete; 113401.7 KB/s
bbcp: At 090413 18:12:06 copy 0% complete; 113695.8 KB/s
bbcp: At 090413 18:12:08 copy 0% complete; 113981.2 KB/s
..

```

Note, that both machines (operating systems) involved into the transfer should allow setting this parameter above the default value. See the [TCP Tuneup](#) section below for further details on this subject.

The best performance would be achieved by setting optimal values for both TCP window size and a number of simultaneous streams. One would need to experiment with these parameters for a specific network/system setup.

Using `bbcp` to transfer LCLS data

Please use the data mover pool to transfer data to your home institution:

```
psexport.slac.stanford.edu
```

The following example will copy the source file `<srcfilename>` from the LCLS mover node `psexport` to the destination file `<dstfilename>` on the local machine, printing the progress every 15 seconds, using 32 streams and a windows size of 2MB:

```
% bbcp -P 15 -s 32 -w 2M psexport.slac.stanford.edu:<srcfilename> <dstfilename>
```

Troubleshooting

No `bbcp` executable available at a remote site

If the `bbcp` executable isn't available in the executable search path of user 'user' at remote host 'SomeRemoteHost' then this will be reported as follows:

```

% bbcp -P 2 user@SomeRemoteHost:/dev/zero /dev/null
bbcp: Command not found.
bbcp: Unable to allocate more than 0 of 4 data streams.

```

The server's ports are blocked by a firewall

Sometimes a firewall (either at SLAC or at a remote site) may block certain (ranges of) ports. Unfortunately, it's hard to diagnose this problem directly. `bbcp` would report this typically as follows:

```
% bbcp -P 2 user@SomeRemoteHost:/dev/zero /dev/null
bbcp: Accept timed out on port 5031
bbcp: Unable to allocate more than 0 of 4 data streams.
Killed by signal 15.
```

Note, that the actual port number may vary in each specific case because `bbcp` would attempt to dynamically allocate the next available (at the remote server machine) port. A workaround for this problem is to use option `-z` which would use reverse connection protocol (i.e., sink to source):

```
% bbcp -z -P 2 user@SomeRemoteHost:/dev/zero /dev/null
user@SomeRemoteHost's password:
bbcp: Creating /dev/null/zero
bbcp: At 090414 02:59:48 copy 0% complete; 5461.3 KB/s
bbcp: At 090414 02:59:50 copy 0% complete; 13639.7 KB/s
..
```

TCP Tuneup

One can find useful tips on how to tune up TCP parameters on machines involved into the data transfer to get the highest data transfer rate with `bbcp`:

- http://pcbunn.cithep.caltech.edu/bbcp/using_bbcp.htm