Red Hat Enterprise Linux Release 9.2 Manual Pages on 'btt.1' command

$ man btt.1

BTT(1)   BTT(1)

NAME

btt - analyse block i/o traces produces by blktrace

SYNOPSIS

btt

[-a | --seek-absolute]

[-A | --all-data]

[-B <output name> | --dump-blocknos=<output name>]

[-d <seconds> | --range-delta=<seconds>]

[-D <dev;...> | --devices=<dev;...>]

[-e <exe,...> | --exes=<exe,...>]

[-h | --help]

[-i <input name> | --input-file=<input name>]

[-I <output name> | --iostat=<output name>]

[-l <output name> | --d2c-latencies=<output name>]

[-L <freq> | --periodic-latencies=<freq>]

[-m <output name> | --seeks-per-second=<output name>]

[-M <dev map> | --dev-maps=<dev map>

[-o <output name> | --output-file=<output name>]

[-p <output name> | --per-io-dump=<output name>]

[-P <output name> | --per-io-trees=<output name>]

[-q <output name> | --q2c-latencies=<output name>]

[-Q <output name> | --active-queue-depth=<output name>]

```
[ -r               | --no-remaps ]
[ -s <output name> | --seeks=<output name> ]
[ -S <interval>    | --iostat-interval=<interval> ]
[ -t <sec>         | --time-start=<sec> ]
[ -T <sec>         | --time-end=<sec> ]
[ -u <output name> | --unplug-hist=<output name> ]
[ -v               | --verbose ]
[ -V               | --version ]
[ -X               | --easy-parse-avgs ]
[ -z <output name> | --q2d-latencies=<output name> ]
[ -Z               | --do-active ]
```

DESCRIPTION

btt is a post-processing tool for the block layer IO tracing tool called blktrace(8). As noted in its documentation, blktrace is a block layer IO tracing mechanism which provides detailed information about request queue operations up to user space.

btt will take in binary dump data from blkparse, and analyse the events, producing a series of output from the analysis. It will also build .dat files containing "range data" -- showing things like Q ac?tivity (periods of time while Q events are being produced), C activity (likewise for command completions), and etc.

Included with the distribution is a simple 3D plotting utility, bno_plot, which can plot the block numbers btt outputs if the -B option is specified. The display will display each IO generated, with the time (seconds) along the X-axis, the block number (start) along the Y-axis and the number of blocks transferred in the IO represented along the Z-axis.

OPTIONS

-a

--seek-absolute

When specified on the command line, this directs btt to calculate seek distances based solely upon the ending block address of one IO, and the start of the next. By default btt uses the concept of the closeness to either the beginning or end of the previous IO. See the Users Manual for more details about seek distances.

-A

--all-data

Normally btt will not print out verbose information concerning per-process and per-device data. If you desire that level of detail you can specify this option.

-B <output name>

--dump-blocknos=<output name>

This option will output absolute block numbers to three files pre?fixed by the specified output name:

prefix_device_r.dat

All read block numbers are output, first column is time (seconds), second is the block number, and the third column is the ending block number.

prefix_device_w.dat

All write block numbers are output, first column is time (seconds), second is the block number, and the third column is the ending block number.

prefix_device_c.dat

All block numbers (read and write) are output, first column is time (seconds), second is the block number, and the third column is the ending block number.

-d <seconds>

--range-delta=<seconds>

btt outputs a file containing Q and C activity, the notion of ac?tive traces simply means that there are Q or C traces occurring within a certain period of each other. The default values is 0.1 seconds; with this option allowing one to change that granularity. The smaller the value, the more data points provided.

-D <dev;...>

--devices=<dev;...>

Normally, btt will produce data for all devices detected in the traces parsed. With this option, one can reduce the analysis to one or more devices provided in the string passed to this option. The device identifiers are the major and minor number of each device, and each device identifier is separated by a colon (:). A valid specifier for devices 8,0 and 8,8 would then be: 8,0:8,8.

-e <exe,...>

--exes=<exe,...>

The -e option supplies the list of executables that will have I/Os analysed.

-h

--help

Shows a short summary of possible command line option

-i <input name>

--input-file <input file>

Specifies the input file to analyse. This should be a trace file produced by blktrace (8).

-I <output name>

--iostat=<output name>

The -I option directs btt to output iostat-like data to the speci? fied file. Refer to the iostat (sysstat) documentation for details on the data columns.

-l <output name>

--d2c-latencies=<output name>

The -l option allows one to output per-IO D2C latencies respec? tively. The supplied argument provides the basis for the output name for each device.

-L <freq>

--periodic-latencies=<freq>

The -L option allows one to output periodic latency information for both Q2C and D2C latencies. The frequency specified will regulate how often an average latency is output -- a floating point value expressing seconds.

-m <output name>

--seeks-per-second=<output name>

Trigger btt to output seeks-per-second information. The first col?umn will contain a time value (seconds), and the second column will indicate the number of seeks per second at that point.

-M <dev map>

--dev-maps=<dev map>

The -M option takes in a file generated by the provided script (gen_disk_info.py), and allows for better output of device names.

-o <output name>

--output-file=<output name>

Specifies the output file name.

-p <output name>

--per-io-dump=<output name>

The -p option will generate a file that contains a list of all IO "sequences" - showing the parts of each IO (Q, A, I/M, D, & C).

-P <output name>

--per-io-trees=<output name>

The -P option will generate a file that contains a list of all IO "sequences" - showing only the Q, D & C operation times. The D & C time values are separated from the Q time values with a vertical bar.

-q <output name>

--q2c-latencies=<output name>

The -q option allows one to output per-IO Q2C latencies respec?tively. The supplied argument provides the basis for the output name for each device.

-Q <output name>

--active-queue-depth=<output name>

The -Q option allows one to output data files showing the time stamp and the depth of active commands (those issued but not com?pleted).

-r

--no-remaps

Ignore remap traces; older kernels did not implement the full remap PDU.

-s <output name>

--seeks=<output name>

The -s option instructs btt to output seek data, the argument pro? vided is the basis for file names output. There are two files per device, read seeks and write seeks.

-S <interval>

--iostat-interval=<interval>

The -S option specifies the interval to use between data output, it defaults to once per second.

-t <sec>

--time-start=<sec>

-T <sec>

--time-end=<sec>

The -t/-T options allow one to set a start and/or end time for analysing - analysing will only be done for traces after -t's argu? ment and before -T's argument. (-t and -T are optional, so if you specify just -t, analysis will occur for all traces after the time specified. Similarly, if only -T is specified, analysis stops after -T's seconds.)

-u <output name>

--unplug-hist=<output name>

This option instructs btt to generate a data file containing his? togram information for unplug traces on a per device basis. It shows how many times an unplug was hit with a specified number of IOs released. There are 21 output values into the file, as follows:

a value of 0 represents 0..4 counts

a value of 1 represents 5..9 counts

a value of 2 represents 10..14 counts

etc, until

a value of 20 represents 100+ counts

The file name(s) generated use the text string passed as an argu?

ment for the prefix, followed by the device identifier in major,mi?nor form, with a .dat extension. For example, with -u up_hist specified on the command line: up_hist_008,032.dat.

-V

--version

Shows the version of btt.

-v

--verbose

Requests a more verbose output.

-X

--easy-parse-avgs

Provide data in an easy-to-parse form and write it to a file with .avg exentsion

-z <output name>

--q2d-latencies=<output name>

The -z option allows one to output per-IO Q2D latencies respec?tively. The supplied argument provides the basis for the output name for each device.

-Z

--do-active

The -Z will output files containing data which can be plotted show?ing per-device (and total system) I/O activity.

AUTHORS

btt was written by Alan D. Brunelle. This man page was created from the blktrace documentation by Bas Zoetekouw.

REPORTING BUGS

Report bugs to <linux-btrace@vger.kernel.org>

COPYRIGHT

Copyright ? 2006 Jens Axboe, Alan D. Brunelle and Nathan Scott.

This is free software. You may redistribute copies of it under the terms of the GNU General Public License <http://www.gnu.org/li?censes/gpl.html>. There is NO WARRANTY, to the extent permitted by law.

This manual page was created for Debian by Bas Zoetekouw. It was derived from the documentation provided by the authors and it may be used, distributed and modified under the terms of the GNU General Public License, version 2.

On Debian systems, the text of the GNU General Public License can be found in /usr/share/common-licenses/GPL-2.

SEE ALSO

The btt Users Guide, which can be found in /usr/share/doc/blktrace/btt.pdf

bno_plot (1), blktrace (8), blkparse (1), verify_blkparse (1), blkrawverify (1), btt (1)

blktrace git-20070910192508   September 29, 2007   BTT(1)