Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'btrecord.8' command

$ man btrecord.8

BTRECORD(8) BTRECORD(8)

NAME

btrecord - recreate IO loads recorded by blktrace

SYNOPSIS

Usage:

btrecord [options] <dev...>

DESCRIPTION

The btrecord and btreplay tools provide the ability to record and re? play IOs captured by the blktrace utility. Attempts are made to main? tain ordering, CPU mappings and time-separation of IOs.

The blktrace utility provides the ability to collect detailed traces from the kernel for each IO processed by the block IO layer. The traces provide a complete timeline for each IO processed, including detailed information concerning when an IO was first received by the block IO layer ? indicating the device, CPU number, time stamp, IO direction, sector number and IO size (number of sectors). Using this information, one is able to replay the IO again on the same machine or another set up entirely.

The basic operating work-flow to replay IOs would be something like:

-

Run blktrace to collect traces. Here you specify the device or devices that you wish to trace and later replay IOs upon.

Note:

the only traces you are interested in are QUEUE requests ?

thus, to save system resources (including storage for traces), one could

specify the -a queue command line option to blktrace.

-

While blktrace is running, you run the workload that you

are interested in.

-

When the work load has completed, you stop the blktrace

utility (thus saving all traces over the complete workload).

-

You extract the pertinent IO information from the traces saved by

blktrace using the btrecord utility. This will parse

each trace file created by blktrace, and crafty IO descriptions

to be used in the next phase of the workload processing.

-

Once btrecord has successfully created a series of data

files to be processed, you can run the btreplay utility which

attempts to generate the same IOs seen during the sample workload phase.

OPTIONS

-d <dir>

--input-directory=<dir>

Set input directory. This option requires a single parameter providing the directory name for where input files are to be found. The default directory is the current directory (.).

-D <dir>

--output-directory=<dir>

Set output directory. This option requires a single parameter providing the directory name for where output files are to be found. The default directory is the current directory (.).

-F

--find-traces

Find trace files automatically This option instructs btreplay to go find all the trace files in the directory specified (either via the -d option, or in the default directory (.).

-h

--help

Show help and exit.

-V

--version

Show version number and exit.

-m <nanoseconds>

--max-bunch-time=<nanoseconds>

The -m option requires a single parameter which specifies an amount of time (in nanoseconds) to include in any one bunch of IOs that are to be processed. The smaller the value, the smaller the number of IOs processed at one time ? perhaps yielding in more realistic replay. However, after a certain point the amount of overhead per bunch may result in additional real re? play time, thus yielding less accurate replay times.

The default value is 10,000,000 nanoseconds (10 milliseconds).

-M <num>

--max-pkts=<num>

Set maximum number of packets per bunch. The -M option requires a single parameter which specifies the maximum number of IOs to store in a single bunch. As with the -m option, smaller values may or may not yield more accurate replay times.

The default value is 8, with a maximum value of up to 512 being supported.

-o <basename>

--output-base=<basename>

Set base name for output files. Each output file has 3 fields:

1.

Device identifier (taken directly from the device name of the

blktrace output file).

2.

btrecord base name ? by default ``replay''.

3.

The CPU number (again, taken directly from the

blktrace output file name).

This option requires a single parameter that will override the

default name (replay), and replace it with the specified value.

-v

--verbose

Enable verbose output. This option will output some simple sta?

tistics at the end of a successful run. Example output is:

sdab:0: 580661 pkts (tot), 126030 pkts (replay), 89809 bunches, 1.4 pkts/bunch

sdab:1: 2559775 pkts (tot), 430172 pkts (replay), 293029 bunches, 1.5 pkts/bunch

sdab:2: 653559 pkts (tot), 136522 pkts (replay), 102288 bunches, 1.3 pkts/bunch

sdab:3: 474773 pkts (tot), 117849 pkts (replay), 69572 bunches, 1.7 pkts/bunch

The meaning of the columns is:

1.

The first field contains the device name and CPU identi?

fier. Thus:

sdab:0: means the device sdab and traces on CPU 0.

2.

The second field contains the total number of packets pro?

cessed for each

device file.

3.

The next field shows the number of packets eligible for re?

play.

4.

The fourth field contains the total number of IO bunches.

5.

The last field shows the average number of IOs per bunch

recorded.

AUTHORS

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

REPORTING BUGS

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

COPYRIGHT

SEE ALSO

The full documentation for btreplay can be found in /usr/share/doc/blk?trace on Debian systems.

blktrace (8), blkparse (1), btreplay (8)