Working flow of record

The main working part of command 'perf record' is defined in the function \_\_cmd\_record in builtin-record.c. This function is defined in these form:

static int \_\_cmd\_record(struct record \*rec, int argc, char \*\*argv)

The working flow of this function starts with the creation of a perf session.

    session = perf\_session\_\_new(file, false, NULL);

Then, if the session is successfully created, perf would firstly synthesize the attributes. This makes the attrs into the events, allowing them to be streamed over the pipe along the rest of the header data. It would also support the adding and removing of the events from perf sessions dynamically. This would call the function of perf\_event\_\_synthesize\_attr() and perf\_event\_\_synthesize\_attrs() in /util/header.c.

If the process has tracepoints, then the function of perf\_event\_\_synthesize\_tracing\_data() would be called. This function is defined also in header.c. It firstly writes the tracing data to a temp file, then writes the data size to pipe. Finally the tracing data from the temp file will be written to the pipe and to the file of perf.data.

After this, perf also has a function synthesizing mmap. This function is called because perf record initializes the recording using perf\_events interface of the system. The records are then written into  mmap pages and a Linux signal is sent to perf record if a page is full. Perf record then stores the records into the data file.

The structure of perf samples is defined in /util/event.h. After synthesizing kernel mmap and threads, the perf tool starts to collect samples. For each sample there is a header set by perf tool, and file size of perf samples could also be calculated.