
Subject: [PATCH 3/7] perf: add ability to change event according to sample (v2)
Posted by [Andrew Vagin](#) on Mon, 28 Nov 2011 09:03:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

It's opposition of perf_session__parse_sample.

v2: fixed mistakes which David Arhen found

Signed-off-by: Andrew Vagin <avagin@openvz.org>

```
tools/perf/util/event.h | 2 +
tools/perf/util/evsel.c | 74 +++++
tools/perf/util/session.h | 9 +++++
3 files changed, 85 insertions(+), 0 deletions(-)
```

```
diff --git a/tools/perf/util/event.h b/tools/perf/util/event.h
```

```
index 357a85b..0493101 100644
```

```
--- a/tools/perf/util/event.h
```

```
+++ b/tools/perf/util/event.h
```

```
@@ -187,5 +187,7 @@ const char *perf_event__name(unsigned int id);
```

```
int perf_event__parse_sample(const union perf_event *event, u64 type,
    int sample_size, bool sample_id_all,
    struct perf_sample *sample, bool swapped);
```

```
+int perf_event__change_sample(union perf_event *event, u64 type,
```

```
+    const struct perf_sample *data, bool swapped);
```

```
#endif /* __PERF_RECORD_H */
```

```
diff --git a/tools/perf/util/evsel.c b/tools/perf/util/evsel.c
```

```
index e426264..d697568 100644
```

```
--- a/tools/perf/util/evsel.c
```

```
+++ b/tools/perf/util/evsel.c
```

```
@@ -494,3 +494,77 @@ int perf_event__parse_sample(const union perf_event *event, u64 type,
```

```
    return 0;
```

```
}
```

```
+
```

```
+int perf_event__change_sample(union perf_event *event, u64 type,
```

```
+    const struct perf_sample *data, bool swapped)
```

```
+{
```

```
+    u64 *array;
```

```
+
```

```
+ /*
```

```
+  * used for cross-endian analysis. See git commit 65014ab3
```

```
+  * for why this goofiness is needed.
```

```
+ */
```

```
+ union {
```

```
+    u64 val64;
```

```
+    u32 val32[2];
```

```

+ } u;
+
+ array = event->sample.array;
+
+ if (type & PERF_SAMPLE_IP) {
+ event->ip.ip = data->ip;
+ array++;
+ }
+
+ if (type & PERF_SAMPLE_TID) {
+ u.val32[0] = data->pid;
+ u.val32[1] = data->tid;
+ if (swapped) {
+ /* undo swap of u64, then swap on individual u32s */
+ u.val32[0] = bswap_32(u.val32[0]);
+ u.val32[1] = bswap_32(u.val32[1]);
+ u.val64 = bswap_64(u.val64);
+ }
+
+ *array = u.val64;
+ array++;
+ }
+
+ if (type & PERF_SAMPLE_TIME) {
+ *array = data->time;
+ array++;
+ }
+
+ if (type & PERF_SAMPLE_ADDR) {
+ *array = data->addr;
+ array++;
+ }
+
+ if (type & PERF_SAMPLE_ID) {
+ *array = data->id;
+ array++;
+ }
+
+ if (type & PERF_SAMPLE_STREAM_ID) {
+ *array = data->stream_id;
+ array++;
+ }
+
+ if (type & PERF_SAMPLE_CPU) {
+ u.val32[0] = data->cpu;
+ if (swapped) {
+ /* undo swap of u64, then swap on individual u32s */
+ u.val32[0] = bswap_32(u.val32[0]);

```

```

+ u.val64 = bswap_64(u.val64);
+ }
+ *array = u.val64;
+ array++;
+ }
+
+ if (type & PERF_SAMPLE_PERIOD) {
+ *array = data->period;
+ array++;
+ }
+
+ return 0;
+}
diff --git a/tools/perf/util/session.h b/tools/perf/util/session.h
index 6e393c9..444f121 100644
--- a/tools/perf/util/session.h
+++ b/tools/perf/util/session.h
@@ -167,6 +167,15 @@ static inline int perf_session__parse_sample(struct perf_session
*session,
    session->header.needs_swap);
}

+static inline int perf_session__change_sample(struct perf_session *session,
+    union perf_event *event,
+    const struct perf_sample *sample)
+{
+ return perf_event__change_sample(event, session->sample_type,
+    sample,
+    session->header.needs_swap);
+}
+
+ struct perf_evsel *perf_session__find_first_evtype(struct perf_session *session,
+    unsigned int type);

--
1.7.1

```
