Subject: [PATCH 2/6] perf: add ability to change event according to sample (v3)
Posted by Andrey Vagin on Wed, 07 Dec 2011 13:55:57 GMT
View Forum Message <> Reply to Message

It's opposition of perf_session__parse_sample.

v2: fixed mistakes which David Arhen found
v3: s/data/sample/
    s/perf_event__change_sample/perf_event__synthesize_sample

Reviewed-by: David Ahern <dsahern@gmail.com>
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 0d80201..cc3cc11 100644
--- a/tools/perf/util/event.h
+++ b/tools/perf/util/event.h
@@ -199,6 +199,8 @@ 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__synthesize_sample(union perf_event *event, u64 type,
+        const struct perf_sample *sample, bool swapped);

 size_t perf_event__fprintf_comm(union perf_event *event, FILE *fp);
 size_t perf_event__fprintf_mmap(union perf_event *event, FILE *fp);
diff --git a/tools/perf/util/evsel.c b/tools/perf/util/evsel.c
index ee68d69..e2d1b22 100644
--- a/tools/perf/util/evsel.c
+++ b/tools/perf/util/evsel.c
@@ -574,3 +574,77 @@ int perf_event__parse_sample(const union perf_event *event, u64 type,

  return 0;
 }
+
+int perf_event__synthesize_sample(union perf_event *event, u64 type,
+        const struct perf_sample *sample, 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 = sample->ip;
+  array++;
+ }
+
+ if (type & PERF_SAMPLE_TID) {
+  u.val32[0] = sample->pid;
+  u.val32[1] = sample->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 = sample->time;
+  array++;
+ }
+
+ if (type & PERF_SAMPLE_ADDR) {
+  *array = sample->addr;
+  array++;
+ }
+
+ if (type & PERF_SAMPLE_ID) {
+  *array = sample->id;
+  array++;
+ }
+
+ if (type & PERF_SAMPLE_STREAM_ID) {
+  *array = sample->stream_id;
+  array++;
+ }
+
+ if (type & PERF_SAMPLE_CPU) {
```

```
+  u.val32[0] = sample->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 = sample->period;
+  array++;
+ }
+
+ return 0;
+}
diff --git a/tools/perf/util/session.h b/tools/perf/util/session.h
index 30e9c6b..4ed73d3 100644
--- a/tools/perf/util/session.h
+++ b/tools/perf/util/session.h
@@ -134,6 +134,15 @@ static inline int perf_session__parse_sample(struct perf_session
*session,
      session->header.needs_swap);
 }

+static inline int perf_session__synthesize_sample(struct perf_session *session,
+          union perf_event *event,
+          const struct perf_sample *sample)
+{
+ return perf_event__synthesize_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
```