Sharing PMU counters across compatible perf events

Song Liu, David Carrillo-Cisneros

More perf events than PMU counters

- Count same metric within different scopes
  - Per CPU, per task, per (nested) cgroup

- Time multiplexing (`perf_rotate_context`)

```
$ perf stat -e ref-cycles,ref-cycles -a -I 1000
# time counts unit events
1.006554343 3,950,838,584 ref-cycles (74.40%)
1.006554343 3,783,373,129 ref-cycles (25.69%)
```

Sharing PMUs

- Compatible perf events can share one counter
  - Compatible: counting exact same metric

- Avoid/Reduce time multiplexing
- Reduce time spent in configuring PMUs

Proposal #1: in core code

(+) One solution for all

(+) Detect compatible events at `perf_event_open`

No overhead for context switch and rotation

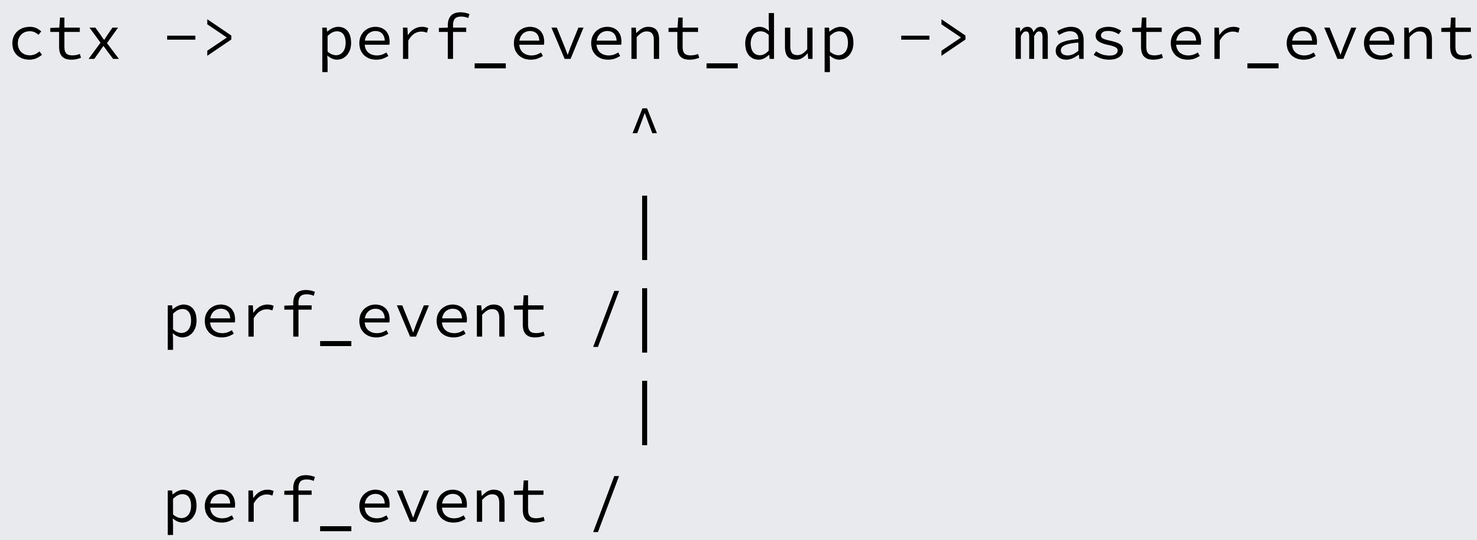
(−) Sharing within same `perf_event_context`

No sharing across CPU events and task events

(−) Complexity

Song Liu: https://lore.kernel.org/lkml/20190226230623.3910393-2-songliubraving@fb.com/

Proposal #1: in core code



Song Liu: https://lore.kernel.org/lkml/20190226230623.3910393-2-songliubraving@fb.com/

Proposal #2: in arch/pmu code

(+) Simpler: Less LoC per arch/pmu

(+) Sharing across CPU events and task events

(−) Detect compatible events in context switch and rotation

Jiri Olsa: https://lore.kernel.org/lkml/20171206114204.GB10580@krava/

Proposal #2: in arch/pmu code

```
XXX_pmu_add() {
   if (find_compatible_event())
       goto out;

   /* existing pmu_add() logic */
}
```

Jiri Olsa: https://lore.kernel.org/lkml/20171206114204.GB10580@krava/

Other ideas?