# perf: screwups

jiri olsa

# SCREWPUS

- threads
- delayed user space data retrievals
- watch command
- cputime PMU
- RDT support
- event triggers
- group sharing
- bpf command
- build menuconfig

# THREADS

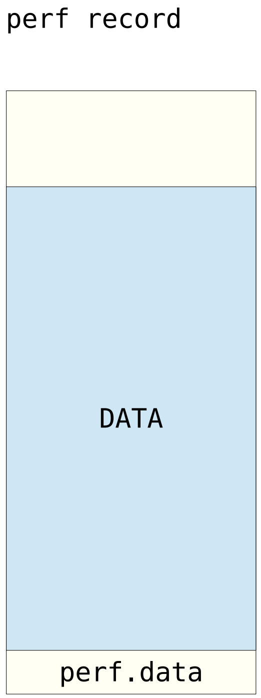
- record
- report
- other report tools (c2c, sched, mem..)

# THREADS REPORT

- original work from Namhyung Kim
- ~30% speed up
- ended up as RFC only

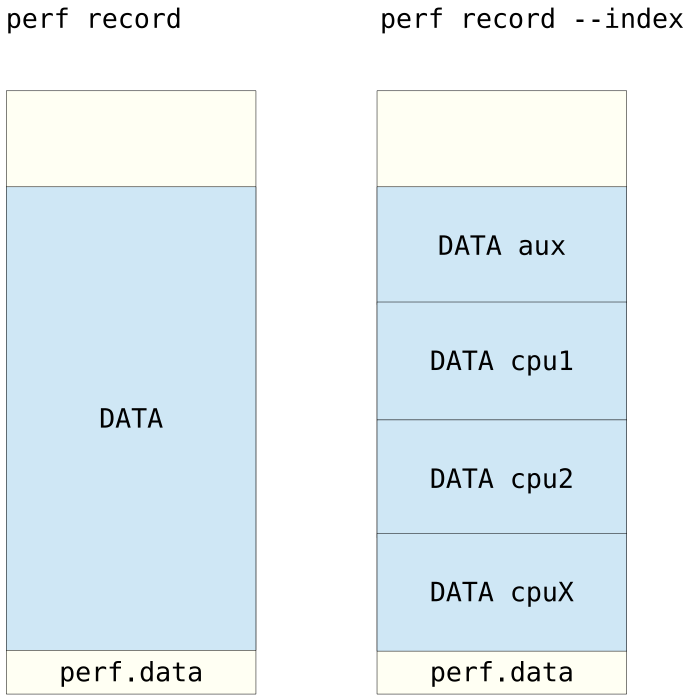
# THREADS REPORT

- original work from Namhyung Kim
- ~30% speed up
- ended up as RFC only



# THREADS REPORT

- original work from Namhyung Kim
- ~30% speed up
- ended up as RFC only



perf record

DATA

perf.data

perf record --index

DATA aux

DATA cpu1

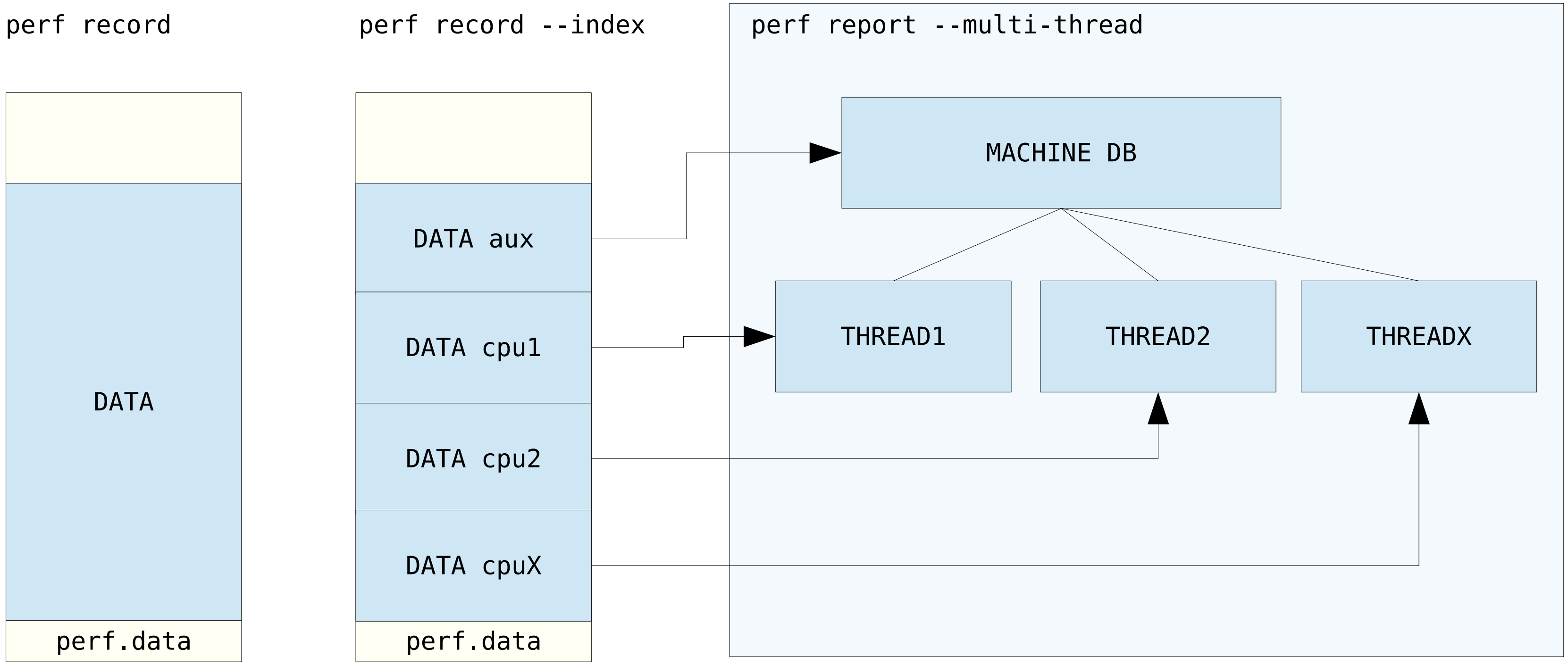
DATA cpu2

DATA cpuX

perf.data

# THREADS REPORT

- original work from Namhyung Kim
- ~30% speed up
- ended up as RFC only



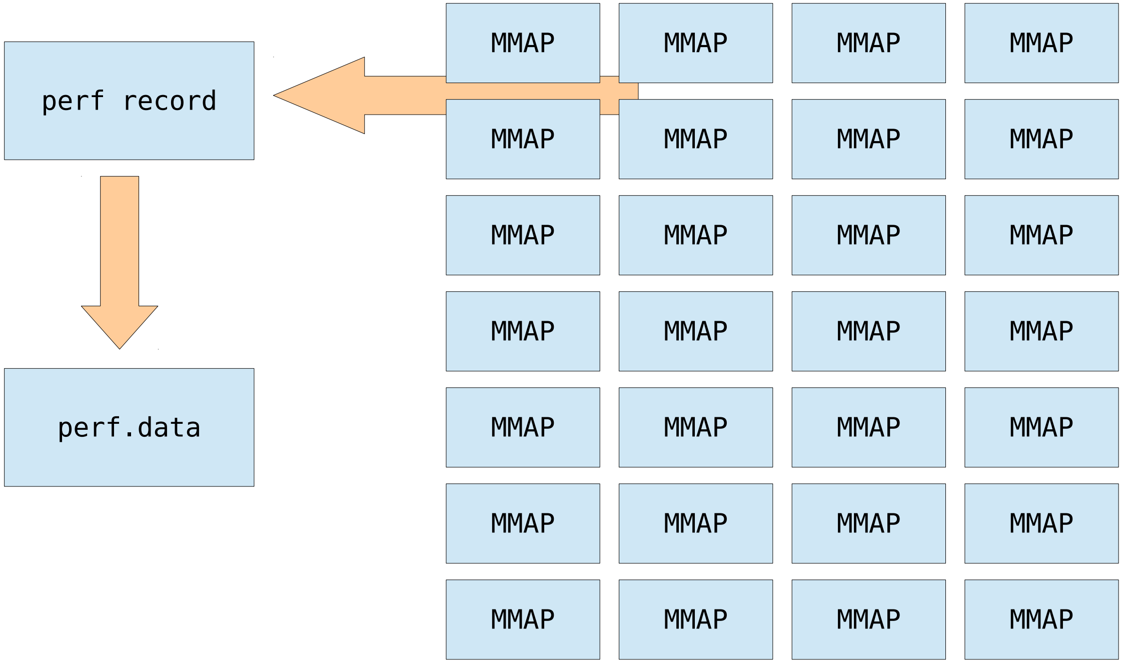
# THREADS REPORT

- RFC:

  https://lore.kernel.org/lkml/1443763159-29098-1-git-send-email-namhyung@kernel.org/

# THREADS RECORD

- one thread to store all CPUs data
- 1 CPU – 1 MMAP



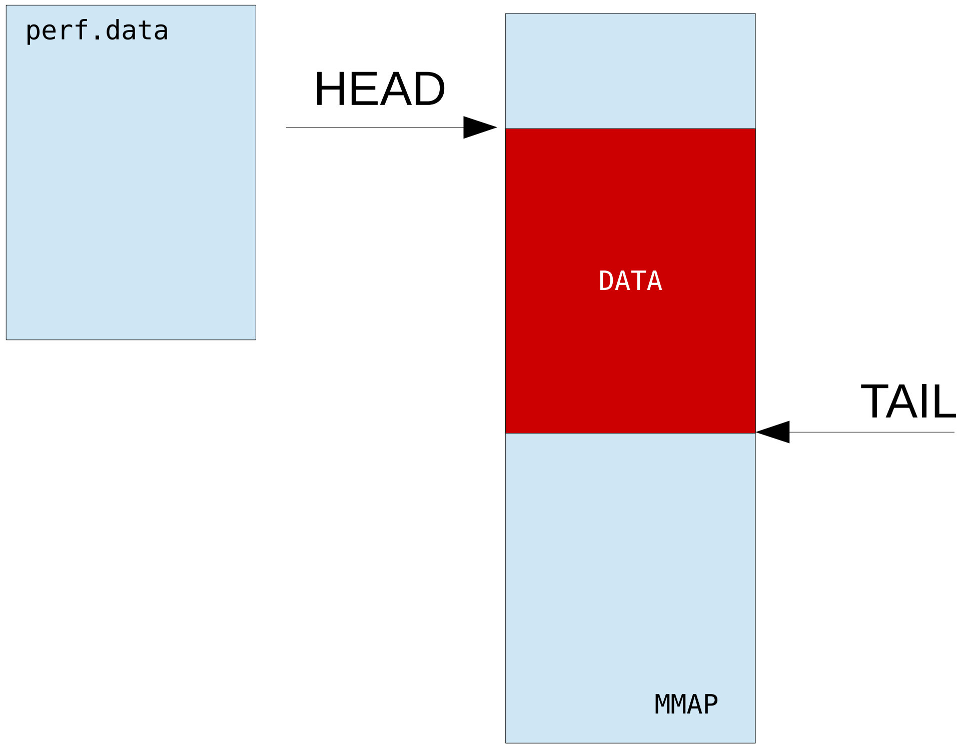
perf record

perf.data

| | | | |
|---|---|---|---|
| MMAP | MMAP | MMAP | MMAP |
| MMAP | MMAP | MMAP | MMAP |
| MMAP | MMAP | MMAP | MMAP |
| MMAP | MMAP | MMAP | MMAP |
| MMAP | MMAP | MMAP | MMAP |
| MMAP | MMAP | MMAP | MMAP |
| MMAP | MMAP | MMAP | MMAP |

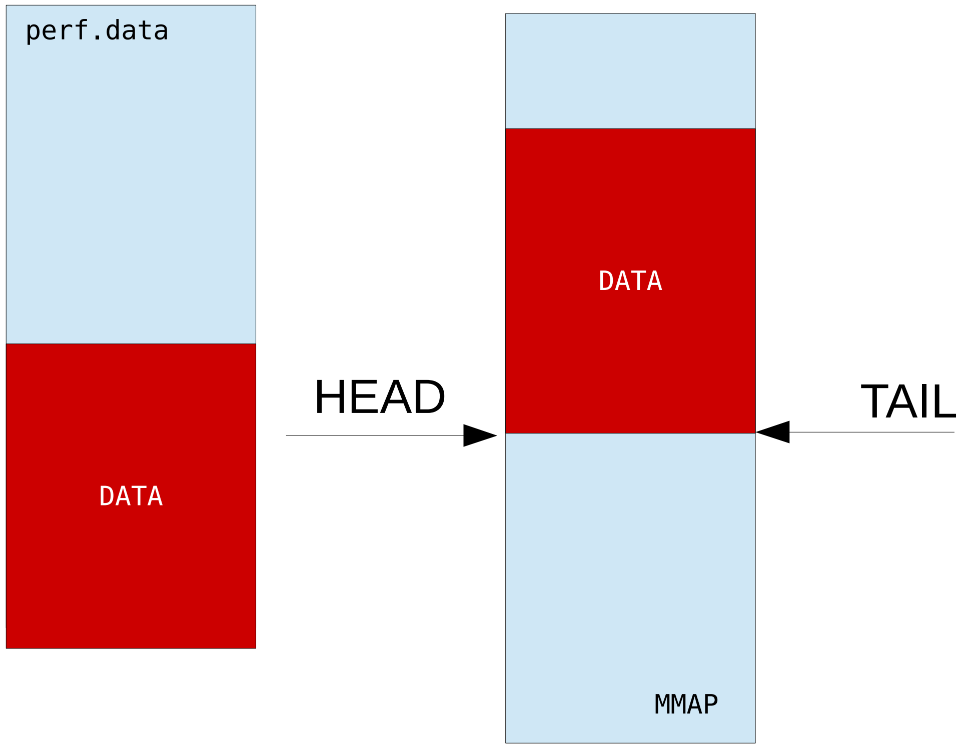
# THREADS RECORD

- one thread to store all CPUs data
- 1 CPU – 1 MMAP



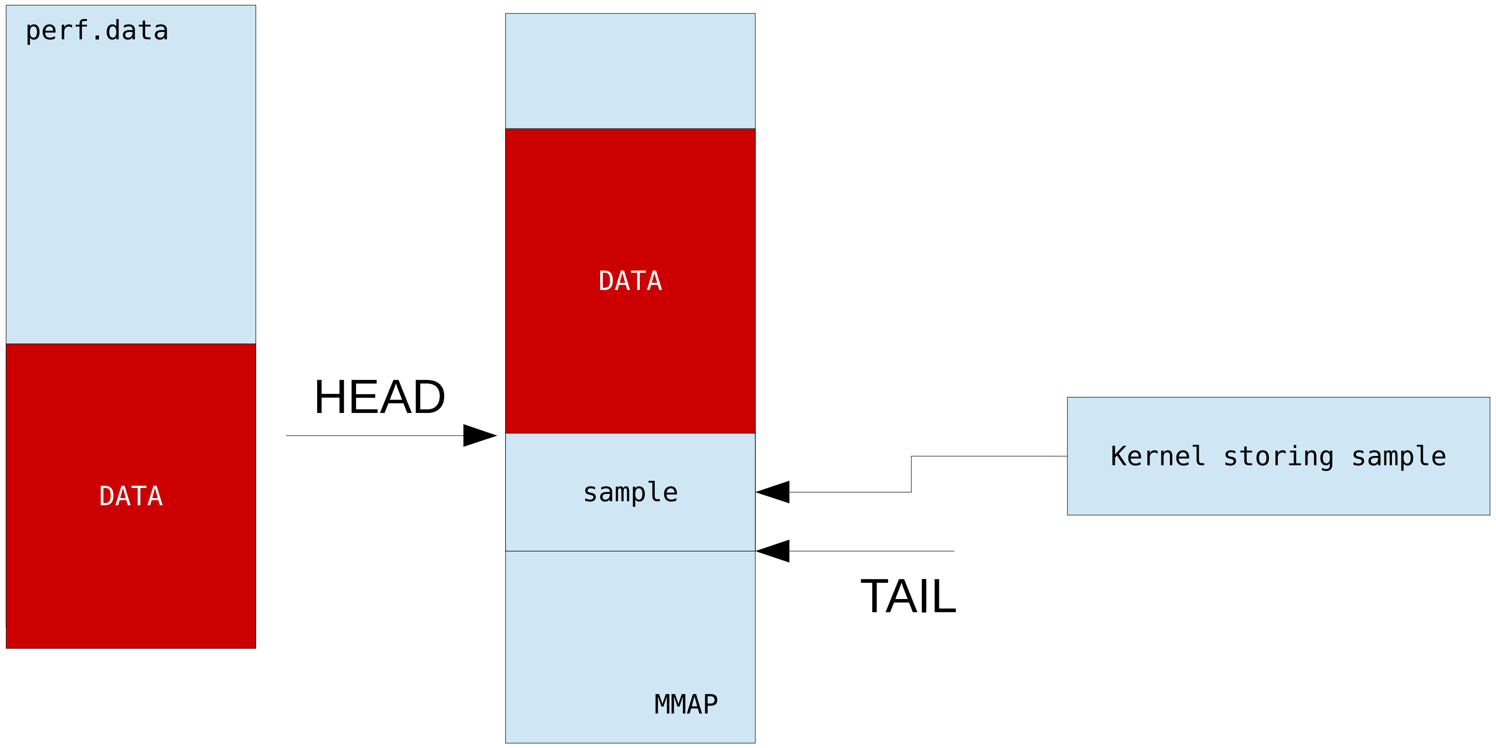
# THREADS RECORD

- one thread to store all CPUs data
- 1 CPU – 1 MMAP



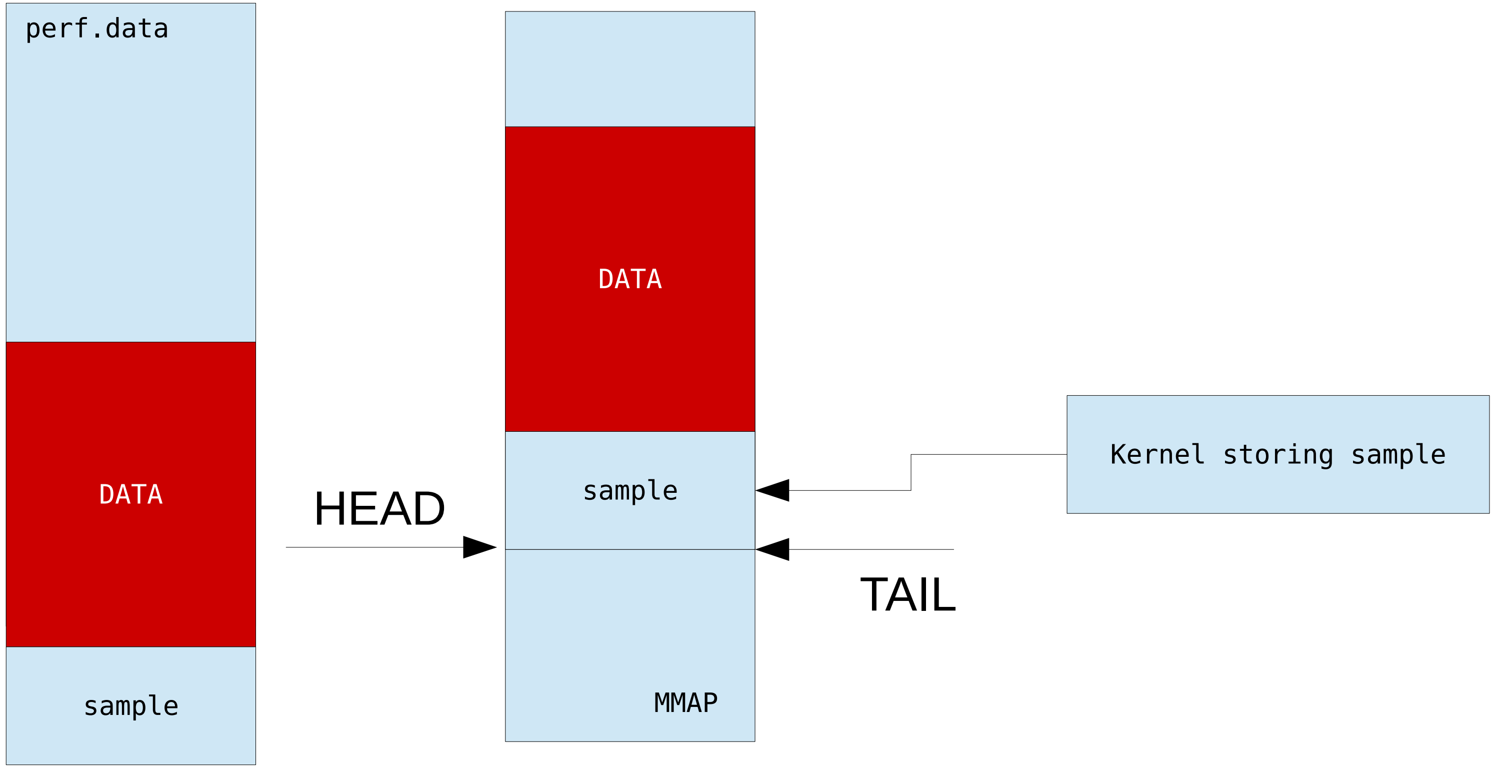
# THREADS RECORD

- one thread to store all CPUs data
- 1 CPU – 1 MMAP



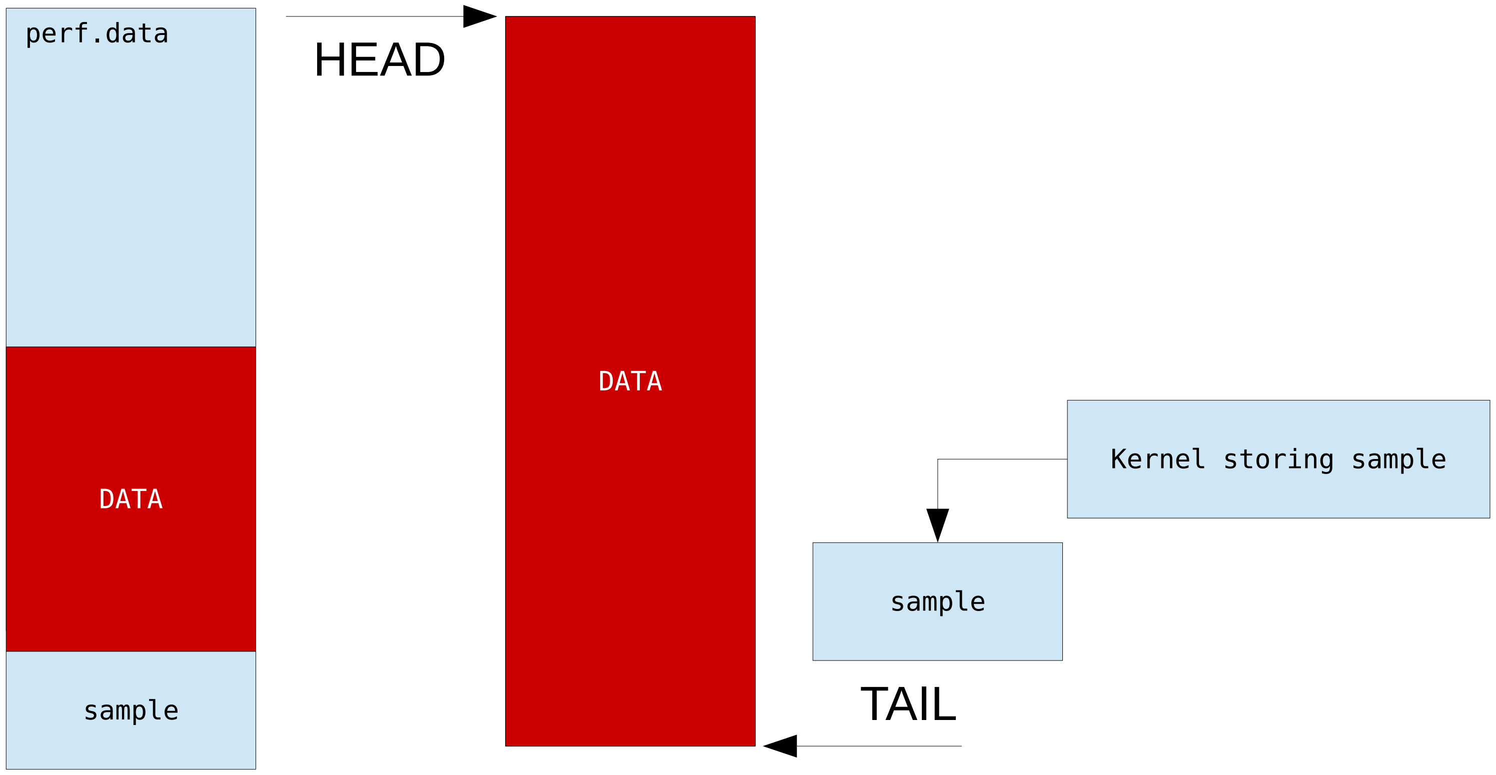
# THREADS RECORD

- one thread to store all CPUs data
- 1 CPU – 1 MMAP



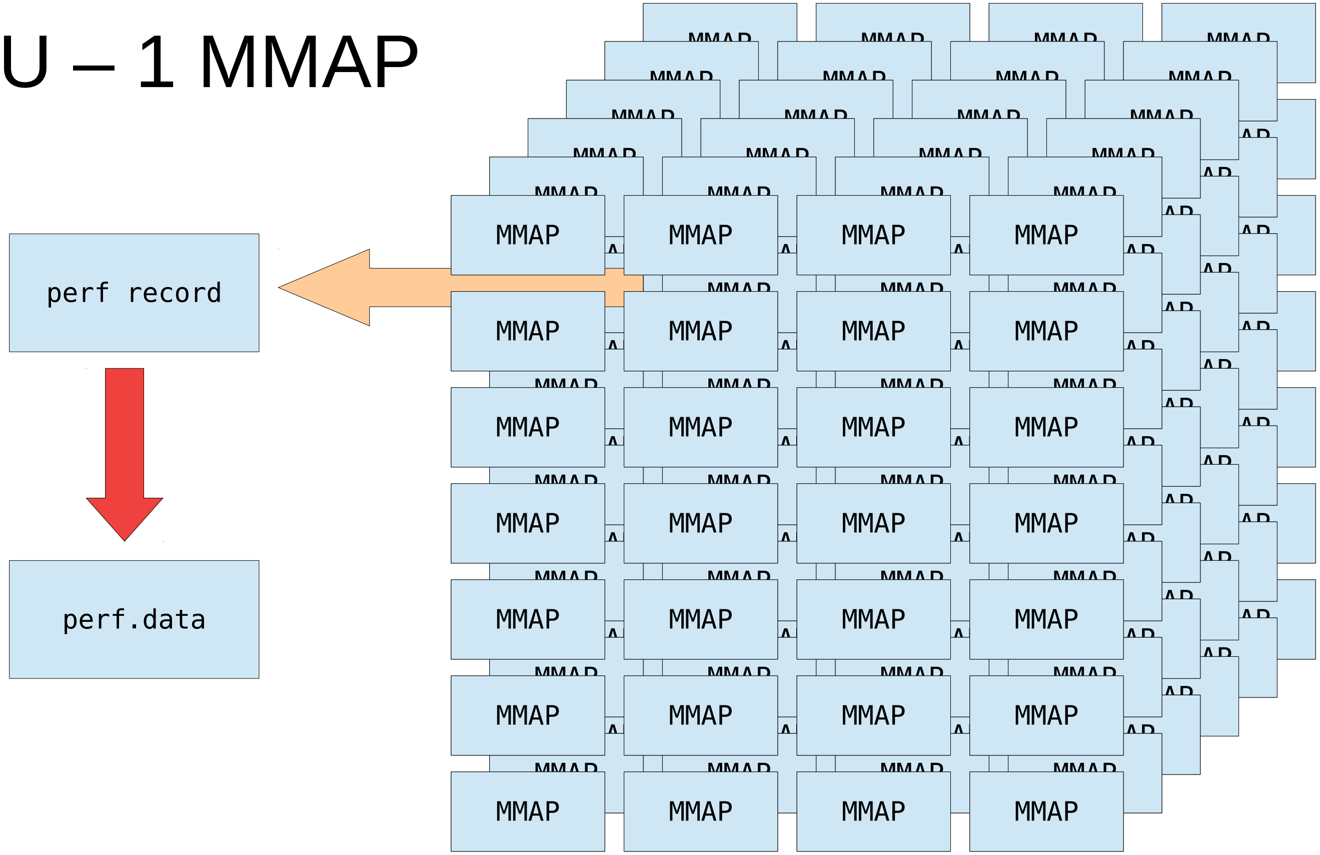
# THREADS RECORD

- one thread to store all CPUs data
- 1 CPU – 1 MMAP



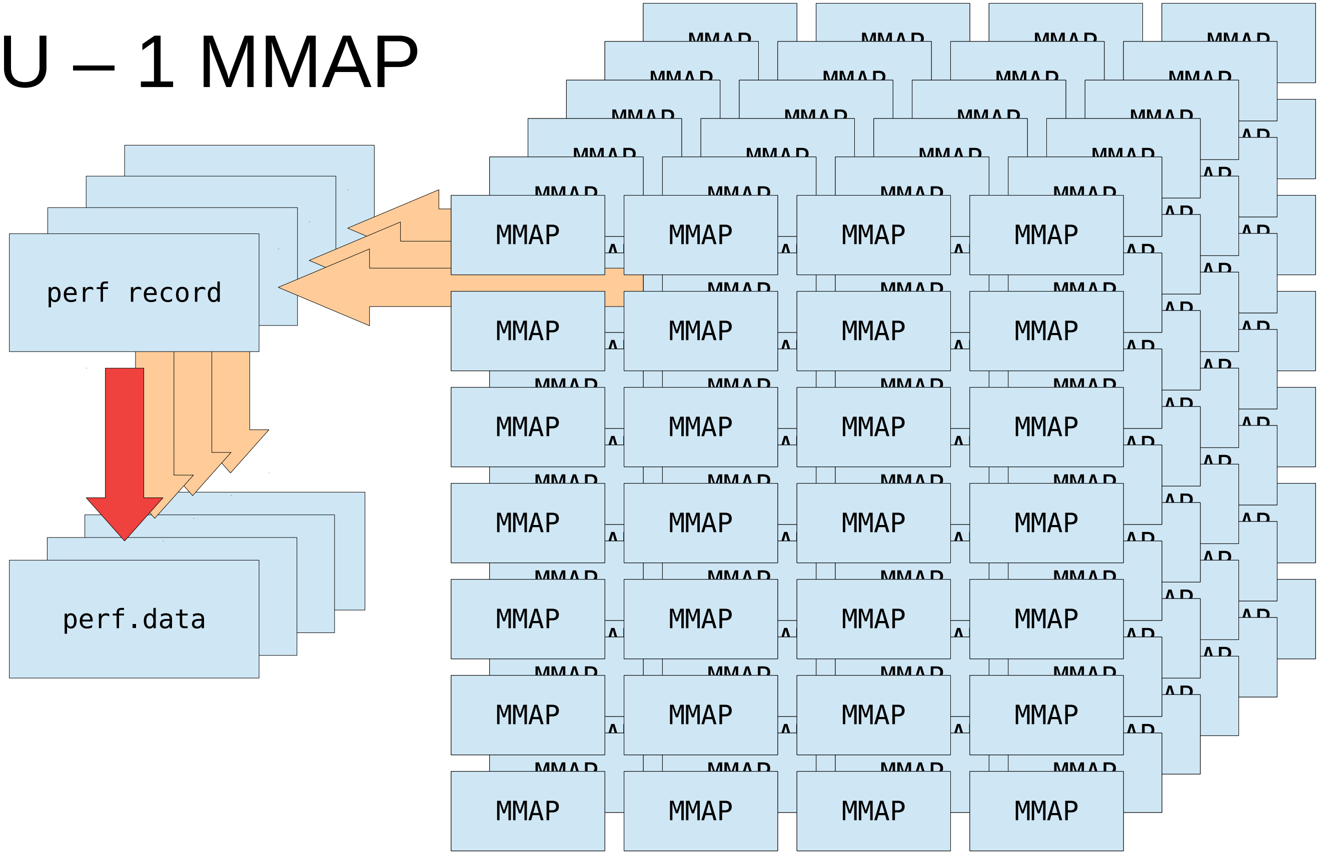
# THREADS RECORD

- one thread to store all CPUs data
- 1 CPU – 1 MMAP



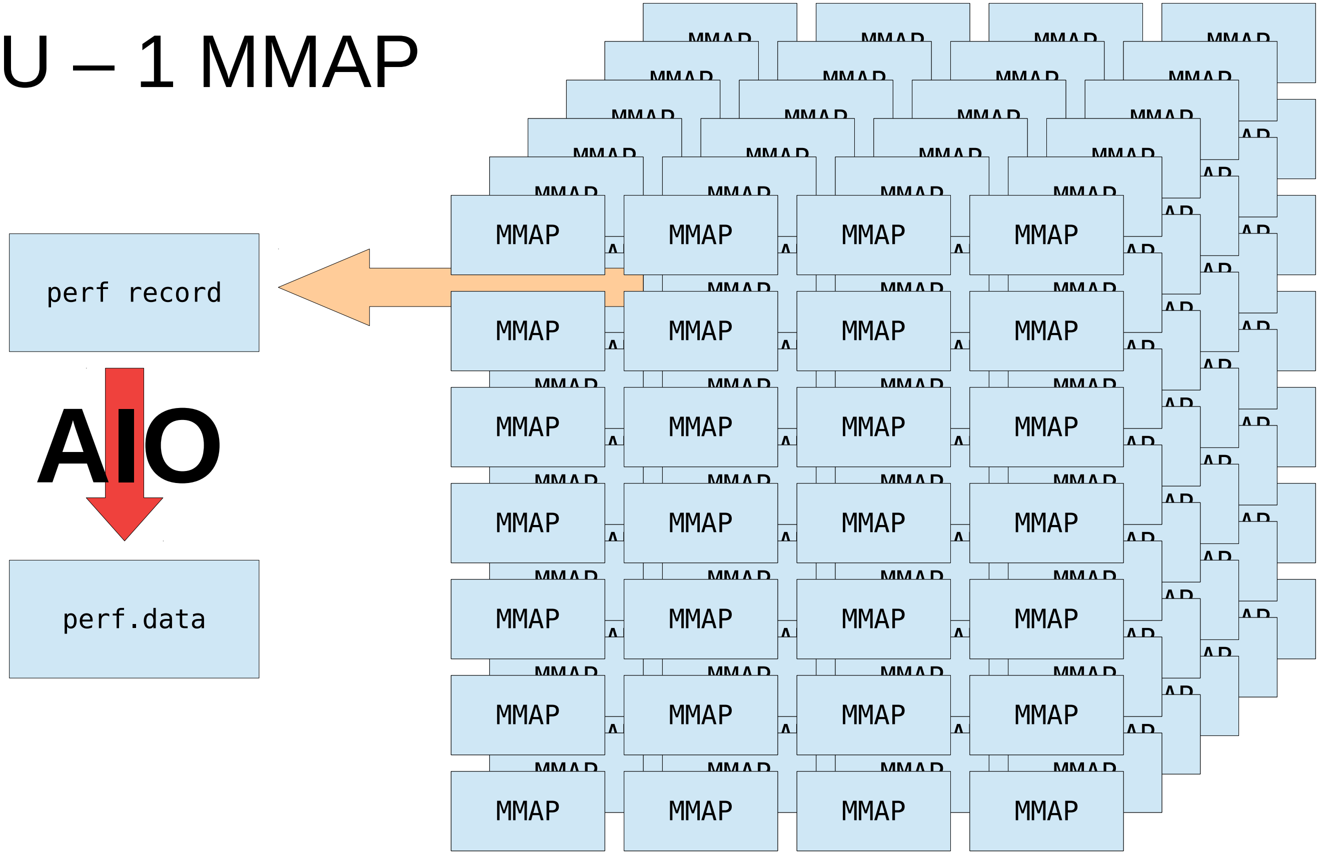
# THREADS RECORD

- one thread to store all CPUs data
- 1 CPU – 1 MMAP



# THREADS RECORD

- one thread to store all CPUs data
- 1 CPU – 1 MMAP



# THREADS RECORD

- **perf record --aio** (Alexey Budankov)

  asynchronous trace streaming via Posix AIO API

  merged in soon
- **perf record --threads**

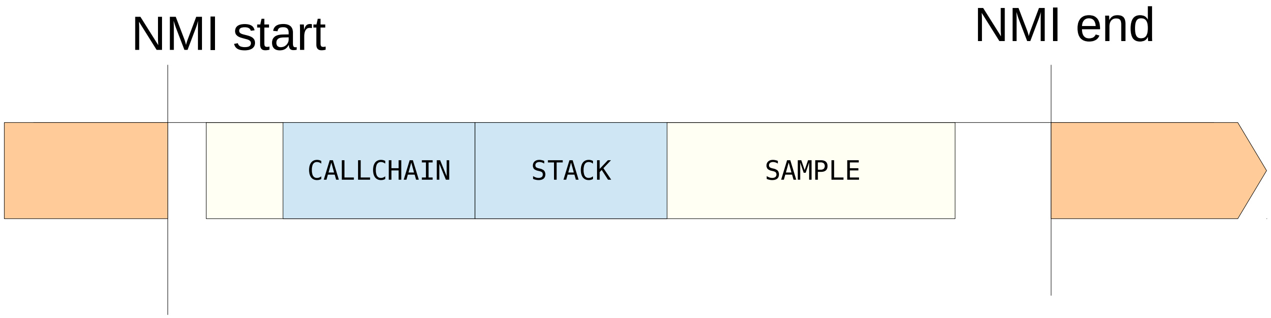
  separate threads

  data in separate files

# DELAYED USER  SPACE DATA RETRIEVAL

- user space data processing out of NMI

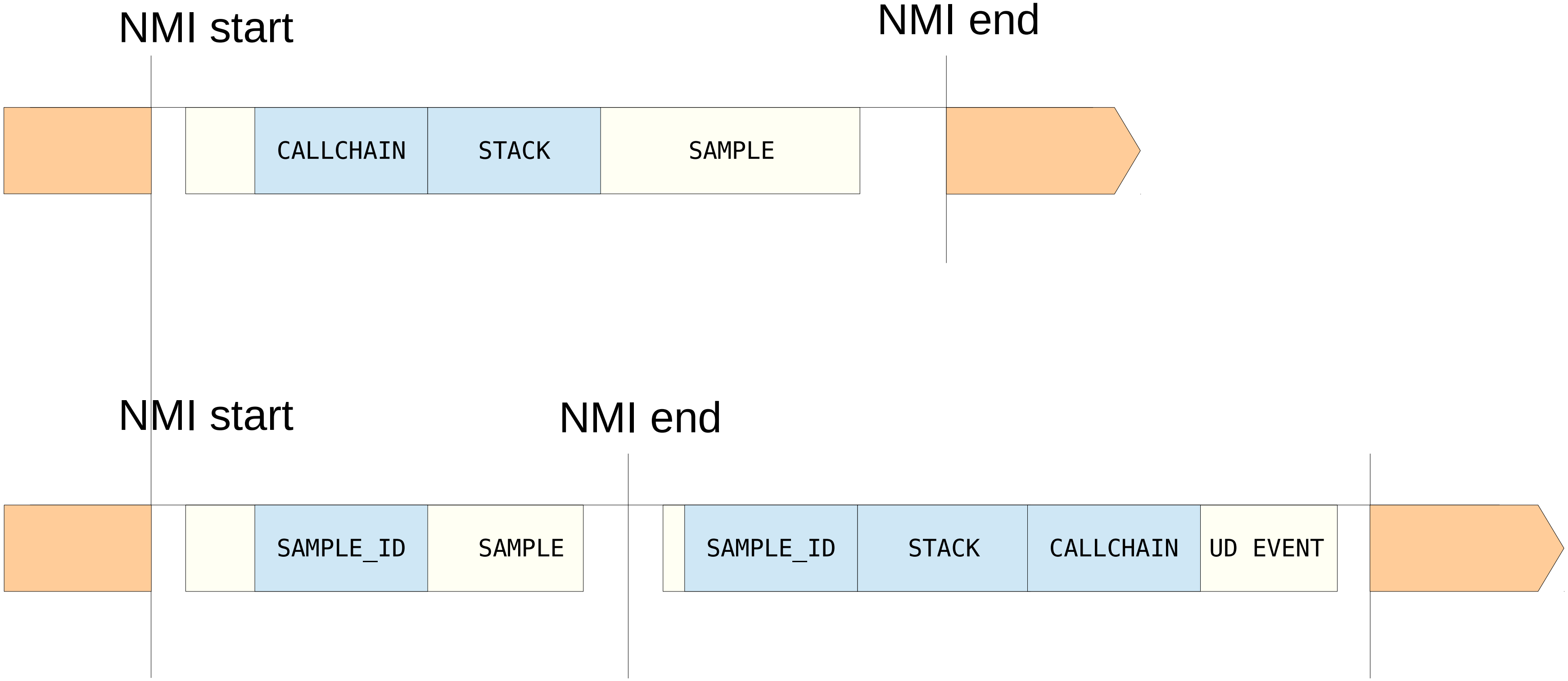
  callchains, stack dump



# DELAYED USER  SPACE DATA RETRIEVAL

- user space data processing out of NMI

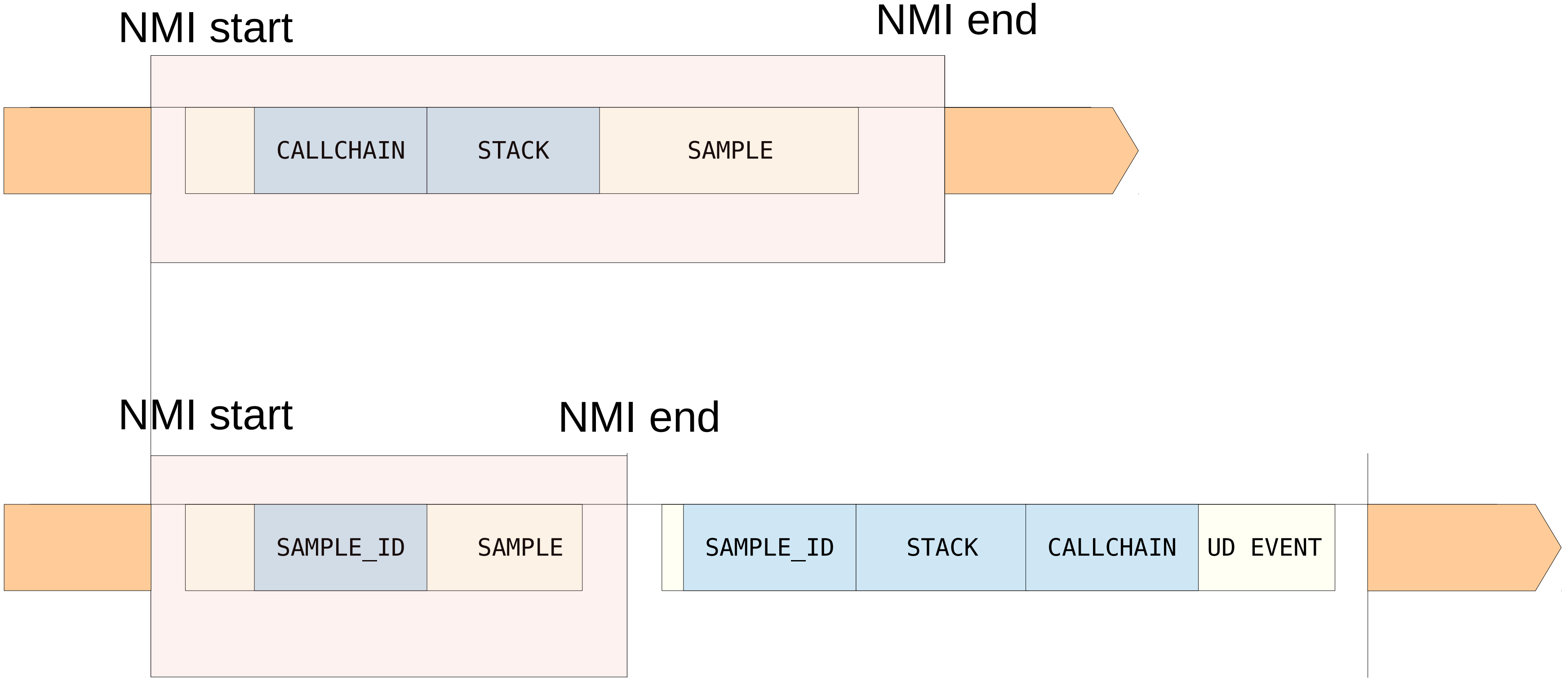
  callchains, stack dump



# DELAYED USER SPACE DATA RETRIEVAL

- user space data processing out of NMI

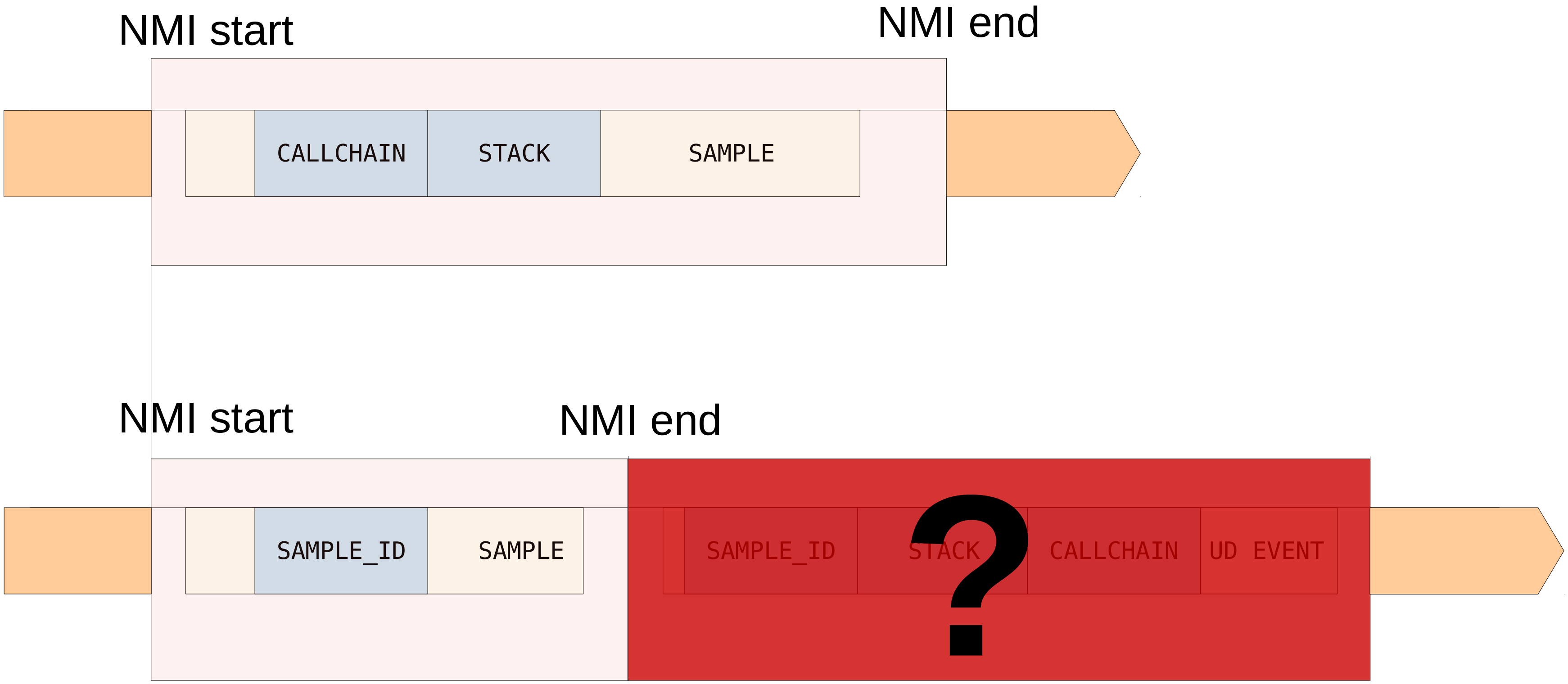
  callchains, stack dump



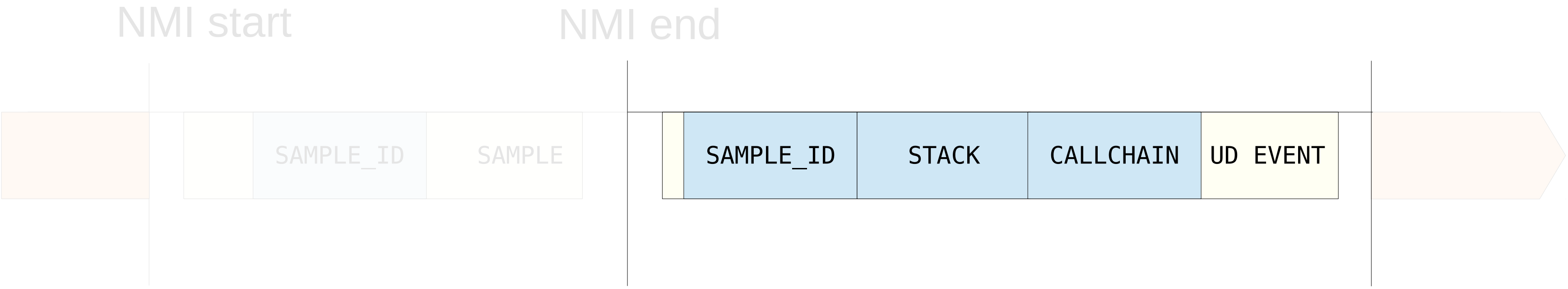
# DELAYED USER SPACE DATA RETRIEVAL

- user space data processing out of NMI

  callchains, stack dump

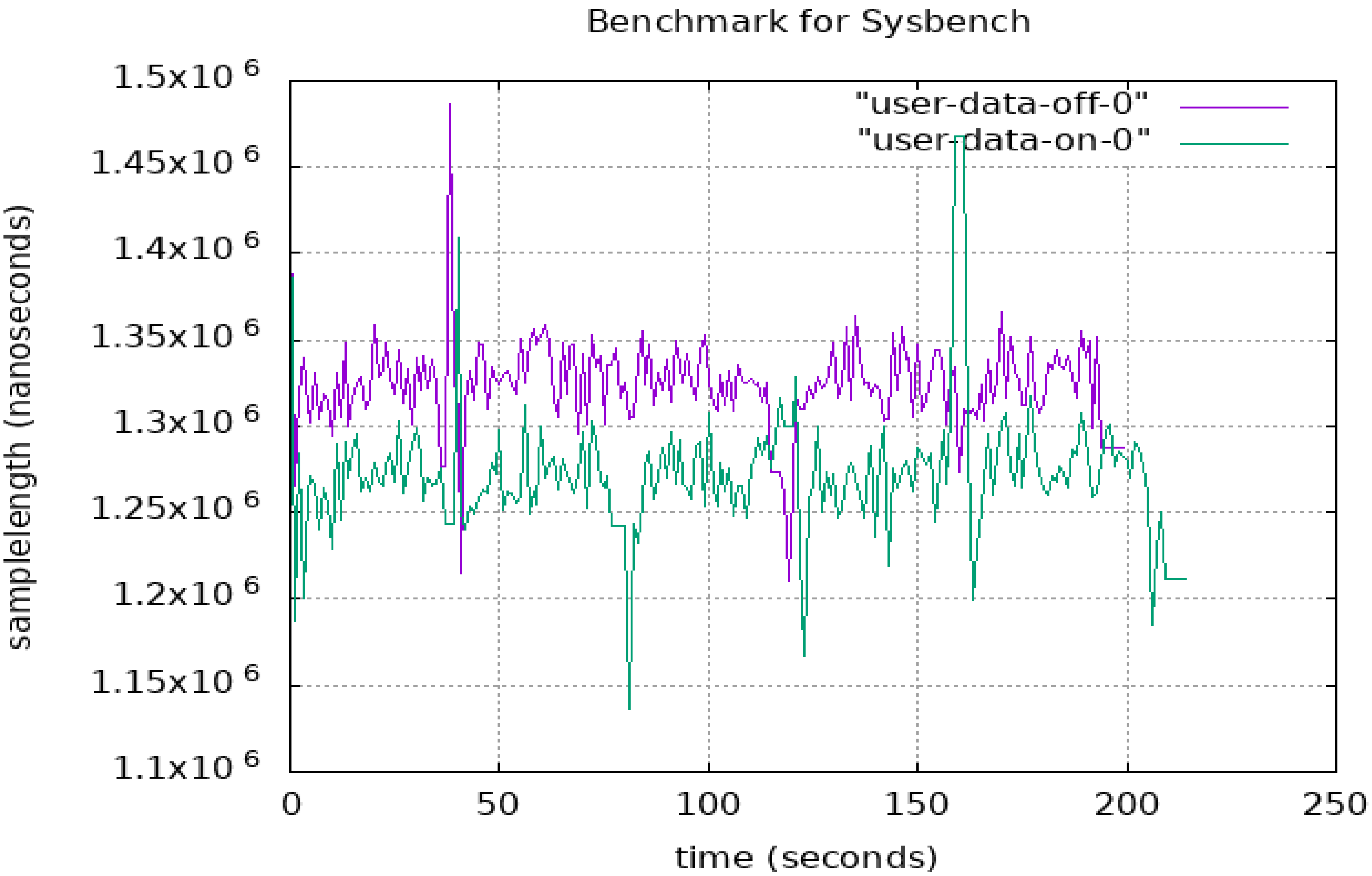


# DELAYED USER  SPACE DATA RETRIEVAL



- taskwork
- trigger window – slow syscall path
- perf tool processing

# DELAYED USER SPACE DATA RETRIEVAL



# DELAYED USER  SPACE DATA RETRIEVAL

- RFC post:

  https://lore.kernel.org/lkml/20180124121114.GA17605@krava/

- git:

  https://git.kernel.org/pub/scm/linux/kernel/git/jolsa/perf.git

  perf/user_data

# WATCH COMMAND

- watches various system files
- allows:
  - display separate fields
  - plot data
  - zero counters
- sched/int files now
- git:

  https://git.kernel.org/pub/scm/linux/kernel/git/jolsa/perf.git

  perf/watch

# CPUTIME PMU

- cputime subsystem, keeps counters:

```
CPUTIME_USER          CPUTIME_NICE      CPUTIME_SYSTEM
CPUTIME_SOFTIRQ       CPUTIME_IRQ       CPUTIME_IDLE
CPUTIME_IOWAIT        CPUTIME_STEAL     CPUTIME_GUEST
CPUTIME_GUEST_NICE
```

- cputime PMU mirrors those counters
- perf stat metric binding

```
perf stat --top/--top-full
```

- no tick issue

# CPUTIME PMU

- RFC post:

  https://lore.kernel.org/lkml/20180606221513.11302-1-jolsa@kernel.org/

- git:

  https://git.kernel.org/pub/scm/linux/kernel/git/jolsa/perf.git

  perf/cputime

# BPF COMMAND

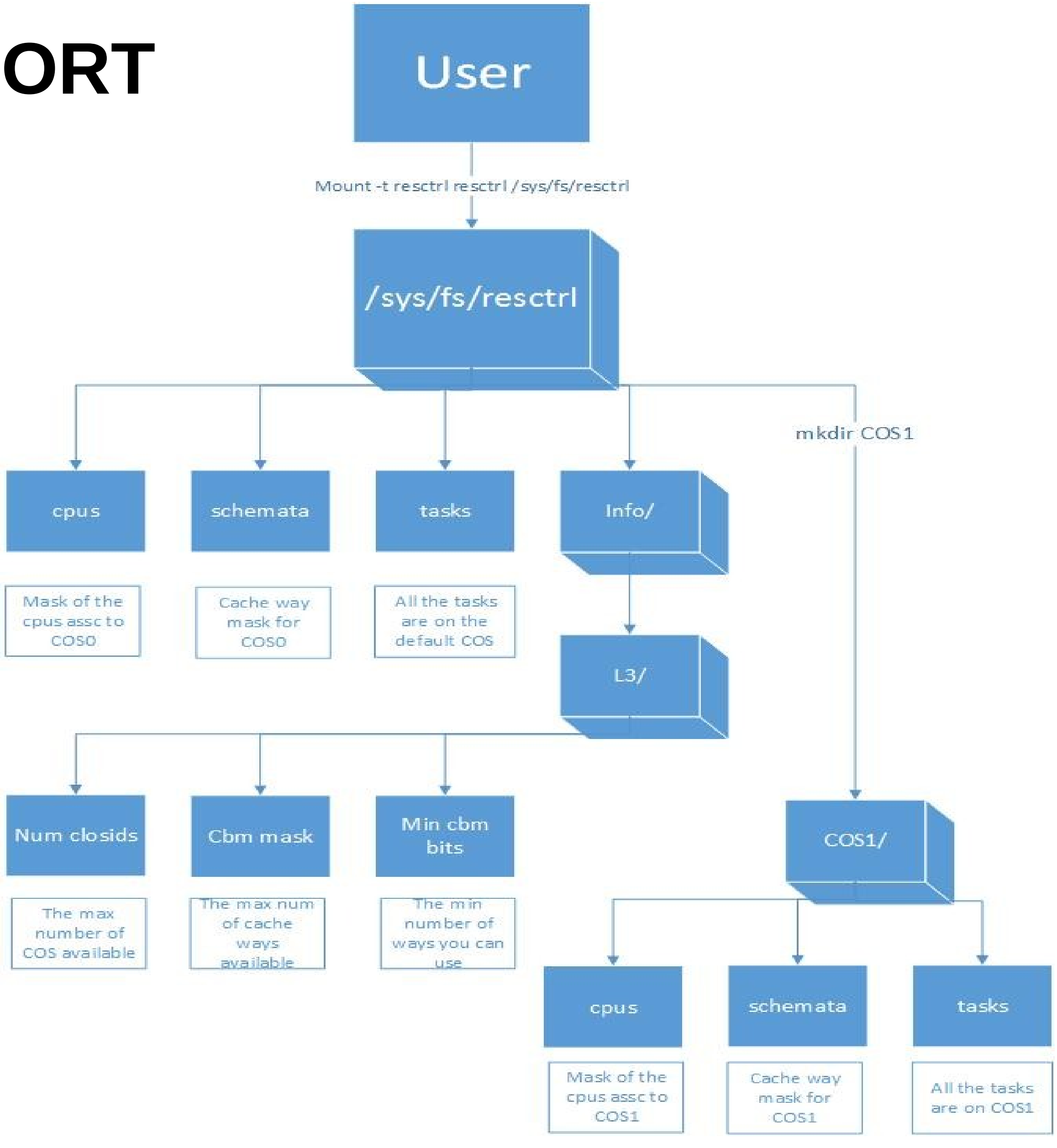
- framework for running BPF programs
- like bcc (python binding), but in C
- pre/post handlers
- event/timer handlers

# RDT SUPPORT

- Intel Resource Director Technology (RDT)

  CMT/MBM/CAT/CDP/MBA/CQM
- resctrl file system
- intel-cmt-cat package
- store resctrl in perf.data (as feature)
- report/load of RDT dump
- need rebase to new RDT design

# RDT SUPPORT


User
Mount -t resctrl resctrl /sys/fs/resctrl
/sys/fs/resctrl
cpus
Mask of the cpus assc to COS0
schemata
Cache way mask for COS0
tasks
All the tasks are on the default COS
Info/
L3/
Num closids
The max number of COS available
Cbm mask
The max num of cache ways available
Min cbm bits
The min number of ways you can use
mkdir COS1
COS1/
cpus
Mask of the cpus assc to COS1
schemata
Cache way mask for COS1
tasks
All the tasks are on COS1

source :https://github.com/intel/intel-cmt-cat/wiki/resctrl

# EVENT TRIGGERS

- https://perf.wiki.kernel.org/index.php/Jolsa_Features_Togle_Event
- allow events to start/stop another event

```
-e 'cycles,irq_entry/on=cycles/,irq_exit/off=cycles/'
```

- perf_event_event/ioctl interface change

# GROUP SHARING

- limit file descriptor usage
- group events do not allocate file descriptors
- perf specific IDs

# BUILD/MENUCONFIG

- make menuconfig
- compile out unneeded stuff (tests)
- cut out dependencies
- code separation

# THANKS, QUESTIONS

Jiri Olsa <jolsa@redhat.com>