tprof on NetBSD

Masanobu SAITOH (msaitoh@n.o)
Ryo SHIMIZU (ryo@n.o)

Who are we?

- We are working for IIJ to develop NetBSD based routers.
- The number of NetBSD developers in our team is 8.

pmc and tprof

- NetBSD had two different performance counter interfaces and commands.
- pmc(1)
  - Monitor PMCs.
  - Simple.
  - Specific to x86
- tprof(8) and tpfmt(1)
  - See the next page

What’s tprof?

- Sampling based profiler
  - Each time a performance counter overflows, the value of the PC at that time is recorded.
- Inspired by IBM AIX’s tprof?
- History on NetBSD:
  - First written by YAMAMOTO Takashi in 2008 (NetBSD 5.0).
    - global_power_events is used for the sampling event and can’t be changed to other event. It’s hardcoded in the backend.
    - x86 only
  - Revamped by Maxime Villard in 2018 (NetBSD 9.0).
    - It allows users to choose which event to count.
    - Generic(MI) PMC interface is implemented.
    - And then, Jared McNeill wrote the code for ARM.

Removal of pmc(1) stuff

- problem
  - pmc(1) had not been maintained for years.
  - The code is duplicated between pmc(1) stuff and tprof stuff. Not shared at all. There are two different kernel interfaces.
  - tprof(8) can use only one performance counter. Supporting multiple counters is in the TODO list.
- So, pmc(1) stuff was removed. NetBSD 9.0 had no pmc(1).

At that time

- We were using pmc(1) at that time.
- We were not using tprof at that time.
- We thought
  - if tprof functionality includes pmc functionality, then removing pmc is OK.
- We did not oppose the proposal to remove pmc(1) stuff.
- Another reason why NetBSD developers working for IIJ didn’t oppose the proposal is that our routers are based on netbsd-8 or prior.

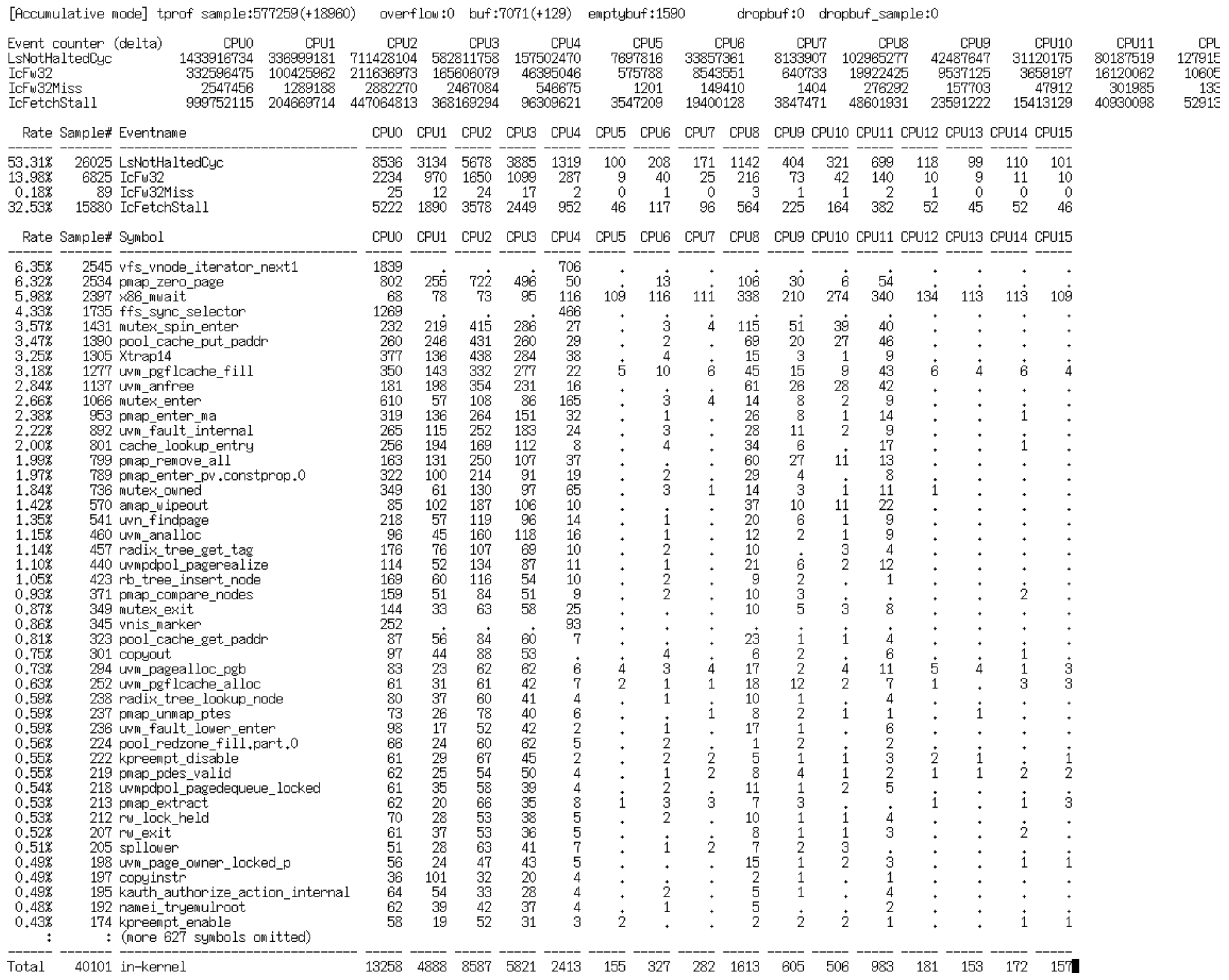
Problems

- We, out team,develop new functionality and improvement on NetBSD-current first and then backport them to netbsd-8.
- Sometimes we used pmc(1) to see some performance counters.
- tprof(8) can use only one counter.
  - It’s important to monitor more than one counter at a time.
  - For example, we cant’ calculate the last level cache’s hit ratio from llc-references and llc-misses.
- NetBSD-current has no pmc(1) anymore.

What’s new in NetBSD 10.0’s tprof?

- Support multiple counters at once.
- Subcommands:
  - list
  - monitor
  - analyze
  - count (<- new)
    - does not do any profiling, only outputs counters every interval.
  - top (<- new)
    - displays profiling results in real-time.
- (It’s not as feature-rich as FreeBSD’s pmcstat or Linux’s perf…)

Demo



```
[Accumulative mode] tprof sample:577259(+18960)   overflow:0  buf:7071(+129)  emptybuf:1590      dropbuf:0  dropbuf_sample:0

Event counter (delta)         CPU0       CPU1       CPU2       CPU3       CPU4       CPU5       CPU6       CPU7       CPU8       CPU9      CPU10      CPU11        CPL
LsNotHaltedCyc          1433916734  336999181  711428104  582811758  157502470    7697816   33857361    8133907  102965277   42487647   31120175   80187519     127915
IcFw32                   332596475  100425962  211636973  165606079   46395046     575788    8543551     640733   19922425    9537125    3659197   16120062      10605
IcFw32Miss                 2547456    1289188    2882270    2467084     546675       1201     149410       1404     276292     157703      47912     301985        133
IcFetchStall             999752115  204669714  447064813  368169294   96309621    3547209   19400128    3847471   48601931   23591222   15413129   40930098      52913

  Rate Sample# Eventname                         CPU0  CPU1  CPU2  CPU3  CPU4  CPU5  CPU6  CPU7  CPU8  CPU9 CPU10 CPU11 CPU12 CPU13 CPU14 CPU15
------ ------- -------------------------------- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
53.31%   26025 LsNotHaltedCyc                    8536  3134  5678  3885  1319   100   208   171  1142   404   321   699   118    99   110   101
13.98%    6825 IcFw32                            2234   970  1650  1099   287     9    40    25   216    73    42   140    10     9    11    10
 0.18%      89 IcFw32Miss                          25    12    24    17     2     0     1     0     3     1     1     2     1     0     0     0
32.53%   15880 IcFetchStall                      5222  1890  3578  2449   952    46   117    96   564   225   164   382    52    45    52    46

  Rate Sample# Symbol                            CPU0  CPU1  CPU2  CPU3  CPU4  CPU5  CPU6  CPU7  CPU8  CPU9 CPU10 CPU11 CPU12 CPU13 CPU14 CPU15
------ ------- -------------------------------- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
 6.35%    2545 vfs_vnode_iterator_next1          1839     .     .     .   706     .     .     .     .     .     .     .     .     .     .     .
 6.32%    2534 pmap_zero_page                     802   255   722   496    50     .    13     .   106    30     6    54     .     .     .     .
 5.98%    2397 x86_mwait                           68    78    73    95   116   109   116   111   338   210   274   340   134   113   113   109
 4.33%    1735 ffs_sync_selector                 1269     .     .     .   466     .     .     .     .     .     .     .     .     .     .     .
 3.57%    1431 mutex_spin_enter                   232   219   415   286    27     .     3     4   115    51    39    40     .     .     .     .
 3.47%    1390 pool_cache_put_paddr               260   246   431   260    29     .     2     .    69    20    27    46     .     .     .     .
 3.25%    1305 Xtrap14                            377   136   438   284    38     .     4     .    15     3     1     9     .     .     .     .
 3.18%    1277 uvm_pgflcache_fill                 350   143   332   277    22     5    10     6    45    15     9    43     6     4     6     4
 2.84%    1137 uvm_anfree                         181   198   354   231    16     .     .     .    61    26    28    42     .     .     .     .
 2.66%    1066 mutex_enter                        610    57   108    86   165     .     3     4    14     8     2     9     .     .     .     .
 2.38%     953 pmap_enter_ma                      319   136   264   151    32     .     1     .    26     8     1    14     .     .     1     .
 2.22%     892 uvm_fault_internal                 265   115   252   183    24     .     3     .    28    11     2     9     .     .     .     .
 2.00%     801 cache_lookup_entry                 256   194   169   112     8     .     4     .    34     6     .    17     .     .     1     .
 1.99%     799 pmap_remove_all                    163   131   250   107    37     .     .     .    60    27    11    13     .     .     .     .
 1.97%     789 pmap_enter_pv.constprop.0          322   100   214    91    19     .     2     .    29     4     .     8     .     .     .     .
 1.84%     736 mutex_owned                        349    61   130    97    65     .     3     1    14     3     1    11     1     .     .     .
 1.42%     570 amap_wipeout                        85   102   187   106    10     .     .     .    37    10    11    22     .     .     .     .
 1.35%     541 uvn_findpage                       218    57   119    96    14     .     1     .    20     6     1     9     .     .     .     .
 1.15%     460 uvm_analloc                         96    45   160   118    16     .     1     .    12     2     1     9     .     .     .     .
 1.14%     457 radix_tree_get_tag                 176    76   107    69    10     .     2     .    10     .     3     4     .     .     .     .
 1.10%     440 uvmpdpol_pagerealize               114    52   134    87    11     .     1     .    21     6     2    12     .     .     .     .
 1.05%     423 rb_tree_insert_node                169    60   116    54    10     .     2     .     9     2     .     1     .     .     .     .
 0.93%     371 pmap_compare_nodes                 159    51    84    51     9     .     2     .    10     3     .     .     .     .     2     .
 0.87%     349 mutex_exit                         144    33    63    58    25     .     .     .    10     5     3     8     .     .     .     .
 0.86%     345 vnis_marker                        252     .     .     .    93     .     .     .     .     .     .     .     .     .     .     .
 0.81%     323 pool_cache_get_paddr                87    56    84    60     7     .     .     .    23     1     1     4     .     .     .     .
 0.75%     301 copyout                             97    44    88    53     .     .     4     .     6     2     .     6     .     .     1     .
 0.73%     294 uvm_pagealloc_pgb                   83    23    62    62     6     4     3     4    17     2     4    11     5     4     1     3
 0.63%     252 uvm_pgflcache_alloc                 61    31    61    42     7     2     1     1    18    12     2     7     1     .     3     3
 0.59%     238 radix_tree_lookup_node              80    37    60    41     4     .     1     .    10     1     .     4     .     .     .     .
 0.59%     237 pmap_unmap_ptes                     73    26    78    40     6     .     .     1     8     2     1     1     .     1     .     .
 0.59%     236 uvm_fault_lower_enter               98    17    52    42     2     .     1     .    17     1     .     6     .     .     .     .
 0.56%     224 pool_redzone_fill.part.0            66    24    60    62     5     .     2     .     1     2     .     2     .     .     .     .
 0.55%     222 kpreempt_disable                    61    29    67    45     2     .     2     2     5     1     1     3     2     1     .     1
 0.55%     219 pmap_pdes_valid                     62    25    54    50     4     .     1     2     8     4     1     2     1     1     2     2
 0.54%     218 uvmpdpol_pagedequeue_locked         61    35    58    39     4     .     2     .    11     1     2     5     .     .     .     .
 0.53%     213 pmap_extract                        62    20    66    35     8     1     3     3     7     3     .     .     1     .     1     3
 0.53%     212 rw_lock_held                        70    28    53    38     5     .     2     .    10     1     1     4     .     .     .     .
 0.52%     207 rw_exit                             61    37    53    36     5     .     .     .     8     1     1     3     .     .     2     .
 0.51%     205 spllower                            51    28    63    41     7     .     1     2     7     2     3     .     .     .     .     .
 0.49%     198 uvm_page_owner_locked_p             56    24    47    43     5     .     .     .    15     1     2     3     .     .     1     1
 0.49%     197 copyinstr                           36   101    32    20     4     .     .     .     2     1     .     1     .     .     .     .
 0.49%     195 kauth_authorize_action_internal     64    54    33    28     4     .     2     .     5     1     .     4     .     .     .     .
 0.48%     192 namei_tryemulroot                   62    39    42    37     4     .     1     .     5     .     .     2     .     .     .     .
 0.43%     174 kpreempt_enable                     58    19    52    31     3     2     .     .     2     2     2     1     .     .     1     1
     :       : (more 627 symbols omitted)
------ ------- -------------------------------- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
Total    40101 in-kernel                        13258  4888  8587  5821  2413   155   327   282  1613   605   506   983   181   153   172   157
```

TODO

- x86:
  - Add counter definitions that newer chips have.
  - Add support fixed PMC (Intel).
    - Currently support general counter only.
  - Get performance counter structure from CPUID 0x0a(Intel) and 0x80000022(AMD).
  - Use Intel PEBS (Processor Event-Based Sampling)
    - to reduce profiling overhead
    - to improve accuracy
  - AMD’s IBS is complexed. Need some investigation to use it.
- Collect events per LWP.
  - Our old pmc had the feature.
- Take some idea from FreeBSD and Linux’s PMC stuff.

Any questions?