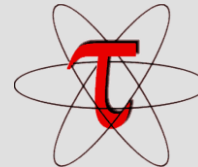


Unified CPU and GPU sampling with the TAU Performance System®

17th International Parallel Tools Workshop



UNIVERSITY
OF OREGON

ParaTools

Jordi Alcaraz
Research Associate
Performance Research Laboratory, OACISS, University of Oregon
jordia@uoregon.edu



Motivation

- Instrumenting entry and exit points provides a general idea of the time spent in an application
- As functions become larger and more complex, entry and exit points provide little information
- Finer-grained analysis is necessary to identify performance bottlenecks
- Profiling every line or instruction of an application is not feasible
 - Increases profiling overhead
 - Generates large amounts of profiling data

Our approach

- Periodically sample the application to identify performance hotspots
 - Low overhead
 - No need to modify the application or executable
- CPU uses signals to interrupt execution and collect samples
- GPUs provide samples using their own APIs
- Integrate CPU and GPU sampling into the TAU Performance System

TAU Performance System®

- Tuning and Analysis Utilities
- Supports profiling and tracing for a variety of runtimes:
 - MPI, CUDA, ROCm, Intel Level Zero, OpenCL, OpenMP, and others.
- Portable across architectures: x86, ARM, CRAY, PowerPC, etc.
- Supports hardware performance counters with PAPI, Likwid and GPU APIs
- Profiles uninstrumented binaries with tau_exec or tau_python
- Can instrument applications using compiler instrumentation, PDT, DyninstAPI and MAQAO
- Tracing is also available with Jumpshot, Vampir and Perfetto

CPU event based sampling

- Architecture independent (ARM, x86_64, PowerPC,...)
- No external API dependencies*
- Stops application at set intervals
 - Time based sampling uses POSIX timers to call a signal
 - * PAPI counters use overflow handler
- Reads the program counter and associates the sampling interval to the sample

CPU event based sampling

Environmental variables

- `TAU_EBS_UNWIND` //Unwind the callstack
- `TAU_EBS_RESOLUTION=line` //Can be set to function or file
- `TAU_EBS_SOURCE=itimer` //In μ s, accepts metrics (`PAPI_TOT_INS`)
- `TAU_EBS_PERIOD=<value>` //Sample each X units

How to use

- `tau_exec -ebs <application>`

CPU uninstrumented profiling

```
NODE 0;CONTEXT 0;THREAD 0:
-----
%Time    Exclusive    Inclusive    #Call    #Subrs    Inclusive Name
          msec      total msec                usec/call
-----
100.0         11         828         1         1    828057 .TAU application
 98.6         816         816         1         0    816427 int taupreload_main(int, char **, char **)
[jalcaraz@gilgamesh mm]$
```

Exclusive : time spent in a function, excluding its calls

Inclusive : time spent by the function and its calls

.TAU application: top-level timer. Includes application initialization.

CPU event based sampling

```
NODE 0;CONTEXT 0;THREAD 0:
-----
%Time    Exclusive    Inclusive    #Call    #Subrs    Inclusive Name
      msec      total msec                usec/call
-----
100.0         11        828         1         1    828057 .TAU application
 98.6         816        816         1         0    816427 int taupreload_main(int, char **, char **)
[jalcaraz@gilgamesh mm]$
```

1 sample every 50 milliseconds

```
NODE 0;CONTEXT 0;THREAD 0:
-----
%Time    Exclusive    Inclusive    #Call    #Subrs    Inclusive Name
      msec      total msec                usec/call
-----
100.0         11        826         1         1    826840 .TAU application
 98.6         814        814         1         0    814851 int taupreload_main(int, char **, char **)
 96.8          0        800        16         0    50000 [CONTEXT] int taupreload_main(int, char **, char **)
 96.8          0        800        16         0    50000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **)
 48.4         400        400         8         0    50000 [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {110}]
 48.4         400        400         8         0    50000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {110}]
 42.3         350        350         7         0    50000 [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {131}]
 42.3         350        350         7         0    50000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {131}]
 6.0          50         50         1         0    50000 [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {109}]
 6.0          50         50         1         0    50000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {109}]
```

CPU event based sampling

```

NODE 0;CONTEXT 0;THREAD 0:
-----
%Time    Exclusive    Inclusive    #Call    #Subrs    Inclusive Name
      msec      total msec
-----
100.0         11         828         1         1    828057 .TAU application
 98.6         816         816         1         0    816427 int taupreload_main(int, char **, char **)
[jalcaraz@gilgamesh mm]$

```

1 sample every millisecond, overhead lower than 1%

```

NODE 0;CONTEXT 0;THREAD 0:
-----
%Time    Exclusive    Inclusive    #Call    #Subrs    Inclusive Name
      msec      total msec
-----
100.0         12         831         1         1    831707 .TAU application
 99.9           0         831         819         0    1015 [CONTEXT] int taupreload_main(int, char **, char **)
 99.9           0         831         819         0    1015 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **)
 98.5         819         819         1         0    819599 int taupreload_main(int, char **, char **)
 49.4         410         410         411         0    1000 [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {110}]
 49.4         410         410         411         0    1000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {110}]
 38.6         320         320         321         0    1000 [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {131}]
 38.6         320         320         321         0    1000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {131}]
 5.5          46         46         46         0    1000 [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {109}]
 5.5          46         46         46         0    1000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {109}]
 4.3          36         36         36         0    1000 [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {130}]
 4.3          36         36         36         0    1000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {130}]
 1.7          14         14         2         0    7002 [SAMPLE] _int_malloc [{/usr/lib64/libc-2.28.so} {0}]
 1.7          14         14         2         0    7002 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] _int_malloc [{/usr/lib64/libc-2.28.so} {0}]
 0.2           1         1         2         0    1000 [SAMPLE] initialize [{/home/users/jalcaraz/tau2/examples/mm/matmult_initialize.c} {11}]
]
 0.2           1         1         2         0    1000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] initialize [{/home/users/jalcaraz/tau2/examples/mm/matmult_initialize.c} {11}]
 0.1           1         1         1         0    1000 [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {129}]
 0.1           1         1         1         0    1000 int taupreload_main(int, char **, char **) => [CONTEXT] int taupreload_main(int, char
**, char **) => [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {129}]

```

CPU sampling and compiler instrumentation

```

NODE 0;CONTEXT 0;THREAD 0:
-----
%Time   Exclusive   Inclusive   #Call   #Subrs   Inclusive Name
      msec      total msec                usec/call
-----
100.0         11         838         1         1   838786 .TAU application
98.6         0.011         827         1         1   827233 int taupreload_main(int, char **, char **)
98.6          4         827         1         2   827222 main [{matmult.c} {224}]
54.6         458         458         1         0   458157 compute [{matmult.c} {100}]
54.6          0         458        458         0   1000 [CONTEXT] compute [{matmult.c} {100}]
54.6          0         458        458         0   1000 compute [{matmult.c} {100}] => [CONTEXT] compute [{matmult.c} {100}]
50.5         424         424        424         0   1000 [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {110}]
50.5         424         424        424         0   1000 compute [{matmult.c} {100}] => [CONTEXT] compute [{matmult.c} {100}] => [SAMPLE] compu
te [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {110}]
43.5          0         365        365         0   1000 [CONTEXT] compute_interchange [{matmult.c} {121}]
43.5          0         365        365         0   1000 compute_interchange [{matmult.c} {121}] => [CONTEXT] compute_interchange [{matmult.c}
{121}]
43.4         364         364         1         0   364448 compute_interchange [{matmult.c} {121}]
39.5         331         331        331         0   1000 [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {131}]
39.5         331         331        331         0   1000 compute_interchange [{matmult.c} {121}] => [CONTEXT] compute_interchange [{matmult.c}
{121}] => [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {131}]
4.1          33         33         34         0   1000 [SAMPLE] compute [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {109}]
4.1          33         33         34         0   1000 compute [{matmult.c} {100}] => [CONTEXT] compute [{matmult.c} {100}] => [SAMPLE] compu
te [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {109}]
3.9          32         32         33         0   1000 [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {130}]
3.9          32         32         33         0   1000 compute_interchange [{matmult.c} {121}] => [CONTEXT] compute_interchange [{matmult.c}
{121}] => [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {130}]
1.8          0         15         4         0   3751 [CONTEXT] main [{matmult.c} {224}]
1.8          0         15         4         0   3751 main [{matmult.c} {224}] => [CONTEXT] main [{matmult.c} {224}]
1.6         13         13         2         0   6503 [SAMPLE] _int_malloc [{/usr/lib64/libc-2.28.so} {0}]
1.6         13         13         2         0   6503 main [{matmult.c} {224}] => [CONTEXT] main [{matmult.c} {224}] => [SAMPLE] _int_malloc
[/{usr/lib64/libc-2.28.so} {0}]
0.2          1         1         2         0   998 [SAMPLE] initialize [{/home/users/jalcaraz/tau2/examples/mm/matmult_initialize.c} {12}]
]
0.2          1         1         2         0   998 main [{matmult.c} {224}] => [CONTEXT] main [{matmult.c} {224}] => [SAMPLE] initialize
[/{home/users/jalcaraz/tau2/examples/mm/matmult_initialize.c} {12}]
0.1          1         1         1         0   1000 [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {129}]
0.1          1         1         1         0   1000 compute_interchange [{matmult.c} {121}] => [CONTEXT] compute_interchange [{matmult.c}
{121}] => [SAMPLE] compute_interchange [{/home/users/jalcaraz/tau2/examples/mm/matmult.c} {129}]

```

GPU sampling

- Adds significant overhead as APIs serialize kernels
- May not work with all GPUs
- Vendor specific limitations and requirements

CUDA sampling

- Provides stall reason
- Does not provide timestamps
- Requires reserving GPU memory
 - May need multiple GBs, more memory=more overhead
 - May crash without enough memory (TAU_CUPTI_PC_HWB in MB)
- Can set sampling interval with TAU_CUPTI_PC_PERIOD
- Interval in cycles
- `tau_exec -cupti_pc <application>`

CUDA sampling

Without sampling nor CUPTI

NODE 0;CONTEXT 0;THREAD 0:

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	13	2,536	1	1	2536052	.TAU application
99.5	311	2,522	1	120	2522693	taupreload_main
63.6	1,613	1,613	30	0	53777	OpenMP_Target compute_target
23.5	596	596	30	0	19900	OpenMP_Parallel_Region main
0.0	1	1	60	0	20	OpenMP_Target main

CUDA sampling

Without sampling nor CUPTI

NODE 0;CONTEXT 0;THREAD 0:

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	13	2,536	1	1	2536052	.TAU application
99.5	311	2,522	1	120	2522693	taupreload_main
63.6	1,613	1,613	30	0	53777	OpenMP_Target compute_target
23.5	596	596	30	0	19900	OpenMP_Parallel_Region main
0.0	1	1	60	0	20	OpenMP_Target main

With CUPTI

NODE 0;CONTEXT 0;THREAD 0:

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	23	2,622	1	1	2622594	.TAU application
99.1	369	2,599	1	120	2599594	taupreload_main
61.6	1,614	1,614	30	0	53817	OpenMP_Target compute_target
23.4	614	614	30	0	20471	OpenMP_Parallel_Region main
0.1	1	1	60	0	22	OpenMP_Target main

CUDA sampling

With CUPTI

NODE 0;CONTEXT 0;THREAD 0:

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	23	2,622	1	1	2622594	.TAU application
99.1	369	2,599	1	120	2599594	taupreload_main
61.6	1,614	1,614	30	0	53817	OpenMP_Target compute_target
23.4	614	614	30	0	20471	OpenMP_Parallel_Region main
0.1	1	1	60	0	22	OpenMP_Target main

With sampling, overhead 100%

NODE 0;CONTEXT 0;THREAD 0:

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	1,240	4,806	1	1	4806060	.TAU application
74.2	818	3,565	1	120	3565997	taupreload_main
33.5	1,611	1,611	30	0	53703	OpenMP_Target compute_target
17.6	844	844	30	0	28136	OpenMP_Parallel_Region main
6.1	291	291	60	0	4866	OpenMP_Target main

CUDA sampling output

Samples are received as a summary.

Total=NumSamples*MeanValue

USER EVENTS Profile :NODE 0, CONTEXT 0, THREAD 1

```
-----
NumSamples   MaxValue   MinValue   MeanValue   Std. Dev.   Event Name
-----
4           236         77        154.5       77.07 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_branch_resolving] OMP OFFLOADING co
mpute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{108}]
2           242         2         122        120 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_branch_resolving] OMP OFFLOADING co
mpute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{109}]
2 4.113E+04   90 2.061E+04 2.052E+04 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_branch_resolving] OMP OFFLOADING co
mpute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{110}]
1           228        228        228         0 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_branch_resolving] OMP OFFLOADING co
mpute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{113}]
4           211         66        134.2      67.13 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_branch_resolving_not_issued] OMP OF
FLOADING compute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{108}]
1           230        230        230         0 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_branch_resolving_not_issued] OMP OF
FLOADING compute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{109}]
2 3.876E+04   84 1.942E+04 1.934E+04 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_branch_resolving_not_issued] OMP OF
FLOADING compute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{110}]
1           202        202        202         0 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_branch_resolving_not_issued] OMP OF
FLOADING compute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{113}]
9           91         1         13.67      27.46 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_dispatch_stall] OMP OFFLOADING comp
ute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{107}]
58          69         1         6          13.8 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_dispatch_stall] OMP OFFLOADING comp
ute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{108}]
20 5.378E+04   1        2841      1.169E+04 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_dispatch_stall] OMP OFFLOADING comp
ute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{109}]
83          3354        4         862.7      666.8 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_dispatch_stall] OMP OFFLOADING comp
ute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{110}]
6           82         1         15         29.97 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_dispatch_stall_not_issued] OMP OFFL
OADING compute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{107}]
32          60         1         6.719      16 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_dispatch_stall_not_issued] OMP OFFL
OADING compute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{108}]
10          137         1         55.9       47.49 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_dispatch_stall_not_issued] OMP OFFL
OADING compute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{109}]
72          1103        1        315.9      260 [cupti sample] Stall reason [smsp_pcsamp_warps_issue_stalled_dispatch_stall_not_issued] OMP OFFL
OADING compute_target [/{storage/users/jalcaraz/sampling/tau2_cuda_stra/examples/openmp/target/matmult.c},{110}]
-----
```

Level Zero sampling

- Low overhead
- Provides stall reason if available
- Does not provide timestamps
- Available Intel(R) Data Center GPU Max Series and later GPUs
 - Needs EUStallSampling event
- Can set sampling interval with `TAU_L0_SAMPLING_INTERVAL`
- Interval in microseconds
- `tau_exec -l0_pc <application>`

Level Zero sampling

Without sampling nor Level Zero

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	181	5,875	1	1	5875851	.TAU application
96.9	219	5,694	1	120	5694823	int taupreload_main(int, char **, char **)
84.1	4,942	4,942	10	0	494260	OpenMP_Target compute_target.DIR.OMP.TARGET.DATA.1
5.8	338	338	20	0	16946	OpenMP_Parallel_Region initialize
2.6	154	154	10	0	15428	OpenMP_Parallel_Region do_work
0.6	33	33	30	0	1115	OpenMP_Target allocateMatrix
0.1	6	6	30	0	201	OpenMP_Target freeMatrix
0.0	0.212	0.212	20	0	11	OpenMP_Target compute_target

Level Zero sampling

Without sampling nor Level Zero

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	181	5,875	1	1	5875851	.TAU application
96.9	219	5,694	1	120	5694823	int taupreload_main(int, char **, char **)
84.1	4,942	4,942	10	0	494260	OpenMP_Target compute_target.DIR.OMP.TARGET.DATA.1
5.8	338	338	20	0	16946	OpenMP_Parallel_Region initialize
2.6	154	154	10	0	15428	OpenMP_Parallel_Region do_work
0.6	33	33	30	0	1115	OpenMP_Target allocateMatrix
0.1	6	6	30	0	201	OpenMP_Target freeMatrix
0.0	0.212	0.212	20	0	11	OpenMP_Target compute_target

Without sampling

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	17	5,991	1	38	5991944	.TAU application
99.7	245	5,973	1	243	5973516	int taupreload_main(int, char **, char **)
86.0	1	5,155	10	133	515582	OpenMP_Target compute_target.DIR.OMP.TARGET.DATA.1
86.0	5,152	5,152	21	0	245342	zeEventHostSynchronize
5.9	355	355	20	0	17764	OpenMP_Parallel_Region initialize
2.8	166	166	10	0	16685	OpenMP_Parallel_Region do_work

Level Zero sampling

Without sampling

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	17	5,991	1	38	5991944	.TAU application
99.7	245	5,973	1	243	5973516	int taupreload_main(int, char **, char **)
86.0	1	5,155	10	133	515582	OpenMP_Target compute_target.DIR.OMP.TARGET.DATA.1
86.0	5,152	5,152	21	0	245342	zeEventHostSynchronize
5.9	355	355	20	0	17764	OpenMP_Parallel_Region initialize
2.8	166	166	10	0	16685	OpenMP_Parallel_Region do_work

With sampling, overhead around 2%

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	301	6,137	1	38	6137308	.TAU application
95.1	207	5,834	1	243	5834727	int taupreload_main(int, char **, char **)
81.2	1	4,985	10	133	498523	OpenMP_Target compute_target.DIR.OMP.TARGET.DATA.1
81.2	4,982	4,982	21	0	237261	zeEventHostSynchronize
6.5	398	398	20	0	19950	OpenMP_Parallel_Region initialize
3.1	189	189	10	0	18937	OpenMP_Parallel_Region do_work
0.6	0.769	35	30	120	1193	OpenMP_Target allocateMatrix
0.6	33	33	35	0	968	zeContextMakeMemoryResident

Level Zero sampling output

USER EVENTS Profile :NODE 0, CONTEXT 0, THREAD 1

NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.	Event Name
10	10	1	5.5	2.872	Correlation ID
10	10	1	5.5	2.872	Correlation ID : .TAU application => [L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}]
10	64	64	64	0	Group Count X
10	64	64	64	0	Group Count X : .TAU application
10	4096	4096	4096	0	Group Count Y
10	4096	4096	4096	0	Group Count Y : .TAU application
10	1	1	1	0	Group Count Z
10	1	1	1	0	Group Count Z : .TAU application
10	64	64	64	0	Group Size X
10	64	64	64	0	Group Size X : .TAU application
10	1	1	1	0	Group Size Y
10	1	1	1	0	Group Size Y : .TAU application
10	1	1	1	0	Group Size Z
10	1	1	1	0	Group Size Z : .TAU application
2	112	8	60	52	Memory Copy Size [MemoryCopy(D2M)]
2	112	8	60	52	Memory Copy Size [MemoryCopy(D2M)] : .TAU application
1	48	48	48	0	Memory Copy Size [MemoryCopy(H2D)]
1	48	48	48	0	Memory Copy Size [MemoryCopy(H2D)] : .TAU application
8	3.277E+04	4	4294	1.076E+04	Memory Copy Size [MemoryCopy(M2D)]
8	3.277E+04	4	4294	1.076E+04	Memory Copy Size [MemoryCopy(M2D)] : .TAU application
5	1	1	1	0	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] { sync.nop null {Compacted,\$2.dst}} OtherStall[Events]
27	166	14	70.74	39.69	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] { sync.nop null {Compacted,\$2.dst}} SbidStall[Events]
27	70	2	34.59	21.06	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] { sync.nop null {Compacted,A@1}} InstrFetchStall[Events]
2	1	1	1	0	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] { sync.nop null {Compacted,A@1}} OtherStall[Events]
4	2	1	1.25	0.433	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] { sync.nop null {Compacted,I@1}} DistStall[Events]
3	1	1	1	0	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] {(W&f3.0) cmp (32 M0) (eq)f3.0 null<1>:d r3.1<0;1,0>:d r3.3<0;1,0>:d } DistStall[Events]
17	4	1	2.471	1.144	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] {(W&f3.0) cmp (32 M0) (eq)f3.0 null<1>:d r3.1<0;1,0>:d r3.3<0;1,0>:d } InstrFetchStall[Events]
18	4	1	1.611	0.8259	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] {(W&f3.0) cmp (32 M0) (gt)f3.0 null<1>:d r3.1<0;1,0>:d r3.3<0;1,0>:d } Active[Events]
4	1	1	1	0	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] {(W&f3.0) cmp (32 M0) (gt)f3.0 null<1>:d r3.1<0;1,0>:d r3.3<0;1,0>:d } InstrFetchStall[Events]
1	1	1	1	0	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] {(W&f3.0) cmp (32 M0) (gt)f3.0 null<1>:d r3.1<0;1,0>:d r3.3<0;1,0>:d } OtherStall[Events]
7	1	1	1	0	[L0] GPU: OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_l0/examples/openmp/target/matmult.c} {105,0}] {(W&f3.0) cmp (32 M0) (gt)f3.0 null<1>:d r3.1<0;1,0>:d r3.3<0;1,0>:d } PipeStall[Events]

ROCm sampling

- Beta feature. We need to enable env variable
ROCPROFILER_PC_SAMPLING_BETA_ENABLED
- Adds significant overhead
- Rocprofiler-sdk can use two different sampling modes
- `tau_exec -rocm_pc <application>`

ROCm sampling modes

Host-trap sampling:

- Software-based using a background kernel thread
- Samples at intervals and outputs the current instruction
- Intervals are defined in nanoseconds
- `tau_exec -rocm_pc_host <application>`

Stochastic sampling:

- Hardware-based using specific hardware inside the GPU
- Provides stalls if an instruction was not issued
- Intervals are defined as cycles using MHz
- `tau_exec -rocm_pc_stoc <application>`

ROCm sampling

AMD Instinct GPU	Architecture	Stochastic PC sampling	Host-trap PC sampling
MI355X	CDNA4	✓	✓
MI350X	CDNA4	✓	✓
MI325X	CDNA3	✓	✓
MI300X	CDNA3	✓	✓
MI300A	CDNA3	✓	✓
MI250X	CDNA2	✗	✓
MI250	CDNA2	✗	✓
MI210	CDNA2	✗	✓

ROCm sampling overhead

Without ROCm profiling

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	566	2,105	1	1	2105222	.TAU application
73.1	103	1,538	1	120	1538351	int taupreload_main(int, char **, char **)
55.8	1,174	1,174	30	0	39150	OpenMP_Target compute_target
7.2	152	152	20	0	7606	OpenMP_Parallel_Region initialize
3.6	76	76	10	0	7646	OpenMP_Parallel_Region do_work
1.4	29	29	30	0	987	OpenMP_Target allocateMatrix
0.1	1	1	30	0	62	OpenMP_Target freeMatrix

ROCm sampling overhead

Without ROCm profiling

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	566	2,105	1	1	2105222	.TAU application
73.1	103	1,538	1	120	1538351	int taupreload_main(int, char **, char **)
55.8	1,174	1,174	30	0	39150	OpenMP_Target compute_target
7.2	152	152	20	0	7606	OpenMP_Parallel_Region initialize
3.6	76	76	10	0	7646	OpenMP_Parallel_Region do_work
1.4	29	29	30	0	987	OpenMP_Target allocateMatrix
0.1	1	1	30	0	62	OpenMP_Target freeMatrix

With ROCm profiling

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive usec/call	Name
100.0	305	2,140	1	444	2140069	.TAU application
73.2	33	1,567	1	192	1567594	int taupreload_main(int, char **, char **)
57.2	2	1,223	30	80	40791	OpenMP_Target compute_target
57.0	1,220	1,220	24	0	50837	hsa_signal_wait_scacquire
6.9	147	147	20	0	7358	OpenMP_Parallel_Region initialize
6.8	145	145	4	0	36437	hsa_amd_memory_fill
5.6	119	119	4	0	29827	hsa_queue_create

ROCm sampling overhead

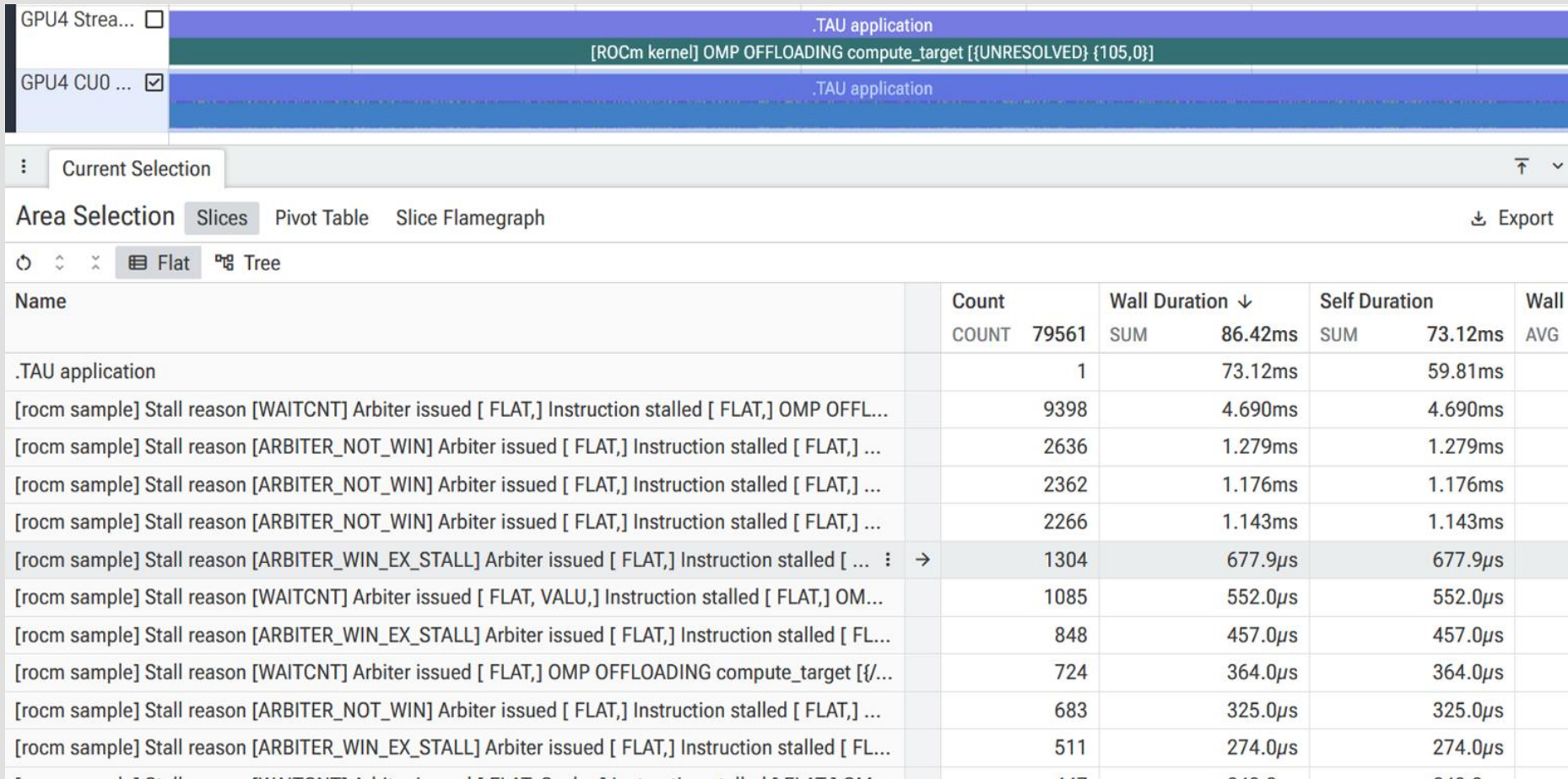
With ROCm profiling

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive Name usec/call
100.0	305	2,140	1	444	2140069 .TAU application
73.2	33	1,567	1	192	1567594 int taupreload_main(int, char **, char **)
57.2	2	1,223	30	80	40791 OpenMP_Target compute_target
57.0	1,220	1,220	24	0	50837 hsa_signal_wait_scacquire
6.9	147	147	20	0	7358 OpenMP_Parallel_Region initialize
6.8	145	145	4	0	36437 hsa_amd_memory_fill
5.6	119	119	4	0	29827 hsa_queue_create

ROCm sampling enabled, overhead 300%

%Time	Exclusive msec	Inclusive total msec	#Call	#Subrs	Inclusive Name usec/call
100.0	4,079	8,583	1	444	8583040 .TAU application
51.1	79	4,385	1	192	4385199 int taupreload_main(int, char **, char **)
45.6	2	3,910	30	80	130338 OpenMP_Target compute_target
45.5	3,906	3,906	24	0	162756 hsa_signal_wait_scacquire
2.3	194	194	20	0	9734 OpenMP_Parallel_Region initialize
1.4	116	116	4	0	29054 hsa_queue_create
1.1	90	90	10	0	9043 OpenMP_Parallel_Region do work

ROCM sampling with tracing



ROCM sampling with tracing

The screenshot displays a ROCm tracing interface. At the top, a timeline shows two tracks: 'GPU4 Stream...' and 'GPU4 CU0 10...'. The 'GPU4 Stream...' track contains a long blue bar labeled '.TAU application' and a green bar labeled '[ROCM kernel] OMP OFFLOADING compute_target [{UNRESOLVED} {105,0}]'. The 'GPU4 CU0 10...' track shows a blue bar labeled '.TAU application' with several small colored icons (purple, blue, green, brown) representing different events. Below the timeline, a 'Current Selection' bar shows a slice: 'Slice [rocm sample] Stall reason [ARBITER_NOT_WIN] Arbiter issued [FLAT,] Instruction stalled [FLAT,] OMP OFFLOADING compute_target [{/home/u... Contextual Options ▾'. The 'Details' panel on the left lists the following information:

- Name**: [rocm sample] Stall reason [ARBITER_NOT_WIN] Arbiter issued [FLAT,] Instruction stalled [FLAT,] OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_rocm/examples/openmp/amd_target/matmult.c}{110}] { global_store_dwordx2 v[4:5], v[8:9], off }
- Category**: tau
- Start time**: 00:00:00.841 107 004
- Duration**: 996ns
- Thread**: GPU4 CU0 [10000003]
- Process**: Rank 0 [1847852]
- SQL ID**: slice[39758] ▾

The 'Histogram' panel on the right shows the text: 'Histogram for '[rocm sample] Stall reason [ARBITER_NOT_WIN] Arbiter issued [FLAT,] Instruction stalled [FLAT,] OMP OFFLOADING compute_target [{/home/users/jalcaraz/sampling/tau2_rocm/examples/openmp/amd_target/matmult.c}{110}] { global_store_dwordx2 v[4:5], v[8:9], off }''. Below the histogram is a 'Scope' dropdown set to 'This track ▾' and a 'Load distribution' button.

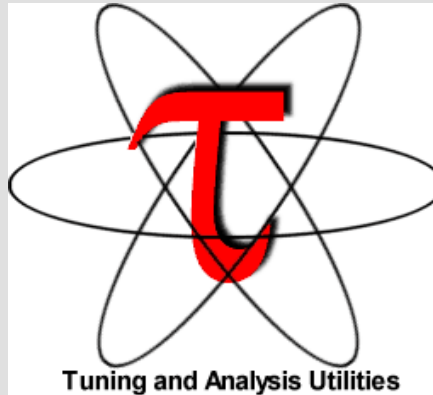
Conclusions

- CPU and GPU sampling can be performed using TAU
- CPU sampling has low overhead
- GPU sampling overhead depends on the application
 - Can be low with single streams
 - Or very high with multiple streams due to kernel serialization

Future Work

- Update to newer Intel API
- Update ROCm support and reduce overhead
- Modify CUDA implementation:
 - Try to print stalled instruction, not only code lines
- Improve readability

Download TAU from U. Oregon



<http://tau.uoregon.edu>

<https://e4s.io> [TAU in Docker/Singularity containers]

Free download, open source, BSD license

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Next-Generation Scientific Software Technologies program, under contract numbers DE-AC02-06CH11357, DE-AC05-00OR22725, DOE SBIR DE-SC0022502, Sandia National Laboratories Contract PO 2542488, and NSF OAC award 2311831.



U.S. DEPARTMENT OF
ENERGY

Office of
Science



- <https://science.osti.gov/ascr>
- <https://pesoproject.org>
- <https://ascr-step.org>
- <https://hpsf.io>
- <https://www.energy.gov/technologytransitions/sbirsttr>



