Enabling Profiling and Analysis Tools for Aurora

Rashawn L. Knapp
Intel, Software and Service Group (SSG)
Technical Computing, Analyzers, and Runtimes

Scalable Tools Workshop, Granlibakken Resort, Lake Tahoe, California
August 3-6, 2015

rashawn.l.knapp@intel.com

Goals

- Enable open source profiling and analysis tools for HPC to run well on Intel's newest and upcoming high-end server platforms.
- Collaboration of Oak Ridge, Argonne and Livermore National Laboratories (CORAL)
  - Intel with partner Cray to deliver two supercomputers to Argonne: Theta in 2016 (8.5 PF) and Aurora in 2018 (180 PF)
    - Knights Landing (KNL) for Theta and beyond for Aurora
- Current work on Xeon Haswell - EP through 2015
- Develop relationships with institutions and tool owners
  - Contribute patches to ensure tool coverage, quality, and performance on Intel platforms
    - Do this on Haswell and repeat on KNL (2016) and again on early Aurora servers
  - Demonstrate a path for all tools on the new platforms via Intel and GNU compilers
    - Why Intel Compilers?
      - Expectation is that these will produce the highest quality code for the Xeon Phi based nodes (especially when first released)
      - We will explore vectorization opportunities for optimization wherever possible.

Current Sample of Tools and Status Overview On Haswell

	Tool/Versions	Description	Status
Low-level tool Foundation	Dyninst 8.2.1	dynamic binary instrumentation tool	GNU and Intel compilations, Test suite completed, Minor change to CMake configuration
	PAPI 5.4.1	interface to sample CPU and off-core performance events	GNU and Intel compilations, Test suite completed, Patch accepted for off-core events
High-level Tools	TAU 2.24.1	profiling and tracing tool for parallel applications, supporting both MPI and OpenMP	Intel Compilation with Intel MPI and Intel C/C++/Fortran compilers, many suite examples tested
	Score-P 1.3	Provides a common interface for high-level tools	2015/16
	Open|Speedshop 2.1	Dynamic Instrumentation tool for Linux: profiling, event tracing for MPI and OpenMP programs. Incorporates Dyninst and PAPI	2015/16
	HPCToolKit 5.3.x r4793	Lightweight sampling measurement tool for HPC; supports PAPI	GNU and Intel compilations with Intel MPI, tests with PAPI and Intel MPI
	Darshan 5.3.2-r4532	IO monitoring tool	2015/16
Low-level Independent	Valgrind 3.10.1	framework for constructing dynamic analysis tools; includes suite of tools including a debugger, and error detection for memory and pthreads.	2015/16
	memcheck	Detects memory errors: stack, heap, memory leaks, and MPI distributed memory. For C and C++.	2015/16
	helgrind	Pthreads error detection: synchronization, incorrect use of pthreads API, potential deadlocks, data races. C, C++, Fortran	2015/16

Dyninst 8.2.1 Overview

Compilers	
GCC 5.1	Completed
Intel 15.03.187	Completed
MPI	
Intel 5.1.038	Completed
MPICH 3.1.4	TBD
Validation	

Test Suite
`./runTests -gcc -g++`
`./runTests -icpc -icc`
	
	363 tests, 353 PASSED, 10 SKIPPED, 0 CRASHED
	329 tests, 319 PASSED, 10 SKIPPED, 0 CRASHED

Examples in DyninstAPI Appendix A	GCC and Intel Ports
Contributions	
- CMake Configuration Change to enable Intel compilers	

Dyninst 8.2.1 Results

Intel 15.0.3 Test results (./runTests -icpc –icc):

```
In total 329 tests ran, 0 CRASHED, 10 SKIPPED, 319 PASSED, and 0 FAILED:
 22 test2_11                  icc    none 64  create   NA      dynamic nonPIC  SKIPPED
100 test2_11                  icpc   none 64  create   NA      dynamic nonPIC  SKIPPED
194 test1_35                  icc    none 64  create   NA      dynamic nonPIC  SKIPPED
195 test1_35                  icc    none 64  rewriter NA      dynamic nonPIC  SKIPPED
196 test1_35                  icpc   none 64  create   NA      dynamic nonPIC  SKIPPED
197 test1_35                  icpc   none 64  rewriter NA      dynamic nonPIC  SKIPPED
310 test_ser_anno             icc    none 64  disk     NA      dynamic nonPIC  SKIPPED
311 test_symtab_ser_funcs     icc    none 64  disk     NA      dynamic nonPIC  SKIPPED
320 test_ser_anno             icpc   none 64  disk     NA      dynamic nonPIC  SKIPPED
321 test_symtab_ser_funcs     icpc   none 64  disk     NA      dynamic nonPIC  SKIPPED
```

GCC 5.1.0 Test Results (./runTests -gcc -g++):

```
In total 363 tests ran, 0 CRASHED, 10 SKIPPED, 353 PASSED, and 0 FAILED:
 22 test2_11                  g++    none 64  create   NA      dynamic nonPIC  SKIPPED
100 test2_11                  gcc    none 64  create   NA      dynamic nonPIC  SKIPPED
194 test1_35                  g++    none 64  create   NA      dynamic nonPIC  SKIPPED
195 test1_35                  g++    none 64  rewriter NA      dynamic nonPIC  SKIPPED
196 test1_35                  gcc    none 64  create   NA      dynamic nonPIC  SKIPPED
197 test1_35                  gcc    none 64  rewriter NA      dynamic nonPIC  SKIPPED
344 test_ser_anno             g++    none 64  disk     NA      dynamic nonPIC  SKIPPED
345 test_symtab_ser_funcs     g++    none 64  disk     NA      dynamic nonPIC  SKIPPED
354 test_ser_anno             gcc    none 64  disk     NA      dynamic nonPIC  SKIPPED
355 test_symtab_ser_funcs     gcc    none 64  disk     NA      dynamic nonPIC  SKIPPED
```

PAPI 5.4.2 Overview

Compilers	
GCC 5.1	Completed
Intel 15.03.187	Completed
MPI (N/A)	
Validation	
ctests	104 tests: 96 PASSED (5 w/warning), 1 FAILED, 6 SKIPPED, 1 does not exist
perf_event	3 tests: 3 PASSED
perf_event_uncore	4 tests: 3 PASSED, 1 SKIPPED
native events (papi_native_avail)	814 base events yields total combination of 11,843 events (2080 added successfully)
Contributions	
- Patch accepted for off-core tests on Haswell-EP	

PAPI 5.4.2 Results

ctests

Total	Passed	Failed	Skipped	Event does not exist
104	96	1	6	1

Failed Test:

- zero.c - *Flops* validation error

Skipped:

- Dat-range.c - Itanium2 only
- calibrate.c – event does not exist
- earprofile.c - Not implemented
- p4_lst_ins.c - Pentium 4 only
- zero_shmem.c - openSHMEM
- zero_smp.c - architecture not included

Event does not exist:

- hlrates - flips, flops, failure

perf_event

Total	Passed	Failed	Skipped	Event does not exist
3	3	0	0	0

Passed Tests: perf_event_offcore_response.c, perf_event_system_wide.c, perf_event_user_kernel.c

native events

- 814 events
- 11,843 events with all possible combinations
  - 2,080 PASS by modifying unit mask value (1-10 tested).
  - 9763 combinations did not pass
    - Some may be important

perf_event_uncore (4 tests)

Total	Passed (3)	Skipped (1)
4	perf_event_uncore, perf_event_uncore_multiple, perf_event_uncore_cbox	perf_event_amd_northbridge.c

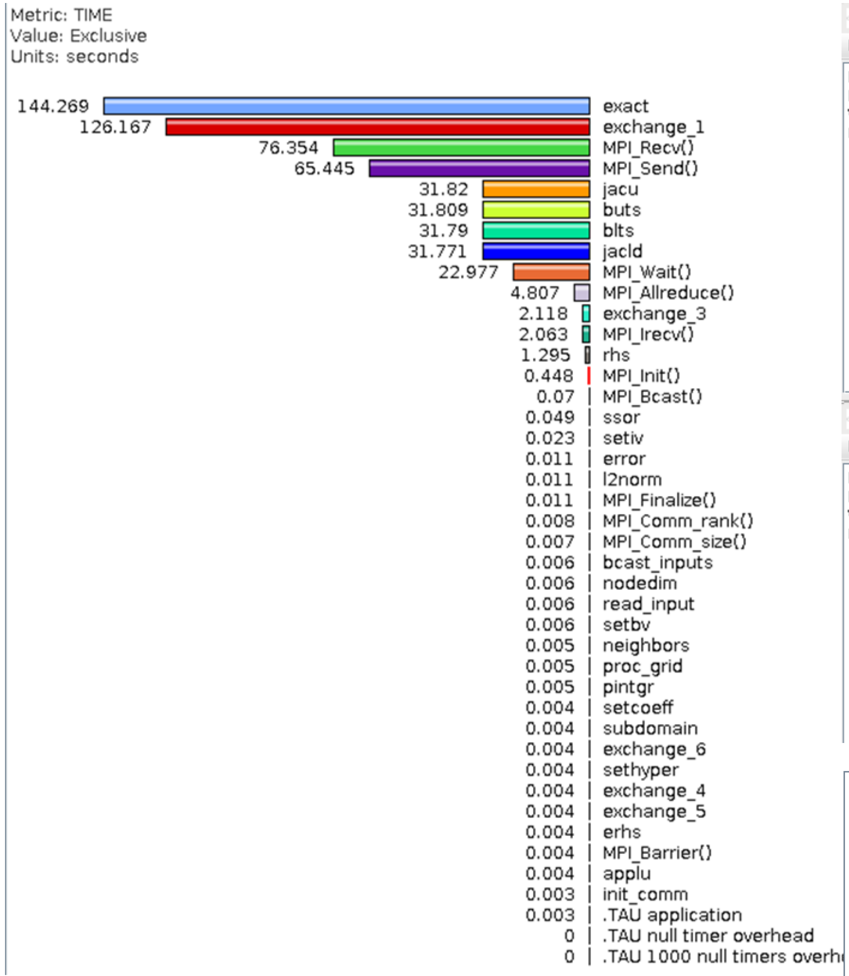
TAU 2.24.1 Overview

Compilers	
GCC 5.1	TBD
Intel 15.03.187	Completed
MPI	
Intel 5.1.038	Completed
MPICH 3.1.4	TBD
Validation	
Suite Examples	MPI and examples incorporating PAPI and Dyninst
Contributions	
- none	

TAU 2.24.1 Results - Examples

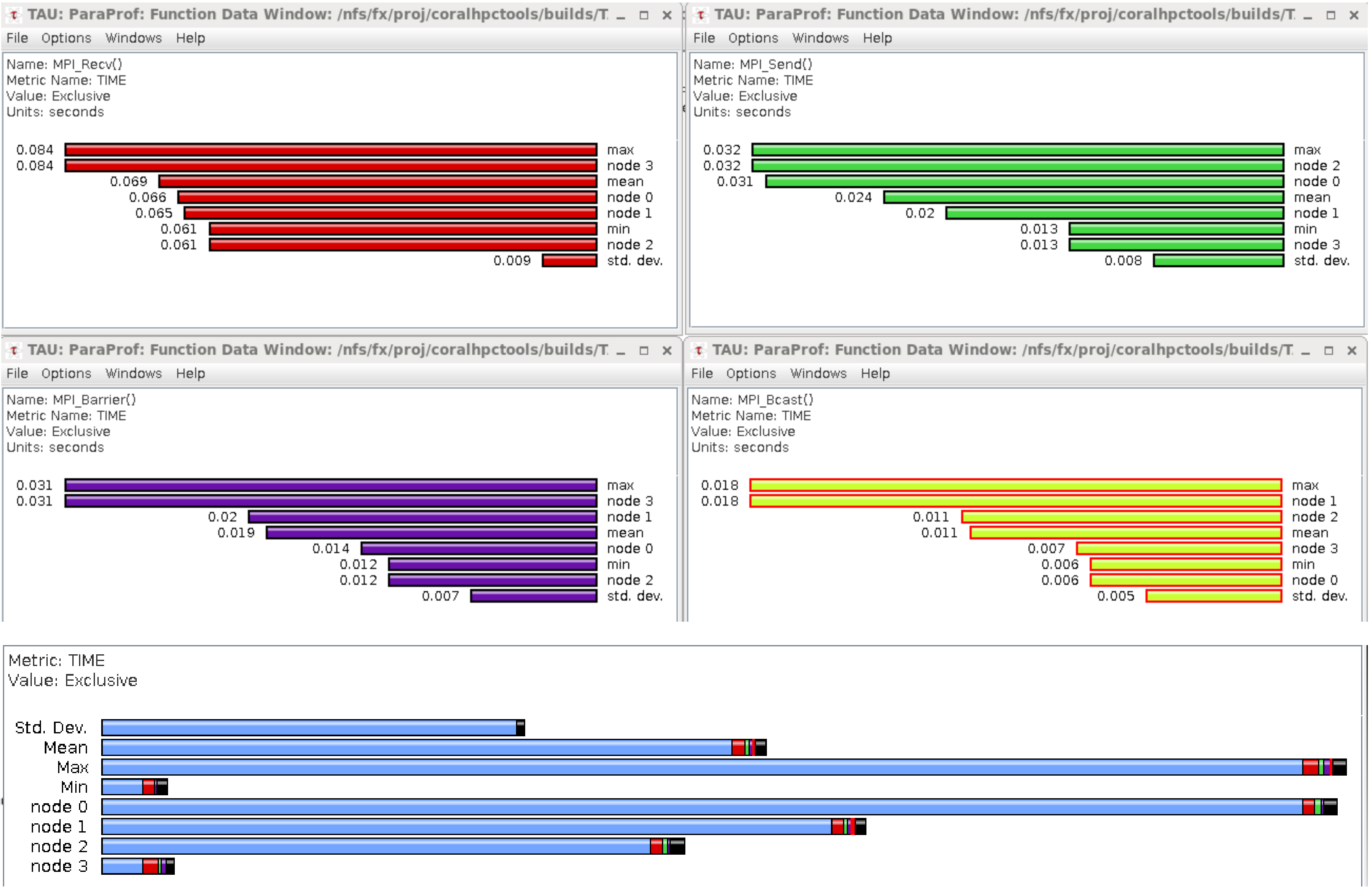
- taucompiler (c, f90, c++, mpic++)
  - tau_cc.sh -tau_makefile=$TAU_MAKEFILE -tau_options=-optCompInst -o ring ring.c
    - mpirun -n 4 -perhost 1 -env I_MPI_FABRICS tcp -hostfile <pathTo>/machines.LINUX ring
  - tau_cxx.sh -tau_makefile=$TAU_MAKEFILE -tau_options=-optCompInst klargest.cpp -o klargest
    - $ ./klargest 100 98
  - tau_f90.sh -tau_makefile=$TAU_MAKEFILE -tau_options=-optCompInst ring.f90 -o ring
    - mpirun -n 4 -perhost 1 -env I_MPI_FABRICS tcp -hostfile <pathTo>/machines.LINUX ring
  - tau_cxx.sh -tau_makefile=$TAU_MAKEFILE -tau_options=-optCompInst -o ring ring.cpp
    - mpirun -n 4 -perhost 1 -env I_MPI_FABRICS tcp -hostfile <pathTo>/machines.LINUX ring
  - paraprof
- taututorial (computePi)*
  - tau_cxx.sh -tau_makefile=$TAU_MAKEFILE -tau_options=-optCompInst computePi.cpp -o computePi
    - mpirun -n 4 -perhost 1 -env I_MPI_FABRICS tcp -hostfile <pathTo>/machines.LINUX computePi
- NPB2.3 (lu.W.4, sp.W.4)
  - mpirun -n 4 -perhost 1 -env I_MPI_FABRICS tcp -hostfile <pathTo>/machines.LINUX [lu.W.4|sp.W.4]
- Dyninst
  - tau_run -T pdt klargest 2500 23
- papi
  - setenv TAU_METRICS TIME:PAPI_TOT_CYC
  - ./simple

TAU 2.24.1 Results



Metric: TIME
Value: Exclusive
Units: seconds

NPB lu.w.4



TAU: ParaProf: Function Data Window: /nfs/fx/proj/coralhpctools/builds/T.
File Options Windows Help

Name: MPI_Recv()
Metric Name: TIME
Value: Exclusive
Units: seconds

TAU: ParaProf: Function Data Window: /nfs/fx/proj/coralhpctools/builds/T.
File Options Windows Help

Name: MPI_Send()
Metric Name: TIME
Value: Exclusive
Units: seconds

TAU: ParaProf: Function Data Window: /nfs/fx/proj/coralhpctools/builds/T.
File Options Windows Help

Name: MPI_Barrier()
Metric Name: TIME
Value: Exclusive
Units: seconds

TAU: ParaProf: Function Data Window: /nfs/fx/proj/coralhpctools/builds/T.
File Options Windows Help

Name: MPI_Bcast()
Metric Name: TIME
Value: Exclusive
Units: seconds

Metric: TIME
Value: Exclusive

ring.c

HPCToolKit Overview

Compilers	
GCC 5.1	Completed
Intel 15.03.187	Completed
MPI	
Intel 5.1.038	Completed
MPICH 3.1.4	TBD
Validation	
Compute Pi (cpi) example	4 nodes, 1 proc/node, PAPI_TOT_CYC and L2_TCM
HPL	4 nodes, 1 proc/node, PAPI_TOT_CYC and L2_TCM
Contributions	
- none	

HPCToolKit Results - CPI

CPI

```
mpiicc -g -O3 cpi.c -o cpi -lm
  hpcstruct ./cpi

  mpirun -np 4 -perhost 1 -env I_MPI_FABRICS tcp -hostfile <pathTo>/machines.LINUX
            hpcrun –t --event PAPI_TOT_CYC@10000 --event  WALLCLOCK@100000 --event PAPI_L2_TCM@10000
  ./cpi
  hpcprof -S cpi.hpcstruct -I ./'*' hpctoolkit-cpi-measurements
  hpctraceviewer hpctoolkit-mmult-database
  hpcviewer hpctoolkit-mmult-database
```

HPCToolKit Results - HPL

HPL

```
mpirun -np 4 -perhost 1 -env I_MPI_FABRICS tcp -hostfile <pathTo>/machines.LINUX \
hpcrun -t --event PAPI_TOT_CYC@10000 --event WALLCLOCK@100000 --event PAPI_L2_TCM@10000  \
./xhpl_intel64
hpcprof -S cpi.hpcstruct -I ./'*' hpctoolkit-cpi-measurements
hpctraceviewer hpctoolkit-mmult-database
hpcviewer hpctoolkit-mmult-database
```

With Instrumentation

```
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR01C2R4        1000   168     1     4               1.14            5.88332e-01
WR01C2R4        2000   168     1     4               0.72            7.38429e+00
```

With out Instrumentation

```
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR01C2R4        1000   168     1     4               0.95            7.04063e-01
WR01C2R4        2000   168     1     4               0.55            9.79022e+00
```

Summary, Challenges, and Next Steps

- Summary
  - We have started and have a plan to ensure that these tools run well on the CORAL machines
  - We have conducted coverage studies up to this point; still need to conduct quality and performance studies
  - We welcome collaboration with the tool groups
    - We will contribute patches as necessary
  - We started with the building block components of high level tools (e.g., Dyninst and PAPI), and we are now incorporating these into the higher level tools (OpenSpeed|Shop, Score-P).

- Challenges
  - We are working on small clusters at this time, but will need to transition to larger clusters to complete the performance studies

- Other open-source tools to consider for this contract?
  - STAT, MRNet

- New Technologies
  - Omni-Path network, NUMA technologies

Acknowledgments

All of the tool groups have been very responsive and helpful.

I want to thank Bill Williams from Dyninst who answered all of my questions regarding building, testing, and using.

Many thanks to the supportive PAPI team in guiding us through upgrading and testing.

And without my colleague, Preeti Suman, we would not have progressed to where we are.

References

CORAL

- http://insidehpc.com/2015/04/intel-build-coral-supercomputers-argonne-200-procurement/
- http://www.hpcwire.com/2015/04/09/argonnes-200-million-supercomputing-award/
- http://insidehpc.com/2015/05/interview-intels-alan-gara-discusses-the-180-petaflop-aurora-supercomputer/

New Technologies

- http://www.intel.com/content/www/us/en/high-performance-computing-fabrics/omni-path-architecture-fabric-overview.html
- http://www.cnet.com/news/intel-and-micron-debut-3d-xpoint-storage-technology-thats-1000-times-faster-than-existing-drives/

Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

Backup PAPI 5.4.2

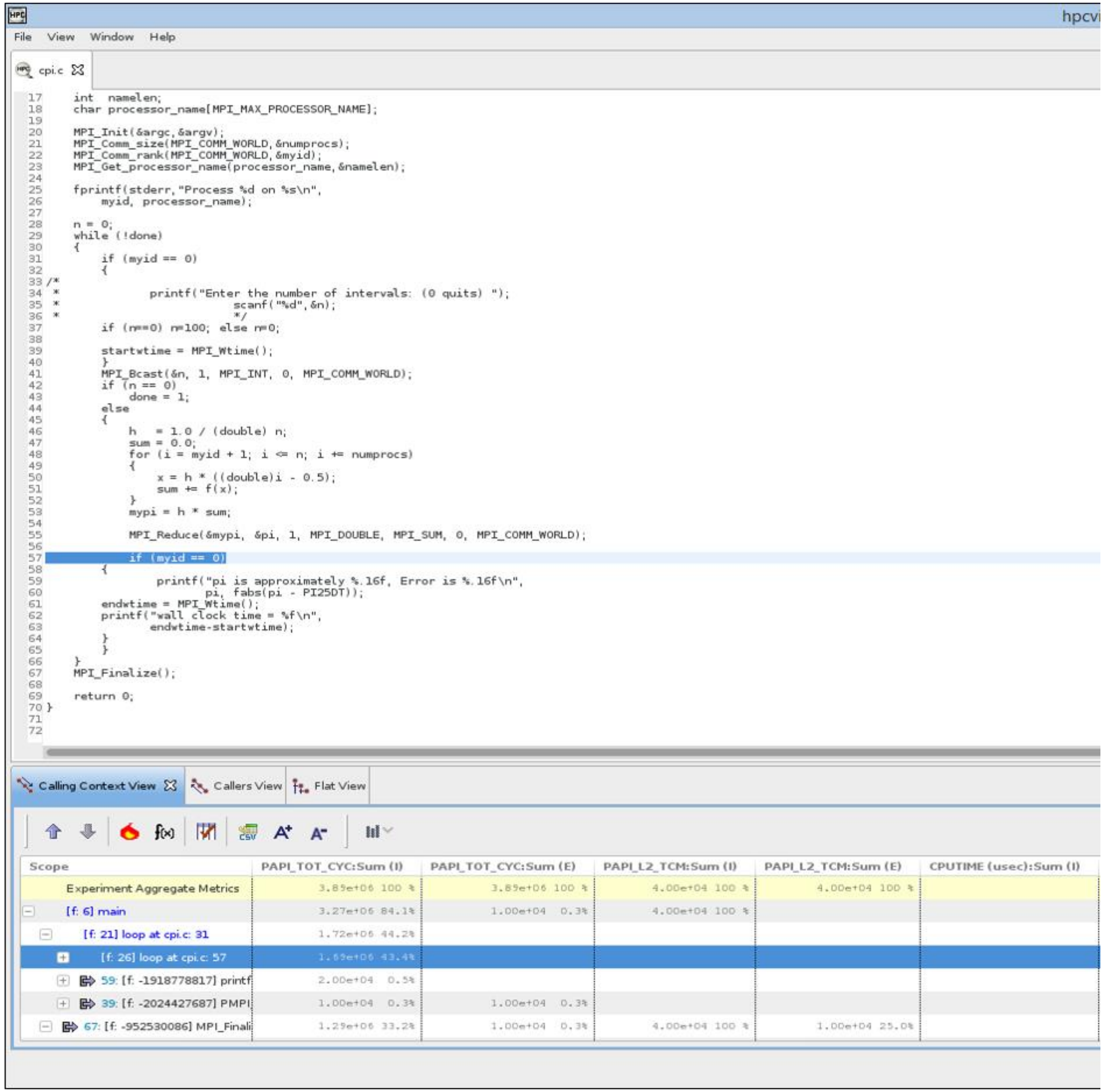
- Kernel upgrade from version 3.10 to 4.0.5, to enable uncore and offcore support on HSW
- Successfully installed PAPI-5.4.2 with GCC 5.1.0 and Intel Compilers
- Successfully added and tested uncore and offcore events to PAPI component tests
- Successfully added and tested imc uncore event support on HSW EP
- Reason for failed tests: disabled floating point counters
- 814 native events enabled on HSW
  - 11843 events extracted from all possible combination of native events and respective unit masks
  - 1848 events were successfully added and 232 events were successfully added after changing the unit mask value, ranging from 1 to 10.
  - 9,763 events that have not been added with the changes to the unit mask value. This returns two evenly distributed error messages: “invalid argument” and “Event does not exist”.

TAU 2.24.1 Backup

Configure:

```
./configure -c++=icpc -cc=icc -fortran=intel \
-pdt=<pathToPDT-3.20-IntelBuild> \
-papi=<pathToPAPI-5.4.2-IntelBuild> \
-PAPIWALLCLOCK -PAPIVIRTUAL -mpi \
-mpiinc=<pathToIntelMPI-5.1.0.38-IntelBuild>/compilers_and_libraries_2016/linux/mpi/intel64/include \
-mpilib=<pathToIntelMPI-5.1.0.38-IntelBuild>/compilers_and_libraries_2016/linux/mpi/intel64/lib \
-tag=IntelMPI5.1-IntelC15.3.187-PAPI5.4.2-Dyninst8.2.1-profiling \
-nocomm -COMPENSATE -PROFILEHEADROOM -PROFILEMEMORY -pthread \
-dyninst=<pathToDyninst-8.2.1-IntelBuild> -CPUTIME -LINUXTIMERS -iowrapper \
-prefix=<pathToTAU-2.24.1-IntelBuild> -bfd=download -unwind=download -pdtcompdir=intel \
-dwarflib=/nfs/fx/proj/coralhpctools/builds/libdwarf/intel/lib64<pathTo-libdwarf-20150507-IntelBuild>
```

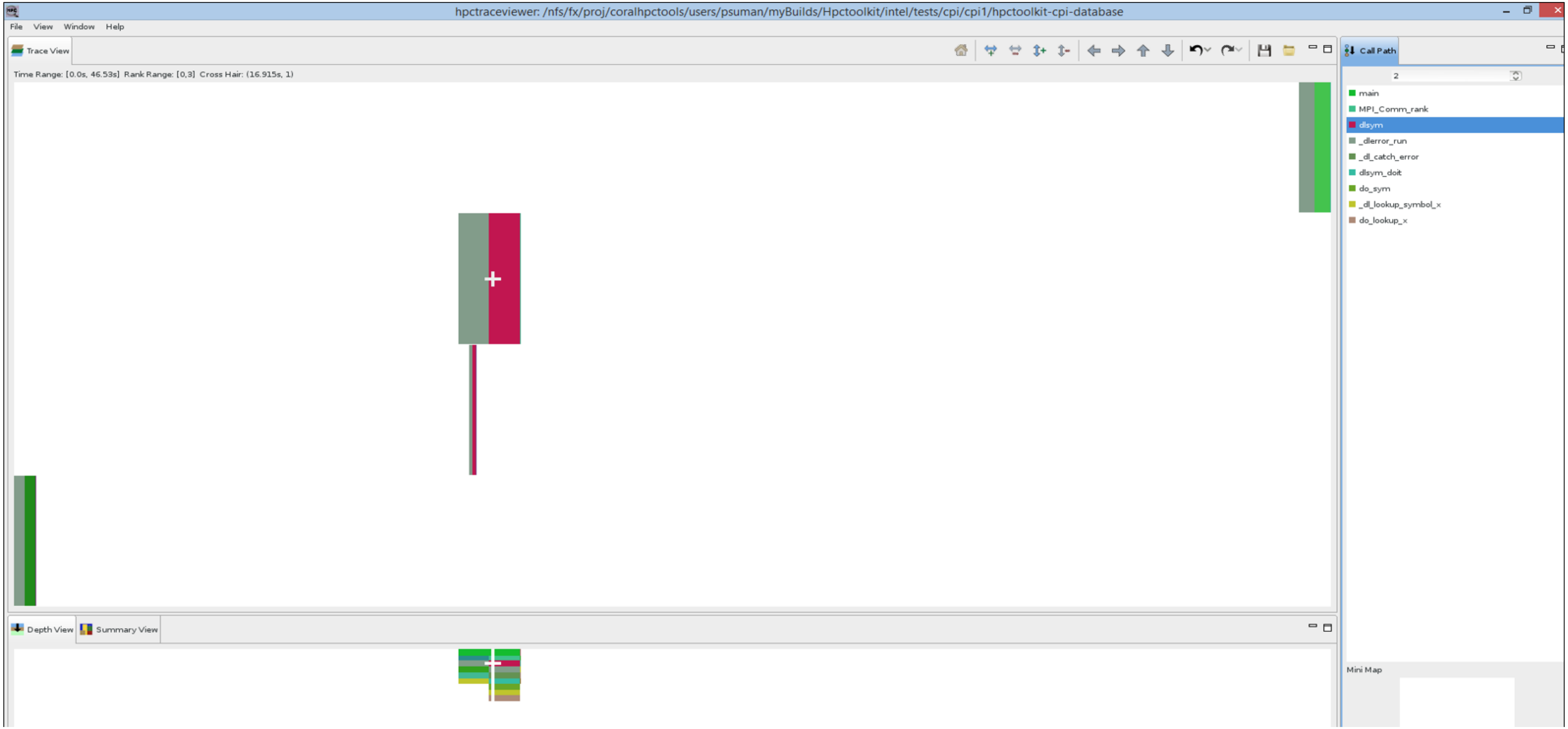
HPCToolKit - Calling Contexts View for cpi Backup



```
17     int  namelen;
18     char processor_name[MPI_MAX_PROCESSOR_NAME];
19
20     MPI_Init(&argc,&argv);
21     MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
22     MPI_Comm_rank(MPI_COMM_WORLD,&myid);
23     MPI_Get_processor_name(processor_name,&namelen);
24
25     fprintf(stderr,"Process %d on %s\n",
26         myid, processor_name);
27
28     n = 0;
29     while (!done)
30     {
31         if (myid == 0)
32         {
33 /*
34  *              printf("Enter the number of intervals: (0 quits) ");
35  *                          scanf("%d",&n);
36  *                          */
37         if (n==0) n=100; else n=0;
38
39         startwtime = MPI_Wtime();
40         }
41         MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
42         if (n == 0)
43             done = 1;
44         else
45         {
46             h   = 1.0 / (double) n;
47             sum = 0.0;
48             for (i = myid + 1; i <= n; i += numprocs)
49             {
50                 x = h * ((double)i - 0.5);
51                 sum += f(x);
52             }
53             mypi = h * sum;
54
55             MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);
56
57             if (myid == 0)
58         {
59                 printf("pi is approximately %.16f, Error is %.16f\n",
60                        pi, fabs(pi - PI25DT));
61         endwtime = MPI_Wtime();
62         printf("wall clock time = %f\n",
63               endwtime-startwtime);
64         }
65         }
66     }
67     MPI_Finalize();
68
69     return 0;
70 }
71
72
```

Scope	PAPI_TOT_CYC:Sum (I)	PAPI_TOT_CYC:Sum (E)	PAPI_L2_TCM:Sum (I)	PAPI_L2_TCM:Sum (E)	CPUTIME (usec):Sum (I)
Experiment Aggregate Metrics	3.89e+06 100 %	3.89e+06 100 %	4.00e+04 100 %	4.00e+04 100 %	
[f: 6] main	3.27e+06 84.1%	1.00e+04 0.3%	4.00e+04 100 %		
[f: 21] loop at cpi.c: 31	1.72e+06 44.2%				
[f: 26] loop at cpi.c: 57	1.69e+06 43.4%				
59: [f: -1918778817] printf	2.00e+04 0.5%				
39: [f: -2024427687] PMPI	1.00e+04 0.3%	1.00e+04 0.3%			
67: [f: -952530086] MPI_Finali	1.29e+06 33.2%	1.00e+04 0.3%	4.00e+04 100 %	1.00e+04 25.0%	

HPCToolKit - Depth View for cpi Backup



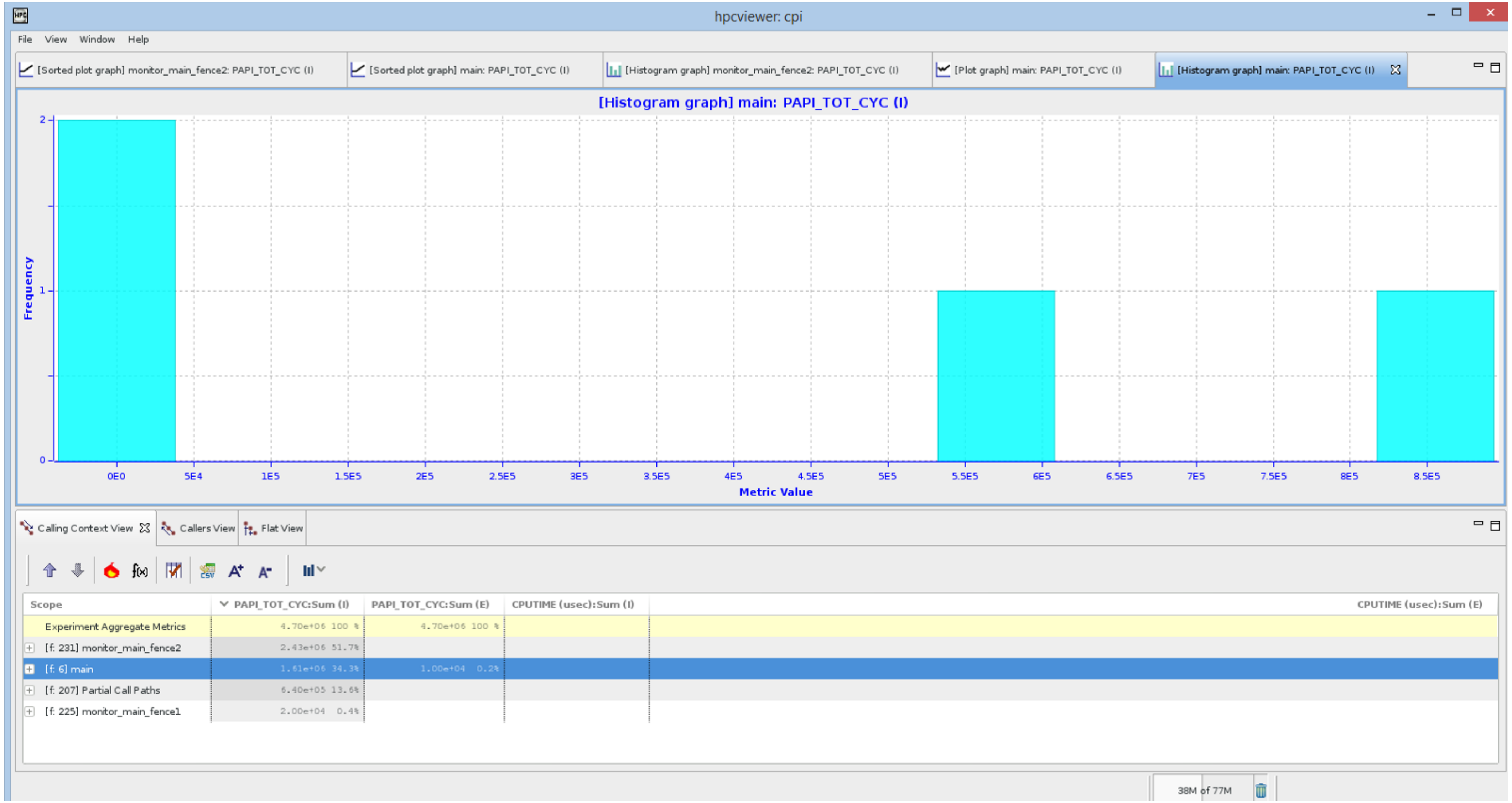
hpctraceviewer: /nfs/fx/proj/coralhpctools/users/psuman/myBuilds/Hpctoolkit/intel/tests/cpi/cpi1/hpctoolkit-cpi-database

Time Range: [0.0s, 46.53s] Rank Range: [0,3] Cross Hair: (16.915s, 1)

2

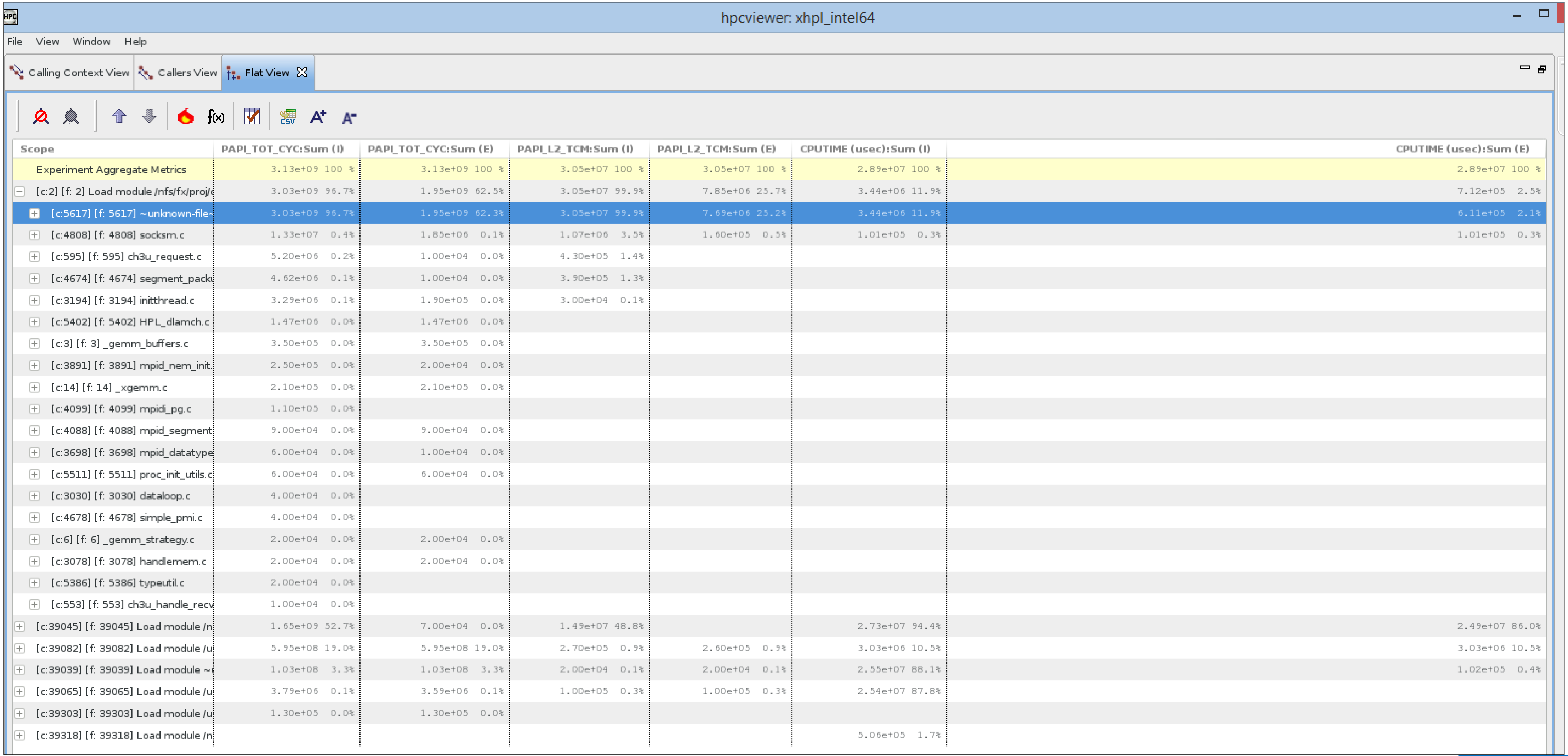
- 🟩 main
- 🟩 MPI_Comm_rank
- 🟥 dlsym
- ■ _dlerror_run
- 🟩 _dl_catch_error
- 🟩 dlsym_doit
- 🟩 do_sym
- 🟨 _dl_lookup_symbol_x
- 🟫 do_lookup_x

HPCToolKit - Histogram for cpi Backup


hpcviewer: cpi
File
View
Window
Help
[Sorted plot graph] monitor_main_fence2: PAPI_TOT_CYC (I)
[Sorted plot graph] main: PAPI_TOT_CYC (I)
[Histogram graph] monitor_main_fence2: PAPI_TOT_CYC (I)
[Plot graph] main: PAPI_TOT_CYC (I)
[Histogram graph] main: PAPI_TOT_CYC (I)
[Histogram graph] main: PAPI_TOT_CYC (I)
Frequency
2
1
0
0E0
5E4
1E5
1.5E5
2E5
2.5E5
3E5
3.5E5
4E5
4.5E5
5E5
5.5E5
6E5
6.5E5
7E5
7.5E5
8E5
8.5E5
Metric Value
Calling Context View
Callers View
Flat View
f(x)
38M of 77M

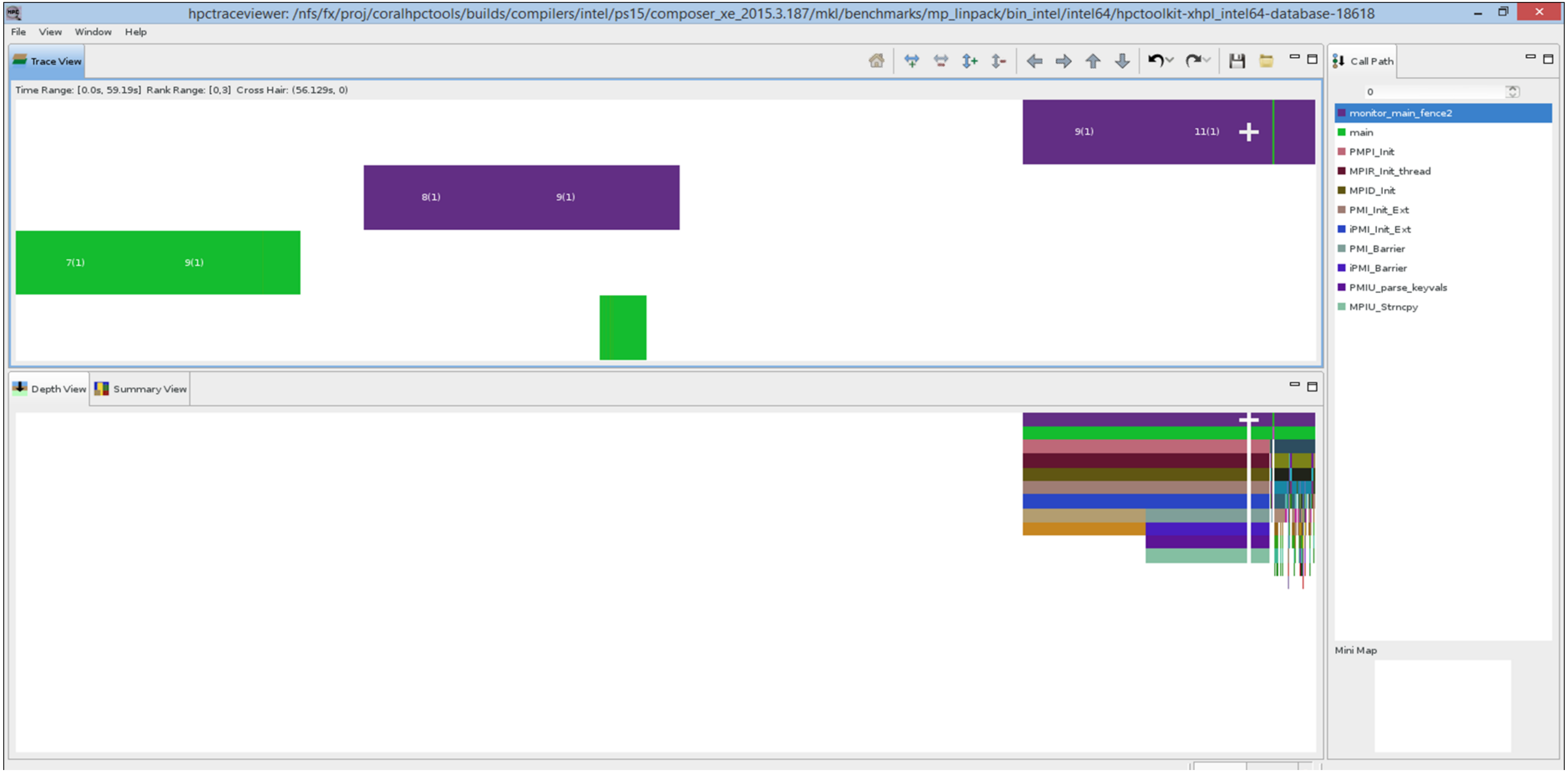
Scope	PAPI_TOT_CYC:Sum (I)	PAPI_TOT_CYC:Sum (E)	CPUTIME (usec):Sum (I)	CPUTIME (usec):Sum (E)
Experiment Aggregate Metrics	4.70e+06 100 %	4.70e+06 100 %		
[f: 231] monitor_main_fence2	2.43e+06 51.7%			
[f: 6] main	1.61e+06 34.3%	1.00e+04 0.2%		
[f: 207] Partial Call Paths	6.40e+05 13.6%			
[f: 225] monitor_main_fence1	2.00e+04 0.4%			

HPCToolKit - Flat View for xhpl_intel64 Backup



Scope	PAPI_TOT_CYC:Sum (I)	PAPI_TOT_CYC:Sum (E)	PAPI_L2_TCM:Sum (I)	PAPI_L2_TCM:Sum (E)	CPUTIME (usec):Sum (I)	CPUTIME (usec):Sum (E)
Experiment Aggregate Metrics	3.13e+09 100 %	3.13e+09 100 %	3.05e+07 100 %	3.05e+07 100 %	2.89e+07 100 %	2.89e+07 100 %
[c:2] [f: 2] Load module /nfs/fx/proj/c	3.03e+09 96.7%	1.95e+09 62.5%	3.05e+07 99.9%	7.85e+06 25.7%	3.44e+06 11.9%	7.12e+05 2.5%
[c:5617] [f: 5617] ~unknown-file~	3.03e+09 96.7%	1.95e+09 62.3%	3.05e+07 99.9%	7.69e+06 25.2%	3.44e+06 11.9%	6.11e+05 2.1%
[c:4808] [f: 4808] socksm.c	1.33e+07 0.4%	1.85e+06 0.1%	1.07e+06 3.5%	1.60e+05 0.5%	1.01e+05 0.3%	1.01e+05 0.3%
[c:595] [f: 595] ch3u_request.c	5.20e+06 0.2%	1.00e+04 0.0%	4.30e+05 1.4%			
[c:4674] [f: 4674] segment_packi	4.62e+06 0.1%	1.00e+04 0.0%	3.90e+05 1.3%			
[c:3194] [f: 3194] initthread.c	3.29e+06 0.1%	1.90e+05 0.0%	3.00e+04 0.1%			
[c:5402] [f: 5402] HPL_dlamch.c	1.47e+06 0.0%	1.47e+06 0.0%				
[c:3] [f: 3] _gemm_buffers.c	3.50e+05 0.0%	3.50e+05 0.0%				
[c:3891] [f: 3891] mpid_nem_init.	2.50e+05 0.0%	2.00e+04 0.0%				
[c:14] [f: 14] _xgemm.c	2.10e+05 0.0%	2.10e+05 0.0%				
[c:4099] [f: 4099] mpidi_pg.c	1.10e+05 0.0%					
[c:4088] [f: 4088] mpid_segment	9.00e+04 0.0%	9.00e+04 0.0%				
[c:3698] [f: 3698] mpid_datatype	6.00e+04 0.0%	1.00e+04 0.0%				
[c:5511] [f: 5511] proc_init_utils.c	6.00e+04 0.0%	6.00e+04 0.0%				
[c:3030] [f: 3030] dataloop.c	4.00e+04 0.0%					
[c:4678] [f: 4678] simple_pmi.c	4.00e+04 0.0%					
[c:6] [f: 6] _gemm_strategy.c	2.00e+04 0.0%	2.00e+04 0.0%				
[c:3078] [f: 3078] handlemem.c	2.00e+04 0.0%	2.00e+04 0.0%				
[c:5386] [f: 5386] typeutil.c	2.00e+04 0.0%					
[c:553] [f: 553] ch3u_handle_recv	1.00e+04 0.0%					
[c:39045] [f: 39045] Load module /n	1.65e+09 52.7%	7.00e+04 0.0%	1.49e+07 48.8%		2.73e+07 94.4%	2.49e+07 86.0%
[c:39082] [f: 39082] Load module /u	5.95e+08 19.0%	5.95e+08 19.0%	2.70e+05 0.9%	2.60e+05 0.9%	3.03e+06 10.5%	3.03e+06 10.5%
[c:39039] [f: 39039] Load module ~	1.03e+08 3.3%	1.03e+08 3.3%	2.00e+04 0.1%	2.00e+04 0.1%	2.55e+07 88.1%	1.02e+05 0.4%
[c:39065] [f: 39065] Load module /u	3.79e+06 0.1%	3.59e+06 0.1%	1.00e+05 0.3%	1.00e+05 0.3%	2.54e+07 87.8%	
[c:39303] [f: 39303] Load module /u	1.30e+05 0.0%	1.30e+05 0.0%				
[c:39318] [f: 39318] Load module /n					5.06e+05 1.7%	

HPCToolKit - Depth View for xhpl_intel64 Backup



hpctraceviewer: /nfs/fx/proj/coralhpctools/builds/compilers/intel/ps15/composer_xe_2015.3.187/mkl/benchmarks/mp_linpack/bin_intel/intel64/hpctoolkit-xhpl_intel64-database-18618

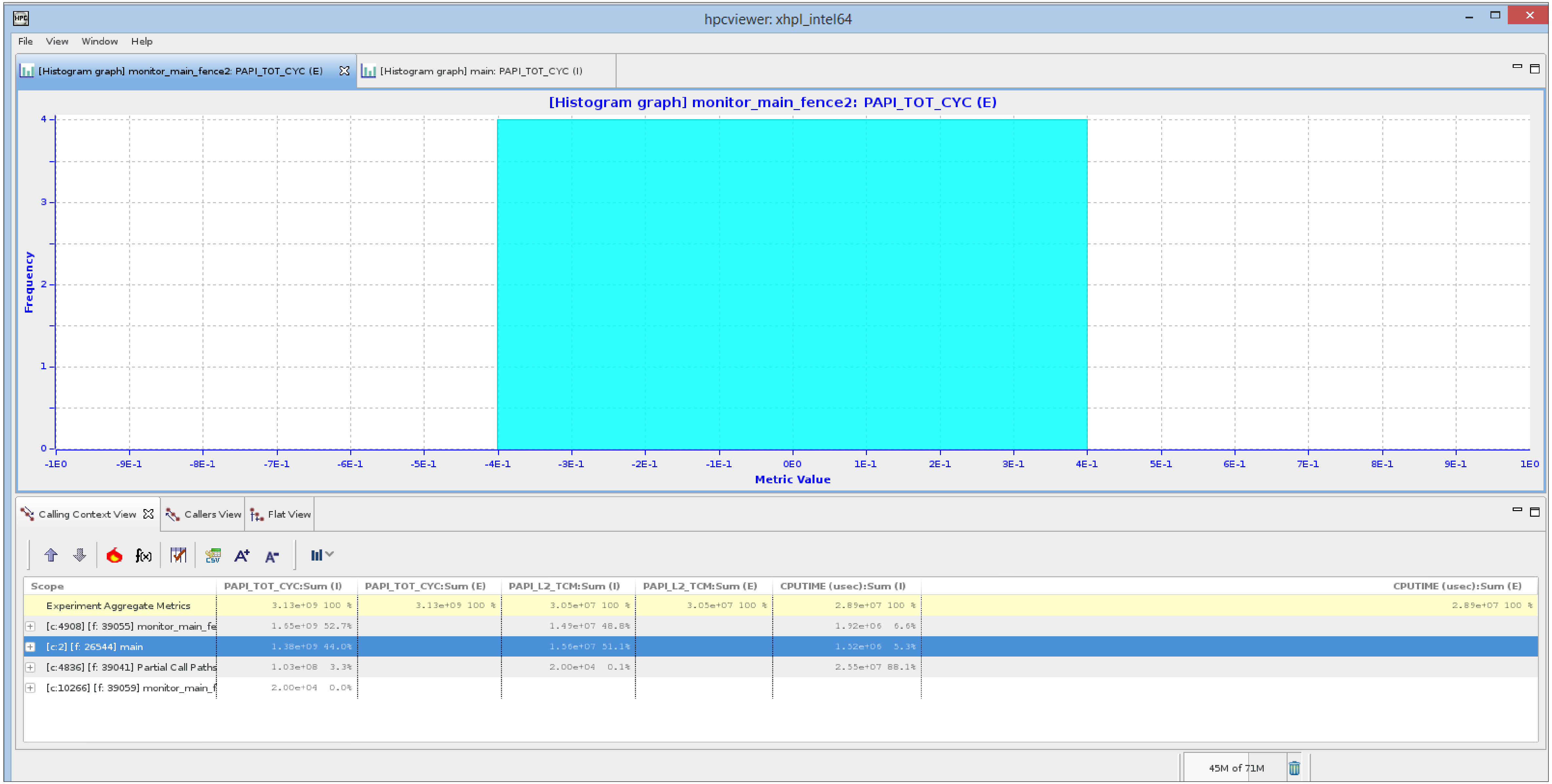
Time Range: [0.0s, 59.19s] Rank Range: [0,3] Cross Hair: (56.129s, 0)

Call Path

0

- 🟪 monitor_main_fence2
- 🟩 main
- 🟥 PMPI_Init
- 🟥 MPIR_Init_thread
- 🟫 MPID_Init
- 🟫 PMI_Init_Ext
- 🟦 iPMI_Init_Ext
- ▨ PMI_Barrier
- 🟦 iPMI_Barrier
- 🟪 PMIU_parse_keyvals
- 🟩 MPIU_Strncpy

HPCToolKit - Histogram for xhpl_intel64 Backup



Scope	PAPI_TOT_CYC:Sum (I)	PAPI_TOT_CYC:Sum (E)	PAPI_L2_TCM:Sum (I)	PAPI_L2_TCM:Sum (E)	CPUTIME (usec):Sum (I)	CPUTIME (usec):Sum (E)
Experiment Aggregate Metrics	3.13e+09 100 %	3.13e+09 100 %	3.05e+07 100 %	3.05e+07 100 %	2.89e+07 100 %	2.89e+07 100 %
[c:4908] [f: 39055] monitor_main_fe	1.65e+09 52.7%		1.49e+07 48.8%		1.92e+06 6.6%	
[c:2] [f: 26544] main	1.38e+09 44.0%		1.56e+07 51.1%		1.52e+06 5.3%	
[c:4836] [f: 39041] Partial Call Paths	1.03e+08 3.3%		2.00e+04 0.1%		2.55e+07 88.1%	
[c:10266] [f: 39059] monitor_main_f	2.00e+04 0.0%					