AMDuProf guide

run and collect data

Run a batch job and collect data:

batch script commands

```
...
#SBATCH --constraint=perf  # perf,nvperf for gpu nodes

cd $SLURM_SUBMIT_DIR
export PATH=/sw/external/amd/AMDuProf_Linux_x64_4.0.341/bin:$PATH

set -v
srun AMDuProfCLI collect --config tbp  -o `pwd`/uprof_tbp  `pwd`/stream.22gb
srun AMDuProfCLI collect --config inst_access  -o `pwd`/uprof_inst_access  `pwd`/stream.22gb
srun AMDuProfCLI collect --config assess  -o `pwd`/uprof_assess  `pwd`/stream.22gb
srun AMDuProfCLI collect --config assess_ext  -o `pwd`/uprof_assess_ext  `pwd`/stream.22gb
```

generate report

After collecting data from a batch job, generate reports with the AMDuProfCLI report option.

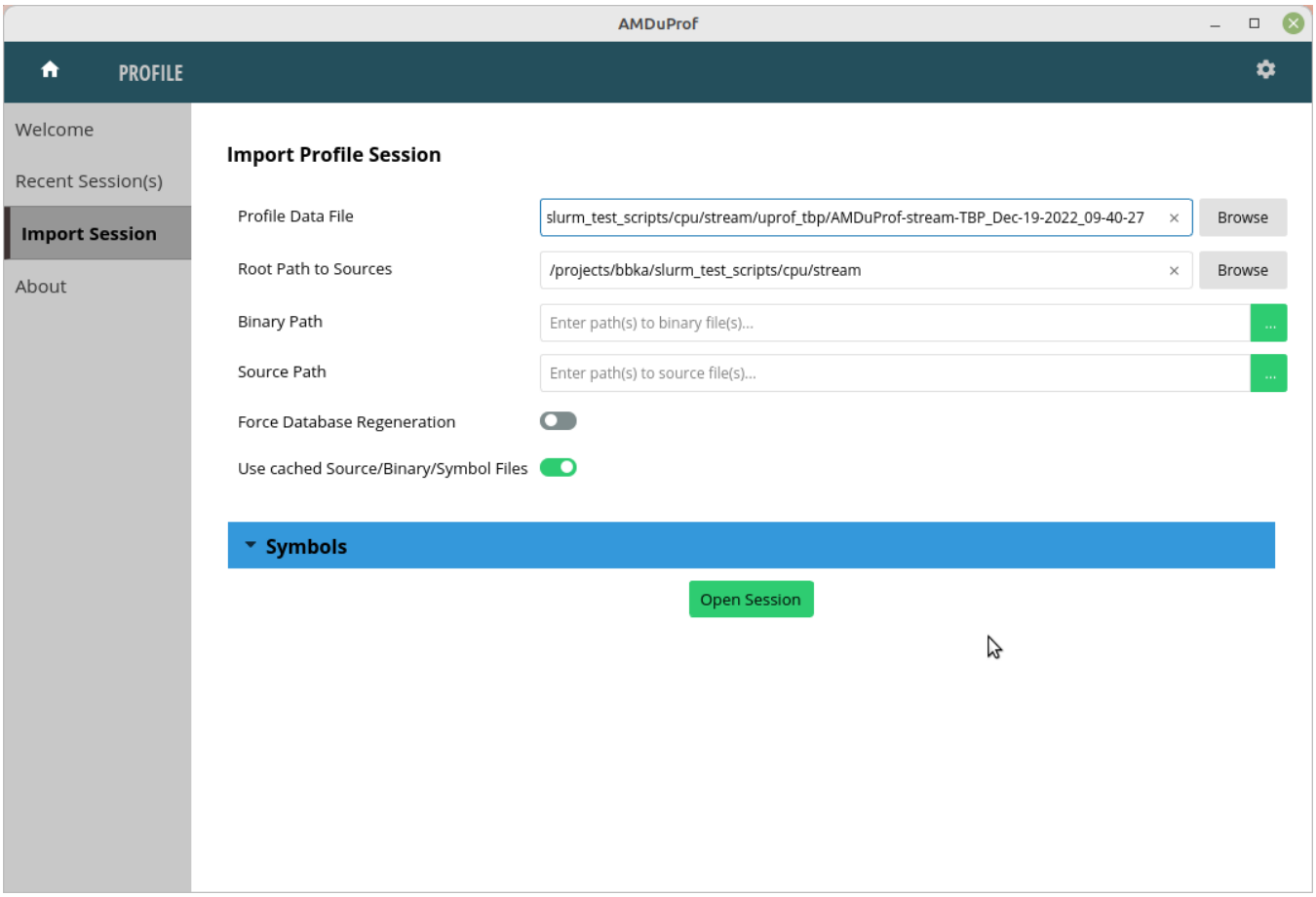
generate report

```
[arnoldg@dt-login03 uprof_tbp]$ export PATH=/sw/external/amd/AMDuProf_Linux_x64_4.0.341/bin:$PATH
[arnoldg@dt-login03 uprof_tbp]$ AMDuProfCLI report -i AMDuProf-stream-TBP_Dec-19-2022_09-40-27/
/sw/external/amd/AMDuProf_Linux_x64_4.0.341/bin/AMDuProfCLI
Translation started ...
Translation finished
Generated database file : cpu
Report generation started...
Generating report file...

Report generation completed...

Generated report file: /projects/bbka/slurm_test_scripts/cpu/stream/uprof_tbp/AMDuProf-stream-TBP_Dec-19-
2022_09-40-27/report.csv
```

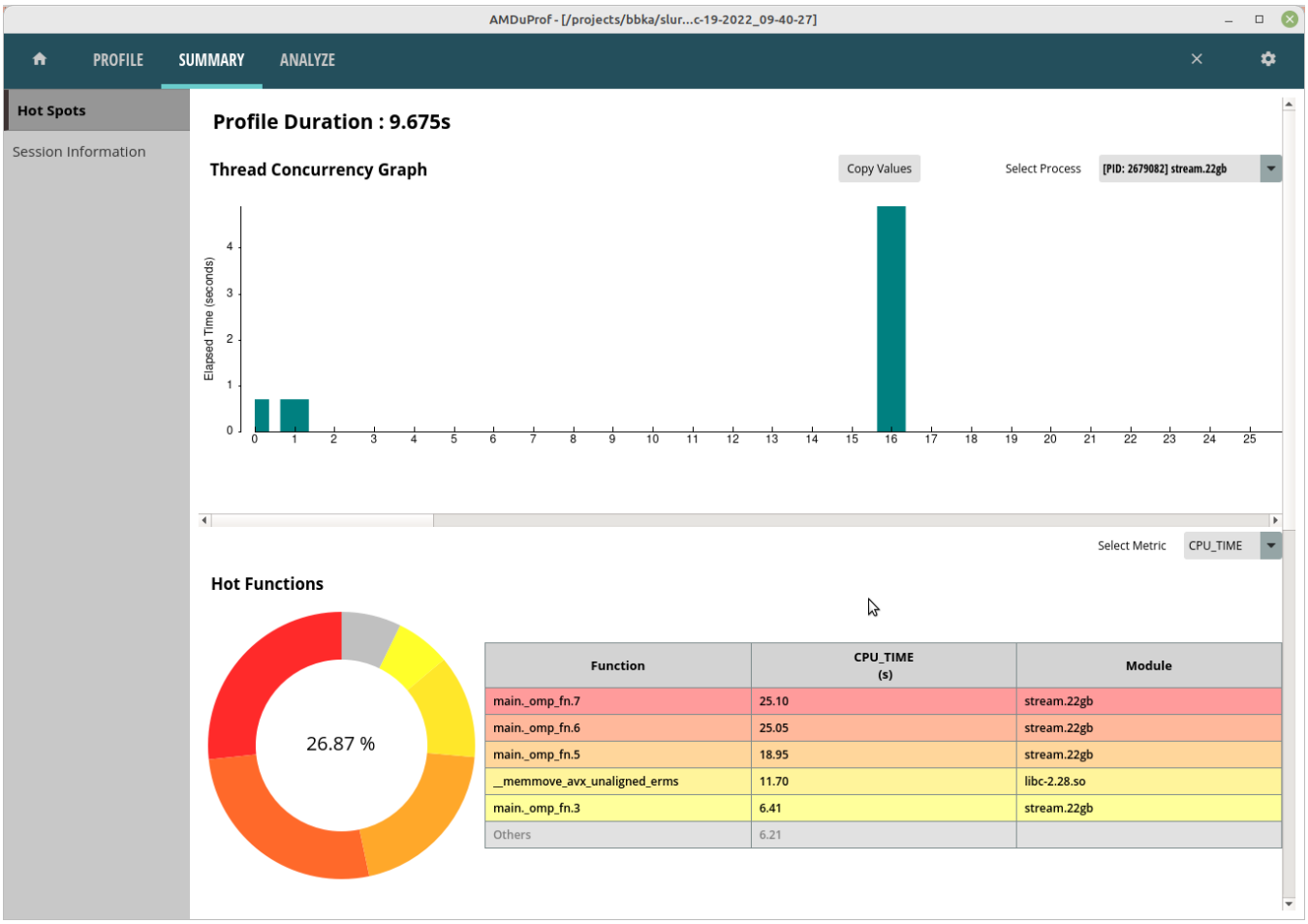
visualize and explore report data

You can view the data in AMDuProf on Delta or locally on a copy you install at your desktop system. If you install locally, you may need to replicate some paths or add paths to the binary in order to get full functionality.

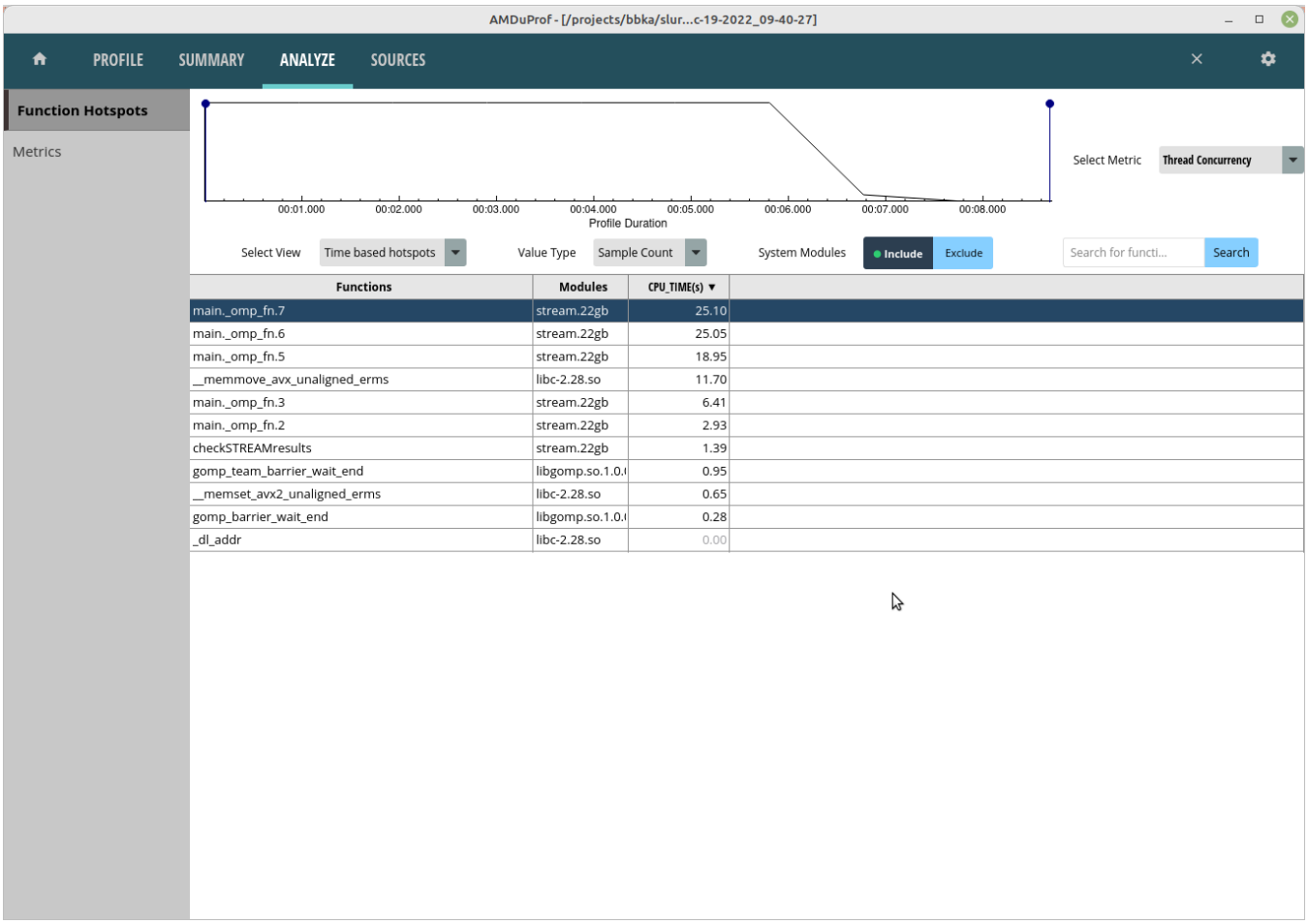
Launch AMDuProf (no CLI suffice for the GUI), and import the profile session from a completed batch job run with AMDuProfCLI collect ...



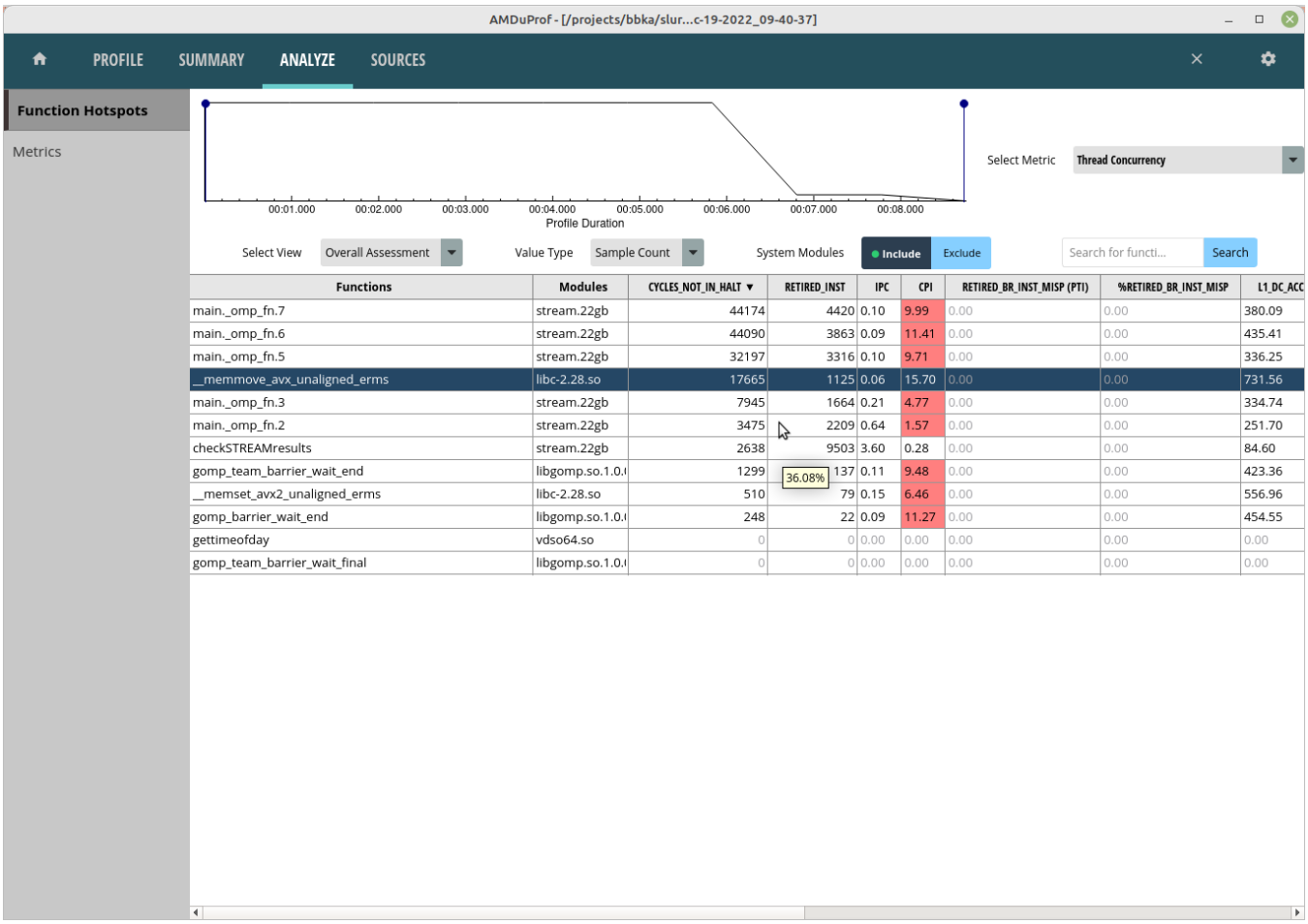
The summary view give a high level overview of how time was spent. This is the tbp time-based-profile summary.



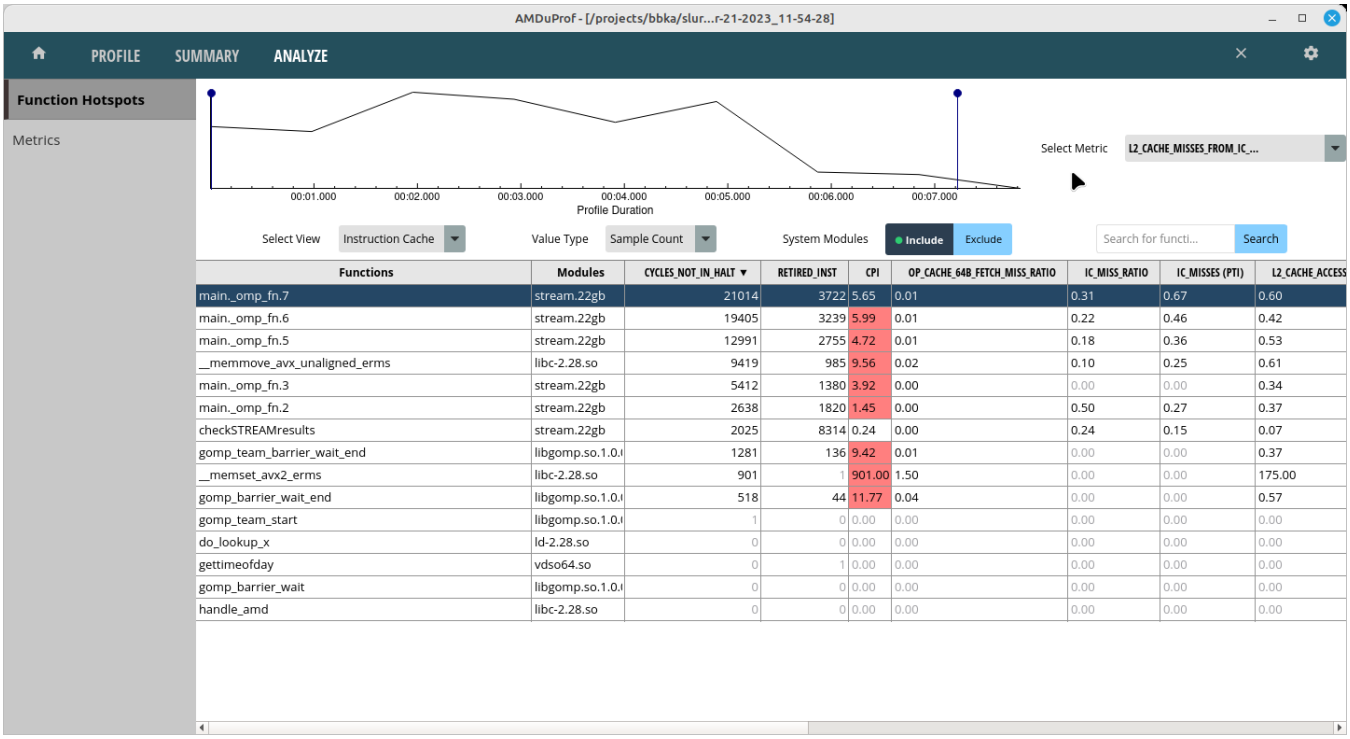
The Analyze tab shows hot routines or lines in more detail. The tbp , assess, and inst_access Analyze views follow.



Functions	Modules	CPU_TIME(s) ▼
main._omp_fn.7	stream.22gb	25.10
main._omp_fn.6	stream.22gb	25.05
main._omp_fn.5	stream.22gb	18.95
__memmove_avx_unaligned_erms	libc-2.28.so	11.70
main._omp_fn.3	stream.22gb	6.41
main._omp_fn.2	stream.22gb	2.93
checkSTREAMresults	stream.22gb	1.39
gomp_team_barrier_wait_end	libgomp.so.1.0.(	0.95
__memset_avx2_unaligned_erms	libc-2.28.so	0.65
gomp_barrier_wait_end	libgomp.so.1.0.(	0.28
_dl_addr	libc-2.28.so	0.00

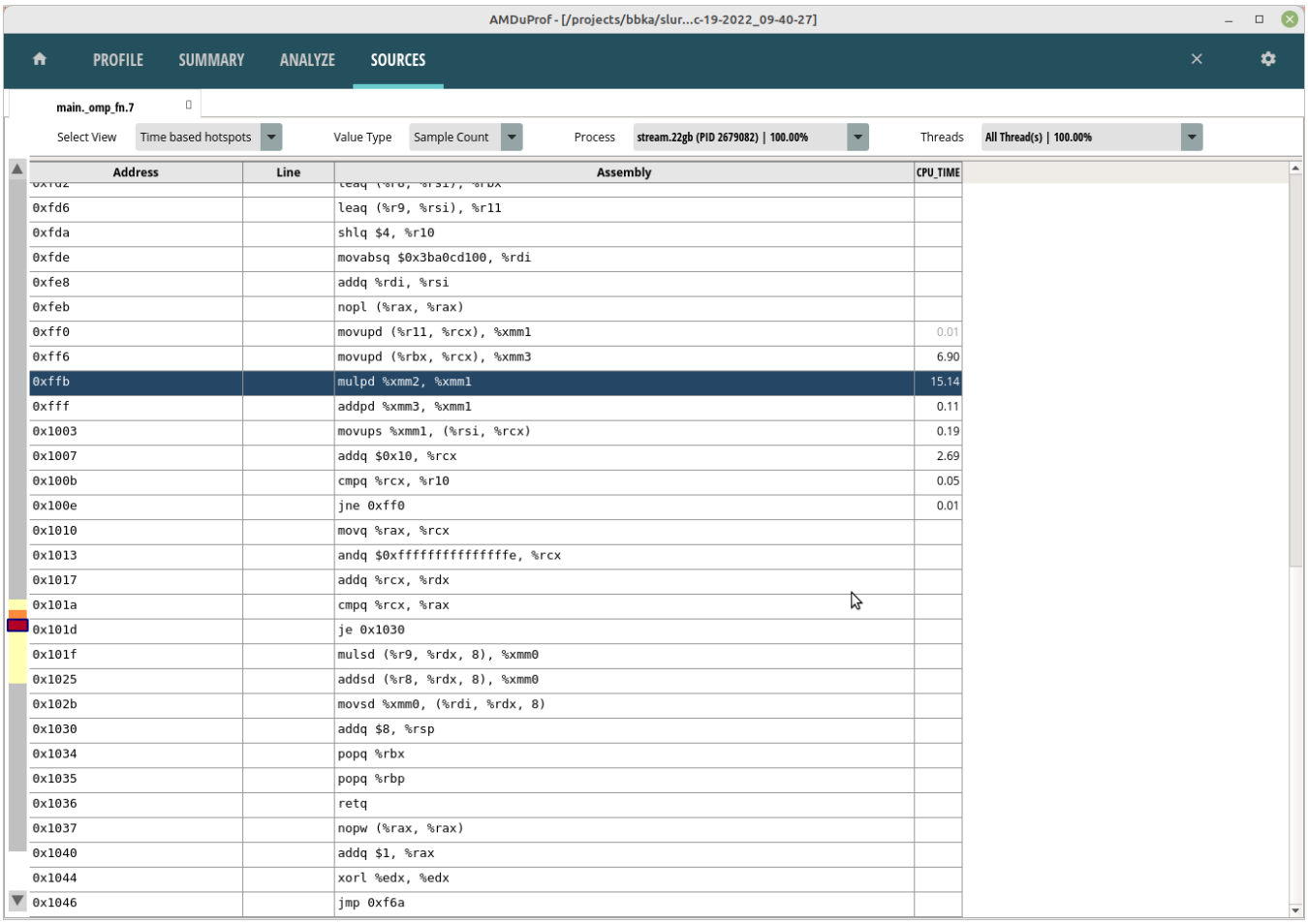


Functions	Modules	CYCLES_NOT_IN_HALT ▼	RETIRED_INST	IPC	CPI	RETIRED_BR_INST_MISP (PTI)	%RETIRED_BR_INST_MISP	L1_DC_ACC
main._omp_fn.7	stream.22gb	44174	4420	0.10	9.99	0.00	0.00	380.09
main._omp_fn.6	stream.22gb	44090	3863	0.09	11.41	0.00	0.00	435.41
main._omp_fn.5	stream.22gb	32197	3316	0.10	9.71	0.00	0.00	336.25
__memmove_avx_unaligned_erms	libc-2.28.so	17665	1125	0.06	15.70	0.00	0.00	731.56
main._omp_fn.3	stream.22gb	7945	1664	0.21	4.77	0.00	0.00	334.74
main._omp_fn.2	stream.22gb	3475	2209	0.64	1.57	0.00	0.00	251.70
checkSTREAMresults	stream.22gb	2638	9503	3.60	0.28	0.00	0.00	84.60
gomp_team_barrier_wait_end	libgomp.so.1.0.(	1299	137	0.11	9.48	0.00	0.00	423.36
__memset_avx2_unaligned_erms	libc-2.28.so	510	79	0.15	6.46	0.00	0.00	556.96
gomp_barrier_wait_end	libgomp.so.1.0.(	248	22	0.09	11.27	0.00	0.00	454.55
gettimeofday	vdso64.so	0	0	0.00	0.00	0.00	0.00	0.00
gomp_team_barrier_wait_final	libgomp.so.1.0.(	0	0	0.00	0.00	0.00	0.00	0.00



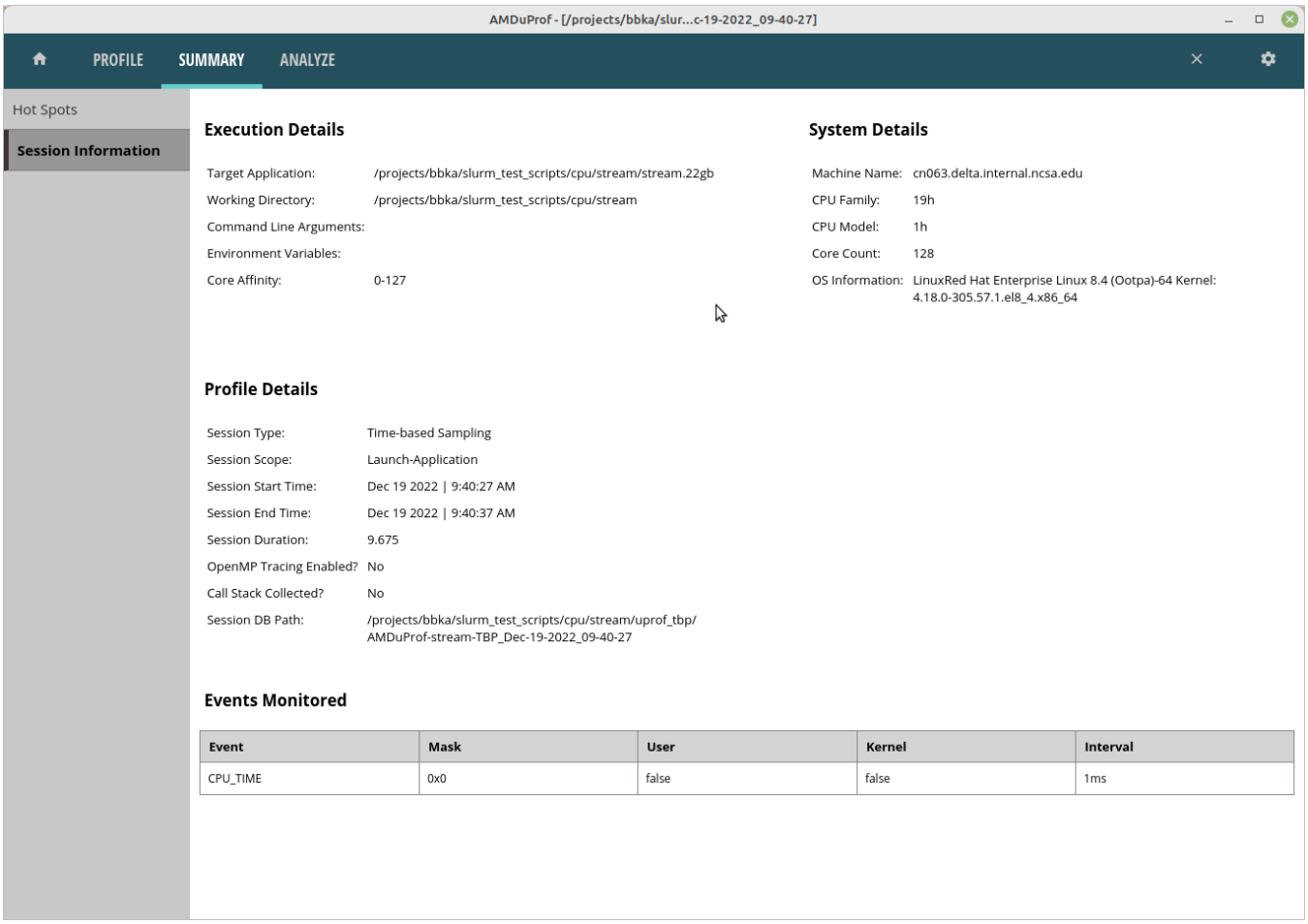
Functions	Modules	CYCLES_NOT_IN_HALT ▼	RETIRED_INST	CPI	OP_CACHE_64B_FETCH_MISS_RATIO	IC_MISS_RATIO	IC_MISSES (PTI)	L2_CACHE_ACCESS
main._omp_fn.7	stream.22gb	21014	3722	5.65	0.01	0.31	0.67	0.60
main._omp_fn.6	stream.22gb	19405	3239	5.99	0.01	0.22	0.46	0.42
main._omp_fn.5	stream.22gb	12991	2755	4.72	0.01	0.18	0.36	0.53
__memmove_avx_unaligned_erms	libc-2.28.so	9419	985	9.56	0.02	0.10	0.25	0.61
main._omp_fn.3	stream.22gb	5412	1380	3.92	0.00	0.00	0.00	0.34
main._omp_fn.2	stream.22gb	2638	1820	1.45	0.00	0.50	0.27	0.37
checkSTREAMresults	stream.22gb	2025	8314	0.24	0.00	0.24	0.15	0.07
gomp_team_barrier_wait_end	libgomp.so.1.0.(	1281	136	9.42	0.01	0.00	0.00	0.37
__memset_avx2_erms	libc-2.28.so	901	1	901.00	1.50	0.00	0.00	175.00
gomp_barrier_wait_end	libgomp.so.1.0.(	518	44	11.77	0.04	0.00	0.00	0.57
gomp_team_start	libgomp.so.1.0.(	1	0	0.00	0.00	0.00	0.00	0.00
do_lookup_x	ld-2.28.so	0	0	0.00	0.00	0.00	0.00	0.00
gettimeofday	vdso64.so	0	1	0.00	0.00	0.00	0.00	0.00
gomp_barrier_wait	libgomp.so.1.0.(	0	0	0.00	0.00	0.00	0.00	0.00
handle_amd	libc-2.28.so	0	0	0.00	0.00	0.00	0.00	0.00

Selecting one of the lines or routines will take you to the Sources view where you can see the assembly used in that portion of code.



Address	Line	Assembly	CPU_TIME
0xfd2		leaq (%r8, %rsi), %rbx	
0xfd6		leaq (%r9, %rsi), %r11	
0xfda		shlq $4, %r10	
0xfde		movabsq $0x3ba0cd100, %rdi	
0xfe8		addq %rdi, %rsi	
0xfeb		nopl (%rax, %rax)	
0xff0		movupd (%r11, %rcx), %xmm1	0.01
0xff6		movupd (%rbx, %rcx), %xmm3	6.90
0xffb		mulpd %xmm2, %xmm1	15.14
0xfff		addpd %xmm3, %xmm1	0.11
0x1003		movups %xmm1, (%rsi, %rcx)	0.19
0x1007		addq $0x10, %rcx	2.69
0x100b		cmpq %rcx, %r10	0.05
0x100e		jne 0xff0	0.01
0x1010		movq %rax, %rcx	
0x1013		andq $0xfffffffffffffffe, %rcx	
0x1017		addq %rcx, %rdx	
0x101a		cmpq %rcx, %rax	
0x101d		je 0x1030	
0x101f		mulsd (%r9, %rdx, 8), %xmm0	
0x1025		addsd (%r8, %rdx, 8), %xmm0	
0x102b		movsd %xmm0, (%rdi, %rdx, 8)	
0x1030		addq $8, %rsp	
0x1034		popq %rbx	
0x1035		popq %rbp	
0x1036		retq	
0x1037		nopw (%rax, %rax)	
0x1040		addq $1, %rax	
0x1044		xorl %edx, %edx	
0x1046		jmp 0xf6a	

The Session Info is under the Summary tab and displays more detail about the profiling session.



References:

https://www.amd.com/content/dam/amd/en/documents/developer/uprof-v4.0-gaGA-user-guide.pdf