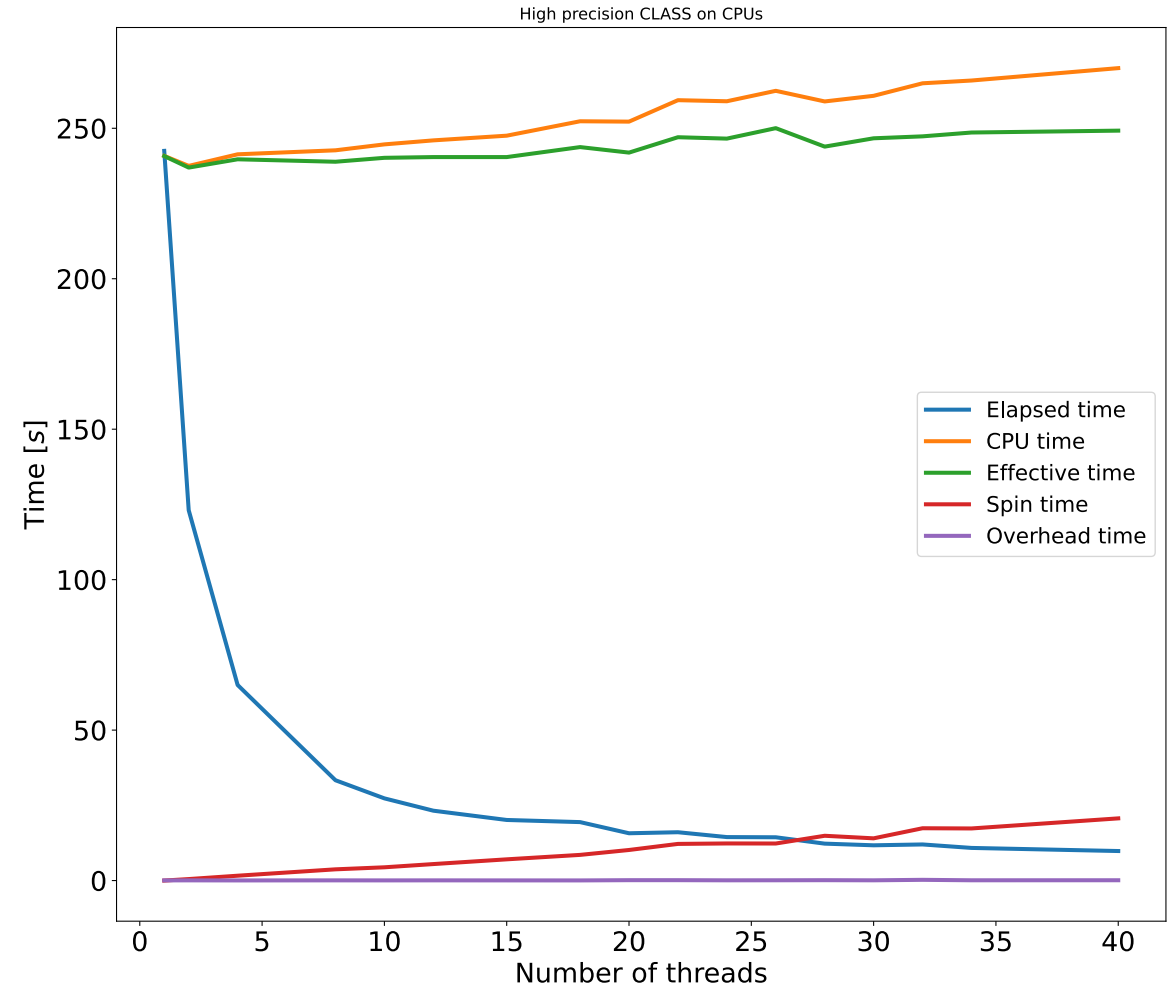# CLASS code profiling on CPUs

- **Elapsed time:** wall time from the beginning to the end of collection
- **CPU time:** time during which the CPU is actively executing your application
- **Effective time:** CPU time spent in the user code. Does not include Spin and Overhead time.
- **Spin time:** Wait Time during which the CPU is busy
- **Overhead time:** CPU time spent on the overhead of known synchronization and threading libraries, such as OpenMP
- **CPU time = Effective time + Spin time + Overhead**

High precision CLASS on CPUs

Profiling of the CLASS code using VTune tool

Hotspots ⓘ

Analysis Configuration   Collection Log   Summary   Bottom-up   Caller/Callee   **Top-down Tree**   Flame Graph   Platform

Grouping: Call Stack

Function Stack	CPU Time: Total ▾ »	CPU Time: Self »	Module	Function (Full)	Source File
▼ Total	100.0%	0s			
▼ _start	97.2%	0s	class	_start	start.S
▼ __libc_start_main	97.2%	0s	libc.so.6	__libc_start_main	
▼ main	97.2%	0s	class	main	class.c
▶ perturbations_init	51.4%	0s	class	perturbations_init	perturbations.c
▶ transfer_init	43.4%	0s	class	transfer_init	transfer.c
▶ lensing_init	1.1%	0s	class	lensing_init	lensing.c
▶ fourier_init	1.0%	0.012s	class	fourier_init	fourier.c
▶ harmonic_init	0.4%	0s	class	harmonic_init	harmonic.c
▶ output_init	0.0%	0s	class	output_init	output.c
▶ thermodynamics_init	0.0%	0s	class	thermodynamics_init	thermodynamics.c
▶ input_init	0.0%	0s	class	input_init	input.c
▶ background_init	0.0%	0s	class	background_init	background.c
▶ __clone	2.4%	0s	libc.so.6	__clone	
▶ _INTERNAL1311483b::__kmp_wait_template<kmp_flag_64<(bool)0, (bool)1>, (bool)1, (bool)0, (b	0.3%	0.784s	libiomp5.so	_INTERNAL1311483b::__kmp_wait_template<kmp_fla...	kmp_wait_release.h
▶ kmp_flag_native<unsigned long long, (flag_type)1, (bool)1>::notdone_check	0.0%	0.088s	libiomp5.so	kmp_flag_native<unsigned long long, (bo...	kmp_wait_release.h
▶ sched_yield	0.0%	0.056s	libc.so.6	sched_yield	

Profiling of the CLASS code using VTune tool

- **Specifically in the transfer module we find:**

Hotspots ⊘

Analysis Configuration   Collection Log   Summary   Bottom-up   Caller/Callee   **Top-down Tree**   Flame Graph   Platform

Grouping: Call Stack

Function Stack	CPU Time: Total ▾	CPU Time: Self	Module	Function (Full)	Source File
▾ Total	100.0%	0s			
▾ _start	97.2%	0s	class	_start	start.S
▾ __libc_start_main	97.2%	0s	libc.so.6	__libc_start_main	
▾ main	97.2%	0s	class	main	class.c
▸ perturbations_init	51.4%	0s	class	perturbations_init	perturbations.c
▾ transfer_init	43.4%	0s	class	transfer_init	transfer.c
▾ [OpenMP fork]	42.6%	0s	libiomp5.so	__kmpc_fork_call	kmp.h
▾ __kmp_fork_call	42.6%	0s	libiomp5.so	__kmp_fork_call	kmp_runtime.cpp
▾ [OpenMP dispatcher]	42.6%	0s	libiomp5.so	__kmp_invoke_task_func	kmp_runtime.cpp
▾ transfer_initompparallel@316	42.6%	0s	class	transfer_initompparallel@316	transfer.c
▾ transfer_compute_for_each_q	42.6%	0.320s	class	transfer_compute_for_each_q	transfer.c
▾ transfer_compute_for_each_l	41.6%	0.528s	class	transfer_compute_for_each_l	transfer.c
▾ transfer_integrate	41.0%	1.852s	class	transfer_integrate	transfer.c
▾ transfer_radial_function	38.5%	8.984s	class	transfer_radial_function	transfer.c
hyperspherical_Hermite4_interpolation_vector_Phi	15.8%	39.208s	class	hyperspherical_Hermite4_interpolation_vector_Phi	hyperspherical.c
hyperspherical_Hermite4_interpolation_vector_Phid2Phi	10.9%	26.912s	class	hyperspherical_Hermite4_interpolation_vector_Phid2Phi	hyperspherical.c
hyperspherical_Hermite4_interpolation_vector_dPhi	6.5%	16.003s	class	hyperspherical_Hermite4_interpolation_vector_dPhi	hyperspherical.c