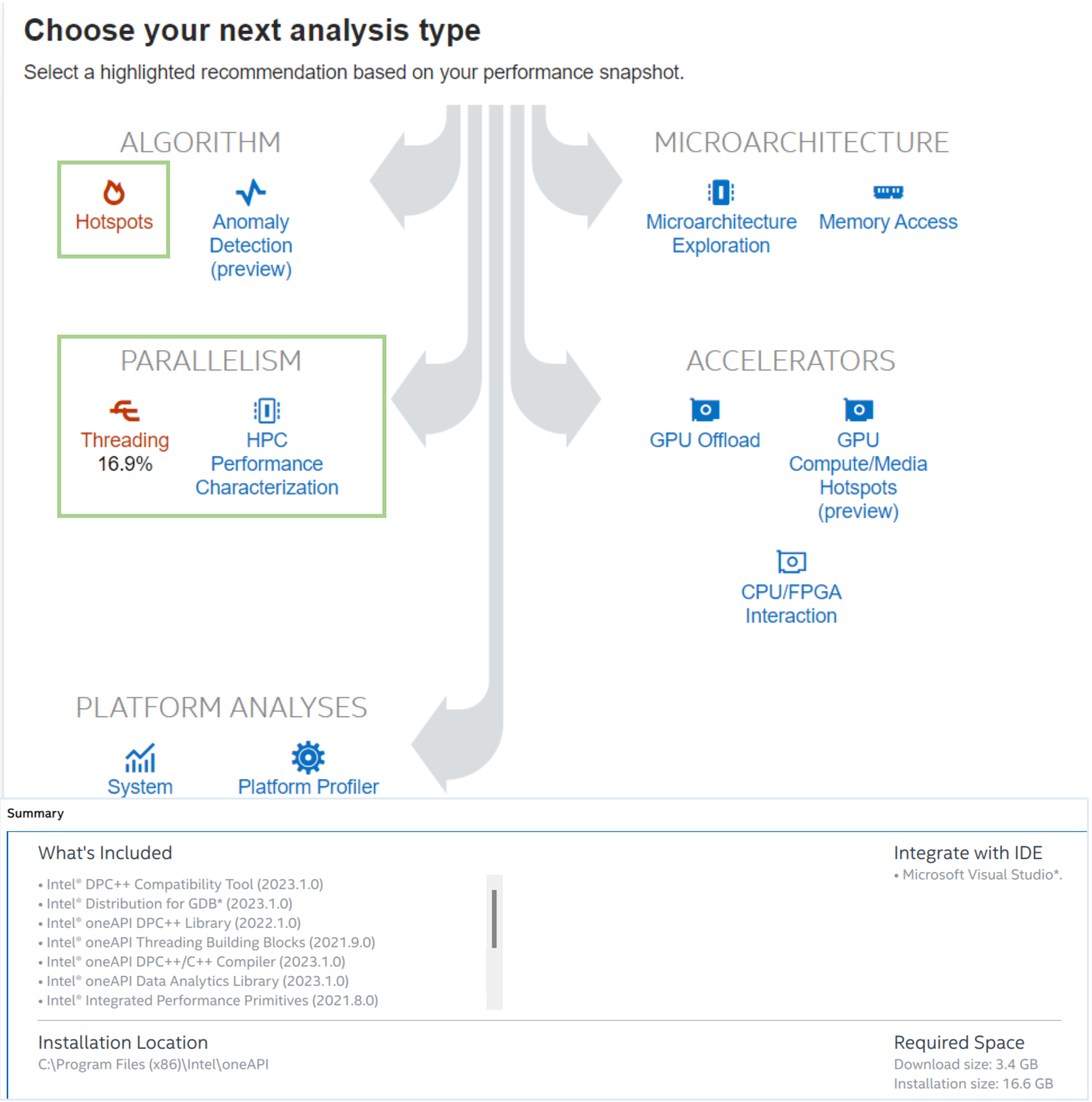
Intel VTune Profiler



Choose your next analysis type

Select a highlighted recommendation based on your performance snapshot.

ALGORITHM
- Hotspots
- Anomaly Detection (preview)

MICROARCHITECTURE
- Microarchitecture Exploration
- Memory Access

PARALLELISM
- Threading 16.9%
- HPC Performance Characterization

ACCELERATORS
- GPU Offload
- GPU Compute/Media Hotspots (preview)
- CPU/FPGA Interaction

PLATFORM ANALYSES
- System
- Platform Profiler

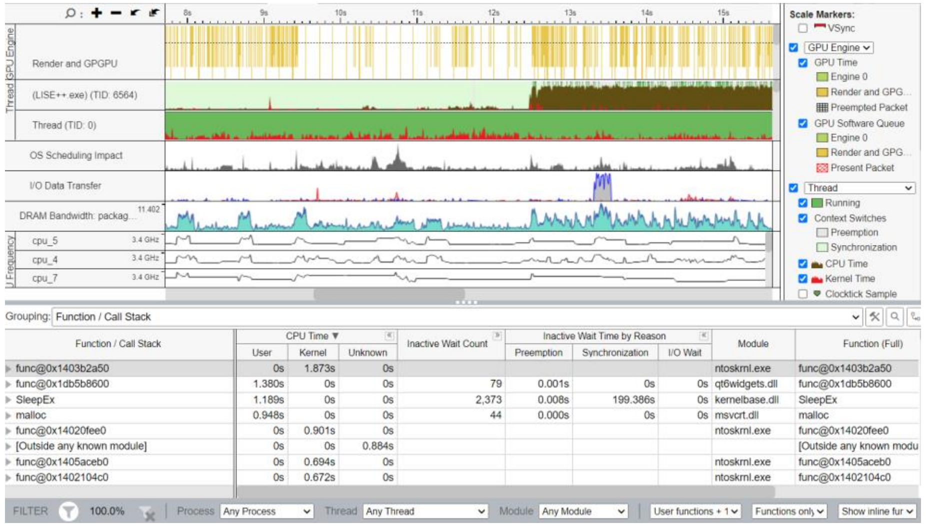
Summary

What's Included
- Intel® DPC++ Compatibility Tool (2023.1.0)
- Intel® Distribution for GDB* (2023.1.0)
- Intel® oneAPI DPC++ Library (2022.1.0)
- Intel® oneAPI Threading Building Blocks (2021.9.0)
- Intel® oneAPI DPC++/C++ Compiler (2023.1.0)
- Intel® oneAPI Data Analytics Library (2023.1.0)
- Intel® Integrated Performance Primitives (2021.8.0)

Integrate with IDE
- Microsoft Visual Studio*.

Installation Location
C:\Program Files (x86)\Intel\oneAPI

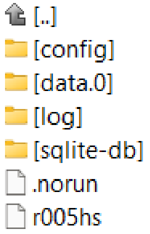
Required Space
Download size: 3.4 GB
Installation size: 16.6 GB



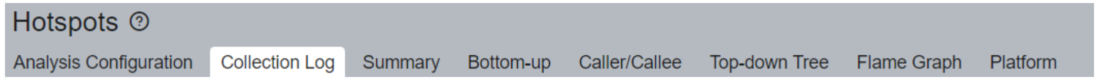
Grouping: Function / Call Stack

Function / Call Stack	CPU Time ▼			Inactive Wait Count	Inactive Wait Time by Reason			Module	Function (Full)
	User	Kernel	Unknown		Preemption	Synchronization	I/O Wait		
func@0x1403b2a50	0s	1.873s	0s					ntoskrnl.exe	func@0x1403b2a50
func@0x1db5b8600	1.380s	0s	0s	79	0.001s	0s	0s	qt6widgets.dll	func@0x1db5b8600
SleepEx	1.189s	0s	0s	2,373	0.008s	199.386s	0s	kernelbase.dll	SleepEx
malloc	0.948s	0s	0s	44	0.000s	0s	0s	msvcrt.dll	malloc
func@0x14020fee0	0s	0.901s	0s					ntoskrnl.exe	func@0x14020fee0
[Outside any known module]	0s	0s	0.884s						[Outside any known modu
func@0x1405aceb0	0s	0.694s	0s					ntoskrnl.exe	func@0x1405aceb0
func@0x1402104c0	0s	0.672s	0s					ntoskrnl.exe	func@0x1402104c0

FILTER 100.0% Process Any Process Thread Any Thread Module Any Module User functions + 1 Functions only Show inline fur



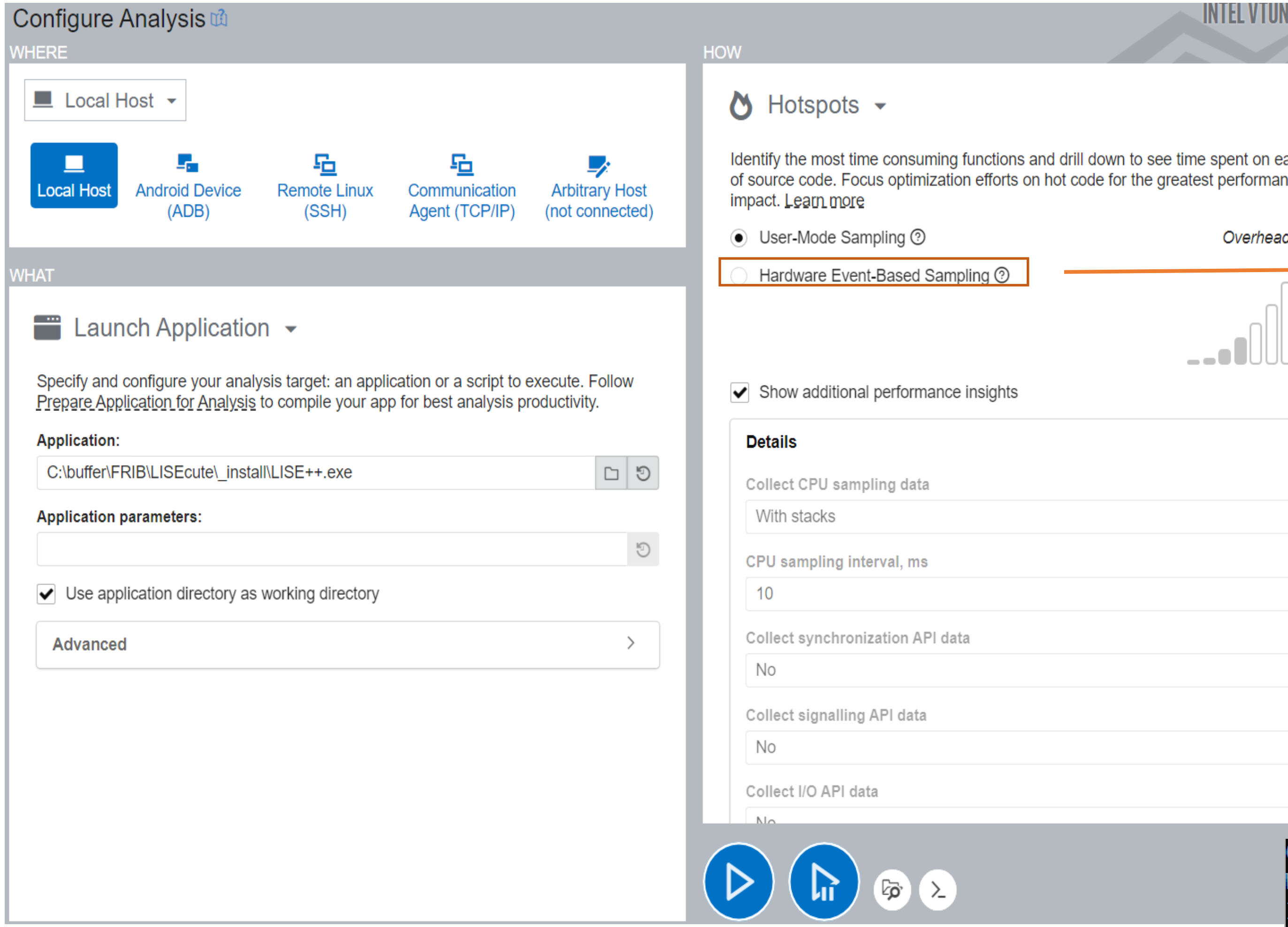
[..]
[config]
[data.0]
[log]
[sqlite-db]
.norun
r005hs



- Detailed Analysis
- Various profiling modes
- Multi-platform support (host/remote configuration)
- Graphs and timeline visualizations
- IDE Integration (MSVC installation option)

Parallelization potential!

Initial Hardware Setup



Configure Analysis

INTEL VTUNE PROFILER

WHERE

Local Host ▾

Local Host | Android Device (ADB) | Remote Linux (SSH) | Communication Agent (TCP/IP) | Arbitrary Host (not connected)

WHAT

Launch Application ▾

Specify and configure your analysis target: an application or a script to execute. Follow Prepare Application for Analysis to compile your app for best analysis productivity.

Application:

C:\buffer\FRIB\LISEcute_install\LISE++.exe

Application parameters:

☑ Use application directory as working directory

Advanced ›

HOW

Hotspots ▾

Identify the most time consuming functions and drill down to see time spent on each line of source code. Focus optimization efforts on hot code for the greatest performance impact. Learn more

● User-Mode Sampling

○ Hardware Event-Based Sampling

Overhead

☑ Show additional performance insights

Details

Collect CPU sampling data: With stacks

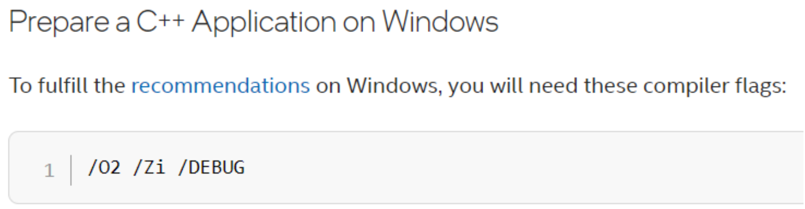
CPU sampling interval, ms: 10

Collect synchronization API data: No

Collect signalling API data: No

Collect I/O API data: No

- Run as administrator
- Install sampling drivers
- Add complier flags to .pro file



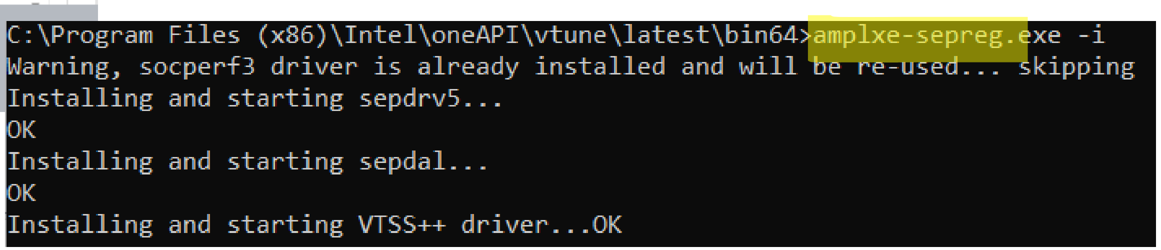
Prepare a C++ Application on Windows

To fulfill the recommendations on Windows, you will need these compiler flags:

```
/O2 /Zi /DEBUG
```

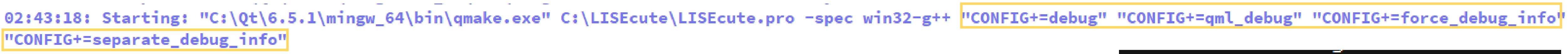
https://www.intel.com/content/www/us/en/docs/vtune-profiler/user-guide/2023-0/install-sampling-drivers-for-windows-targets.html

https://www.intel.com/content/www/us/en/developer/articles/code-sample/vtune-profiler-sampling-driver-downloads.html

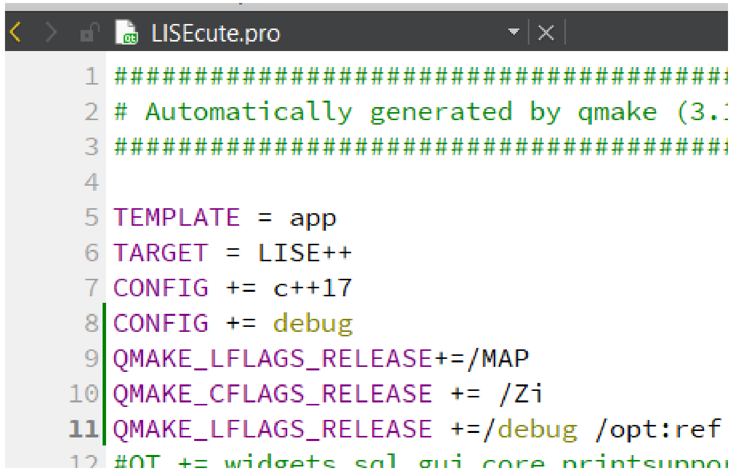


```
C:\Program Files (x86)\Intel\oneAPI\vtune\latest\bin64>amplxe-sepreg.exe -i
Warning, socperf3 driver is already installed and will be re-used... skipping
Installing and starting sepdrv5...
OK
Installing and starting sepdal...
OK
Installing and starting VTSS++ driver...OK
```

Intel VTune Source Code Instructions



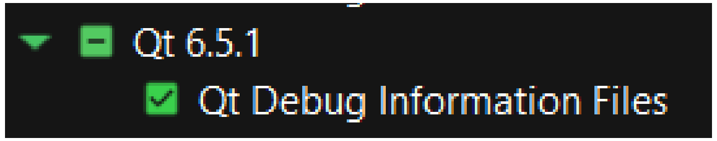
```
02:43:18: Starting: "C:\Qt\6.5.1\mingw_64\bin\qmake.exe" C:\LISEcute\LISEcute.pro -spec win32-g++ "CONFIG+=debug" "CONFIG+=qml_debug" "CONFIG+=force_debug_info"
"CONFIG+=separate_debug_info"
```



LISEcute.pro

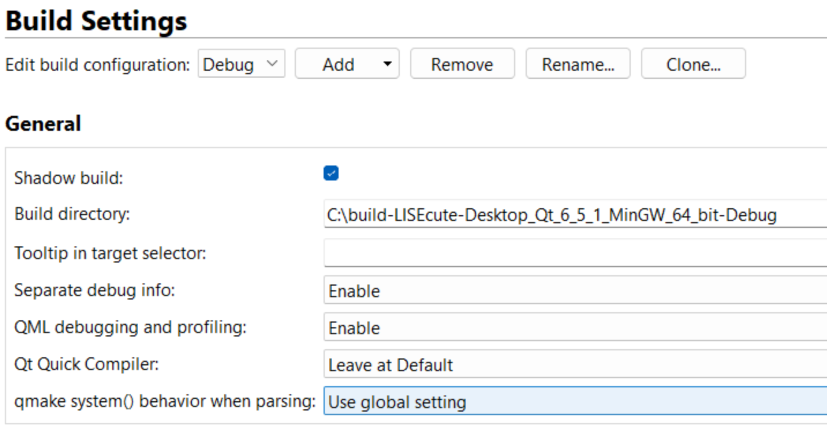
```
######################################
# Automatically generated by qmake (3.
######################################

TEMPLATE = app
TARGET = LISE++
CONFIG += c++17
CONFIG += debug
QMAKE_LFLAGS_RELEASE+=/MAP
QMAKE_CFLAGS_RELEASE += /Zi
QMAKE_LFLAGS_RELEASE +=/debug /opt:ref
#QT += widgets sql gui core printsuppo
```



- Qt 6.5.1
  - ☑ Qt Debug Information Files

1. Complier flags on .pro file
   - https://stackoverflow.com/questions/9234337/qt-no-map-pdb-files-generated-for-windows-release-builds



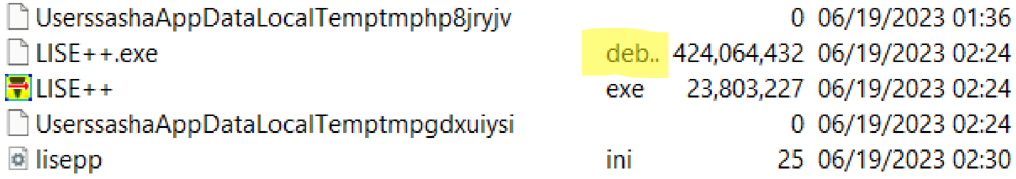
Build Settings

General

Shadow build:	☑
Build directory:	C:\build-LISEcute-Desktop_Qt_6_5_1_MinGW_64_bit-Debug
Tooltip in target selector:	
Separate debug info:	Enable
QML debugging and profiling:	Enable
Qt Quick Compiler:	Leave at Default
qmake system() behavior when parsing:	Use global setting

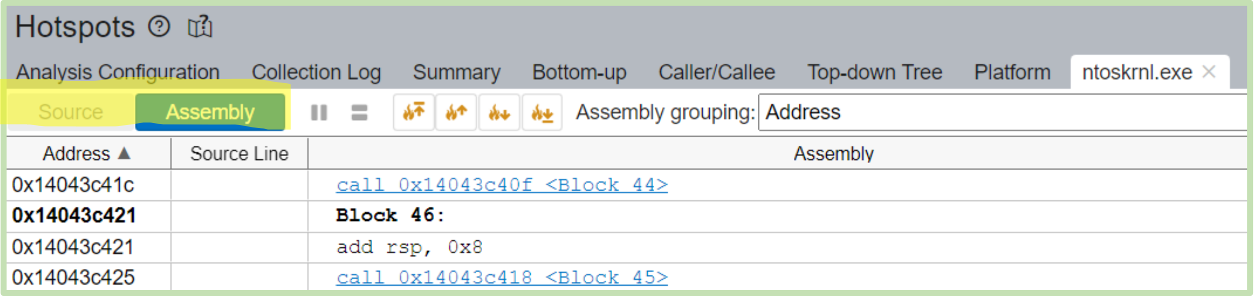
2. Check debug options in Projects > Build > General Build Settings for debug configuration

3. Build > Clean
4. Build > Run qmake
5. Debug to generate



UserssashaAppDataLocalTemptmphp8jryjv		0	06/19/2023 01:36
LISE++.exe	deb..	424,064,432	06/19/2023 02:24
LISE++	exe	23,803,227	06/19/2023 02:24
UserssashaAppDataLocalTemptmpgdxuiysi		0	06/19/2023 02:24
lisepp	ini	25	06/19/2023 02:30

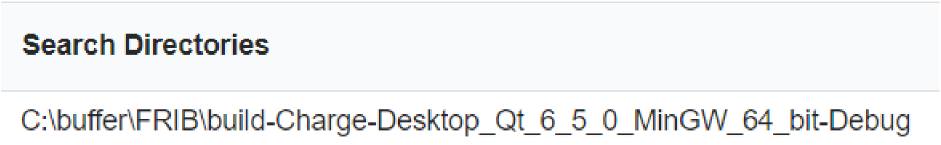
Intel VTune Search Directories



Assembly grouping: Address

Address ▲	Source Line	Assembly
0x14043c41c		call 0x14043c40f <Block 44>
0x14043c421		**Block 46:**
0x14043c421		add rsp, 0x8
0x14043c425		call 0x14043c418 <Block 45>

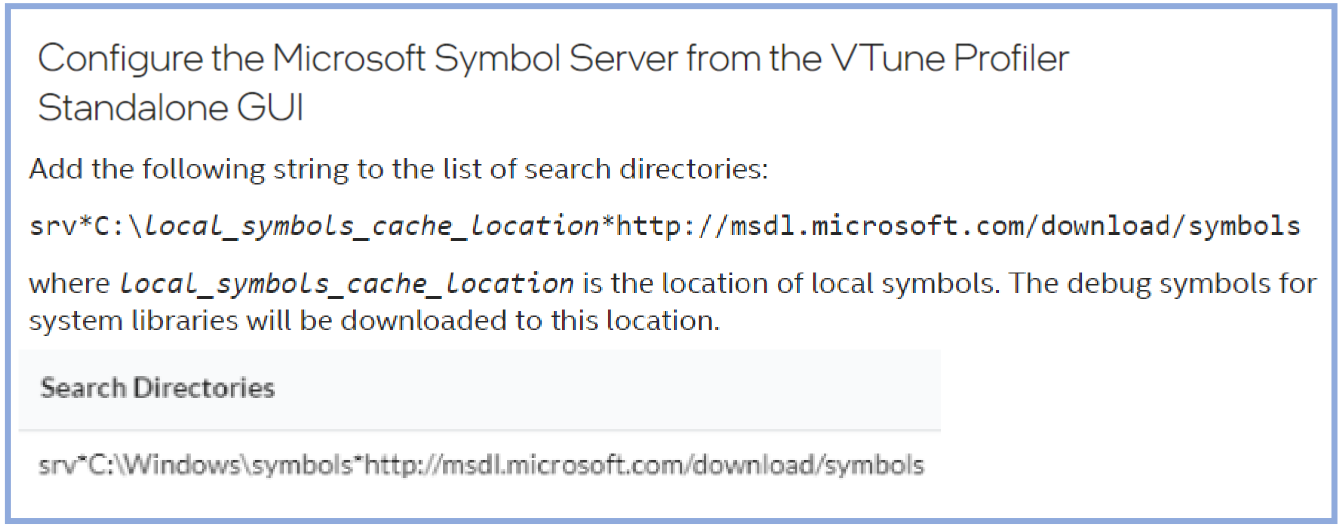
1. Add the directory in Intel VTune where debug symbols(.pdb files) are located
   - Configure Analysis
   - Search Sources/Binaries



Search Directories

C:\buffer\FRIB\build-Charge-Desktop_Qt_6_5_0_MinGW_64_bit-Debug

- Additional resources:
  - [Debug Information for Windows Application Binaries]
  - [Debug Information for Windows System Libraries]



Configure the Microsoft Symbol Server from the VTune Profiler Standalone GUI

Add the following string to the list of search directories:

```
srv*C:\local_symbols_cache_location*http://msdl.microsoft.com/download/symbols
```

where *local_symbols_cache_location* is the location of local symbols. The debug symbols for system libraries will be downloaded to this location.

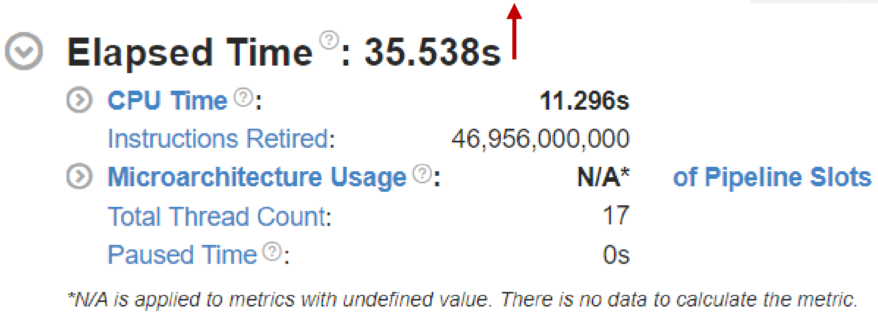
Search Directories

srv*C:\Windows\symbols*http://msdl.microsoft.com/download/symbols

Usage of Profiler

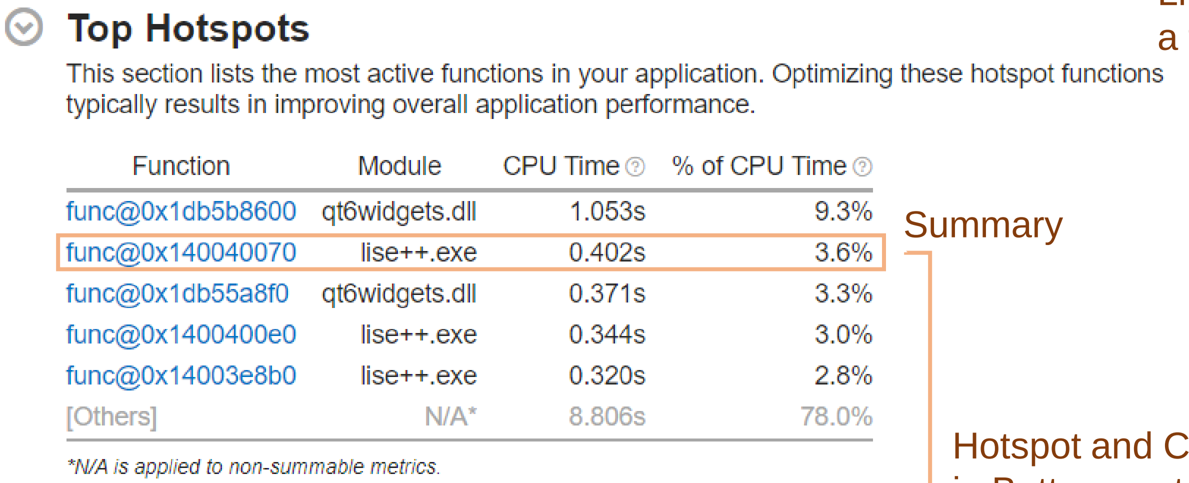
Hardware-based sample testing only the transmission calculations for all nuclei

Transmission calculation: All nuclei


Elapsed Time: 35.538s

CPU Time:	11.296s
Instructions Retired:	46,956,000,000
Microarchitecture Usage:	N/A* of Pipeline Slots
Total Thread Count:	17
Paused Time:	0s

*N/A is applied to metrics with undefined value. There is no data to calculate the metric.


Top Hotspots

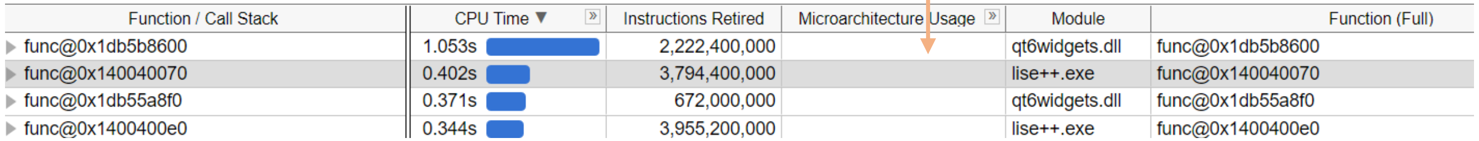
This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time	% of CPU Time
func@0x1db5b8600	qt6widgets.dll	1.053s	9.3%
func@0x140040070	lise++.exe	0.402s	3.6%
func@0x1db55a8f0	qt6widgets.dll	0.371s	3.3%
func@0x1400400e0	lise++.exe	0.344s	3.0%
func@0x14003e8b0	lise++.exe	0.320s	2.8%
[Others]	N/A*	8.806s	78.0%

*N/A is applied to non-summable metrics.

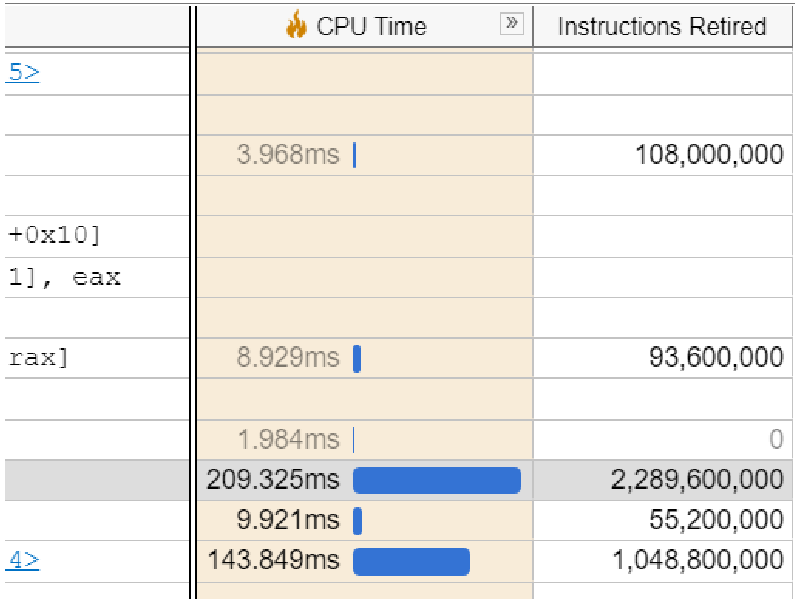
Summary

Hotspot and Callers Analysis in Bottom-up tab

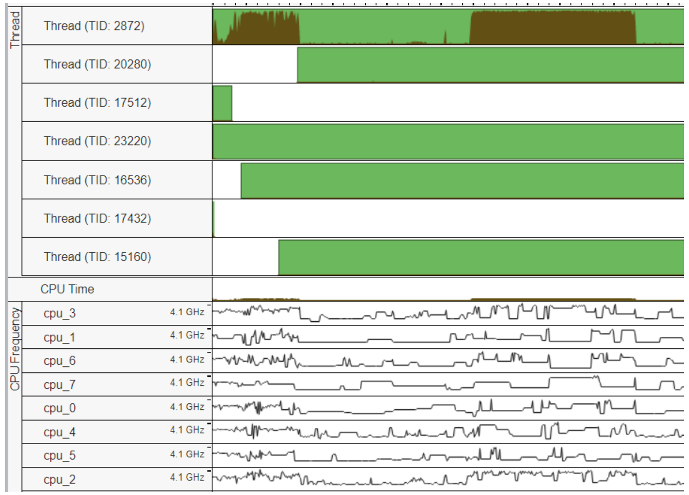


Function / Call Stack	CPU Time ▼	Instructions Retired	Microarchitecture Usage	Module	Function (Full)
▸ func@0x1db5b8600	1.053s	2,222,400,000		qt6widgets.dll	func@0x1db5b8600
▸ func@0x140040070	0.402s	3,794,400,000		lise++.exe	func@0x140040070
▸ func@0x1db55a8f0	0.371s	672,000,000		qt6widgets.dll	func@0x1db55a8f0
▸ func@0x1400400e0	0.344s	3,955,200,000		lise++.exe	func@0x1400400e0

Line by line analysis in a tab for a function



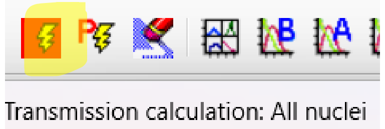
	CPU Time	Instructions Retired
5>		
	3.968ms	108,000,000
+0x10]		
1], eax		
rax]	8.929ms	93,600,000
	1.984ms	0
	209.325ms	2,289,600,000
	9.921ms	55,200,000
4>	143.849ms	1,048,800,000



Hotspot Summary User vs Kernel Mode

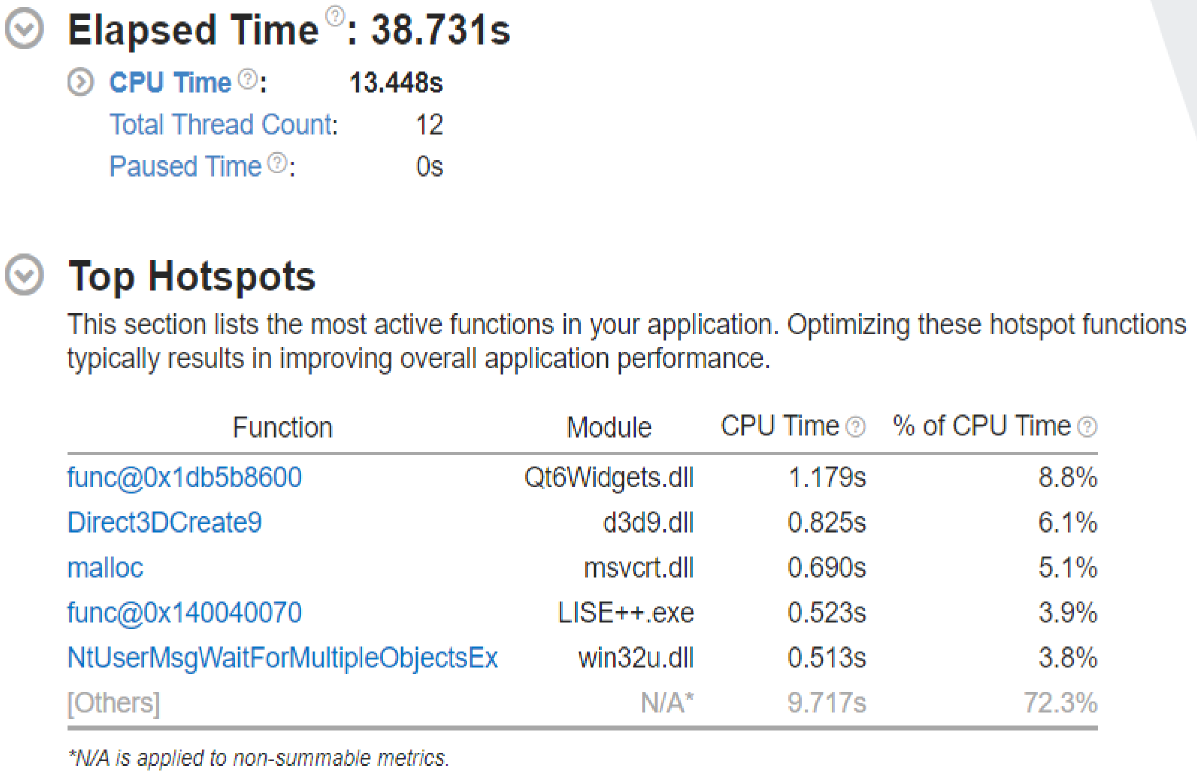
User-Mode Sampling

All samples test only the transmission calculations for all nuclei


Transmission calculation: All nuclei

Use this mode for:

- Profiles longer than a few seconds
- Profiling a single process or a process-tree
- Profiling Python and Intel runtimes



Elapsed Time: 38.731s

CPU Time:	13.448s
Total Thread Count:	12
Paused Time:	0s

Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time	% of CPU Time
func@0x1db5b8600	Qt6Widgets.dll	1.179s	8.8%
Direct3DCreate9	d3d9.dll	0.825s	6.1%
malloc	msvcrt.dll	0.690s	5.1%
func@0x140040070	LISE++.exe	0.523s	3.9%
NtUserMsgWaitForMultipleObjectsEx	win32u.dll	0.513s	3.8%
[Others]	N/A*	9.717s	72.3%

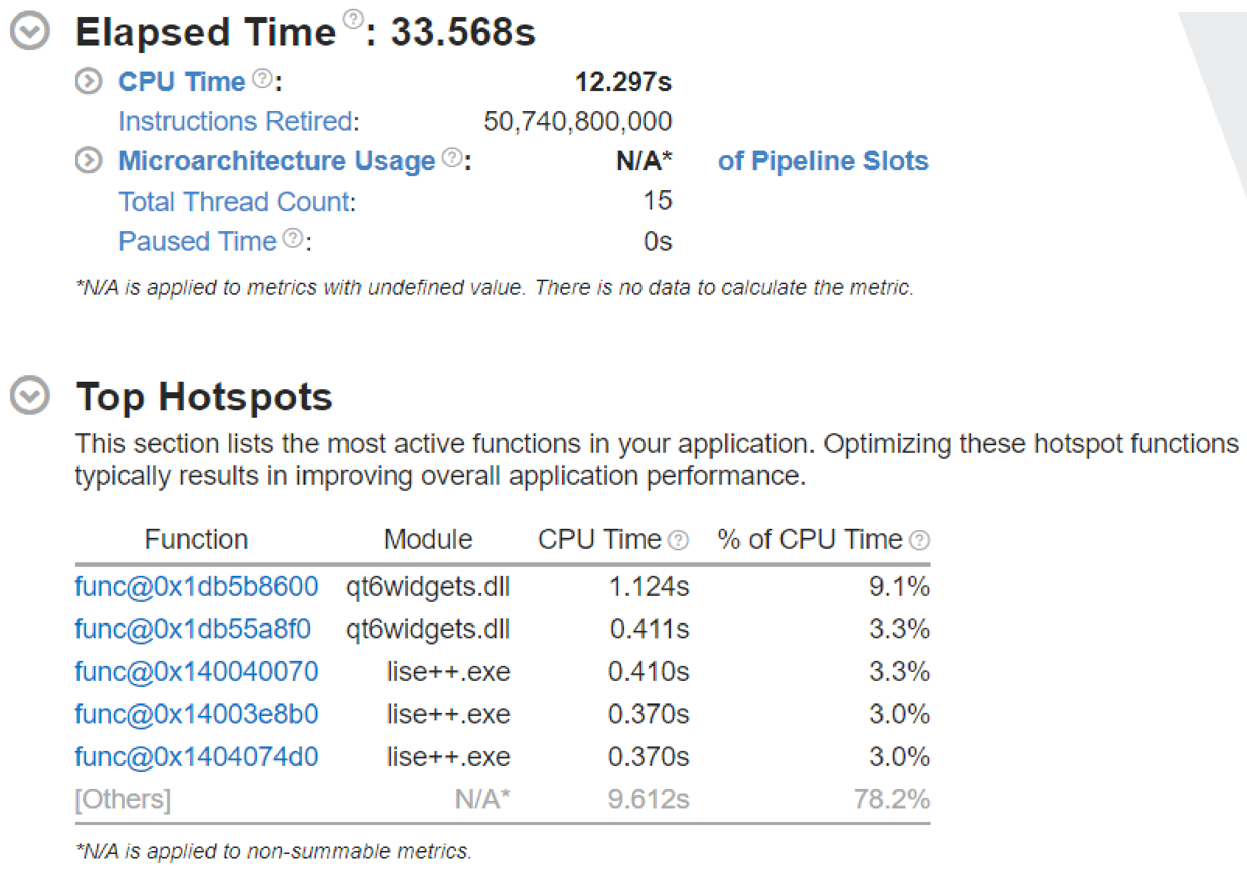
*N/A is applied to non-summable metrics.

Hardware Event-Based Sampling

1 ms CPU sampling interval

Use this mode for:

- Profiles shorter than a few seconds
- Profiling all processes on a system, including kernel



Elapsed Time: 33.568s

CPU Time:	12.297s	
Instructions Retired:	50,740,800,000	
Microarchitecture Usage:	N/A*	of Pipeline Slots
Total Thread Count:	15	
Paused Time:	0s	

*N/A is applied to metrics with undefined value. There is no data to calculate the metric.

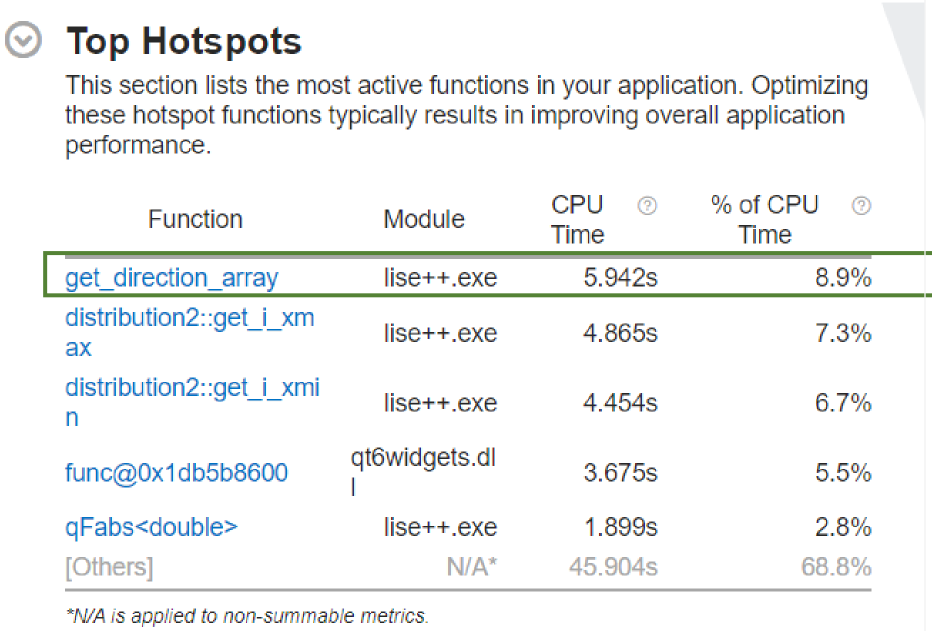
Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time	% of CPU Time
func@0x1db5b8600	qt6widgets.dll	1.124s	9.1%
func@0x1db55a8f0	qt6widgets.dll	0.411s	3.3%
func@0x140040070	lise++.exe	0.410s	3.3%
func@0x14003e8b0	lise++.exe	0.370s	3.0%
func@0x1404074d0	lise++.exe	0.370s	3.0%
[Others]	N/A*	9.612s	78.2%

*N/A is applied to non-summable metrics.

L_Distr2.cpp Optimization



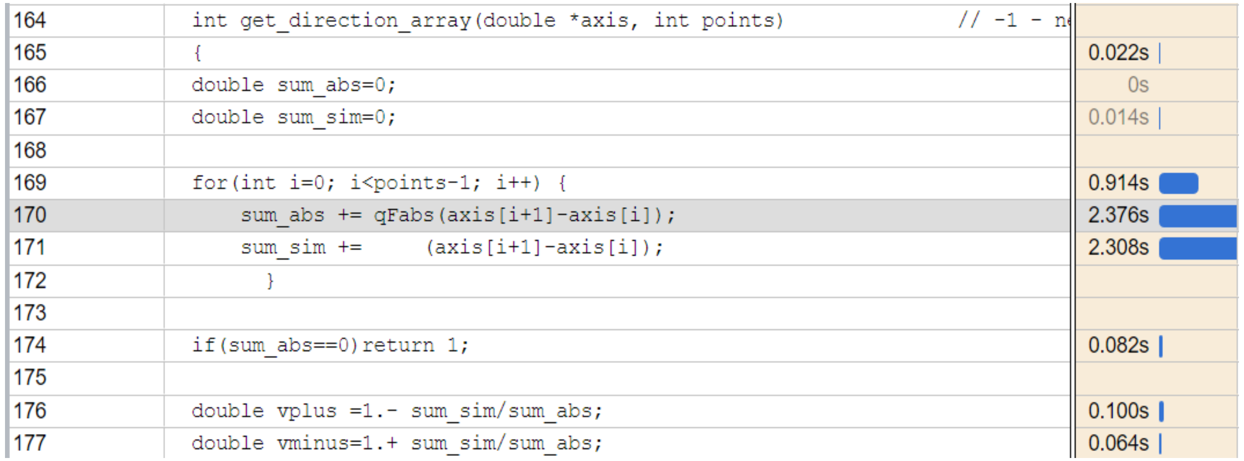
Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time	% of CPU Time
get_direction_array	lise++.exe	5.942s	8.9%
distribution2::get_i_xmax	lise++.exe	4.865s	7.3%
distribution2::get_i_xmin	lise++.exe	4.454s	6.7%
func@0x1db5b8600	qt6widgets.dll	3.675s	5.5%
qFabs<double>	lise++.exe	1.899s	2.8%
[Others]	N/A*	45.904s	68.8%

*N/A is applied to non-summable metrics.

v 16.15.13

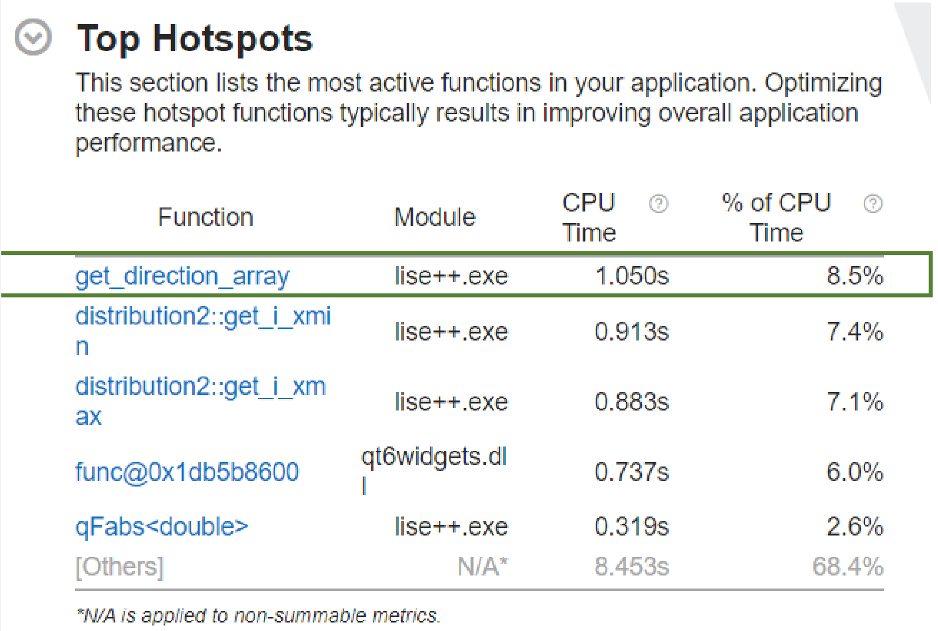


```
int get_direction_array(double *axis, int points)          // -1 - n
{                                                           0.022s
double sum_abs=0;                                               0s
double sum_sim=0;                                           0.014s

for(int i=0; i<points-1; i++) {                             0.914s
    sum_abs += qFabs(axis[i+1]-axis[i]);                    2.376s
    sum_sim +=      (axis[i+1]-axis[i]);                    2.308s
      }

if(sum_abs==0)return 1;                                     0.082s

double vplus =1.- sum_sim/sum_abs;                          0.100s
double vminus=1.+ sum_sim/sum_abs;                          0.064s
```



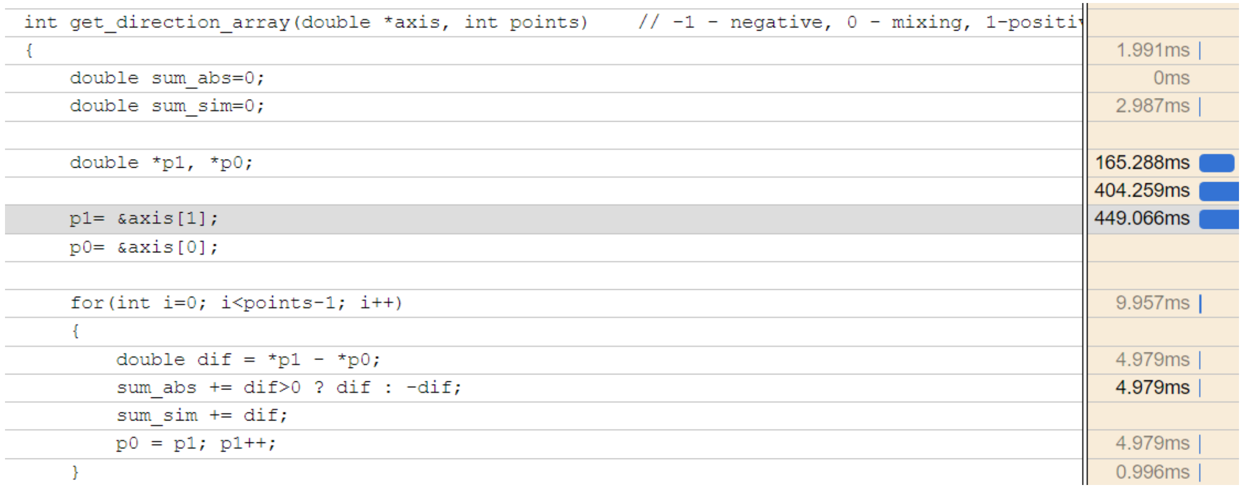
Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time	% of CPU Time
get_direction_array	lise++.exe	1.050s	8.5%
distribution2::get_i_xmin	lise++.exe	0.913s	7.4%
distribution2::get_i_xmax	lise++.exe	0.883s	7.1%
func@0x1db5b8600	qt6widgets.dll	0.737s	6.0%
qFabs<double>	lise++.exe	0.319s	2.6%
[Others]	N/A*	8.453s	68.4%

*N/A is applied to non-summable metrics.

v 16.15.17



```
int get_direction_array(double *axis, int points)    // -1 - negative, 0 - mixing, 1-positi
{                                                                   1.991ms
    double sum_abs=0;                                                   0ms
    double sum_sim=0;                                               2.987ms

    double *p1, *p0;                                              165.288ms
                                                                  404.259ms
    p1= &axis[1];                                                 449.066ms
    p0= &axis[0];

    for(int i=0; i<points-1; i++)                                   9.957ms
    {
        double dif = *p1 - *p0;                                     4.979ms
        sum_abs += dif>0 ? dif : -dif;                              4.979ms
        sum_sim += dif;
        p0 = p1; p1++;                                              4.979ms
    }                                                               0.996ms
```

FRIB

Facility for Rare Isotope Beams
U.S. Department of Energy Office of Science
Michigan State University