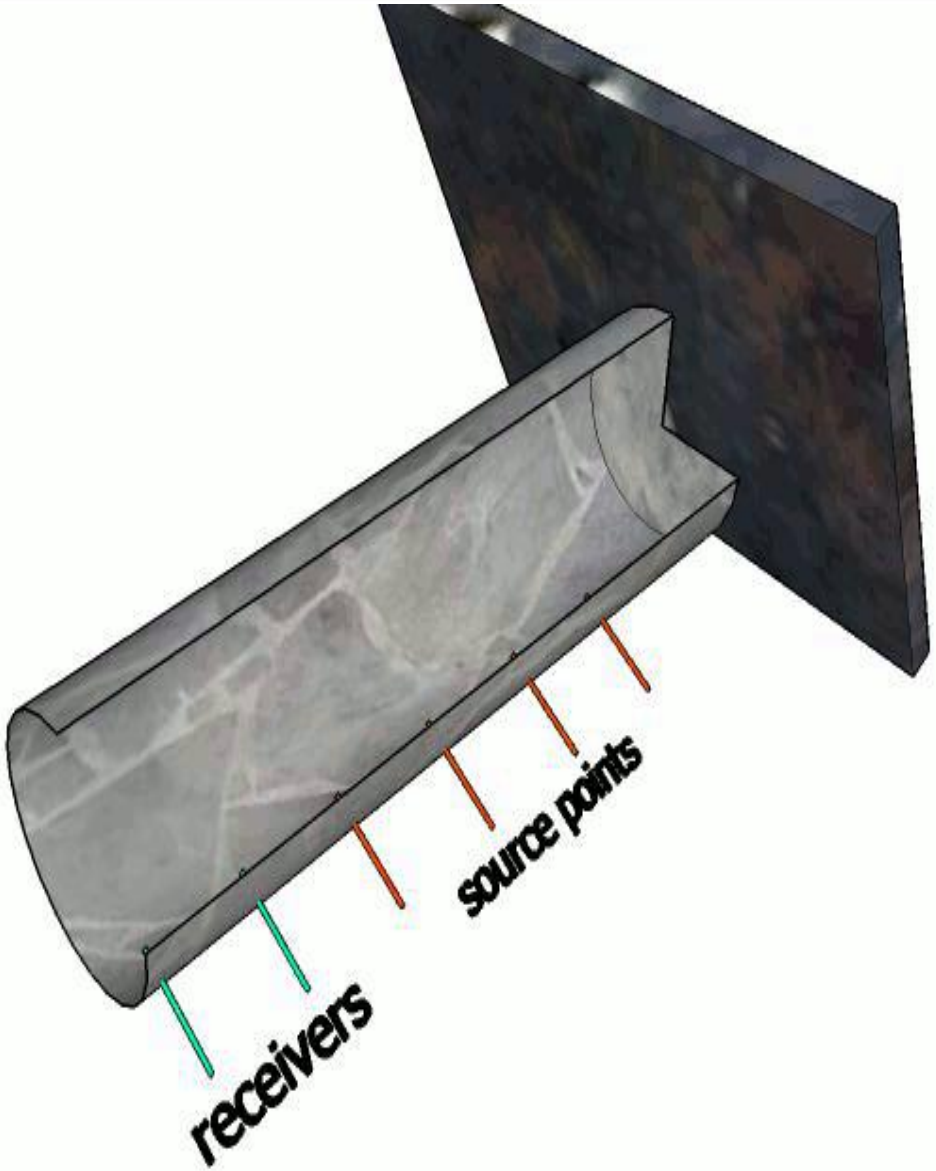
Применение гибридного OpenMP/GPGPU подхода для ускорения вычислений 3D поля скорости по сейсмическим данным

Singularis Lab, Ltd.

Volgograd State Technical University

О чём пойдёт речь

- Система сейсморазведки при прокладке туннелей
- Оптимизация и ускорение вычислительных процедур



Тестовое оборудование:

Intel i5-4570

GeForce GT 640

Исследование проблем производительности

Analysis Target | Analysis Type | Collection Log | Summary | Bottom-up | Caller/Callee

Elapsed Time: 318.848s

Total Thread Count:	1
Overhead Time:	0s
Spin Time:	0s
CPU Time:	317.616s
Paused Time:	0s

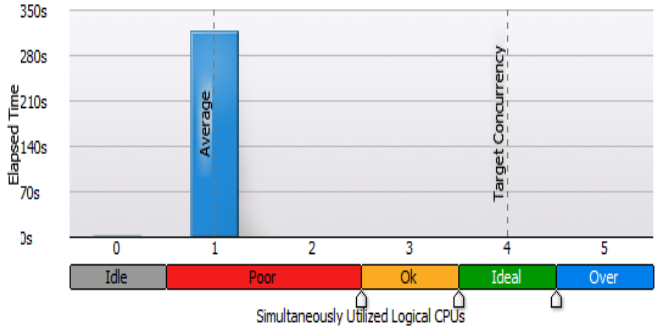
Top Hotspots

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

Function	CPU Time
tripd2	95.632s
put_velocity_along_ray_path	66.868s
get_velocity_along_ray_path	41.185s
vave2int1drecursivesmooth3	25.408s
smooth_slowness1d_noalloc	24.056s
[Others]	64.467s

CPU Usage Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. S
the Thread Concurrency level if a thread is executing code on a CPU while it is logically waiting. Try to keep your

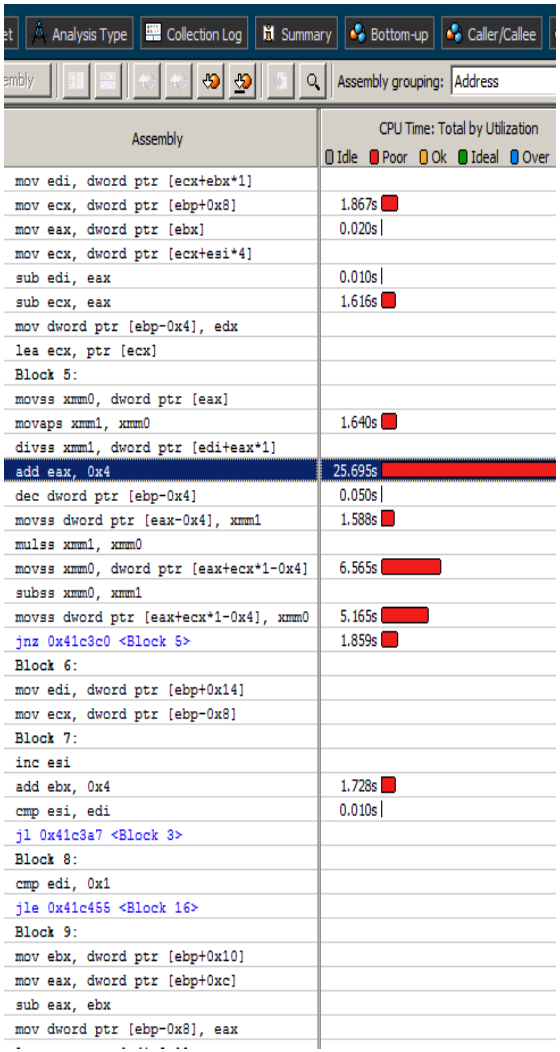


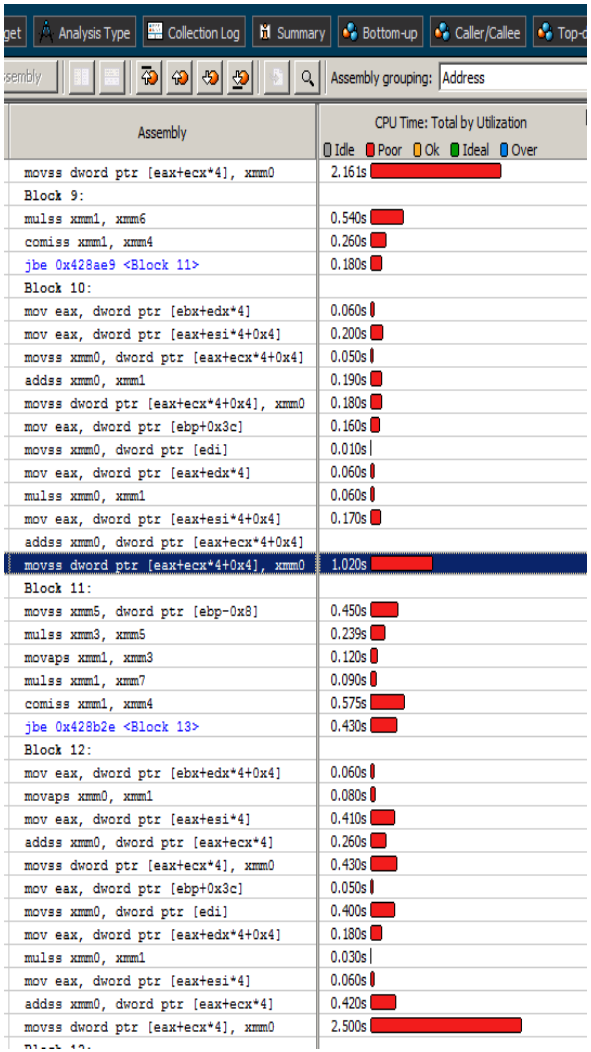
Analysis Target | Analysis Type | Collection Log | Summary | Bottom-up | Caller/Callee | Top-down Tree

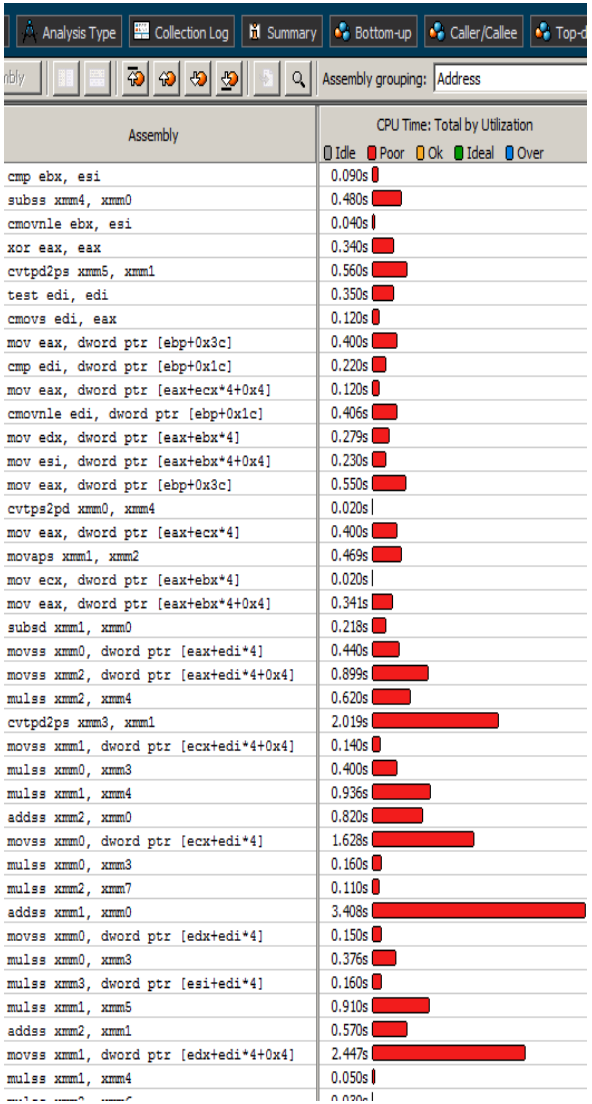
Grouping: Function / Call Stack

Function / Call Stack	CPU Time by Utilization (Idle Poor Ok Ideal Over)	Ov. an.	Module	
tripd2	95.632s	0s	tspvpick3d.exe	tripd2
put_velocity_along_ray_path	66.868s	0s	tspvpick3d.exe	put_vel
get_velocity_along_ray_path	41.185s	0s	tspvpick3d.exe	get_vel
vave2int1drecursivesmooth3	25.408s	0s	tspvpick3d.exe	vave2ir
smooth_slowness1d_noalloc	24.056s	0s	tspvpick3d.exe	smooth_
sdot	19.040s	0s	tspvpick3d.exe	sdot
vsm1d_noalloc	16.677s	0s	tspvpick3d.exe	vsm1d_
compute_straight_ray_path	11.433s	0s	tspvpick3d.exe	computi
localxcor3	9.758s	0s	tspvpick3d.exe	localxcc
_libm_sse2_sqrt_precise	1.862s	0s	MSVCR110.dll	_libm_s
compute_gagc	1.808s	0s	tspvpick3d.exe	computi
find_max	0.879s	0s	tspvpick3d.exe	find_ma
null1	0.440s	0s	tspvpick3d.exe	null1
null2	0.430s	0s	tspvpick3d.exe	null2
fprintf	0.325s	0s	MSVCR110.dll	fprintf
vave2int3drecursive	0.300s	0s	tspvpick3d.exe	vave2ir
parabola_extremum_3point	0.263s	0s	tspvpick3d.exe	parabol
func@0x1000d0c0	0.224s	0s	TSPNGDataC ...	func@0
dc_getPoint	0.200s	0s	TSPNGDataC ...	dc_getF
quant	0.100s	0s	tspvpick3d.exe	quant
fill_neighborhood	0.080s	0s	tspvpick3d.exe	fill_neig
func@0x10007e90	0.060s	0s	TSPNGDataC ...	func@0
find_vector_maxima	0.060s	0s	tspvpick3d.exe	find_ve
std::basic_istream<char,struct std::char_traits<char> >:	0.047s	0s	MSVCP90.dll	std::ba
smooth_slowness3d	0.040s	0s	tspvpick3d.exe	smooth_
_startOneArgErrorHandling	0.040s	0s	MSVCR110.dll	_startO
neighborhood_equalization_med	0.040s	0s	tspvpick3d.exe	neighbo
main	0.040s	0s	tspvpick3d.exe	main
Selected 1 row(s):	95.632s	0s		

tripd2 put_velocity get_velocity





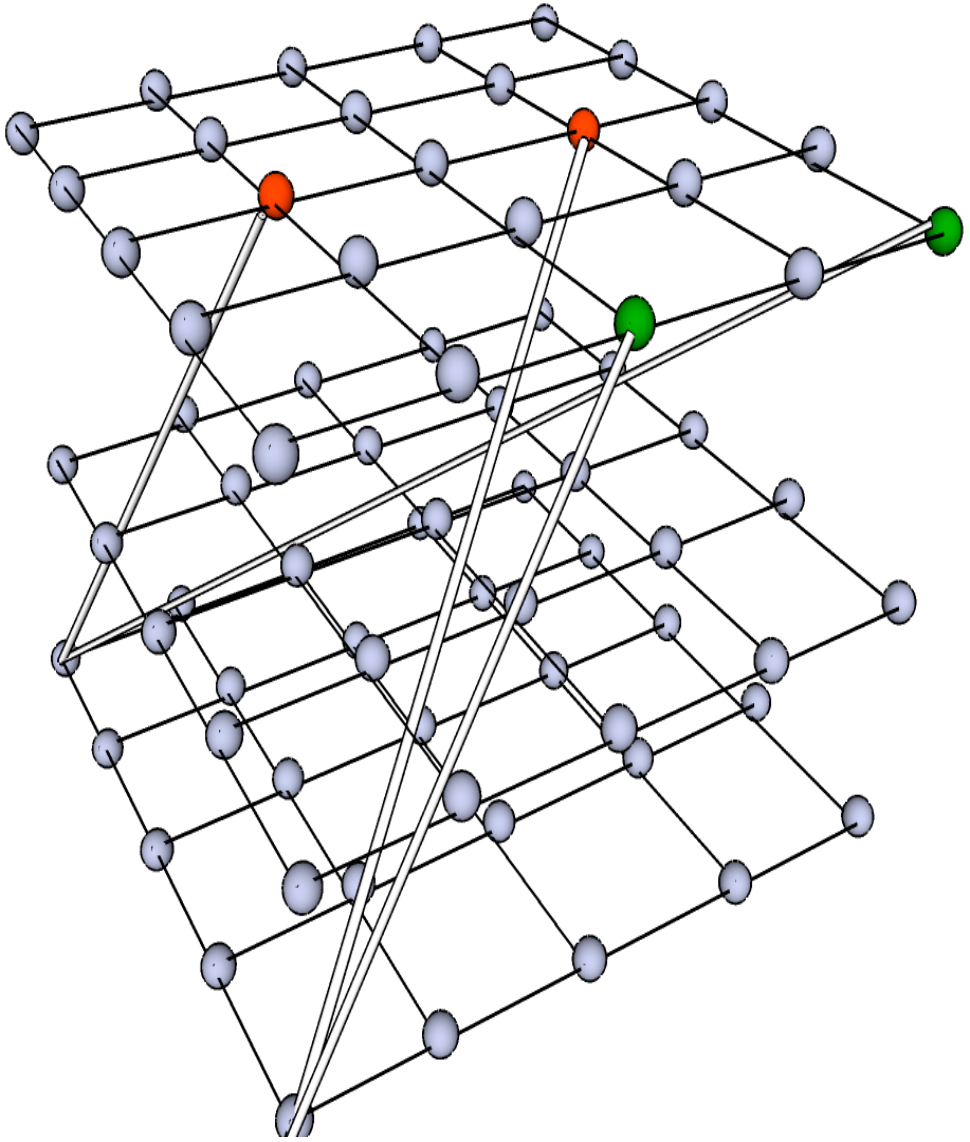


Проблемы доступа к данным

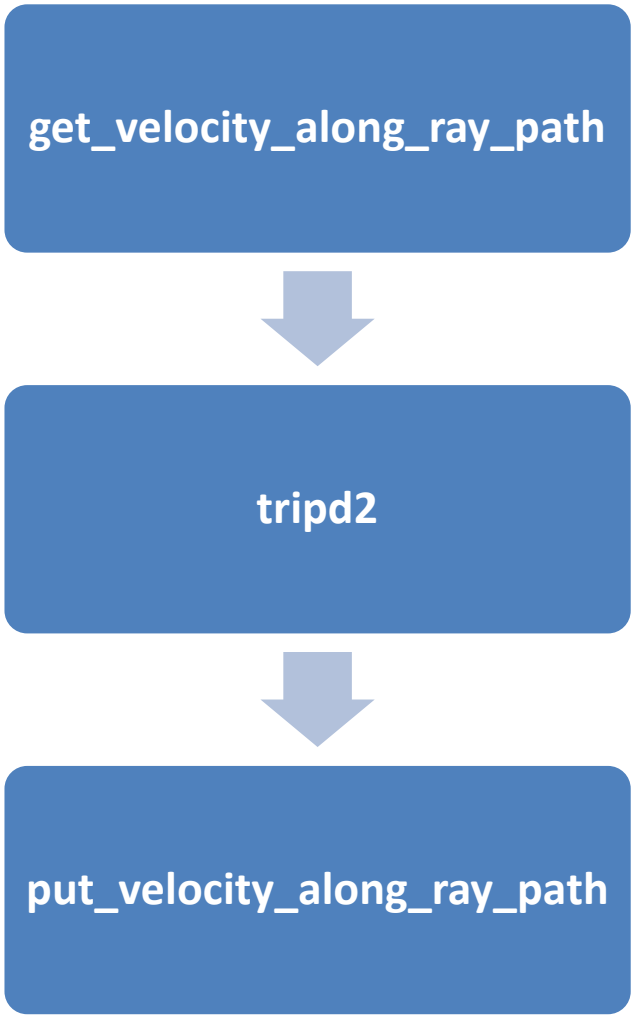
Symbol + Offset	DC accesses	DC misses	Misalign access	Ret inst	DTLB L1M L2H	DTLB L1M L2M	DC refills L2/NB
put_velocity_along_ray_path_p	1069808	140624	236	2062302	37716	1752	406600
tripd2_simplify_p	471328	1926	76	873384	718	42	2638
velocities::get_velocity_along_ray_path	440766	63922	154	1121879	17362	612	103980
qcOutput::muladd	313606	296	589284	590939	50	4	246
vsm1d_noalloc	270030	8036	60	618220	3182	66	49296
velocities::compute_straight_ray_path	218116	3440	586	439061	1344	22	18290
smooth_slowness1d_noalloc	104490	1632	58	438653	704	12	7464
velocities::vave2int1drecursivesmooth3	82818	1692	90018	373051	706	12	7530
qcOutput::multiply	34826	200	55774	40199	6	8	534
qcOutput::localxcor3	34676	68	3348	97221	6	4	512
find_max	6820	22	4	23335			4
velocities::vave2int3drecursiveomp1	5396	3716	2	9067	1008	90	3288
parabola_extremum_3point	3460	132		5076	24		188
null1	2680	270	4	15735	20	12	2516
sqrt	1850	228	410	2383		4	228
fill_neighborhood	1196	34		3668	36		46
tripd2	1092	14		2287	4		138
main	1040			1547	2		
qcOutput::find_vector_maxima	956	12		1831			78
quant	714			1391			2
neighborhood_equalization	586	14	2	1038			34
vsm3d	496	12		806	2		100
border_vs_main_grid	432			1121			26
neighborhood_equalization_med	336	26		861			54
null3	318	10		537		6	182
moment_simpl	300	4		180	2		4
qcOutput::compute_stackword_2d	108	4		319			2
find_max2d	98	10		256			60
std::vector<float,std::allocator<float> >::resize	82	8		15			30
smooth_slowness3d	70	2		318			26
vcomp_atomic_add_r4	64	2		202			
qcOutput::update_average_velocity	44	2	2	64			8
qcOutput::CalcIntVelDipQualomp1	42	10		55	4	10	26

Velocity along ray path

- Проблемы:
  - невыровненный доступ к памяти
  - промахи кэша
- Решение
  - уменьшить куб
  - реорганизовать работу с памятью



Логическая структура программы

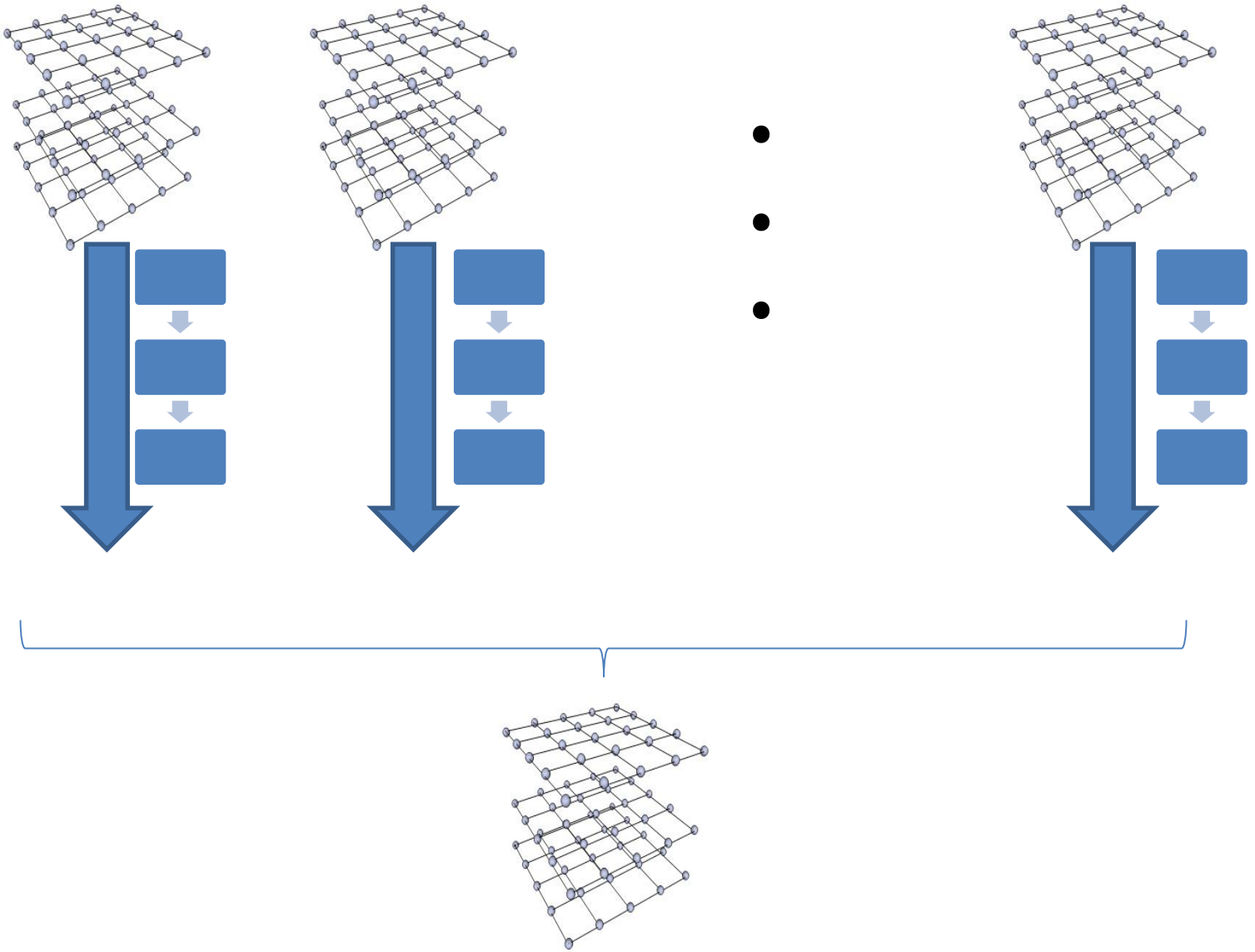


вычисление траекторий лучей в кубе скоростей

сглаживание скорости вдоль траектории луча

модификация куба скоростей

Разделение вычислений между потоками



Результат с SSE intrinsics, выравниванием памяти и OpenMP

Analysis Target | Analysis Type | Collection Log | Summary | Bottom-up | Caller/Callee

Elapsed Time: 66.978s

Total Thread Count:	11
Overhead Time:	0.020s
Spin Time:	3.934s
CPU Time:	249.417s
Paused Time:	0s

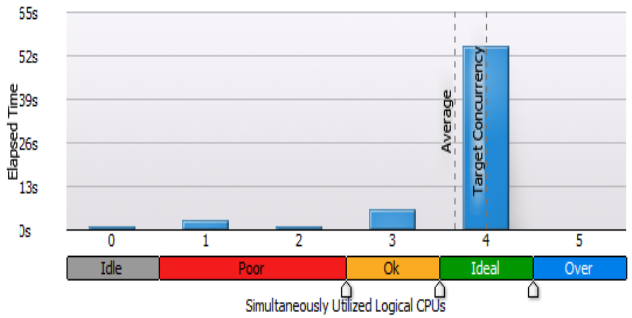
Top Hotspots

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

Function	CPU Time
fastvelocities::tripd2	77.024s
fastvelocities::put_velocity_along_ray_path	52.637s
fastvelocities::get_velocity_along_ray_path	44.502s
fastvelocities::vave2int1drecursivesmooth3	25.351s
fastcube::muladd	12.340s
[Others]	37.564s

CPU Usage Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. S
the Thread Concurrency level if a thread is executing code on a CPU while it is logically waiting. Try to keep your T



Analysis Target | Analysis Type | Collection Log | Summary | Bottom-up | Caller/Callee

Elapsed Time: 318.848s

Total Thread Count:	1
Overhead Time:	0s
Spin Time:	0s
CPU Time:	317.616s
Paused Time:	0s

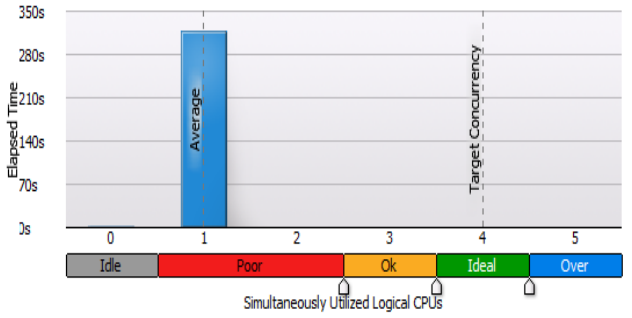
Top Hotspots

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

Function	CPU Time
tripd2	95.632s
put_velocity_along_ray_path	66.868s
get_velocity_along_ray_path	41.185s
vave2int1drecursivesmooth3	25.408s
smooth_slowness1d_noalloc	24.056s
[Others]	64.467s

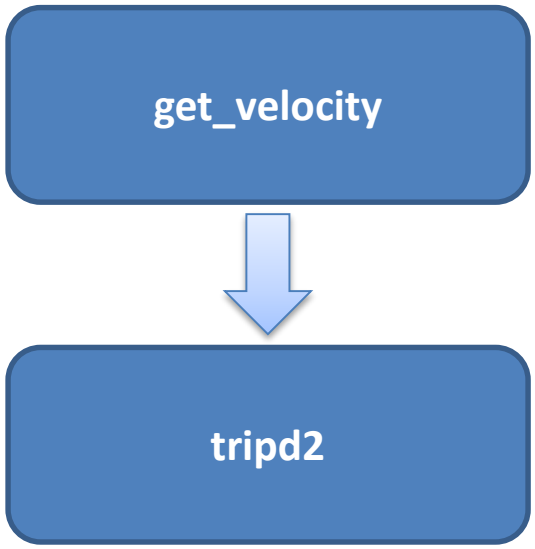
CPU Usage Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Sp
the Thread Concurrency level if a thread is executing code on a CPU while it is logically waiting. Try to keep your T



Использование GPU ускорителя

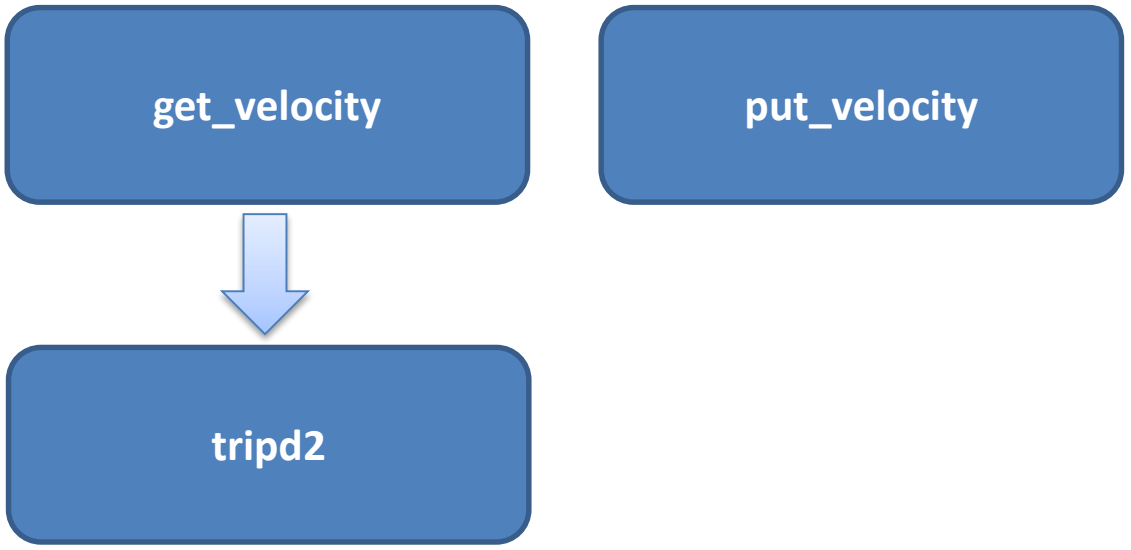
- Step 1: GPU



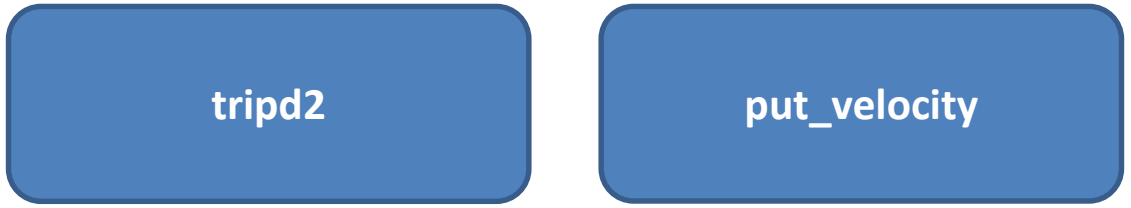
GPU

- Step 2

CPU



- Step 3



Сравнение результатов

Исходный код

Elapsed Time: 318.848s

Total Thread Count:	1
Overhead Time:	0s
Spin Time:	0s
CPU Time:	317.616s
Paused Time:	0s

CPU only

Elapsed Time: 66.978s

Total Thread Count:	11
Overhead Time:	0.020s
Spin Time:	3.934s
CPU Time:	249.417s
Paused Time:	0s

CPU+GPU

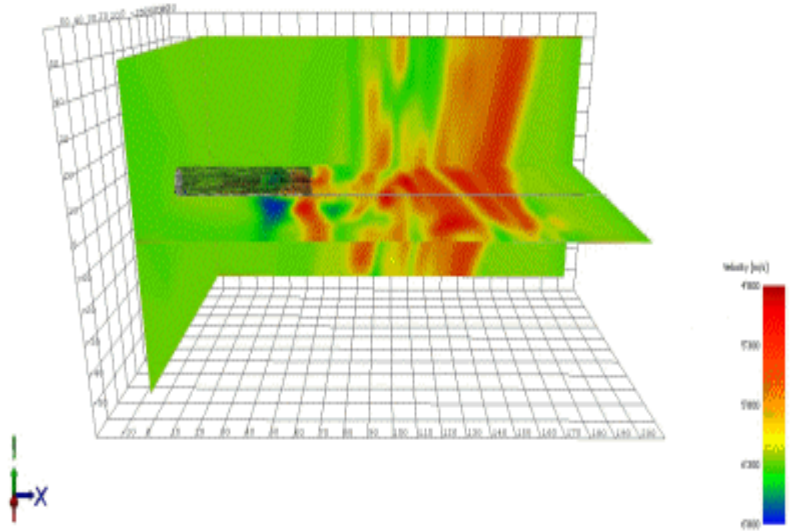
Elapsed Time: 35.057s

Total Thread Count:	19
Overhead Time:	0.319s
Spin Time:	32.426s

A significant portion of CPU time is spent waiti
by backing off then descheduling), or adjustin

CPU Time:	121.823s
Paused Time:	0s

Заключение



- Хотспоты были проанализированы с помощью Intel VTune Amplifier
- Приложение было ускорено за счёт одновременных вычислений на GPU и CPU
- Итоговое ускорение на тестовой машине более 9 раз

Анимация Amberg Technologies (http://www.ambergtechnologies.ch)