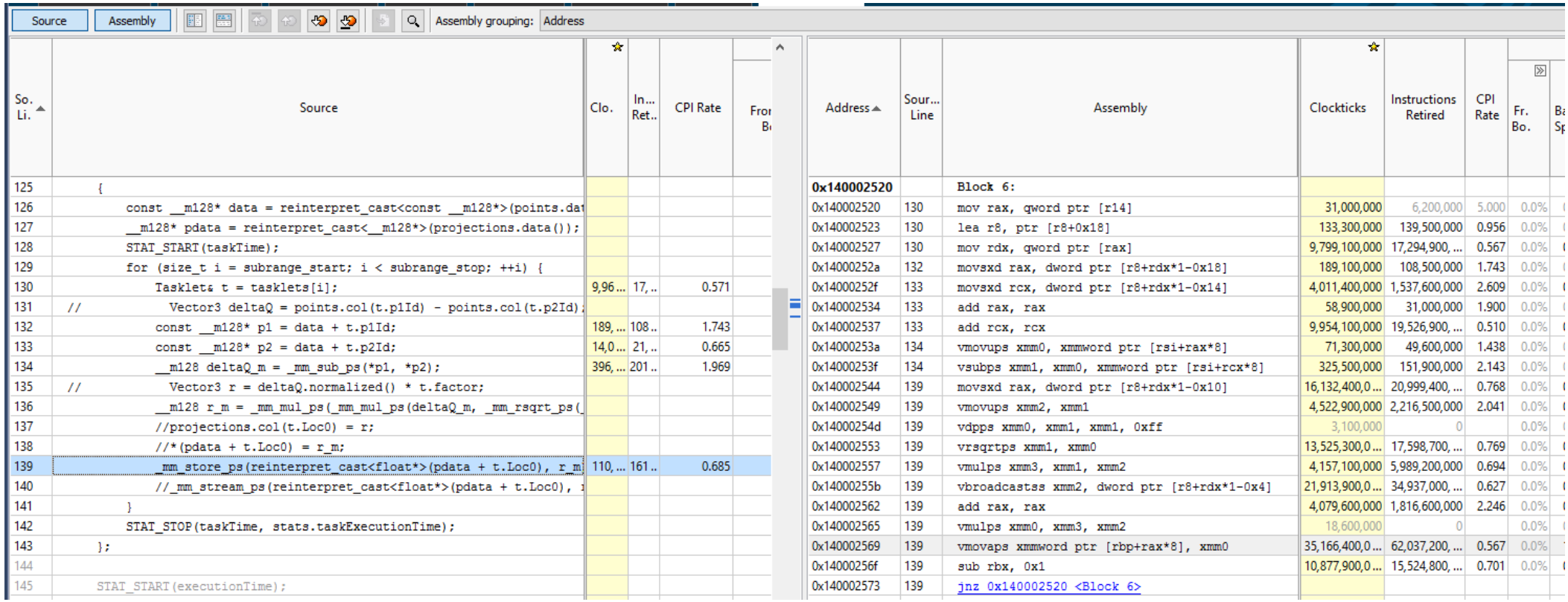
1. Default compiler generated code

So. Li.	Source	Clo.	In... Ret..	CPI Rate	F
125	{				
126	const __m128* data = reinterpret_cast<const __m128*>(points.data())				
127	__m128* pdata = reinterpret_cast<__m128*>(projections.data());				
128	STAT_START(taskTime);				
129	for (size_t i = subrange_start; i < subrange_stop; ++i) {				
130	Tasklet& t = tasklets[i];	9,73 ...	16, ..	0.607	
131	Vector3 deltaQ = points.col(t.p1Id) - points.col(t.p2Id);				
132	const __m128* p1 = data + t.p1Id;	784, ...	802 ..	0.977	
133	const __m128* p2 = data + t.p2Id;	13,5 ...	19, ..	0.680	
134	__m128 deltaQ_m = _mm_sub_ps(*p1, *p2);	1,23 ...	988 ..	1.248	
135	Vector3 r = deltaQ.normalized() * t.factor;				
136	__m128 r_m = _mm_mul_ps(_mm_mul_ps(deltaQ_m, _mm_rsqrt_ps(_mm_d				
137	//projections.col(t.Loc0) = r;				
138	*(pdata + t.Loc0) = r_m;	110, ...	152 ..	0.723	
139	//_mm_store_ps(reinterpret_cast<float*>(pdata + t.Loc0), r_m);				
140	//_mm_stream_ps(reinterpret_cast<float*>(pdata + t.Loc0), r_m);				
141	}				
142	STAT_STOP(taskTime, stats.taskExecutionTime);				
143	};				
144					

Address	Sour... Line	Assembly	Clockticks	Instructions Retired	CPI Rate	L... Fr. Bo.	L... Ba. Sp.
0x140002520		**Block 6:**					
0x140002520	130	mov rax, qword ptr [r14]	260,400,000	102,300,000	2.545	0.0%	0.0
0x140002523	130	lea r8, ptr [r8+0x18]	331,700,000	142,600,000	2.326	0.0%	0.0
0x140002527	130	mov rdx, qword ptr [rax]	9,138,800,000	15,788,300, ...	0.579	0.2%	0.0
0x14000252a	132	movsxd rax, dword ptr [r8+rdx*1-0x18]	784,300,000	802,900,000	0.977	0.0%	0.1
0x14000252f	133	movsxd rcx, dword ptr [r8+rdx*1-0x14]	4,519,800,000	1,608,900,000	2.809	0.0%	0.0
0x140002534	133	add rax, rax	133,300,000	105,400,000	1.265	0.0%	0.0
0x140002537	133	add rcx, rcx	8,921,800,000	18,249,700, ...	0.489	0.3%	0.0
0x14000253a	134	vmovups xmm0, xmmword ptr [rsi+rax*8]	694,400,000	725,400,000	0.957	0.0%	0.1
0x14000253f	134	vsubps xmm1, xmm0, xmmword ptr [rsi+rcx*8]	539,400,000	263,500,000	2.047	0.0%	0.1
0x140002544	138	movsxd rax, dword ptr [r8+rdx*1-0x10]	15,459,700,0 ...	18,789,100, ...	0.823	0.4%	0.0
0x140002549	138	vdpps xmm2, xmm1, xmm1, 0xff	5,040,600,000	2,597,800,000	1.940	0.0%	0.0
0x14000254f	138	vrsqrtps xmm0, xmm2	13,562,500,0 ...	17,139,900, ...	0.791	0.2%	0.5
0x140002553	138	vmulps xmm4, xmm0, xmm1	4,141,600,000	5,685,400,000	0.728	0.1%	0.0
0x140002557	138	vbroadcastss xmm1, dword ptr [r8+rdx*1-0x4]	18,054,400,0 ...	26,883,200, ...	0.672	0.5%	0.0
0x14000255e	138	add rax, rax	7,604,300,000	5,648,200,000	1.346	0.0%	0.2
0x140002561	138	vmulps xmm0, xmm4, xmm1	415,400,000	263,500,000	1.576	0.0%	0.0
0x140002565	138	vmovups xmmword ptr [rbp+rax*8], xmm0	35,423,700,0 ...	60,490,300, ...	0.586	0.2%	2.7
0x14000256b	138	sub rbx, 0x1	10,313,700,0 ...	14,703,300, ...	0.701	0.1%	0.3
0x14000256f	138	jnz 0x140002520 <Block 6>					

2. _mm_store (vmovaps)



Source | Assembly | Assembly grouping: Address

So. Li.	Source	Clo.	In... Ret..	CPI Rate	Fro... Bl...
125	{				
126	const __m128* data = reinterpret_cast<const __m128*>(points.dat				
127	__m128* pdata = reinterpret_cast<__m128*>(projections.data());				
128	STAT_START(taskTime);				
129	for (size_t i = subrange_start; i < subrange_stop; ++i) {				
130	Tasklet& t = tasklets[i];	9,96...	17,..	0.571	
131	// Vector3 deltaQ = points.col(t.p1Id) - points.col(t.p2Id);				
132	const __m128* p1 = data + t.p1Id;	189,...	108..	1.743	
133	const __m128* p2 = data + t.p2Id;	14,0...	21,..	0.665	
134	__m128 deltaQ_m = _mm_sub_ps(*p1, *p2);	396,...	201..	1.969	
135	// Vector3 r = deltaQ.normalized() * t.factor;				
136	__m128 r_m = _mm_mul_ps(_mm_mul_ps(deltaQ_m, _mm_rsqrt_ps(				
137	//projections.col(t.Loc0) = r;				
138	//*(pdata + t.Loc0) = r_m;				
139	_mm_store_ps(reinterpret_cast<float*>(pdata + t.Loc0), r_m	110,...	161..	0.685	
140	//_mm_stream_ps(reinterpret_cast<float*>(pdata + t.Loc0), r				
141	}				
142	STAT_STOP(taskTime, stats.taskExecutionTime);				
143	};				
144					
145	STAT_START(executionTime);				

Address	Sour... Line	Assembly	Clockticks	Instructions Retired	CPI Rate	Fr. Bo.	Ba... Sp...
0x140002520		**Block 6:**					
0x140002520	130	mov rax, qword ptr [r14]	31,000,000	6,200,000	5.000	0.0%	
0x140002523	130	lea r8, ptr [r8+0x18]	133,300,000	139,500,000	0.956	0.0%	
0x140002527	130	mov rdx, qword ptr [rax]	9,799,100,000	17,294,900,...	0.567	0.0%	
0x14000252a	132	movsxd rax, dword ptr [r8+rdx*1-0x18]	189,100,000	108,500,000	1.743	0.0%	
0x14000252f	133	movsxd rcx, dword ptr [r8+rdx*1-0x14]	4,011,400,000	1,537,600,000	2.609	0.0%	
0x140002534	133	add rax, rax	58,900,000	31,000,000	1.900	0.0%	
0x140002537	133	add rcx, rcx	9,954,100,000	19,526,900,...	0.510	0.0%	
0x14000253a	134	vmovups xmm0, xmmword ptr [rsi+rax*8]	71,300,000	49,600,000	1.438	0.0%	
0x14000253f	134	vsubps xmm1, xmm0, xmmword ptr [rsi+rcx*8]	325,500,000	151,900,000	2.143	0.0%	
0x140002544	139	movsxd rax, dword ptr [r8+rdx*1-0x10]	16,132,400,0...	20,999,400,...	0.768	0.0%	
0x140002549	139	vmovups xmm2, xmm1	4,522,900,000	2,216,500,000	2.041	0.0%	
0x14000254d	139	vdpps xmm0, xmm1, xmm1, 0xff	3,100,000	0		0.0%	
0x140002553	139	vrsqrtps xmm1, xmm0	13,525,300,0...	17,598,700,...	0.769	0.0%	
0x140002557	139	vmulps xmm3, xmm1, xmm2	4,157,100,000	5,989,200,000	0.694	0.0%	
0x14000255b	139	vbroadcastss xmm2, dword ptr [r8+rdx*1-0x4]	21,913,900,0...	34,937,000,...	0.627	0.0%	
0x140002562	139	add rax, rax	4,079,600,000	1,816,600,000	2.246	0.0%	
0x140002565	139	vmulps xmm0, xmm3, xmm2	18,600,000	0		0.0%	
0x140002569	139	vmovaps xmmword ptr [rbp+rax*8], xmm0	35,166,400,0...	62,037,200,...	0.567	0.0%	
0x14000256f	139	sub rbx, 0x1	10,877,900,0...	15,524,800,...	0.701	0.0%	
0x140002573	139	jnz 0x140002520 <Block 6>					

3. _mm_stream_ps (vmovntps)

So. Li.	Source	Clo.	In... Ret..	CPI Rate	Fro... B...
125	{				
126	const __m128* data = reinterpret_cast<const __m128*>(points.dat				
127	__m128* pdata = reinterpret_cast<__m128*>(projections.data());				
128	STAT_START(taskTime);				
129	for (size_t i = subrange_start; i < subrange_stop; ++i) {				
130	Tasklet& t = tasklets[i];	10,5...	18,..	0.583	
131	// Vector3 deltaQ = points.col(t.p1Id) - points.col(t.p2Id);				
132	const __m128* p1 = data + t.p1Id;	223,...	117..	1.895	
133	const __m128* p2 = data + t.p2Id;	11,9...	21,..	0.546	
134	__m128 deltaQ_m = _mm_sub_ps(*p1, *p2);	285,...	145..	1.957	
135	// Vector3 r = deltaQ.normalized() * t.factor;				
136	__m128 r_m = _mm_mul_ps(_mm_mul_ps(deltaQ_m, _mm_rsqrt_ps(				
137	//projections.col(t.Loc0) = r;				
138	//*(pdata + t.Loc0) = r_m;				
139	//_mm_store_ps(reinterpret_cast<float*>(pdata + t.Loc0), r_				
140	_mm_stream_ps(reinterpret_cast<float*>(pdata + t.Loc0), r_r	100,...	159..	0.631	
141	}				
142	STAT_STOP(taskTime, stats.taskExecutionTime);				
143	};				
144					
145	STAT_START(executionTime);				

Address	Sour... Line	Assembly	Clockticks	Instructions Retired	CPI Rate	Fr. Bo.	B... S...
0x140002520		**Block 6:**					
0x140002520	130	mov rax, qword ptr [r14]	24,800,000	3,100,000	8.000	0.0%	
0x140002523	130	lea r8, ptr [r8+0x18]	198,400,000	170,500,000	1.164	0.0%	
0x140002527	130	mov rdx, qword ptr [rax]	10,372,600,0...	18,014,100,...	0.576	0.0%	
0x14000252a	132	movsxd rax, dword ptr [r8+rdx*1-0x18]	223,200,000	117,800,000	1.895	0.0%	
0x14000252f	133	movsxd rcx, dword ptr [r8+rdx*1-0x14]	1,934,400,000	654,100,000	2.957	0.0%	
0x140002534	133	add rax, rax	52,700,000	9,300,000	5.667	0.0%	
0x140002537	133	add rcx, rcx	9,972,700,000	21,259,800,...	0.469	0.0%	
0x14000253a	134	vmovups xmm0, xmmword ptr [rsi+rax*8]	40,300,000	58,900,000	0.684	0.0%	
0x14000253f	134	vsubps xmm1, xmm0, xmmword ptr [rsi+rcx*8]	244,900,000	86,800,000	2.821	0.0%	
0x140002544	140	movsxd rax, dword ptr [r8+rdx*1-0x10]	13,054,100,0...	20,249,200,...	0.645	0.0%	
0x140002549	140	vmovups xmm2, xmm1	1,763,900,000	694,400,000	2.540	0.0%	
0x14000254d	140	vdpps xmm0, xmm1, xmm1, 0xff					
0x140002553	140	vrsqrtps xmm1, xmm0	12,043,500,0...	17,552,200,...	0.686	0.0%	
0x140002557	140	vmulps xmm3, xmm1, xmm2	3,134,100,000	5,211,100,000	0.601	0.0%	
0x14000255b	140	vbroadcastss xmm2, dword ptr [r8+rdx*1-0x4]	21,926,300,0...	35,085,800,...	0.625	0.0%	
0x140002562	140	add rax, rax	1,581,000,000	601,400,000	2.629	0.0%	
0x140002565	140	vmulps xmm0, xmm3, xmm2					
0x140002569	140	vmovntps xmmword ptr [rbp+rax*8], xmm0	36,576,900,0...	64,458,300,...	0.567	0.0%	
0x14000256f	140	sub rbx, 0x1	10,716,700,0...	15,775,900,...	0.679	0.0%	
0x140002573	140	jnz 0x140002520 <Block 6>					

Profiling Results (General Exploration)

1. Default compiler generated code

1 thread

Elapsed Time: 40.732s

Clockticks	135,814,100,000	
Instructions Retired	190,101,300,000	
CPI Rate	0.714	
MUX Reliability	0.924	
Front-End Bound	2.5%	of Pipeline Slots
Bad Speculation	0.2%	of Pipeline Slots
Back-End Bound	53.5%	of Pipeline Slots
Memory Bound	24.1%	of Pipeline Slots
L1 Bound	7.4%	of Clockticks
DTLB Overhead	2.0%	of Clockticks
Loads Blocked by Store Forwarding	0.0%	of Clockticks
Lock Latency	0.0%	of Clockticks
Split Loads	0.0%	of Clockticks
4K Aliasing	2.8%	of Clockticks
FB Full	7.0%	of Clockticks
L2 Bound	0.0%	of Clockticks
L3 Bound	1.5%	of Clockticks
DRAM Bound	15.5%	of Clockticks
Memory Bandwidth	1.8%	of Clockticks
Memory Latency	93.7%	of Clockticks
Store Bound	0.0%	of Clockticks
Core Bound	29.4%	of Pipeline Slots
Divider	0.0%	of Clockticks
Port Utilization	26.5%	of Clockticks
Cycles of 0 Ports Utilized	24.2%	of Clockticks
Cycles of 1 Port Utilized	24.0%	of Clockticks
Cycles of 2 Ports Utilized	26.9%	of Clockticks
Cycles of 3+ Ports Utilized	29.1%	of Clockticks
Port 0	29.1%	of Clockticks
Port 1	34.0%	of Clockticks
Port 2	34.3%	of Clockticks
Port 3	35.1%	of Clockticks
Port 4	7.3%	of Clockticks
Port 5	35.1%	of Clockticks
Retiring	43.9%	of Pipeline Slots
Total Thread Count	2	
Paused Time	3.042s	

2 threads

Elapsed Time: 34.198s

Clockticks	217,976,500,000	
Instructions Retired	190,054,800,000	
CPI Rate	1.147	
MUX Reliability	0.984	
Front-End Bound	1.5%	of Pipeline Slots
Bad Speculation	0.3%	of Pipeline Slots
Back-End Bound	72.6%	of Pipeline Slots
Memory Bound	53.0%	of Pipeline Slots
L1 Bound	6.8%	of Clockticks
DTLB Overhead	1.2%	of Clockticks
Loads Blocked by Store Forwarding	0.0%	of Clockticks
Lock Latency	0.0%	of Clockticks
Split Loads	0.0%	of Clockticks
4K Aliasing	1.9%	of Clockticks
FB Full	15.5%	of Clockticks
L2 Bound	0.0%	of Clockticks
L3 Bound	8.7%	of Clockticks
DRAM Bound	37.3%	of Clockticks
Memory Bandwidth	10.9%	of Clockticks
Memory Latency	82.9%	of Clockticks
Store Bound	0.0%	of Clockticks
Core Bound	19.6%	of Pipeline Slots
Divider	0.0%	of Clockticks
Port Utilization	18.1%	of Clockticks
Cycles of 0 Ports Utilized	49.5%	of Clockticks
Cycles of 1 Port Utilized	17.5%	of Clockticks
Cycles of 2 Ports Utilized	16.6%	of Clockticks
Cycles of 3+ Ports Utilized	15.7%	of Clockticks
Port 0	18.5%	of Clockticks
Port 1	21.9%	of Clockticks
Port 2	22.2%	of Clockticks
Port 3	22.6%	of Clockticks
Port 4	4.6%	of Clockticks
Port 5	22.0%	of Clockticks
Retiring	25.7%	of Pipeline Slots
Total Thread Count	5	
Paused Time	2.890s	

2. _mm_store (vmovaps)

1 thread			2 threads		
Elapsed Time: 40.674s			**Elapsed Time: 34.483s**		
Clockticks	135,448,300,000		Clockticks	219,783,800,000	
Instructions Retired	200,080,200,000		Instructions Retired	200,179,400,000	
CPI Rate	0.677		CPI Rate	1.098	
MUX Reliability	0.933		MUX Reliability	0.960	
Front-End Bound	**0.1%**	**of Pipeline Slots**	**Front-End Bound**	**0.6%**	**of Pipeline Slots**
Bad Speculation	**0.3%**	**of Pipeline Slots**	**Bad Speculation**	**0.2%**	**of Pipeline Slots**
Back-End Bound	**56.2%**	**of Pipeline Slots**	**Back-End Bound**	**71.7%**	**of Pipeline Slots**
Memory Bound	**23.7%**	**of Pipeline Slots**	**Memory Bound**	**50.1%**	**of Pipeline Slots**
L1 Bound	**6.5%**	**of Clockticks**	**L1 Bound**	**1.4%**	**of Clockticks**
DTLB Overhead	1.8%	of Clockticks	DTLB Overhead	1.3%	of Clockticks
Loads Blocked by Store Forwarding	0.0%	of Clockticks	Loads Blocked by Store Forwarding	0.0%	of Clockticks
Lock Latency	0.0%	of Clockticks	Lock Latency	0.0%	of Clockticks
Split Loads	0.0%	of Clockticks	Split Loads	0.0%	of Clockticks
4K Aliasing	3.2%	of Clockticks	4K Aliasing	1.8%	of Clockticks
FB Full	8.1%	of Clockticks	FB Full	17.3%	of Clockticks
L2 Bound	0.0%	of Clockticks	L2 Bound	1.7%	of Clockticks
L3 Bound	**0.8%**	**of Clockticks**	**L3 Bound**	**9.0%**	**of Clockticks**
DRAM Bound	**14.1%**	**of Clockticks**	**DRAM Bound**	**36.0%**	**of Clockticks**
Memory Bandwidth	10.3%	of Clockticks	Memory Bandwidth	14.2%	of Clockticks
Memory Latency	**89.7%**	**of Clockticks**	**Memory Latency**	**83.4%**	**of Clockticks**
Store Bound	**0.0%**	**of Clockticks**	**Store Bound**	**0.0%**	**of Clockticks**
Core Bound	**32.5%**	**of Pipeline Slots**	**Core Bound**	**21.6%**	**of Pipeline Slots**
Divider	0.0%	of Clockticks	Divider	0.0%	of Clockticks
Port Utilization	**26.4%**	**of Clockticks**	**Port Utilization**	**20.8%**	**of Clockticks**
Cycles of 0 Ports Utilized	20.9%	of Clockticks	Cycles of 0 Ports Utilized	51.3%	of Clockticks
Cycles of 1 Port Utilized	24.7%	of Clockticks	Cycles of 1 Port Utilized	17.6%	of Clockticks
Cycles of 2 Ports Utilized	27.9%	of Clockticks	Cycles of 2 Ports Utilized	17.1%	of Clockticks
Cycles of 3+ Ports Utilized	**30.0%**	**of Clockticks**	**Cycles of 3+ Ports Utilized**	**16.4%**	**of Clockticks**
Port 0	31.3%	of Clockticks	Port 0	18.2%	of Clockticks
Port 1	36.6%	of Clockticks	Port 1	21.3%	of Clockticks
Port 2	36.8%	of Clockticks	Port 2	21.5%	of Clockticks
Port 3	37.7%	of Clockticks	Port 3	22.0%	of Clockticks
Port 4	7.8%	of Clockticks	Port 4	4.4%	of Clockticks
Port 5	37.1%	of Clockticks	Port 5	21.5%	of Clockticks
Retiring	**43.4%**	**of Pipeline Slots**	**Retiring**	**27.5%**	**of Pipeline Slots**
Total Thread Count	2		Total Thread Count	3	
Paused Time	2.899s		Paused Time	2.911s	

3. _mm_stream_ps (vmovntps)

1 thread			2 threads		
Elapsed Time: 37.623s			**Elapsed Time: 26.798s**		
Clockticks:	124,325,500,000		Clockticks:	165,974,000,000	
Instructions Retired:	200,154,600,000		Instructions Retired:	200,064,700,000	
CPI Rate:	0.621		CPI Rate:	0.830	
MUX Reliability:	0.962		MUX Reliability:	0.972	
Front-End Bound:	**0.2%**	**of Pipeline Slots**	**Front-End Bound:**	**0.6%**	**of Pipeline Slots**
Bad Speculation:	**0.2%**	**of Pipeline Slots**	**Bad Speculation:**	**0.2%**	**of Pipeline Slots**
Back-End Bound:	**57.5%**	**of Pipeline Slots**	**Back-End Bound:**	**64.1%**	**of Pipeline Slots**
Memory Bound:	**21.6%**	**of Pipeline Slots**	**Memory Bound:**	**35.5%**	**of Pipeline Slots**
L1 Bound:	**8.1%**	**of Clockticks**	**L1 Bound:**	**2.6%**	**of Clockticks**
DTLB Overhead:	2.1%	of Clockticks	DTLB Overhead:	1.5%	of Clockticks
Loads Blocked by Store Forwarding:	0.0%	of Clockticks	Loads Blocked by Store Forwarding:	0.0%	of Clockticks
Lock Latency:	0.0%	of Clockticks	Lock Latency:	0.0%	of Clockticks
Split Loads:	0.0%	of Clockticks	Split Loads:	0.0%	of Clockticks
4K Aliasing:	3.4%	of Clockticks	4K Aliasing:	2.6%	of Clockticks
FB Full:	0.0%	of Clockticks	FB Full:	3.4%	of Clockticks
L2 Bound:	0.0%	of Clockticks	L2 Bound:	0.0%	of Clockticks
L3 Bound:	**0.3%**	**of Clockticks**	**L3 Bound:**	**4.3%**	**of Clockticks**
DRAM Bound:	**9.9%**	**of Clockticks**	**DRAM Bound:**	**23.5%**	**of Clockticks**
Memory Bandwidth:	11.6%	of Clockticks	Memory Bandwidth:	8.5%	of Clockticks
Memory Latency:	**88.4%**	**of Clockticks**	**Memory Latency:**	**89.8%**	**of Clockticks**
Store Bound:	**0.0%**	**of Clockticks**	**Store Bound:**	**0.0%**	**of Clockticks**
Core Bound:	**35.9%**	**of Pipeline Slots**	**Core Bound:**	**28.7%**	**of Pipeline Slots**
Divider:	0.0%	of Clockticks	Divider:	0.0%	of Clockticks
Port Utilization:	**23.7%**	**of Clockticks**	**Port Utilization:**	**24.6%**	**of Clockticks**
Cycles of 0 Ports Utilized:	15.9%	of Clockticks	Cycles of 0 Ports Utilized:	33.8%	of Clockticks
Cycles of 1 Port Utilized:	22.1%	of Clockticks	Cycles of 1 Port Utilized:	21.2%	of Clockticks
Cycles of 2 Ports Utilized:	26.3%	of Clockticks	Cycles of 2 Ports Utilized:	22.5%	of Clockticks
Cycles of 3+ Ports Utilized:	**29.3%**	**of Clockticks**	**Cycles of 3+ Ports Utilized:**	**22.9%**	**of Clockticks**
Port 0:	33.3%	of Clockticks	Port 0:	24.2%	of Clockticks
Port 1:	36.5%	of Clockticks	Port 1:	29.5%	of Clockticks
Port 2:	36.9%	of Clockticks	Port 2:	29.7%	of Clockticks
Port 3:	37.7%	of Clockticks	Port 3:	30.6%	of Clockticks
Port 4:	7.9%	of Clockticks	Port 4:	6.4%	of Clockticks
Port 5:	36.2%	of Clockticks	Port 5:	29.8%	of Clockticks
Retiring:	**42.1%**	**of Pipeline Slots**	**Retiring:**	**35.0%**	**of Pipeline Slots**
Total Thread Count:	3		Total Thread Count:	3	
Paused Time:	2.919s		Paused Time:	2.883s	

Profiling Results (General Exploration)

1. Default compiler generated code (1 thread)

1 thread

Elapsed Time: 41.265s

CPU Time:	37.264s	
Memory Bound:	25.9%	of Pipeline Slots
L1 Bound:	4.4%	of Clockticks
L2 Bound:	0.3%	of Clockticks
L3 Bound:	2.9%	of Clockticks
DRAM Bound:	13.1%	of Clockticks
Memory Bandwidth:	35.5%	of Clockticks
Memory Latency:	62.4%	of Clockticks
Loads:	79,844,995,278	
Stores:	10,008,015,012	
LLC Miss Count:	12,000,360	
Average Latency (cycles):	12	
Total Thread Count:	2	
Paused Time:	2.867s	

System Bandwidth

This section provides various system bandwidth-related properties detected by the product. These values are used to define default High, Medium and Low bandwidth utilization thresholds for the Bandwidth Utilization Histogram and to scale overtime bandwidth graphs in the Bottom-up view.

Max DRAM System Bandwidth: 27 GB

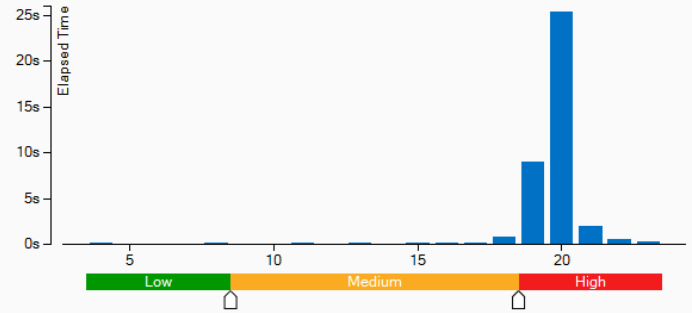
Bandwidth Utilization

Explore bandwidth utilization over time using the histogram and identify memory objects or functions with maximum contribution to the high bandwidth utilization.

Bandwidth Domain: DRAM, GB/sec

Bandwidth Utilization Histogram

This histogram displays the wall time the bandwidth was utilized by certain value. Use sliders at the bottom of the histogram to define thresholds for Low, Medium and High utilization levels. You can use these bandwidth utilization types in the Bottom-up view to group data and see all functions executed during a particular utilization type. To learn bandwidth capabilities, refer to your system specifications or run appropriate benchmarks to measure them; for example, Intel Memory Latency Checker can provide maximum achievable DRAM and QPI bandwidth.



2 threads

Elapsed Time: 34.560s

CPU Time:	60.919s	
Memory Bound:	50.5%	of Pipeline Slots
L1 Bound:	4.8%	of Clockticks
L2 Bound:	0.0%	of Clockticks
L3 Bound:	5.5%	of Clockticks
DRAM Bound:	39.2%	of Clockticks
Memory Bandwidth:	18.4%	of Clockticks
Memory Latency:	80.3%	of Clockticks
Loads:	77,600,927,958	
Stores:	9,780,014,670	
LLC Miss Count:	28,200,846	
Average Latency (cycles):	25	
Total Thread Count:	4	
Paused Time:	2.910s	

System Bandwidth

This section provides various system bandwidth-related properties detected by the product. These values are used to define default High, Medium and Low bandwidth utilization thresholds for the Bandwidth Utilization Histogram and to scale overtime bandwidth graphs in the Bottom-up view.

Max DRAM System Bandwidth: 27 GB

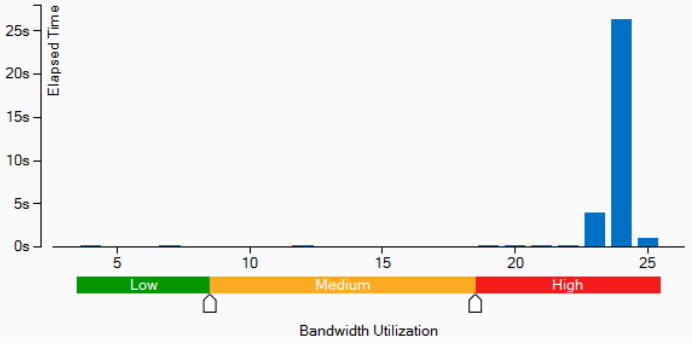
Bandwidth Utilization

Explore bandwidth utilization over time using the histogram and identify memory objects or functions with maximum contribution to the high bandwidth utilization.

Bandwidth Domain: DRAM, GB/sec

Bandwidth Utilization Histogram

This histogram displays the wall time the bandwidth was utilized by certain value. Use sliders at the bottom of the histogram to define thresholds for Low, Medium and High utilization levels. You can use these bandwidth utilization types in the Bottom-up view to group data and see all functions executed during a particular utilization type. To learn bandwidth capabilities, refer to your system specifications or run appropriate benchmarks to measure them; for example, Intel Memory Latency Checker can provide maximum achievable DRAM and QPI bandwidth.



2. _mm_store (vmovaps)

1 thread

Elapsed Time: 41.966s

CPU Time:	37.846s	
Memory Bound:	24.6%	of Pipeline Slots
L1 Bound:	5.4%	of Clockticks
L2 Bound:	0.0%	of Clockticks
L3 Bound:	2.4%	of Clockticks
DRAM Bound:	13.9%	of Clockticks
Memory Bandwidth:	16.9%	of Clockticks
Memory Latency:	83.1%	of Clockticks
Loads:	76,450,693,452	
Stores:	9,636,014,454	
LLC Miss Count:	12,000,360	
Average Latency (cycles):	12	
Total Thread Count:	4	
Paused Time:	2.919s	

System Bandwidth

This section provides various system bandwidth-related properties detected by the product. These values are used to define default High, Medium and Low bandwidth utilization thresholds for the Bandwidth Utilization Histogram and to scale overtime bandwidth graphs in the Bottom-up view.

Max DRAM System Bandwidth: 26 GB

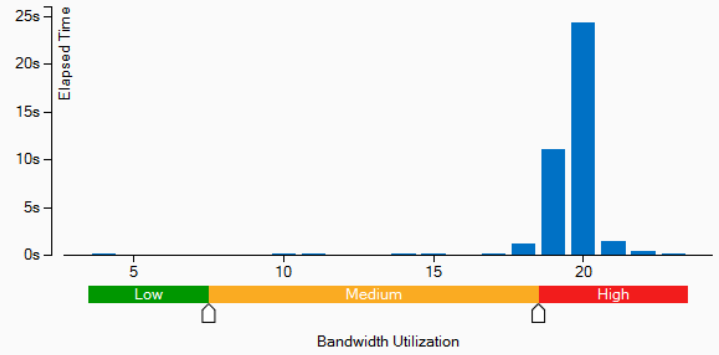
Bandwidth Utilization

Explore bandwidth utilization over time using the histogram and identify memory objects or functions with maximum contribution to the high bandwidth utilization.

Bandwidth Domain: DRAM, GB/sec

Bandwidth Utilization Histogram

This histogram displays the wall time the bandwidth was utilized by certain value. Use sliders at the bottom of the histogram to define thresholds for Low, Medium and High utilization levels. You can use these bandwidth utilization types in the Bottom-up view to group data and see all functions executed during a particular utilization type. To learn bandwidth capabilities, refer to your system specifications or run appropriate benchmarks to measure them; for example, Intel Memory Latency Checker can provide maximum achievable DRAM and QPI bandwidth.



2 threads

Elapsed Time: 34.825s

CPU Time:	61.276s	
Memory Bound:	51.4%	of Pipeline Slots
L1 Bound:	5.2%	of Clockticks
L2 Bound:	0.0%	of Clockticks
L3 Bound:	5.8%	of Clockticks
DRAM Bound:	39.3%	of Clockticks
Memory Bandwidth:	21.9%	of Clockticks
Memory Latency:	78.1%	of Clockticks
Loads:	77,247,517,356	
Stores:	9,624,014,436	
LLC Miss Count:	30,600,918	
Average Latency (cycles):	25	
Total Thread Count:	3	
Paused Time:	2.932s	

System Bandwidth

This section provides various system bandwidth-related properties detected by the product. These values are used to define default High, Medium and Low bandwidth utilization thresholds for the Bandwidth Utilization Histogram and to scale overtime bandwidth graphs in the Bottom-up view.

Max DRAM System Bandwidth: 26 GB

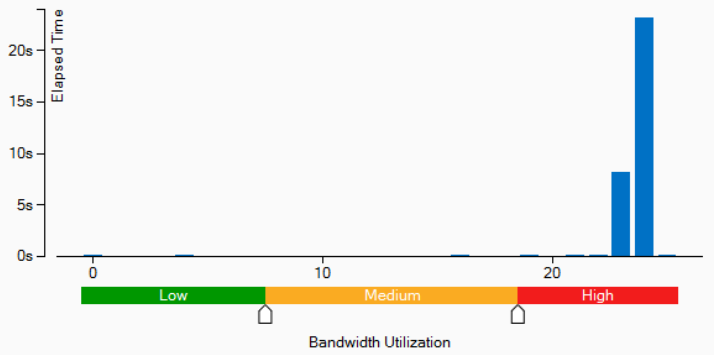
Bandwidth Utilization

Explore bandwidth utilization over time using the histogram and identify memory objects or functions with maximum contribution to the high bandwidth utilization.

Bandwidth Domain: DRAM, GB/sec

Bandwidth Utilization Histogram

This histogram displays the wall time the bandwidth was utilized by certain value. Use sliders at the bottom of the histogram to define thresholds for Low, Medium and High utilization levels. You can use these bandwidth utilization types in the Bottom-up view to group data and see all functions executed during a particular utilization type. To learn bandwidth capabilities, refer to your system specifications or run appropriate benchmarks to measure them; for example, Intel Memory Latency Checker can provide maximum achievable DRAM and QPI bandwidth.



3. _mm_stream_ps (vmovntps)

1 thread

Elapsed Time: 38.334s

CPU Time:	34.331s	
Memory Bound:	18.0%	of Pipeline Slots
L1 Bound:	3.6%	of Clockticks
L2 Bound:	0.7%	of Clockticks
L3 Bound:	0.3%	of Clockticks
DRAM Bound:	10.2%	of Clockticks
Memory Bandwidth:	10.5%	of Clockticks
Memory Latency:	88.9%	of Clockticks
Loads:	75,286,658,532	
Stores:	9,420,014,130	
LLC Miss Count:	6,600,198	
Average Latency (cycles):	10	
Total Thread Count:	2	
Paused Time:	2.929s	

System Bandwidth

This section provides various system bandwidth-related properties detected by the product. These values are used to define default High, Medium and Low bandwidth utilization thresholds for the Bandwidth Utilization Histogram and to scale overtime bandwidth graphs in the Bottom-up view.

Max DRAM System Bandwidth: 26 GB

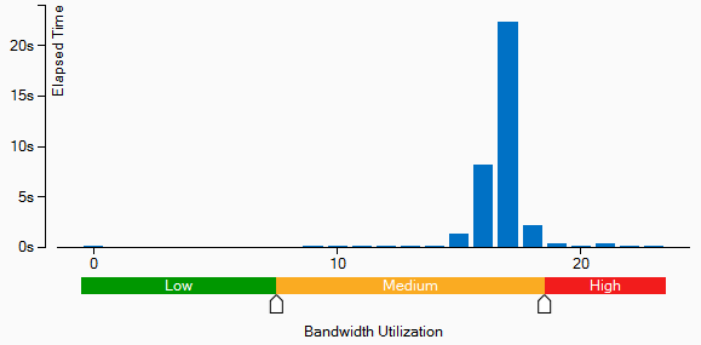
Bandwidth Utilization

Explore bandwidth utilization over time using the histogram and identify memory objects or functions with maximum contribution to the high bandwidth utilization.

Bandwidth Domain: DRAM, GB/sec

Bandwidth Utilization Histogram

This histogram displays the wall time the bandwidth was utilized by certain value. Use sliders at the bottom of the histogram to define thresholds for Low, Medium and High utilization levels. You can use these bandwidth utilization types in the Bottom-up view to group data and see all functions executed during a particular utilization type. To learn bandwidth capabilities, refer to your system specifications or run appropriate benchmarks to measure them; for example, Intel Memory Latency Checker can provide maximum achievable DRAM and QPI bandwidth.



2 threads

Elapsed Time: 27.243s

CPU Time:	46.625s	
Memory Bound:	37.7%	of Pipeline Slots
L1 Bound:	3.5%	of Clockticks
L2 Bound:	0.0%	of Clockticks
L3 Bound:	3.5%	of Clockticks
DRAM Bound:	26.5%	of Clockticks
Memory Bandwidth:	9.4%	of Clockticks
Memory Latency:	89.7%	of Clockticks
Loads:	78,896,366,820	
Stores:	9,960,014,940	
LLC Miss Count:	16,800,504	
Average Latency (cycles):	16	
Total Thread Count:	3	
Paused Time:	2.925s	

System Bandwidth

This section provides various system bandwidth-related properties detected by the product. These values are used to define default High, Medium and Low bandwidth utilization thresholds for the Bandwidth Utilization Histogram and to scale overtime bandwidth graphs in the Bottom-up view.

Max DRAM System Bandwidth: 26 GB

Bandwidth Utilization

Explore bandwidth utilization over time using the histogram and identify memory objects or functions with maximum contribution to the high bandwidth utilization.

Bandwidth Domain: DRAM, GB/sec

Bandwidth Utilization Histogram

This histogram displays the wall time the bandwidth was utilized by certain value. Use sliders at the bottom of the histogram to define thresholds for Low, Medium and High utilization levels. You can use these bandwidth utilization types in the Bottom-up view to group data and see all functions executed during a particular utilization type. To learn bandwidth capabilities, refer to your system specifications or run appropriate benchmarks to measure them; for example, Intel Memory Latency Checker can provide maximum achievable DRAM and QPI bandwidth.