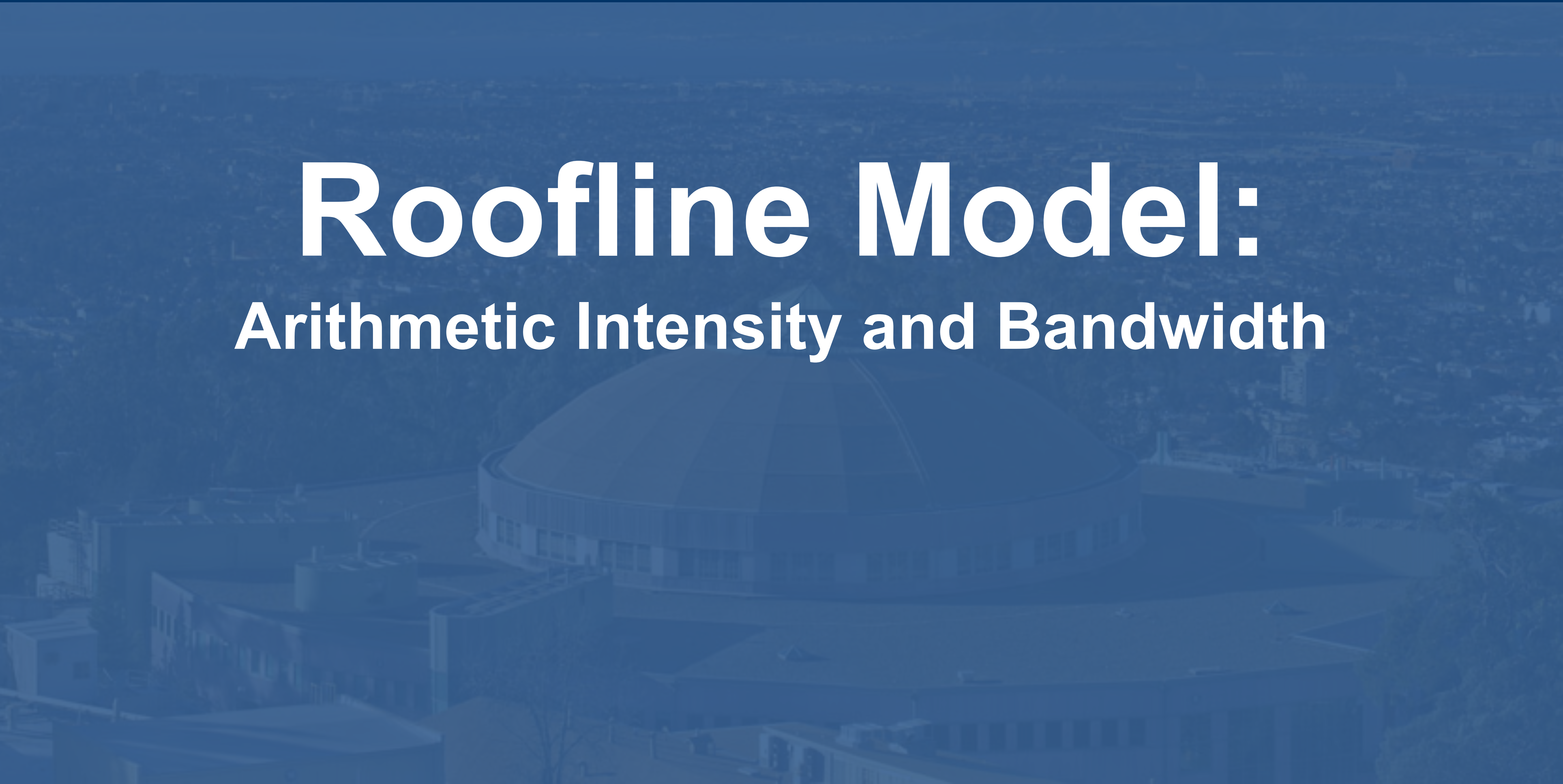
BERKELEY LAB
BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF
ENERGY

Advisor Hand-On:

Stencil Example

Samuel Williams
Computational Research Division
Lawrence Berkeley National Lab
SWWilliams@lbl.gov

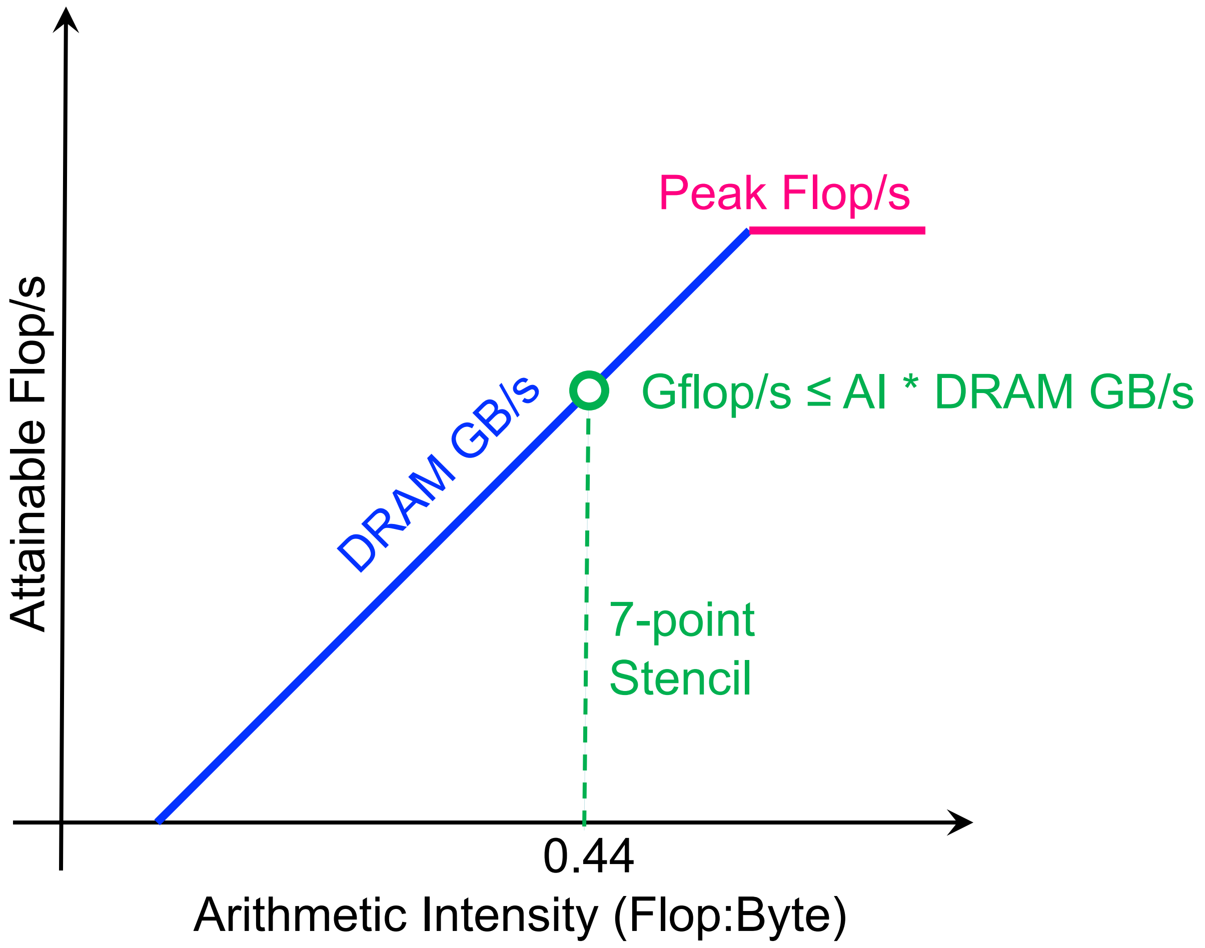


Roofline Model:

Arithmetic Intensity and Bandwidth

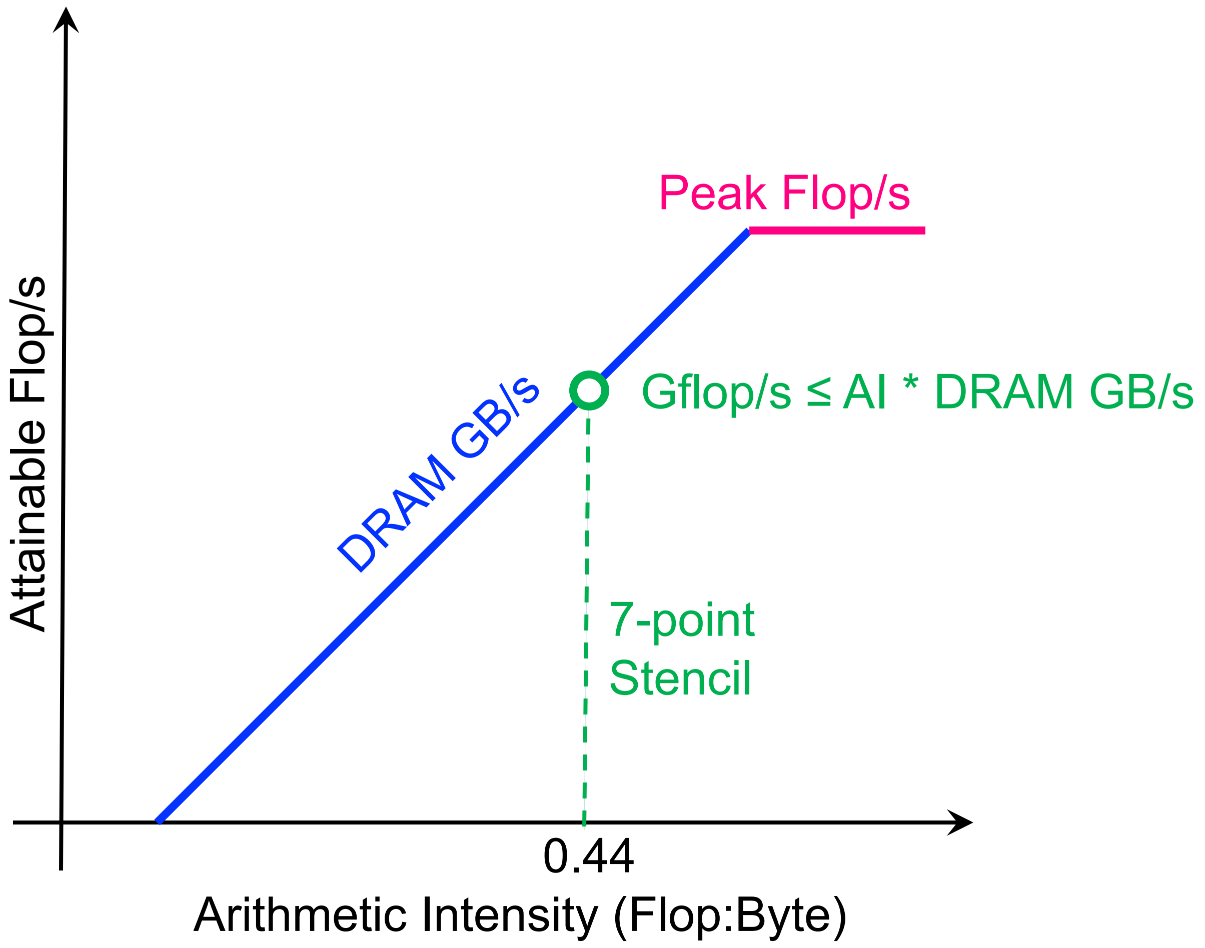
Stencil Example

- Consider our 7-point constant coefficient stencil…
  - 7 flops per point
  - 8 memory references (7 reads, 1 store)
  - An ideal cache can filter all but 1 read and 1 write per point (compulsory misses)
  - **AI = 0.44 flops per byte**
- Actual performance can suffer from…
  - Capacity misses in the L2/L3
  - Failure to vectorize / optimally vectorize
  - Superfluous write allocations



Stencil Example in Advisor

- Let's walk thru a series of progressively more optimized stencil implementations
- We'll use Advisor's Integrated Roofline functionality[1] to highlight how optimization changes AI at different levels of the cache



[1] Technology Preview, not in official product roadmap so far. This version will be made available during the hands-on component of this tutorial.

Naive Version (ver0)

- Naive version (ver0)…
  - 7-point stencil
  - 512^3 grid (padded to 514^3)
  - OpenMP on outer loop
  - ➢ **pragma to prevent vectorization**
  - ➢ **8MB/thread cache working set**
  - ➢ **Low DRAM/L3 Arithmetic Intensity**

```
#pragma omp parallel for
for(k=1;k<dim+1;k++){
for(j=1;j<dim+1;j++){
  #pragma novector
  for(i=1;i<dim+1;i++){
    int ijk = i*iStride + j*jStride + k*kStride;
    new[ijk] = -6.0*old[ijk        ]
                  + old[ijk-iStride]
                  + old[ijk+iStride]
                  + old[ijk-jStride]
                  + old[ijk+jStride]
                  + old[ijk-kStride]
                  + old[ijk+kStride];
}}}
```

Baseline Version (ver1)

- Eliminate novector
- Simplify Address calculation
- Still has issues…
  - **8MB/thread cache working set**
  - **8MB*16 threads > L3**
  - **Low DRAM/L3 Arithmetic Intensity**

```
#pragma omp parallel for
for(k=1;k<dim+1;k++){
for(j=1;j<dim+1;j++){
for(i=1;i<dim+1;i++){
  int ijk = i + j*jStride + k*kStride;
  new[ijk] = -6.0*old[ijk        ]
                 + old[ijk-iStride]
                 + old[ijk+iStride]
                 + old[ijk-jStride]
                 + old[ijk+jStride]
                 + old[ijk-kStride]
                 + old[ijk+kStride];
}}}
```

Tiled Version (ver2)

- Apply 2D (8x8x512) loop tiling…
  - ➢ **Cache working set reduced to ~164KB**
  - ➢ **Should improve L3 and DRAM AI (but not L2)**

```
#pragma omp parallel for schedule(static,1)
for(tile=0;tile<jTiles*kTiles;tile++){
  int kLo = TILE*(tile/jTiles);
  int jLo = TILE*(tile%jTiles);
  for(k=kLo;k<kLo+TILE;k++){
  for(j=jLo;j<jLo+TILE;j++){
  for(i=0;i<dim;i++){
    int ijk = i + j*jStride + k*kStride;
    new[ijk] = -6.0*old[ijk        ]
                  + old[ijk-1      ]
                  + old[ijk+1      ]
                  + old[ijk-jStride]
                  + old[ijk+jStride]
                  + old[ijk-kStride]
                  + old[ijk+kStride];
}}}
```

Padded Version (ver3)

- Same code, but
  - Unit-stride padded to 520 (multiple of 8)
  - First non-ghost zone aligned to 64B
  - Should facilitate vectorization / eliminate peel and remainder loops
- Larger dimension = larger array…
  - ➢ **AIs should be a 1% lower**

```
#pragma omp parallel for schedule(static,1)
for(tile=0;tile<jTiles*kTiles;tile++){
  int kLo = TILE*(tile/jTiles);
  int jLo = TILE*(tile%jTiles);
  for(k=kLo;k<kLo+TILE;k++){
  for(j=jLo;j<jLo+TILE;j++){
  for(i=0;i<dim;i++){
    int ijk = i + j*jStride + k*kStride;
    new[ijk] = -6.0*old[ijk        ]
                  + old[ijk-1      ]
                  + old[ijk+1      ]
                  + old[ijk-jStride]
                  + old[ijk+jStride]
                  + old[ijk-kStride]
                  + old[ijk+kStride];
}}}
```

SIMD/Cache Bypass Version (ver4)

- Once data has been aligned, force SIMDization and alignment
- Write allocate cache is inflating total data movement by 50%...
  - Use non-temporal store to bypass cache
  - ➢ **DRAM AI should be a 50% higher and close to the ideal 0.44 flop/byte**

```
#pragma omp parallel for schedule(static,1)
for(tile=0;tile<jTiles*kTiles;tile++){
  int kLo = TILE*(tile/jTiles);
  int jLo = TILE*(tile%jTiles);
  for(k=kLo;k<kLo+TILE;k++){
  for(j=jLo;j<jLo+TILE;j++){
    #pragma omp simd aligned(new,old:64)
    #pragma vector nontemporal
    for(i=0;i<jStride;i++){
      int ijk = i + j*jStride + k*kStride;
      new[ijk] = -6.0*old[ijk        ]
                    + old[ijk-1      ]
                    + old[ijk+1      ]
                    + old[ijk-jStride]
                    + old[ijk+jStride]
                    + old[ijk-kStride]
                    + old[ijk+kStride];
}}}
```

Using Advisor

Open https://nxcloud01.nersc.gov in Browser

- Login using your temporary account.
- Create a shell on cori using your temporary account
- Load Advisor

```
module load advisor/2018.integrated_roofline
```

- We’ve pre-collected/surveyed the stencil benchmark for you:

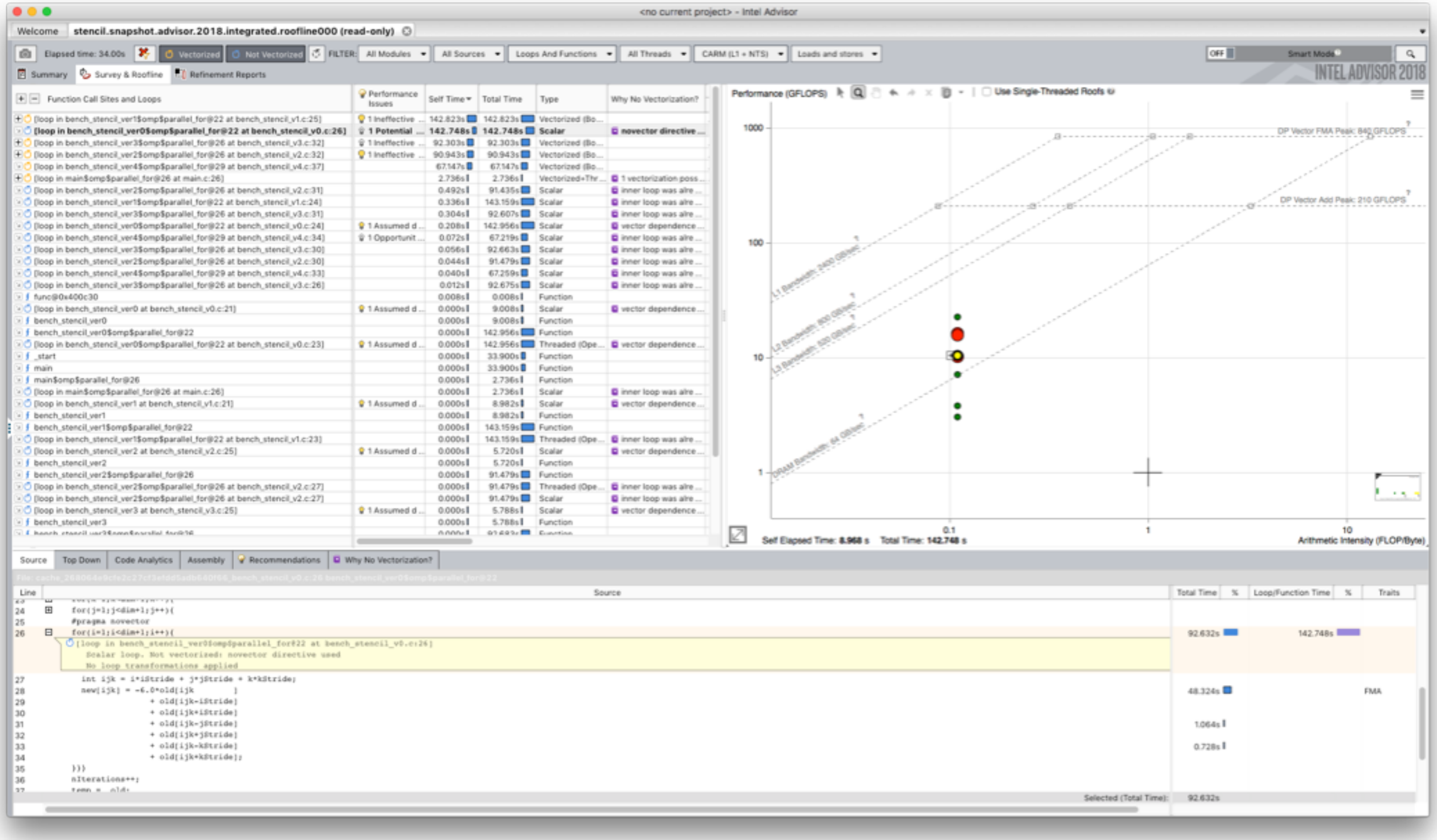
```
cp $ADVISOR_XE_2018_DIR/ECP-meeting-tutorial/* ~
```

- Launch advisor:

```
advixe-gui
```

Ver0

- Advisor notes that ver0 didn’t vectorize



<no current project> - Intel Advisor

Welcome | stencil.snapshot.advisor.2018.integrated.roofline000 (read-only)

Elapsed time: 34.00s | Vectorized | Not Vectorized | FILTER: All Modules | All Sources | Loops And Functions | All Threads | CARM (L1 + NTS) | Loads and stores | OFF Smart Mode

Summary | Survey & Roofline | Refinement Reports

Function Call Sites and Loops	Performance Issues	Self Time	Total Time	Type	Why No Vectorization?
[loop in bench_stencil_ver1$omp$parallel_for@22 at bench_stencil_v1.c:25]	1 Ineffective ...	142.823s	142.823s	Vectorized (Bo...	
[loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:26]	**1 Potential ...**	**142.748s**	**142.748s**	**Scalar**	**novector directive ...**
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:32]	1 Ineffective ...	92.303s	92.303s	Vectorized (Bo...	
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:32]	1 Ineffective ...	90.943s	90.943s	Vectorized (Bo...	
[loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:37]		67.147s	67.147s	Vectorized (Bo...	
[loop in main$omp$parallel_for@26 at main.c:26]		2.736s	2.736s	Vectorized+Thr...	1 vectorization poss ...
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:31]		0.492s	91.435s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver1$omp$parallel_for@22 at bench_stencil_v1.c:24]		0.336s	143.159s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:31]		0.304s	92.607s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:24]	1 Assumed d ...	0.208s	142.956s	Scalar	vector dependence ...
[loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:34]	1 Opportunit ...	0.072s	67.219s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:30]		0.056s	92.663s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:30]		0.044s	91.479s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:33]		0.040s	67.259s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:26]		0.012s	92.675s	Scalar	inner loop was alre ...
func@0x400c30		0.008s	0.008s	Function	
[loop in bench_stencil_ver0 at bench_stencil_v0.c:21]	1 Assumed d ...	0.000s	9.008s	Scalar	vector dependence ...
bench_stencil_ver0		0.000s	9.008s	Function	
bench_stencil_ver0$omp$parallel_for@22		0.000s	142.956s	Function	
[loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:23]	1 Assumed d ...	0.000s	142.956s	Threaded (Ope...	vector dependence ...
_start		0.000s	33.900s	Function	
main		0.000s	33.900s	Function	
main$omp$parallel_for@26		0.000s	2.736s	Function	
[loop in main$omp$parallel_for@26 at main.c:26]		0.000s	2.736s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver1 at bench_stencil_v1.c:21]	1 Assumed d ...	0.000s	8.982s	Scalar	vector dependence ...
bench_stencil_ver1		0.000s	8.982s	Function	
bench_stencil_ver1$omp$parallel_for@22		0.000s	143.159s	Function	
[loop in bench_stencil_ver1$omp$parallel_for@22 at bench_stencil_v1.c:23]		0.000s	143.159s	Threaded (Ope...	inner loop was alre ...
[loop in bench_stencil_ver2 at bench_stencil_v2.c:25]	1 Assumed d ...	0.000s	5.720s	Scalar	vector dependence ...
bench_stencil_ver2		0.000s	5.720s	Function	
bench_stencil_ver2$omp$parallel_for@26		0.000s	91.479s	Function	
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:27]		0.000s	91.479s	Threaded (Ope...	inner loop was alre ...
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:27]		0.000s	91.479s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver3 at bench_stencil_v3.c:25]	1 Assumed d ...	0.000s	5.788s	Scalar	vector dependence ...
bench_stencil_ver3		0.000s	5.788s	Function	
bench_stencil_ver3$omp$parallel_for@26		0.000s	91.693s	Function	

Performance (GFLOPS)
Use Single-Threaded Roofs
1000
100
10
1
0.1
1
10
Arithmetic Intensity (FLOP/Byte)
DP Vector FMA Peak: 840 GFLOPS
DP Vector Add Peak: 210 GFLOPS
L1 Bandwidth: 2400 GB/sec
L2 Bandwidth: 800 GB/sec
L3 Bandwidth: 520 GB/sec
DRAM Bandwidth: 64 GB/sec
INTEL ADVISOR 2018

Self Elapsed Time: **8.968** s Total Time: **142.748** s

Source | Top Down | Code Analytics | Assembly | Recommendations | Why No Vectorization?

File: cache_268064e9cfe2c27cf3efdd5adb640f66_bench_stencil_v0.c:26 bench_stencil_ver0$omp$parallel_for@22

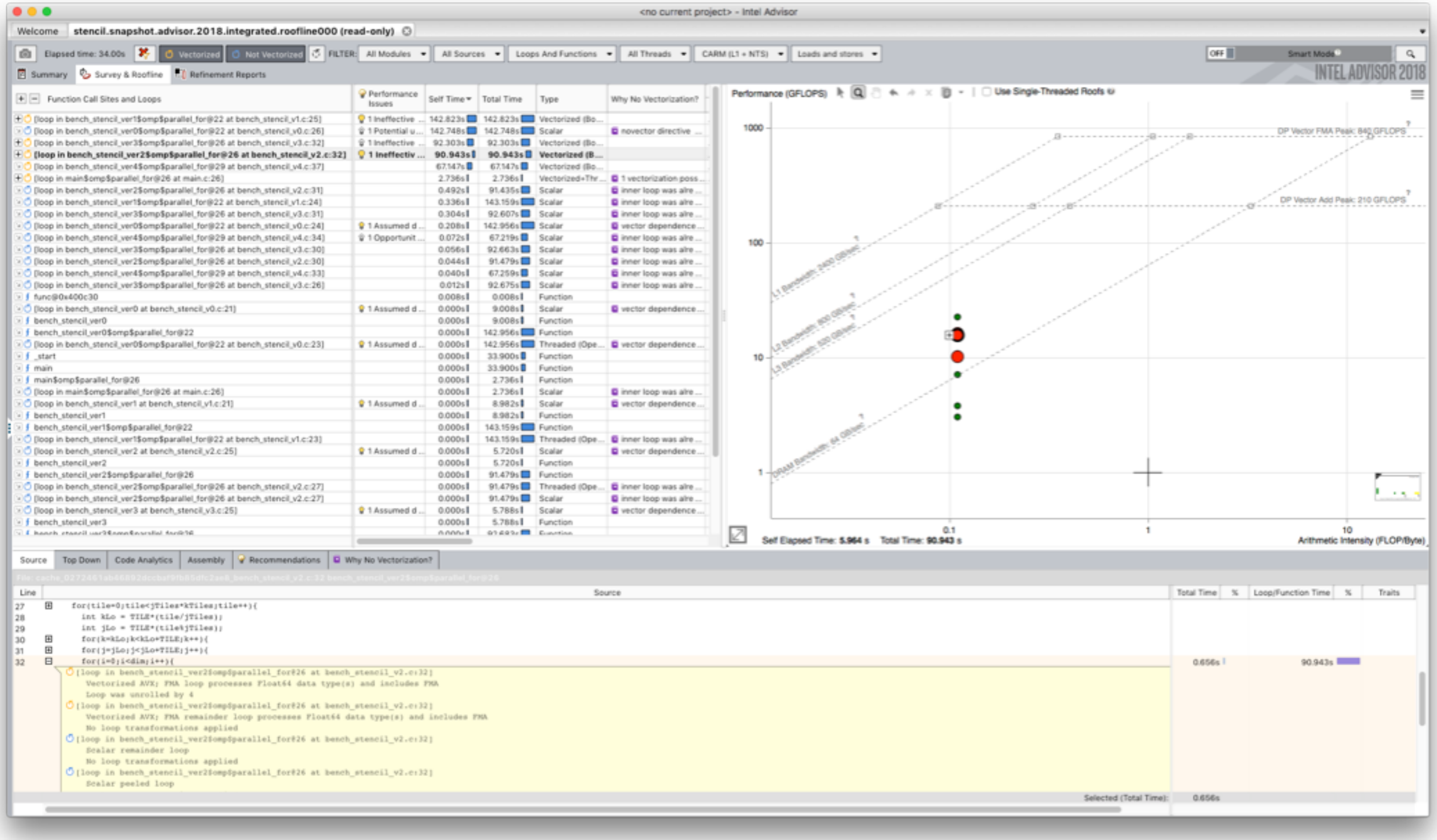
```
for(k=1;k<dim+1;k++){
for(j=1;j<dim+1;j++){
#pragma novector
for(i=1;i<dim+1;i++){
  [loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:26]
   Scalar loop. Not vectorized: novector directive used
   No loop transformations applied
  int ijk = i*iStride + j*jStride + k*kStride;
  new[ijk] = -6.0*old[ijk         ]
                 + old[ijk-iStride]
                 + old[ijk+iStride]
                 + old[ijk-jStride]
                 + old[ijk+jStride]
                 + old[ijk-kStride]
                 + old[ijk+kStride];
}}}
nIterations++;
temp = old;
```

Line	Total Time	%	Loop/Function Time	%	Traits
26	92.632s		142.748s		
28	48.324s				FMA
31	1.064s				
33	0.728s				

Selected (Total Time): 92.632s

Ver2 (tiled)

- Ver2 showed no improvement in L1 AI.
- All variants present the same number of loads/stores to the L1 cache (compiler failed to register block)



<no current project> - Intel Advisor
Welcome
stencil.snapshot.advisor.2018.integrated.roofline000 (read-only)
Vectorized
Not Vectorized
Smart Mode
OFF
INTEL ADVISOR 2018
Summary
Survey & Roofline
Refinement Reports
Performance (GFLOPS)
Use Single-Threaded Roofs
1000
100
10
1
0.1
1
10
Arithmetic Intensity (FLOP/Byte)
DP Vector FMA Peak: 840 GFLOPS
DP Vector Add Peak: 210 GFLOPS
L1 Bandwidth: 2400 GB/sec
L2 Bandwidth: 800 GB/sec
L3 Bandwidth: 520 GB/sec
DRAM Bandwidth: 64 GB/sec
Source
Top Down
Code Analytics
Assembly
Recommendations
Why No Vectorization?

Elapsed time: 34.00s

FILTER: All Modules | All Sources | Loops And Functions | All Threads | CARM (L1 + NTS) | Loads and stores

Function Call Sites and Loops	Performance Issues	Self Time	Total Time	Type	Why No Vectorization?
[loop in bench_stencil_ver1$omp$parallel_for@22 at bench_stencil_v1.c:25]	1 Ineffective ...	142.823s	142.823s	Vectorized (Bo...	
[loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:26]	1 Potential u...	142.748s	142.748s	Scalar	novector directive ...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:32]	1 Ineffective ...	92.303s	92.303s	Vectorized (Bo...	
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:32]	**1 Ineffectiv ...**	**90.943s**	**90.943s**	**Vectorized (B...**	
[loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:37]		67.147s	67.147s	Vectorized (Bo...	
[loop in main$omp$parallel_for@26 at main.c:26]		2.736s	2.736s	Vectorized+Thr...	1 vectorization poss ...
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:31]		0.492s	91.435s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver1$omp$parallel_for@22 at bench_stencil_v1.c:24]		0.336s	143.159s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:31]		0.304s	92.607s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:24]	1 Assumed d ...	0.208s	142.956s	Scalar	vector dependence ...
[loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:34]	1 Opportunit ...	0.072s	67.219s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:30]		0.056s	92.663s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:30]		0.044s	91.479s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:33]		0.040s	67.259s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:26]		0.012s	92.675s	Scalar	inner loop was alre ...
func@0x400c30		0.008s	0.008s	Function	
[loop in bench_stencil_ver0 at bench_stencil_v0.c:21]	1 Assumed d ...	0.000s	9.008s	Scalar	vector dependence ...
bench_stencil_ver0		0.000s	9.008s	Function	
bench_stencil_ver0$omp$parallel_for@22		0.000s	142.956s	Function	
[loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:23]	1 Assumed d ...	0.000s	142.956s	Threaded (Ope...	vector dependence ...
_start		0.000s	33.900s	Function	
main		0.000s	33.900s	Function	
main$omp$parallel_for@26		0.000s	2.736s	Function	
[loop in main$omp$parallel_for@26 at main.c:26]		0.000s	2.736s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver1 at bench_stencil_v1.c:21]	1 Assumed d ...	0.000s	8.982s	Scalar	vector dependence ...
bench_stencil_ver1		0.000s	8.982s	Function	
bench_stencil_ver1$omp$parallel_for@22		0.000s	143.159s	Function	
[loop in bench_stencil_ver1$omp$parallel_for@22 at bench_stencil_v1.c:23]		0.000s	143.159s	Threaded (Ope...	inner loop was alre ...
[loop in bench_stencil_ver2 at bench_stencil_v2.c:25]	1 Assumed d ...	0.000s	5.720s	Scalar	vector dependence ...
bench_stencil_ver2		0.000s	5.720s	Function	
bench_stencil_ver2$omp$parallel_for@26		0.000s	91.479s	Function	
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:27]		0.000s	91.479s	Threaded (Ope...	inner loop was alre ...
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:27]		0.000s	91.479s	Scalar	inner loop was alre ...
[loop in bench_stencil_ver3 at bench_stencil_v3.c:25]	1 Assumed d ...	0.000s	5.788s	Scalar	vector dependence ...
bench_stencil_ver3		0.000s	5.788s	Function	
bench_stencil_ver3$omp$parallel_for@26		0.000s	91.683s	Function	

Self Elapsed Time: 5.964 s Total Time: 90.943 s

File: cache_0272461ab46892dccbaf9fb85dfc2ae8_bench_stencil_v2.c:32 bench_stencil_ver2$omp$parallel_for@26

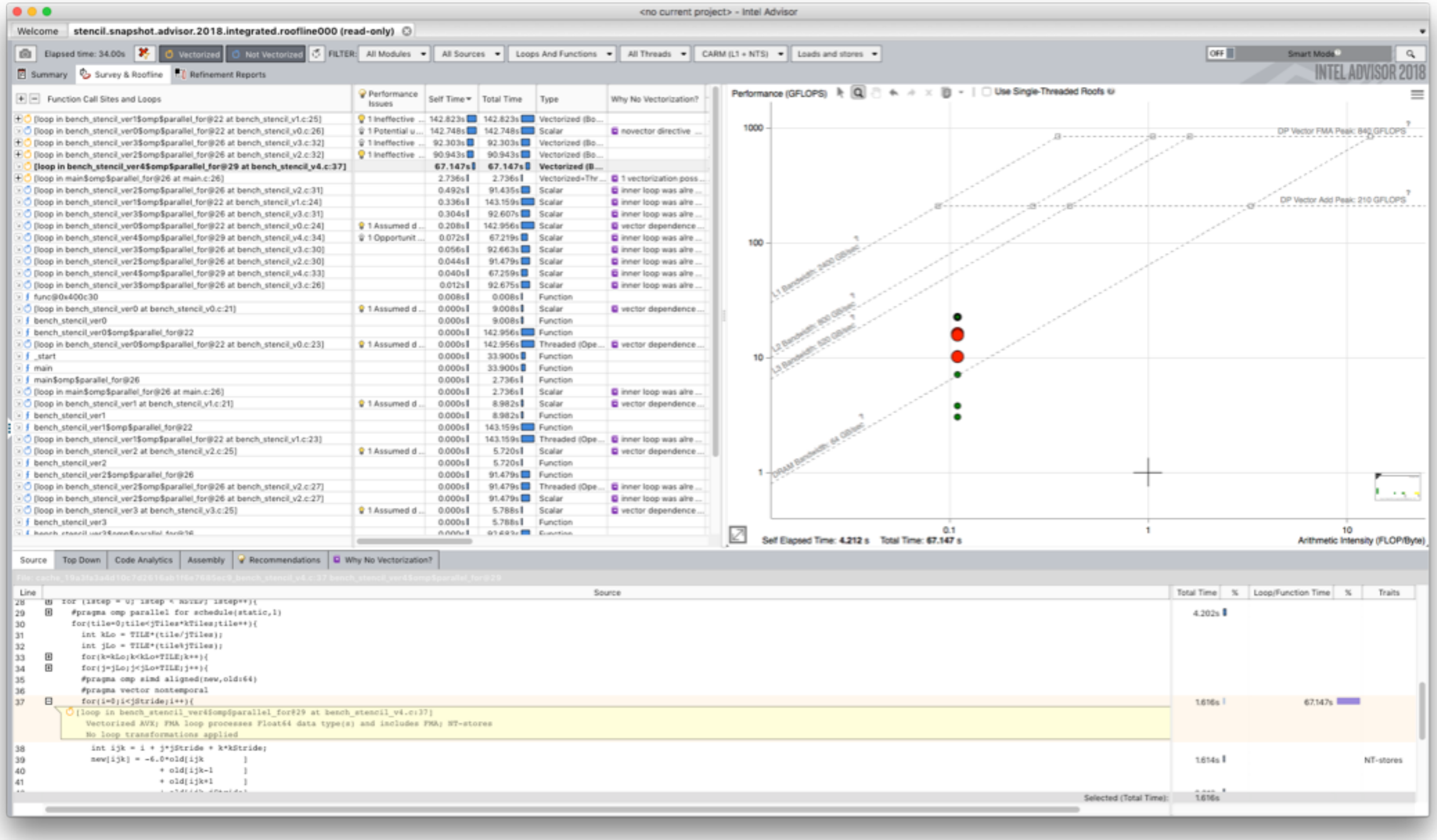
Line	Source	Total Time	%	Loop/Function Time	%	Traits
32	for(i=0;i<dim;i++){	0.656s		90.943s		

```
for(tile=0;tile<jTiles*kTiles;tile++){
  int kLo = TILE*(tile/jTiles);
  int jLo = TILE*(tile%jTiles);
  for(k=kLo;k<kLo+TILE;k++){
  for(j=jLo;j<jLo+TILE;j++){
  for(i=0;i<dim;i++){
   [loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:32]
     Vectorized AVX; FMA loop processes Float64 data type(s) and includes FMA
     Loop was unrolled by 4
   [loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:32]
     Vectorized AVX; FMA remainder loop processes Float64 data type(s) and includes FMA
     No loop transformations applied
   [loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:32]
     Scalar remainder loop
     No loop transformations applied
   [loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:32]
     Scalar peeled loop
```

Selected (Total Time): 0.656s

Ver4 (cache bypass)

- Advisor notes the presence of NTS
- L1 AI is flops divided by load+store+NTS
- Hence, NTS does not change L1 AI



Function Call Sites and Loops	Performance Issues	Self Time	Total Time	Type	Why No Vectorization?
[loop in bench_stencil_ver1$omp$parallel_for@22 at bench_stencil_v1.c:25]	1 Ineffective ...	142.823s	142.823s	Vectorized (Bo...	
[loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:26]	1 Potential u...	142.748s	142.748s	Scalar	novector directive ...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:32]	1 Ineffective ...	92.303s	92.303s	Vectorized (Bo...	
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:32]	1 Ineffective ...	90.943s	90.943s	Vectorized (Bo...	
[loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:37]		**67.147s**	**67.147s**	**Vectorized (B...**	
[loop in main$omp$parallel_for@26 at main.c:26]		2.736s	2.736s	Vectorized+Thr...	1 vectorization poss...
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:31]		0.492s	91.435s	Scalar	inner loop was alre...
[loop in bench_stencil_ver1$omp$parallel_for@22 at bench_stencil_v1.c:24]		0.336s	143.159s	Scalar	inner loop was alre...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:31]		0.304s	92.607s	Scalar	inner loop was alre...
[loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:24]	1 Assumed d...	0.208s	142.956s	Scalar	vector dependence...
[loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:34]	1 Opportunit...	0.072s	67.219s	Scalar	inner loop was alre...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:30]		0.056s	92.663s	Scalar	inner loop was alre...
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:30]		0.044s	91.479s	Scalar	inner loop was alre...
[loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:33]		0.040s	67.259s	Scalar	inner loop was alre...
[loop in bench_stencil_ver3$omp$parallel_for@26 at bench_stencil_v3.c:26]		0.012s	92.675s	Scalar	inner loop was alre...
func@0x400c30		0.008s	0.008s	Function	
[loop in bench_stencil_ver0 at bench_stencil_v0.c:21]	1 Assumed d...	0.000s	9.008s	Scalar	vector dependence...
bench_stencil_ver0		0.000s	9.008s	Function	
bench_stencil_ver0$omp$parallel_for@22		0.000s	142.956s	Function	
[loop in bench_stencil_ver0$omp$parallel_for@22 at bench_stencil_v0.c:23]	1 Assumed d...	0.000s	142.956s	Threaded (Ope...	vector dependence...
_start		0.000s	33.900s	Function	
main		0.000s	33.900s	Function	
main$omp$parallel_for@26		0.000s	2.736s	Function	
[loop in main$omp$parallel_for@26 at main.c:26]		0.000s	2.736s	Scalar	inner loop was alre...
[loop in bench_stencil_ver1 at bench_stencil_v1.c:21]	1 Assumed d...	0.000s	8.982s	Scalar	vector dependence...
bench_stencil_ver1		0.000s	8.982s	Function	
bench_stencil_ver1$omp$parallel_for@22		0.000s	143.159s	Function	
[loop in bench_stencil_ver1$omp$parallel_for@22 at bench_stencil_v1.c:23]		0.000s	143.159s	Threaded (Ope...	inner loop was alre...
[loop in bench_stencil_ver2 at bench_stencil_v2.c:25]	1 Assumed d...	0.000s	5.720s	Scalar	vector dependence...
bench_stencil_ver2		0.000s	5.720s	Function	
bench_stencil_ver2$omp$parallel_for@26		0.000s	91.479s	Function	
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:27]		0.000s	91.479s	Threaded (Ope...	inner loop was alre...
[loop in bench_stencil_ver2$omp$parallel_for@26 at bench_stencil_v2.c:27]		0.000s	91.479s	Scalar	inner loop was alre...
[loop in bench_stencil_ver3 at bench_stencil_v3.c:25]	1 Assumed d...	0.000s	5.788s	Scalar	vector dependence...
bench_stencil_ver3		0.000s	5.788s	Function	
bench_stencil_ver3$omp$parallel_for@26		0.000s	91.683s	Function	

Performance (GFLOPS)
Use Single-Threaded Roofs
1000
100
10
1
DP Vector FMA Peak: 840 GFLOPS
DP Vector Add Peak: 210 GFLOPS
L1 Bandwidth: 2400 GB/sec
L2 Bandwidth: 800 GB/sec
L3 Bandwidth: 520 GB/sec
DRAM Bandwidth: 64 GB/sec
0.1
1
10
Arithmetic Intensity (FLOP/Byte)
Self Elapsed Time: 4.212 s
Total Time: 67.147 s
Source
Top Down
Code Analytics
Assembly
Recommendations
Why No Vectorization?

File: cache_19a3fa3a4d10c7d2616ab1f6e7685ec9_bench_stencil_v4.c:37 bench_stencil_ver4$omp$parallel_for@29

```
for (istep = 0; istep < NSTEP; istep++){
  #pragma omp parallel for schedule(static,1)
  for(tile=0;tile<jTiles*kTiles;tile++){
    int kLo = TILE*(tile/jTiles);
    int jLo = TILE*(tile%jTiles);
    for(k=kLo;k<kLo+TILE;k++){
    for(j=jLo;j<jLo+TILE;j++){
    #pragma omp simd aligned(new,old:64)
    #pragma vector nontemporal
    for(i=0;i<jStride;i++){
   [loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:37]
     Vectorized AVX; FMA loop processes Float64 data type(s) and includes FMA; NT-stores
     No loop transformations applied
      int ijk = i + j*jStride + k*kStride;
      new[ijk] = -6.0*old[ijk        ]
                    + old[ijk-1      ]
                    + old[ijk+1      ]
```

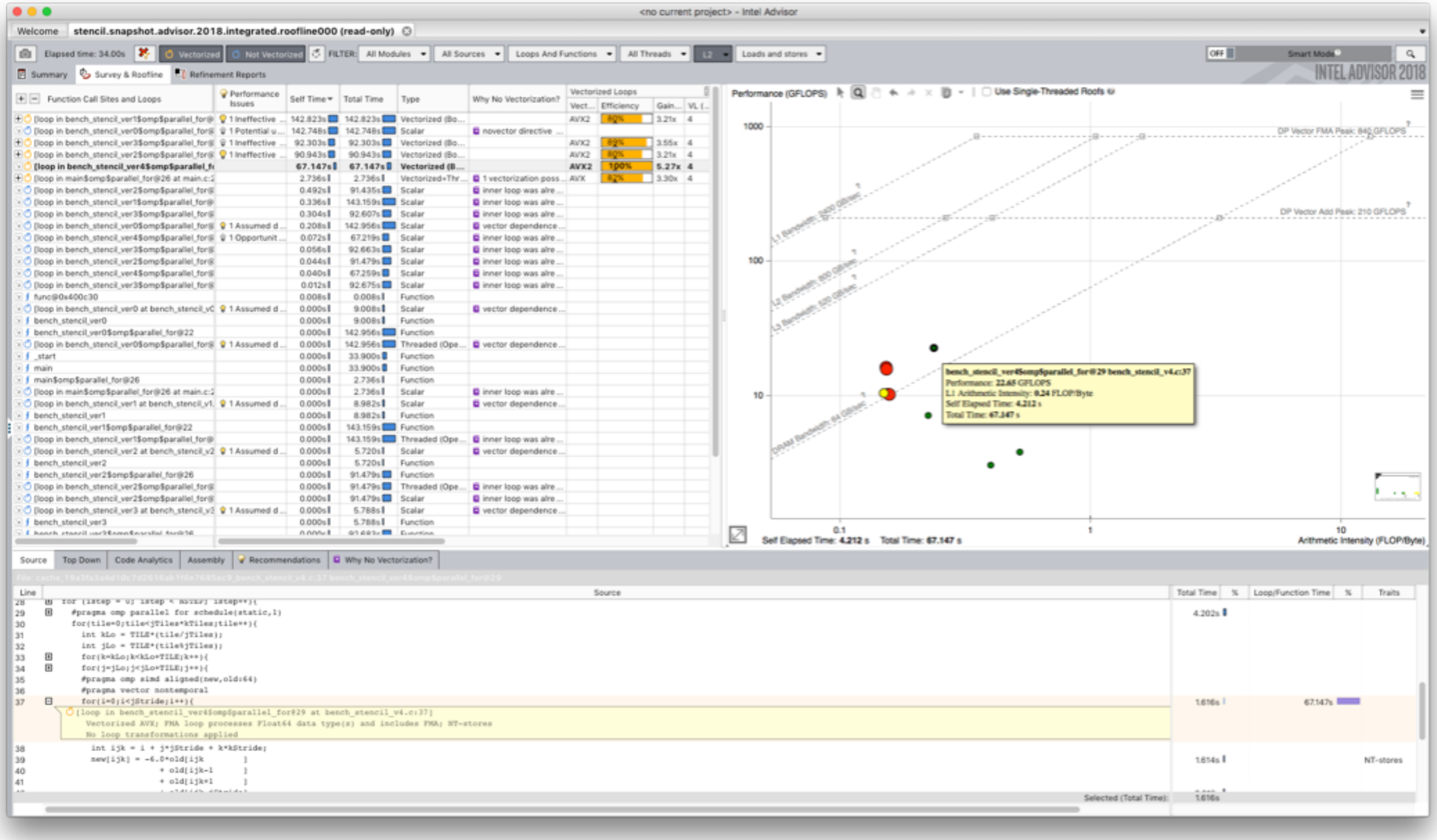
Line	Total Time	%	Loop/Function Time	%	Traits
29	4.202s				
37	1.616s		67.147s		
39	1.614s				NT-stores

Selected (Total Time): 1.616s

L2 AI

- Observe tiling did not change the L2 AI
- Why? Although working set is smaller, it still doesn’t fit in the L1 (164KB > 32KB)

- Ver4 has higher AI.
- Why? NTS bypassed the L2.



Function Call Sites and Loops	Performance Issues	Self Time	Total Time	Type	Why No Vectorization?	Vect...	Efficiency	Gain...	VL (...
[loop in bench_stencil_ver1$omp$parallel_for@	1 Ineffective ...	142.823s	142.823s	Vectorized (Bo...		AVX2	80%	3.21x	4
[loop in bench_stencil_ver0$omp$parallel_for@	1 Potential u...	142.748s	142.748s	Scalar	novector directive ...				
[loop in bench_stencil_ver3$omp$parallel_for@	1 Ineffective ...	92.303s	92.303s	Vectorized (Bo...		AVX2	89%	3.55x	4
[loop in bench_stencil_ver2$omp$parallel_for@	1 Ineffective ...	90.943s	90.943s	Vectorized (Bo...		AVX2	80%	3.21x	4
[loop in bench_stencil_ver4$omp$parallel_fo		**67.147s**	**67.147s**	**Vectorized (B...**		**AVX2**	**100%**	**5.27x**	**4**
[loop in main$omp$parallel_for@26 at main.c:2		2.736s	2.736s	Vectorized+Thr...	1 vectorization poss...	AVX	82%	3.30x	4
[loop in bench_stencil_ver2$omp$parallel_for@		0.492s	91.435s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver1$omp$parallel_for@		0.336s	143.159s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3$omp$parallel_for@		0.304s	92.607s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver0$omp$parallel_for@	1 Assumed d...	0.208s	142.956s	Scalar	vector dependence...				
[loop in bench_stencil_ver4$omp$parallel_for@	1 Opportunit...	0.072s	67.219s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3$omp$parallel_for@		0.056s	92.663s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver2$omp$parallel_for@		0.044s	91.479s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver4$omp$parallel_for@		0.040s	67.259s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3$omp$parallel_for@		0.012s	92.675s	Scalar	inner loop was alre...				
func@0x400c30		0.008s	0.008s	Function					
[loop in bench_stencil_ver0 at bench_stencil_v0	1 Assumed d...	0.000s	9.008s	Scalar	vector dependence...				
bench_stencil_ver0		0.000s	9.008s	Function					
bench_stencil_ver0$omp$parallel_for@22		0.000s	142.956s	Function					
[loop in bench_stencil_ver0$omp$parallel_for@	1 Assumed d...	0.000s	142.956s	Threaded (Ope...	vector dependence...				
_start		0.000s	33.900s	Function					
main		0.000s	33.900s	Function					
main$omp$parallel_for@26		0.000s	2.736s	Function					
[loop in main$omp$parallel_for@26 at main.c:2		0.000s	2.736s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver1 at bench_stencil_v1.	1 Assumed d...	0.000s	8.982s	Scalar	vector dependence...				
bench_stencil_ver1		0.000s	8.982s	Function					
bench_stencil_ver1$omp$parallel_for@22		0.000s	143.159s	Function					
[loop in bench_stencil_ver1$omp$parallel_for@		0.000s	143.159s	Threaded (Ope...	inner loop was alre...				
[loop in bench_stencil_ver2 at bench_stencil_v2	1 Assumed d...	0.000s	5.720s	Scalar	vector dependence...				
bench_stencil_ver2		0.000s	5.720s	Function					
bench_stencil_ver2$omp$parallel_for@26		0.000s	91.479s	Function					
[loop in bench_stencil_ver2$omp$parallel_for@		0.000s	91.479s	Threaded (Ope...	inner loop was alre...				
[loop in bench_stencil_ver2$omp$parallel_for@		0.000s	91.479s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3 at bench_stencil_v3	1 Assumed d...	0.000s	5.788s	Scalar	vector dependence...				
bench_stencil_ver3		0.000s	5.788s	Function					

bench_stencil_ver4$omp$parallel_for@29 bench_stencil_v4.c:37
Performance: 22.65 GFLOPS
L1 Arithmetic Intensity: 0.24 FLOP/Byte
Self Elapsed Time: 4.212 s
Total Time: 67.147 s

Self Elapsed Time: 4.212 s Total Time: 67.147 s

File: cache_19a3fa3add10c7d2616ab1f6e7685ec9_bench_stencil_v4.c:37 bench_stencil_ver4$omp$parallel_for@29

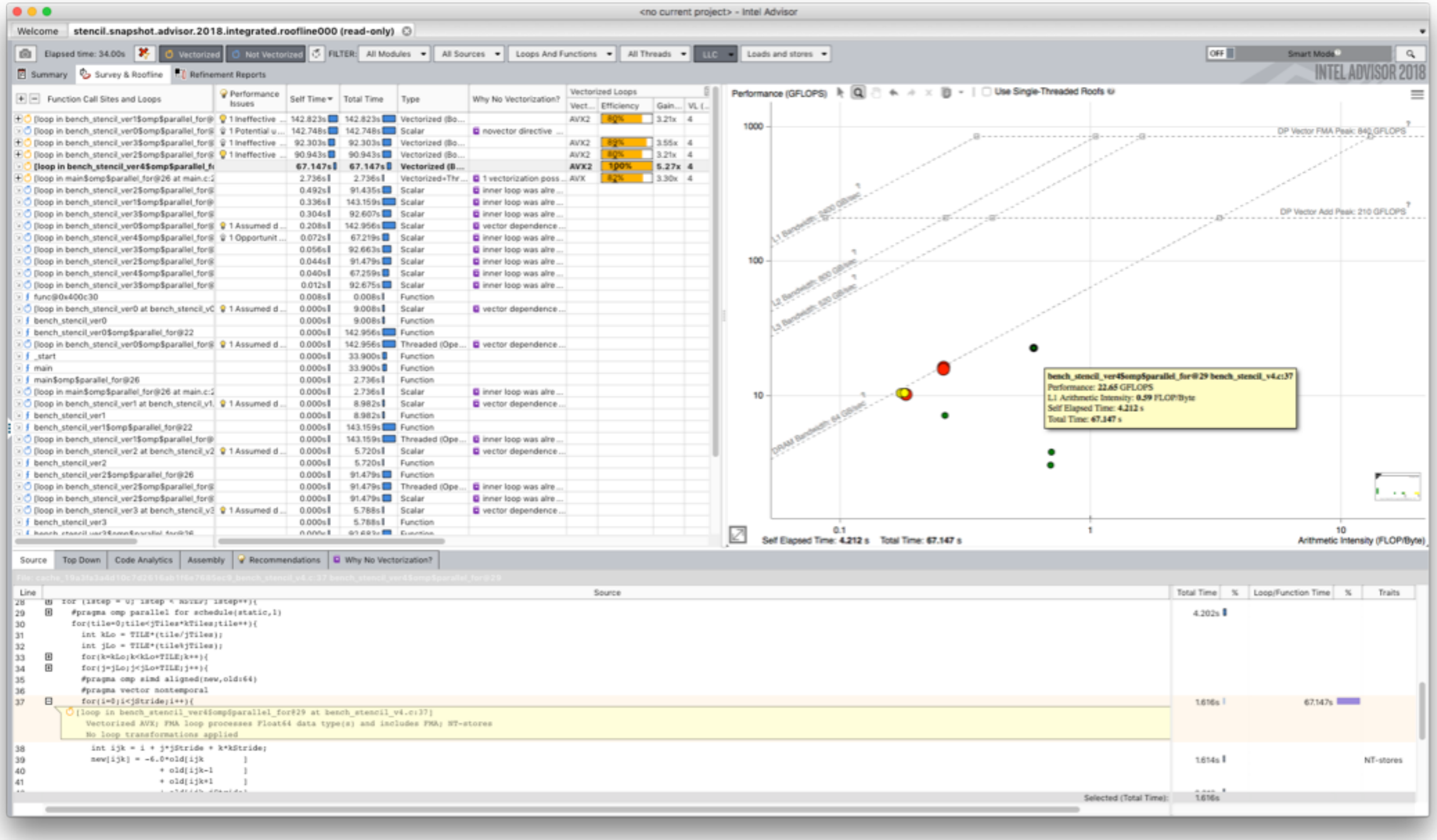
```
for (istep = 0; istep < NSTEP; istep++){
  #pragma omp parallel for schedule(static,1)
  for(tile=0;tile<jTiles*kTiles;tile++){
    int kLo = TILE*(tile/jTiles);
    int jLo = TILE*(tile%jTiles);
    for(k=kLo;k<kLo+TILE;k++){
    for(j=jLo;j<jLo+TILE;j++){
    #pragma omp simd aligned(new,old:64)
    #pragma vector nontemporal
    for(i=0;i<jStride;i++){
      [loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:37]
      Vectorized AVX; FMA loop processes Float64 data type(s) and includes FMA; NT-stores
      No loop transformations applied
      int ijk = i + j*jStride + k*kStride;
      new[ijk] = -6.0*old[ijk      ]
                     + old[ijk-1      ]
                     + old[ijk+1      ]
```

Total Time	%	Loop/Function Time	%	Traits
4.202s				
1.616s		67.147s		
1.614s				NT-stores

Selected (Total Time): 1.616s

LLC AI

- ver0 and ver1 have low LLC AI
- ver2 and ver3 (tiled) have substantially higher AI (working set fits in LLC)
- Ver4 has even higher LLC AI because NTS bypassed the LLC
- **Performance looks correlated with DRAM BW?**



Function Call Sites and Loops	Performance Issues	Self Time	Total Time	Type	Why No Vectorization?	Vectorized Loops			
						Vect...	Efficiency	Gain...	VL (...
[loop in bench_stencil_ver1$omp$parallel_for@	1 Ineffective ...	142.823s	142.823s	Vectorized (Bo...		AVX2	80%	3.21x	4
[loop in bench_stencil_ver0$omp$parallel_for@	1 Potential u...	142.748s	142.748s	Scalar	novector directive ...				
[loop in bench_stencil_ver3$omp$parallel_for@	1 Ineffective ...	92.303s	92.303s	Vectorized (Bo...		AVX2	89%	3.55x	4
[loop in bench_stencil_ver2$omp$parallel_for@	1 Ineffective ...	90.943s	90.943s	Vectorized (Bo...		AVX2	80%	3.21x	4
[loop in bench_stencil_ver4$omp$parallel_f		**67.147s**	**67.147s**	**Vectorized (B...**		**AVX2**	**100%**	**5.27x**	**4**
[loop in main$omp$parallel_for@26 at main.c:2		2.736s	2.736s	Vectorized+Thr...	1 vectorization poss...	AVX	82%	3.30x	4
[loop in bench_stencil_ver2$omp$parallel_for@		0.492s	91.435s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver1$omp$parallel_for@		0.336s	143.159s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3$omp$parallel_for@		0.304s	92.607s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver0$omp$parallel_for@	1 Assumed d ...	0.208s	142.956s	Scalar	vector dependence ...				
[loop in bench_stencil_ver4$omp$parallel_for@	1 Opportunit ...	0.072s	67.219s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3$omp$parallel_for@		0.056s	92.663s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver2$omp$parallel_for@		0.044s	91.479s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver4$omp$parallel_for@		0.040s	67.259s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3$omp$parallel_for@		0.012s	92.675s	Scalar	inner loop was alre...				
func@0x400c30		0.008s	0.008s	Function					
[loop in bench_stencil_ver0 at bench_stencil_v0	1 Assumed d ...	0.000s	9.008s	Scalar	vector dependence ...				
bench_stencil_ver0		0.000s	9.008s	Function					
bench_stencil_ver0$omp$parallel_for@22		0.000s	142.956s	Function					
[loop in bench_stencil_ver0$omp$parallel_for@	1 Assumed d ...	0.000s	142.956s	Threaded (Ope...	vector dependence ...				
_start		0.000s	33.900s	Function					
main		0.000s	33.900s	Function					
main$omp$parallel_for@26		0.000s	2.736s	Function					
[loop in main$omp$parallel_for@26 at main.c:2		0.000s	2.736s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver1 at bench_stencil_v1.	1 Assumed d ...	0.000s	8.982s	Scalar	vector dependence ...				
bench_stencil_ver1		0.000s	8.982s	Function					
bench_stencil_ver1$omp$parallel_for@22		0.000s	143.159s	Function					
[loop in bench_stencil_ver1$omp$parallel_for@		0.000s	143.159s	Threaded (Ope...	inner loop was alre...				
[loop in bench_stencil_ver2 at bench_stencil_v2	1 Assumed d ...	0.000s	5.720s	Scalar	vector dependence ...				
bench_stencil_ver2		0.000s	5.720s	Function					
bench_stencil_ver2$omp$parallel_for@26		0.000s	91.479s	Function					
[loop in bench_stencil_ver2$omp$parallel_for@		0.000s	91.479s	Threaded (Ope...	inner loop was alre...				
[loop in bench_stencil_ver2$omp$parallel_for@		0.000s	91.479s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3 at bench_stencil_v3	1 Assumed d ...	0.000s	5.788s	Scalar	vector dependence ...				
bench_stencil_ver3		0.000s	5.788s	Function					
bench_stencil_ver3$omp$parallel_for@26		0.000s	92.683s	Function					

Performance (GFLOPS)
Use Single-Threaded Roofs
1000
100
10
0.1
1
10
Arithmetic Intensity (FLOP/Byte)
DP Vector FMA Peak: 840 GFLOPS
DP Vector Add Peak: 210 GFLOPS
L1 Bandwidth: 3400 GB/sec
L2 Bandwidth: 800 GB/sec
L3 Bandwidth: 620 GB/sec
DRAM Bandwidth: 64 GB/sec

bench_stencil_ver4$omp$parallel_for@29 bench_stencil_v4.c:37
Performance: 22.65 GFLOPS
L1 Arithmetic Intensity: 0.59 FLOP/Byte
Self Elapsed Time: 4.212 s
Total Time: 67.147 s

Self Elapsed Time: 4.212 s Total Time: 67.147 s

```
for (istep = 0; istep < NSTEP; istep++){
  #pragma omp parallel for schedule(static,1)
  for(tile=0;tile<jTiles*kTiles;tile++){
    int kLo = TILE*(tile/jTiles);
    int jLo = TILE*(tile%jTiles);
    for(k=kLo;k<kLo+TILE;k++){
    for(j=jLo;j<jLo+TILE;j++){
    #pragma omp simd aligned(new,old:64)
    #pragma vector nontemporal
    for(i=0;i<jStride;i++){
     [loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:37]
      Vectorized AVX; FMA loop processes Float64 data type(s) and includes FMA; NT-stores
      No loop transformations applied
      int ijk = i + j*jStride + k*kStride;
      new[ijk] = -6.0*old[ijk       ]
                    + old[ijk-1     ]
                    + old[ijk+1     ]
```

4.202s

1.616s 67.147s

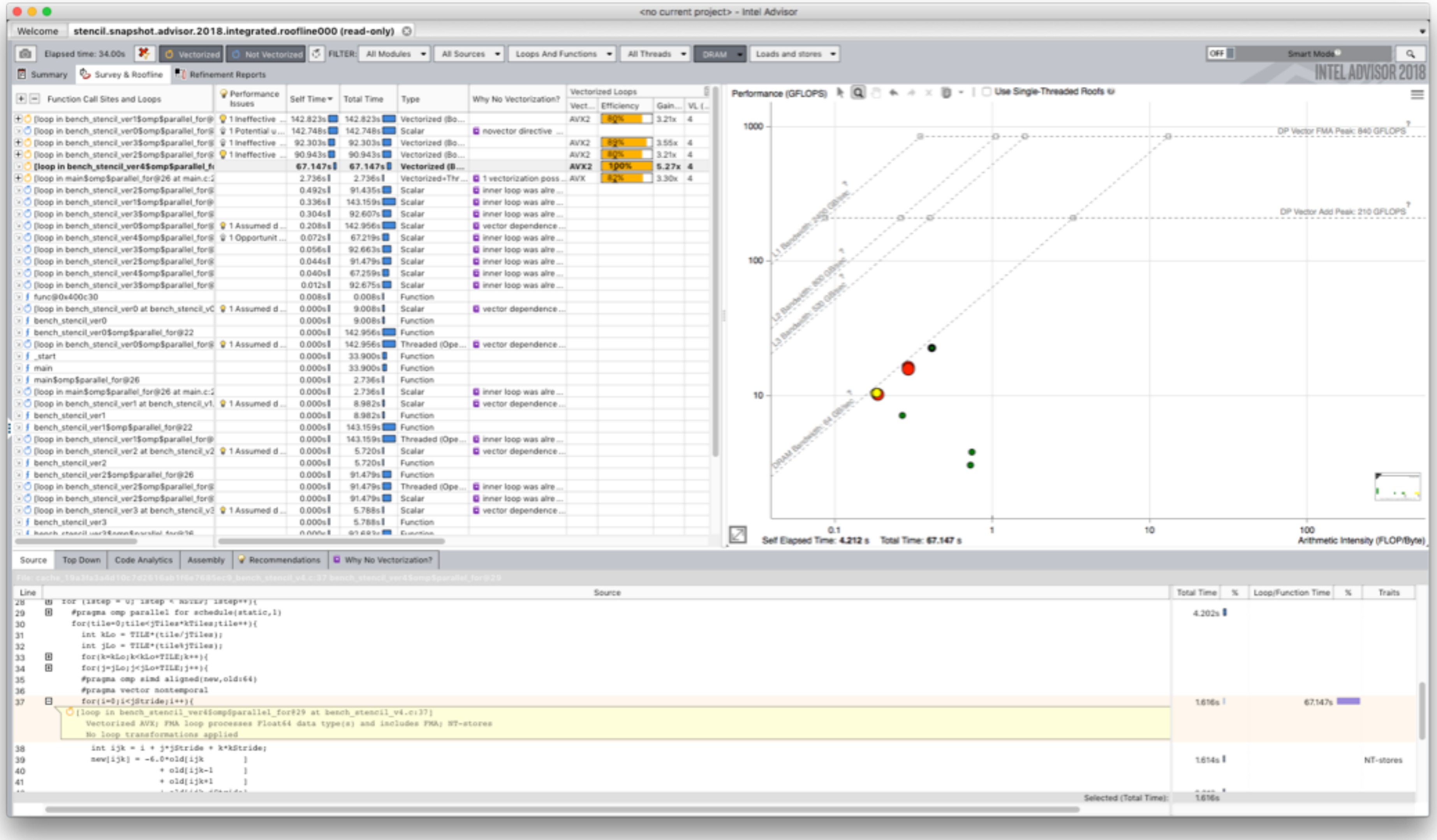
1.614s NT-stores

Selected (Total Time): 1.616s

DRAM AI

- Ver4 DRAM AI drops, but others remain the same
- Performance is now correlated with STREAM bandwidth

➢ **Ver4 DRAM AI ~ 0.41 (close to ideal 0.44)**



Function Call Sites and Loops	Performance Issues	Self Time	Total Time	Type	Why No Vectorization?	Vectorized Loops Vect...	Efficiency	Gain...	VL (...
[loop in bench_stencil_ver1$omp$parallel_for@	1 Ineffective ...	142.823s	142.823s	Vectorized (Bo...		AVX2	80%	3.21x	4
[loop in bench_stencil_ver0$omp$parallel_for@	1 Potential u...	142.748s	142.748s	Scalar	novector directive ...				
[loop in bench_stencil_ver3$omp$parallel_for@	1 Ineffective ...	92.303s	92.303s	Vectorized (Bo...		AVX2	89%	3.55x	4
[loop in bench_stencil_ver2$omp$parallel_for@	1 Ineffective ...	90.943s	90.943s	Vectorized (Bo...		AVX2	80%	3.21x	4
[loop in bench_stencil_ver4$omp$parallel_fo		**67.147s**	**67.147s**	**Vectorized (B...**		**AVX2**	**100%**	**5.27x**	**4**
[loop in main$omp$parallel_for@26 at main.c:2		2.736s	2.736s	Vectorized+Thr...	1 vectorization poss...	AVX	82%	3.30x	4
[loop in bench_stencil_ver2$omp$parallel_for@		0.492s	91.435s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver1$omp$parallel_for@		0.336s	143.159s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3$omp$parallel_for@		0.304s	92.607s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver0$omp$parallel_for@	1 Assumed d ...	0.208s	142.956s	Scalar	vector dependence ...				
[loop in bench_stencil_ver4$omp$parallel_for@	1 Opportunit ...	0.072s	67.219s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3$omp$parallel_for@		0.056s	92.663s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver2$omp$parallel_for@		0.044s	91.479s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver4$omp$parallel_for@		0.040s	67.259s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3$omp$parallel_for@		0.012s	92.675s	Scalar	inner loop was alre...				
func@0x400c30		0.008s	0.008s	Function					
[loop in bench_stencil_ver0 at bench_stencil_v0	1 Assumed d ...	0.000s	9.008s	Scalar	vector dependence ...				
bench_stencil_ver0		0.000s	9.008s	Function					
bench_stencil_ver0$omp$parallel_for@22		0.000s	142.956s	Function					
[loop in bench_stencil_ver0$omp$parallel_for@	1 Assumed d ...	0.000s	142.956s	Threaded (Ope...	vector dependence ...				
_start		0.000s	33.900s	Function					
main		0.000s	33.900s	Function					
main$omp$parallel_for@26		0.000s	2.736s	Function					
[loop in main$omp$parallel_for@26 at main.c:2		0.000s	2.736s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver1 at bench_stencil_v1.	1 Assumed d ...	0.000s	8.982s	Scalar	vector dependence ...				
bench_stencil_ver1		0.000s	8.982s	Function					
bench_stencil_ver1$omp$parallel_for@22		0.000s	143.159s	Function					
[loop in bench_stencil_ver1$omp$parallel_for@		0.000s	143.159s	Threaded (Ope...	inner loop was alre...				
[loop in bench_stencil_ver2 at bench_stencil_v2	1 Assumed d ...	0.000s	5.720s	Scalar	vector dependence ...				
bench_stencil_ver2		0.000s	5.720s	Function					
bench_stencil_ver2$omp$parallel_for@26		0.000s	91.479s	Function					
[loop in bench_stencil_ver2$omp$parallel_for@		0.000s	91.479s	Threaded (Ope...	inner loop was alre...				
[loop in bench_stencil_ver2$omp$parallel_for@		0.000s	91.479s	Scalar	inner loop was alre...				
[loop in bench_stencil_ver3 at bench_stencil_v3	1 Assumed d ...	0.000s	5.788s	Scalar	vector dependence ...				
bench_stencil_ver3		0.000s	5.788s	Function					
bench_stencil_ver3$omp$parallel_for@26		0.000s	92.683s	Function					

Self Elapsed Time: **4.212** s Total Time: **67.147** s

File: cache_19a3fa3a4d10c7d2616ab1f6e7685ec9_bench_stencil_v4.c:37 bench_stencil_ver4$omp$parallel_for@29

```
for (istep = 0; istep < NSTEP; istep++){
  #pragma omp parallel for schedule(static,1)
  for(tile=0;tile<jTiles*kTiles;tile++){
    int kLo = TILE*(tile/jTiles);
    int jLo = TILE*(tile%jTiles);
    for(k=kLo;k<kLo+TILE;k++){
    for(j=jLo;j<jLo+TILE;j++){
    #pragma omp simd aligned(new,old:64)
    #pragma vector nontemporal
    for(i=0;i<jStride;i++){
     [loop in bench_stencil_ver4$omp$parallel_for@29 at bench_stencil_v4.c:37]
      Vectorized AVX; FMA loop processes Float64 data type(s) and includes FMA; NT-stores
      No loop transformations applied
      int ijk = i + j*jStride + k*kStride;
      new[ijk] = -6.0*old[ijk        ]
                    + old[ijk-1      ]
                    + old[ijk+1      ]
```

Total Time	%	Loop/Function Time	%	Traits
4.202s				
1.616s		67.147s		
1.614s				NT-stores

Selected (Total Time): 1.616s

Questions

Open https://nxcloud01.nersc.gov in Browser

- Login using your temporary account.
- Create a shell on cori using your temporary account
- Load Advisor

```
module load advisor/2018.integrated_roofline
```

- Source the environment variables:

```
source $ADVISOR_XE_2018_DIR/advixe-vars.sh
```

- We've pre-collected data for you:

```
cp $ADVISOR_XE_2018_DIR/ECP-meeting-tutorial/* ~
```

- Launch advisor:

```
advixe-gui
```