REGIONALES HOCHSCHUL-
RECHENZENTRUM KAISERSLAUTERN

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

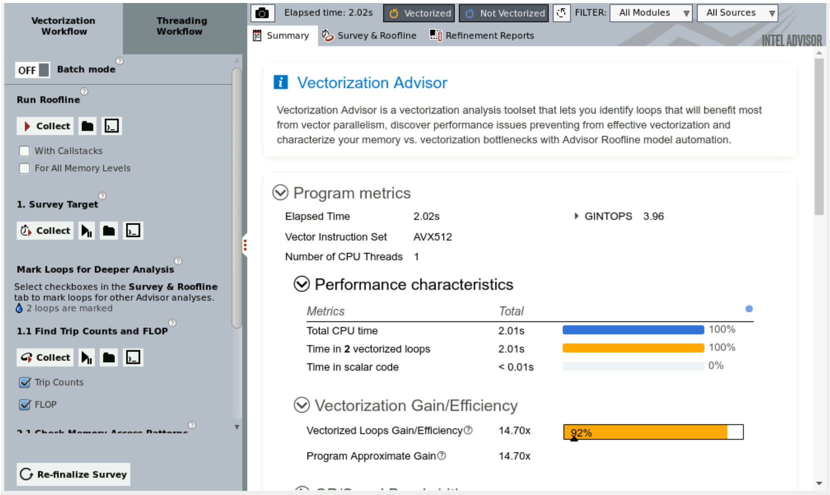
RHRK-Tutorial

Vectorization: The Intel Advisor Tool

Course instructor: Gabriele Monaco
In charge: Dr. Josef Schüle, RHRK

Intel Vectorization Advisor

- **Vectorization optimization and shared memory assistance tool**
- **Targets C, C++, C# and Fortran**
    - License available for TUK members
- **Supports codes with SSE, AVX, AVX2 and AVX-512 generated by both the Intel compiler and GCC**
    - Also explicit threading libraries like OpenMP
- **Run with**
    - module add intel/latest
    - advixe-gui



Select checkboxes in the **Survey & Roofline** tab to mark loops for other Advisor analyses.
2 loops are marked

Vectorization Advisor

Vectorization Advisor is a vectorization analysis toolset that lets you identify loops that will benefit most from vector parallelism, discover performance issues preventing from effective vectorization and characterize your memory vs. vectorization bottlenecks with Advisor Roofline model automation.

Program metrics

Elapsed Time	2.02s
Vector Instruction Set	AVX512
Number of CPU Threads	1
GINTOPS	3.96

Performance characteristics

Metrics	*Total*	
Total CPU time	2.01s	100%
Time in **2** vectorized loops	2.01s	100%
Time in scalar code	< 0.01s	0%

Vectorization Gain/Efficiency

Vectorized Loops Gain/Efficiency	14.70x	92%
Program Approximate Gain	14.70x	

Simple code example

- We can compile source codes with icc and give them to the tool
- For it to match the binary with the source code correctly we need debug symbols (-g option)
- The tool is going to suggest us which options to use if we want to exploit the current hardware
    - If we compile without avx512 suppor t, it can warn us

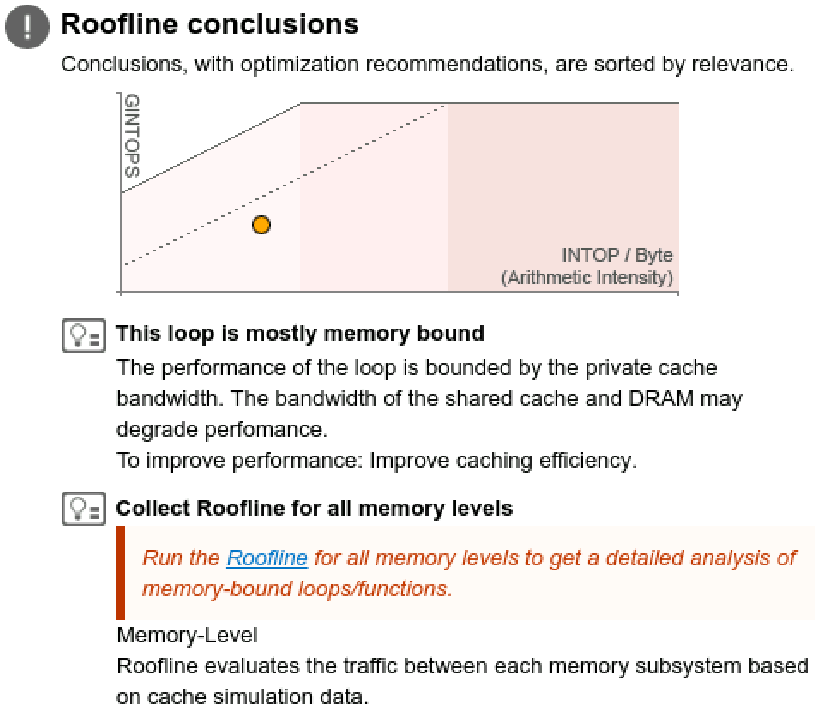
```
#include <stdio.h>
#define OUTER 1000000
#define MAX 6400

int main() {
    int i,io;
    int c[MAX],a[MAX],b[MAX];
    for(io=0;io<OUTER;io++) {
        for (i=0;i<MAX;i++)
            c[i]=a[i]=b[i]=24;
        for (i=0;i<MAX;i++)
            c[i]=a[i]+b[i];
    }
    printf("%d\n",c[3]);
}
```

```
icc -g -qopt-report=5 -O -xCORE-AVX512 -o example main.c
```

Optimizing with the tool

- **The tool can easily spot weaknesses in our code/binary**
  - Compiled with lower capabilities
  - Some loops aren' t vectorized
  - Time spent for memory operations
- **Straightforward to use but it's typically not enough for deep optimization!**
- **Remember to leave out the -g from compilation in production code!**



Roofline conclusions

Conclusions, with optimization recommendations, are sorted by relevance.

This loop is mostly memory bound

The performance of the loop is bounded by the private cache bandwidth. The bandwidth of the shared cache and DRAM may degrade perfomance.

To improve performance: Improve caching efficiency.

Collect Roofline for all memory levels

Run the Roofline for all memory levels to get a detailed analysis of memory-bound loops/functions.

Memory-Level

Roofline evaluates the traffic between each memory subsystem based on cache simulation data.

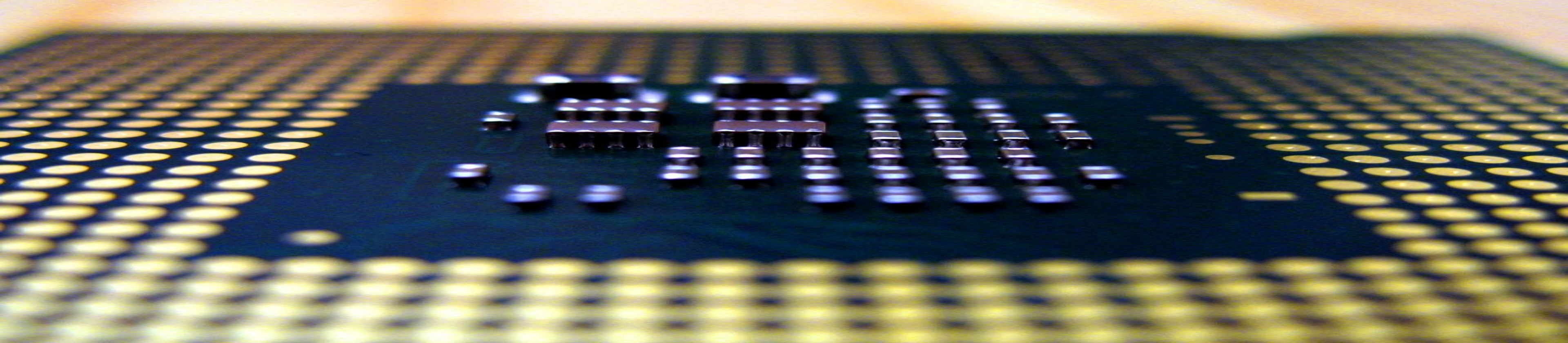
GCC compatibility

- **The tool works perfectly with ICC but the GNU compiler is supported as well**
  - **Still we need to add the -g option for debugging symbols**
  - **Be careful while adding AVX512 as all wanted extensions shall be included**
    - -mavx512f and -mavx512cd may be a good option
    - -march=skylake-avx512 also targets directly that processor family
- **The results may look a little different compared with icc (as we have seen the compilation is sometimes different)**
  - **Not quite noticeable on this simple example**

```
gcc -g -fopt-info-vec -ftree-vectorize -march=skylake-avx512
-o example main.c
```

Farewell

- **The Advisor tool is usually not saving the day, but if you came so far, you know how to solve those problems on your own!**
  - **We know what to avoid and how to structure the code**
  - **We know how to set the compiler from source and command line**
  - **We know how to efficiently access memory**
  - **We know how to get away with complicated cases with extensions**
    - The rest will come with practice!



- **High Performance Computing on Elwetritsch**

Vielen Dank
Thank You