



Islas  
Canarias



# Vectorization studies: Haswell and Cilk+

Juan José Fumero Alfonso

[juan.jose.fumero.alfonso@cern.ch](mailto:juan.jose.fumero.alfonso@cern.ch)

CERN openlab summer student 2013



# Outline

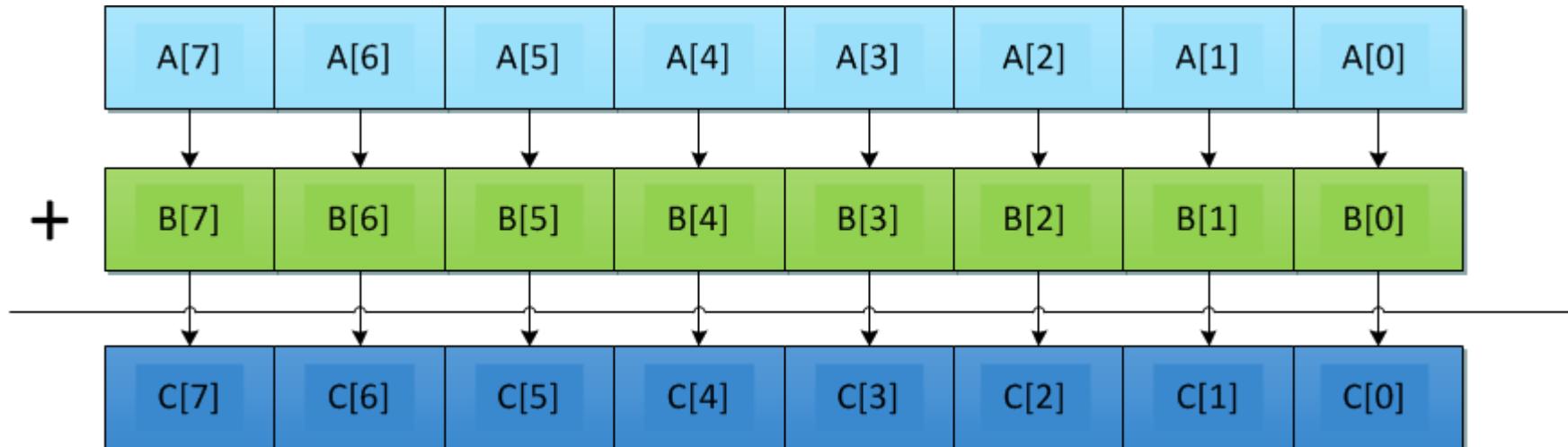
1. Motivation
2. MLFit
3. Geant5 and CilkPlus Compiler
4. CilkPlus implementations of Geant5
5. Conclusions



# Exploiting parallelism inside the core

```
for (int i = 0; i < n; i++) {  
    c[i] = a[i] + b[i];  
}
```

```
vmovups .L8(%rip), %ymm0  
vaddps .L.9(%rip), %ymm0,  
%ymm1  
vmovups %ymm1, 32(%rsp)  
vmovups %ymm1, 64(%rsp)
```



# MLfit

- It is a simplified version of the RooFit package
- Component of ROOT framework used in likelihood based data analysis.
- Benchmark used at CERN openlab as a representative of data analysis
- Written in OpenMP and MPI, Cilk+ and TBB
- It is prepared for XEON PHI



# MLfit (II)

- Sample of N events (energies, masses, ...)
- Maximum Likelihood fit technique is used to estimate the values of free parameters, minimizing the Negative Log-Likelihood (NLL) function.

$$NLL = \sum_{j=1}^s n_j - \sum_{i=1}^N \left( \ln \sum_{j=1}^s n_j \mathcal{P}_j(x_i; \theta_j) \right)$$

$j$  species (signals, backgrounds)

$n_j$  number of events

$\mathcal{P}_j$  probability density function (PDF)

$\theta_j$  Free parameters in the PDFs



# Environment

Microarchitecture	Processor	Frq (MHz)	#Cores/Threads	L1	L2	L3
Sandy Bridge	Intel Xeon CPU E5-2690 (x2)	2900	16/32	32 K	256K	20M
Ivy Bridge	Intel Xeon CPU E3-1265L v2	2500	4/8	32 K	256K	8M
Haswell	Intel Xeon CPU E3-1285L V3	3100	4/8	32 K	256K	8M



# Compilation options (Intel)

Option	Description
<b>-xsse</b>	The compiler enables SSE3, SSE2 and SSE1 vector code
<b>-xsse4.2</b>	Intel Compiler may generate instructions from SSE to SSE4.1 and SSE4.2
<b>-xavx</b>	Intel Compiler generates instructions for AVX (256 bits) if the processor allows it.
<b>-xcore_avx2</b>	Intel Compiler generates AVX2 vector code. It is only enabled on the Haswell microarchitecture.
<b>-no-fma</b>	Intel Compiler enables FMA by default when AVX2 is used. This option is needed to disable FMA and compare AVX2 vector code with AVX

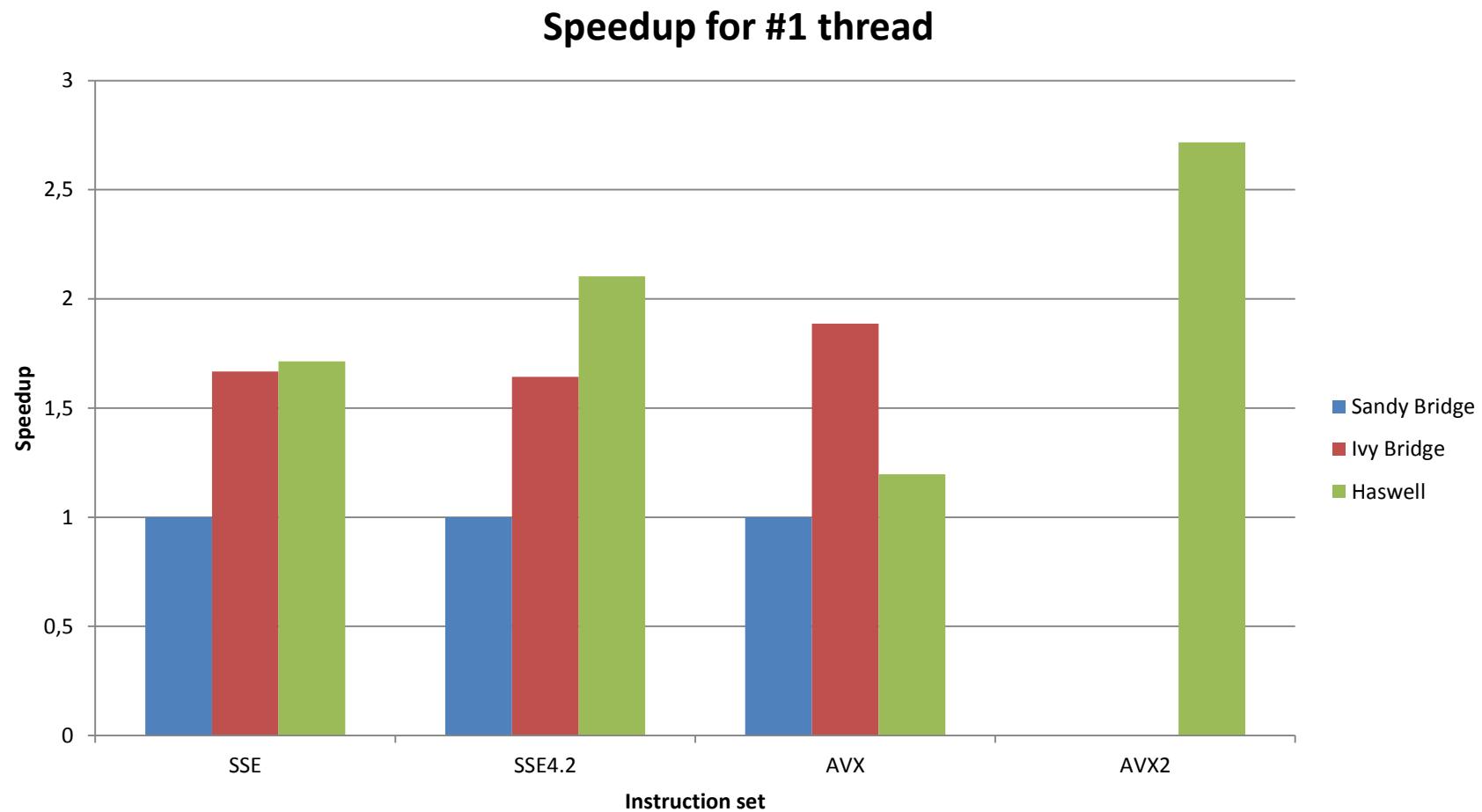


# Experiments

- It has been run three times per instance
- We have got the the best time for analysis
- We have run the program with two different sizes in each microarchitecture for each Vector Instruction Set
  - 500,000 events
  - 1,000,000 events

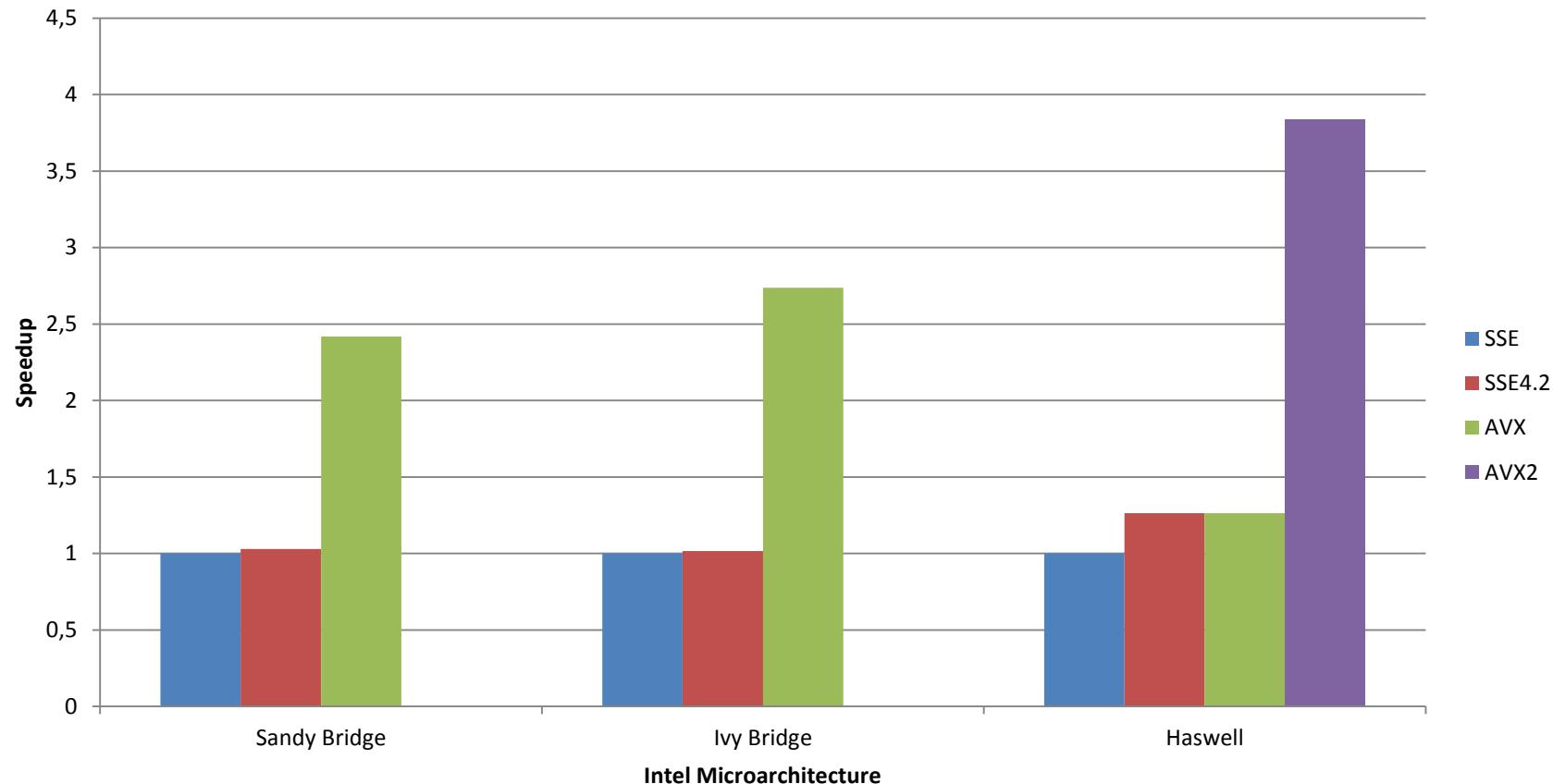


# Relative speedup

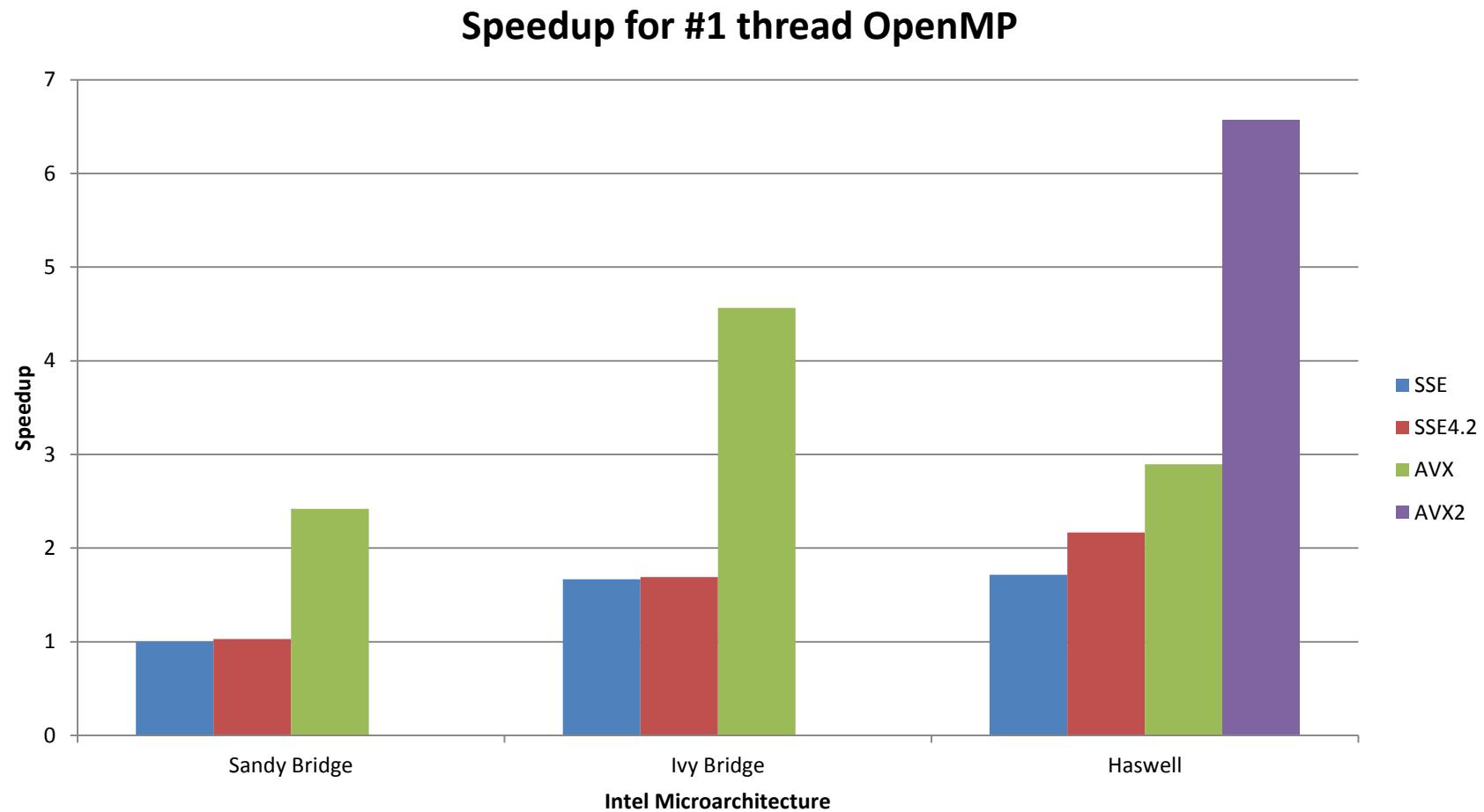


# Relative speedup

Speedup comparison #1 thread OpenMP



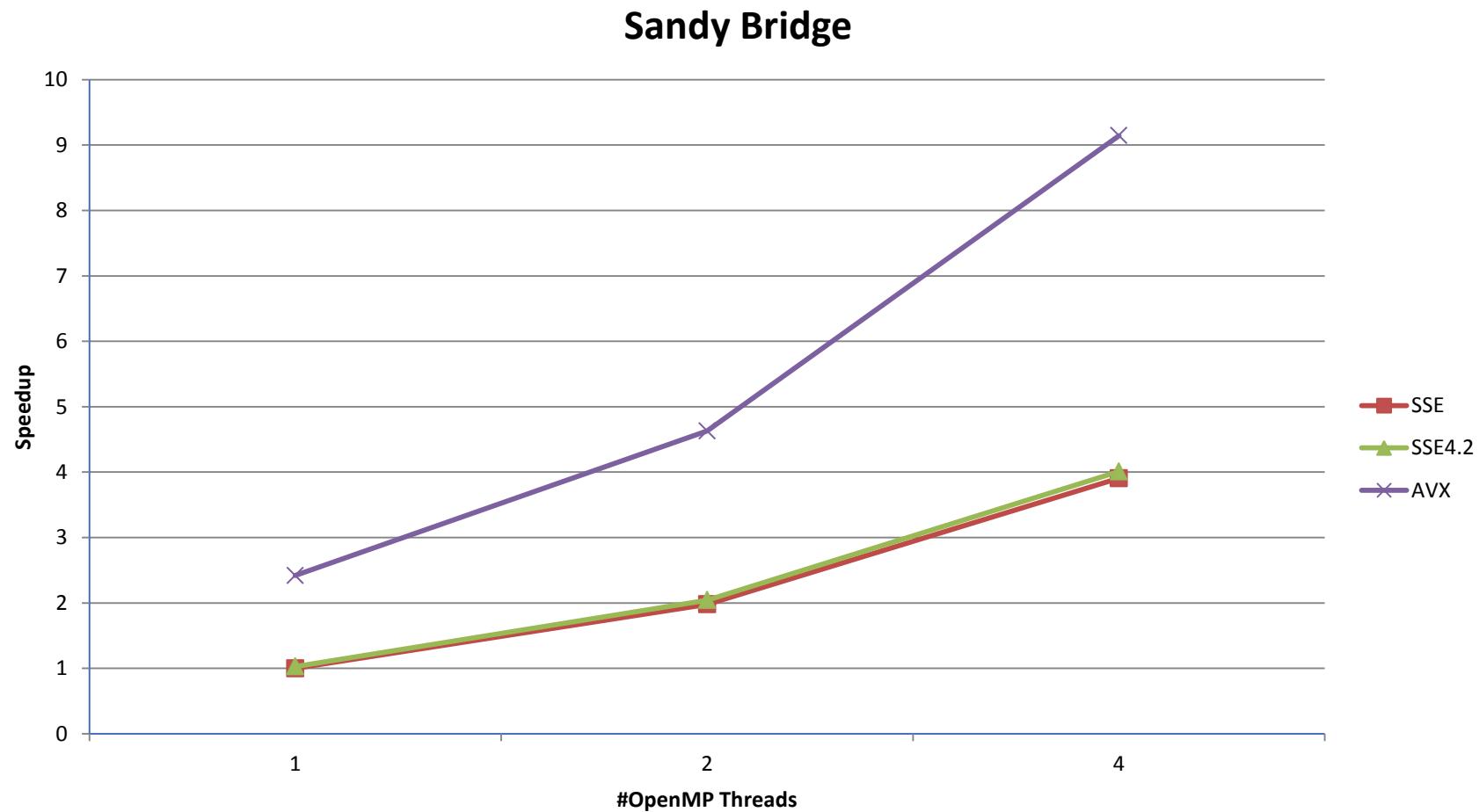
# Comparison – Microarchitectures



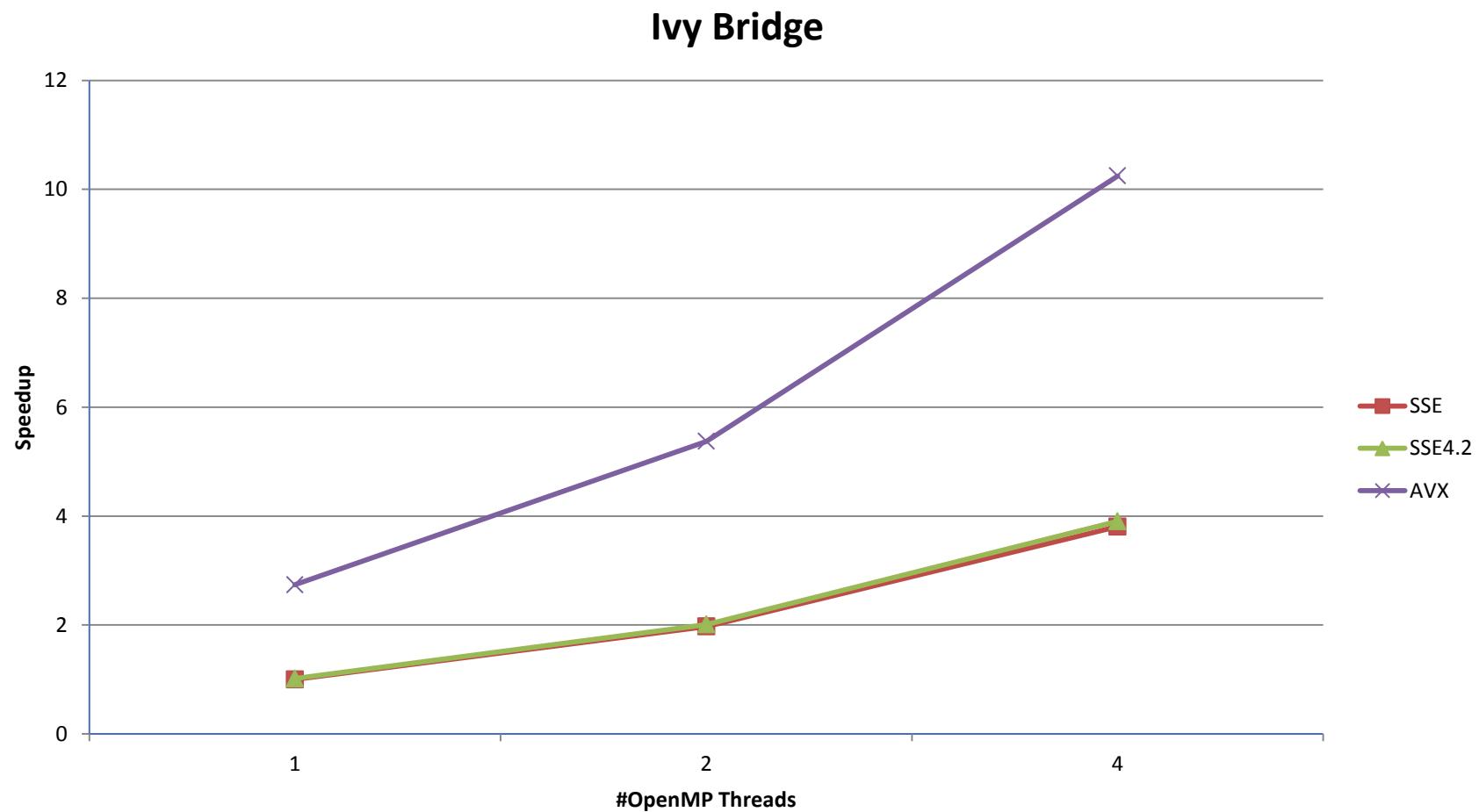


Islas  
Canarias

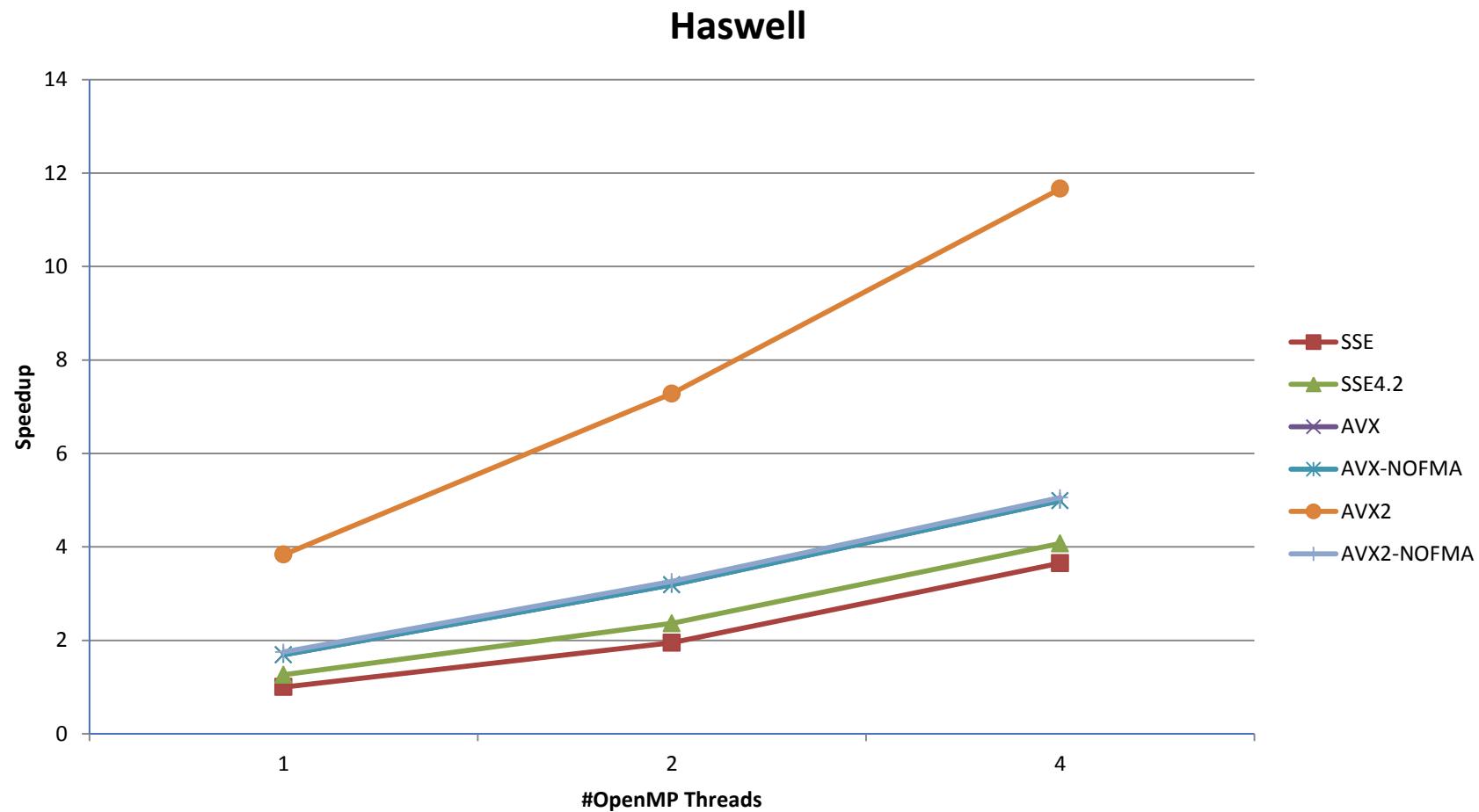
# 1M of events (4 threads, Sandy)



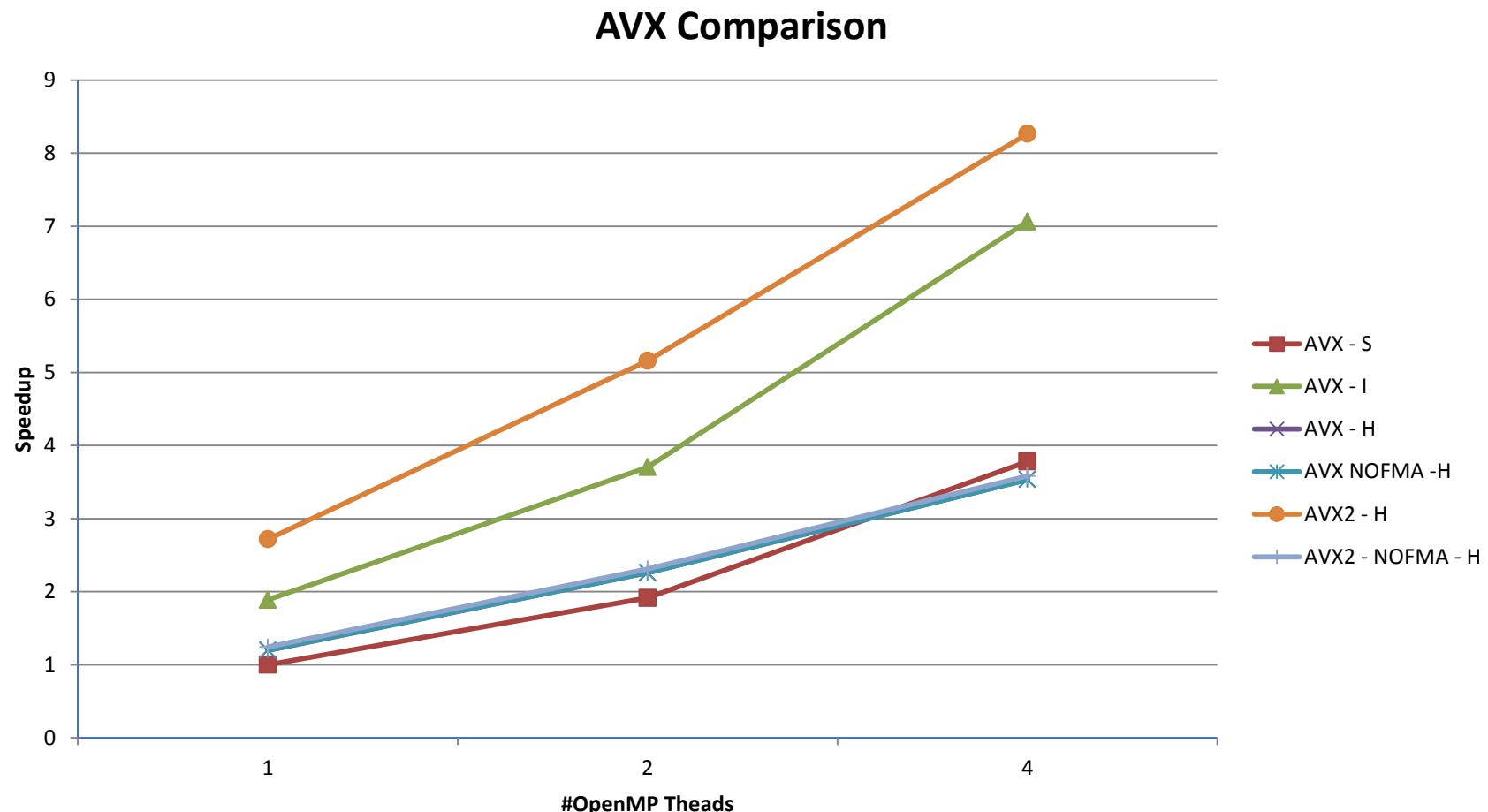
# 1M of events (II), 4threads - Ivy



# 1M of events(III), 4 threads-Haswell

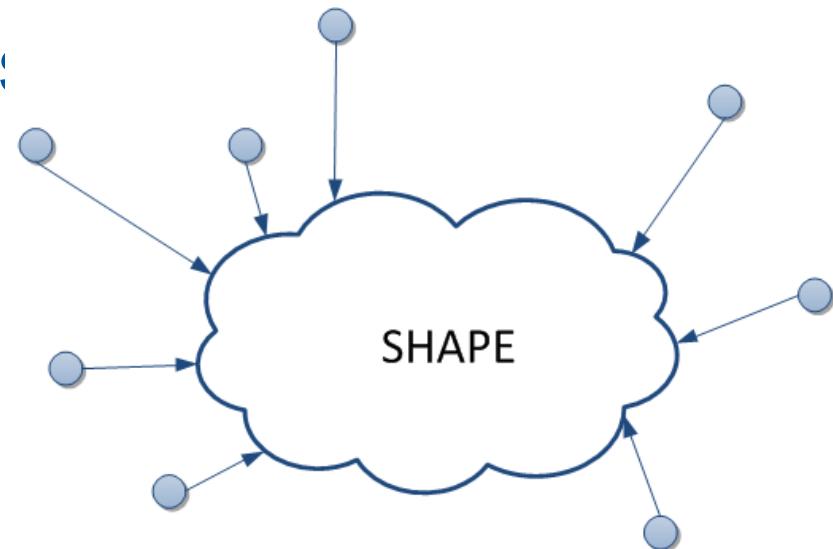


# 1M events – AVX compararison



# Geant5 prototype

- Detector simulation is one of the most CPU intensive tasks in modern High Energy Physics.
- Geant5 represents a benchmark for these kinds of simulations:



# CilkPlus Compiler



- Cilk+ is an extension to the C/C++ languages to support data and tasks parallelism
- New tokens to express the tasks parallelism
- New syntax to express the data parallelism (CEAN)
- Quickest way to harness the power of both multicore and vector processing
- Intel implementation and GCC-4.8.1 CilkPlus branch (newest)

# TGeoBBox\_v

- Five different implementations
  1. CilkPlus small vectors
  2. CilkPlus array notation processing all particles
  3. CilkPlus with tiling (extra space)
  4. CilkPlus less memory
  5. CilkPlus processing small tiles

Original from: S. Swenzel



# Original code

```
newpt[0] = p[0][k] - origin[0];
saf[0] = TMath::Abs(newpt[0])-par[0];
factor = (saf[0]>=stepmax[k]) ? TGeoShape::Big() : 1
in = (saf[0]<0);
```

```
newpt[1] = p[1][k] - origin[1];
saf[1] = TMath::Abs(newpt[1])-par[1];
factor *= (saf[1]>=stepmax[k]) ? TGeoShape::Big() : 1;
in = in & (saf[1]<0);
```

```
newpt[2] = p[2][k] - origin[2];
saf[2] = TMath::Abs(newpt[2])-par[2];
factor *= (saf[2]>=stepmax[k]) ? TGeoShape::Big() : 1.;
in = in & (saf[2]<0);
```



# DistanceFromOutside (Cilk1)

```
newpt[0] = p[0][k] - origin[0];
saf[0] = TMath::Abs(newpt[0])-par[0];
factor = (saf[0]>=stepmax[k]) ? TGeoShape::Big() :
1
in = (saf[0]<0);

newpt[1] = p[1][k] - origin[1];
saf[1] = TMath::Abs(newpt[1])-par[1];
factor *= (saf[1]>=stepmax[k]) ? TGeoShape::Big() :
1;
in = in & (saf[1]<0);

newpt[2] = p[2][k] - origin[2];
saf[2] = TMath::Abs(newpt[2])-par[2];
factor *= (saf[2]>=stepmax[k]) ? TGeoShape::Big() :
1.;
in = in & (saf[2]<0);
```

## Original

```
factor[:] = 1;
double point_aux[3]    = { p[0][k], p[1][k],
                           p[2][k] };

newpt[0:3] = point_aux[0:3] - origin[0:3];
saf[:] = abs(newpt[:])-par[:];
if ( saf[:] >= stepmax[k] )
    factor[:] = geobig;
else
    factor[:] = 1;
factor_scalar =
__sec_reduce_mul(factor[:]);
```

## Cilk1



# Cilk+ & Geant5

- We have implemented several approaches
- We have studied the ASM code
- We have studied perf output
- Waiting for feedback of Intel
- Analysis with Intel
- Maybe new approach no the solution



# Conclusions and future

- The vector instruction set AVX and AVX2 show very good performance in combination with FMA.
- Array notation with CilkPlus is very easy to use easy and to learn. Waiting for feedback
- Running MLFit with GCC
- Exploring Intel Xeon PHI?
- GPUs?



# Thank you

Juan José Fumero

[juan.jose.fumero.alfonso@cern.ch](mailto:juan.jose.fumero.alfonso@cern.ch)



# Comments, suggestions, ...



# Bibliography

1. A Guide to Auto-vectorization with Intel® C++ Compilers. Intel Documentation.
2. Sverre Jarp, Afio Lazzaro, Andrzej Nowak, Liviu Valsan. *Comparison os Software Technologies for Vectorization and Parallelization*. CERN OpenLAB 2012 – Whitepaper.
3. John L. Hennessy, David A. Patterson. *Computer Architecture. Quantitative Approach*. Fifth Edition (The Morgan Kaufmann Series in Computer Architecture Design).
4. Intel CilkPlus Documentation
5. Michael McCool, James Reinders, Arch Robinson. Structured Parallel Programming: Patterns for Efficient Computation. MK 2013.

