Performance study of detector alignment software

› **26/08/2015**

Estefania Serrano

Supervisor:

Pawel Szostek

CERN openlab

Millepede II

- **Least square fit application.**
- **Written in Fortran.**
- **A lot of methods for solving equation systems.**
  - Most of them memory optimized . -> Medium problem 25 GB on memory
  - Computationally expensive. -> Whole week -end of execution
- **Already using parallelization.**
  - OpenMP in some sections

http://www.desy.de/~kleinwrt/MP2/doc/html/index.html

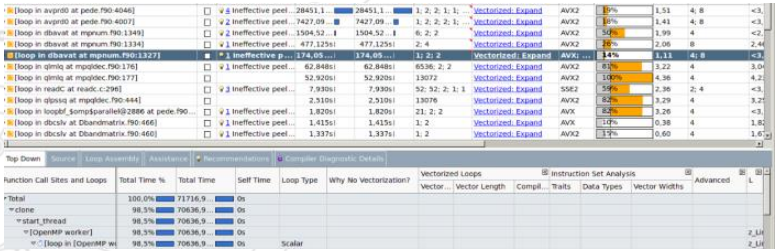
http://www.desy.de/

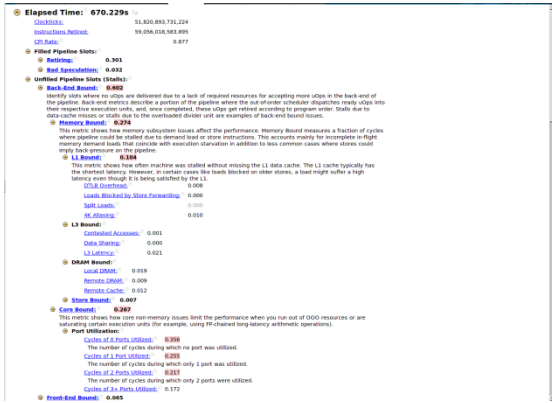
Is it possible to optimize it?

First steps: Characterization

- Intel Compiler adaptation.
- Study of the hotspots:

Intel VTune





Intel Advisor

Vectorization

- **Compiler pragmas**

!DIR$ ASSUME_ALIGNED A: 64, V: 64

- **Instruction set changes**

-axAVX2

- **Code modifications**

b(j)=b(j)+globalMatD(offset+j)*x(j)

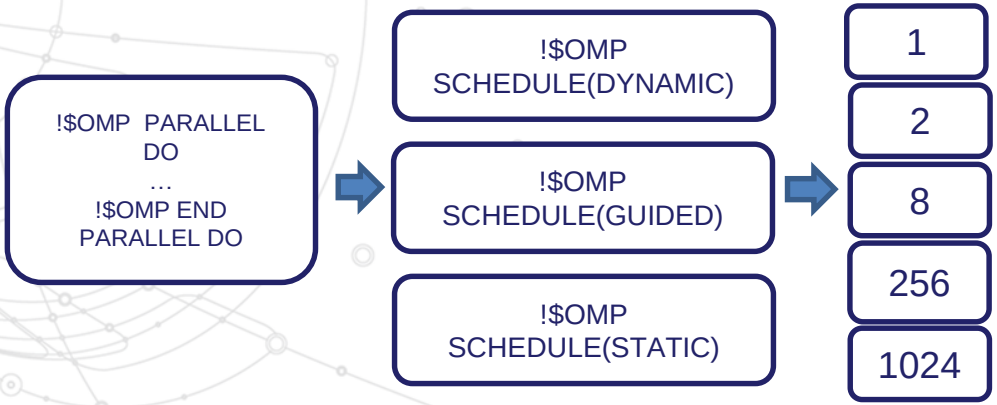
b(j)=b(j)+globalMatD(newj)*x(j)

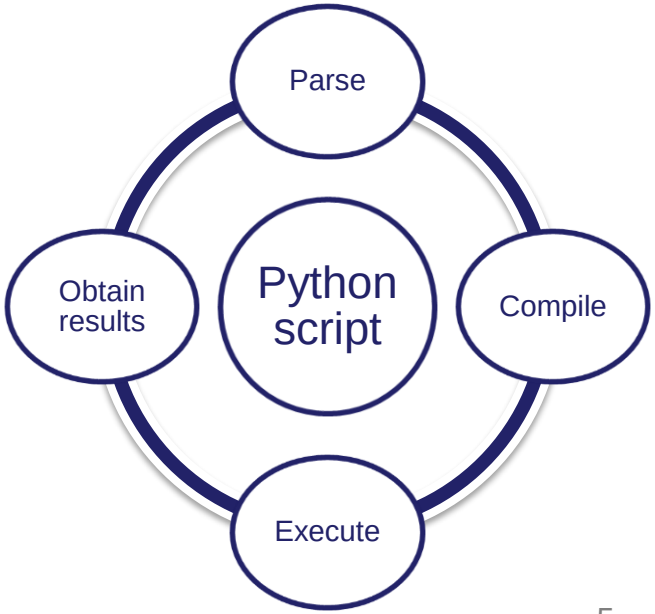
Obvious changes for the programmer…

Not so obvious for the compiler!!!

Parallelization

- **Parallelized sections with high imbalance**
  - Some threads get more work than others
- **Optimization space: 170 executions**





Xeon Phi

Coprocessor = big number of cores

Not a guarantee of better performance!!

Requirements:

- Good vectorization and alignment ✓
- Small sequential sections
- Small memory usage
- Not much use of I/O

Initial experiments:

Xeon processor:	200 s
Xeon Phi:	2000 s

10x slower!
Still work to do

www.clipartbest.com

http://nexnet.files.wordpress.com/2013/02/kliponious-black-tick.png

Conclusion

- **Some room for optimization**
  - Medium problem: 1900 s to 1700 s
- **Advises for the future:**
  - Modern programming language
  - Optimized mathematical libraries

Even fixed some bugs!!

“It is much easier to create an optimized application than optimizing an already created one.”

http://pngimg.com/upload/bug_PNG3980.png

Performance study of detector alignment software

› **26/08/2015**

Estefania Serrano

Supervisor:

Pawel Szostek

CERN openlab