Tuning ECMWF's RAPS9 benchmark for Intel® platforms

N. Panov

Intel Corp.,
pr. ak. Lavrentieva 6/1, Novosibirsk, Russia, 630090
nikita.v.panov@intel.com

Plan

- Porting steps
- Tuning efforts
- Scalability
- Future works

Notations and abbreviations

- Woodcrest processor – Dual Core Intel® Xeon® Processor model 5160
- ppn – processes per node (the number of cores used on a node)
- MVAPICH – an MPI implementation from the Ohio State University

Hardware and software

256 nodes **Woodcrest**, 2 sockets / 2 cores each, 3.0 GHz; 4 MB L2; 8GB RAM;
Infiniband Interconnection

Red Hat Enterprise Linux 3.0.
Intel® Fortran and C compilers v9.1 for Linux.
Intel® MPI 2.0, MVAPICH-0.9.7

Porting efforts

- We ported ECMWF RAPS9 both on Itanium2® and Xeon® processors
- We made 37 changes in makefiles and scripts
- There are eight modifications in the source codes
    - *$RAPS9/src/ec/source/signal_trap.c*
    - *$RAPS9/src/dummy/src/errtra.F*
    - *$RAPS9/src/ifsaux/support/envsub.F90*
    - *$RAPS9/src/ifs/parallel/slcomm2a.F90*
- Adapted the sources to use Intel® MKL (Math Kernel Library), continue more integration
- We were able to run forecast models T21A, T21R, T399, T511, T799 and 4D-VAR T159 case on both platforms

Intel® Performance Optimization Tools

- **Intel® Compilers –** specific optimizations for Intel processors.
- **VTune™** – Intel® Performance Analyzer. It makes application performance tuning easier with a graphical user interface.
- **Intel® Trace Collector & Analyzer.** The tool allows analyzing and optimizing high-performance applications on clusters of Intel® processor-based systems.
- **Intel® Math Kernel Library** - highly optimized, thread-safe math routines for maximum performance.

Intel® Compilers Features

Efficiency:

Inherent ability to highly optimize codes for all Intel processors

Ease of Use:

Automatic optimization features make it easier to obtain highly optimized target code

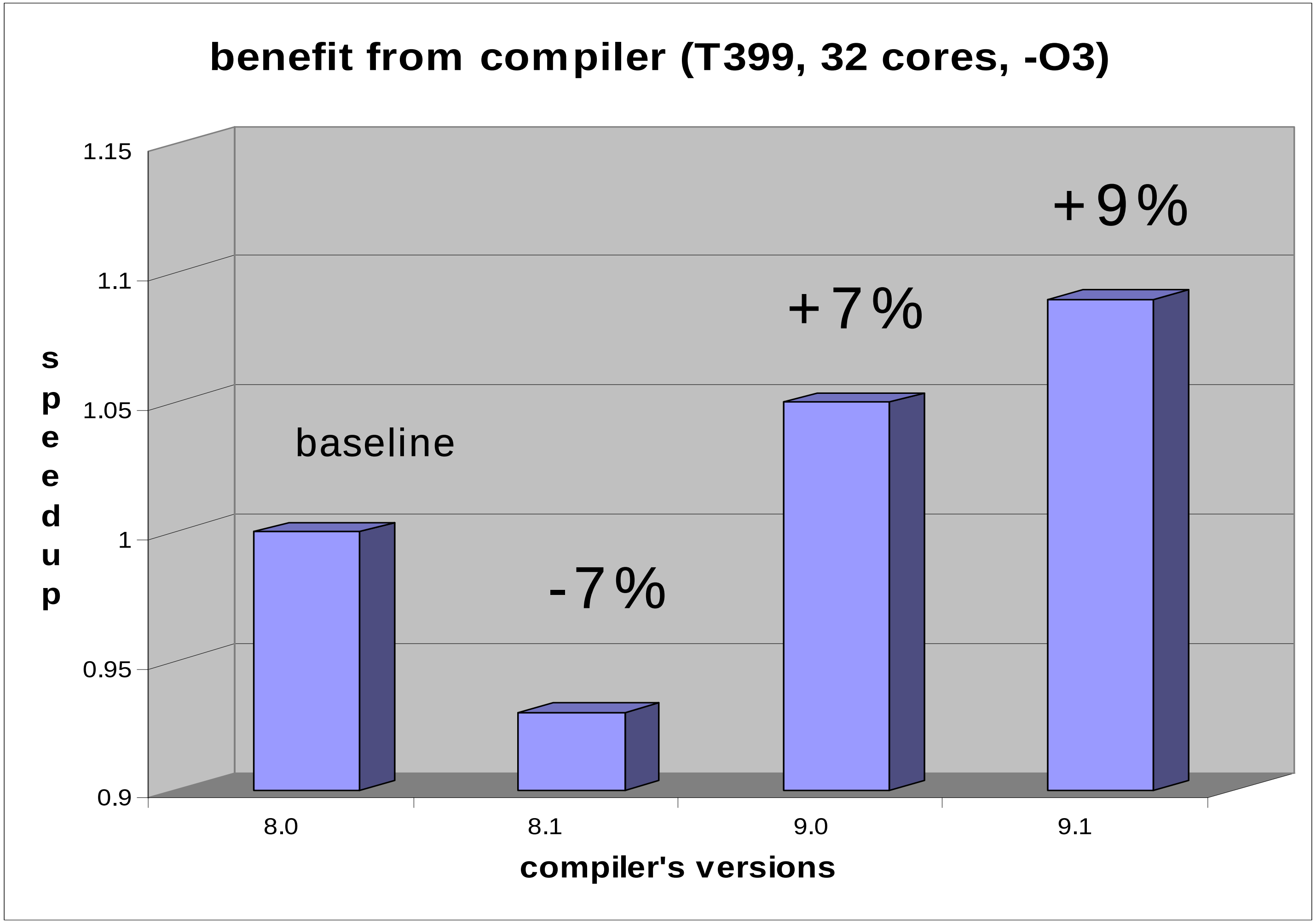
Intel® Premier Support:

- training
- best known methods
- problem fixes & workarounds

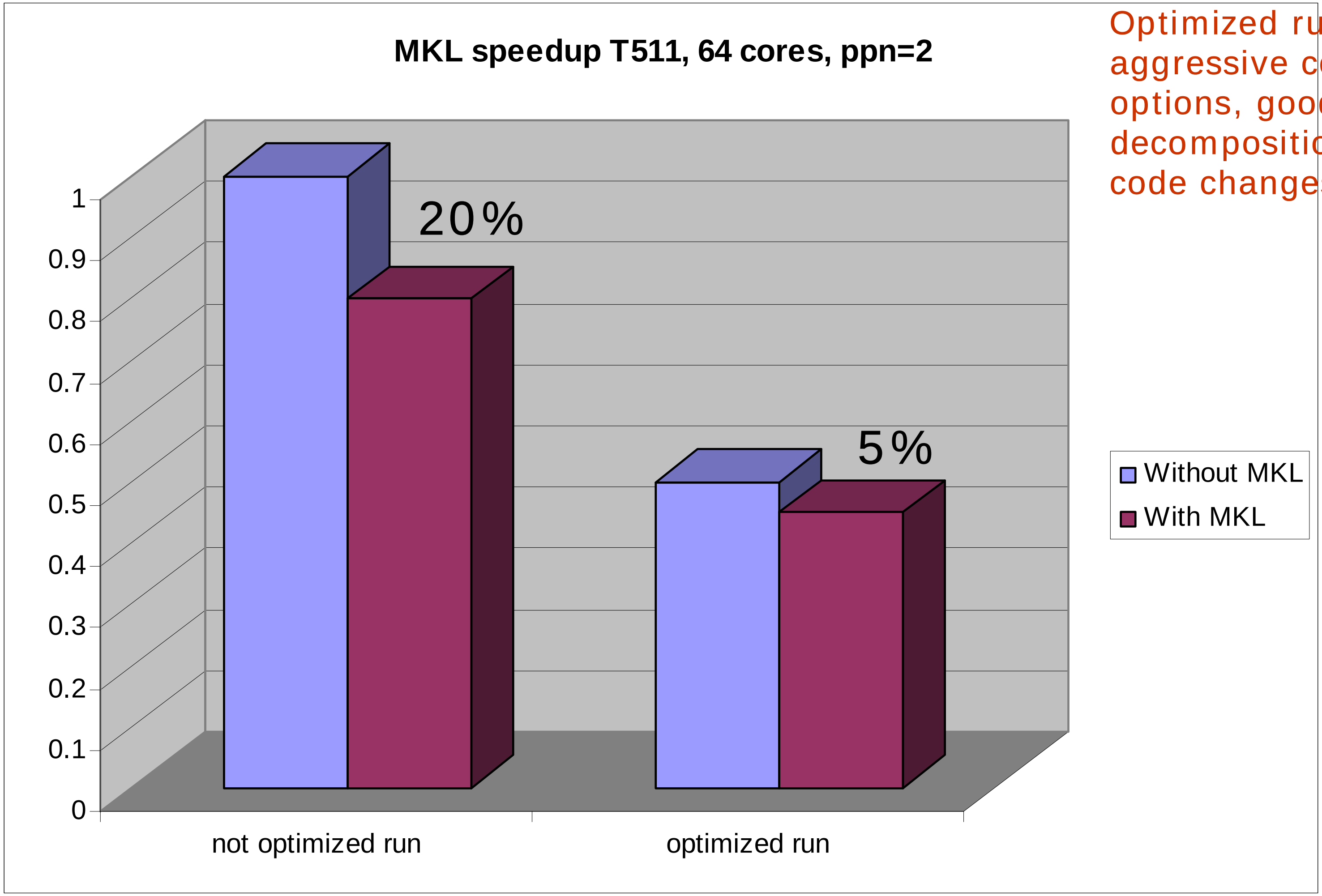
Intel® compilers use:

- Speculative memory accesses
- Advanced branch prediction
- Software pipelining for Intel® Itanium®
- Specific optimization for Woodcrest

Tuning efforts

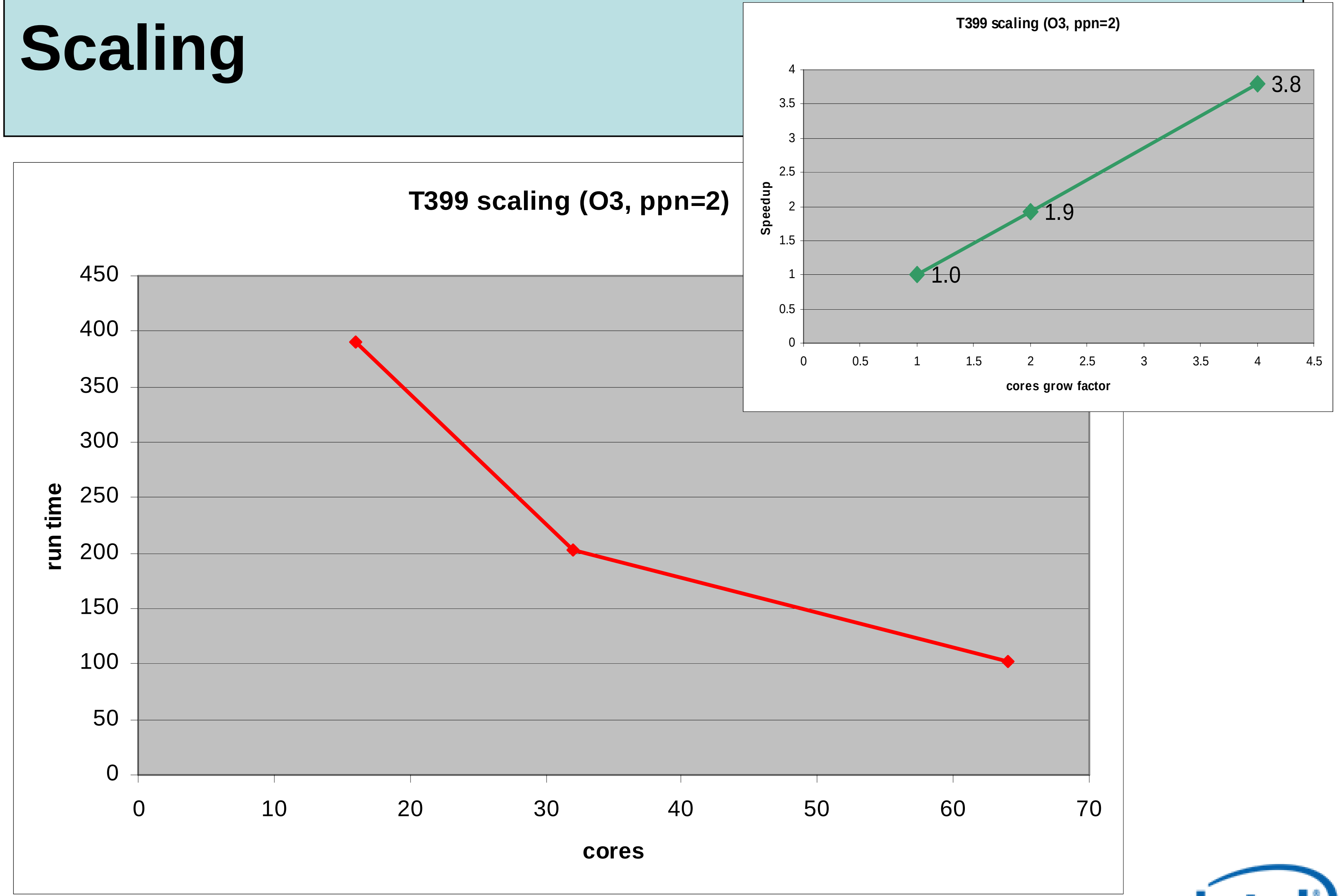


Tuning efforts



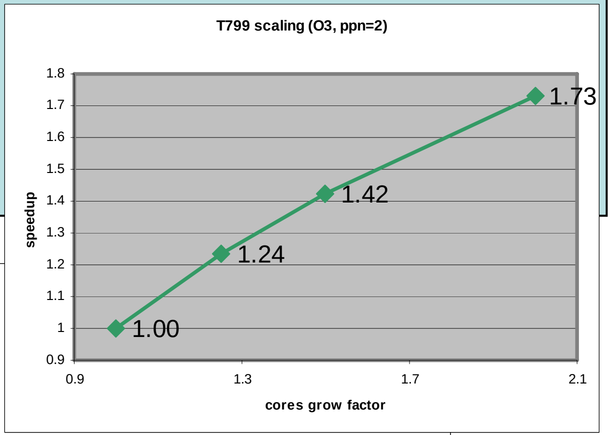
Optimized run means aggressive compiler’s options, good decomposition and minor code changes

Scaling

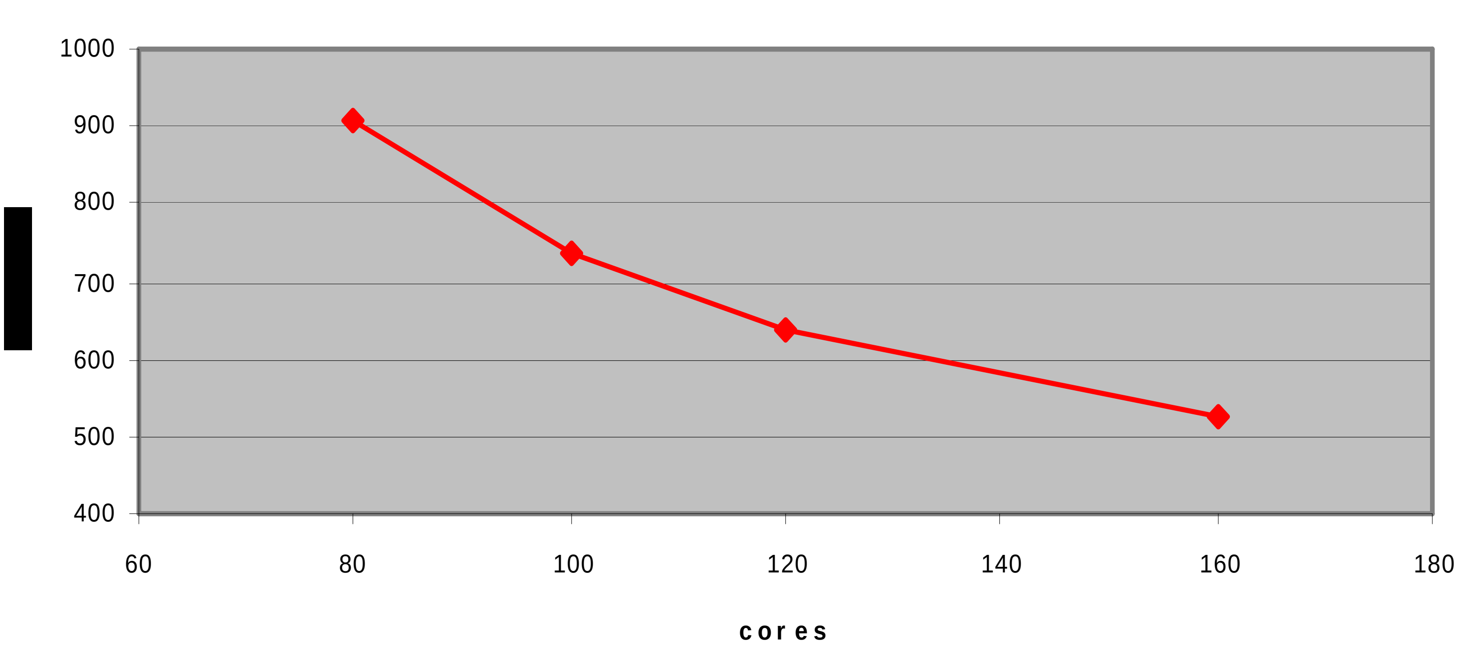


Dual-Core Intel® Xeon® 5160 processor, 2-socket sys., 3.0GHz, 4MB L2 cache, 8GB Memory

Scaling

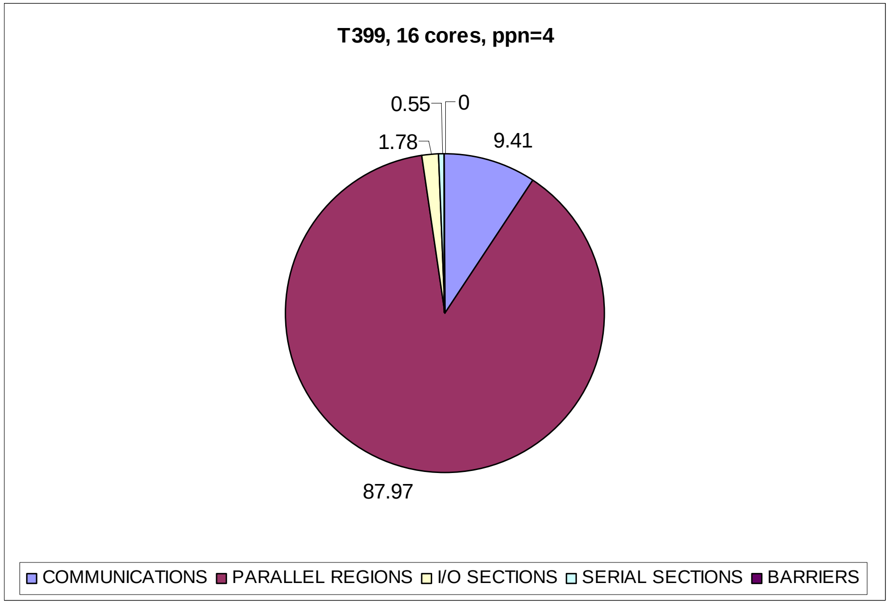


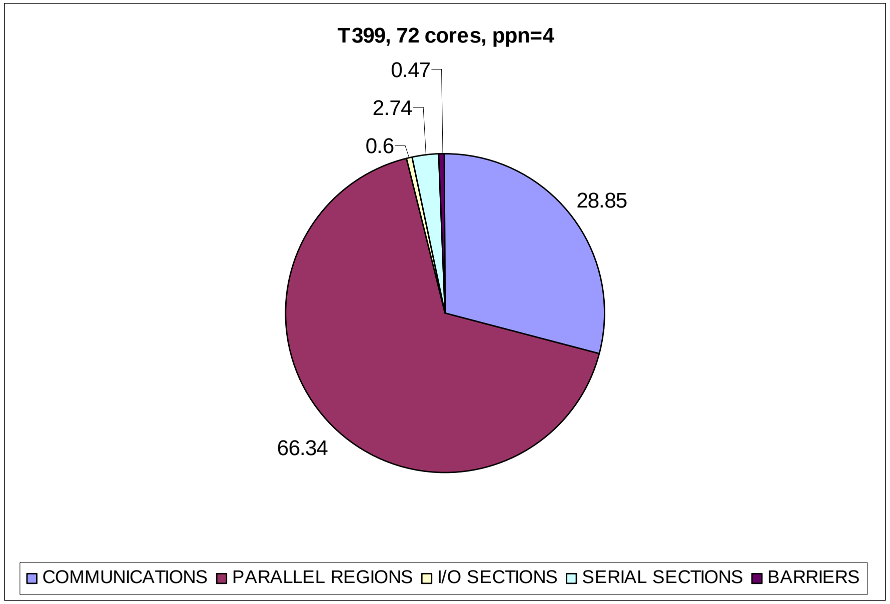
T799 scaling (O3, ppn=2)



Dual-Core Intel® Xeon® 5160 processor, 2-socket sys., 3.0GHz, 4MB L2 cache, 8GB Memory

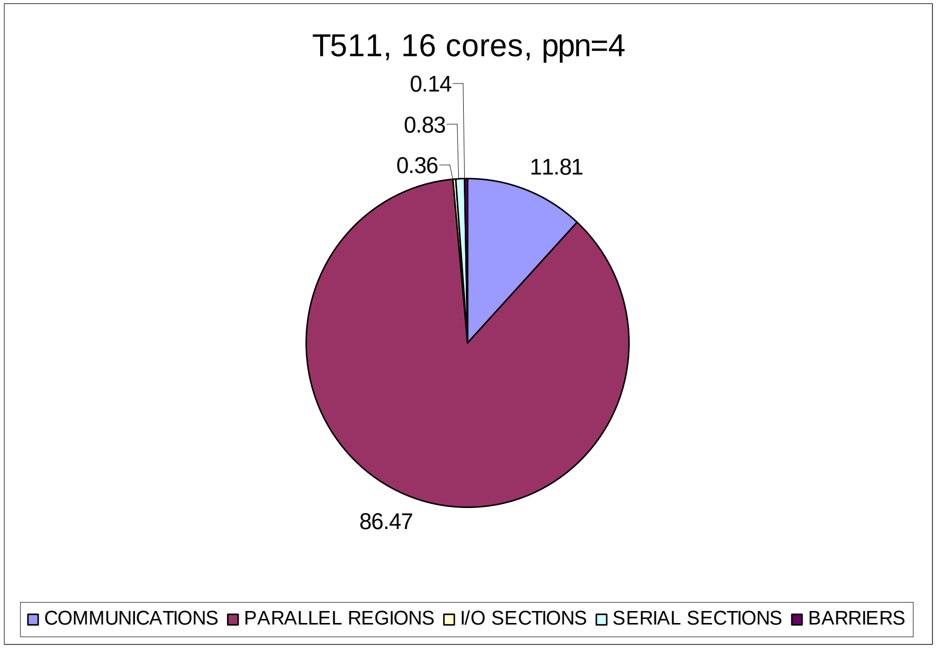
T399 CPU-time as percent of total

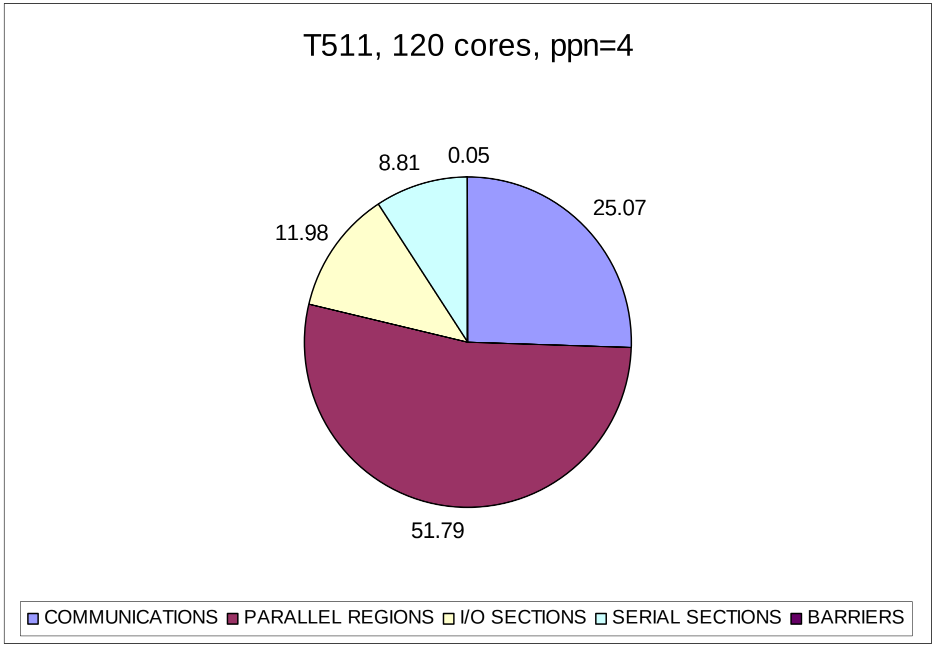




Dual-Core Intel® Xeon® 5160 processor, 2-socket sys., 3.0GHz, 4MB L2 cache, 8GB Memory

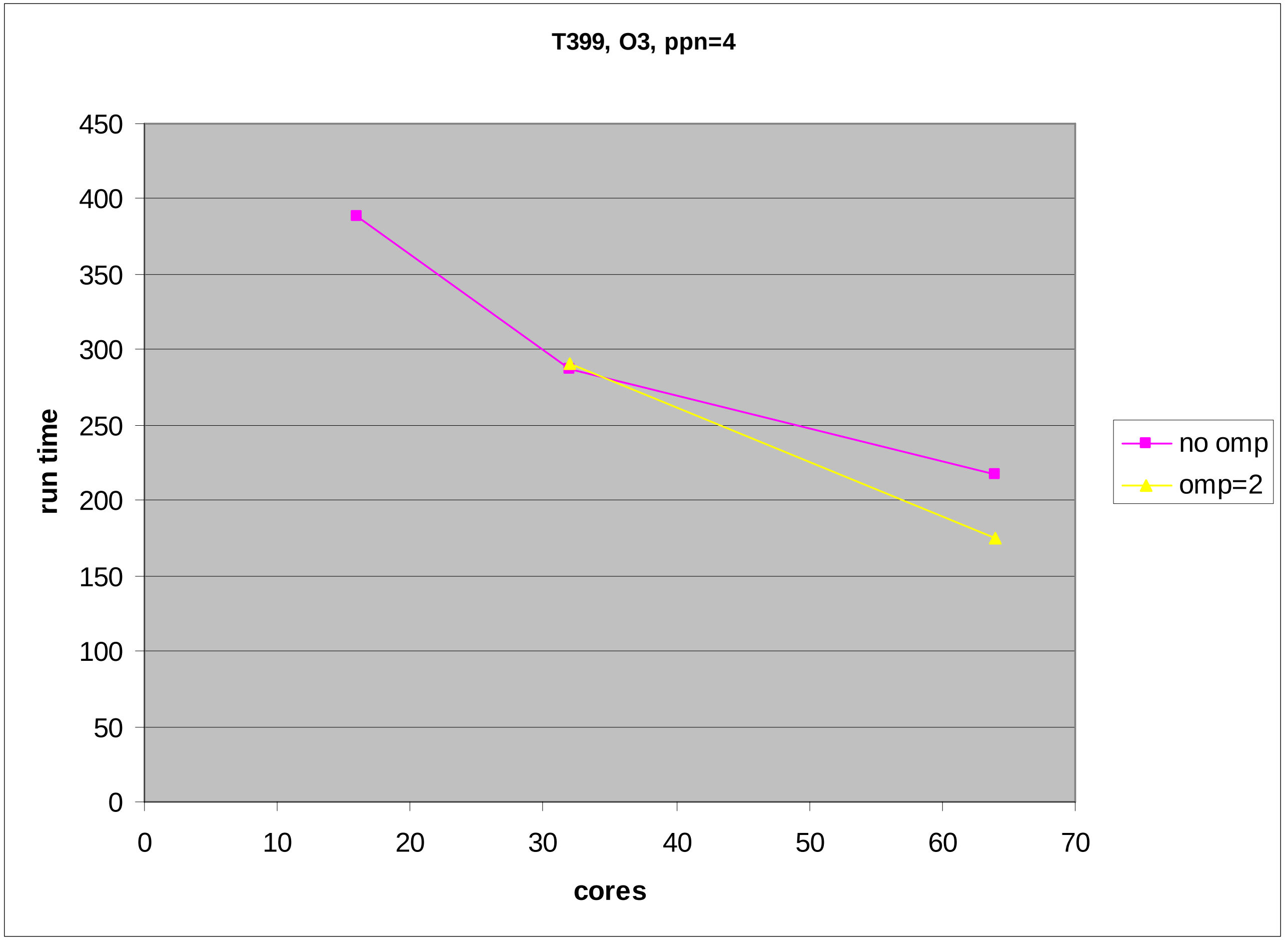
T511 CPU-time as percent of total





Dual-Core Intel® Xeon® 5160 processor, 2-socket sys., 3.0GHz, 4MB L2 cache, 8GB Memory

Benefits of Hybrid optimization



Dual-Core Intel® Xeon® 5160 processor, 2-socket sys., 3.0GHz, 4MB L2 cache, 8GB Memory

4D-Var T159 case

4D-Var T159 case, O3, 32 cores, ppn=4	
ifstraj_0	402
ifstmin_0	736
ifstraj_1	119
ifstmin_1	602
ifstraj_2	352

Dual-Core Intel® Xeon® 5160 processor, 2-socket sys., 3.0GHz, 4MB L2 cache, 8GB Memory

Conclusions:

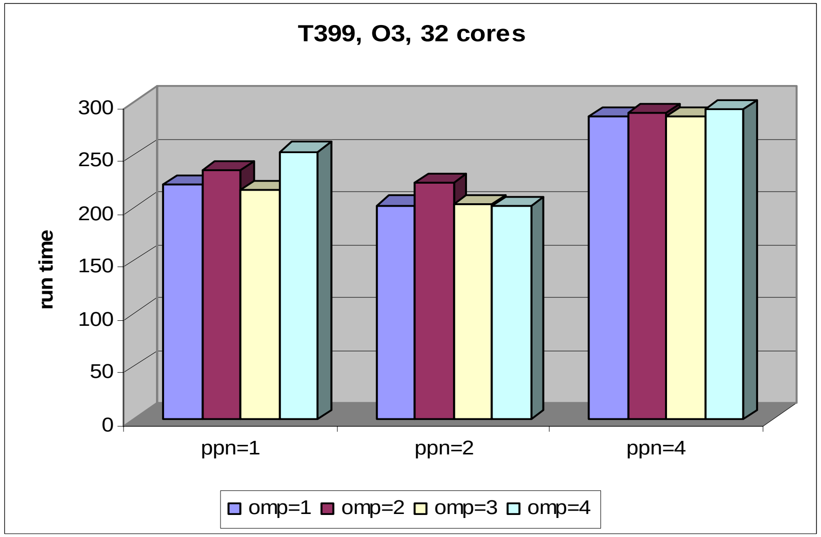
- ECMWF IFS RAPS ported on Altix and Woodcrest clusters.
- All benchmarks passed validation on both systems.
- Running benchmarks on Intel’s platforms showed high efficiency of the processor both in computations and scalability.
- There is a number of hot-spots and bottle necks that have not be processed yet.
- We are going to explore more intensively Intel® MKL for benchmark optimization: to use not only its vectorized math functions but more complex routines and solvers.

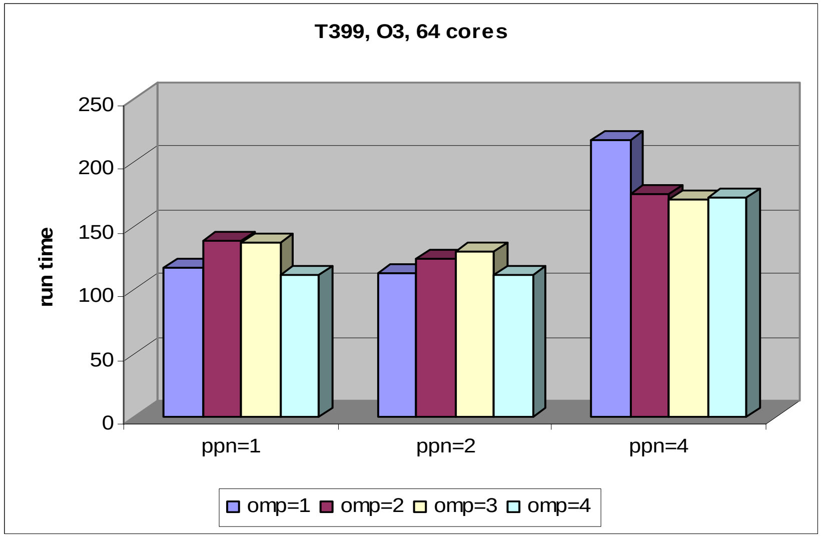
Future works

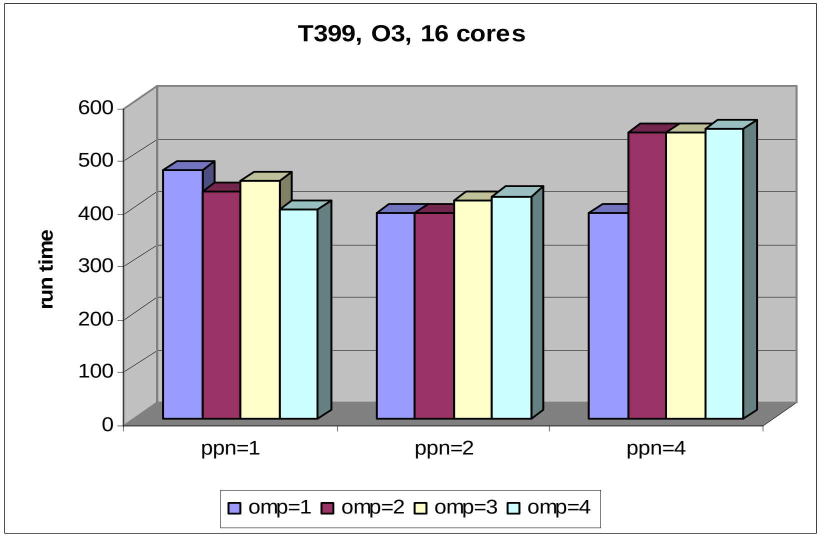
- Compiler optimizations (switches, directives, …)
- Optimization of MPI collaborations
- Using more Intel MKL in the code (FFT, solvers, …)
- Tuning the code

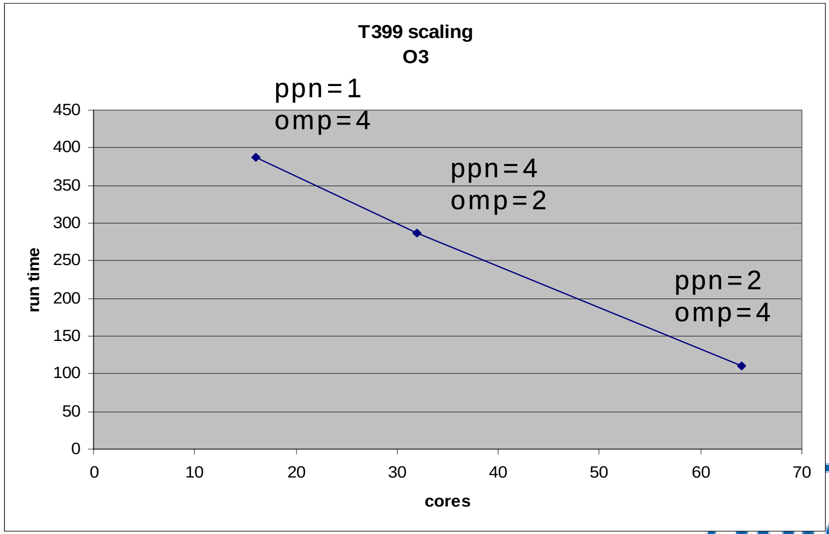
Backup

Scaling


T399, O3, 32 cores
run time
300
250
200
150
100
50
0
ppn=1
ppn=2
ppn=4
□ omp=1 □ omp=2 □ omp=3 □ omp=4

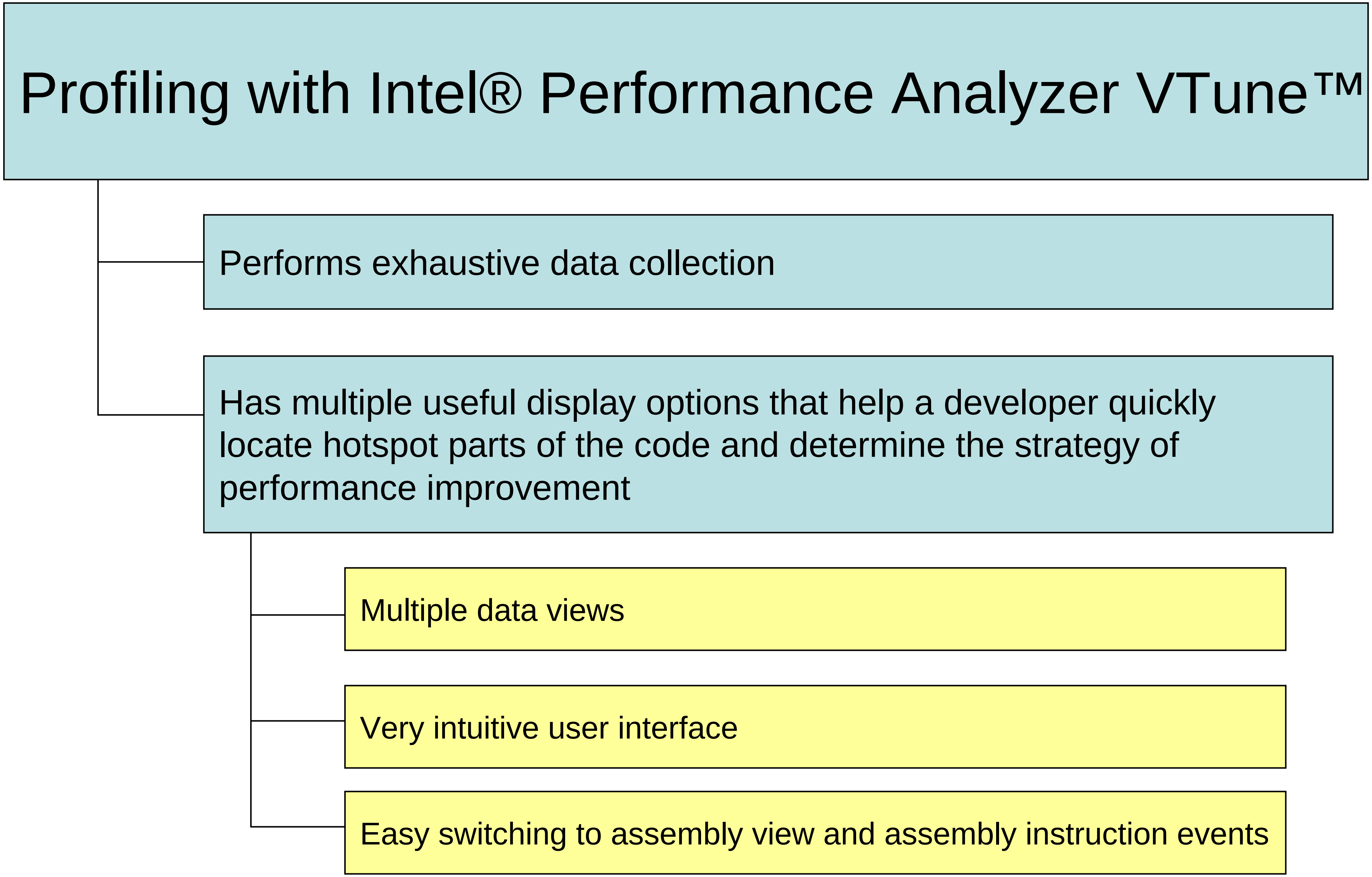

T399, O3, 64 cores
run time
250
200
150
100
50
0
ppn=1
ppn=2
ppn=4
□ omp=1 □ omp=2 □ omp=3 □ omp=4


T399, O3, 16 cores
run time
600
500
400
300
200
100
0
ppn=1
ppn=2
ppn=4
□ omp=1 □ omp=2 □ omp=3 □ omp=4


T399 scaling
O3
ppn=1
omp=4
ppn=4
omp=2
ppn=2
omp=4
run time
450
400
350
300
250
200
150
100
50
0
0
10
20
30
40
50
60
70
cores

Useful Intel® Compiler Options for Woodcrest

- **-O2** Turns on default optimizations for speed
- **-O3** Enables -*O2* optimization level and performs more aggressive optimizations, in particular, loop transformations
- **-ip/ipo** Enables single−/multi−file interprocedural optimizations
- **-no-prec-div, -no-prec-sqrt** Enable use of faster but slightly less accurate algorithms for division and square root (it may affect floating-point accuracy)
- **-xT** Enables use of specific optimization for Woodcrest
- **-unroll0** Disables unrolling the loops in the file



Profiling with Intel® Performance Analyzer VTune™

- Performs exhaustive data collection
- Has multiple useful display options that help a developer quickly locate hotspot parts of the code and determine the strategy of performance improvement
  - Multiple data views
  - Very intuitive user interface
  - Easy switching to assembly view and assembly instruction events