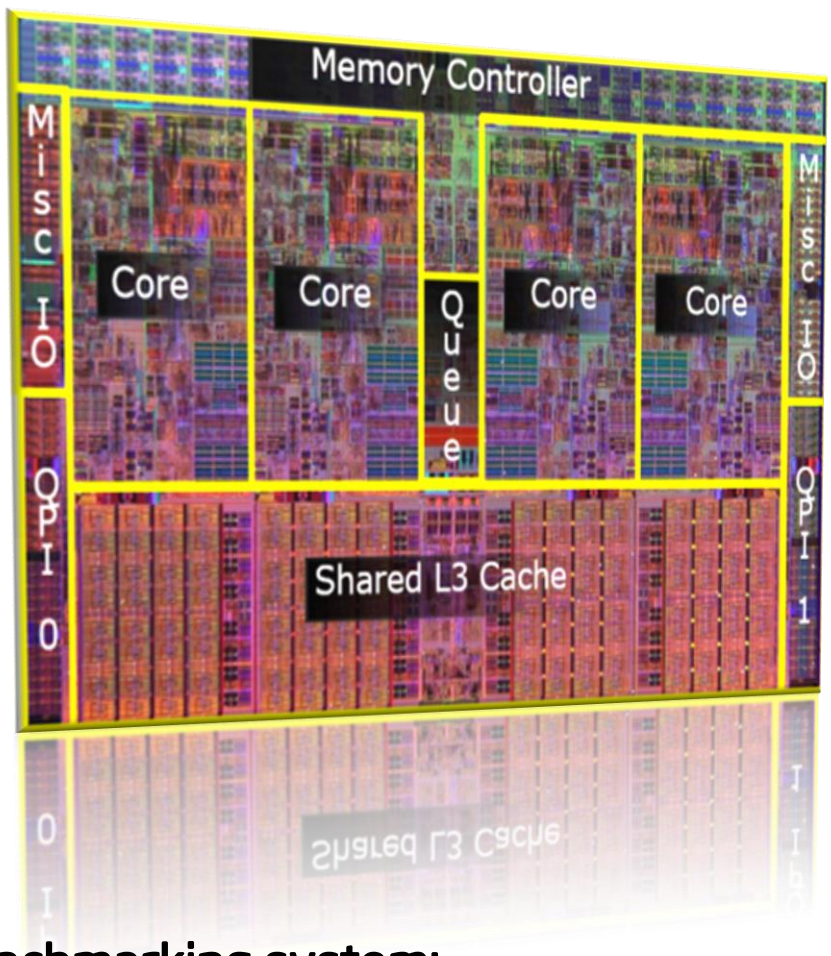
Improving WRF performance on the latest Intel® based platforms

Roman Dubtsov*, Alexander Kosenkov, Alexander Semenov, Dmitry Shkurko
{roman.s.dubtsov,alex.kosenkov,alexander.l.semenov,dmitry.v.shkurko}@intel.com

Executive summary

- WRF 3.0.1 benchmarking results on latest Intel® Xeon® processor 5500 series show substantial improvement over previous generation of processors
- Intel® software tools can be used to visibly improve WRF performance
- Beta version of latest WRF release – WRFV3.1 – has been tested on major Intel® based platforms

Intel® Xeon® processor 5500 series overview and benchmarking system setup



- 4 cores with dedicated L2 and L1 caches
  - 2 hardware threads per core if Intel® Hyper-Threading Technology (Intel® HT Technology) is enabled
- Shared L3 cache up to 8MB
- Intel® Turbo Boost Technology which increases CPU performance on demand
- Integrated 3-channel DDR3 memory controller up to 1333MHz
- 2 Intel® QuickPath Interconnect links up to 6.4GT/s
  - 1st link connects sockets
  - 2nd link connects socket with I/O controller

Benchmarking system:

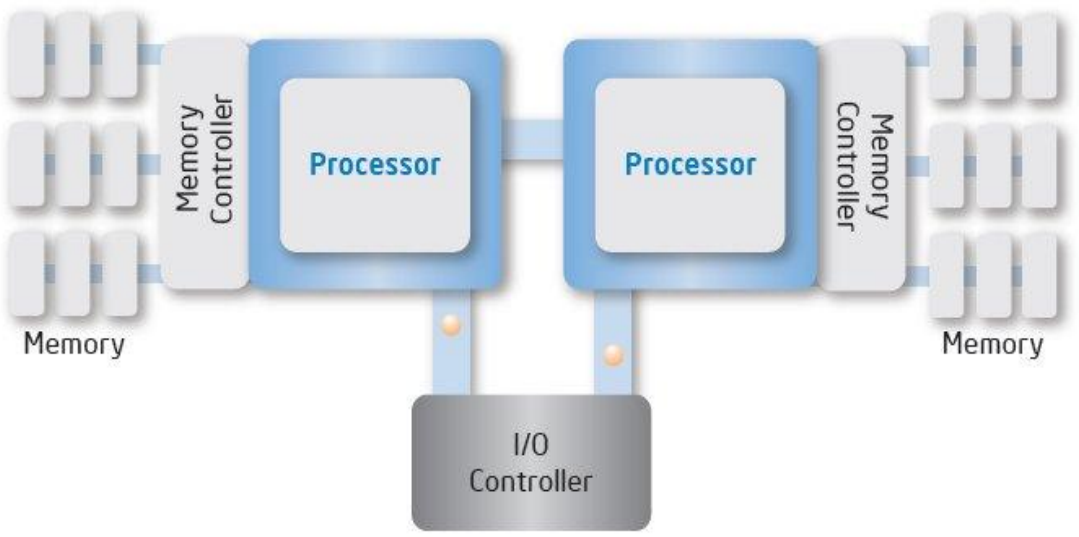
More than 200 nodes and growing

Hardware:

- 2x Intel® Xeon® 5560 processor: 2.8GHz, 8MB L3 cache; 8 cores per node
- Mellanox ConnectX* QDR InfiniBand* with Cisco* router; fat tree topology
- Lustre* and Panasas* cluster file systems

Software:

- RedHat Enterprise Linux* 5.x
- OFED 1.3.1

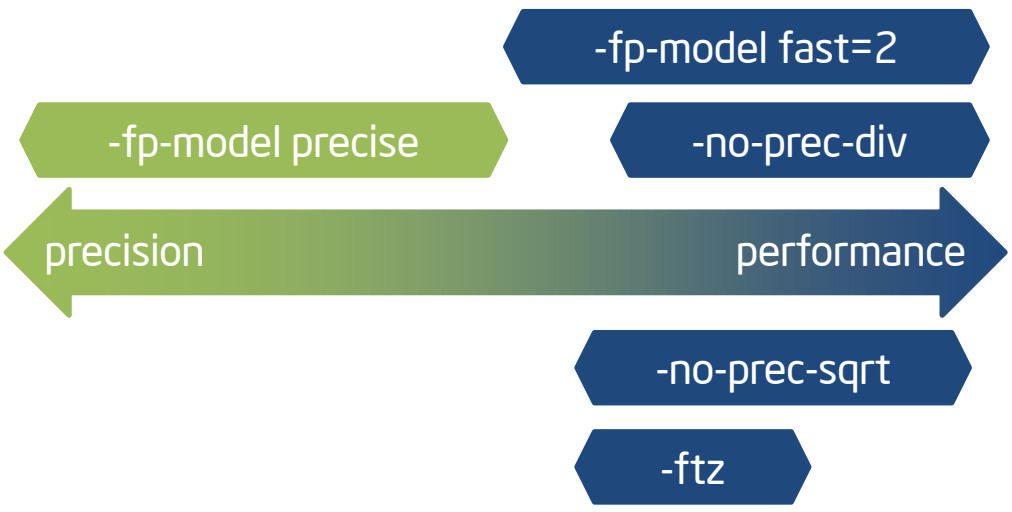


General optimization hints

Compiler flags optimizations

Precision settings:

Intel® compilers provide settings to trade performance for precision and vice versa:



Vectorization settings (11.1.x):

- -xSSE2, -xSSE3, -xSSSE3, -xSSE4.2: use SSEx for vectorization
- -xHost: use whatever SSE implementation is available on this machine

Optimizations for parallel performance

Process and thread affinity settings

- Process pinning improve cache utilization and ensure memory allocation on right NUMA node
- Intel® MPI library 3.2.x supports automatic process pinning (enabled by default)
  - support for hybrid MPI + OpenMP jobs
  - fine control over pinning settings for power users
- OpenMP threads can be pinned using KMP_AFFINITY environment variable

Exploiting Intel® HT Technology flexibly

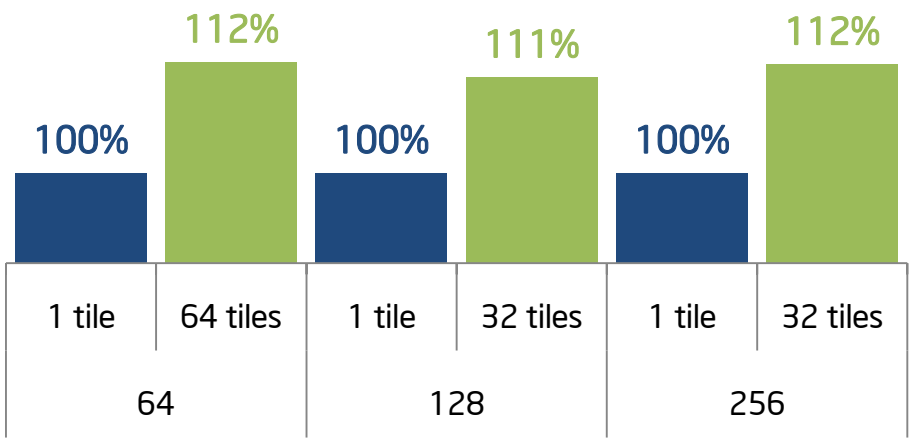
- Hides memory latency; good for workloads with low computations to memory accesses ratio or on high core counts
- Enabling/disabling Intel® HT Technology requires a reboot. If degradation is observed, it is advised to run application with single software thread per core to obtain same performance as with Intel® HT Technology disabled while leaving it enabled in BIOS.

WRF-specific optimizations

Memory access optimizations:

- WRF divides per-process patch into tiles that are processed separately by different OpenMP threads; it is possible to specify more tiles than OpenMP threads
- As tiles are processed separately; having multiple tiles reduces application working data set improving cache and memory bandwidth utilization

Speedup from multiple tiles/process for WRFV3/CONUS2.5km

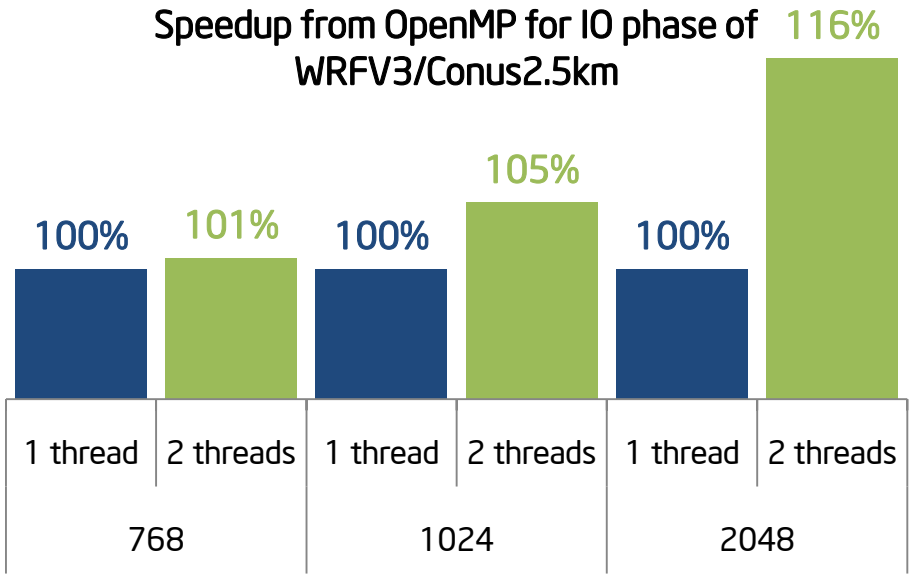


Optimization of communications:

Hybrid OpenMP + MPI configuration may reduce overhead from communications:

- Workloads with nesting use collective communications during forcing/feedback and during IO phase to gather/scatter data

Speedup from OpenMP for IO phase of WRFV3/Conus2.5km



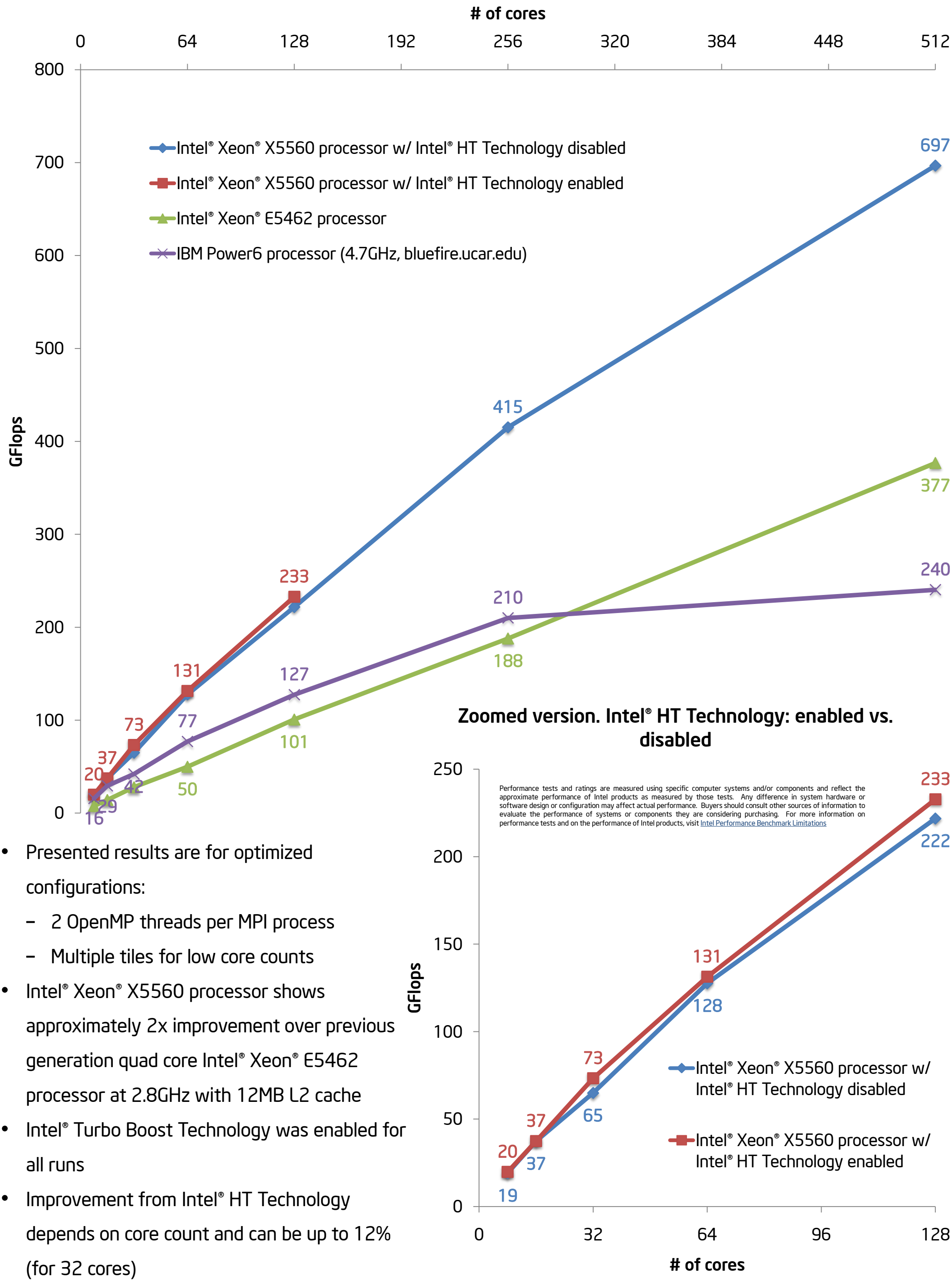
- Lower impact from latency for point-to-point communications during halo exchange

Benchmarking configuration:

- Intel® C/C++ and Fortran compilers 11.1.x
  - -O3 -xSSE4.2 -ip -fp-model fast=2 -no-prec-div –no-prec-sqrt –openmp
- Intel® MPI library 3.2.x

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

Benchmarking WRFV3.0.1/CONUS12km

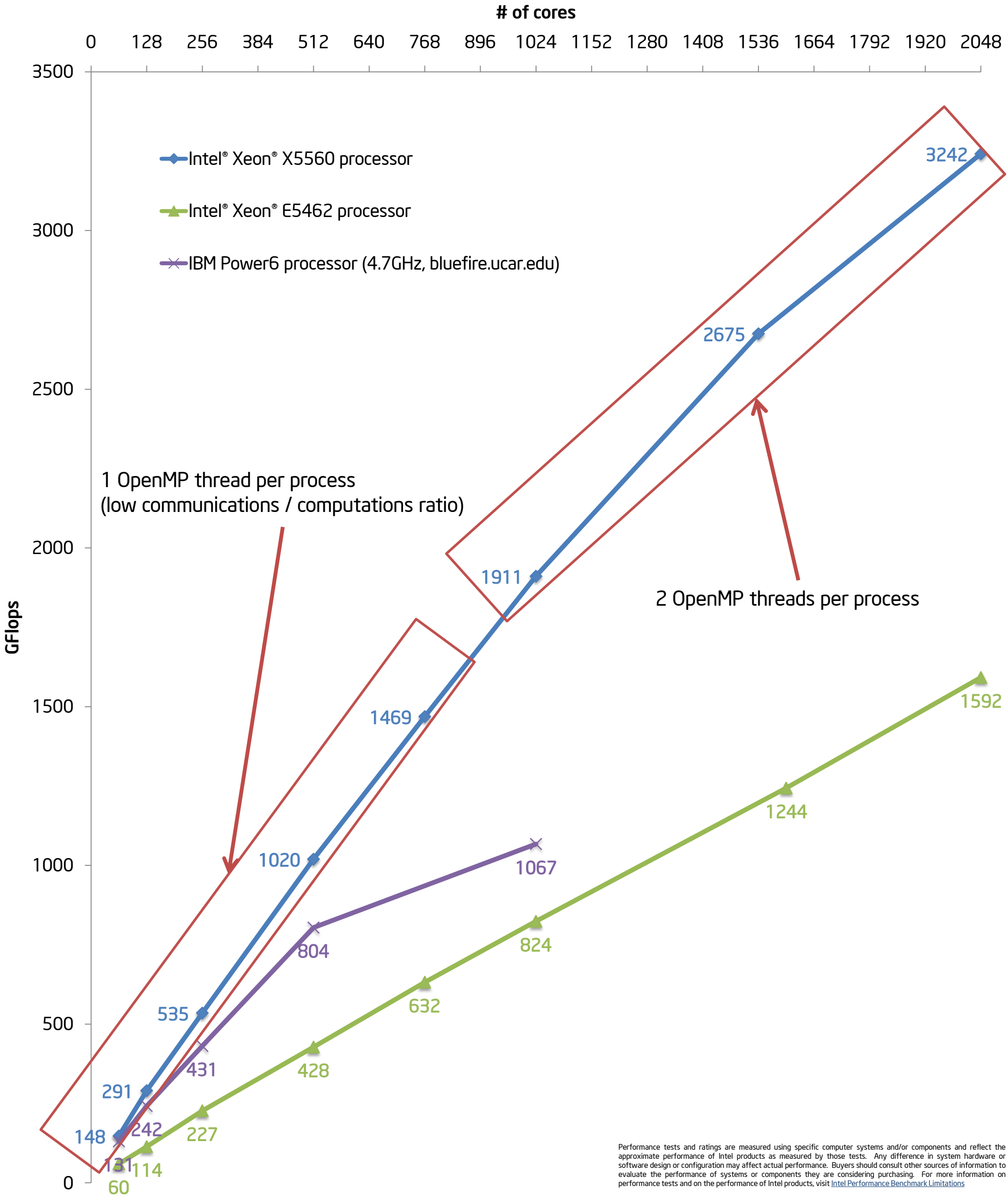


- Presented results are for optimized configurations:
  - 2 OpenMP threads per MPI process
  - Multiple tiles for low core counts
- Intel® Xeon® X5560 processor shows approximately 2x improvement over previous generation quad core Intel® Xeon® E5462 processor at 2.8GHz with 12MB L2 cache
- Intel® Turbo Boost Technology was enabled for all runs
- Improvement from Intel® HT Technology depends on core count and can be up to 12% (for 32 cores)

Zoomed version. Intel® HT Technology: enabled vs. disabled

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

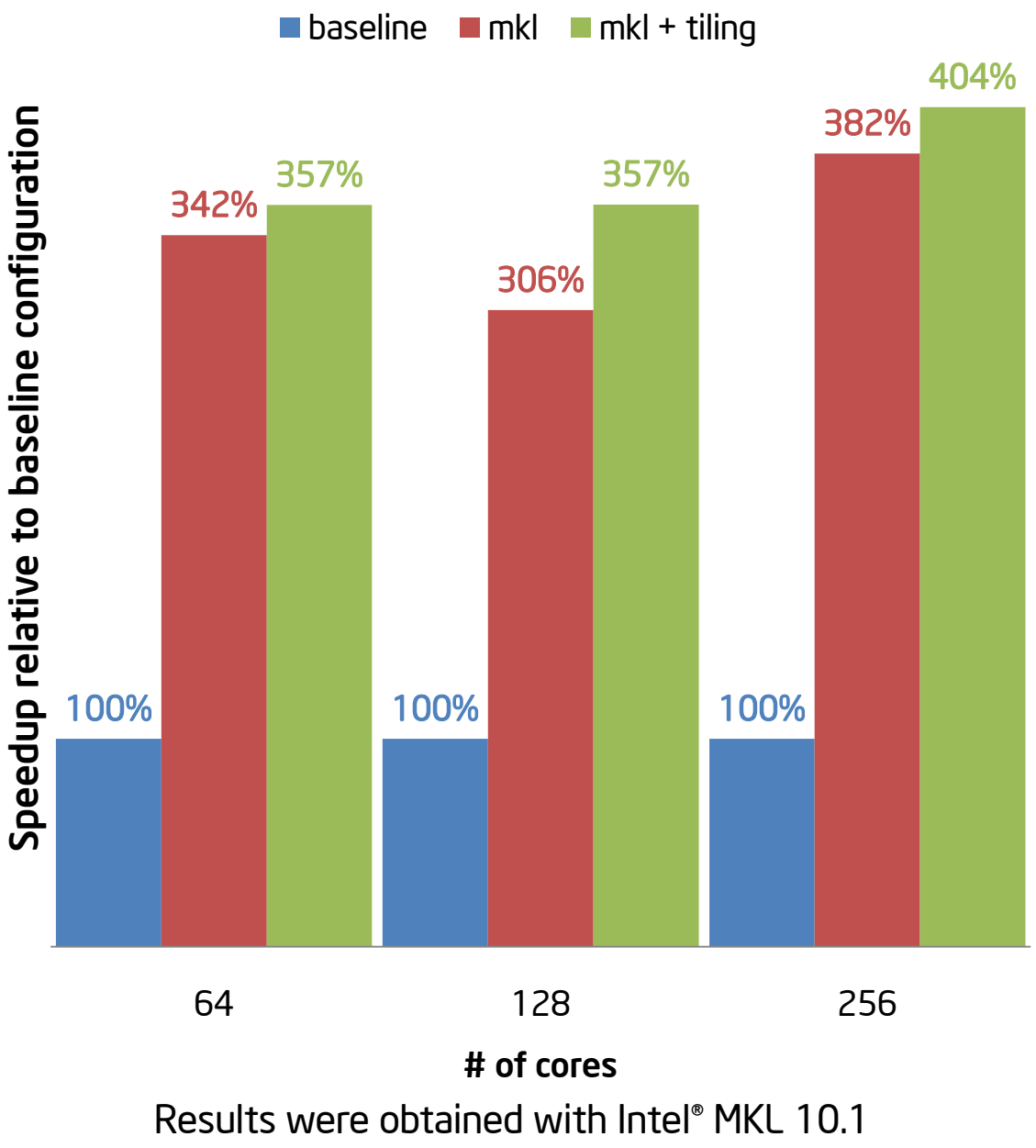
Benchmarking WRFV3.0.1/CONUS2.5km



Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

- Presented results are for optimized configurations
- Intel® Xeon® X5560 processor shows > 2x improvement over previous generation Intel® Xeon® E5462 processor
- Intel® Turbo Boost Technology was enabled for all runs
- No improvement from Intel® HT Technology was observed for this workload

Improving performance of Global WRF by using Intel® Math Kernel Library (Intel® MKL)



Results were obtained with Intel® MKL 10.1

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

- One of hotspots in planet-wide climate simulations is polar filter: it can account for up to 70% of computational step time
- Filtering is done using FFT and, by default, WRF uses fftpack5 FFT library developed in NCAR in mid-90's and not tuned for modern processors
- Replacing calls to fftpack5 with calls to Intel® MKL DFTI improved simulation performance > 3x despite additional overhead caused by pre- and post-processing of data that was required to match FFT definition used by fftpack5
- Increase in number of tiles brings further improvements
- Source code patch available on request

WRFV3.1 on Intel® processors

- WRFV3.1 beta (ARW core) was tested on major Intel® platforms used with WRF:
  - Linux on Intel® 64 and Intel® Itanium® 2
  - Apple Macs with Intel® processors
- No problems with WRF code were found in this release
- Build-time configuration boilerplates remained the same
  - Serial, SM, DM and SM+DM configurations supported
  - Supported MPIs: MPICH, Intel® MPI library, OpenMPI, SGI MPT
- Some compiler issues remain; please visit "Building WRF with the Intel® Compilers" webpage or contact authors for more info. Update for WRFV3.x and WPS is in progress.

Summary and acknowledgements

Intel® Xeon® 5500 series processors provide competitive performance and present a compelling improvement over previous generation of Intel processors.

Acknowledgements:

- John Michalakes (NCAR)
- IBM Power6 results by Davide Del Vento (NCAR)