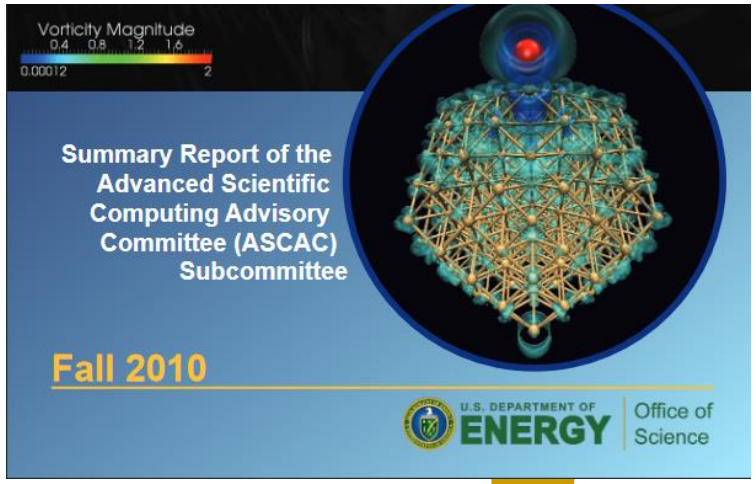
TALE(S) OF PARALLEL APPLICATION ENABLING

CCDSC - October 2016

Joseph Curley, Sr. Director Platform & Ecosystem Enabling

Enterprise and Government Group

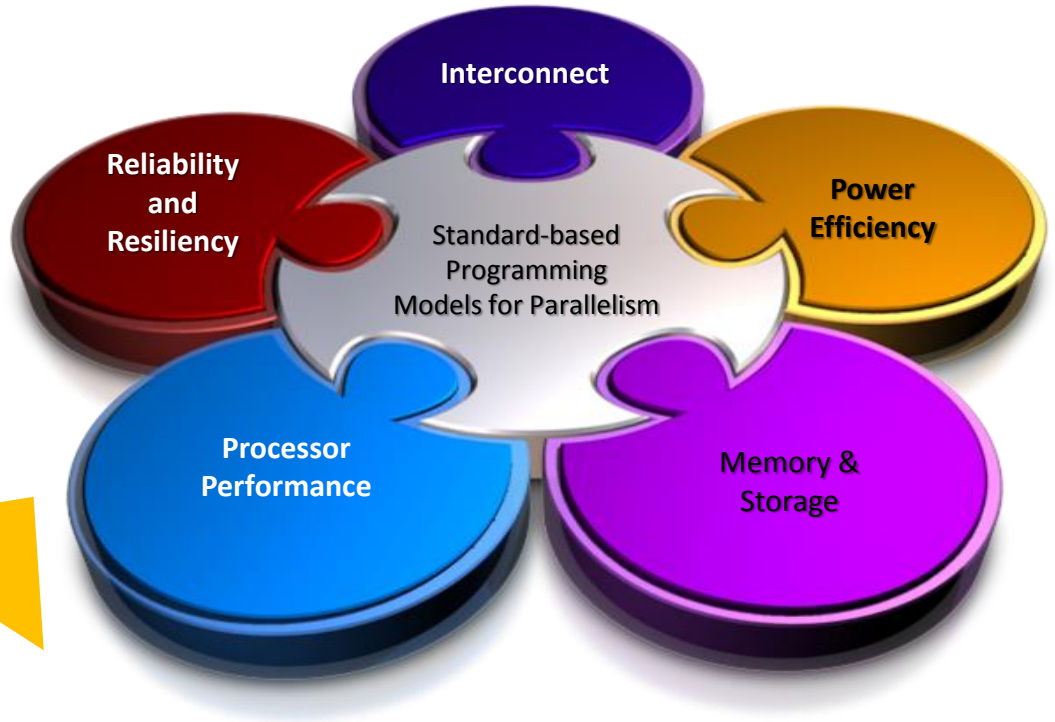
Has it really been 6 years, already?



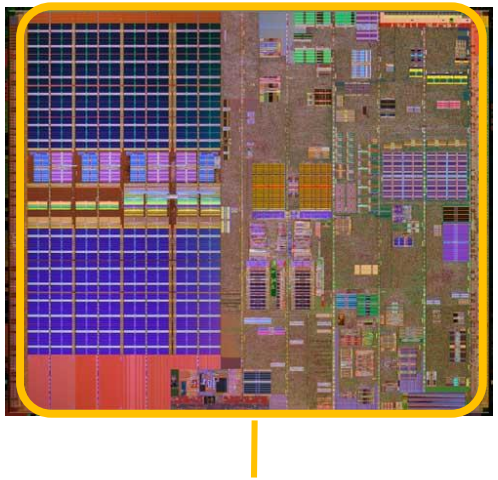
"Reducing power consumption"

"Coping with run-time errors"

"Exploiting Massive Parallelism"



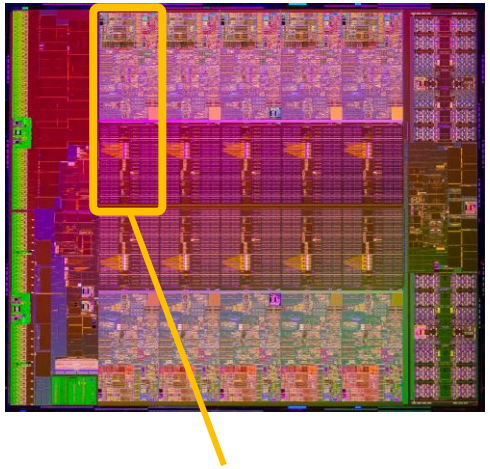
Increasingly Parallel CPUs



2005 “Irwindale” - 1 core

1 Thread per Core

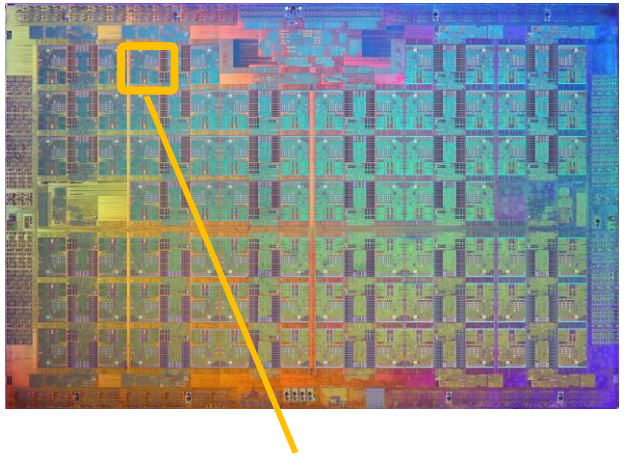
SIMD 4



2013 “Ivy Bridge” - 1 of 10 cores

2 Threads per Core

SIMD 8



2016 “Knights Landing” - 1 of 68 cores

4 Threads per Core

SIMD 16

The Kind of Thing We Wanted To Hear

"VERY Big advantage: Minimal Change to code base, all switched on/off by a single compiler option and a single compiler flag" *

– Martijn Marsman, Universitaet Wien, VASP speaking about OpenMP work on VASP application in collaboration with Intel at MCC-UKCP-ECPP workshop, 21-1-2016 Daresbury, United Kingdom

But…

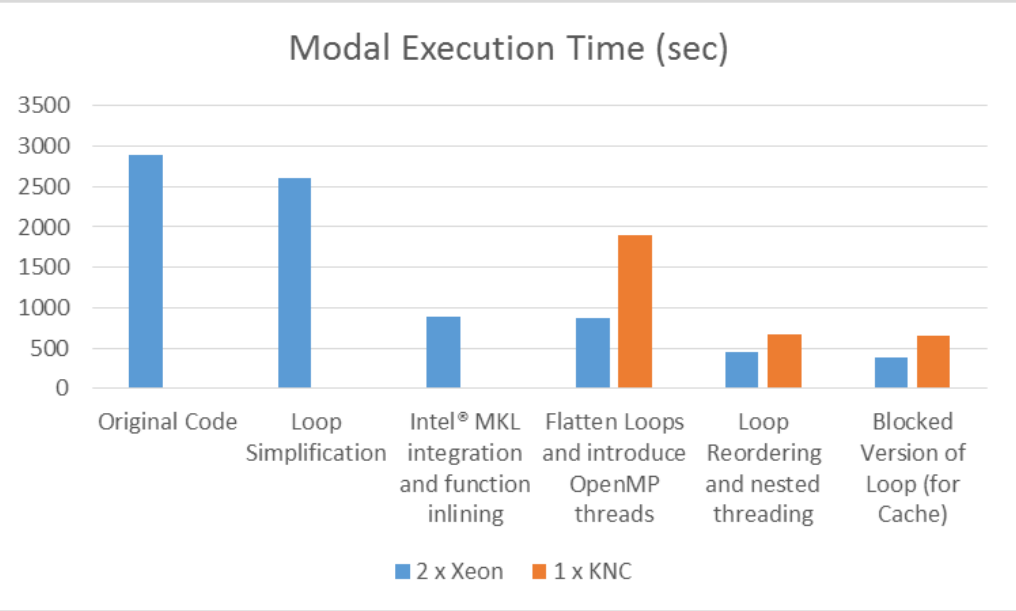
*It can also be a big disadvantage, in that it inhibits more revolutionary approaches, which we will get to later…

easy_to_program != programming_is_easy

MODAL – Cosmology Code in the Cambridge DAMTP Intel® Parallel Computing Center

Objective – reduce from compute years to compute days

→Traditional optimization approaches yield 6.5x Xeon Speedup, KNC at .6 (1 KNC >= 1 Xeon)



	2 x Xeon	1 x KNC	Speedup to Baseline	KNC Speedup
Original Code	2887			
Loop Simplification	2610		0.1	
Intel® MKL integration and function inlining	882		2.3	
Flatten Loops and introduce OpenMP threads	865.9	1901.6	2.3	0.5
Loop Reordering and nested threading	450.6	667.9	5.4	0.7
Blocked Version of Loop (for Cache)	385.6	655	6.5	0.6

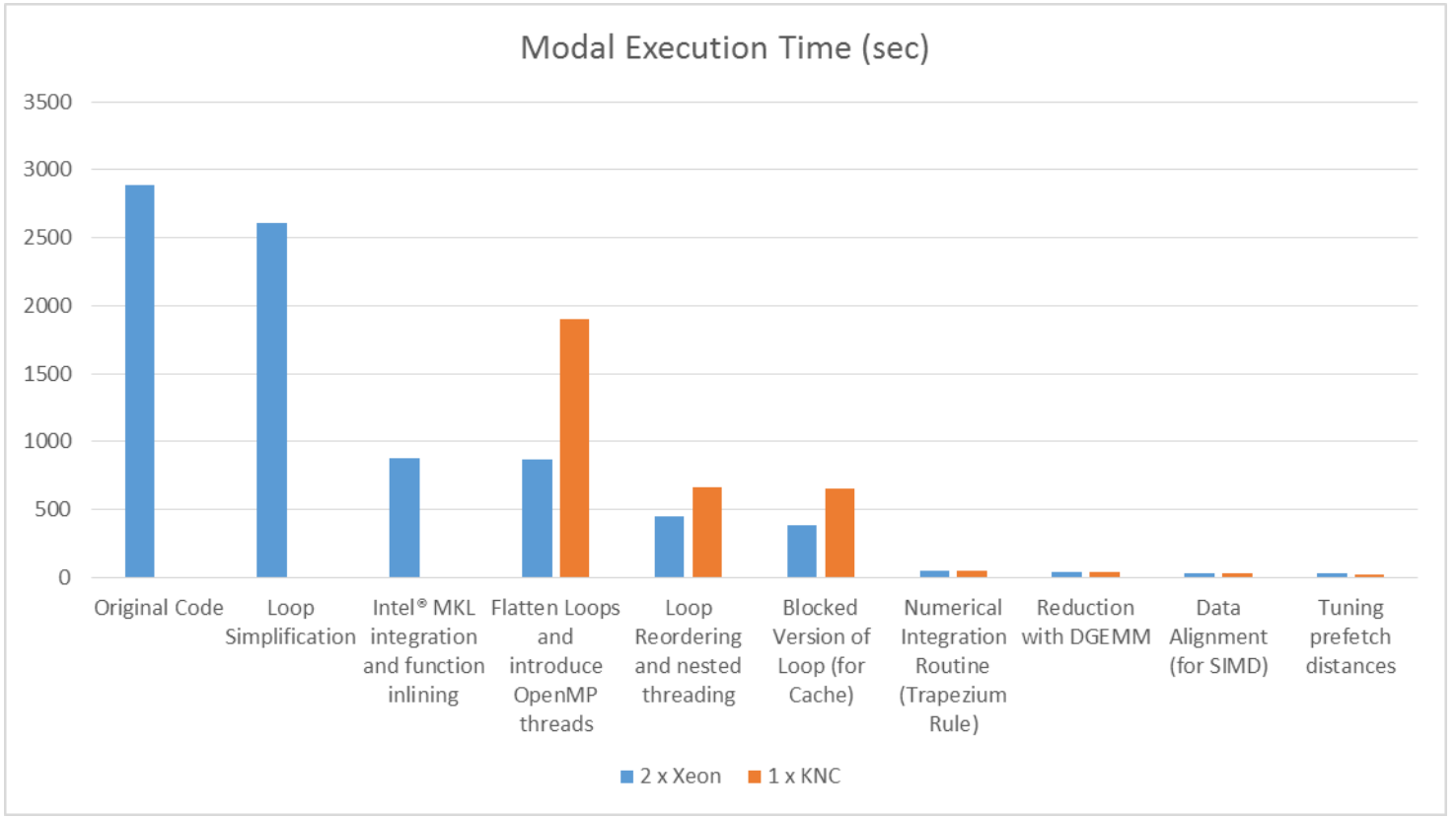
Performance was measured by Cambridge DAMTP on their SGI UV2000 with 192 Intel® Xeon® E5 4650 CPUs, with 128 GBs of DDR memory. Xeon Phi numbers are measured on Intel9r) Xeon Phi™ 5110P coprocessors running at 1.065 GHz with 8 gigabytes of GDDR5 memory.
Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at www.intel.com.
Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.
See optimization notice

Should one conclude Modal should be a "Xeon" application and just be satisfied with the 6.5x speedup?

Revaluating Codes for the Many Core Era

	2 x Xeon	1 x KNC
Original Code	2887	
Loop Simplification	2610	
Intel® MKL integration and function inlining	882	
Flatten Loops and introduce OpenMP threads	865.9	1901.6
Loop Reordering and nested threading	450.6	667.9
Blocked Version of Loop (for Cache)	385.6	655
Numerical Integration Routine (Trapezium Rule)	46.9	49.5
Reduction with DGEMM	37.4	37.7
Data Alignment (for SIMD)	35.1	34.5
Tuning prefetch distances	34.3	26.6

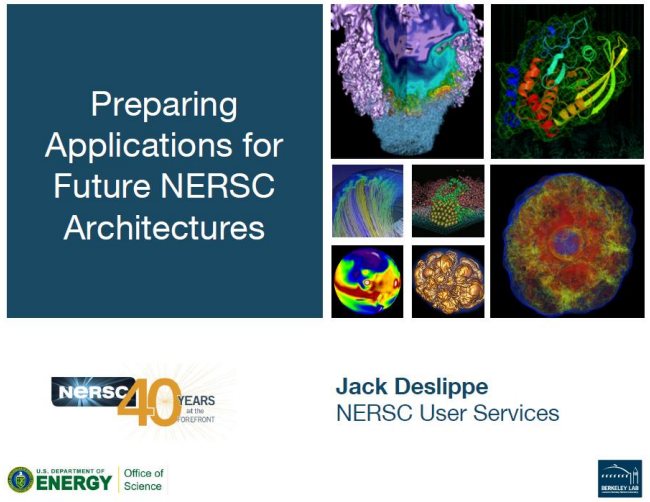
Re-thinking the compute algorithm for many core: speedup from 6.5x to 60x.

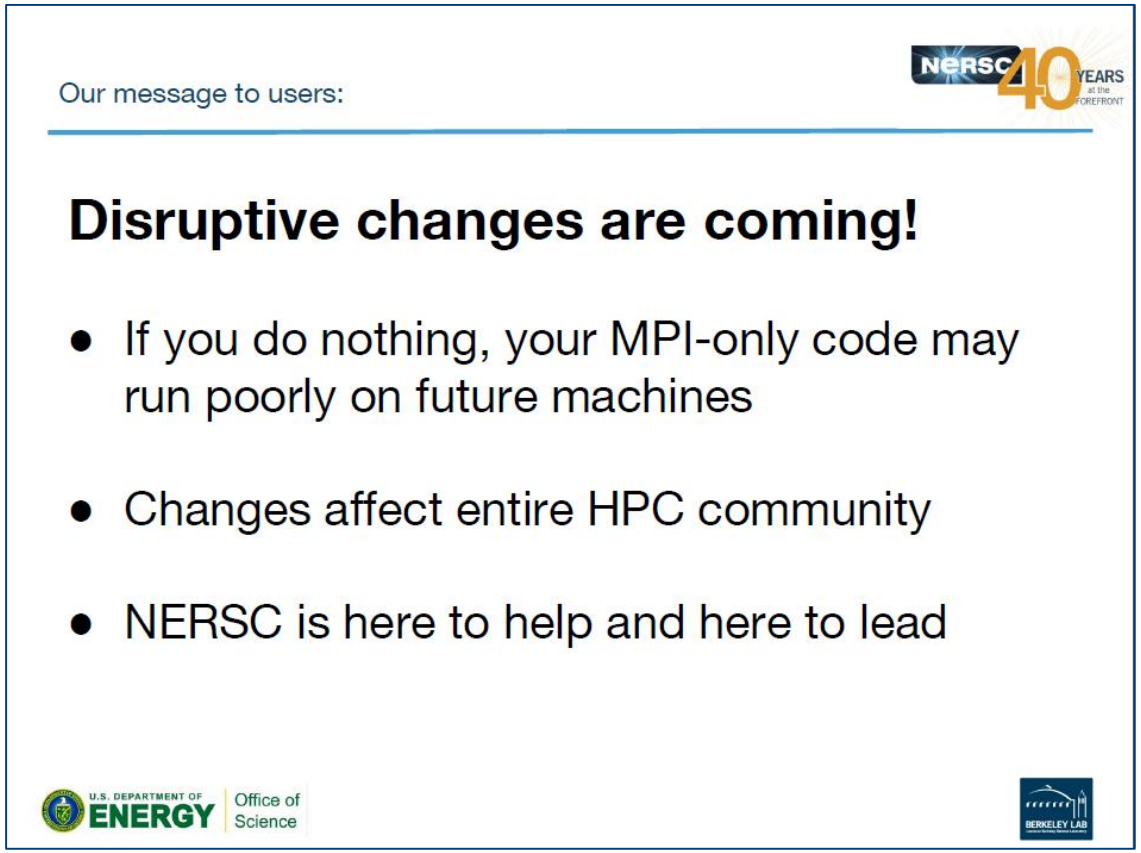


Algorithmic re-design for many core yielded a 10x improvement, followed by new incremental improvements.

Preparing applications for massive core counts…

October 2013 – Jack Deslippe, NERSC





PARSEC – From MPI to Hybrid Parallelism

PARSEC IS…

- DFT code "without" a basis set (real-space grid)
- Designed to be massively parallel, with simple "physics" layer
- A few decades of development and publications
- Efficient non-linear solver from Yousef Saad et al.
- Challenges towards Cori/KNL different than other DFT codes –
  - "Non-canonical" stencil code
  - Grid partitioning and data preconditioning
  - Tall-skinny subspace

THE DEVELOPMENT TEAM

- Multinational (US, Europe, Israel, India)
- Active research on new features - "Physics"
- …multiple "production" branches
- 25 year legacy of FORTRAN code
- NESAP and code modernization – Ariel Biller (PhD chemistry)

PARSEC in a nutshell	GOOD or BAD?
Solver using BLAS, LAPACK only one "legacy" library doing CG	Good
3D Hamiltonian as sparse matrix, Wavefunctions & density as 1D arrays	Neither
Real-space grid naively "flattened" and distributed	Bad & good
Potentially free of all-to-all in kernel	WOW!!
Serial portions in pre/post	Bad

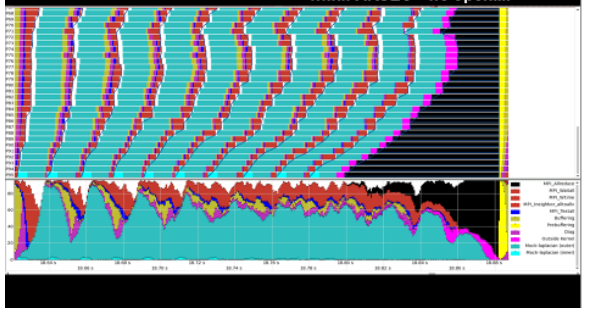
"BEFORE"

ITAC trace without annotations, excessive barriers.

HYBRID VERSION 1

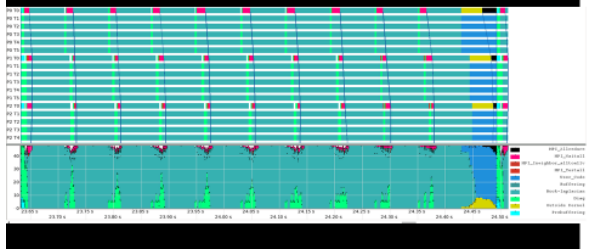
96 CPUS, 96 MPI PROCS

miniPARSEC – no openMP



16X6 THREADS

Same kernel, neighborhood alltoallv
Each MPI PROC has own NUMA domain
30% faster!



NEXT VERSION:

Better overlap, dynamic load balancing in openMP (guided)
Tried to do manual progression with mpi_test

(Note the PARSEC developers have let me know these slides are out of date, b) they are 4x faster than this and, c) there is more coming. But the point of the slide isn't about absolute performance)

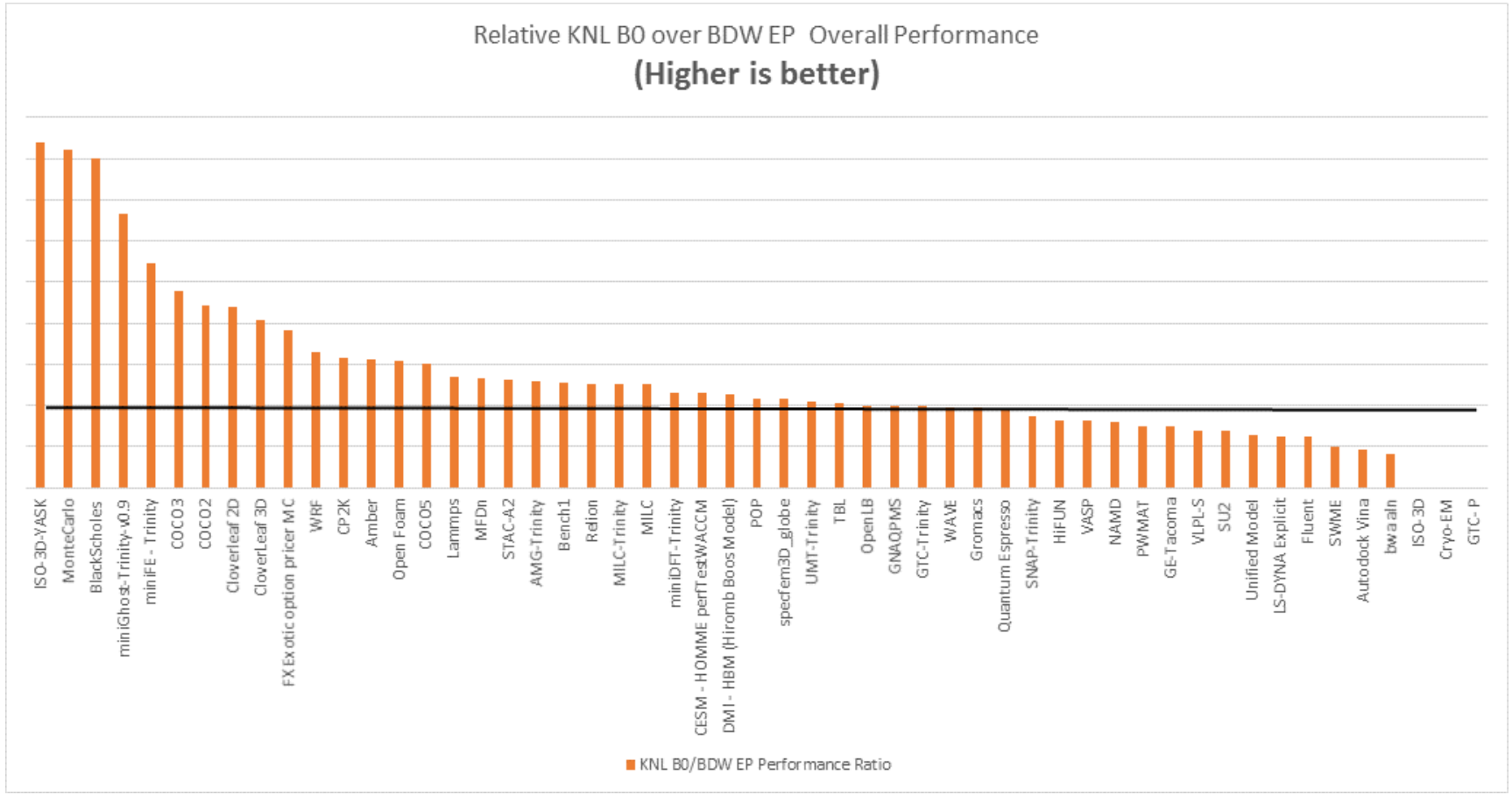
June 2016 performance survey of Knights Landing

Baseline is 2x18 Core Intel® Xeon Processor E5 2600 v4 family

Minimal Optimization - roughly out-of-the-box

Plenty of optimization headroom – SIMD, flat v. cache study, etc.

Lowest-end results roughly = 1 Broadwell
Highest-end results roughly = 8 Broadwells



One Challenge: The un-optimized performance may be good enough to ignore "Revolutionary" benefits

Performance was measured at Intel in Labs by Intel employees. Configurations information can be found on the following slide.
Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at www.intel.com.
Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.
See optimization notice

Summary

Legacy codes run – throughput ‘surprisingly’ good

Measurable benefits from Hybrid OpenMP + MPI

Good results on Knights Landing != Optimal results on Knights Landing


intel®
experience
what's inside™

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPETY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel may make changes to specifications and product descriptions at any time, without notice.

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Sandy Bridge and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to http://www.intel.com/performance

Intel, Core, Xeon, VTune, Cilk, Intel and Intel Sponsors of Tomorrow. and Intel Sponsors of Tomorrow. logo, and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Hyper-Threading Technology: Requires an Intel® HT Technology enabled system, check with your PC manufacturer. Performance will vary depending on the specific hardware and software used. Not available on all Intel® Core™ processors. For more information including details on which processors support HT Technology, visit http://www.intel.com/info/hyperthreading

Intel® 64 architecture: Requires a system with a 64-bit enabled processor, chipset, BIOS and software. Performance will vary depending on the specific hardware and software you use. Consult your PC manufacturer for more information. For more information, visit http://www.intel.com/info/em64t

Intel® Turbo Boost Technology: Requires a system with Intel® Turbo Boost Technology capability. Consult your PC manufacturer. Performance varies depending on hardware, software and system configuration. For more information, visit http://www.intel.com/technology/turboboost

Optimization Notice

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

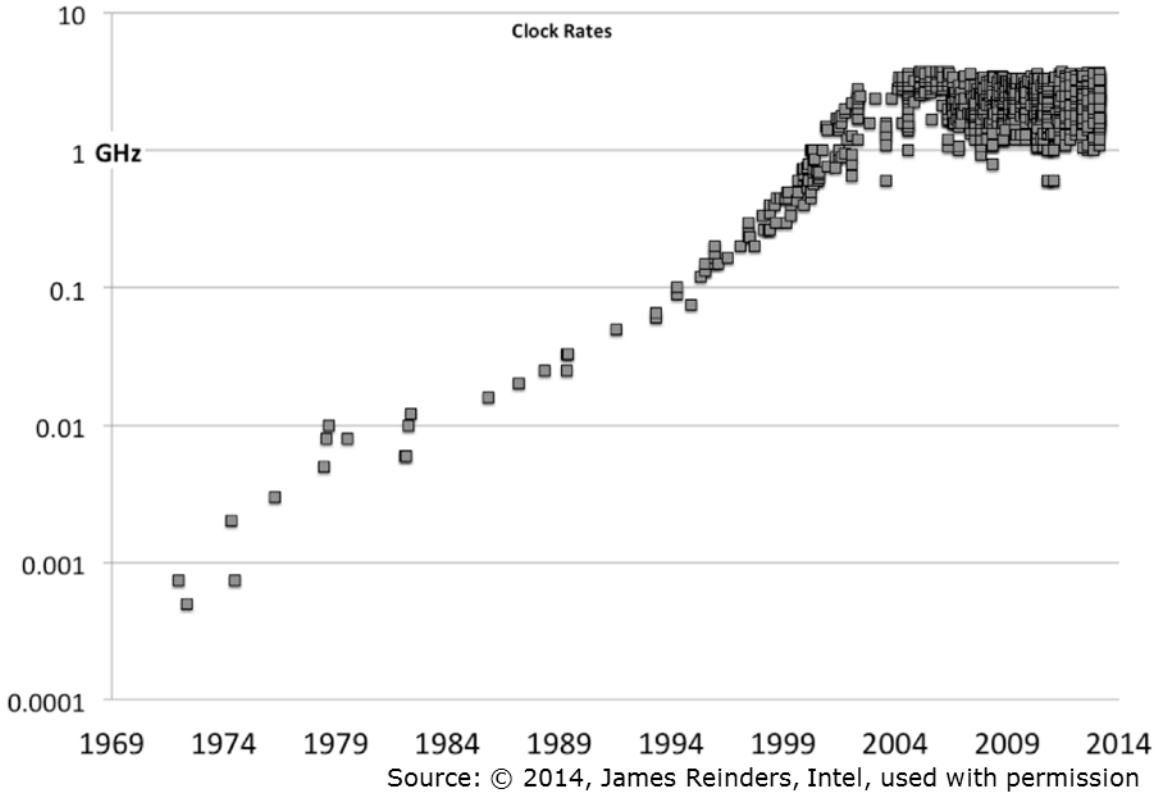
Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

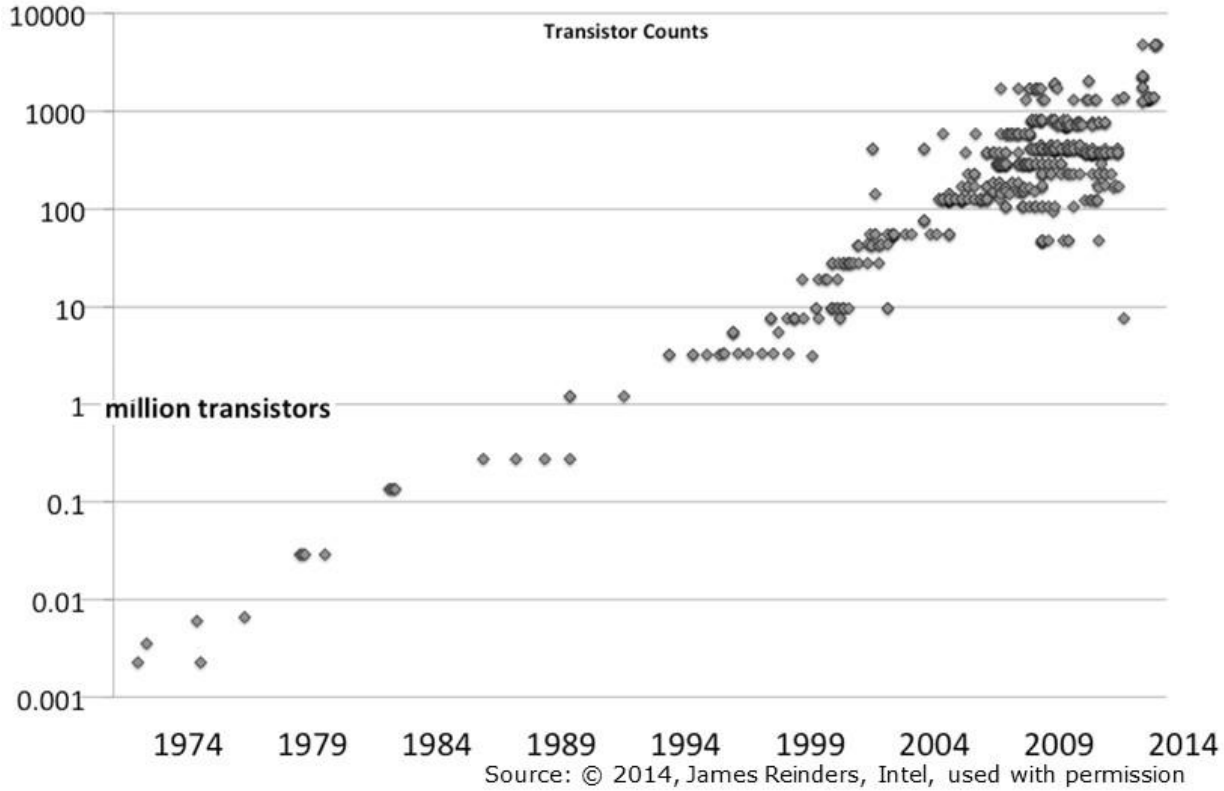
Notice revision #20110804

The “Right Hand Turn”

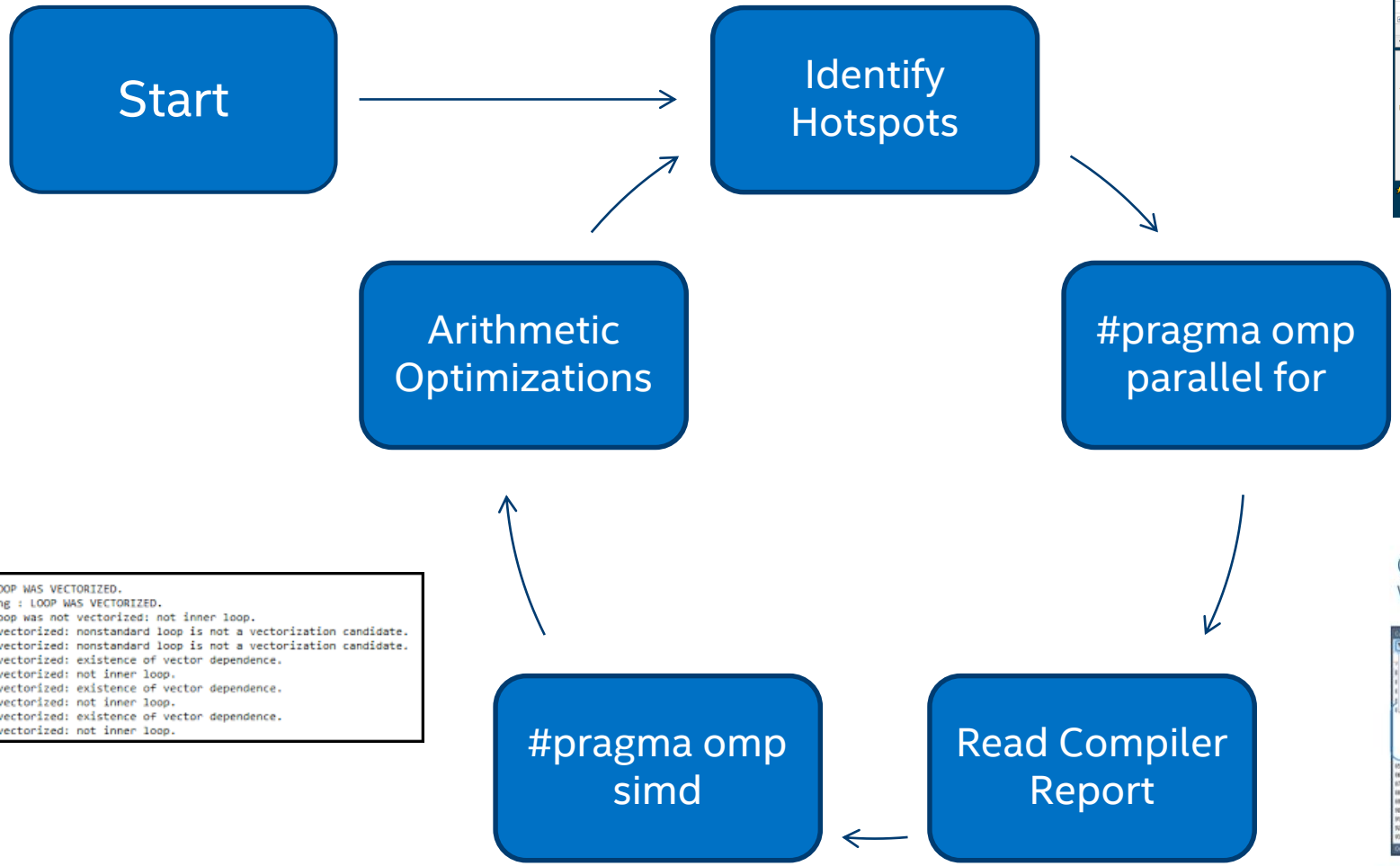
While IPC has grown through microarchitecture, frequency no longer provides ‘a free lunch’

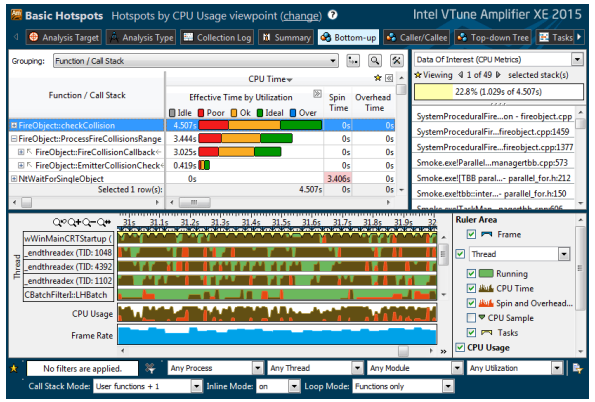


But “Moore’s Law” has pressed on

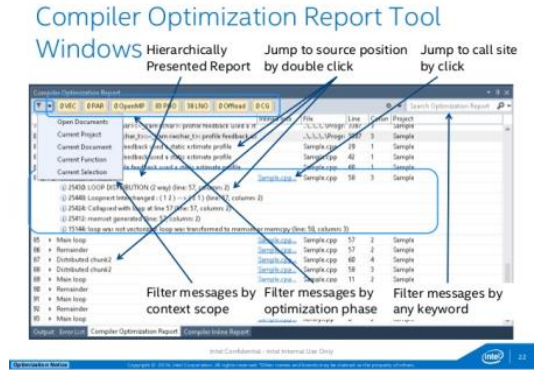


The “Evolutionary” Approach



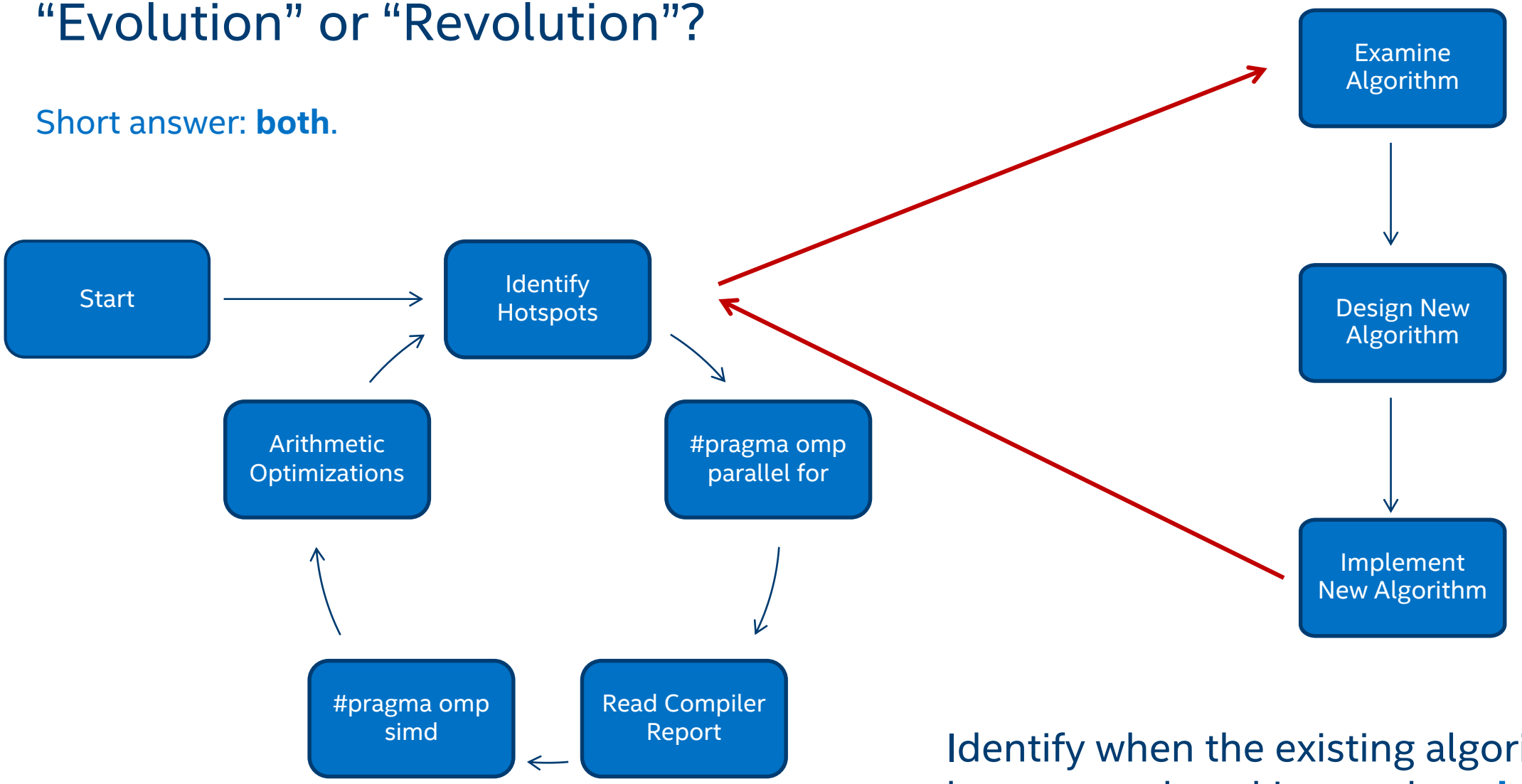



```
\main.cpp(30): warning : LOOP WAS VECTORIZED.
\scalar_dep.cpp(80): warning : LOOP WAS VECTORIZED.
\main.cpp(47): warning : loop was not vectorized: not inner loop.
): warning : loop was not vectorized: nonstandard loop is not a vectorization candidate.
): warning : loop was not vectorized: nonstandard loop is not a vectorization candidate.
): warning : loop was not vectorized: existence of vector dependence.
): warning : loop was not vectorized: not inner loop.
): warning : loop was not vectorized: existence of vector dependence.
): warning : loop was not vectorized: not inner loop.
): warning : loop was not vectorized: existence of vector dependence.
): warning : loop was not vectorized: not inner loop.
```



2nd IPCC Meeting - ICHEC
"Evolution" or "Revolution"?

Short answer: **both**.



Identify when the existing algorithm has stopped working, and **re-think**