
# LOI DE MOORE, LOI D'AMDHAL, QUEL FUTUR? LA VISION D'INTEL

**Philippe Thierry**
Sr Staff Engineer
Intel Corp.

**Marc Dollfus**
Responsable HPC secteur public et recherche
Intel France

Idris - 8 janvier 2009



## Legal Disclaimers

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 http://www.intel.com/performance/resources/limits.htm or call (U.S.) 1-800-628-8686 or 1-916-356-3104.

Results were derived using simulations run on an architecture simulator or model. Any difference in system hardware or software design or configuration may affect actual performance*

All dates and products specified are for planning purposes only and are subject to change without notice

Relative performance is calculated by assigning a baseline value of 1.0 to one benchmark result, and then dividing the actual benchmark result for the baseline platform into each of the specific benchmark results of each of the other platforms, and assigning them a relative performance number that correlates with the performance improvements reported.

SPEC, SPECint2000, SPECfp2000, SPECint2006, SPECfp2006, SPECjbb, are trademarks of the Standard Performance Evaluation Corporation. See http://www.spec.org for more information.

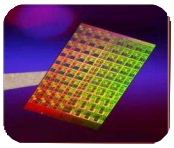
Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor series, not across different processor sequences. See http://www.intel.com/products/processor_number for details.

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications. All dates and products specified are for planning purposes only and are subject to change without notice

Intel, Intel Xeon, Intel Core microarchitecture, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

## Intel in High Performance Computing : the big picture



Advanced HPC R&D

Leading Process Manufacturing

Leading performance, performance/watt

Dedicated renowned expertise

Large scale clusters for test & optimization

Broad SW tools portfolio


Defined HPC application platform

Cluster Ready

Platform building blocks

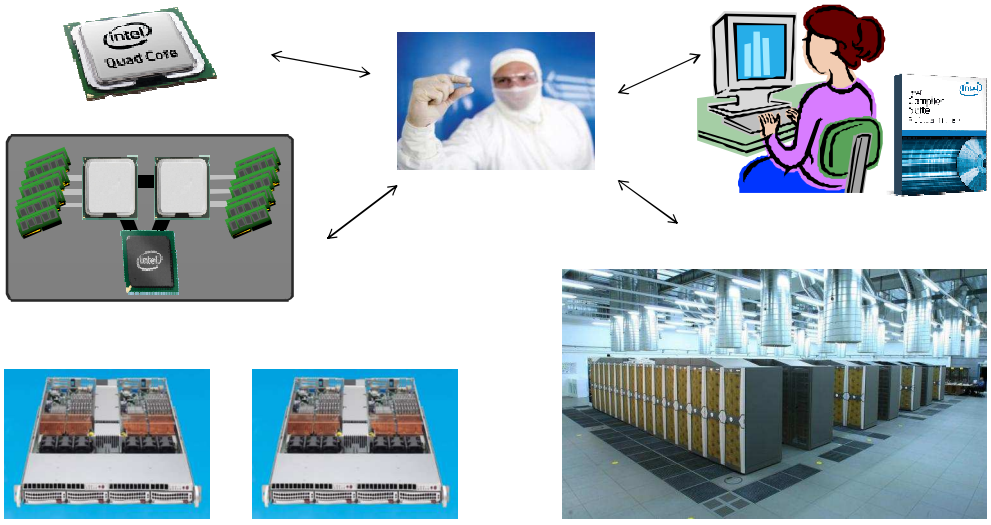
**A long term commitment to HPC**





## Agenda: from the micro-arch to the end users

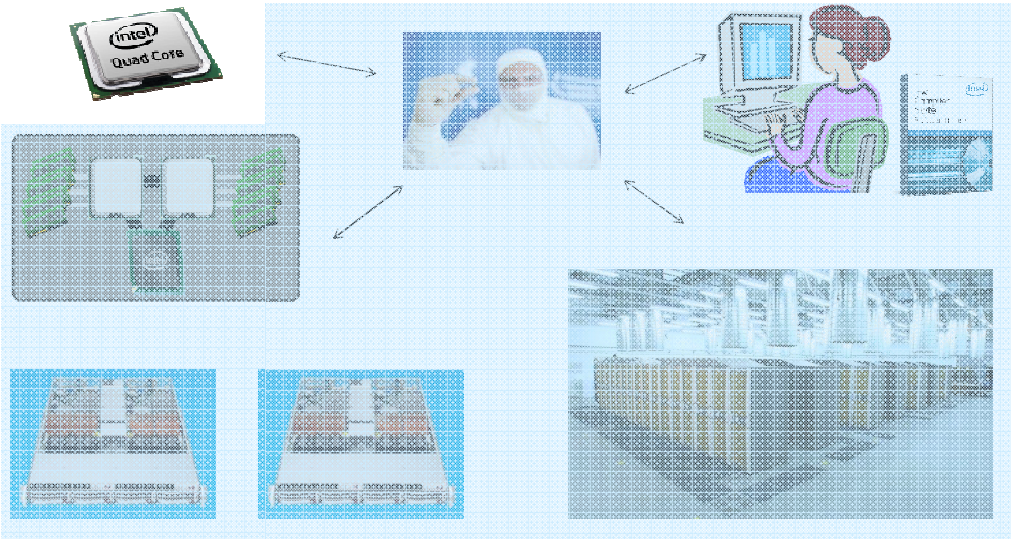

## Agenda: from the micro-arch to the end users





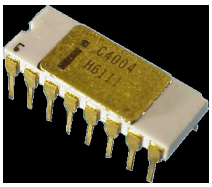


## Agenda: from the micro-arch to the end users



## What G. Moore predicted ?

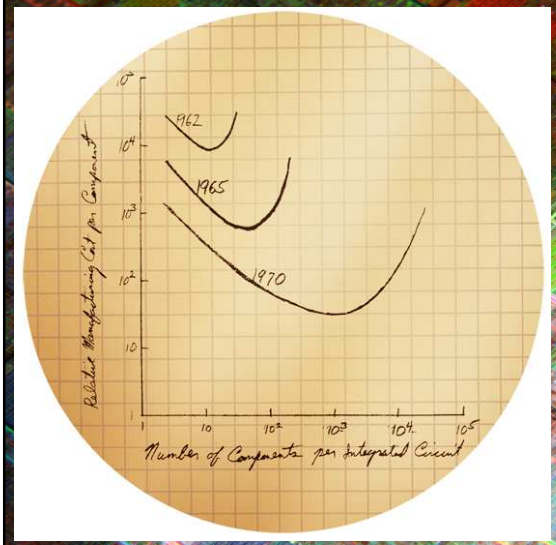
- 1971 Intel® 4004 : 4/8 bits, 108 kHz, 2300 transistors
- 1972 Intel® 8008 : 8 bits, 200 kHz, 3500 transistors
- 1974 Intel® 8080 : 8 bits, 2 Mhz, 4500 transistors
- 1978 Intel® 8086-8088 : 16 bits, 5 MHz,
- 1982 Intel® 80286 : 16/32 bits, plus de 100.000 transistors
- 1985 Intel® 80386 : 32 bits, 275.000 transistors
- 1989 Intel® 80486 : 32 bits, 25 à 66 MHz, 1,6 millions de transistors
- 1993 Intel® Intel® Pentium® : 32 bits, 75 à 133 MHz, 3,1 millions de transistors
- 1995 Intel® Pentium® Pro : 32 bits, 150 à 200 MHz, 5,5 millions de transistors
- 1997 Intel® Intel® Pentium® II : 32 bits, 7,5 millions de transistors
- 1999 Intel® Pentium® III : 32 bits, 450 à 600 MHz, 9,5 millions de transistors
- 1999 Intel® Celeron® : 32 bits, 9,5 millions de transistors
- 2000 Intel® Pentium® 4 : 32 bits, 1,5 GHz, 42 millions de transistors

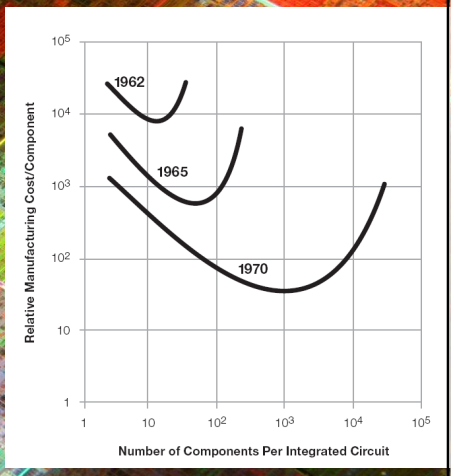







## Not a simple linear function !



"The number of transistors incorporated in a chip will approximately double every 24 months."

Gordon Moore, Intel Co-founder

FOR simple circuits, the cost per component is nearly inversely proportional to the number of components, the result of the equivalent piece of semiconductor in the equivalent package containing more components. But as components are added , decreased yields more than compensate for the increased complexity, tending to raise the cost per component.

Thus, there is a minimum cost at any given time in the evolution of the technology.

…

If we look ahead five years, a plot of costs suggests that the minimum cost per component might be expected in circuits with about 1,000 components per circuit

In 1970, the manufacturing cost per component can be expected to be only a tenth of the present cost.

The complexity for minimum component costs has increased at a rate of roughly a factor of two per year . Certainly over the short term this rate can be expected to continue, if not to increase.
Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years.
That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000.

I believe that such a large circuit can be built on a single wafer.

 *Electronics, Volume 38, Number 8, April 19, 1965*



## Conclusion intermédiaire 1

- La loi de Moore n'a jamais signifiée
  - plus de performance
  - plus de hpc, plus de //ism, plus de scalabilité

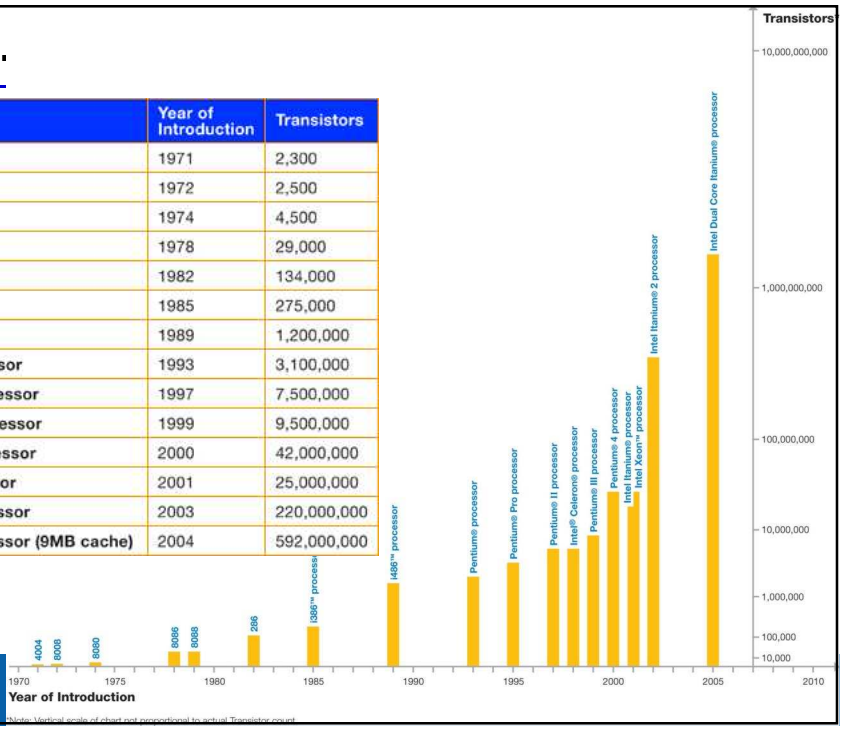
Juste
« il existe un tjs un bon rapport prix / nb de transitor sur un circuit !! »

Implicitement , le nb de transistor augmente et les capacités du dit circuit aussi ;-)

Mais la seule (?) loi qui domine en HPC reste la loi d'Amdhal et assimilée

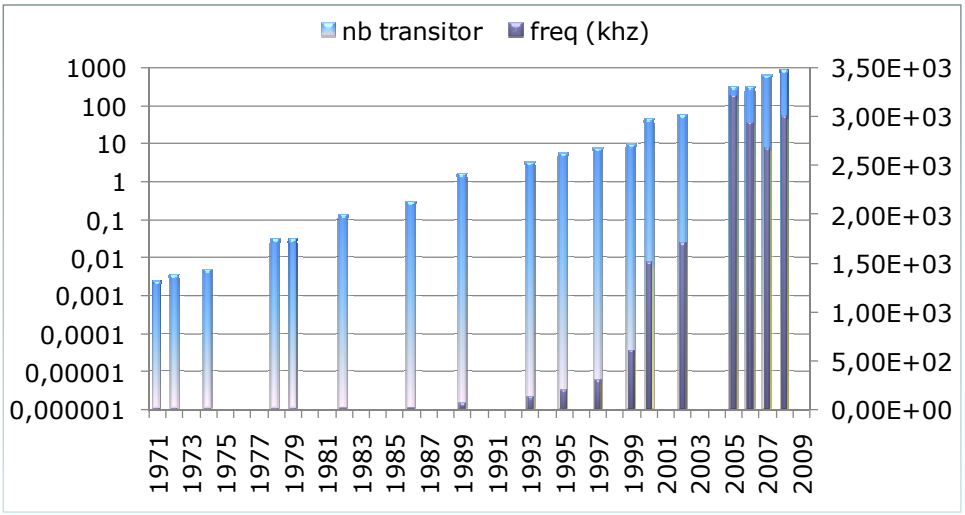
## However .

| Microprocessor | Year of Introduction | Transistors |
|---|---|---|
| 4004 | 1971 | 2,300 |
| 8008 | 1972 | 2,500 |
| 8080 | 1974 | 4,500 |
| 8086 | 1978 | 29,000 |
| Intel286 | 1982 | 134,000 |
| Intel386™ processor | 1985 | 275,000 |
| Intel486™ processor | 1989 | 1,200,000 |
| Intel® Pentium® processor | 1993 | 3,100,000 |
| Intel® Pentium® II processor | 1997 | 7,500,000 |
| Intel® Pentium® III processor | 1999 | 9,500,000 |
| Intel® Pentium® 4 processor | 2000 | 42,000,000 |
| Intel® Itanium® processor | 2001 | 25,000,000 |
| Intel® Itanium® 2 processor | 2003 | 220,000,000 |
| Intel® Itanium® 2 processor (9MB cache) | 2004 | 592,000,000 |



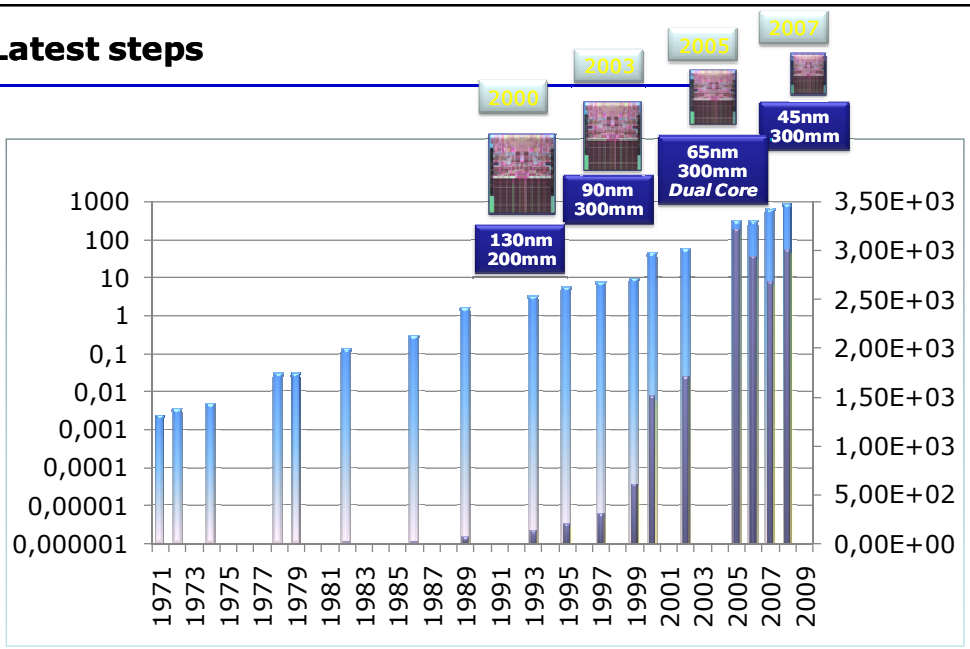

11

## Latest steps




12

## Latest steps

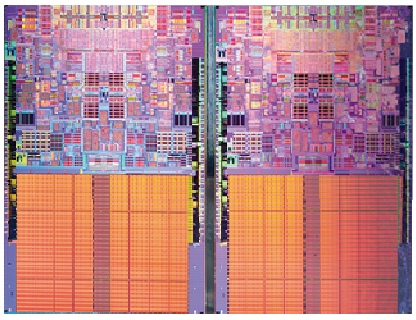







## 45 nm Hi-k Intel Processor

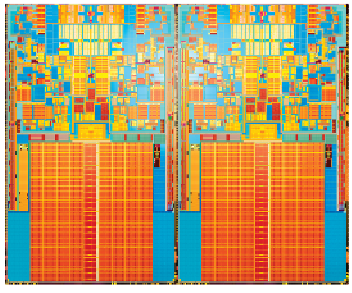
**Quad core Intel® Xeon® 53xx (Clovertown)**
**65 nm**



143 mm$^{2*}$ 143 mm$^{2*}$

582m Transistors
8 MB Cache

**Quad core Intel® Xeon® 54xx (Harpertown)**
**45 nm Hi-k**



107 mm$^{2*}$ 107 mm$^{2*}$

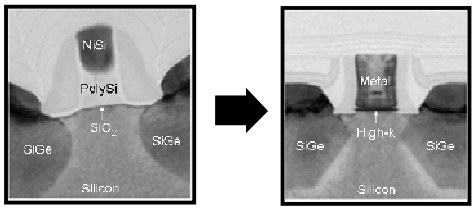
820m Transistors
12 MB Cache



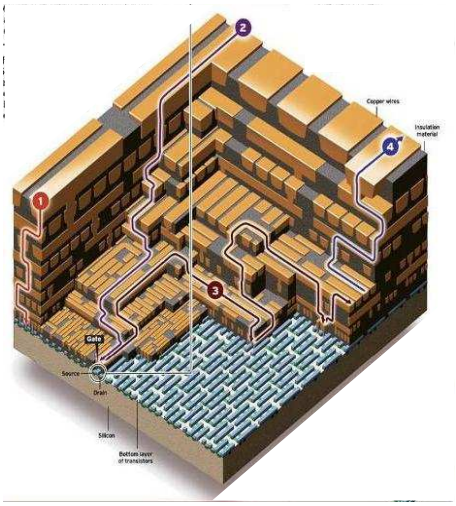
*Source: Intel
Note: die picture sizes are approximate

## Industry's First 45 nm High-K + Metal Gate Transistor

| | |
|---|---|
| Improved Transistor Density | ~2x |
| Improved Transistor Switching Speed | >20% |
| Reduced Transistor Switching Power | ~30% |
| Reduction in gate oxide leakage power | >10x |

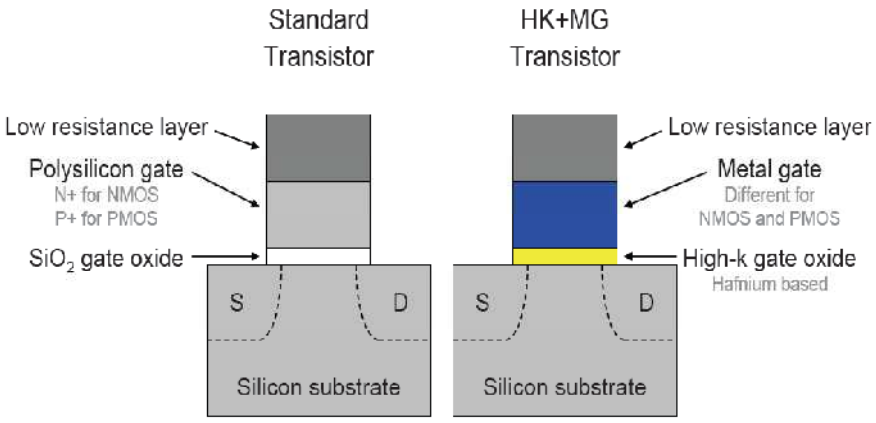



65 nm 45 nm






## High-k + metal gate transistors



## High-k + metal gate transistors

**Metal Gate**

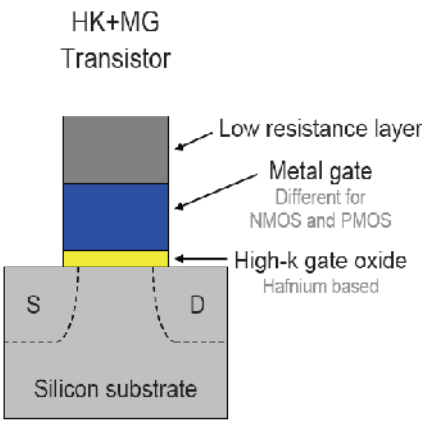
- Increases the gate field effect

**High-k Dielectric**

- Increases the gate field effect
- Allows use of thicker dielectric layer to reduce gate leakage

**HK + MG Combined**

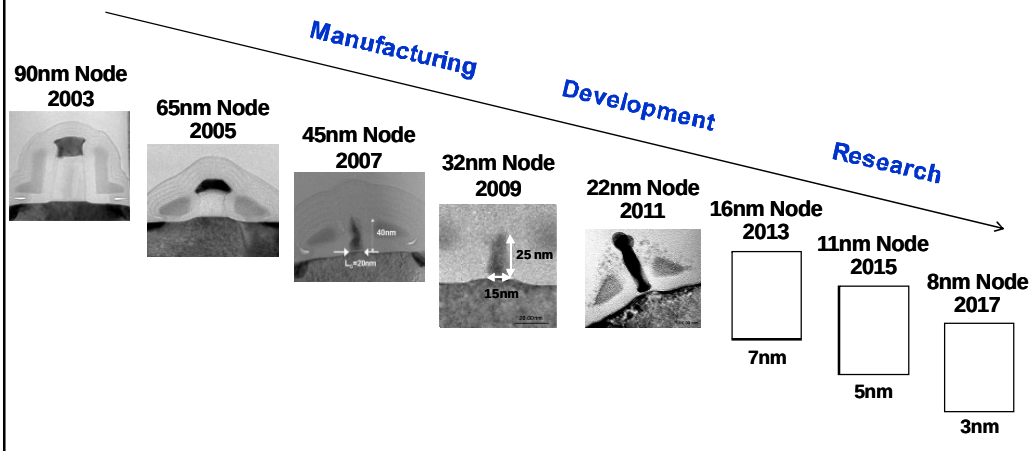
- Drive current increased >20% (>20% higher performance)
- Or source-drain leakage reduced >5x
- Gate oxide leakage reduced >10x



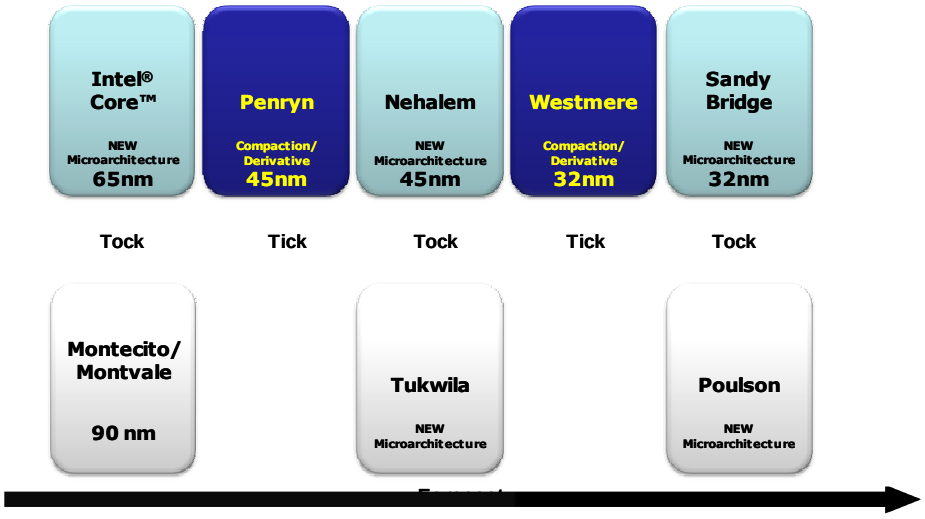

*« The implementation of high-k and metal gate materials marks the biggest change in transistor technology since the introduction of polysilicon gate MOS transistors in the late 1960s », G. Moore, 2007*




## Intel Driving Moore's Law into the Future



# Tick - Tock

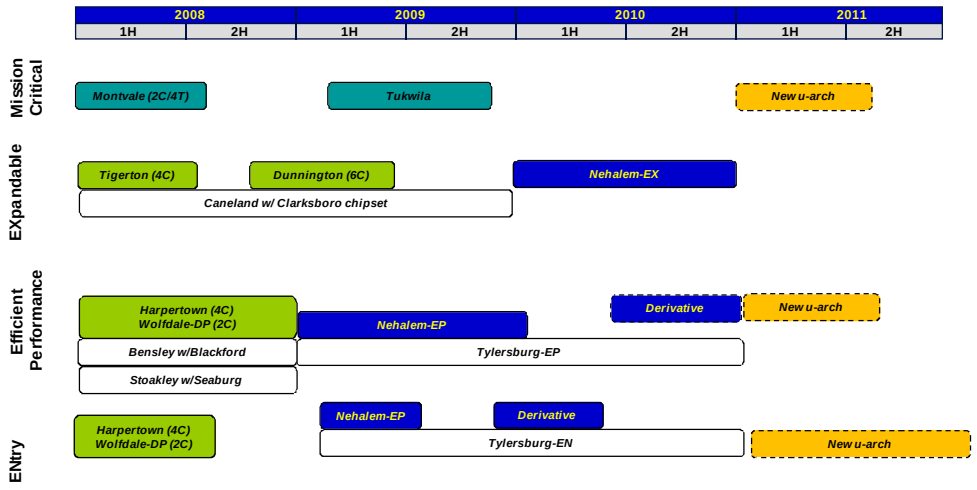






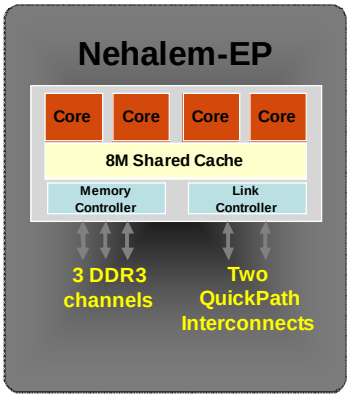



# Intel® HPC Roadmap



## Nehalem-EP

- 4 cores
- 8M on-chip Shared Cache
- **3-level cache hierarchy**
  - **32k I-Cache + 32k D-cache**
  - **New 256k L2 cache *per core***
  - **New shared last level cache**
  - **Inclusive Cache Policy**
- Simultaneous Multi-Threading capability (SMT)
- QuickPath interconnect
  - **Point-to-Point**
  - **2 links per CPU socket**
  - **1 for connection to other socket**
  - **1 for connection to chipset**
  - **Up to 6.4 GT/sec (12.8 GB/sec) in each direction per link ( Fully duplex)**
- Integrated QuickPath Memory Controller (DDR3)
  - Up to 18 DIMMs
  - 800/1066/1333MHz DDR3
- New instructions
- Power: 130W, 95W, 80W, 60W




**Socket:**

- New LGA 1366 Socket

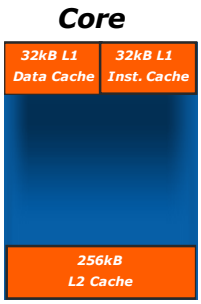



## Nehalem Processor Microarchitecture

- Derived from Core Microarchitecture
  - Similar in pipeline length
- Processor Performance features
  - Greater parallelism
    - » Example: 33% more instructions in flight
  - More efficient algorithms
    - » Example: Faster handling of unaligned cache accesses
  - Better branch prediction
- New 3-level cache hierarchy
- New 2-level TLB hierarchy

## Intel Smart Cache – Core Caches

- 1st level caches
  - 32kB Instruction cache
  - 32kB Data Cache
    - Support more L1 misses in parallel than Core 2
- 2nd level Cache
  - New cache introduced in Nehalem
  - Unified (holds code and data)
  - 256 kB per core
  - ***Performance:*** Very low latency
  - ***Scalability:*** As core count increases, reduce pressure on shared cache

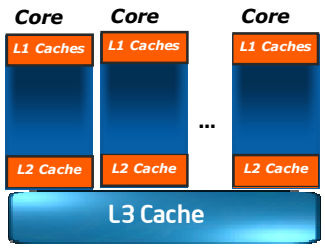






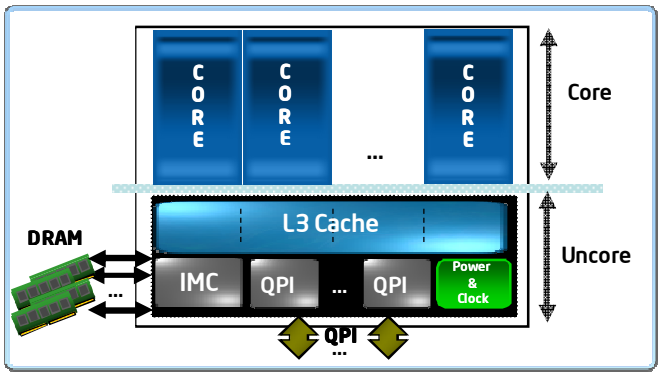
## Intel Smart Cache -- 3rd Level Cache

- New 3rd level cache
- Shared across all cores
- Size depends on # of cores
  - Quad-core: Up to 8MB
  - ***Scalability:***
    - Built to vary size with varied core counts
    - Built to easily increase L3 size in future parts
- Inclusive cache policy for best ***performance***
  - Address residing in L1/L2 ***must*** be present in 3rd level cache



## Designed For Modularity

2008 – 2009 Servers & Desktops

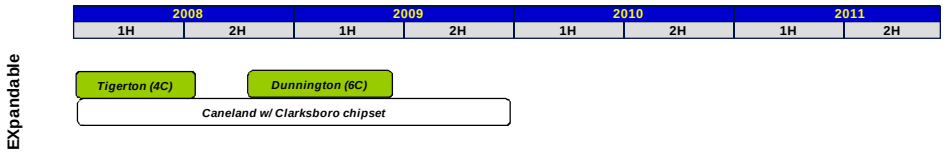


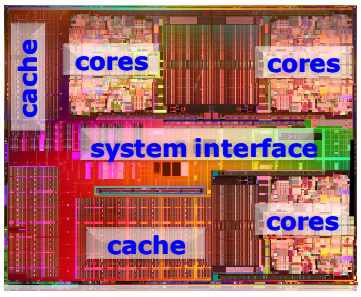

Differentiation in the "Uncore":






## Expandable (EX) Platform Roadmap




- 45nm high-k technology
- 1.9B transistors
- 16 MB L3 cache
- Caneland socket compatible
- Latest Intel virtualization technologies
- Launched

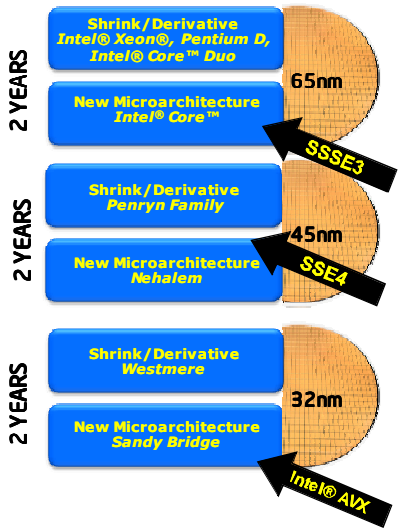
## Intel innovations

**Evolution of Instruction Set Architecture (ISA) :**

- 1979 - x87 Numeric data extensions
- 1997 - MMX™ - 64b SIMD ISA extension
- 1999-2007 – SSEx - 128b SIMD ISA extensions
- Starting 2010 – Intel® AVX – 256b SIMD ISA extensions

**Benefits of Intel's ISA**

- Available across a wide range of platforms
  - No additional hw required
  - Investment protection for your development costs and effort
- Backward and forward compatible to future ISAs
- A suite of tools and compilers to make development easy





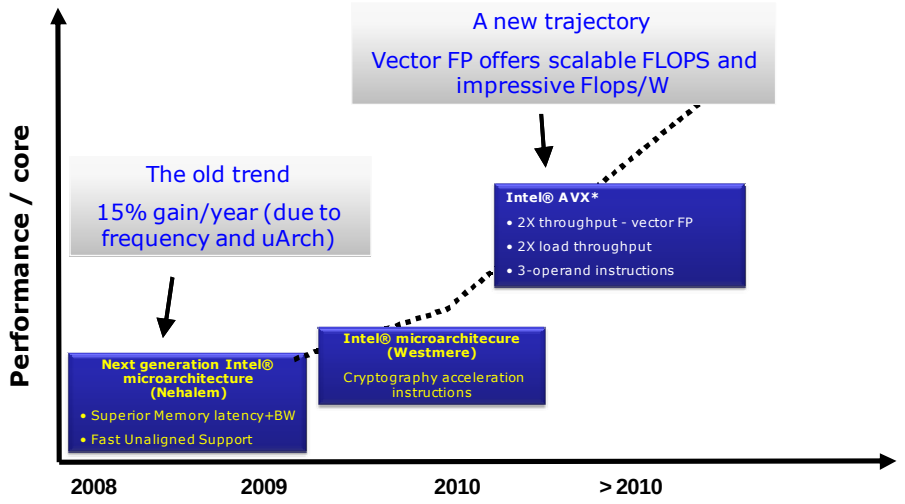


## Key Intel® Advanced Vector Extensions : Intel® AVX

| *KEY FEATURES* | *BENEFITS* |
|---|---|
| Wider Vectors<br>Increased from 128 bits to 256 bits | Up to 2x peak Flops/s<br>with good power efficiency |
| Enhanced Data Rearrangement<br>Use new 256 bit primitives to broadcast, mask loads and permute | Organize, access and pull only necessary data more quickly and efficiently |
| Three and four Operands,<br>Non Destructive Syntax<br>Designed for efficiency and extensibility | Fewer register copies, better register use |
| Flexible unaligned memory access support | More opportunities to combine load and compute operations |

Intel® AVX is a general purpose architecture, expected to supplant SSE in all applications used today

## Intel Instruction Set Innovation Continues



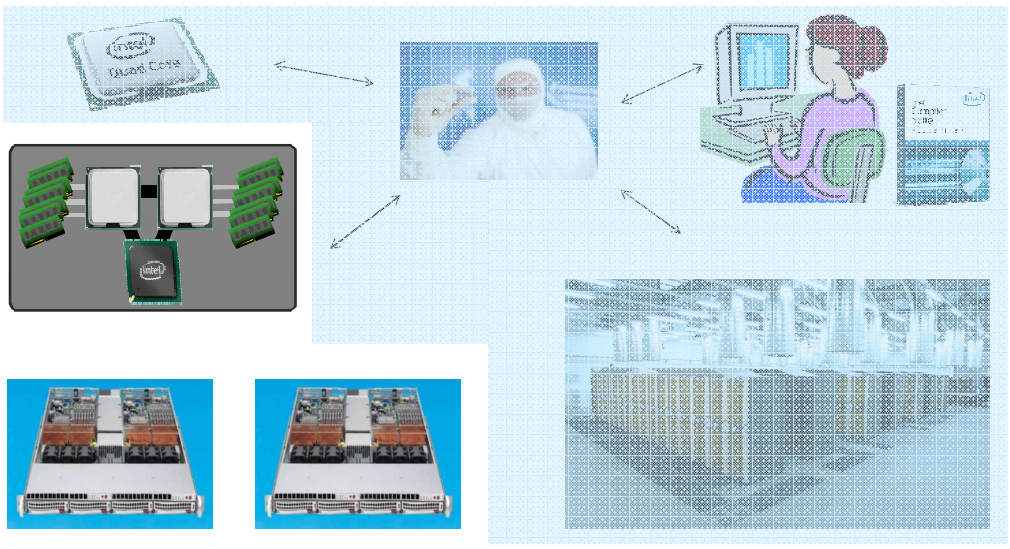

*All timeframes, dates and products are subject to change without further notification*

* Intel® AVX = Intel® Advanced Vector Extensions (Intel® AVX)

29



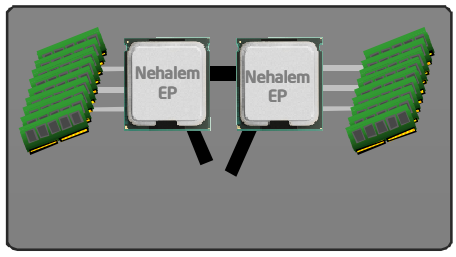
## Agenda: from the micro-arch to the end users

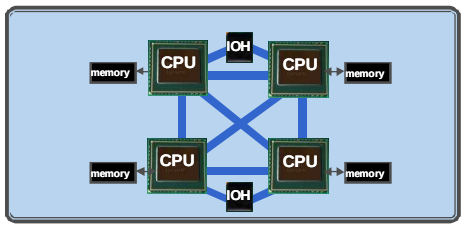


30

## QuickPath Interconnect

- Nehalem introduces new QuickPath Interconnect (QPI)
- ***High bandwidth***, ***low latency*** point to point interconnect
- Up to 6.4 GT/sec initially
  - 6.4 GT/sec -> 12.8 GB/sec
  - Bi-directional link -> 25.6 GB/sec per link
- Highly ***scalable*** for systems with varying # of sockets



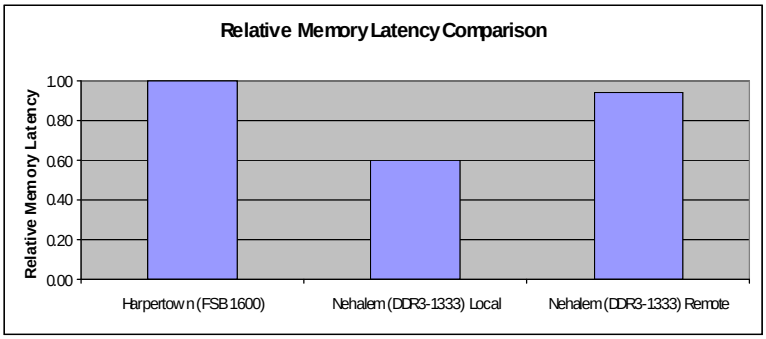







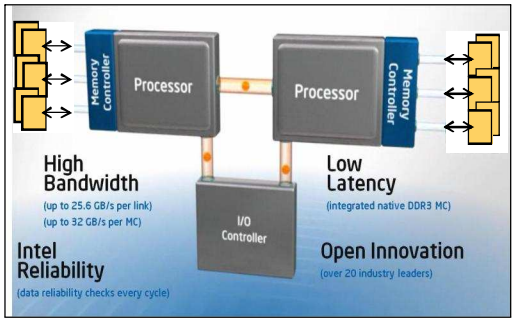
## Memory Latency Comparison

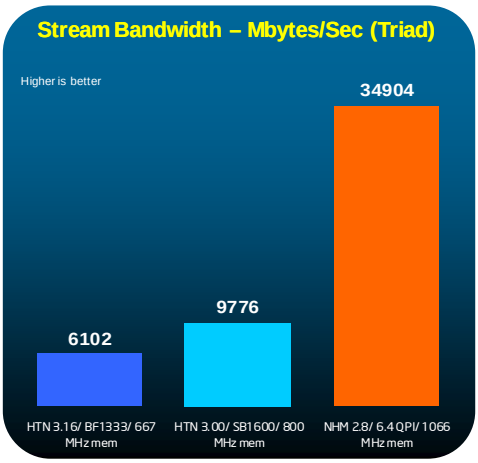
- ***Low memory latency*** critical to high performance
- Design integrated memory controller for low latency
- Need to optimize both local and remote memory latency
- Effective memory latency depends per application/OS
  - Percentage of local vs. remote accesses
  - NHM has lower latency regardless of mix



# Intel QuickPath Architecture

***Intel® QuickPath interconnect plus integrated Intel® QuickPath memory controllers***



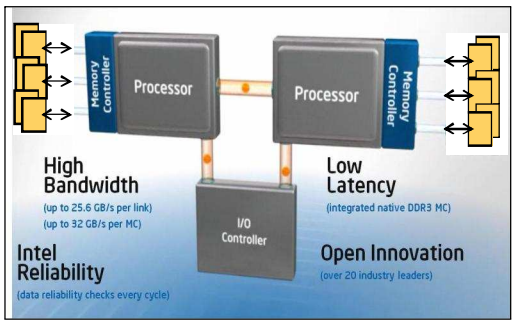

Source: Intel. All future products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

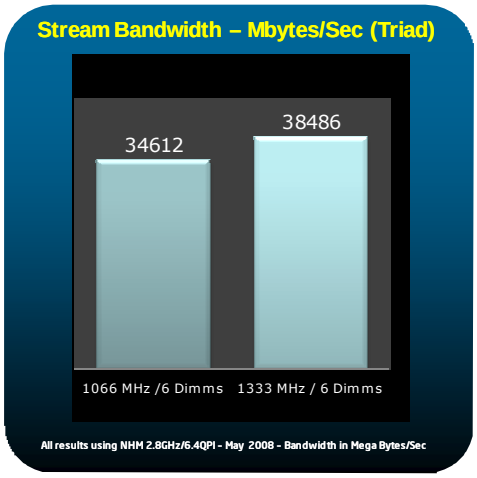




# Intel QuickPath Architecture

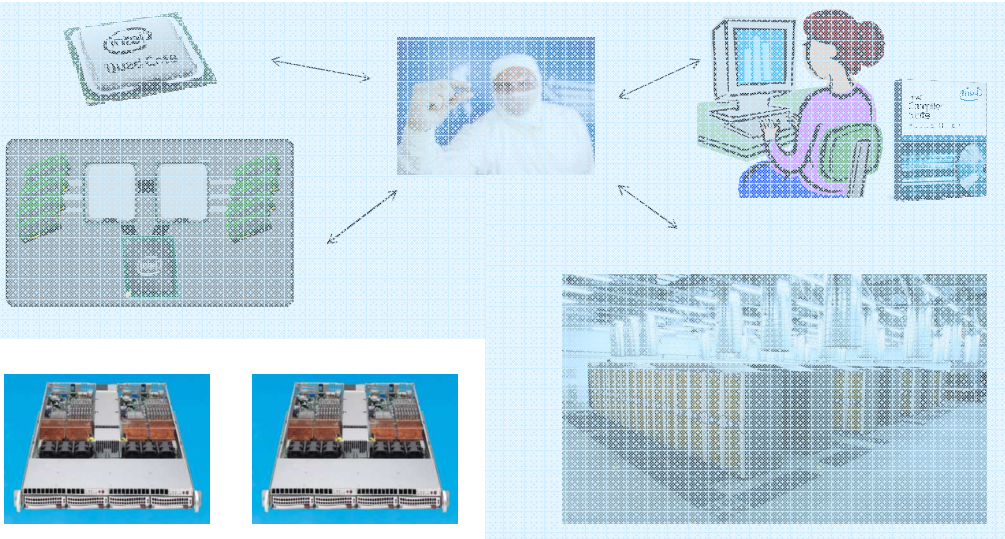
***Intel® QuickPath interconnect plus integrated Intel® QuickPath memory controllers***



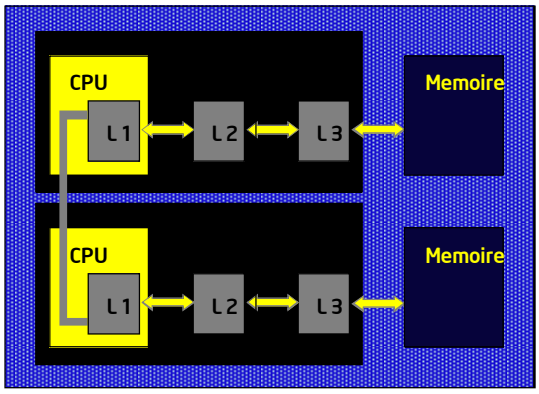

Source: Intel. All future products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

## Agenda: from the micro-arch to the end users






## Where are the limitations ?




I / O

Interconnect.

Intra and Inter socket communications

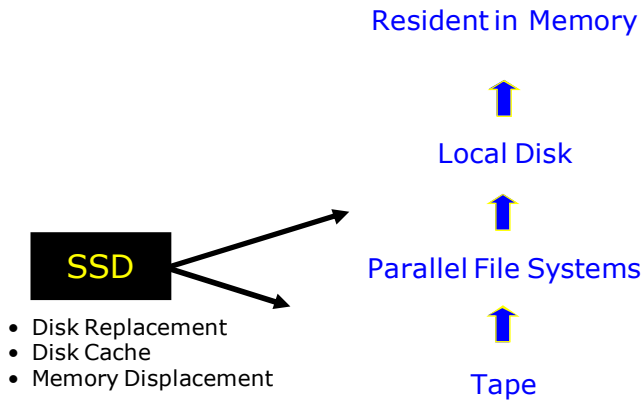
Number, size and performance of the cores ?

Size, BW of Cache memories ?

Memory BW ?

Inter nodes Communications ?

## I/O Trends

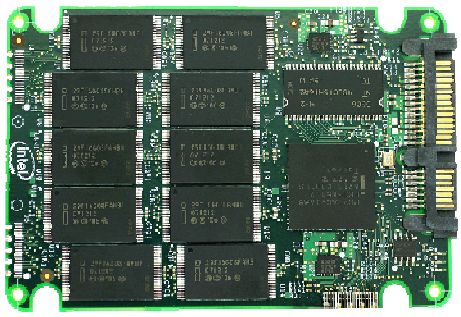



TeraScale Memory Research





## Intel® High Performance: *SATA Solid State Drives - The new trend*



**SATA 3.0 Gb/s Interface**

**1.8" & 2.5" Form Factors**

**32GB to 80GB+ Capacities**

**Power**

| | |
|---|---|
| **Active (avg)** | **2.0W** |
| **Standby** | **0.1W** |

**Weight:**

**2.5" SSD 87g (+/- 2g)**

**1.8" SSD 40g (+/- 2g)**

[1]IOmeter results vs Seagate Momentus® 7200.2 SATA 3Gb/s Hard Drive
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.

# Intel® High Performance Solid-State Drives

"X25-E"
*Enterprise Class*

***High performance SATA Drive***

- 2.5" FF
- 32GB / 64GB SLC
- 240 / 170 R/W (MB/s)
- Architected for write intensive highly random workloads

SLC – Enterprise Class SSD

"X18-M"
*SFF Mobile Client*

***Performance SFF Mobile SATA Drive***

- 1.8" FF
- 80GB / 160GB MLC
- 240 / 70 R/W (MB/s)

"X25-M"
*Mobile Client*

***Performance Mobile SATA Drive***

- 2.5" FF
- 80GB / 160GB MLC
- 240 / 70 R/W (MB/s)

MLC – Client/Consumer Class SSD

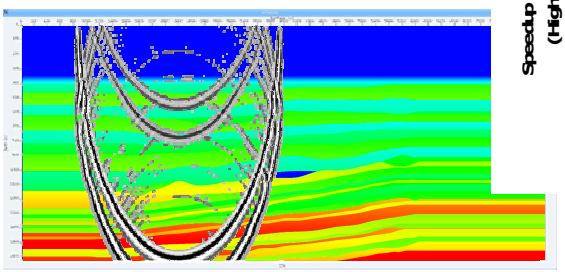
39 *SLC & MLC (Multi-Level Cell) NAND*

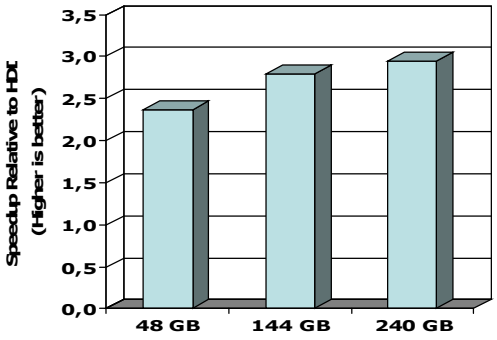


# Experiment - Finite Difference Code

*(Simulate RTM Snapshots)*

- 2D elastic wave propagation
- Small 2-homogeneous layered model
- Disk Replacement Test
  - HDD – 4x150GB SATA RAID-0
  - SSD – 4x80GB MLC Samples RAID-0



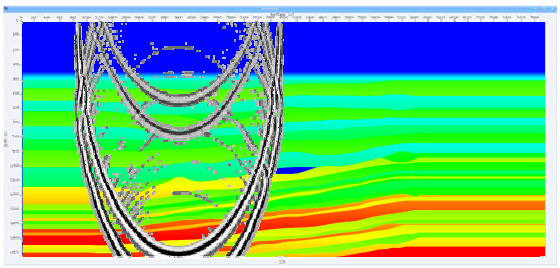


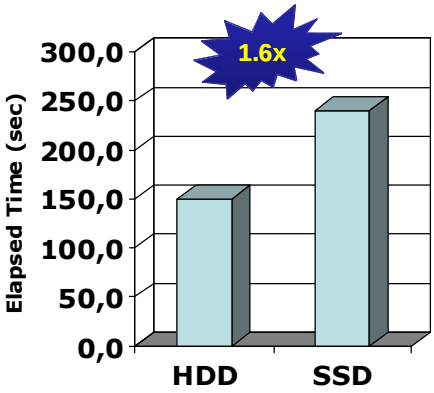

**Potential: 2-3x Gain with this Example**

40

## Experiment - Finite Difference Code

- 2D elastic wave propagation
- Small 2-homogeneous layered model
- Disk Replacement Test
  - HDD – 2x250GB SATA RAID-0
  - SSD – 2x32GB SLC Samples RAID-0

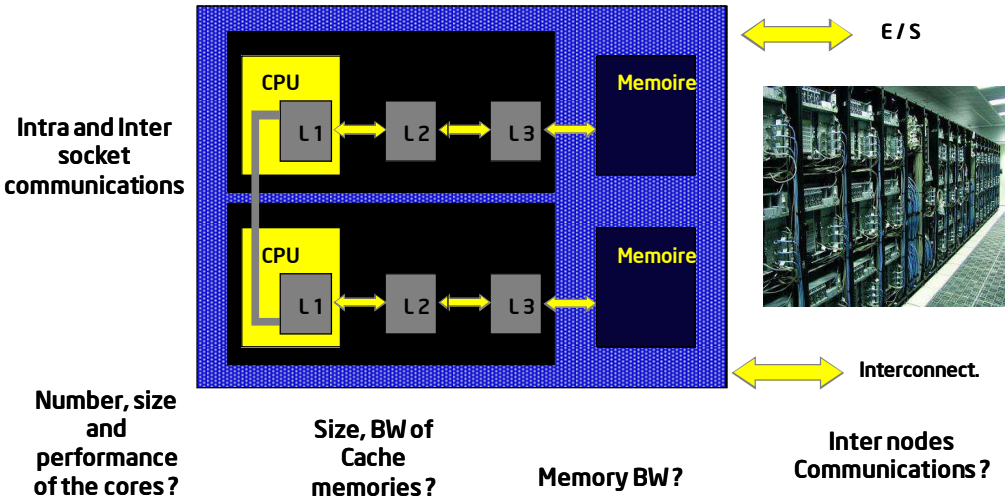




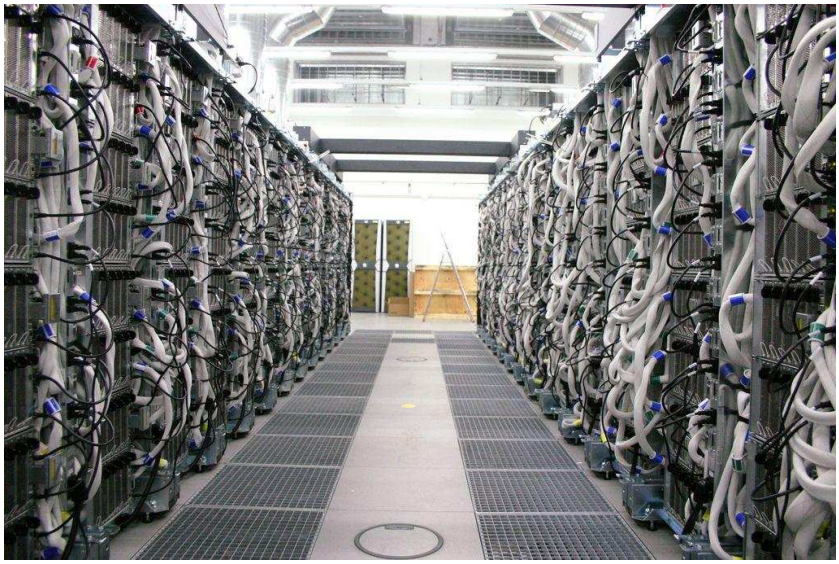

**1 shot gather : 12GB ouput**




## Where are the limitations ?



## Interconnect ?

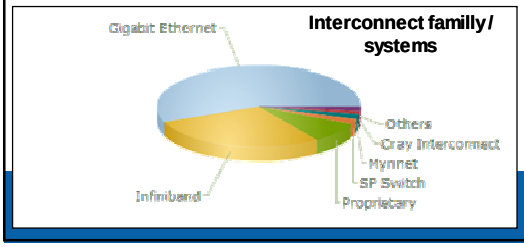


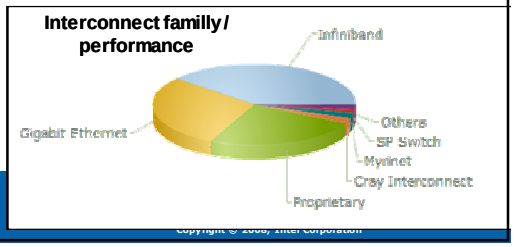




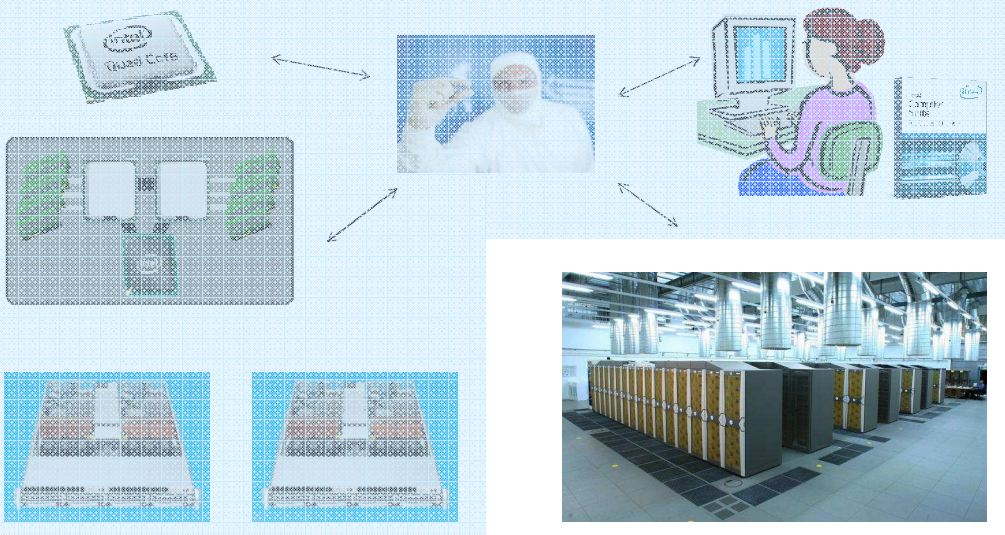
## 11/2008 Top500 statistics

| Interconnect Family | Count | Share % | Rmax Sum (GF) | Rpeak Sum (GF) | Processor Sum |
|---|---|---|---|---|---|
| Myrinet | 10 | 2.00 % | 350290 | 488934 | 56576 |
| Quadrics | 4 | 0.80 % | 122220 | 147507 | 21040 |
| Gigabit Ethernet | 282 | 56.40 % | 4948233 | 9795163 | 941748 |
| Infiniband | 141 | 28.20 % | 6549813 | 8721697 | 841730 |
| Crossbar | 1 | 0.20 % | 35860 | 40960 | 5120 |
| Mixed | 1 | 0.20 % | 66567 | 82944 | 13824 |
| NUMAlink | 3 | 0.60 % | 122554 | 137625 | 21504 |
| SP Switch | 10 | 2.00 % | 229541 | 273754 | 34208 |
| Proprietary | 42 | 8.40 % | 4143049 | 5243830 | 1108169 |
| Cray Interconnect | 6 | 1.20 % | 359197 | 469470 | 73004 |
| **Totals** | **500** | **100%** | **16927325.79** | **25401883.80** | **3116923** |



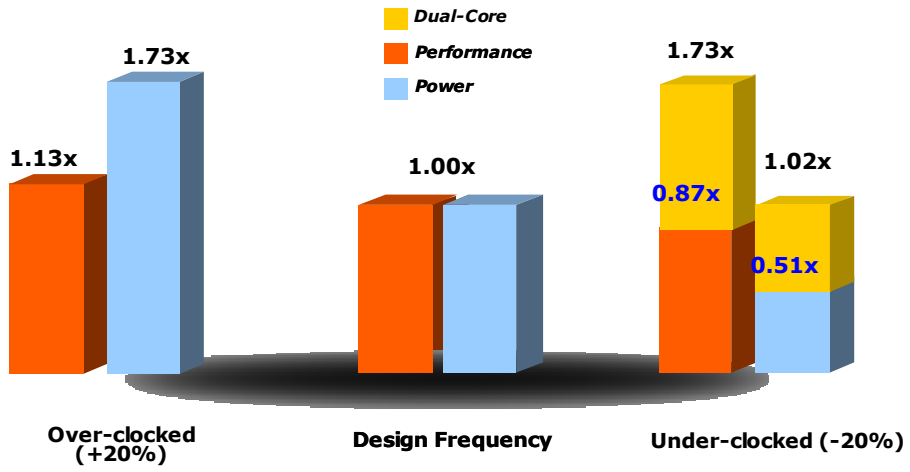
## Agenda: from the micro-arch to the end users



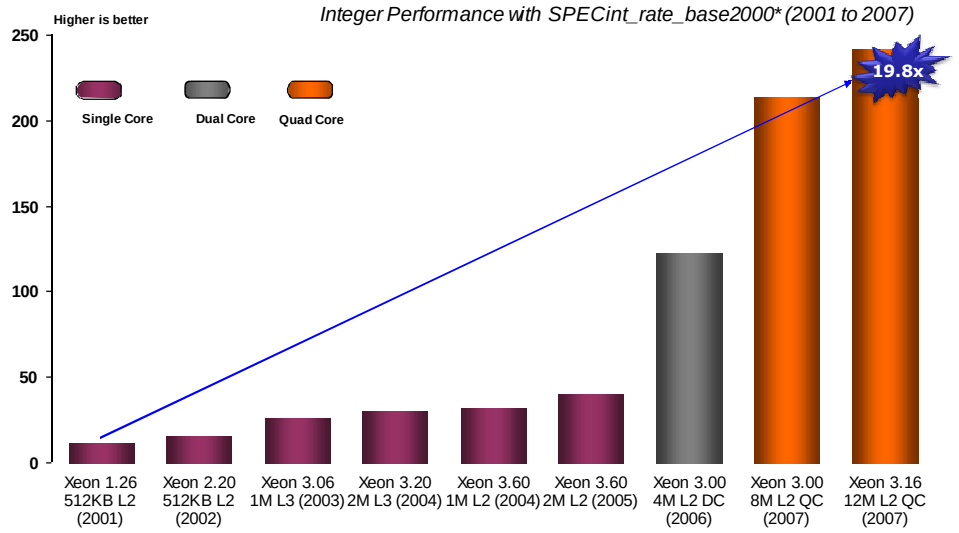




## Multi-core motivation



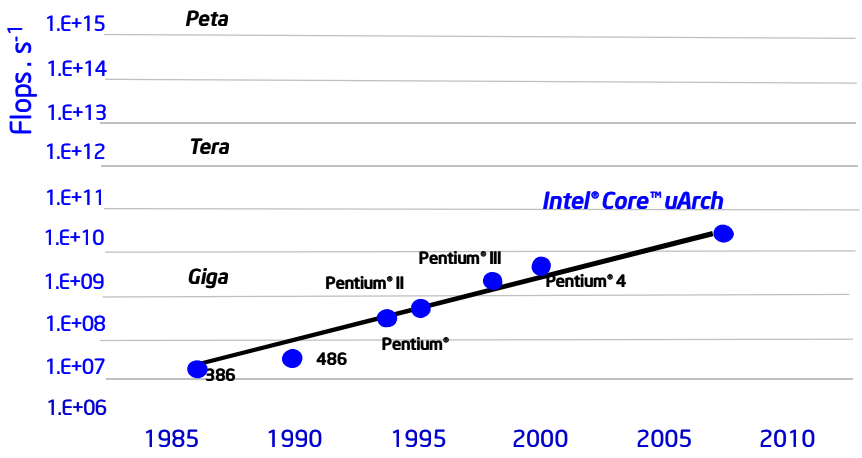
## Multi-core advantage





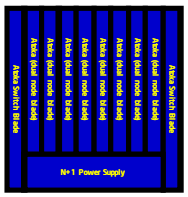


## Petaflops/s with today CPU ?

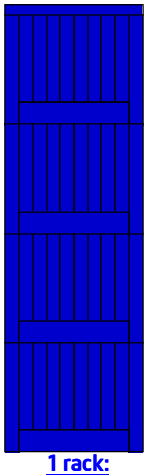


## Global warming of the datacenter

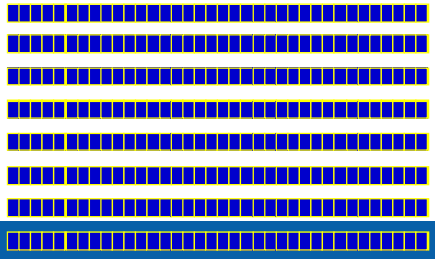
1U:
170 GF
2x 450 W




1 module :
1.36 TF
~5.6 kW



1 rack:
5.44 TF
~22.4 kW



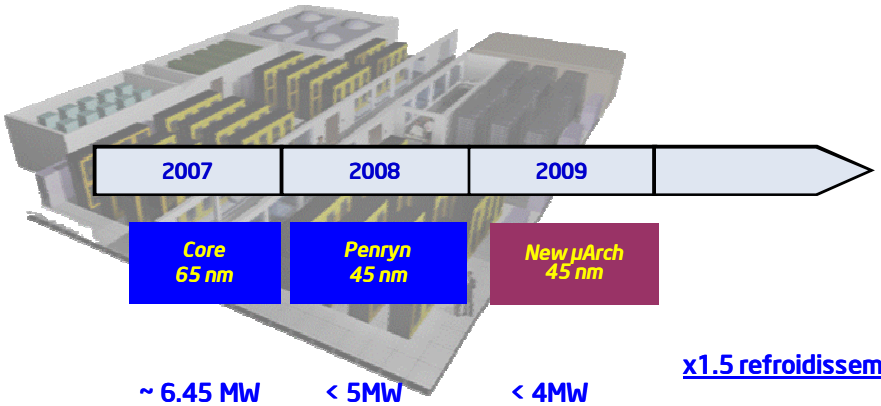
288 racks:
1.6 PF
147 k cores
6.45 MW





## Petaflops : decrease of the power consumption ?

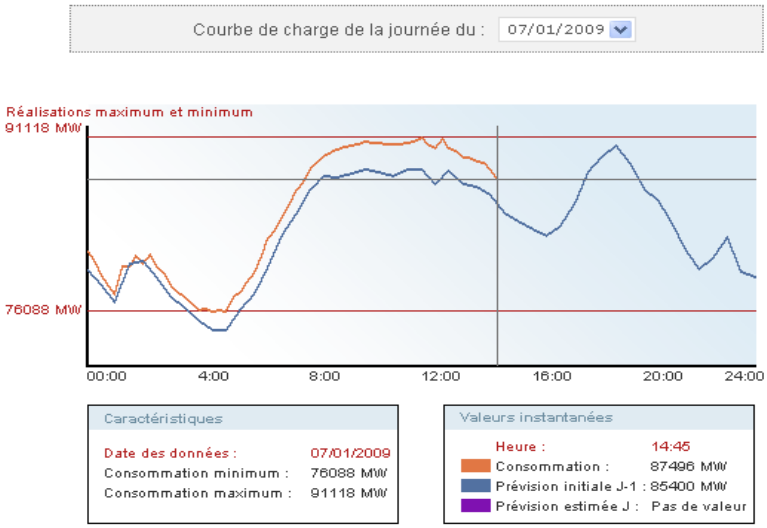
10 MW à 0.05 € / kWh
> 5 M€ / an (500 € / hour )




~ 6.45 MW
< 5MW
< 4MW

x1.5 refroidissement

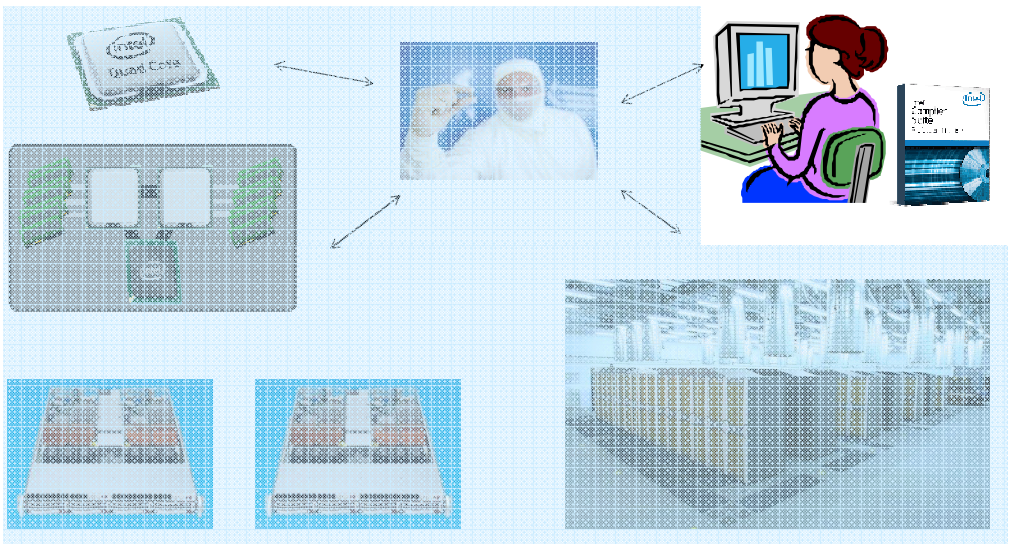
## French power consumption




51 http://www.rte-france.com/



## Agenda: from the micro-arch to the end users



52

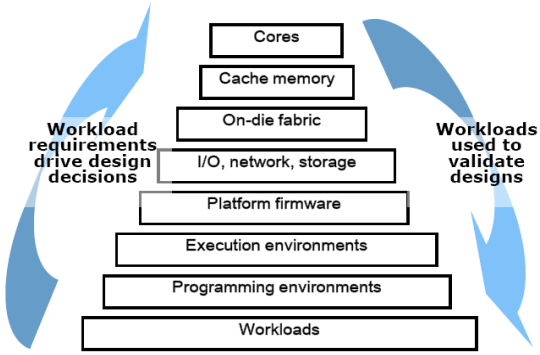
## HPC enabling@ Softwares & Services Group (SSG)

- tuning / optimization
- training
- benchmarking
- architecture (roadmap)
- topology (interconnect, i/o)
- softwares
- early cpu testing
- collaboration
- co-design

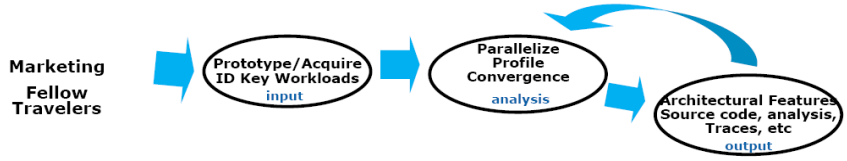
- HPC end user:
  - develop their own codes
  - uses ISV codes
  - benchmarking
- ISV
- Universities / Research Institutes
- OEMs




## Intel point of view : Emerging application research




- Identify
- Characterize
- Optimize
- Enhance HW
- Validate



## End users point of view : Physics and computing needs increase

Increase problem size

(at constant time and physics)

Decrease elapsed time.

(at constant size and physics)

Increase physics complexity

(at constant elapsed time and size )





## HPC or Scientific Computing landscape

Universities / Research Institutes

Wks

Labs

Meso center

National center

ISV

< 500 Gflops

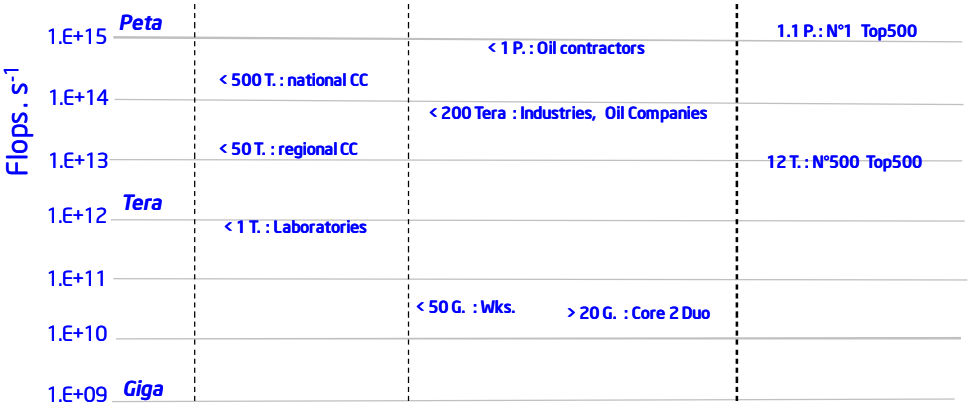
R & D in the Industry

< 10 Tflops

Production in the Industry

20 Tf to < 1 Pflops

## Availaible Computing power … in 2008



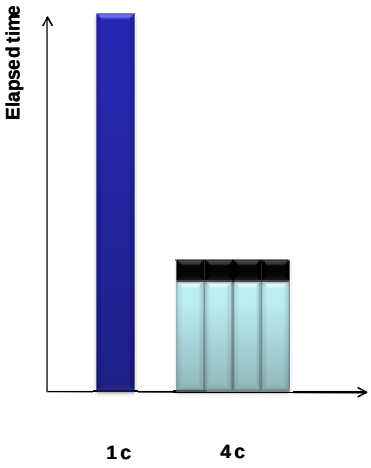


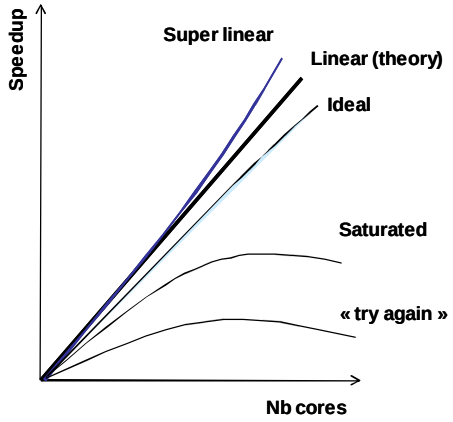


## The Picket Fence (Amdahl revisited)

| # People | 1 | 10 | 100 |
|---|---|---|---|
| Prepare | 1 | 1.1 | 1.2 |
| Paint | 10 | 1 | .1 |
| Clean-up | 1 | 1.1 | 1.2 |
| Total | 12 hours | 3.2 hours | 2.5 hours |
| Speed-up | 1 | 3.75 | 4.8 |
| Efficiency | 100% | 37.5% | 4.8% |
| Brush | 1 € | 10 € | 100 € |
| Beer cooling | 0.12 € | 0.32 € | 2.5 € |

## Parallel computing

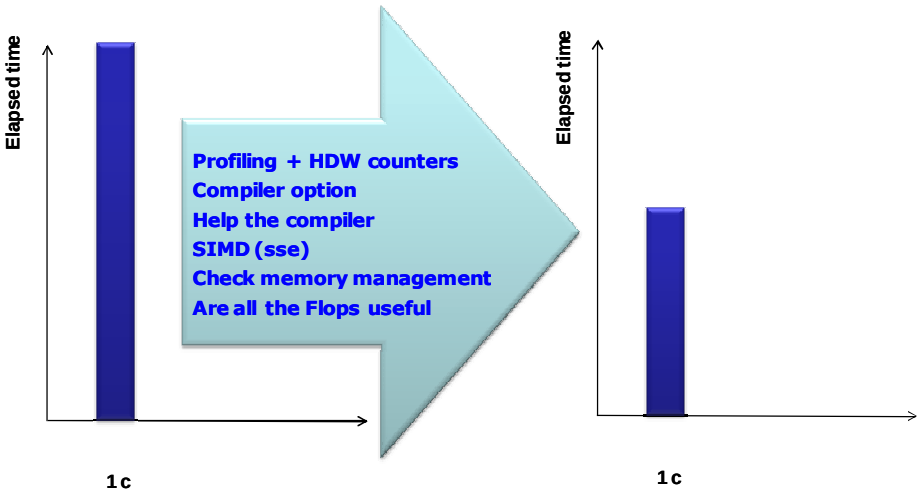


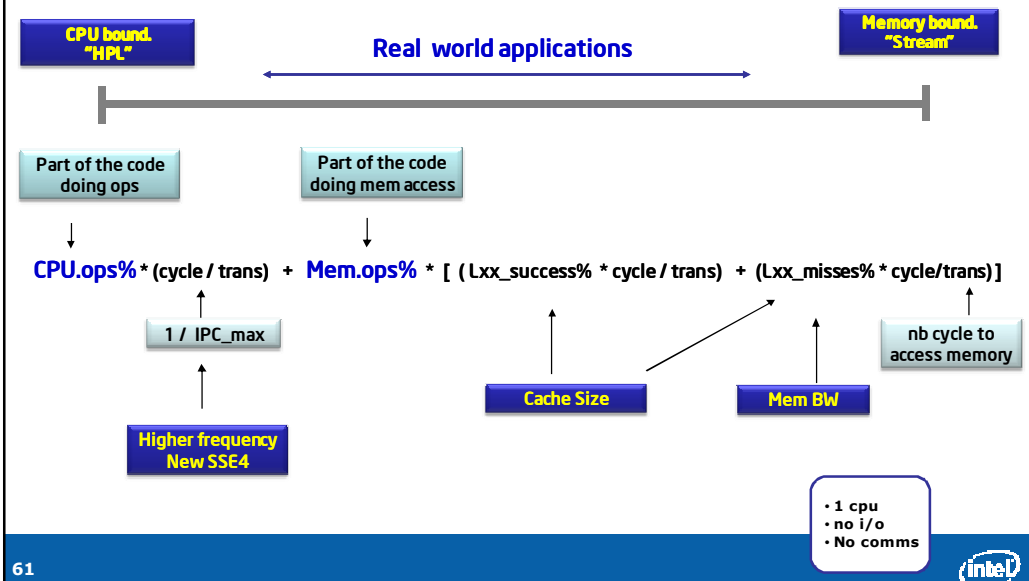




## First step : do it serial



## Apps classification and performance analysis







## Intel® Software Development Products

- Intel® Compilers
  - The best way to get application performance on Intel® multi-core processors – both for C/C++ and Fortran
- Intel® VTune™ Performance Analyzers
  - Identify bottlenecks i and optimize multi-core performance
- Intel® Performance Libraries
  - Highly optimized, thread-safe, multimedia and HPC math functions

## C++ and Fortran Compilers 11.0

| OS | IA-32 | IA-64 | Intel® 64 |
|---|---|---|---|
| Windows* | √ | √ | √ |
| Linux* | √ | √ | √ |
| Mac OS* X | √ | | √ |

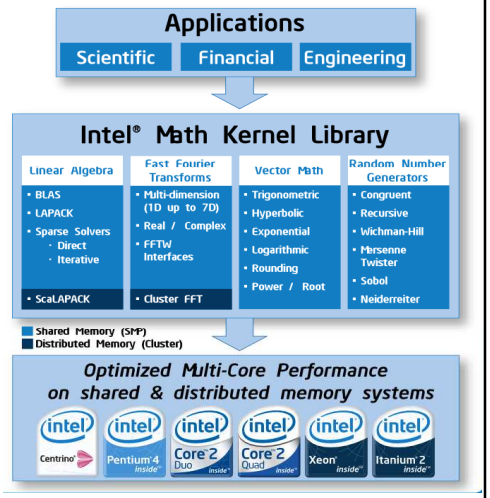
- OpenMP 3.0 support
- C++ lambda functions
- Windows* products: Microsoft Visual Studio* 2005 and 2008 support
- Linux* products: Eclipse* CDT 4.0 support
- Windows Server* 2008 Support
- Parallel debugger for IA-32/Intel®64 Linux
- Option to emit FE diagnostics related to threading
- /MP option for generating compilation sessions in parallel
- Implementation of valarray using IPP for array notation
- Improved install framework based on Intel® PSET 2.0
- Default to be Intel® Pentium® 4 (QxW) compatible
- Increased F2003 feature support beyond 10.1:
  - ENUMERATOR, MIN/MAX and friends, IEEE FP, ASSOCIATE, PROCEDURE POINTER, ABSTRACT INTERFACE, TYPE EXTENDS, struct ctor change, array ctor changes, ALLOCATE scalar
- Native Intel®64 Compiler (Win/Linux)
- Compilers available only in "Pro" packaging (no more "Standard")




## Intel® Math Kernel Library 10.0

The ***flagship*** math processing library

- Simplify multi-threaded application development
  - Extensively threaded math functions with excellent scaling
  - ***New*** Threading in Vector Math Functions
  - ***New*** OpenMP compatibility library supports Microsoft and GNU OpenMP implementations
- Maximize application performance
  - Automatic runtime processor detection ensures great performance on whatever processor your application is running on.
  - ***New*** optimizations for latest Intel processors
  - ***New*** cluster functionality is now standard




| Windows* | Linux* | Mac* | IA32 | Intel64 | IA64 | Multicore |
|---|---|---|---|---|---|---|
| √ | √ | | √ | √ | √ | √ |

## MKL 10.0

### Intel® Math Kernel Library 10.0 – Sparse Solvers

Product Information | Evaluation Center | Support Resources | Documentation

Overview | BLAS/LAPACK | ScaLAPACK | Sparse Solvers | Fast Fourier Transforms | Vector Math | Vector Statistics | LINPACK

The Intel Math Kernel Library includes sparse solvers that use both direct and indirect/iterative methods.

| Matrix Types | | Intel® Math Kernel Library Sparse Solvers | |
|---|---|---|---|
| | | Direct | Indirect/Iterative |
| General | | PARDISO (d, z) (Parallel Direct Solver) | FGMRES (d) |
| Symmetric | Positive | PARDISO (d, z) (Parallel Direct Solver) | Conjugate Gradient (d) |
| | Indefinite | PARDISO (d, z) (Parallel Direct Solver) | |

d: Supports double-precision data
z: Supports double-precision, complex data




## Math Functions Using SVML Library

Short vector math library (SVML) provides efficient software implementations:

- sin/cos/tan
- asin/acos/atan
- sinh/cosh/tanh
- asinh/acosh/atanh
- log10/ln
- exp/pow

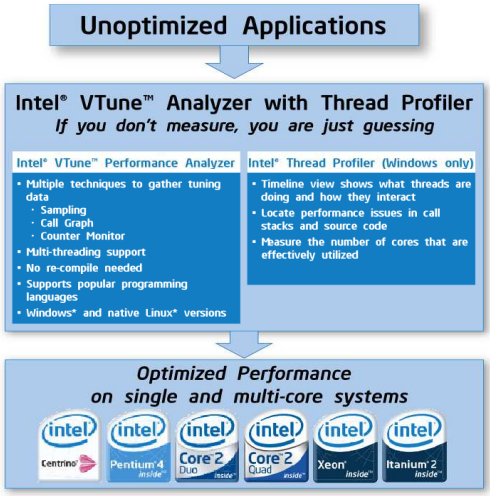
- and many more

# Intel® VTune™ Analyzer 9.0

***Quickly*** find performance bottlenecks

- Simplify multi-threaded application development
  - Tune process or thread parallel code
  - Works on standard debug builds without recompiling
- Maximize application performance
  - Stall cycle accounting for Core™2 Duo and Core™2 Quad processors
  - Low overhead sampling
  - Graphical call graph
  - View results on source or assembly
- Includes Intel® Thread Profiler

| Windows* | Linux* | Mac* | IA32 | Intel64 | IA64 | Multicore |
|---|---|---|---|---|---|---|
| √ | √ | | √ | √ | √ | √ |







# Whatif.intel.com

**Opportunity to kick the tires, give feedback, experiment – put you "in the loop early" – before we fold into final products.**

**Also - have *direct* connection with the developers and researchers in Intel.**


**Intel® Adaptive Spike-Based Solver**
Cluster OpenMP* for Intel® Compilers
Intel® Platform Modeling with Machine Learning
Intel® Decimal Floating-Point Math Library
Intel® Location Technologies Software Development Kit 1.0 (LTSDK)
**Intel® C++ Parallelism Exploration Compiler, Prototype Edition**
Integrated Debugger for Java*/JNI Environments
**Intel® Performance Tuning Utility 3.0**

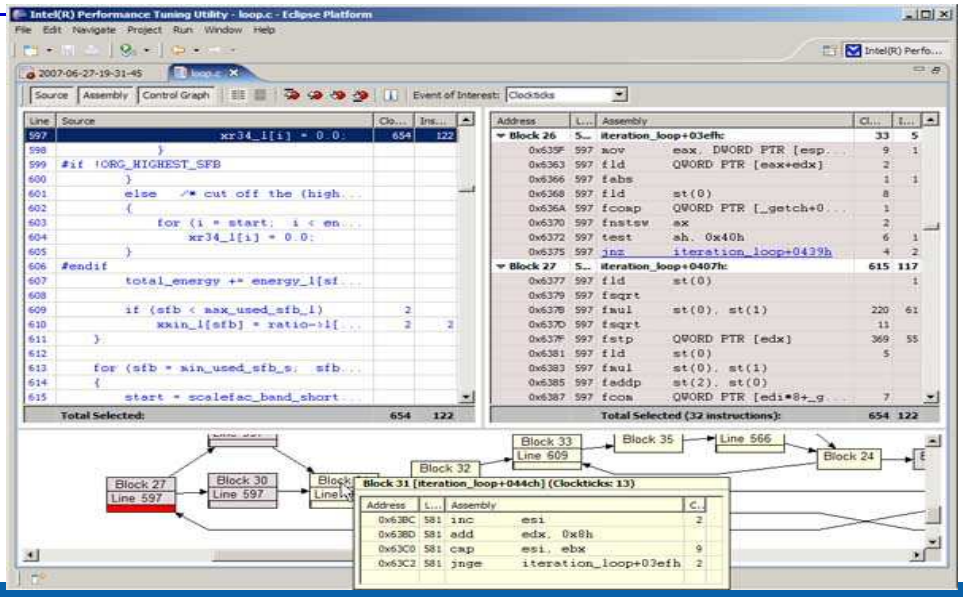
## Intel® Performance Tuning Utility 3.0

- **NEW in v3.0**
  - **Result difference Extension**
    - Semi-automated compiler comparison
  - **Events per CPU**
  - **Data Access Profiling updates**
  - **Module Aliases**
  - **CPU Frequency Changing**
  - **Heap Profiling GUI**
  - **Instrumentation-based Call Graph and Call Count**
- **EXISTING from previous version**
  - **Statistical Call Graph**
  - **Basic Block Analysis**
  - **Events over IP graph**
  - **Loop Analysis**

*Learn More at: http://whatif.intel.com*




## Intel® Performance Tuning Utility 3.0



## Agenda (for next time)

- **Compiler**
  - **Reports**
  - **Vectorization**
  - **Inlining**
  - **PGO**
  - **Blocking**
  - **Unrolling**
  - **Floating points**



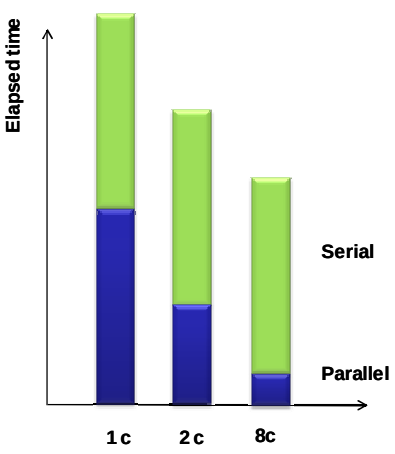

## References


- "The Software Vectorization Handbook" by Aart Bik
- "The Software Optimization Cookbook" by Richard Gerber
- Intel® C++ Compiler User's Guide
- IA32 Optimization Reference Manual
- High Performance Computing, 2nd Edition by Kevin Dowd and Charles Severance
- "Floating Point Calculations and the ANSI C, C++ and Fortran Standard"
- http://cache-www.intel.com/cd/00/00/33/01/330130_330130.pdf
- Details on the SSE-4 instruction set
- http://cache-www.intel.com/cd/00/00/32/26/322663_322663.pdf
- Web-based and classroom training
  - www.intel.com/software/college
- White papers and technical notes
  - www.intel.com/ids
  - www.intel.com/software/products
- And have a look at
- http://whatif.intel.com
- Product support resources
  - www.intel.com/software/products/support

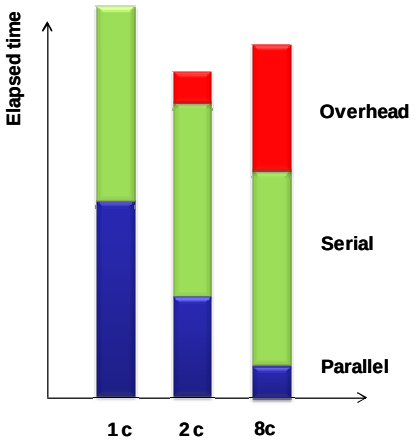




## Parallel computing: Amdhal law




In theory



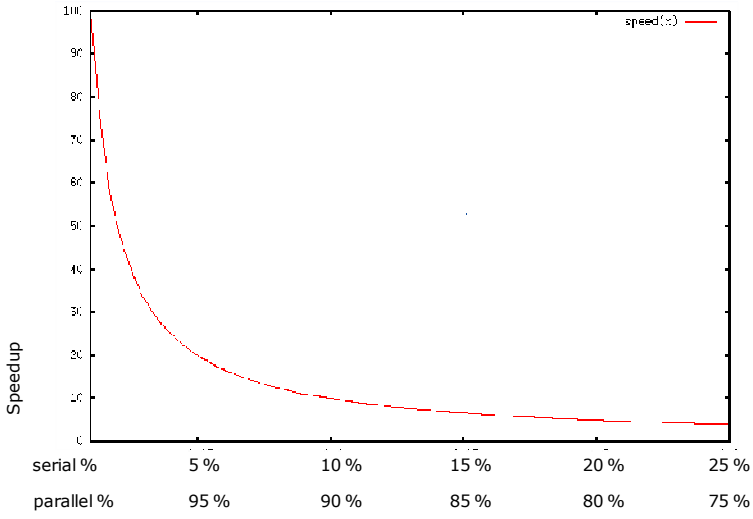

In practice




## Speedup using Amdhal

**Using infinite nb of processors**

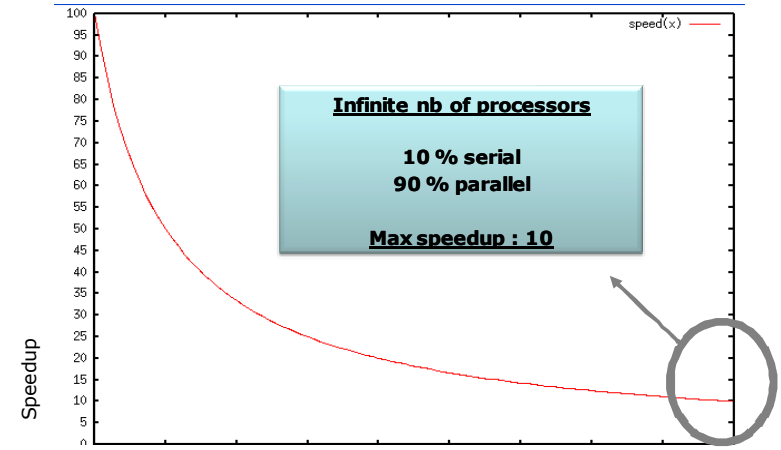
**lim (Speedup) = 1 / serial part**



## (Zoom)

**Using infinite nb of processors**

**lim (Speedup) = 1 / serial part**

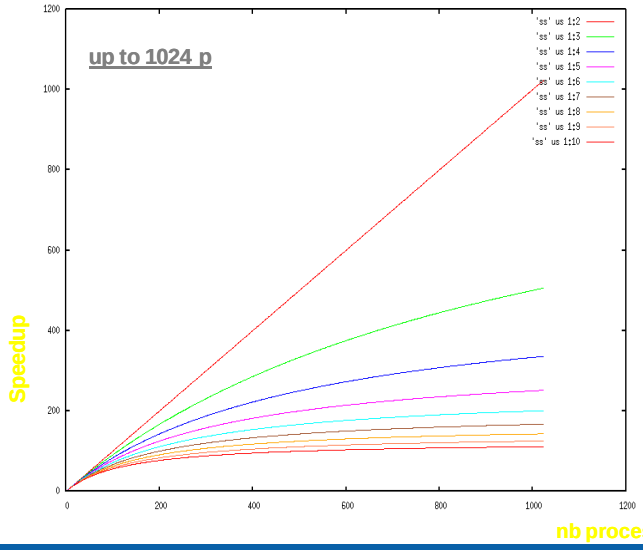



| | | | |
|---|---|---|---|
| serial % | 2 % | 5 % | 10 % |
| parallel % | 98% | 95 % | 90 % |





## Speedup curves from Amdhal law




100 %

parallel %

99.999 %

99.992 %

## Amdhal's Hypothesis (among others)

Amdhal law hypothesis:

- Code alone on the machine — Just a dream
- Not affected by the OS — OS tuning
- No overhead due to
  - -//ism (comms) — Hdw /sftw
  - - i/o's — Hdw /sftw
  - - Memory access (bw, latency) — hdw
- Load balancing is perfect — sftw
- serial optimization are not affected by //ism — sftw




## Real world

But when nb of processes increases :

- Amount of work / core decreases (can increase data locality) — Workload, sftw
- communications (size / nb) and sync do not decrease as the work / core — Hdw /sftw, sftw
- Load balancing becomes important

## Conclusion intermédiaire

**La loi d'amdhal**

**- n'est pas suffisante pour simuler précisément le comportement d'une appli**

**- Mais elle reste simple et didactique**

**- elle montre bien son « asymptote »**

**- d'autres approximations existent**

**- on peut passer sa vie à en obtenir de meilleures**

**- Permet d'estimer un taux de //ism**

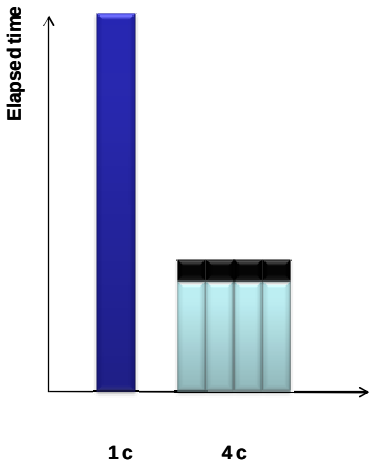
**(associé avec un profiling serial du nb de flops/s)**

**- Son aspect didactique permet de quantifier l'effort à (ou ne pas) fournir pour optimiser le code**

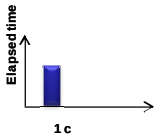




## Parallel computing

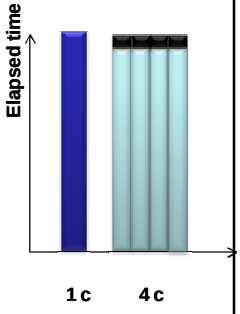



**0 -** • Serial Optim. Done (We reached XX % of the peak)

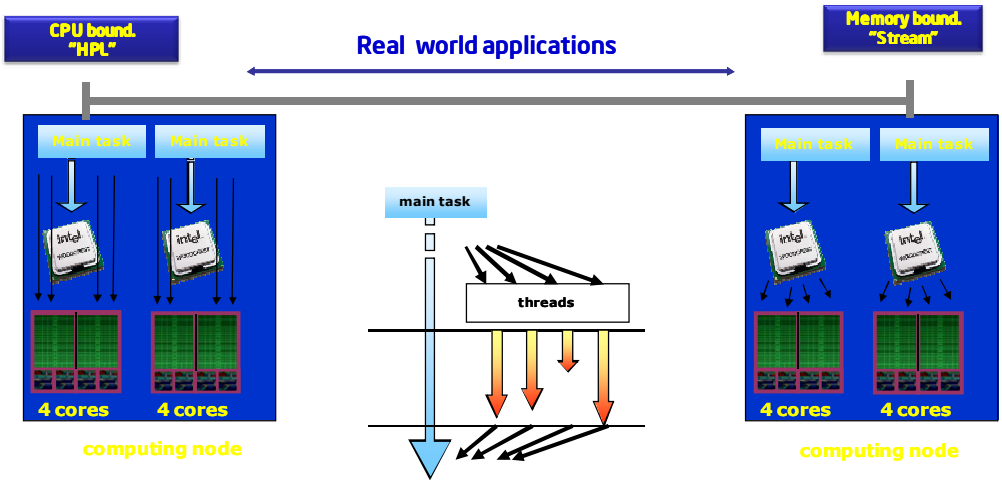



**1 -** • Work on the ovehead to become « super linear »

**2 -** •Take advantage of //ism to run 4 x bigger workloads ?



## Application classification vs. X-core




Try Find a trade-off between langages / apps / platform to limit stalls and overhead




## Most used

**Coarse and fine level parallelism :**

- Domain decomposition using MPI
- Secondary MPI level if coarse enough
- OpenMP / pthread at the finest level

(within a SMP node)

- Asynchron communications
- Asynchron and parallel i/o

→ 1 to 10k cores

*

→ 1 to 8 cores

~ max of 10 to 100k cores

## Levels of Parallelism

| | Serial Core Level | Multi-Core/SMP Node Level | Multi-Node Cluster Level |
|---|---|---|---|
| Programming Model, Implementation | C/C++ FORTRAN95 | Auto-Parallelization OpenMP* TBB | Cluster OpenMP Intel MPI |
| Correctness & Debugging | IDB | Thread Checker IDB | MPI Correctness Checker IDB-MPP |
| Performance Libraries | MKL IPP | MKL IPP | Cluster MKL |
| Performance Analysis | VTune™ | VTune Thread Profiler | Trace Analyzer |

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




## Intel® Software Development Products

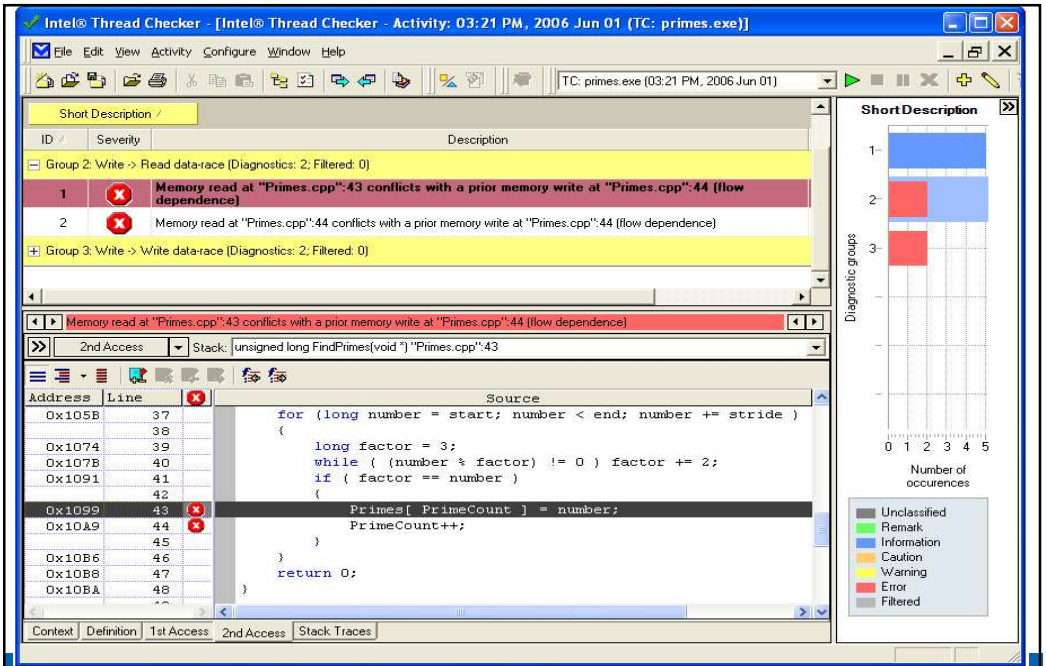
- Intel® Threading Analysis Tools
  - Find threading errors and optimize threaded applications for maximum performance
- Intel® Threading Building Blocks
  - C++ template-based runtime library that simplifies writing multithreaded applications for performance and scalability
- Intel® Cluster Tools
  - Create, analyze, optimize and deploy cluster-based applications


## Intel® Thread Checker v3.1

- Simplify multi-threaded application development
  - Detects challenging data races and deadlocks before they occur
  - Pinpoints errors to the source code line
  - Works on standard debug builds without recompiling
- Full compatibility with 32 & 64 bit Windows* and Linux* development environments
  - Includes command line interface for Windows and Linux
  - Batch scripts integration for regression test runs
- Support OpenMP*, WIN32 and PThreads, TBB

| Windows* | Linux* | Mac* | IA32 | Intel64 | IA64 | Multicore |
|---|---|---|---|---|---|---|
| √ | √ | | √ | √ | | √ |



## Intel® Thread Profiler v3.1

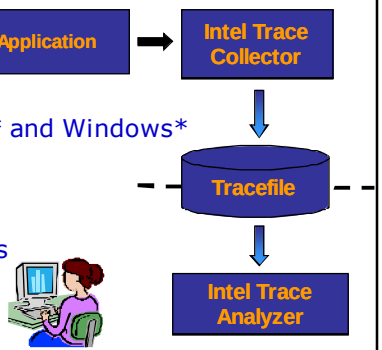
- ***Pinpoint*** threading inefficiencies to maximize threaded performance
  - Simplify multi-threaded application development
    - Easier to Use - Recall custom configuration settings
    - Faster to Use - User selectable stack walking
    - Support for Threading Building Blocks API
  - Maximize application performance
    - View application concurrency level to ensure full core utilization
    - Identify where thread related overhead impacts performance
    - Find out which created threads are active and which are inactive
  - Utilize one threading profiler for 32 & 64 bit Windows*
  - Included with VTune™ Analyzer for Windows*

| Windows* | Linux* | Mac* | IA32 | Intel64 | IA64 | Multicore |
|---|---|---|---|---|---|---|
| √ | √ | | √ | √ | | √ |

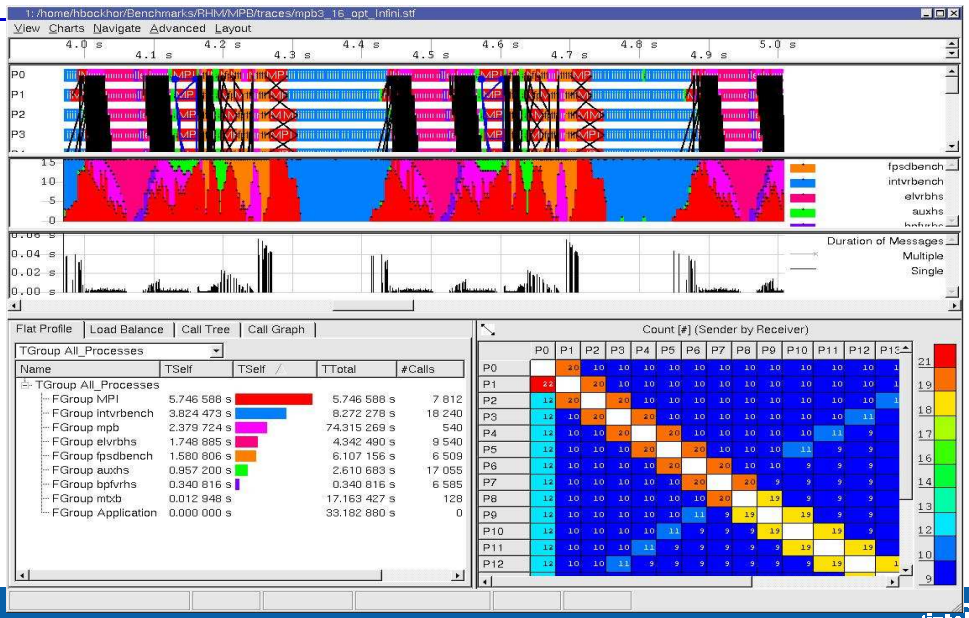



## Intel® Trace Analyzer and Collector 7.1

- Increase productivity and cluster application performance
- Very low impact
- Lightweight statistics mode
- Excellent scalability on time and processors
- Compiler and binary instrumentation
- MPI Correctness Checking library GUI on Linux* and Windows*
- Comparison of multiple trace files
- Timeline display for performance counters
- Powerful new aggregation and filtering functions
- Better and faster GUI
- Embedded Help



# Intel® Trace Analyzer








# Technical Information about Parallelism

- Watch Intel Software Development Products innovative online series produced by CNET Networks International at:

- ZDNet Asia: http://www.zdnetasia.com/whiteboard/0,39069063,62032144,00.htm
- ZDNet Australia: http://www.builderau.com.au/whiteboards/intel/video-1.htm
- ZDNet China: Coming soon
- ZDNet France: http://www.zdnet.fr/special/whiteboard-intel/
- ZDNet Germany: http://www.zdnet.de/specials/whiteboard-series/
- ZDNet Japan: Coming soon
- ZDNet UK: http://resources.zdnet.co.uk/articles/video/0,1000002009,39289127,00.htm

- **Go-parallel.com**
  - Resource for technical information about:
    - Parallelism
    - Optimization
    - Threading
    - Hpc

- **Multithreading Tools and Techniques for Software Developers: A Technical Webinar Series**
  - Sign up today!

## Conclusion intermédiaire

**le Petaflops soutenu semble difficile à atteindre …**

- Opposition « loi d'Amdhal » versus nombre de coeurs croissants
- + le cout, la consommation, l'espace



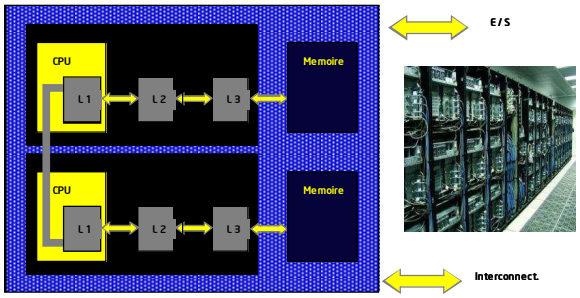

## Conclusion intermédiaire

**le Petaflops est possible si …**

- les plateformes deviennent :
  - beaucoup plus "parallèles" ou plus spécialisées
  - beaucoup plus puissante
  - beaucoup plus fiables ?
- les applications deviennent :
  - moins sensibles aux problèmes matériels
  - 100x plus parallèles
  - après un profond remaniement …

## Conclusion intermédiaire : Problèmes à résoudre

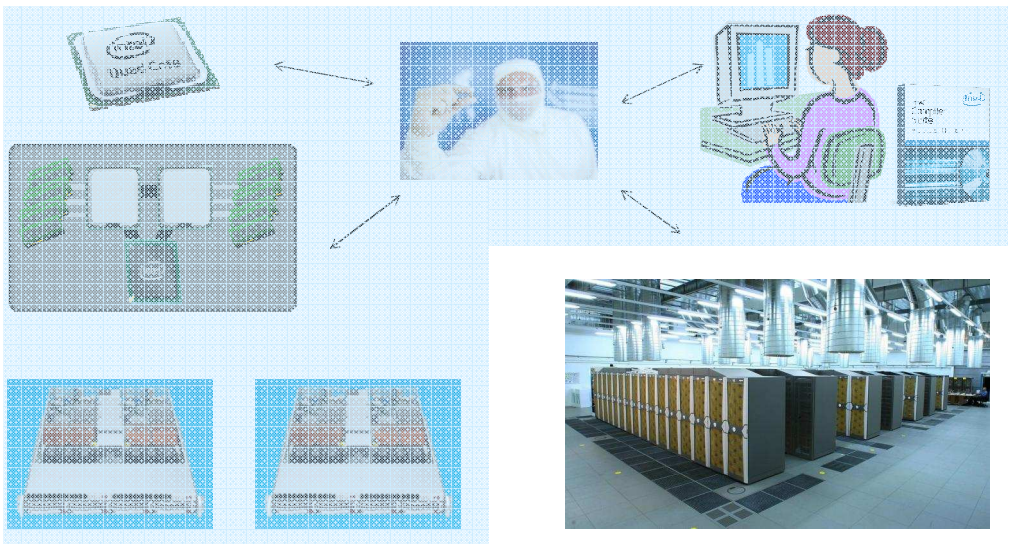
- Hardware (électronique)
  - Taille , consommation, type de « transistors »
  - Communications : latence et BW
    - CPU ←→ memoires ←→ i/o
    - Sockets ←→ Sockets
    - Cores ←→ cores



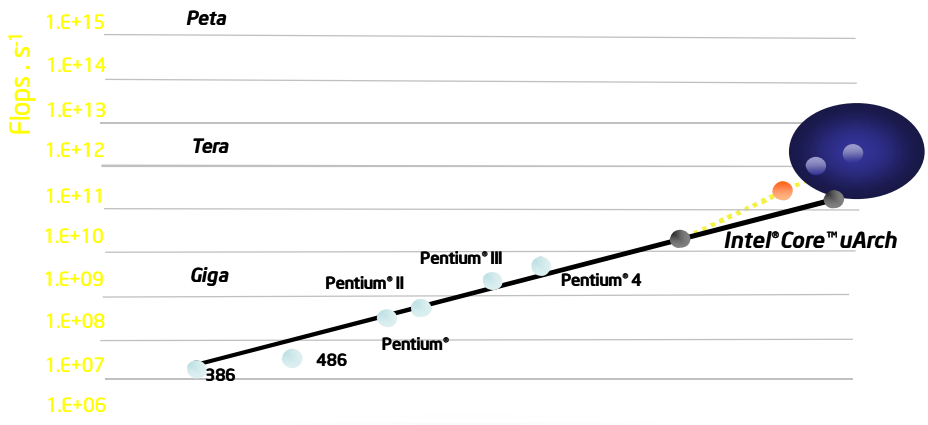




## Agenda: from the micro-arch to the end users



## Petaflops : too much cores to handle ?




*How to get the Petaflops with less powerfull cores ?*

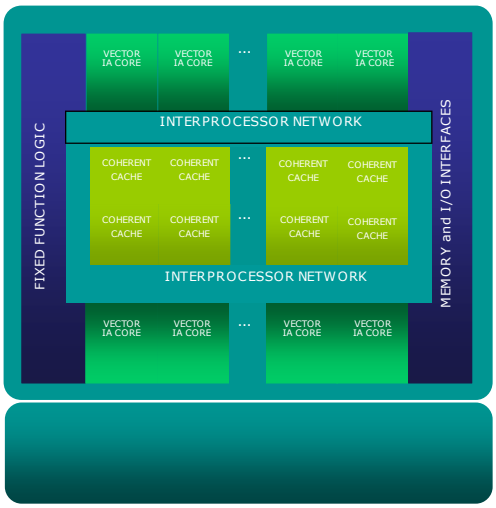




## Accelerators – Observations

- Accelerators may work well at an algorithm level but typically do not meet speedup goals from an application standpoint.
- Multi-core performance has not been fully exploited
  - Projected accelerator speedup's are typically biased to comparing against single CPU or single core instead of complete systems (2S, 4S, Clusters)
- Software agility is lacking
  - Development, debug, optimization
  - Support, maintainability across generations of devices
- IT deployment criteria is difficult to meet

## Throughput Computing




**Intel Many Core Breakthroughs**

- Array Of Fully Programmable IA Cores
- Innovative Hardware Caching Architecture
- Scales To Tera FLOPS
- Single Development Environment
- Single Software Executable

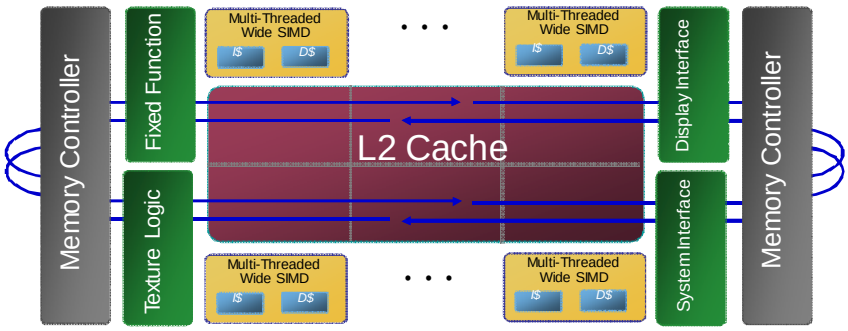
Larrabee is Intel's first many core processor under development

Each core is a complete Intel processor

Efficient inter-block communication

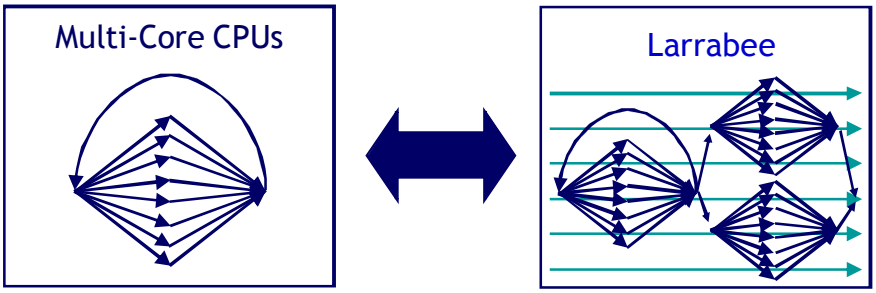



## Larrabee Block Diagram




- Cores communicate on a wide ring bus
  - Fast access to memory and fixed function blocks
  - Fast access for cache coherency
- L2 cache is partitioned among the cores
  - Provides high aggregate bandwidth
  - Allows data replication & sharing

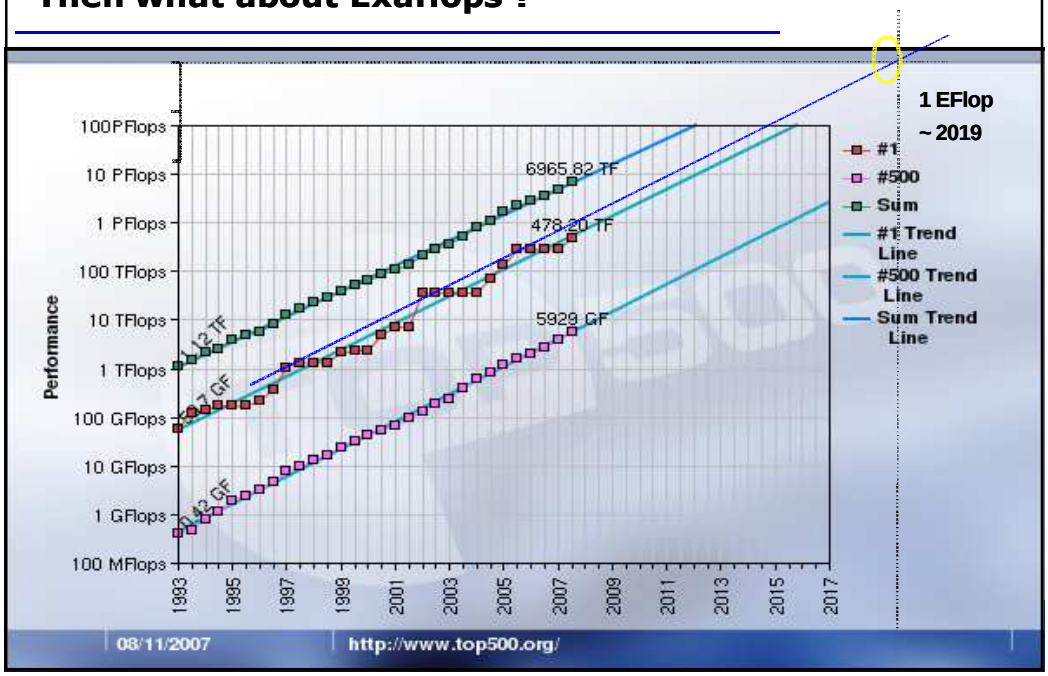
## Braided Parallelism Support




- Intel many-core combines best of GPU and CPU
  - IA many core can submit work to itself
  - Create "braided parallelism" that intermixes *data-* and *task-parallelism* with *sequential code*
- SW compatibility , Same data formats
- High performance Math libraries




## Then what about Exaflops ?



## ExaFlops Peak with today CPUs ?

peak perf.

48 GF *22 => 1 TF *1000 = > 1 PF * 1000 => 1 Exa Flop

power (board)

225 W => 5000 W => 5 MW => 5 GW

x1.5 for cooling

nb. cores

4 cores => 88 cores => 88 K cores => 88 M cores

101

## Low power for the ExaFlops ?

peak perf.

1 TF *1000 = > 1 PF * 1000 => 1 Exa Flops

power.

100 W (including cooling) => 0.1 MW (including cooling) => 100 MW (including cooling)

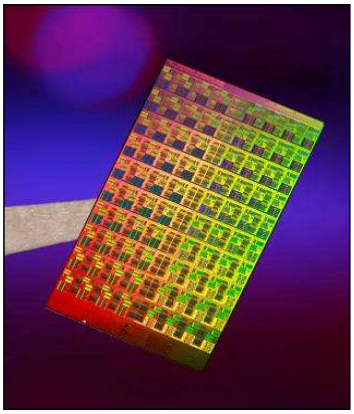
nb. cores

1 "CPU" => 1 K "CPU" => 1M "CPU"

102

## Teraflops Research Chip

**100 Million Transistors • 80 cores • 275mm2 • 62 W • 1 TFlops**

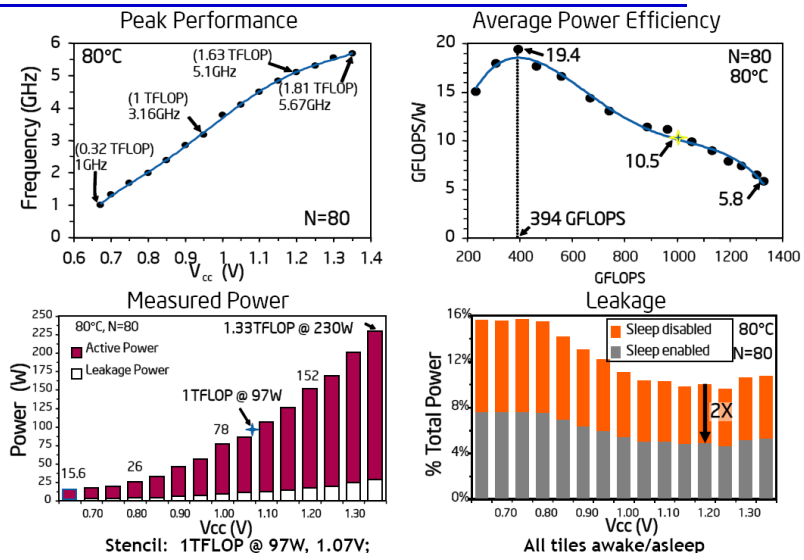


| Frequency | Voltage | Power | Bisection Bandwidth | Performance |
|---|---|---|---|---|
| 3.16 GHz | 0.95 V | 62W | 1.62 Terabits/s | 1.01 Teraflops |
| 5.1 GHz | 1.2 V | 175W | 2.61 Terabits/s | 1.63 Teraflops |
| 5.7 GHz | 1.35 V | 265W | 2.92 Terabits/s | 1.81 Teraflops |





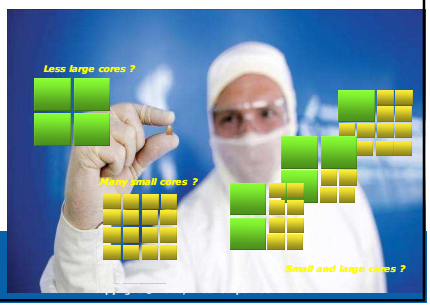
## Teraflops Research Chip




Stencil: 1TFLOP @ 97W, 1.07V; All tiles awake/asleep

## Core size (at constant 100-200—300 Watts/die) ??

- **Many (~1000--2000) small cores:**
  - Think of using low power (say mobility) x86 Processors and putting many of them on a die.
  - In-order, SSE-n with 2, 4 or perhaps 8 ops per clock 1--2 GHz clock
  - Many threads per core
- **A bunch of really big cores (512):**
  - very wide (32 ops) Vector units
  - 4 / 8 threads per core
  - 2 / 4 GHz





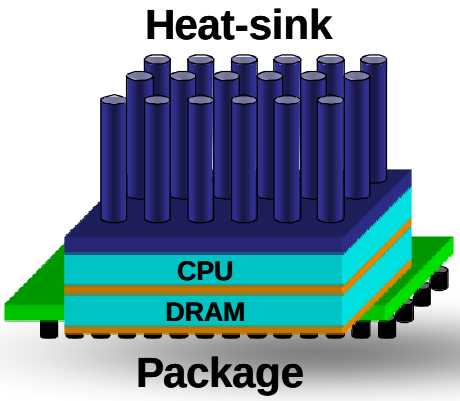

## Memory solutions

- We cannot match memory bandwidth to off-package memories with increase in processing speed.
- We are hitting a wall in terms of number of pins, signaling area, and signaling power.
- Memory speeds are advancing more slowly than CPU speeds

**When we have many tera-ops processors**
**We'll need many TB/s Memory systems**

| | Actual values | "Tera - value" | Gain |
|---|---|---|---|
| Performance | 20 Gflops / p | 1 Tflops / p | ~50 x |
| BW | 12 GB/s | ~1.2 TB/s | ~100 x |
| Latency | 400 cycles | 20 cycles | ~20 x |

## Memory Bandwidth futures: 3D Die Stacking

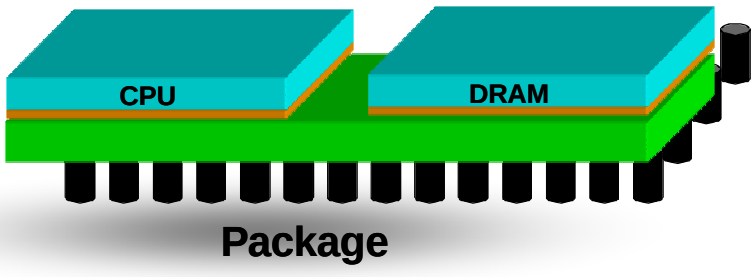



- Power and IO signals go through DRAM to CPU
- Thin DRAM die
- Through DRAM vias

*DRAM, Voltage Regulators, and High Voltage I/O All on the 3D integrated die*

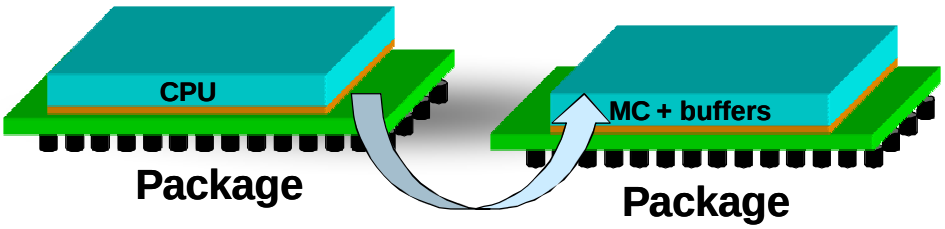



## Memory Bandwidth options: DRAM on Pkg




*DRAM, CPU integrated on die*

## Memory Bandwidth options




Replace on-pkg MC with very fast flex links to an on-board MC




## Possible Technology insertion timeline

- 2011: terascale processors
- 2015: terascale Si-photonics
- 2015: Aggressive memory stacking
  - fault tolerant networks
- 2015: 2--10 Tera-ops processors
  - optical memory interconnects
- 2018: 10's of tera-ops processors
  - "All" memory is on package
- 2018: 10's of terabits/sec/fiber

## Summary

- Intel research is addressing the challenges of parallel computing with Intel platforms
  - Teraflop hardware performance within mainstream power and cost constraints
  - ISA enhancements to address emerging workload requirements
  - Language and runtimes to better support parallel programming models
- Intel is developing hardware and software technologies to enable Tera-scale computing
  - Extending our proven standard compliant tools to many-core
- Prepare for Tera-scale by optimizing for multi-core
  - Scalable algorithms
  - Parallel SW tools – OpenMP 3.0, TBB, MKL & IPP libraries, MPI
  - Experiment with whatif.intel.com offerings and give us your feedback