Performance Engineering of Seismic Simulations for Exascale Architectures

December 07, 2015

Olaf Schenk

Institute of Computational Science

Universita della Svizzera italiana, Lugano

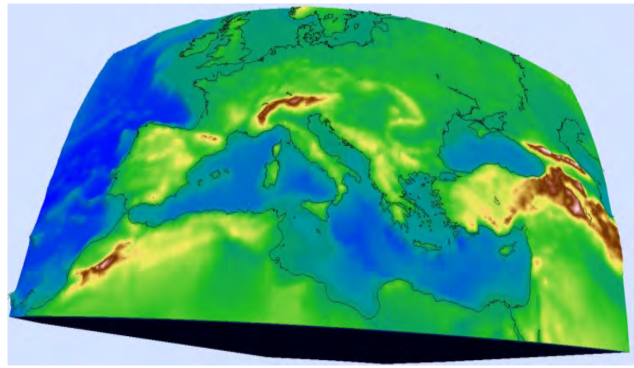
Joint work with:

M. Rietmann (USI), W. Vanroose (U Antwerp, Intel ExaScale Lab)

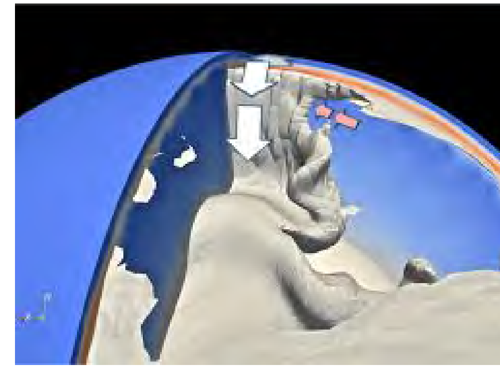
Y. Cui (SCEC), A. Fichtner (ETH Zurich), J. Tromp (Princeton)

Applications of Large-Scale Optimization and HPC

Power-Grid Optimization under Uncertainty (Stochastic Programming)



Computational Wave Propagation



Seismic Inversion / Global Tomography

SNF Projects (2006-...)

$$\min_{\mathbf{x}} F(\mathbf{x})$$
$$\text{subject to } c_i(\mathbf{x}) = 0 \quad \text{for } i \in \mathcal{E}$$
$$c_i(\mathbf{x}) \geq 0 \quad \text{for } i \in \mathcal{I}$$

U.S. DEPARTMENT OF ENERGY INCITE LEADERSHIP COMPUTING

pasc Platform for Advanced Scientific Computing

HP2C High Performance and High Productivity Computing

TOP 500 List of Supercomputers

Rank	Site	System	Cores	Rmax in Tflops/s	Rpeak in Tflops/s	Power (KW)
1	National Super Computer Center China	**Tianhe-2 (MilkyWay-2)** - Intel Xeon Intel Xeon Phi	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge United States	**Titan** - Cray XK7 Opteron + NVIDIA K20x	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQ, IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN, Japan	K computer, SPARC64 Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne United States	**Mira** - BlueGene/Q, IBM	786,432	8,586.6	10,066.3	3,945
6	CSCS, Switzerland	**Piz Daint** - Cray XC30, Intel Xeon , NVIDIA K20x	115,984	6,271.0	7,788.9	2,325
7	Texas Advanced Computing Center, US	**Stampede** - Intel Xeon, Intel Xeon Phi, Dell	462,462	5,168.1	8,520.1	4,510
8	Forschungszentrum Juelich (FZJ), Germany	**JUQUEEN** - BlueGene/Q, IBM	458,752	5,008.9	5,872.0	2,301
9	DOE/NNSA/LLNL United States	**Vulcan** - BlueGene/Q, IBM	393,216	4,293.3	5,033.2	1,972
10	Leibniz Rechenzentrum Germany	**SuperMUC** - Xeon E5-2680 IBM	147456	2,897.0	3,185.1	3,423

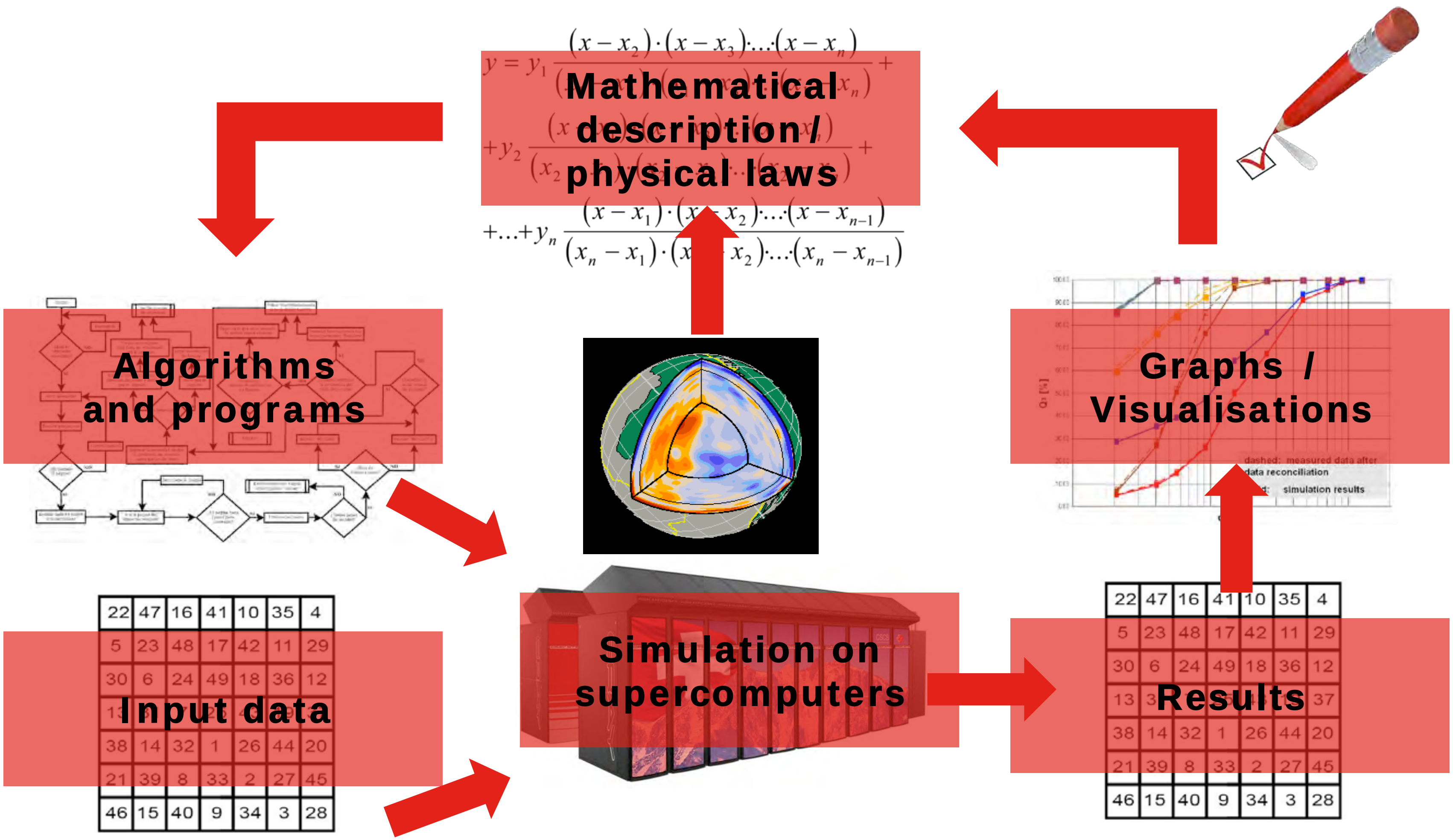
Agenda

- **Swiss Platform for Advanced Scientific Computing**
  - Supercomputing Architectures
- **Performance Characteristics of Many-Core Architectures**
  - Roofline model, Arithmetic intensity
- **Structured Grid Simulations on Many-Core Architectures**
  - High-Productivity & High-Performance Stencil Compiler Framework
- **Parallel Nonlinear Optimization Methods**
- **Conclusion**

HP2C & PASC

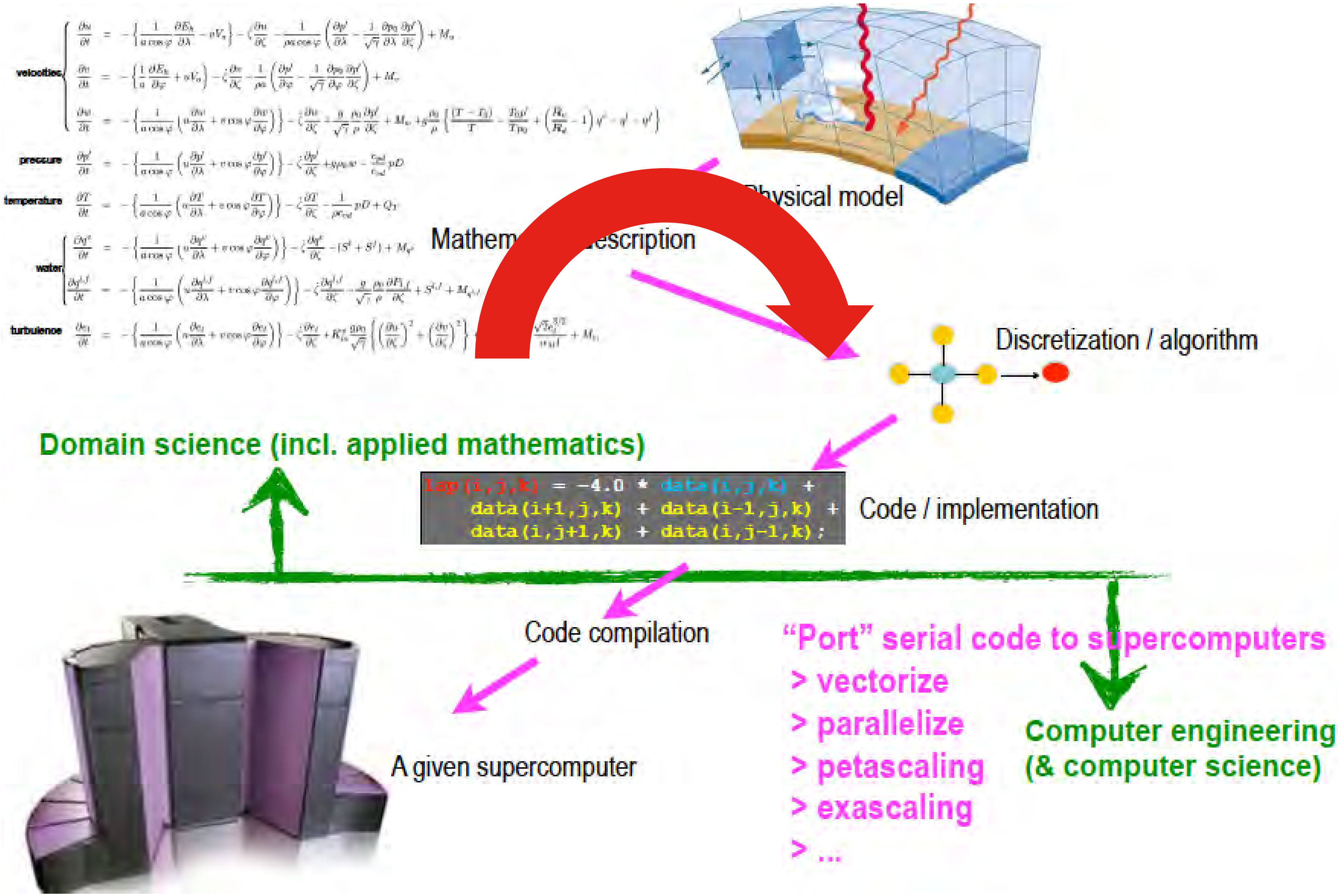
Swiss Platform for Advanced Scientific Computing

How do computational scientists work?



Università della Svizzera italiana

Università della Svizzera italiana | Faculty of Informatics | Institute of Computational Science ICS



Università della Svizzera italiana

Università della Svizzera italiana | Faculty of Informatics | Institute of Computational Science ICS

The Swiss Platform for High-Performance and High-Productivity Computing (2010-2013)

Goal of HP2C:

- **Enable computational sciences to make effective use of next generation supercomputers**
- **New Supercomputing building for CSCS in proximity to academic institution (USI)**
- **Facility for the next generation of peta- / exacale machines in Switzerland**
- **New Institute of Computational Science at USI Lugano**
- **HP2C Project: Scientific Computing Research in Cooperation with Swiss Universities**

What is a supercomputer?

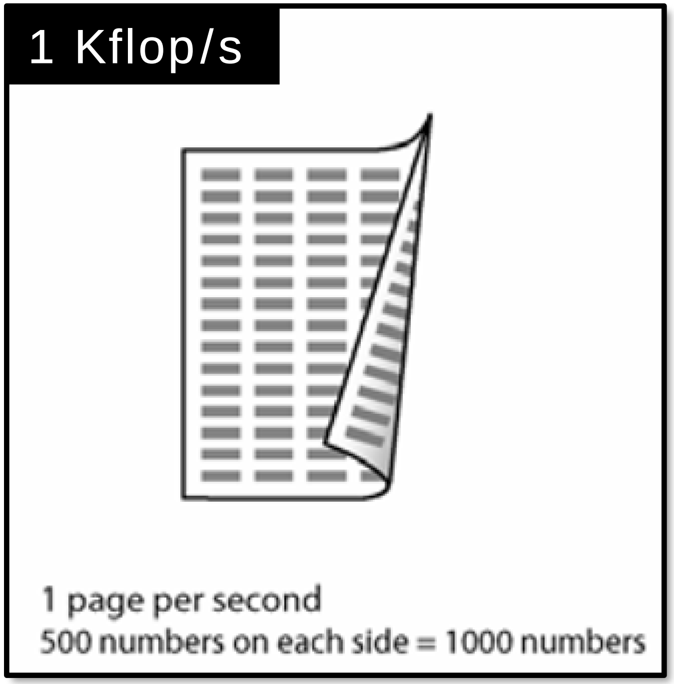
Units of Measure in Computing

- **High Performance Computing (HPC) units are:**
  - Flops: floating point operations
  - Flop/s: floating point operations per second
  - Bytes: size of data (double precision floating point number is 8)
- **Typical sizes are millions, billions, trillions…**

Mega	**Mflop/s** = 10^{6} flop/sec	Mbyte = 10^{6} byte
Giga	**Gflop/s** = 10^{9} flop/sec	Gbyte = 10^{9} byte
Tera	**Tflop/s** = 10^{12} flop/sec	Tbyte = 10^{12} byte
Peta	**Pflop/s** = 10^{15} flop/sec	Pbyte = 10^{15} byte
Exa	**Eflop/s** = 10^{18} flop/sec	Ebyte = 10^{18} byte

Units of Measure in Computing

- **Let us say you can print:**
  5 columns of 100 number each; on both sides of the page = 1000 numbers (Kflop) in one second (1 Kflop/s)



Units of Measure in Computing

- **Let us say you can print:**

  1000 pages about 10 cm = 10^6 numbers (Mflop)

  2 reams of paper per seconds **(1 Mflop/s)**

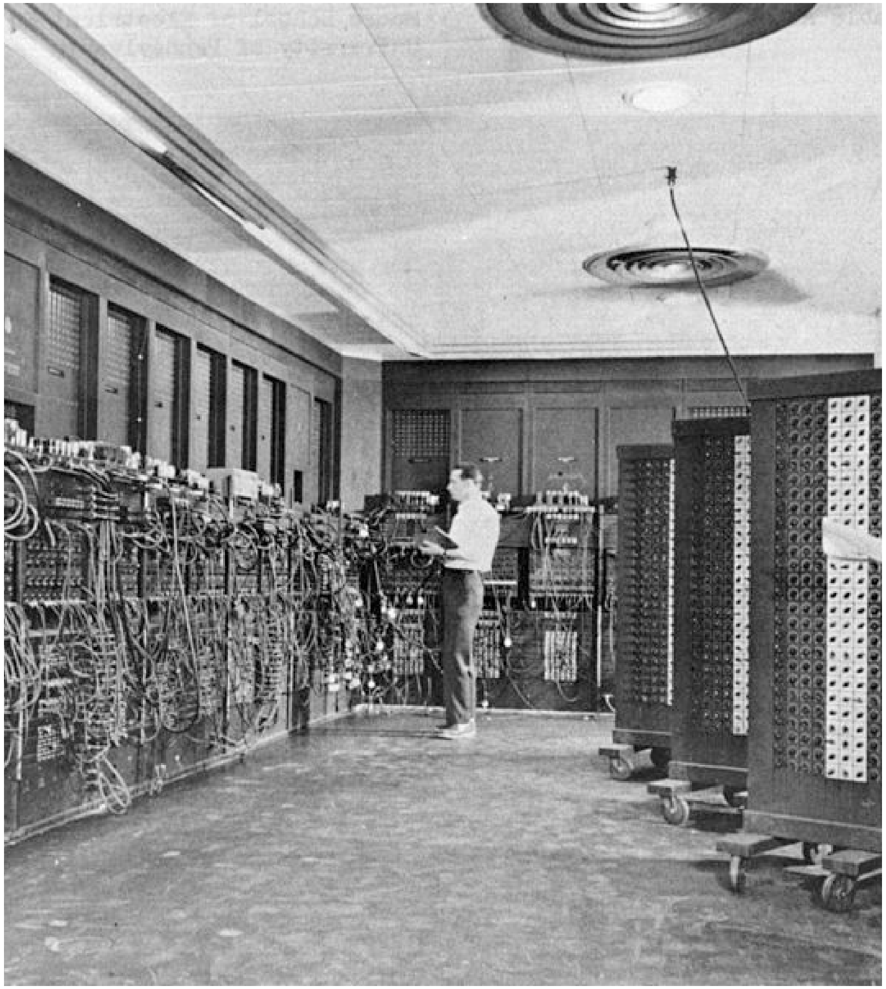
Units of Measure in Computing

- **Let us say you can print:**
  10^{15} numbers (Pflop) = 100,000 km
  (1/4 distance to the moon) stack printed per second (1Pflop/s)

Electronic Numerical Integrator And Computer

ENIAC, USA, 1946

- **Ballistic Calculations**
- **Size**
  - 27 t
  - 2.4 m × 0.9 m × 30 m
  - 150 kW
- **Cost: $5’900’000**



5 KFLOPS

Earth Simulator, Japan, 2002

- **Run global climate models**
- **Size**
  - Interconnect 14 m x 13 m
  - Computer 41 m x 40 m
  - 6.4 MW
- **Cost $400,000,000**

35.86 TFLOPS

CSCS - The new office building

The computer building: machine room

Cray XC30 „Piz Daint", Switzerland, 2015

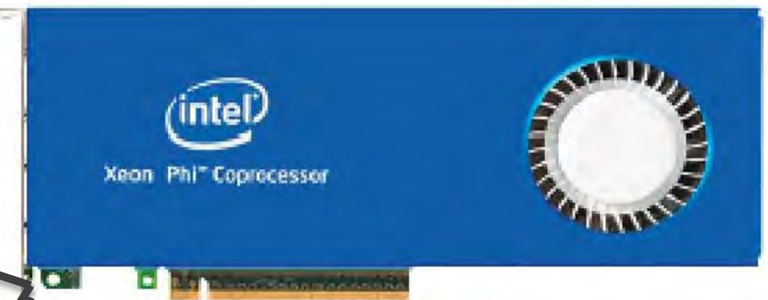
- **User Lab for Swiss Scientists**
- **115‘984 cores - 272 TB of RAM**
- **4PB TB local disks**

- **Size**
  - 23 t
  - 47 m^2
  - 2,325 MW

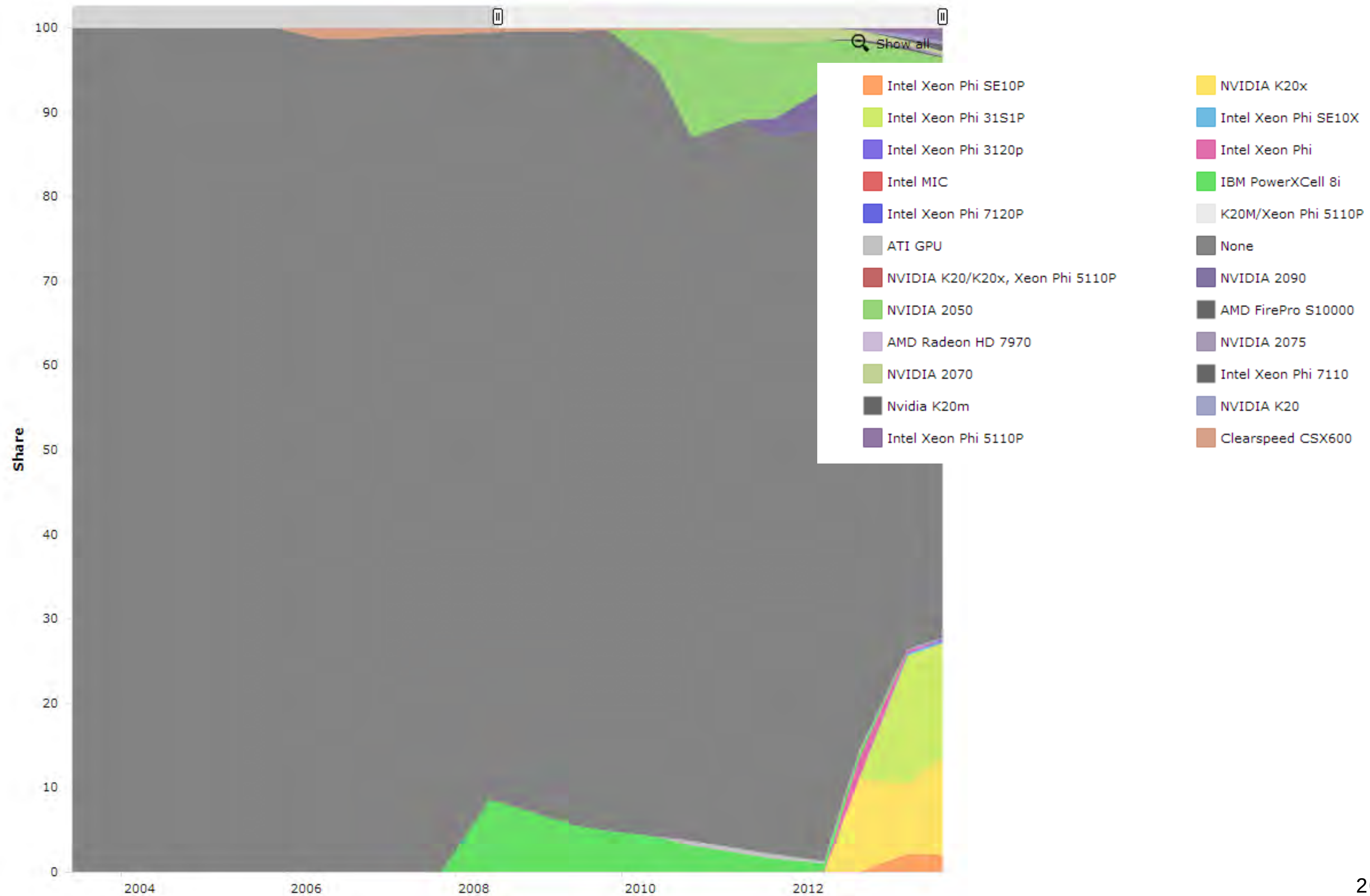
6.700 PFlops

TOP 500 List of Supercomputers (June 2014)

Rank	Site	System				
1	National Super Computer Center China	**Tianhe-2 (MilkyWay-2)** - Intel Xeon Phi				
2	DOE/SC/Oak Ridge United States	**Titan** - Cray XK7 Opteron + NVIDIA K20x				
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQ, IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN, Japan	K computer, SPARC64 Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne United States	**Mira** - BlueGene/Q, IBM				
6	CSCS, Switzerland	**Piz Daint** - Cray XC30, Intel Xeon , NVIDIA K20x				
7	Texas Advanced Computing Center, US	**Stampede** - Intel Xeon, Intel Xeon Phi, Dell				
8	Forschungszentrum Juelich (FZJ), Germany	**JUQUEEN** - BlueGene/Q, IBM				
9	DOE/NNSA/LLNL United States	**Vulcan** - BlueGene/Q, IBM	393,216	4,293.3	5,033.2	1,972
10	Leibniz Rechenzentrum Germany	**SuperMUC** - Xeon E5-2680 IBM	147456	2,897.0	3,185.1	3,423



Accelerators / Development over time



Three types of modern accelerators

GPU: NVIDIA Tesla K20c

Kepler GK110, 28 nm

13 mp × 192 cores @ 0.71 GHz

5 GB GDDR5 @ 2.6 GHz

225W

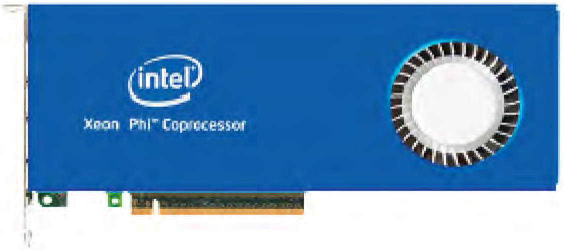
GPU: Radeon HD 7970

Graphics Core Next, 28 nm

32 mp × 64 cores @ 1 GHz

3GB GDDR5 @ 1.5 GHz

250W



MIC: Intel Xeon Phi 3120A

Knights Corner (KNC), 22 nm

57 cores @ 1.1 GHz

6GB GDDR5 @ 1.1 GHz

300W

up to 4 threads per core

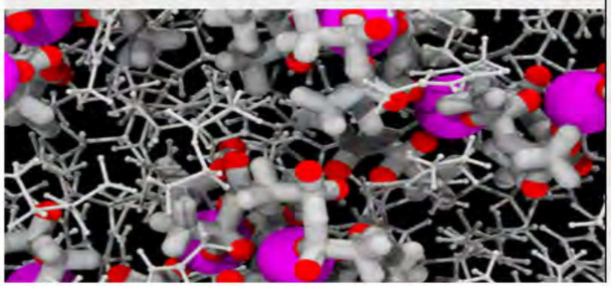
512-bit vectorization (AVX-512)

Swiss Platform for Advanced Scientific Computing (PASC)

pasc
Platform for Advanced Scientific Computing

ABOUT | NEWS | NETWORKS | PROJECTS | ACTIVITIES | CONTACTS

Materials Simulations Network



Events

Platform of Advanced Scientific Computing Conference
14.10.2013

The first Platform of Advanced Scientific Computing Conference (PASC14) will take place on June 2...

Latest News

Additional Co-Design Projects accepted
10.01.2014

2014 Call for Co-Design Projects
10.01.2014

Welcome to the Swiss **Platform for Advanced Scientific Computing** (PASC) - PASC is a structuring project jointly supported by the Swiss University Conference (SUC) and the Council of Federal Institutes of Technology (ETH Board).

PASC is coordinated by the Università della Svizzera italiana (USI) in collaboration with CSCS, the Swiss National Supercomputing Centre of the ETH Zurich, and with the other Swiss universities and the EPF Lausanne.

The platform's overarching goal is to position Swiss computational sciences in the emerging exascale-era. It is complementary to the supercomputing-hardware-focused elements of the Swiss High-Performance and Networking (HPCN) initiative. The PASC consolidates and builds on the achievements of the current High-Performance and High-Productivity Computing (HP2C) project which supported 13 large-scale projects in the period 2009-2013.

PASC aims to promote joint effort to address key scientific issues in different domain sciences through interdisciplinary collaborations between domain scientists, computational scientists, software developers, computing centres and hardware developers. Thus, PASC builds on the principle of co-design, namely that software codes exploiting the potential of the next generation of computing architectures need to be jointly and interactively developed by these actors throughout the whole value chain.

SUK·CUS
Swiss university conference

ETH BOARD

http://www.pasc-ch.org

Università
della
Svizzera
italiana

PASC16 Conference, June 08-10, 2016, EPFL – Lausanne, Cosponsored by ACM, www.pasc16.org

PA SC 16 Platform for Advanced Scientific Computing Conference

Lausanne Switzerland | 08-10 June 2016

Association for Computing Machinery sighpc

Call for Papers

The Platform for Advanced Scientific Computing (PASC) is inviting submissions for the Papers Session of its next conference (PASC16) to be held from June 8 to 10, 2016 at the SwissTech Convention Center, located on the campus of the EPFL in Lausanne, Switzerland.

The PASC Conference is a leading event for researchers in computational science and high-performance computing. PASC16 builds on a successful history with 350 international attendees in 2015. PASC's structure enables efficient communication between various areas arranged in eight domain-specific tracks. The PASC papers program is soliciting high-quality contributions in all of these areas. Papers will be presented during the PASC16 Conference and published in the ACM Digital Library.

Areas of interest include (but are not limited to):

- Implementation strategies for computational science applications
- Programming languages and models for science domains
- Tools for application development
- Domain-specific libraries or frameworks
- Use of heterogeneous or advanced computing for scientific applications

To ensure the highest quality contributions, the ACM publication process includes multiple stages of review. Following the first round of reviews, authors whose submissions are conditionally accepted will have the opportunity to revise their manuscripts based on feedback prior to a second round of reviews. To ensure a timely dissemination of research results, contributors are required to work according to the following schedule:

- Abstract submission: **January 15, 2016**
- Full paper submission: **January 22, 2016**
- First review notification: **February 26, 2016**
- Revised submission: **March 11, 2016**
- Final review notification: **April 7, 2016**

Organization

- Conference co-chairs: **Jan Hesthaven** (EPF Lausanne, Switzerland), **Nicola Marzari** (EPF Lausanne, Switzerland), **Olaf Schenk** (Università della Svizzera italiana, Switzerland) and **Laurent Villard** (EPF Lausanne, Switzerland)
- Papers co-chairs: **Torsten Hoefler** (ETH Zurich, Switzerland) and **David Keyes** (King Abdullah University of Science and Technology, Saudi Arabia)

Editorial Board of the Scientific Tracks

The review process is organized in tracks. The editor of each track selects appropriate reviewers who are experts in the relevant area.

- **Climate & Weather**: Michael Wehner (Lawrence Berkeley National Laboratory & University of California, USA)
- **Computer Science & Mathematics**: David Keyes (King Abdullah University of Science and Technology, Saudi Arabia)
- **Emerging Domains**: Omar Ghattas (The University of Texas, USA)
- **Engineering**: George Biros (The University of Texas, USA)
- **Life Sciences**: Ioannis Xenarios (Swiss Institute of Bioinformatics, Switzerland)
- **Materials**: Mark van Schilfgaarde (King's College London, UK)
- **Physics**: George Lake (University of Zurich, Switzerland)
- **Solid Earth**: Jeroen Tromp (Princeton University, USA)

Submissions will be reviewed double blind (authors should not be listed and a reasonable effort should be made to anonymize the paper, e.g., referring in third person to own previous works).

Papers should be in the ACM proceedings format and should be no more than 10 pages in length (www.acm.org/publications/article-templates/proceedings-template.html).

Contributions are to be submitted online at www.pasc16.org. The submission system will open at the end of November 2015.

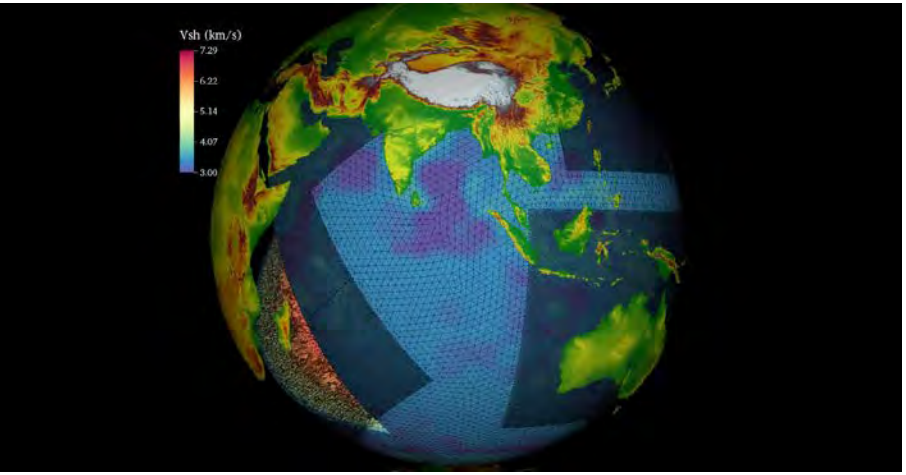
- PASC is delighted to launch a Call for Abstracts for its next conference PASC15 cosponsored by the Association for Computing Machinery (ACM).

sighpc ACM Special Interest Group on High Performance Computing

acm Association for Computing Machinery

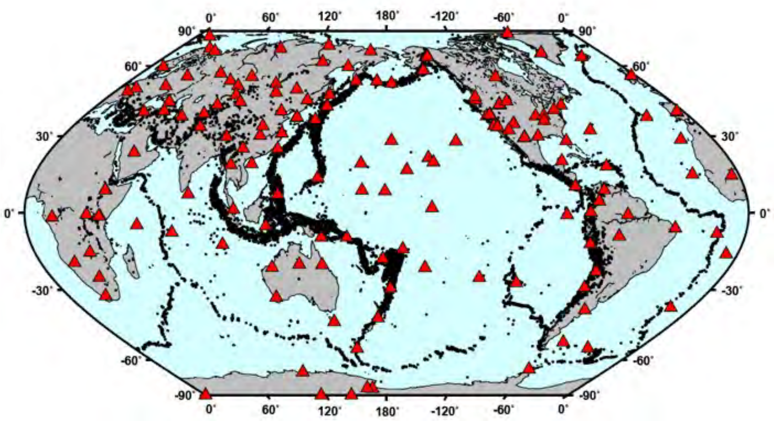
Advancing Computing as a Science & Profession

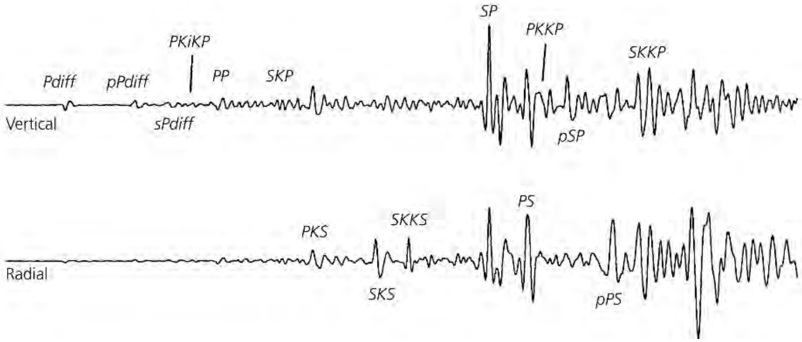
Geoscale / GeoPC projects (ETHZ, USI, CSCS)



Earth models for seismic (compressional/shear) velocities

Seismicity and seismometers





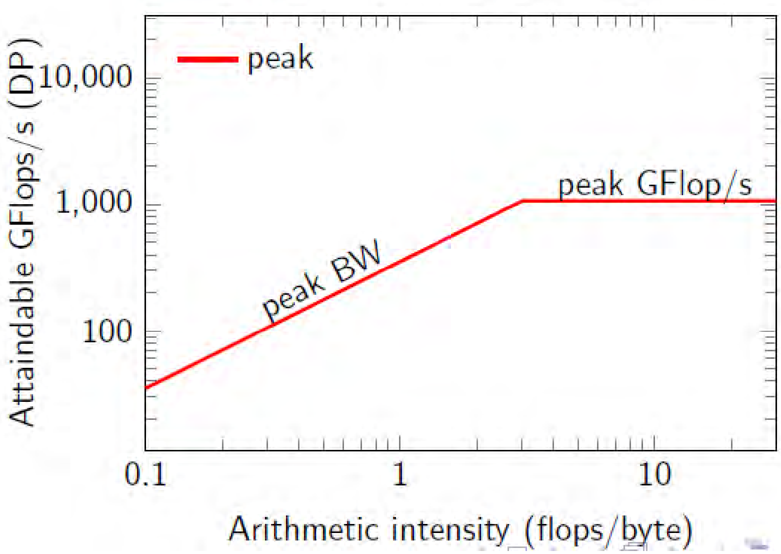
Performance Characteristics of Many-Core Architectures

Roofline Model

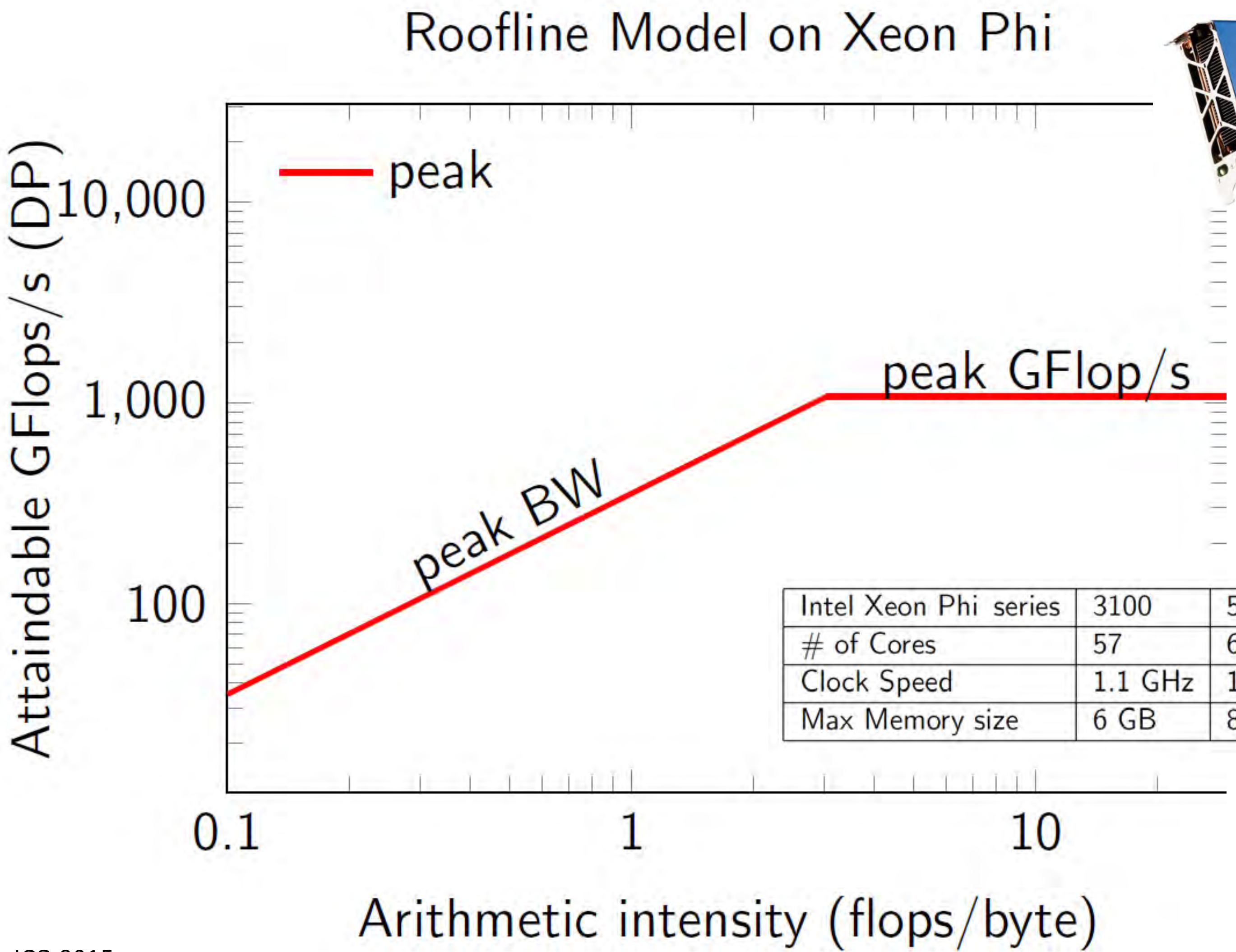
- **Arithmetic intensity**: $q = \frac{\text{floating-point operations}}{\text{byte off-chip memory traffic}}$
- High $q \rightarrow$ compute bound (dense algebra, FFT, ...)
- Low $q \rightarrow$ bandwidth bound (sparse algebra, stencils, ...)
- Performance Gflop/s = min(Peak Gflop/s, Stream BW$\times q$)
- Roofline gives upperbound for performance for given q

The Roofline Model: A pedagogical tool for program analysis and optimization (Williams, Patterson, 2008)

Roofline Model on Xeon Phi

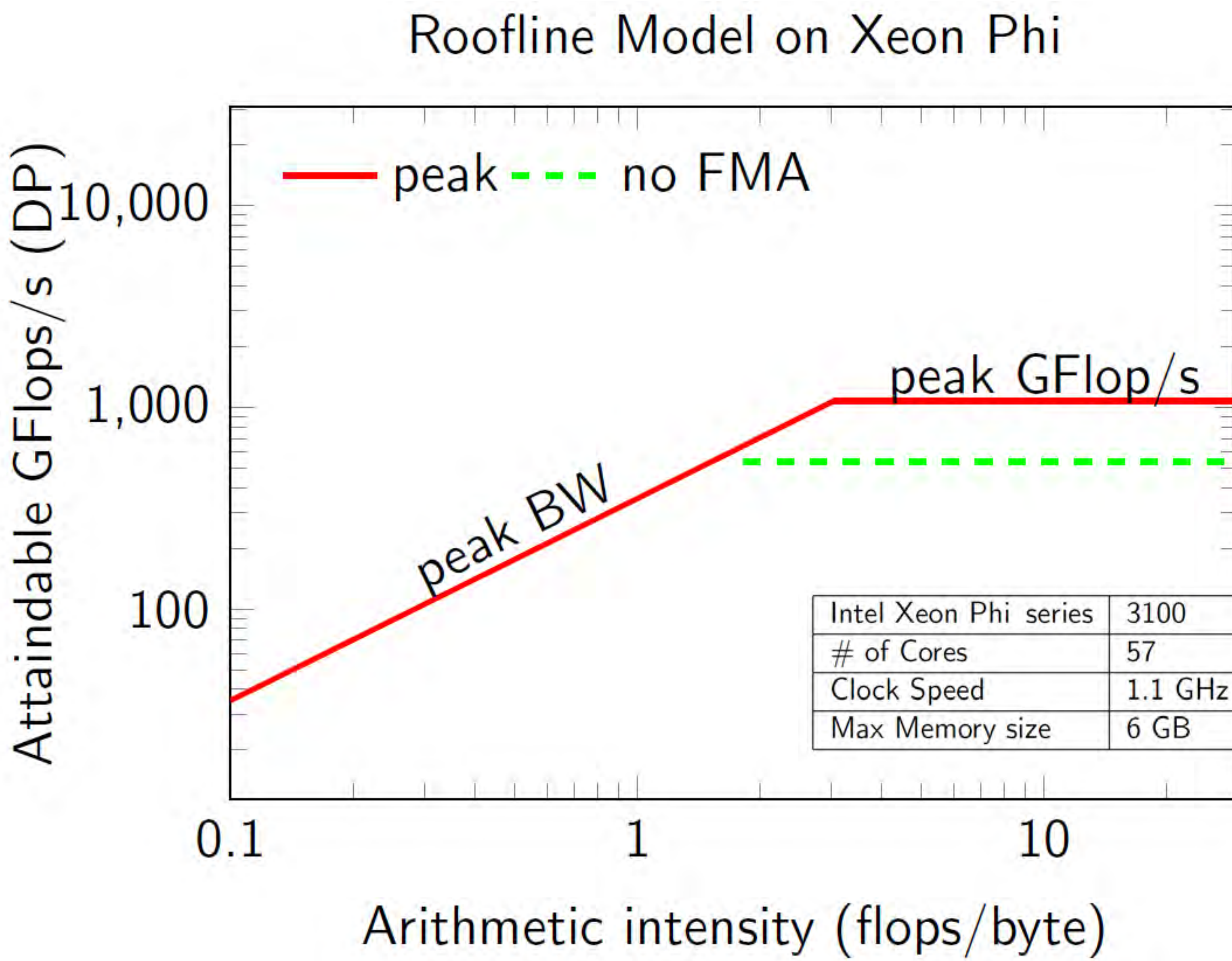


Roofline Model on Intel Xeon Phi



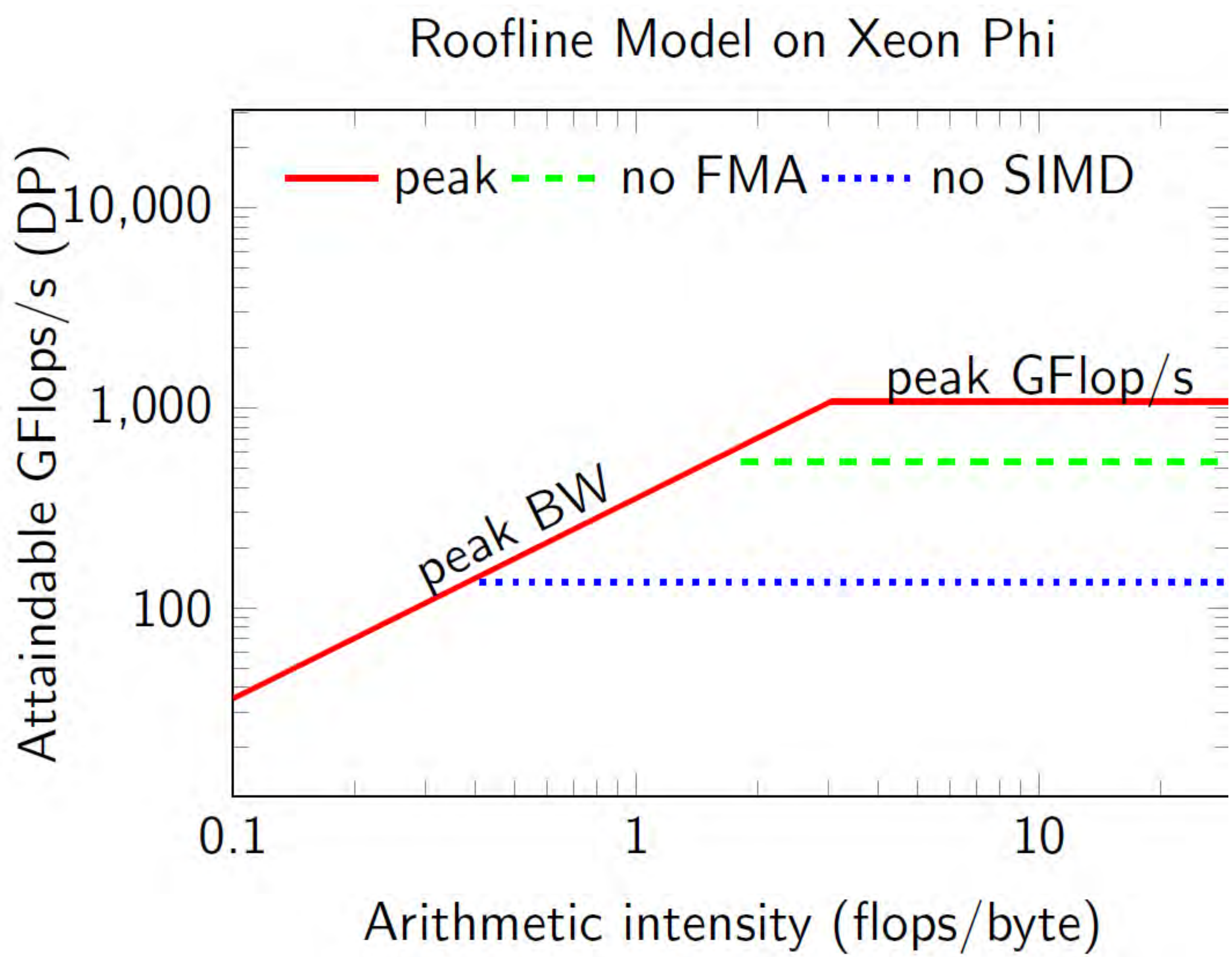
Intel Xeon Phi series	3100	5100	7100
# of Cores	57	60	61
Clock Speed	1.1 GHz	1.053 GHz	1.238 GHz
Max Memory size	6 GB	8 GB	16 GB

Roofline Model on Intel Xeon Phi



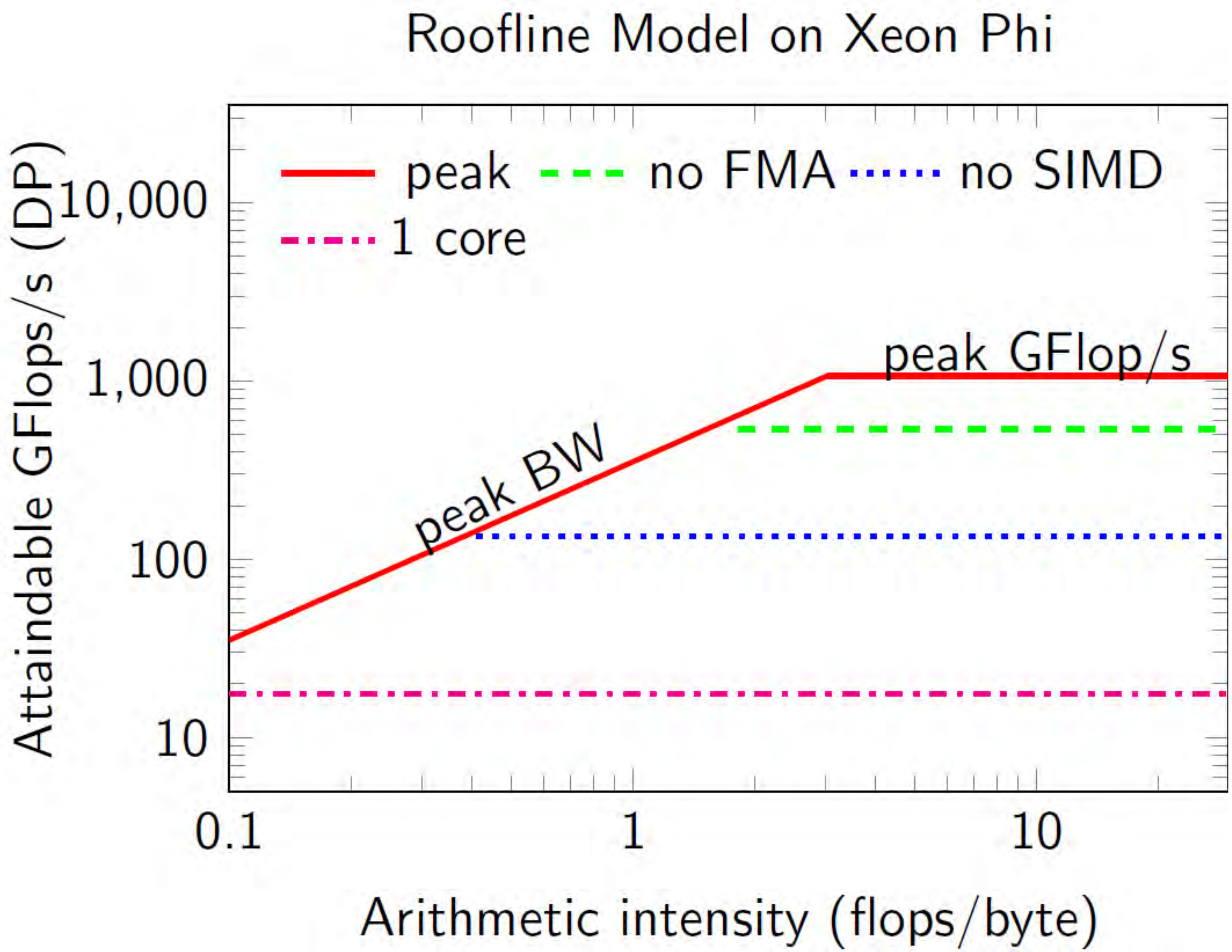
Intel Xeon Phi series	3100	5100	7100
# of Cores	57	60	61
Clock Speed	1.1 GHz	1.053 GHz	1.238 GHz
Max Memory size	6 GB	8 GB	16 GB

Roofline Model on Intel Xeon Phi



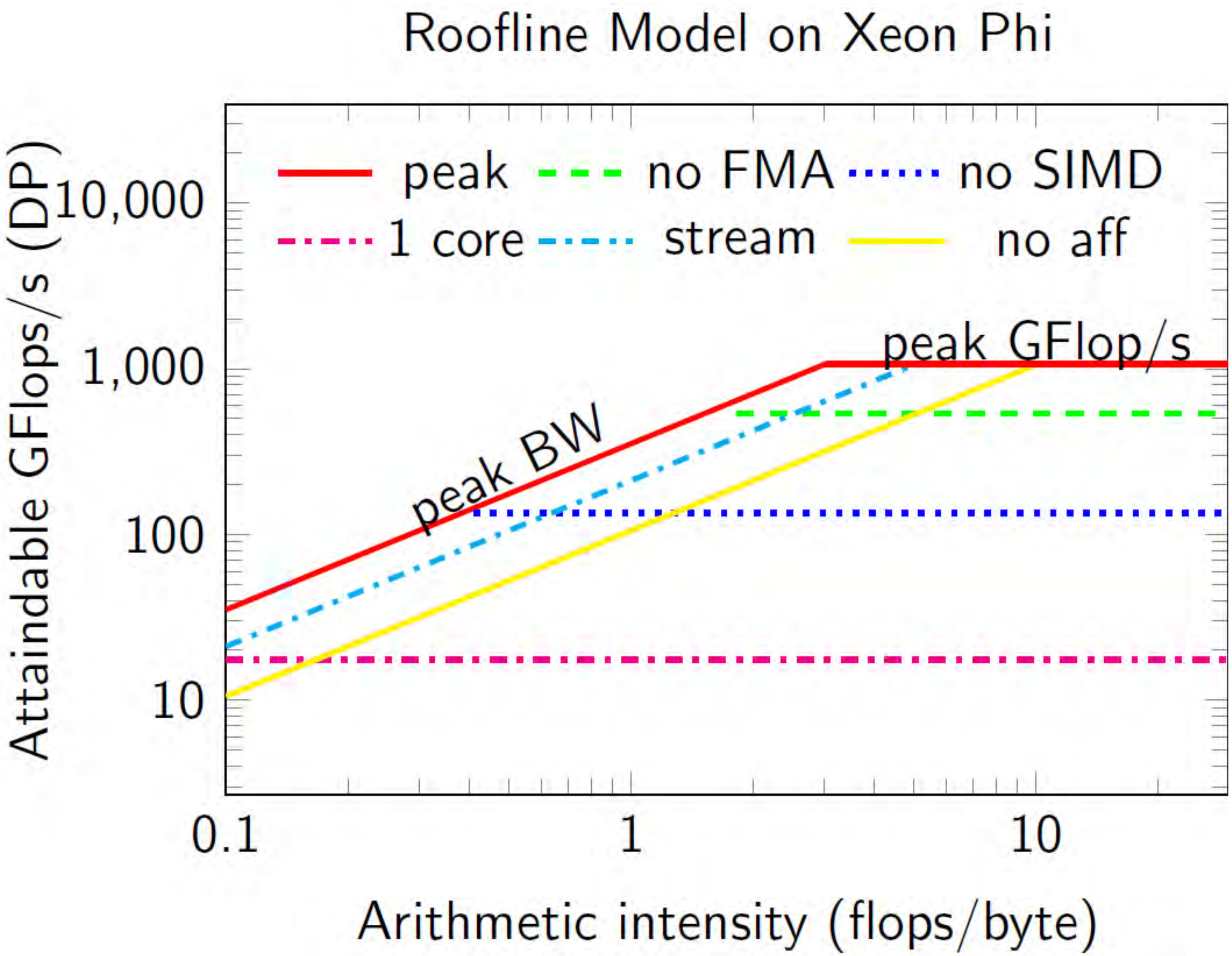
Intel Xeon Phi series	3100	5100	7100
# of Cores	57	60	61
Clock Speed	1.1 GHz	1.053 GHz	1.238 GHz
Max Memory size	6 GB	8 GB	16 GB

Roofline Model on Intel Xeon Phi



Intel Xeon Phi series	3100	5100	7100
# of Cores	57	60	61
Clock Speed	1.1 GHz	1.053 GHz	1.238 GHz
Max Memory size	6 GB	8 GB	16 GB

Roofline Model on Intel Xeon Phi



Intel Xeon Phi series	3100	5100	7100
# of Cores	57	60	61
Clock Speed	1.1 GHz	1.053 GHz	1.238 GHz
Max Memory size	6 GB	8 GB	16 GB

Seismic Structured Grid Simulations on Many-Core Architectures

PDE Solution Techniques

Numerical PDE Solvers

Methods

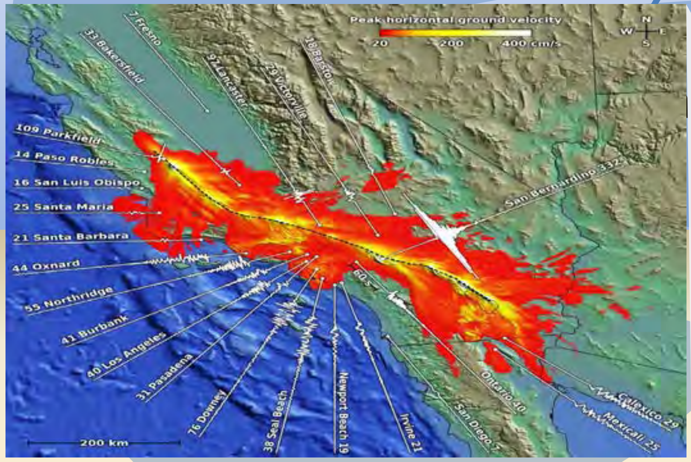
Finite Element Method

Spectral Methods

ce Method

Finite Volume Method

Operators



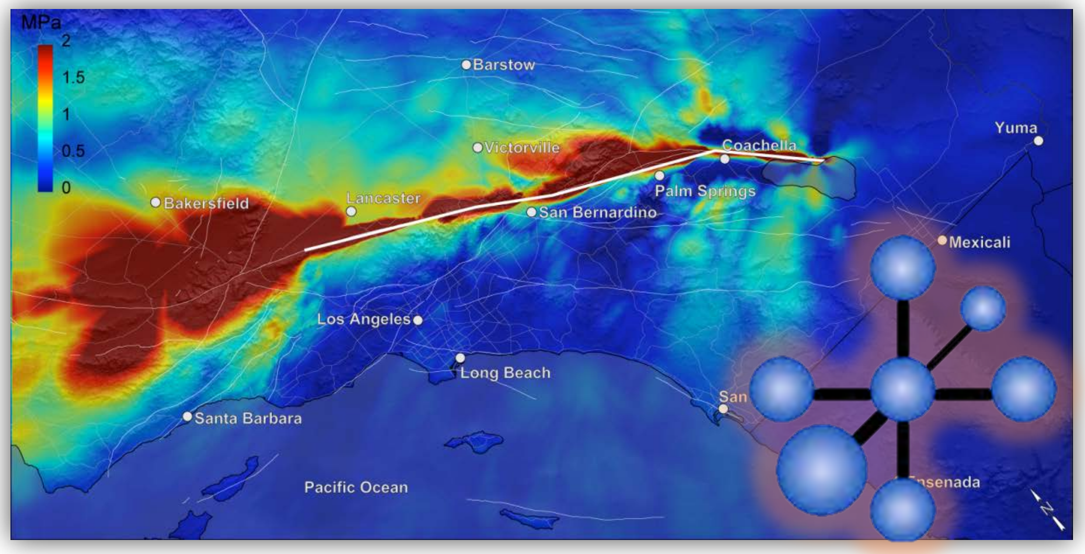
Stencils

FFT

Università della Svizzera italiana

Università della Svizzera italiana | Faculty of Informatics | Institute of Computational Science ICS

AWP-ODC: Earthquakes & Seismic hazard



$$\frac{\partial \dot{\mathbf{u}}}{\partial t} = \rho^{-1}\nabla \cdot \boldsymbol{\sigma}$$

$$\frac{\partial \boldsymbol{\sigma}}{\partial t} = \lambda(\nabla \cdot \dot{\mathbf{u}})\mathbf{I} + \mu(\nabla\dot{\mathbf{u}} + \nabla\dot{\mathbf{u}}^{T})$$

Coulomb failure stress changes in a simulation of an earthquake on the southern San Andreas Fault

Image courtesy: Southern California Earthquake Center

.

- AWP: Scientific modeling code for anelastic waves
- Capable of simulate accurate earthquake wave propagations
- Used to conduct multiple significant SCEC simulations
- 600 x 300 x 80 km domain, 100m resolution, 14.4 billion grids, 50k time steps.
- Gordon Bell finalist (SC 2010), 220 TFlop/s on 223K Jaguar cores

Scalability of the AWP-ODC Stencil-Code on Jaguar

TABLE 2
EVOLUTION OF AWP-ODC

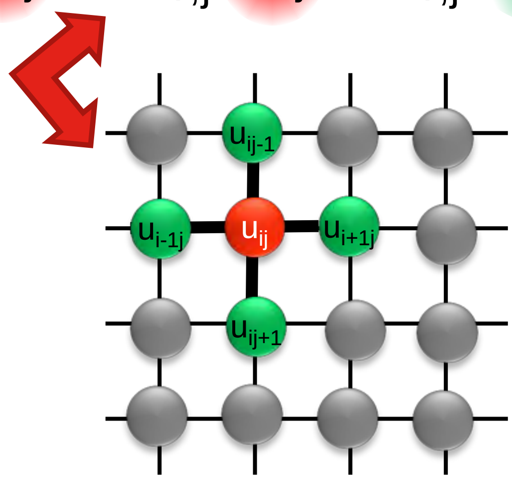
Year	Code version	Simulations	Optimization	SCEC alloc. SUs	Sustain. Tflop/s
2004	1.0	TeraShake-K	MPI tuning	0.5M	0.04
2005	2.0	TeraShake-D	I/O tuning	1.4M	0.68
2006	3.0	PN MQuake	partition. mesh	1.0M	1.44
2007	4.0	ShakeOut-K	incorp. SGSN	15M	7.29
2008	5.0	ShakeOut-D	asynchronous	27M	49.9
2009	6.0	W2W	single CPU opt	32M	86.7
2010	7.0 7.1 7.2	M8	overlap cache blocking reduced comm	61M	220

Highly scalable AWP-ODC code: **220 TFlop/s sustained** on **220k cores** (Jaguar)

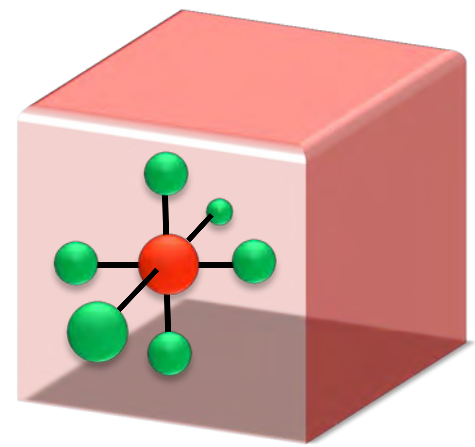
What Is a Stencil?

- Weighted sum of subset of neighbors of a grid point

$$u_{i,j} \leftarrow c^{(0)}_{i,j} u_{i,j} + c^{(1)}_{i,j} u_{i-1,j} + c^{(2)}_{i,j} u_{i+1,j} + c^{(3)}_{i,j} u_{i,j-1} + c^{(4)}_{i,j} u_{i,j+1}$$



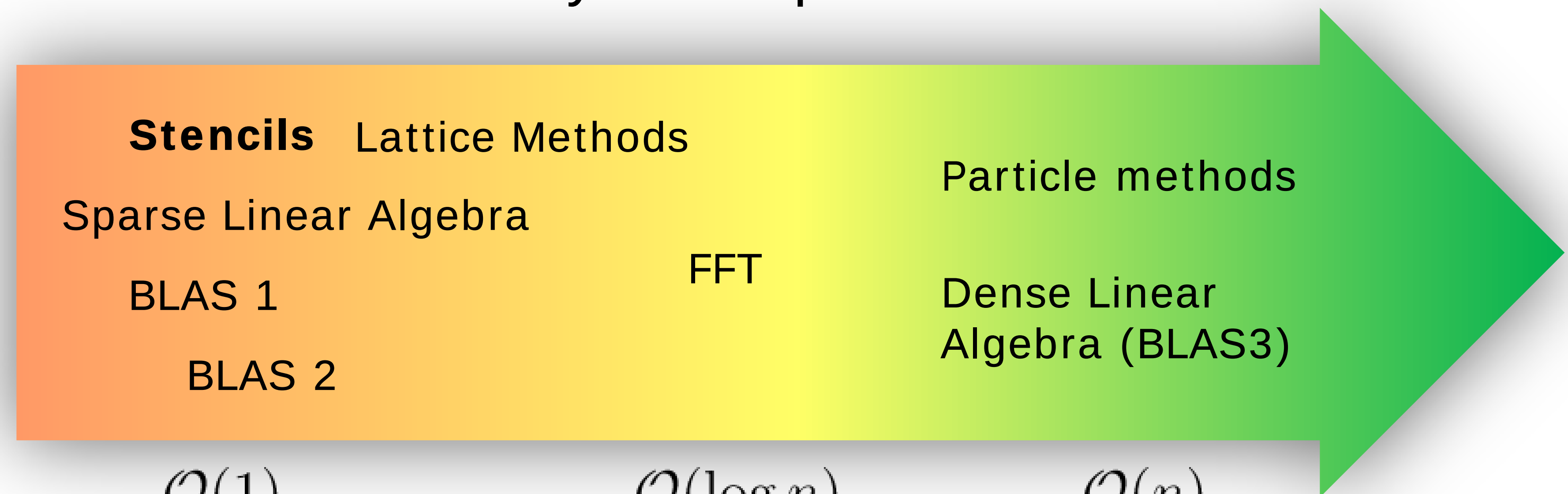
5-point stencil in a regular 2D grid



7-point stencil in 3D

Challenge: Arithmetic Intensity

Arithmetic Intensity := Flops / Transferred Data

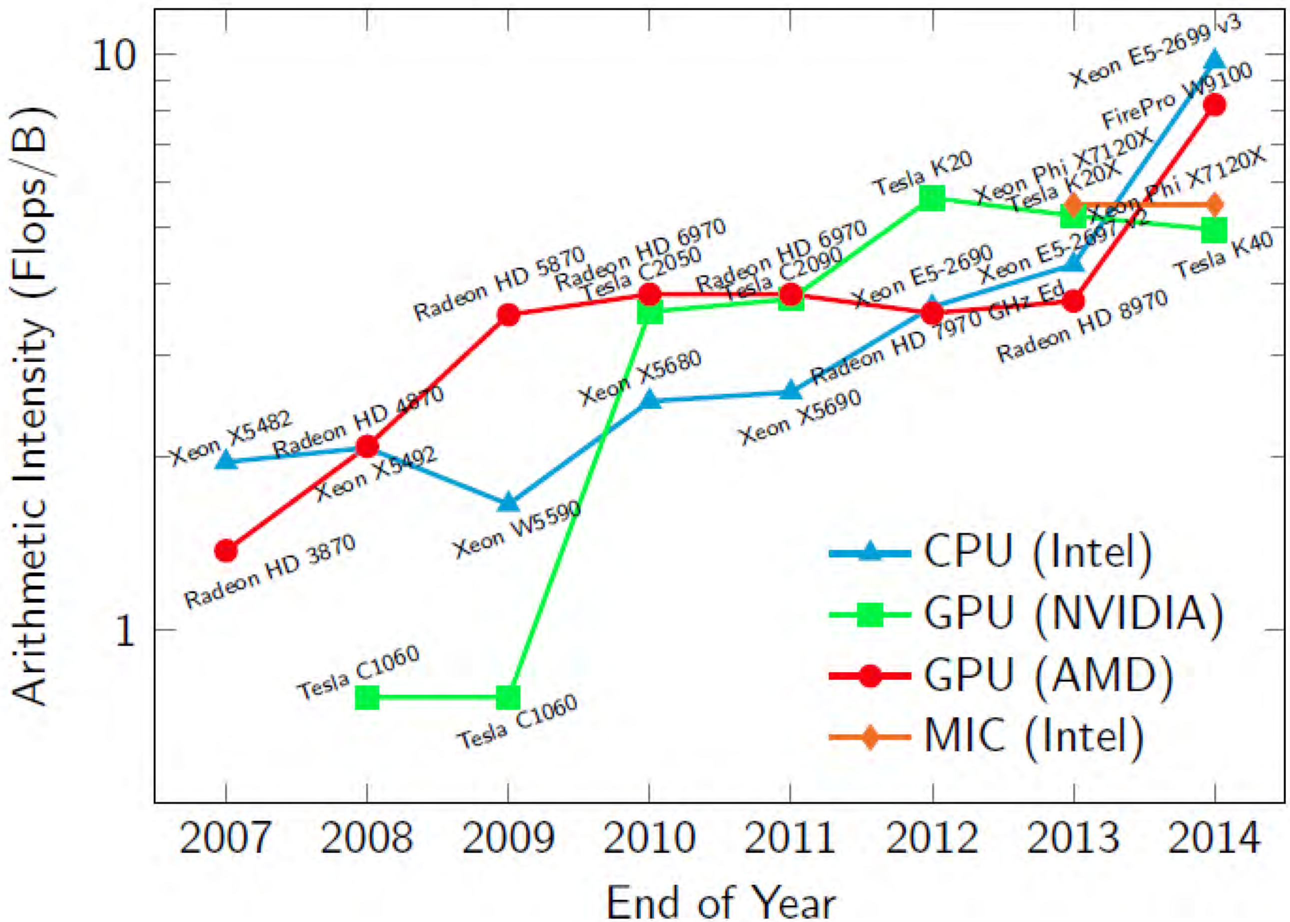


$\mathcal{O}(1)$ $\mathcal{O}(\log n)$ $\mathcal{O}(n)$

Low arithmetic intensity $\Rightarrow$ memory bandwidth bound

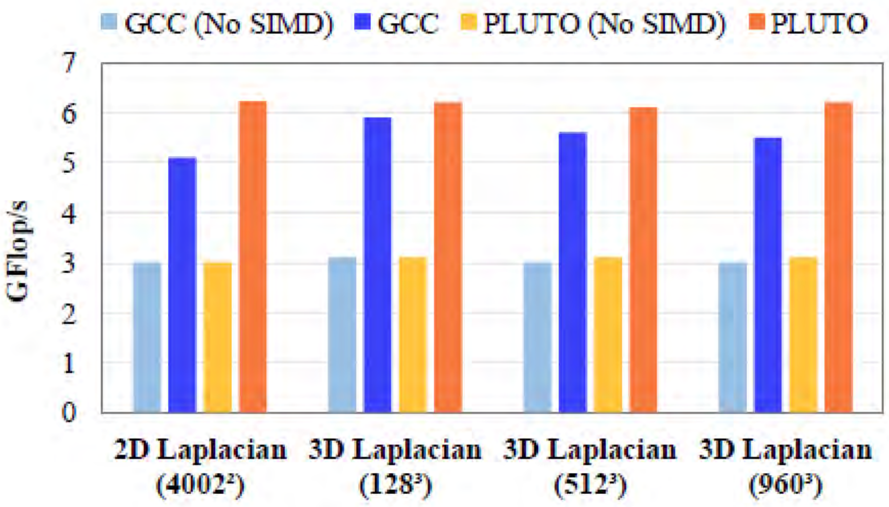
High arithmetic intensity $\Rightarrow$ processor bound

Arithmetic Intensity



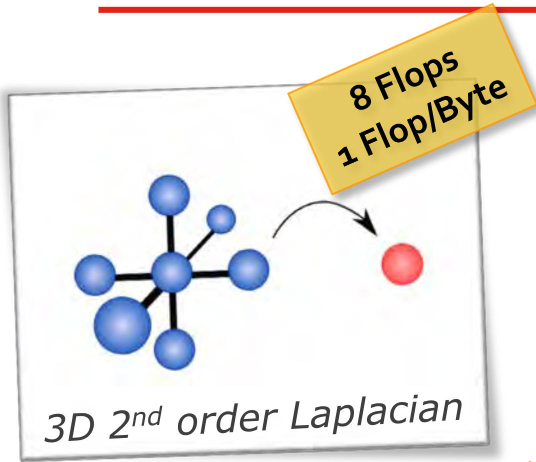
- Arithmetic intensity to fully utilize the floating-point capabilities of HPC units on recent microarchitectures

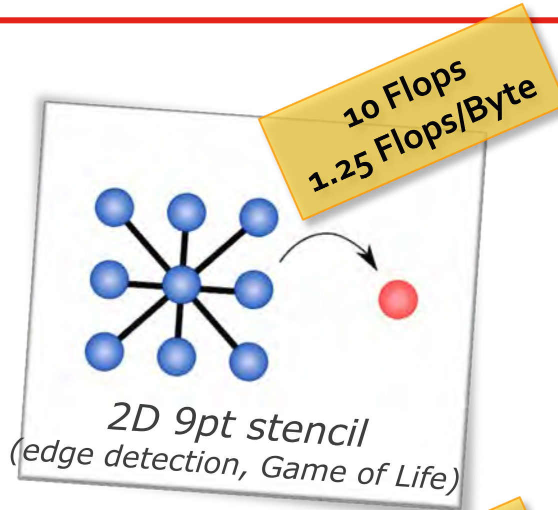
Impact of AVX vectorization

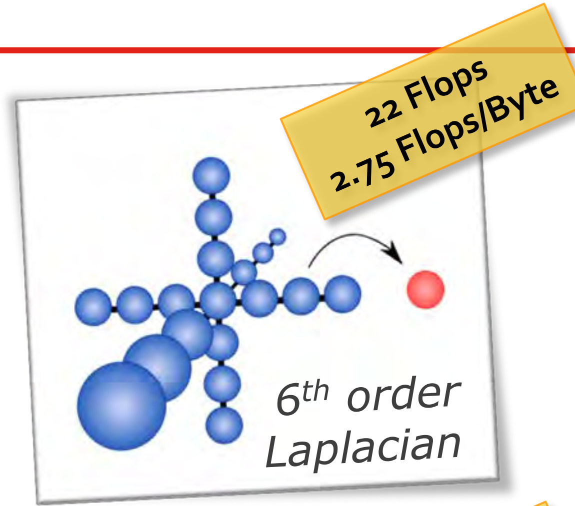


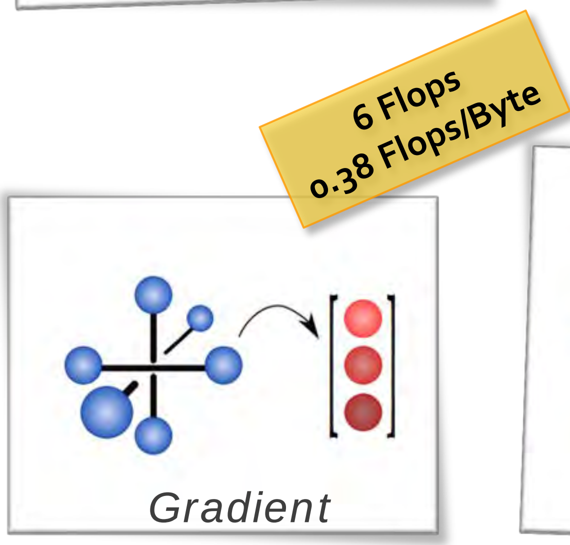
- Impact of AVX vectorization for various stencils using 1 (on the left) and 10 cores (on the right) on one socket of an Intel Xeon 2660v2 IvyBridge.

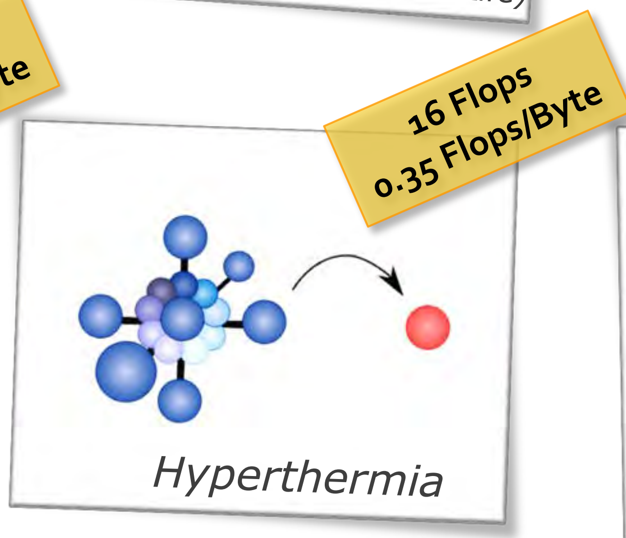
Stencil Variety

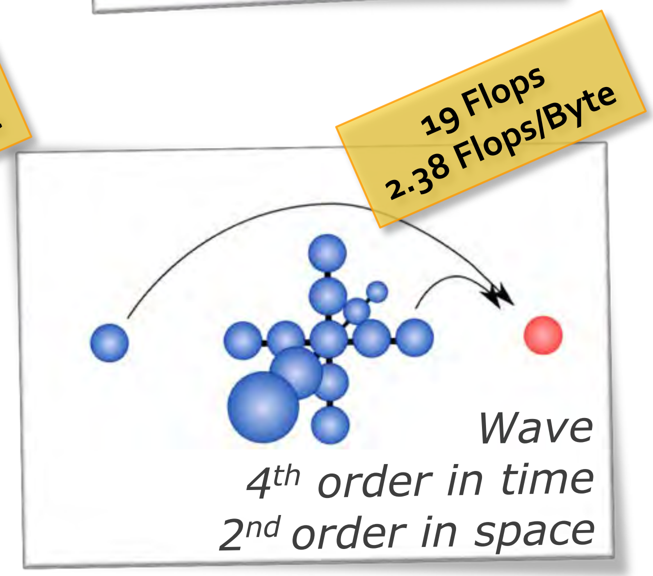










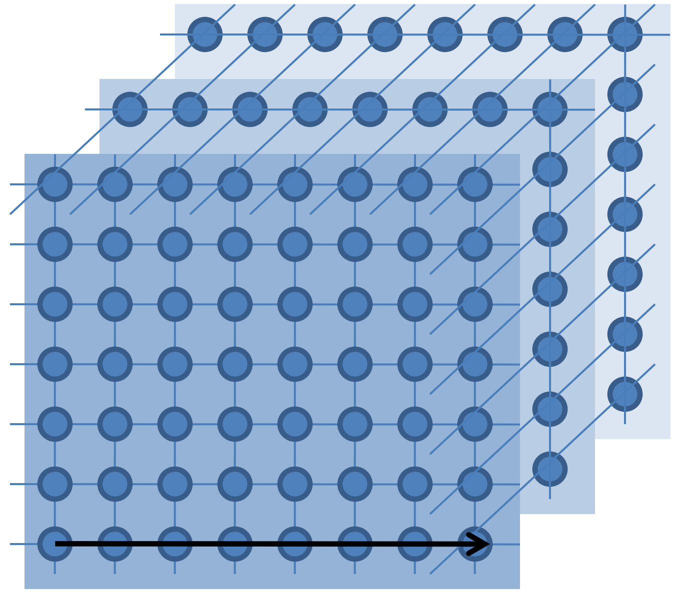


Reduce Memory Traffic on Multi-/Manycores

- Stencil performance usually limited by memory bandwidth
- **Goal**: Increase performance by minimizing memory traffic
  - Even more important for many/multicore!

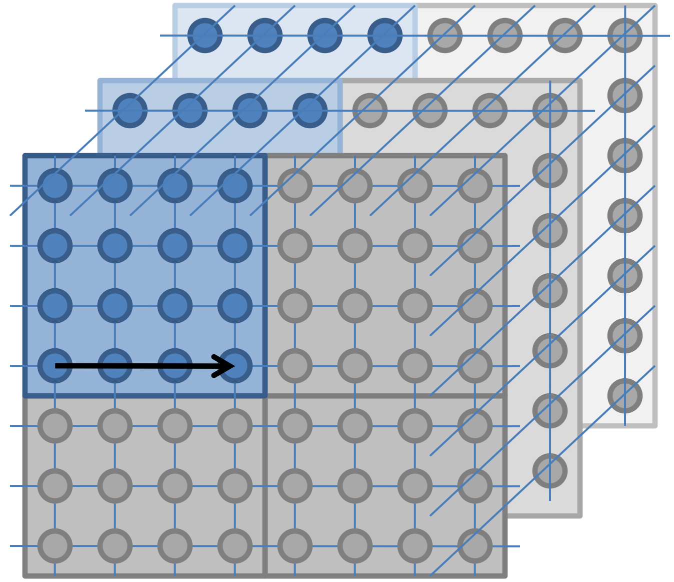
- Concentrate on getting reuse both:
    - within an iteration (**spatial blocking**)
    - across time iterations (**temporal blocking**, Ax, A^2x, ..., A^kx)

Naïve Grid Traversal



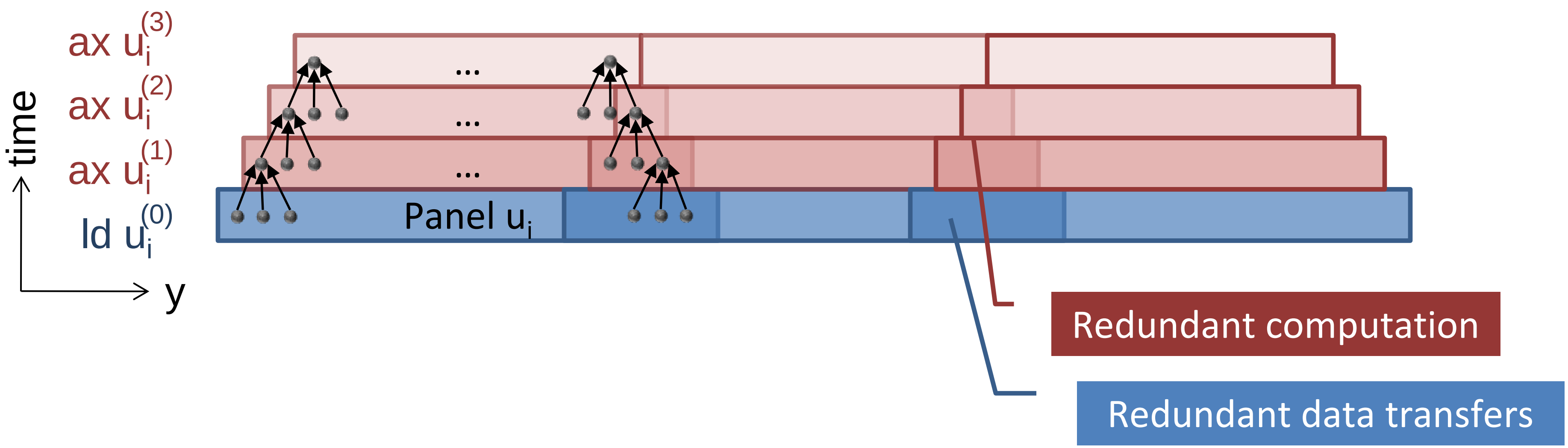
- Traverse the grid in the “usual” way
- Locality not exploited
- Performance will suffer if grid doesn’t fit into cache

Spatial Cache Blocking – Reuse data in space



- 3D block partitioning
- Reuse data within an iteration

Temporal Cache Blocking – Reuse data in time

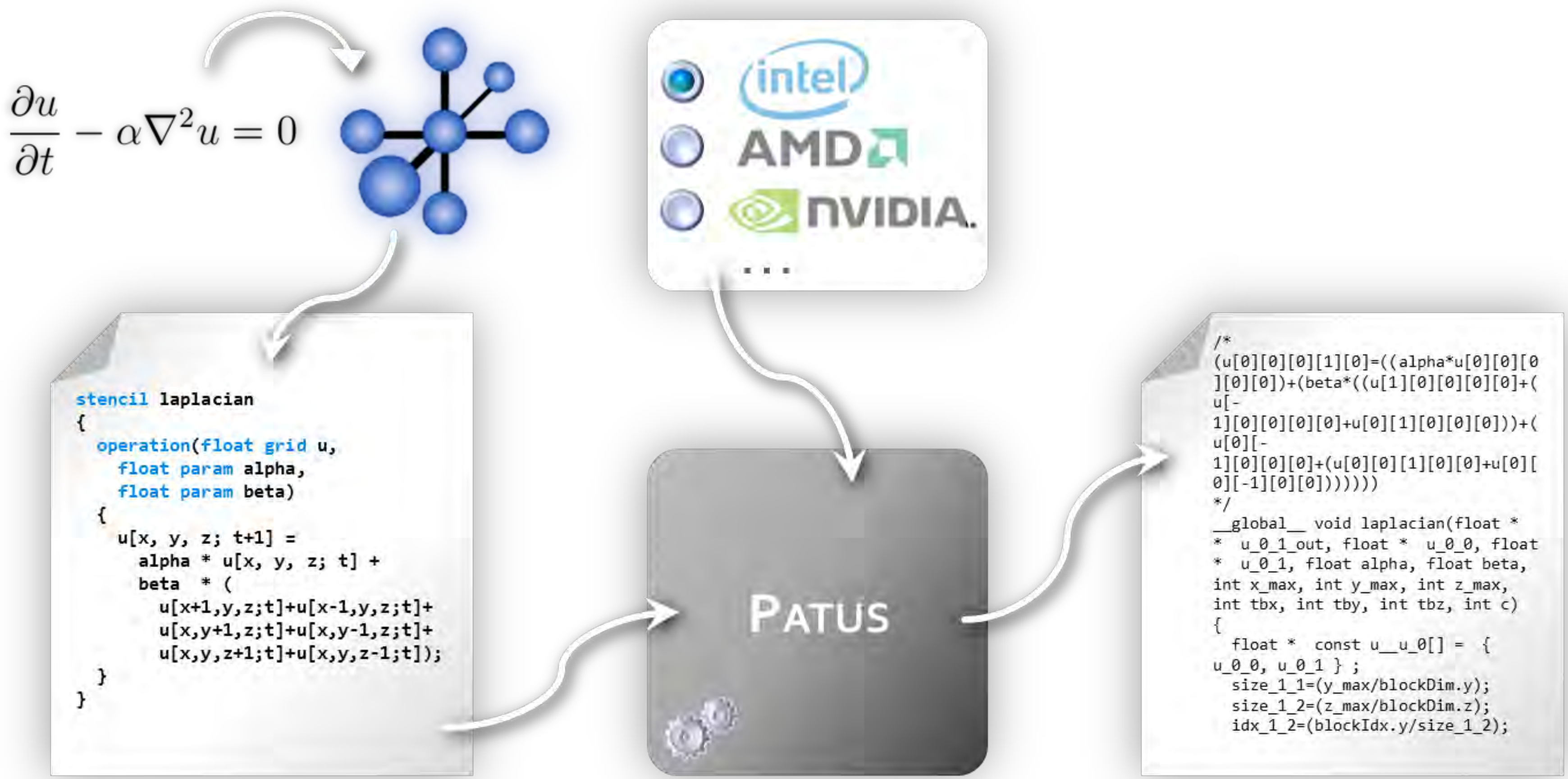


Sizes of the panels shrink with each stencil sweep

(-) Redundant computation and data transfers

(+) Easily parallelizable along the horizontal (y) axis

PATUS: Parallel Autotunic of Stencil Codes



PATUS: Code Optimization Techniques

- **NUMA optimization (NUMA-aware data initialization)**
- **Cache blocking, block parallelization**
- **Explicit vectorization**
- **Loop unrolling**
- **Inline assembly**
  - Efficient index calculations
  - Register reuse
  - Software prefetching
  - Optimal instruction scheduling
- **Auto-tuning**

Specifying a Stencil

Code Optimization Techniques

- NUMA optimization (NUMA-aware data initialization)
- Explicit Vectorization
- Cache and time blocking, block parallelization
- Loop unrolling
- **Inline assembly code is generated**
- Auto-tuning

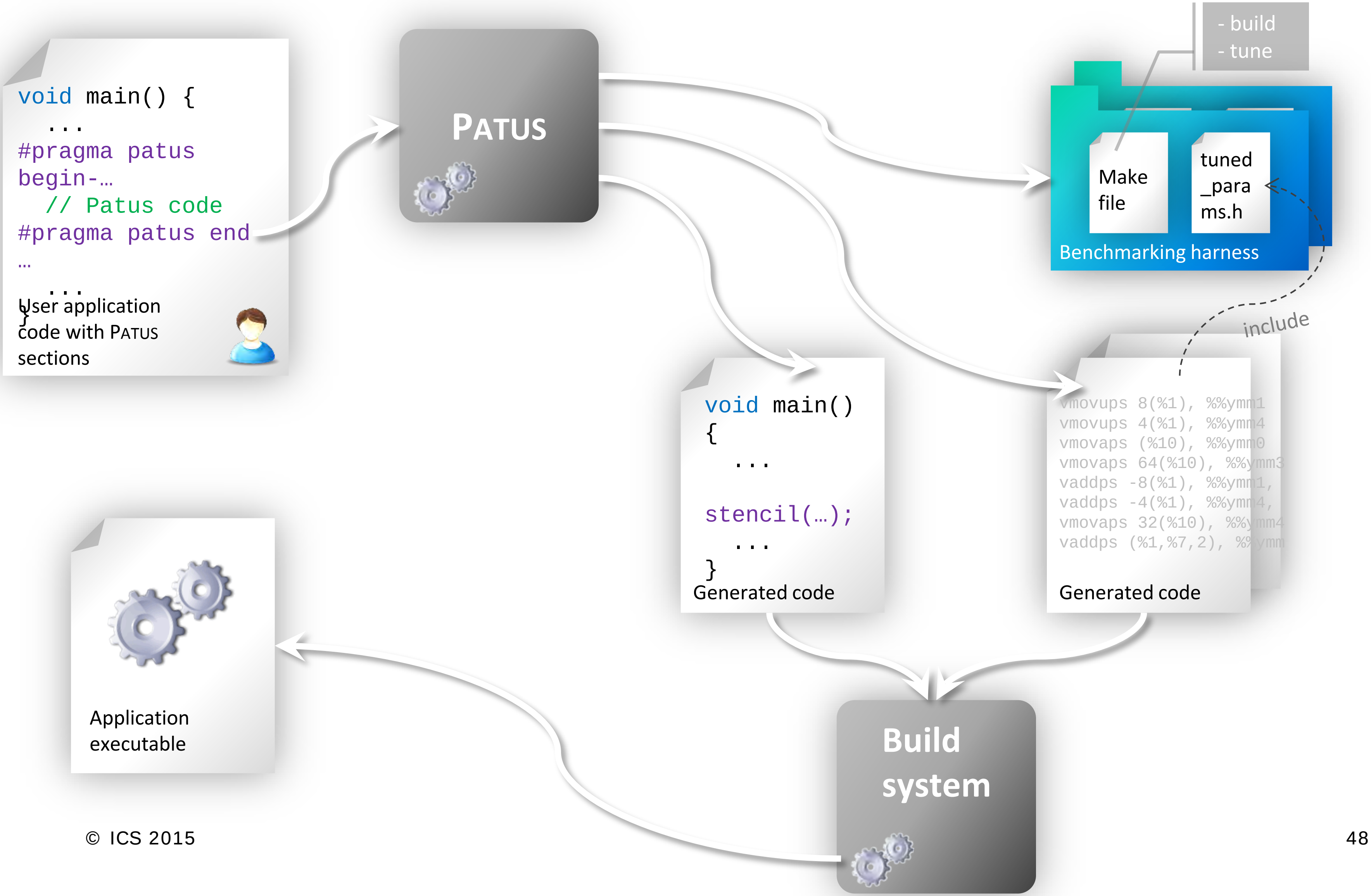
```
for (p3_idx_z=v2_idx_z; p3_idx_z<v2_idx_z_max; p3_idx_z+=1)
{
 for (p3_idx_y=v2_idx_y; p3_idx_y<(v2_idx_y_max-1);
p3_idx_y+=2)
 {
  p3_idx_x=v2_idx_x;
  _idx0=((x_max*((y_max*p3_idx_z)+p3_idx_y))+p3_idx_x);
  __asm__ __volatile__ (
   "mov %2, %%rax\n\t"
   "add $31, %%rax\n\t"
   "and $31, %%rax\n\t"
   "sub $32, %%rax\n\t"
   "neg %%rax\n\t"
   "shr $2, %%rax\n\t"
   "cmp %%rax, %11\n\t"
   "cmovng %11, %%rax\n\t"
   "mov %%rax, %%rbx\n\t"
   "or %%rax, %%rax\n\t"
   "jz 1f\n\t"
   "vmovups 4(%1), %%ymm1\n\t"
   "vmovups 32(%10), %%ymm0\n\t"
   "vaddps -4(%1), %%ymm1, %%ymm3\n\t"
   "vaddps (%1,%7), %%ymm3, %%ymm3\n\t"
   "vaddps (%1,%9), %%ymm3, %%ymm3\n\t"
   "vaddps (%1,%8), %%ymm3, %%ymm3\n\t"
   "vmovups 64(%10), %%ymm1\n\t"
   "vmovups 8(%1), %%ymm2\n\t"
   "vaddps (%1,%6), %%ymm3, %%ymm3\n\t"
   "vmulps %%ymm0, %%ymm3, %%ymm0\n\t"
   ...
   "vmovups %%ymm2, (%2)\n\t"
   "shl $2, %%rax\n\t"
   "addq %%rax, %0\n\t"
   "addq %%rax, %1\n\t"
   "addq %%rax, %2\n\t"
...
```

prologue loop header

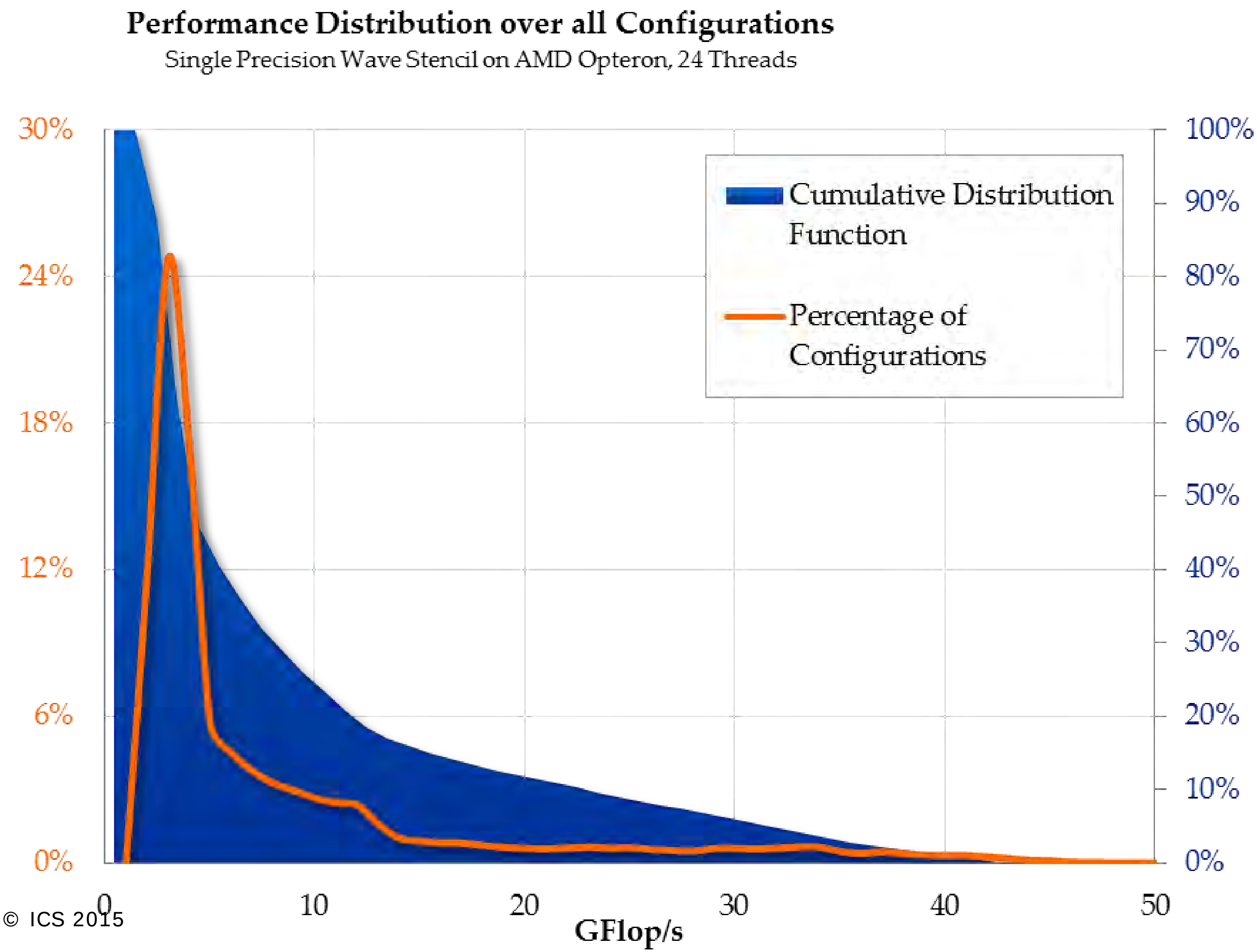
stencil comp. (AVX)

next grid pt

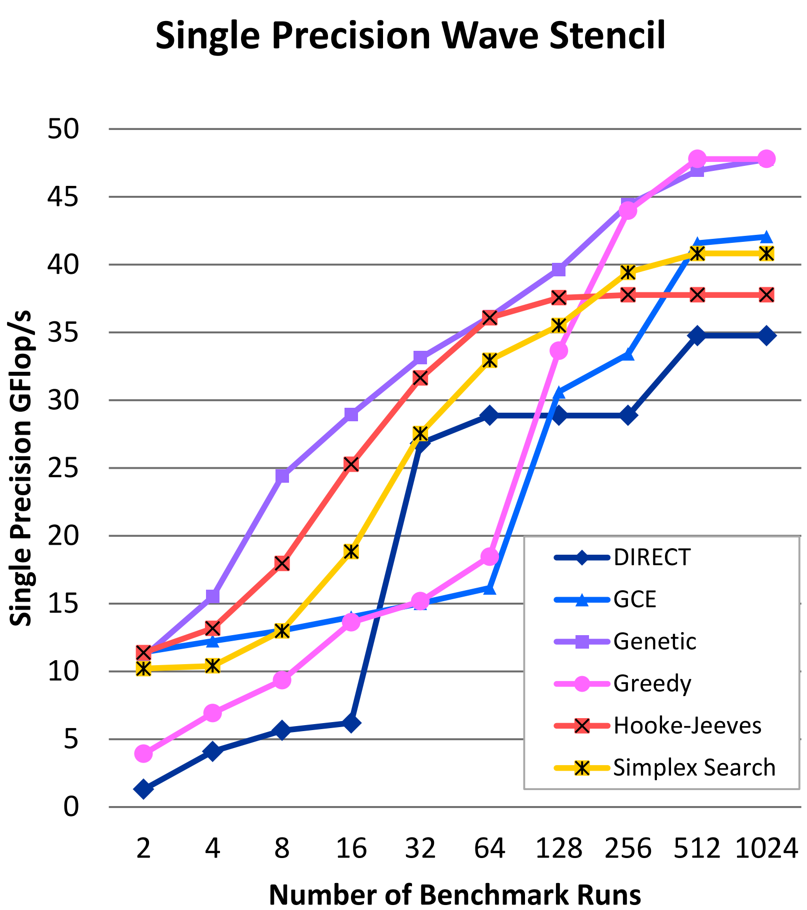
Integration & Application-Specific Tuning

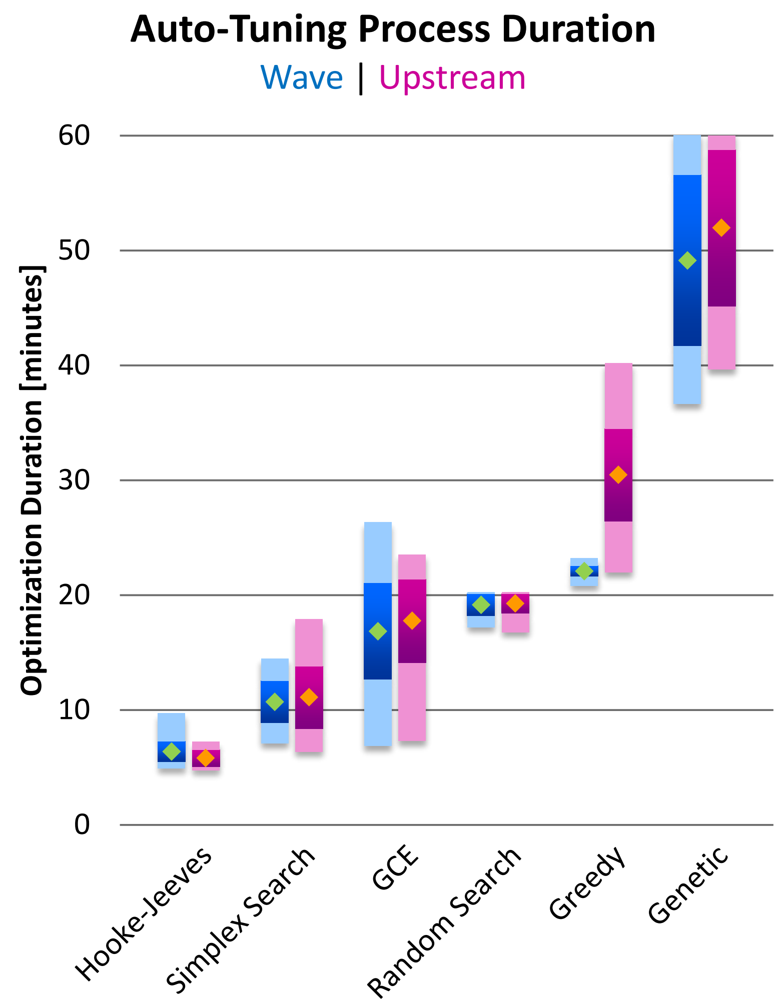


Auto-Tuning (Single-Precision Wave Stencil)

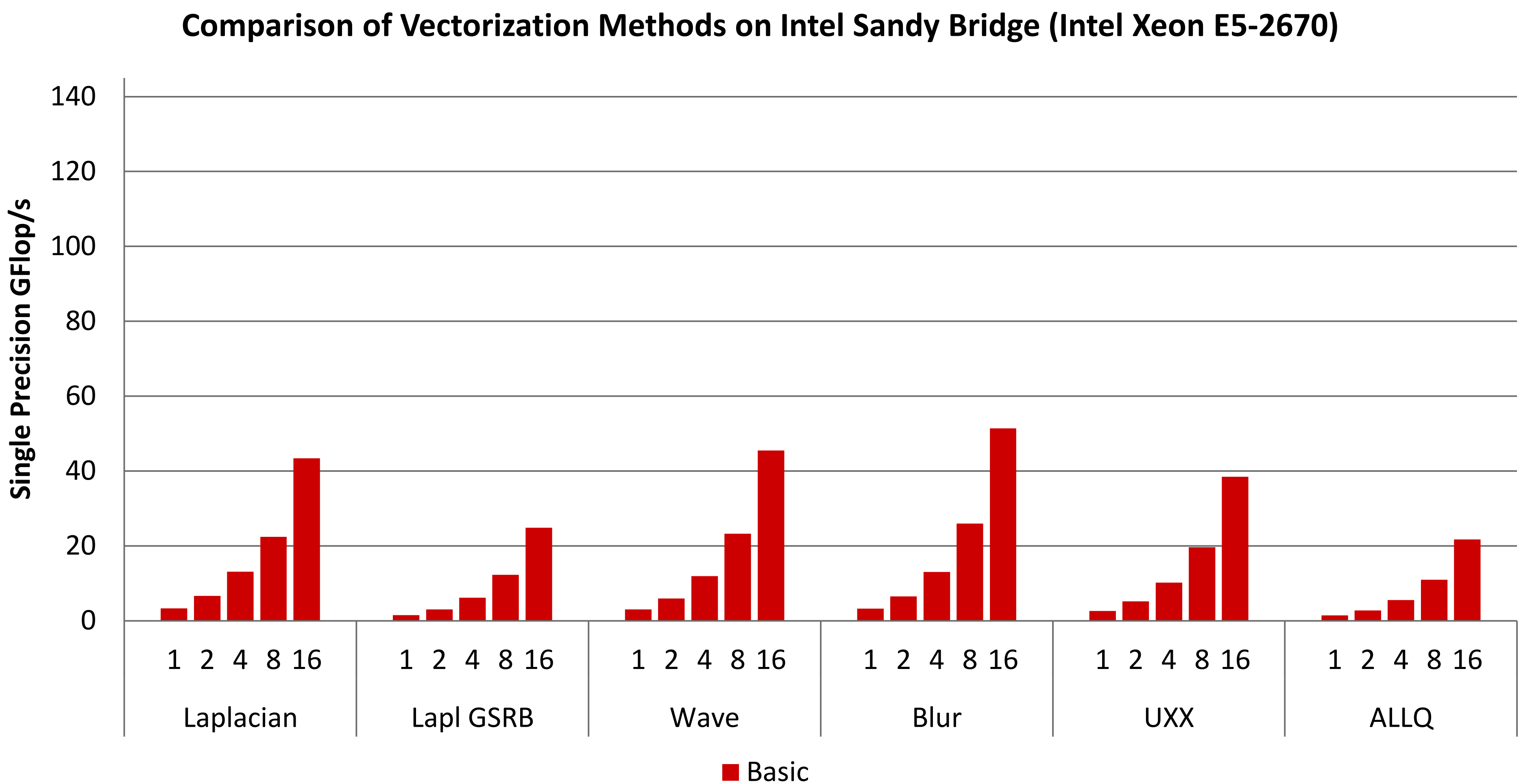


Search Methods (Single-Precision Wave Stencil)

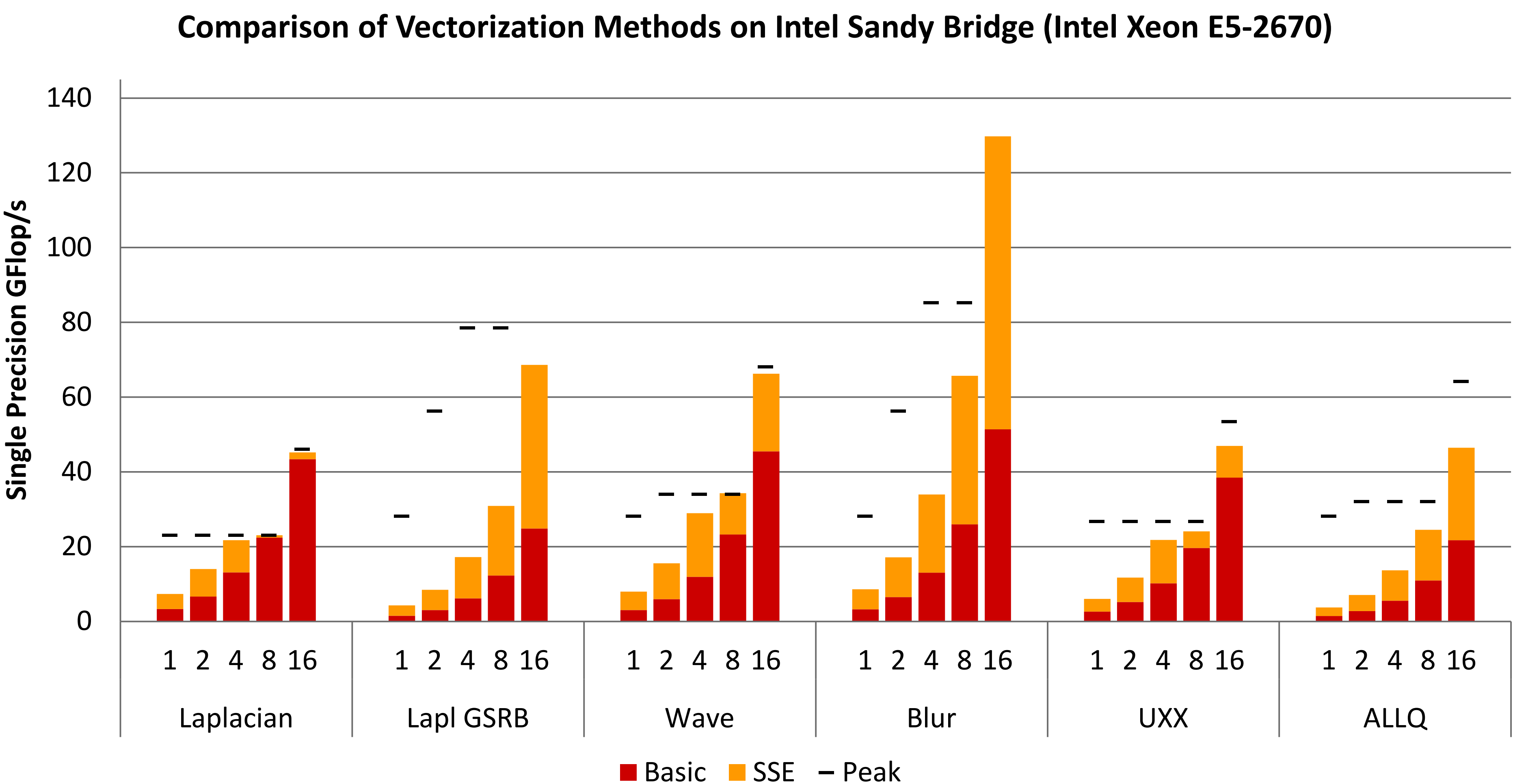




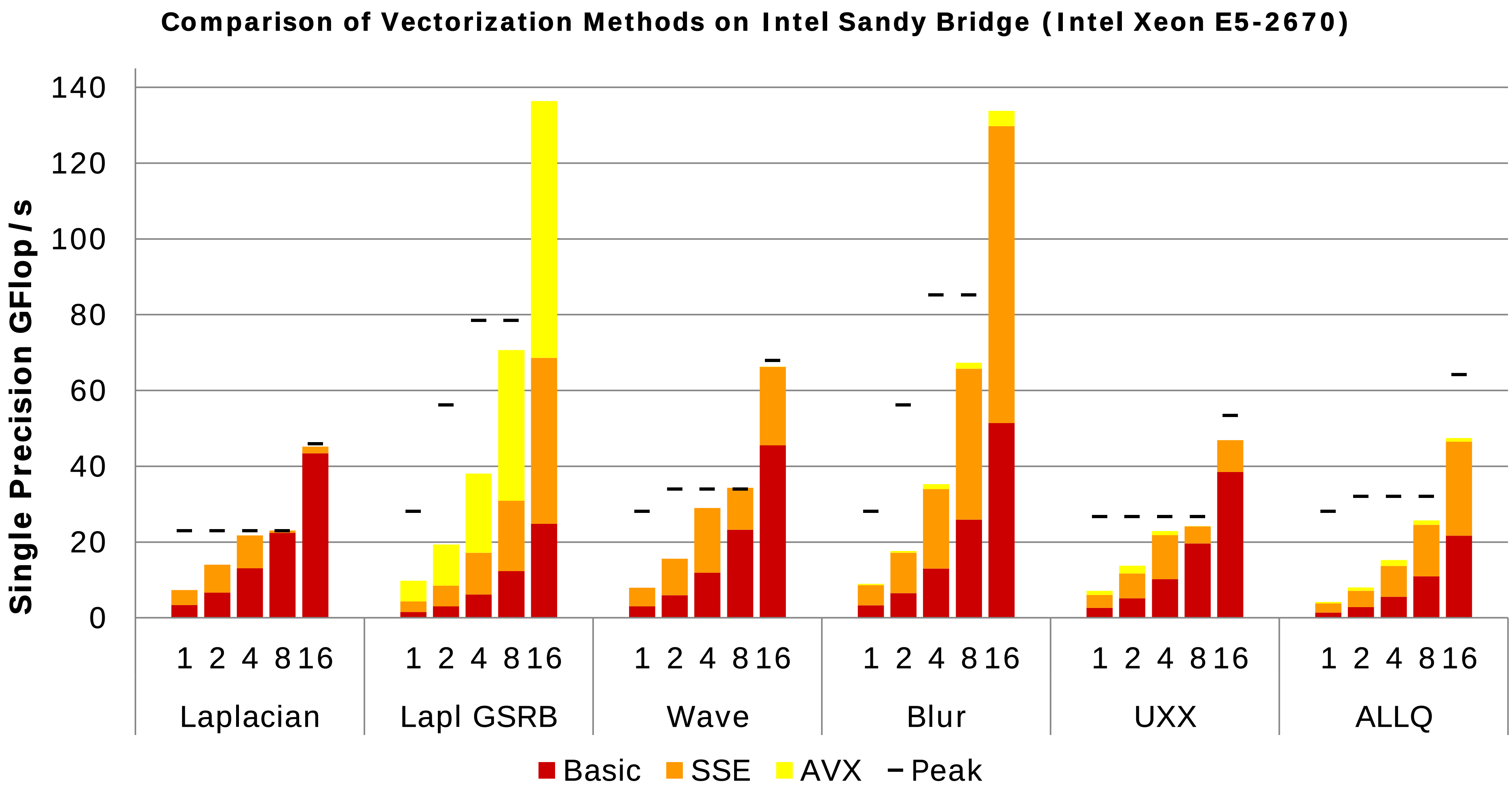
Stencil Kernel Benchmarks



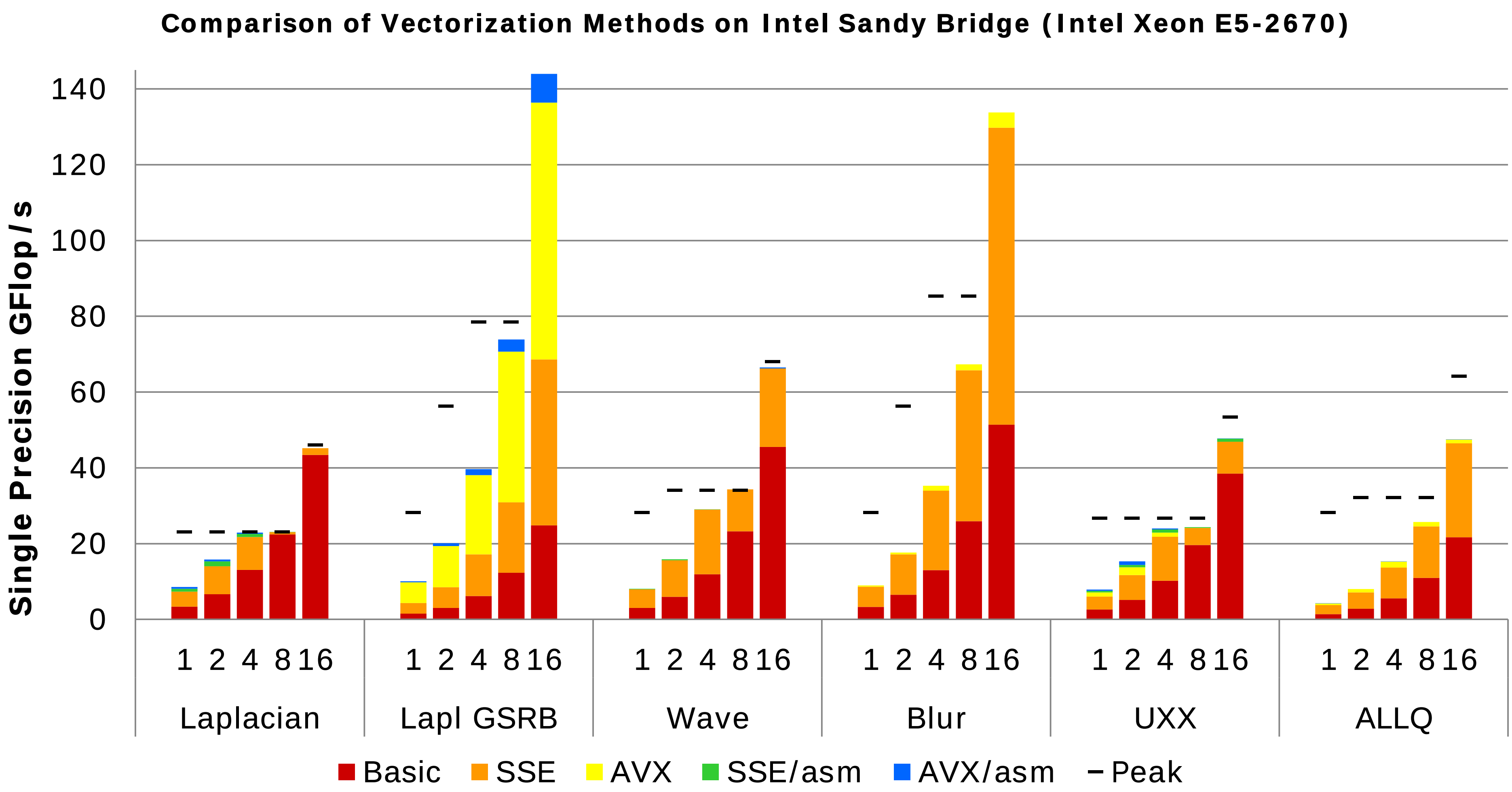
Stencil Kernel Benchmarks



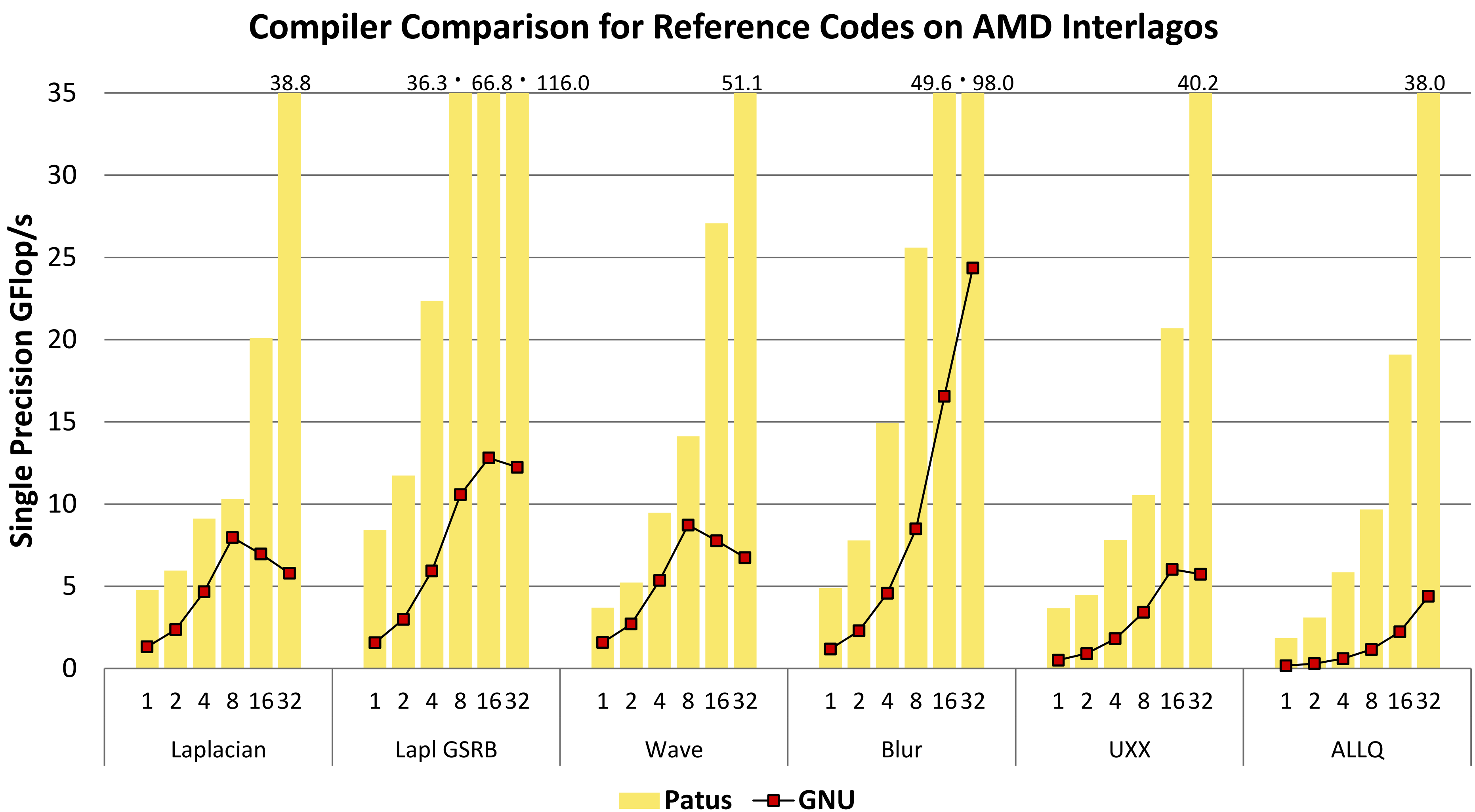
Stencil Kernel Benchmarks



Stencil Kernel Benchmarks

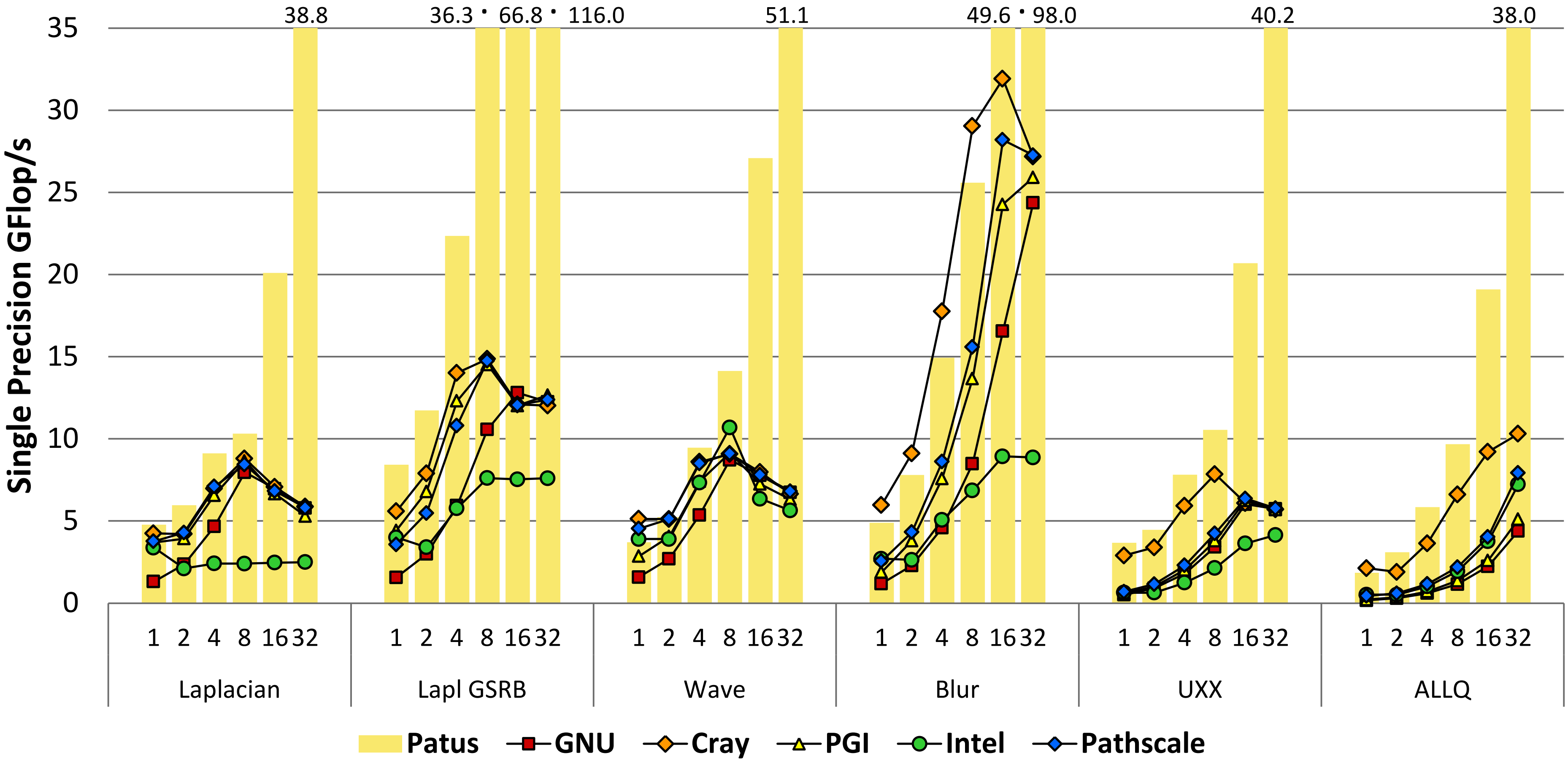


Compiler Optimization Comparison



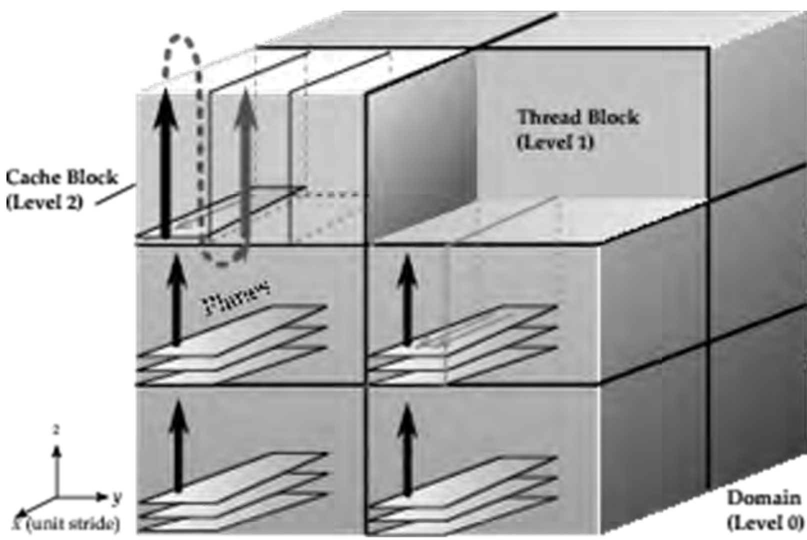
Compiler Optimization Comparison

Compiler Comparison for Reference Codes on AMD Interlagos

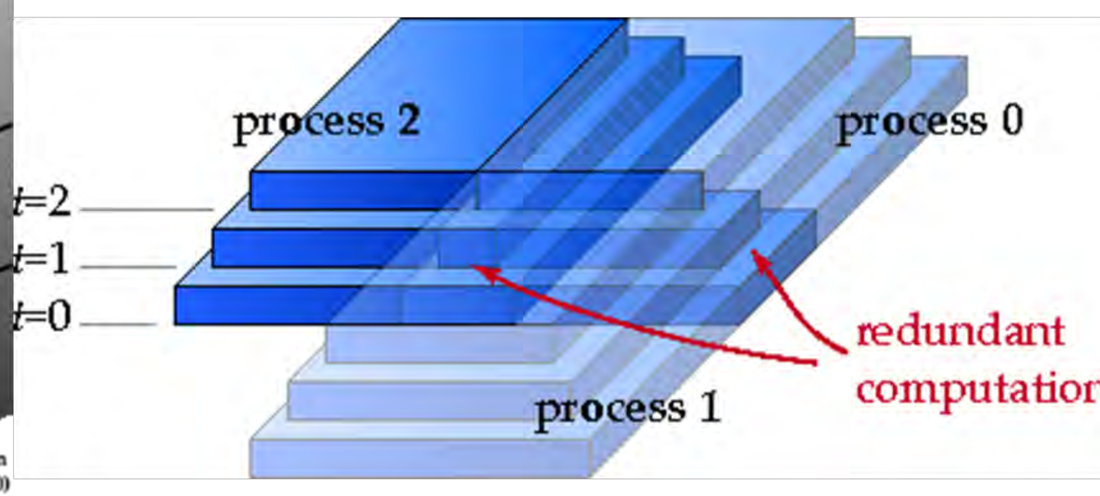


Earthquakes&seismic hazard / AWP-ODC Stencil Kernels

Kernel	Description	Discretization	Flops/Stencil	Arith. Intens.
uxx1	Velocity in one direction	4th order	20 Flops	0.83 Flop/Byte
xy1	Diagonal stress in one direction	4th order	16 Flops	0.80 Flop/Byte
xyz1	Stresses parallel to axes	4th order	90 Flops	2.04 Flop/Byte
xyzq	Stresses parallel to axes in viscous mode	4th order	129 Flops	1.61 Flop/Byte

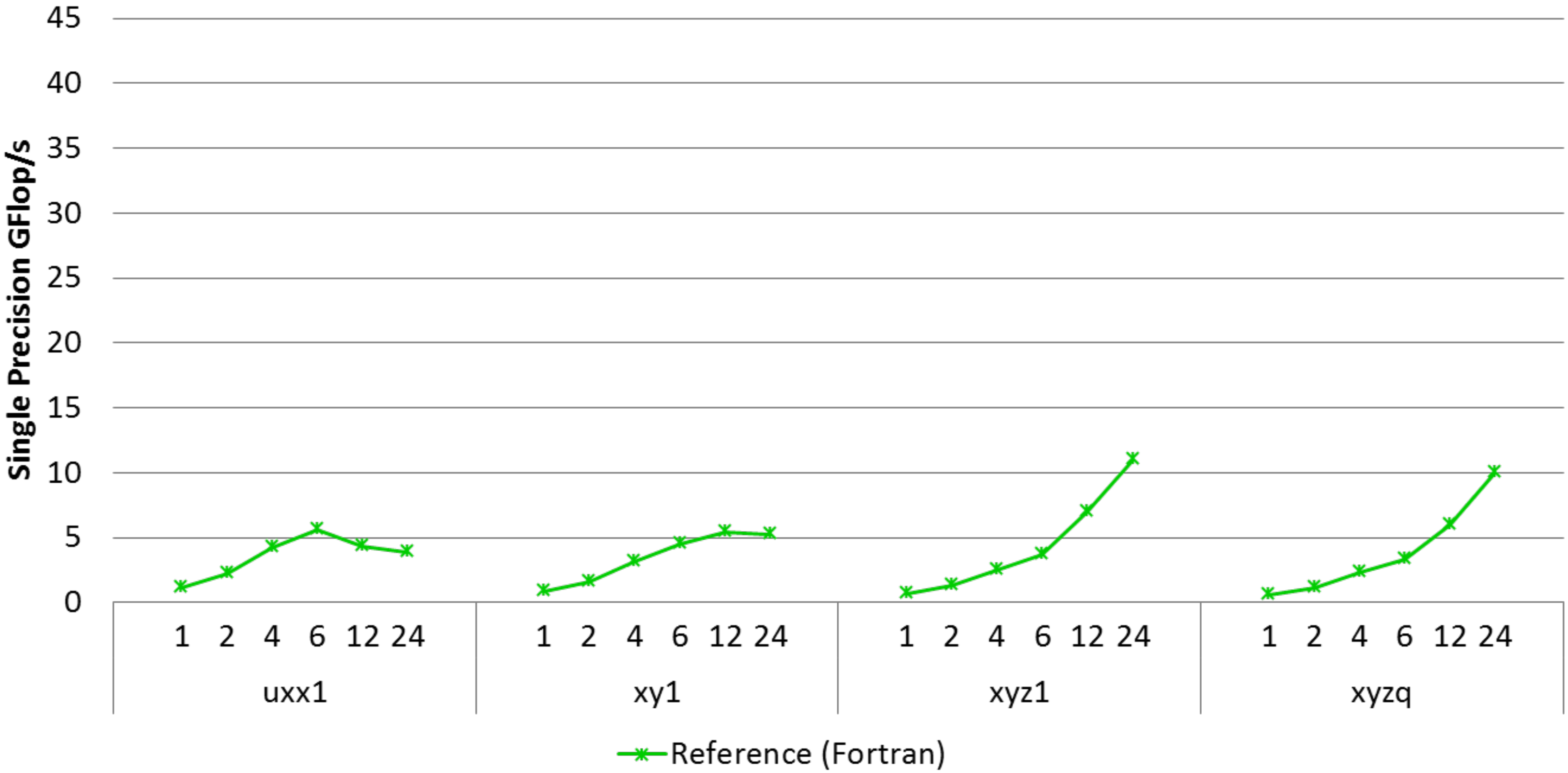


spatial blocking



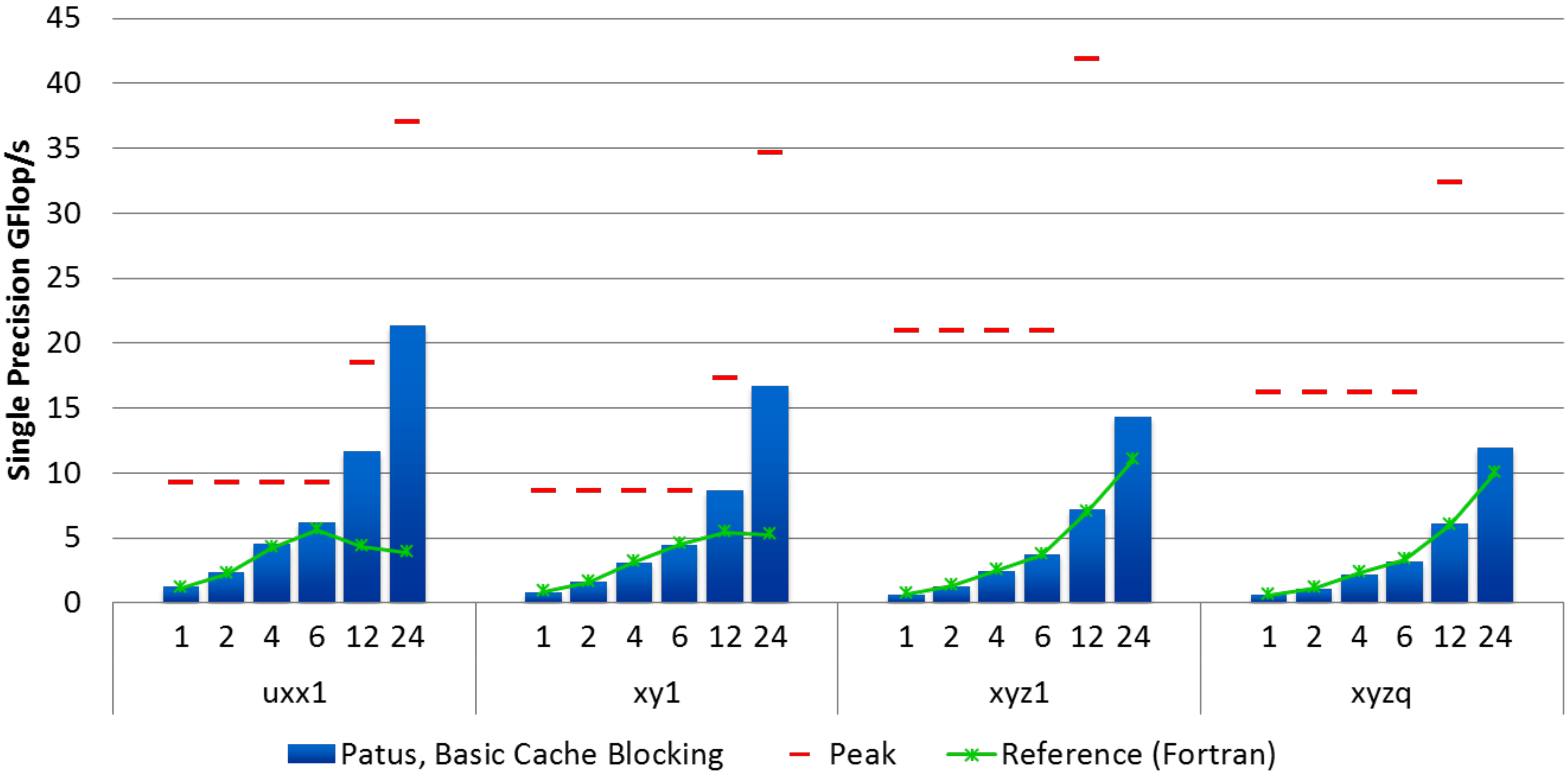
temporal blocking, $Ax, A^2x, ..., A^kx$

Performance Benchmarks AWP-ODC Code on AMD Opteron "Magny Cours"



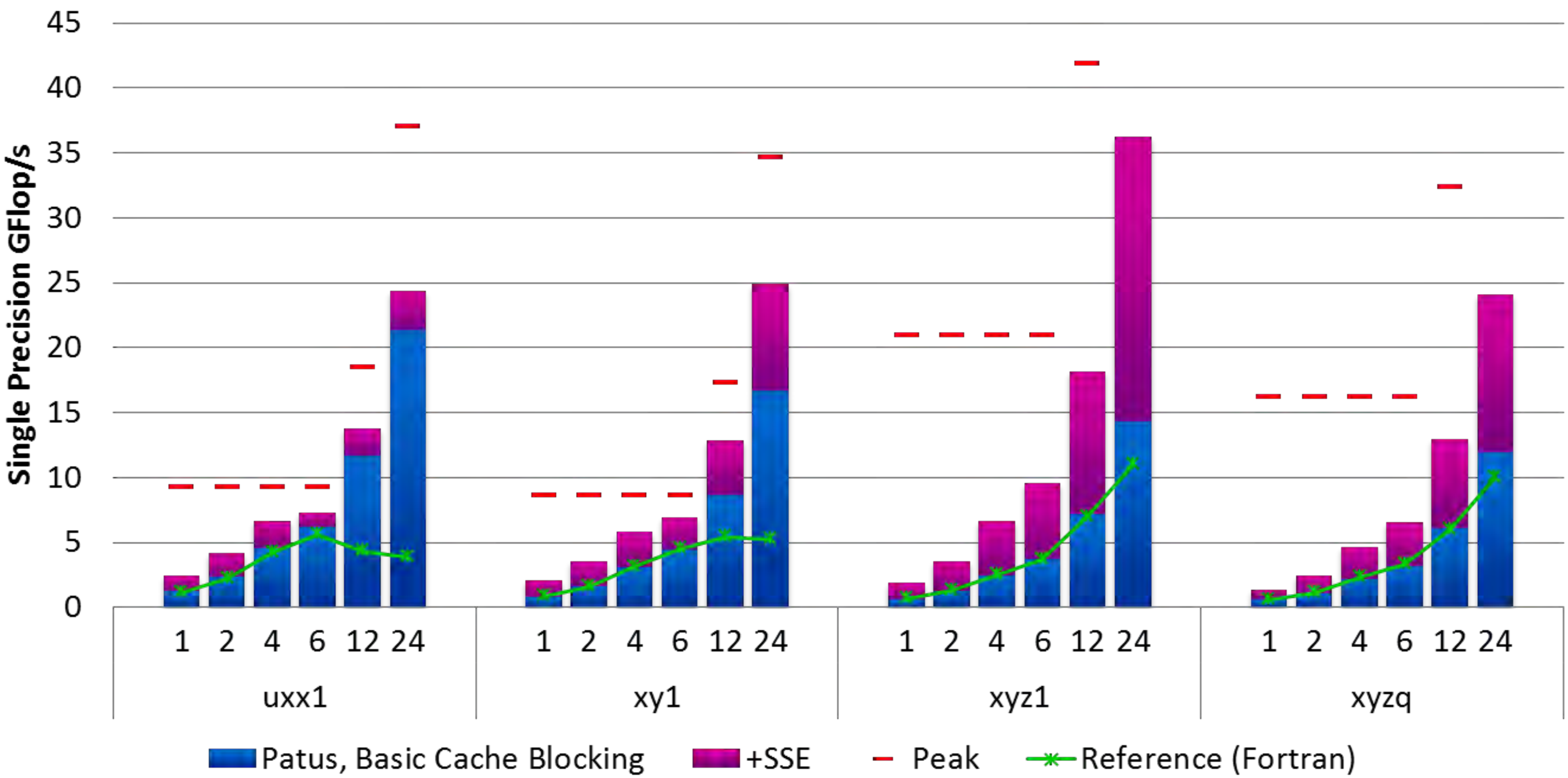
M. Christen, O. Schenk et al., *PATUS: A Code Generation and Autotuning Framework For Parallel Iterative Stencil Computations on Modern Microarchitectures*, SC12, IPDPS 2009, IPDPS 2010, IPDPS 2011

Performance Benchmarks AWP Code on AMD Opteron “Magny Cours”



M. Christen, O. Schenk et al., *PATUS: A Code Generation and Autotuning Framework For Parallel Iterative Stencil Computations on Modern Microarchitectures*, IPDPS 2009, IPDPS 2010, IPDPS 2011

Performance Benchmarks AWP Code on AMD Opteron "Magny Cours"



M. Christen, O. Schenk et al., *PATUS: A Code Generation and Autotuning Framework For Parallel Iterative Stencil Computations on Modern Microarchitectures*, IPDPS 2009, IPDPS 2010, IPDPS 2011

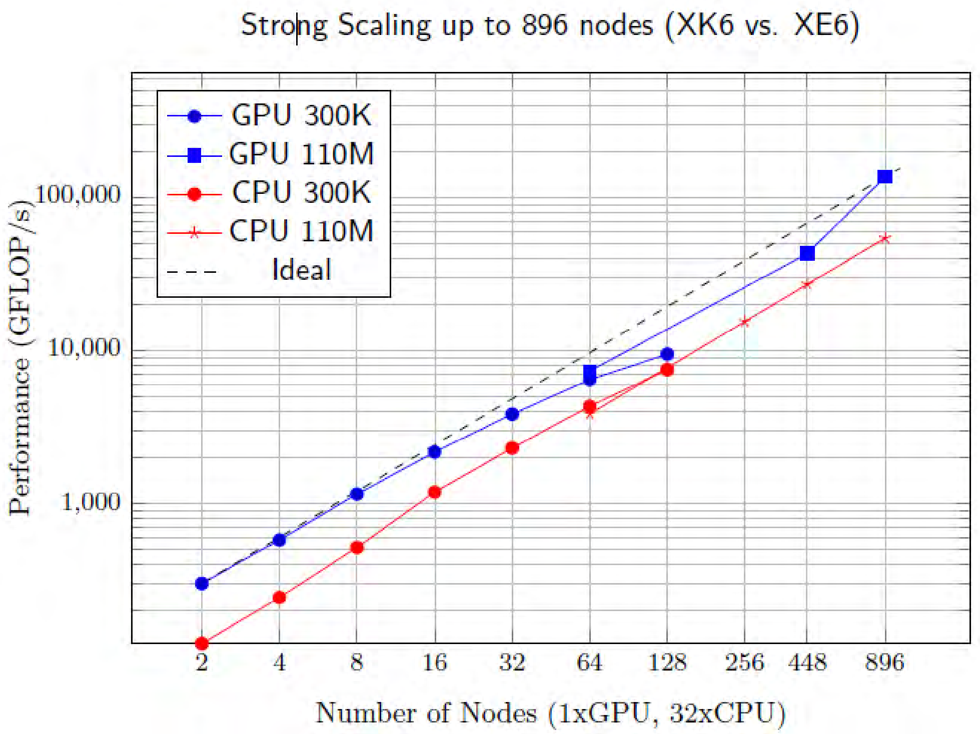
Roofline Performance Model For Unstructured Grids in Seismic Simulations

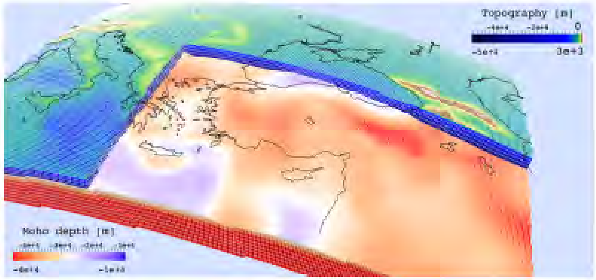
PASC Project: GPU Version of SPECFEM3D

- **SPECFEM3D** is a higher-order finite element code that simulates elastic/acoustic waves on arbitrary hexahedral meshes.
- Written using Fortran90 and MPI
- Excellent performance and scalability (> 90%)
- Large user community across large array of applications
- **GPU Code** for **forward** and **adjoint** system with SPECFEM3D.
- Seismic imaging via adjoint methods

HP2C: Strong Scaling for Case Study Turkey Earthquakes

- 19M mesh covering Europe, Middle East/Northern Africa.



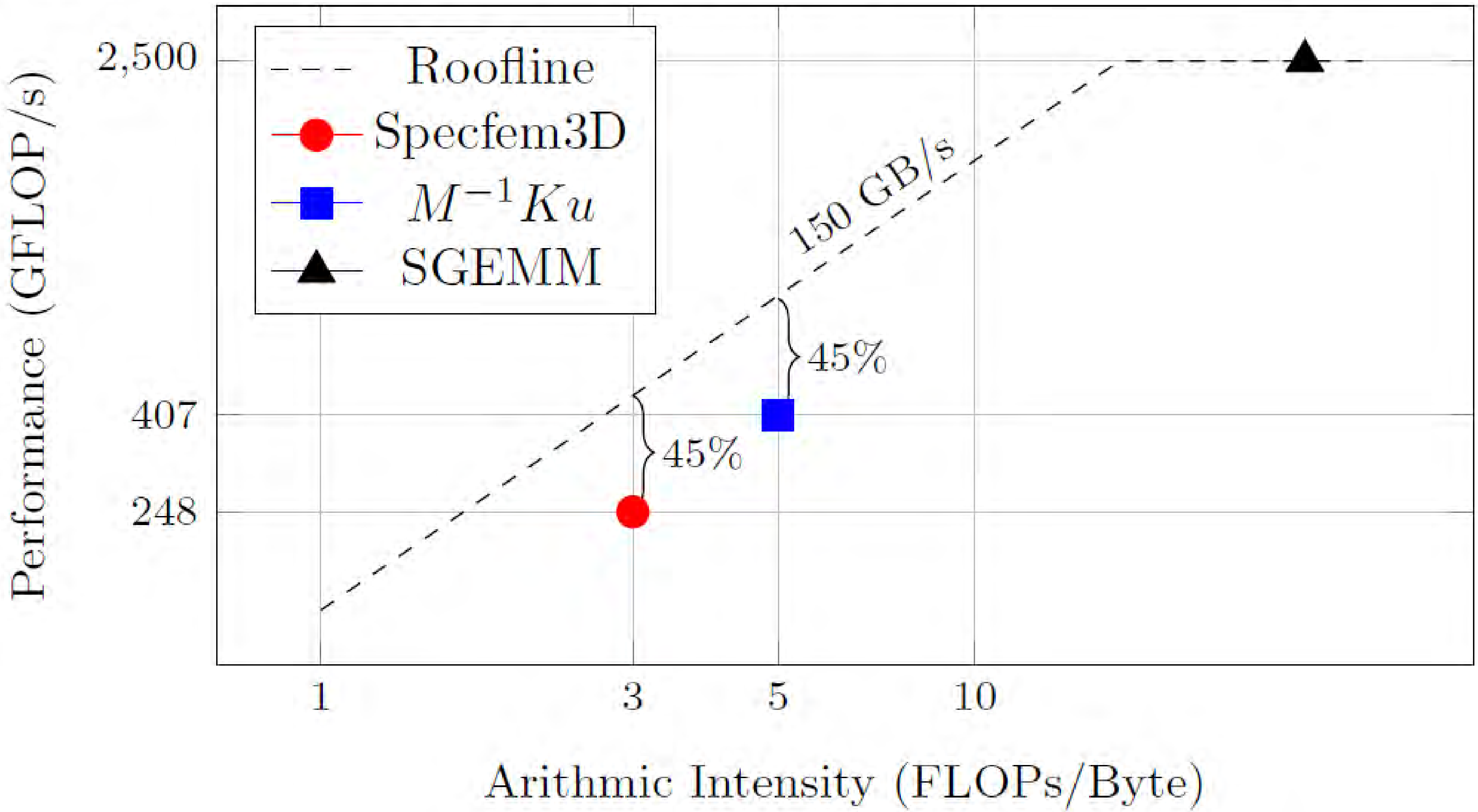


M. Rietmann, O.S. et.al., Forward and Adjoint Simulations of Seismic Wave Propagation on Emerging Large-Scale GPU Architectures, **ACM / IEEE Supercomputing 2012**

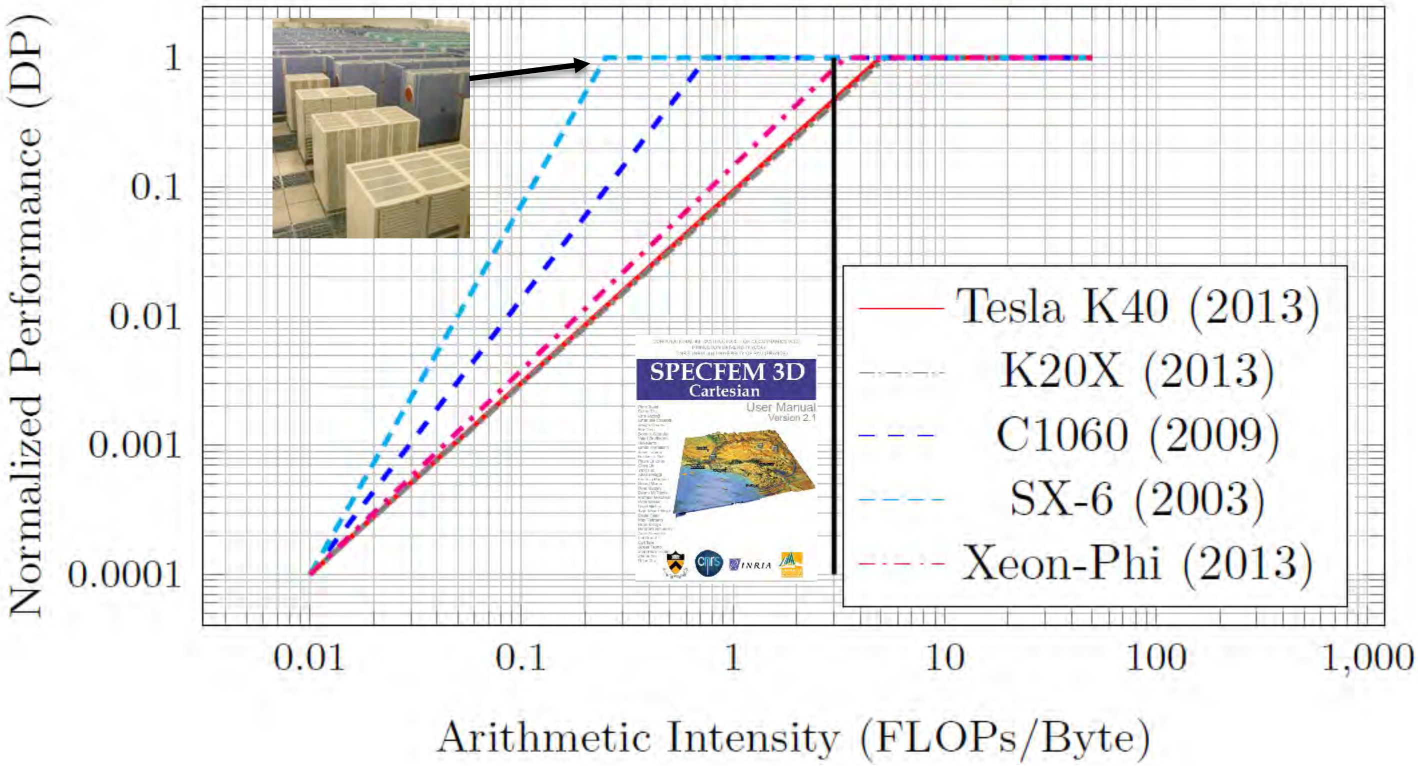
- Similar excellent performance on future manycore architectures? --> **Roofline Model**

SPECFEM and Roofline Model on GPU Tesla K20

Arithmetic intensity of SPECFEM: $q = 3$



SPECFEM and Roofline Trend Model (2003-2015)



- Arithmetic intensity for Peak: 0.25(2003), 0.8(2009), 6.2 (2013)
- **Re-design** to increase **arithmetic intensity** on **accelerators.**

Interior-point methods for large scale seismic optimization on high-performance computers

Inequality constrained minimization

Inequality constrained minimization

$$\min_{\mathbf{x},\mathbf{x_0}} f_0(\mathbf{x}, \mathbf{x}_0)$$

$$\text{s.t.} \quad f_i(\mathbf{x}, \mathbf{x}_0) \geq 0, \quad i = 1, \ldots, m$$

$$A(\mathbf{x}_0) \cdot \mathbf{x} = b_j, \quad j = 1, \ldots, N_e$$

with f_i **nonconvex**, twice continuously differentiable, $A(x_0)$ full rank, $\mathbf{x}_0$ control variables, $\mathbf{x}$ state variables.

Ongoing project:

- Exploiting structure in very large-scale **interior-point optimization**
- Software to solve QPs or NLPs on massively-parallel computers
- DOE INCITE projects on "Titan" (100M CPU h on Cray XK7) and "Mira" (BG/Q)
- 1.95 billion uncertain parameters, 1.94 billion constraints, 25K control variables on "Titan" under "real-time" constraints (30 min).

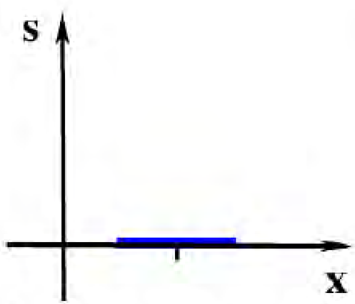
First Order Optimality Conditions

Simplex Method:

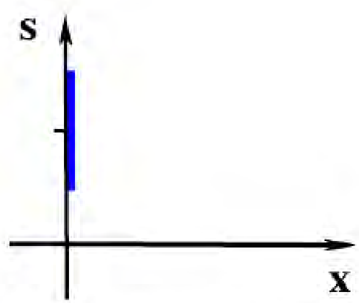
$$
\begin{aligned}
Ax &= b \\
A^T y + s &= c \\
XSe &= 0 \\
x, s &\geq 0
\end{aligned}
$$

Interior Point Method:

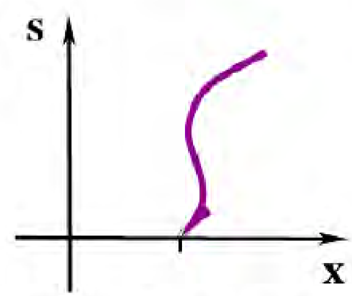
$$
\begin{aligned}
Ax &= b \\
A^T y + s &= c \\
XSe &= \mu e \\
x, s &\geq 0
\end{aligned}
$$



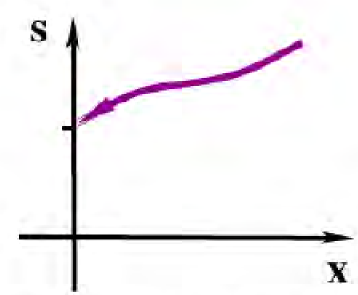
Basic: x > 0, s = 0



Nonbasic: x = 0, s > 0



"Basic": x > 0, s = 0



"Nonbasic": x = 0, s > 0

Nocedal, Wright, **Numerical Optimization**, Springer, 2006.

Convergence of IPM

		Number of IPM Iterations		
Scenarios	Variables	standard	correctors	warm-started
100	105K	23	20	7
200	209K	64	25	9
800	836K	28	22	11
1200	1.6M	33	26	12
2400	3.1M	29	21	9

Theory IPM converge in $O(\sqrt{n})$ iterations

Practise IPM converge in $O(1)$ to $O(log\ n)$ iterations

. . . but one iteration may be VERY expensive

(Slide Source: J. Gondzio, School of Mathematics, University of Edinburgh)

KKT systems in IPMs for LP, QP, and NLP

LP
$$\begin{pmatrix} \Theta^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} f \\ d \end{pmatrix}$$

QP
$$\begin{pmatrix} Q + \Theta^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} f \\ d \end{pmatrix}$$

NLP
$$\begin{pmatrix} Q(x,y) + \Theta_P^{-1} & A^T \\ A & -\Theta_D^{-1} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} f \\ d \end{pmatrix}$$

$\rightarrow$ another regularization is related to Hessian modification

Linear algebra of primal-dual interior-point methods (IPM)

Convex quadratic problem

$$\min \frac{1}{2} x^T Qx + c^T x$$
$$\text{s.t.} Ax = b$$
$$x > 0$$

IPM Linear System

$$\begin{pmatrix} Q+\Delta & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- Multi-stage stage stochastic programming
- nested arrow-shaped linear system (modulo a permutation)
- N is the number of scenarios

$$\begin{pmatrix} Q_1 & W_1^T & & & & & & 0 & 0 \\ W_1 & 0 & & & & & & T_1 & 0 \\ & & Q_2 & W_2^T & & & & 0 & 0 \\ & & W_2 & 0 & & & & T_2 & 0 \\ & & & & \ddots & & & \vdots & \vdots \\ & & & & & Q_N & W_N^T & 0 & 0 \\ & & & & & W_N & 0 & T_N & 0 \\ 0 & T_1^T & 0 & T_2^T & \dots & 0 & T_N^T & Q_0 & W_0^T \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & W_0 & 0 \end{pmatrix}$$

Parallel Solution Procedure for KKT System

N scenarios distributed across $\mathcal{P}$ processes. $\mathcal{N}_p$ is set of scenarios assigned to process $p \in \mathcal{P}$. Each process $p \in \mathcal{P}$ executes the following steps:

(factorization phase)

1.1. Factorize $L_i D_i L_i^T = K_i$ for each $i \in \mathcal{N}_p$.

1.2. Compute SC contribution $S_i = B_i^T K_i^{-1} B_i$ for each $i \in \mathcal{N}_p$.

1.3. Accumulate $C_p = -\sum_{i \in \mathcal{N}_p} S_i$. On process 1, let $C_1 = C_1 + K_0$.

2. Reduce SC matrix $C = \sum_{r \in \mathcal{P}} C_r$ to process 1.

3. Factorize SC matrix $L_c D_c L_c^T = C$ in process 1.

(solve phase)

4.1. Solve $w_i = L_i^{-T} D_i^{-1} L_i^{-1} r_i$ for each $i \in \mathcal{N}_p$.Compute $v_p = \sum_{i \in \mathcal{N}_p} B_i^T w_i$.

4.2. On process 1, let $v_1 = v_1 + r_0$.

5. Reduce $v_0 = \sum_{i \in \mathcal{N}_p} v_i$ to process 1.

6.1. Solve $\Delta z_0 = C^{-1} v_0 = L_c^{-T} D_c^{-1} L_c^{-1} v_0$ in process 1.

6.2. Process 1 broadcasts z_0 to all other processes.

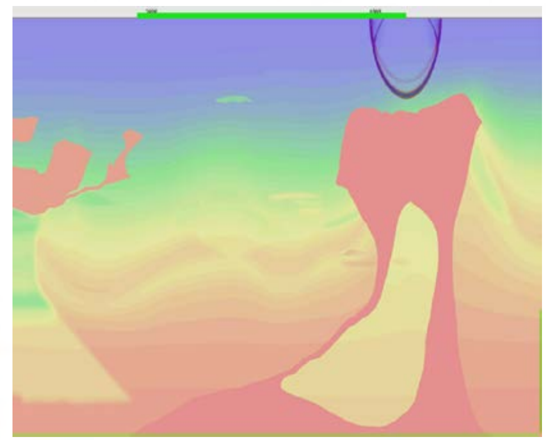
7. Solve $\Delta z_i = L_i^{-T} D_i^{-1} L_i^{-1}(B_i \Delta z_0 - r_i)$ for each $i \in \mathcal{N}_p$.

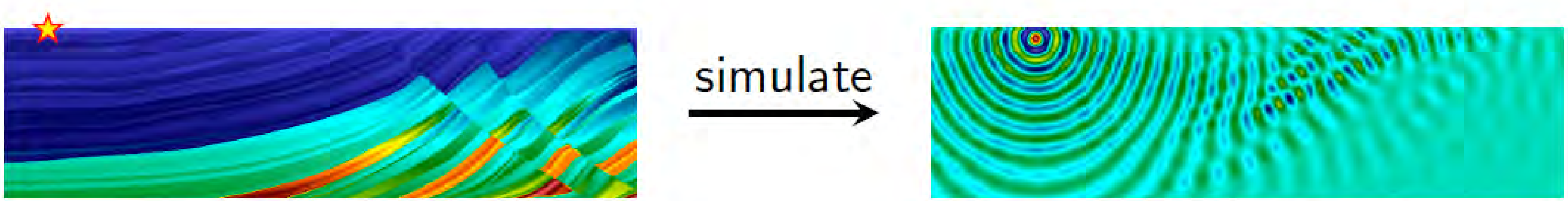
Application 3D Seismic Imaging (ETH, USI, SCEC)

- Simulate subsurface wave y
- Helmholtz equation:

$$A(u, y) = -\textbf{grad} \cdot (u(x)^2 \textbf{grad}\, y(x)) - \omega^2 y(x) - f(x) = 0 \qquad (1)$$

- Plug in parameters
  - Temporal frequency ω
  - Wave source f
  - Wave speed $u(x)$
- Simulate (discretize and solve) $\longrightarrow y(x)$

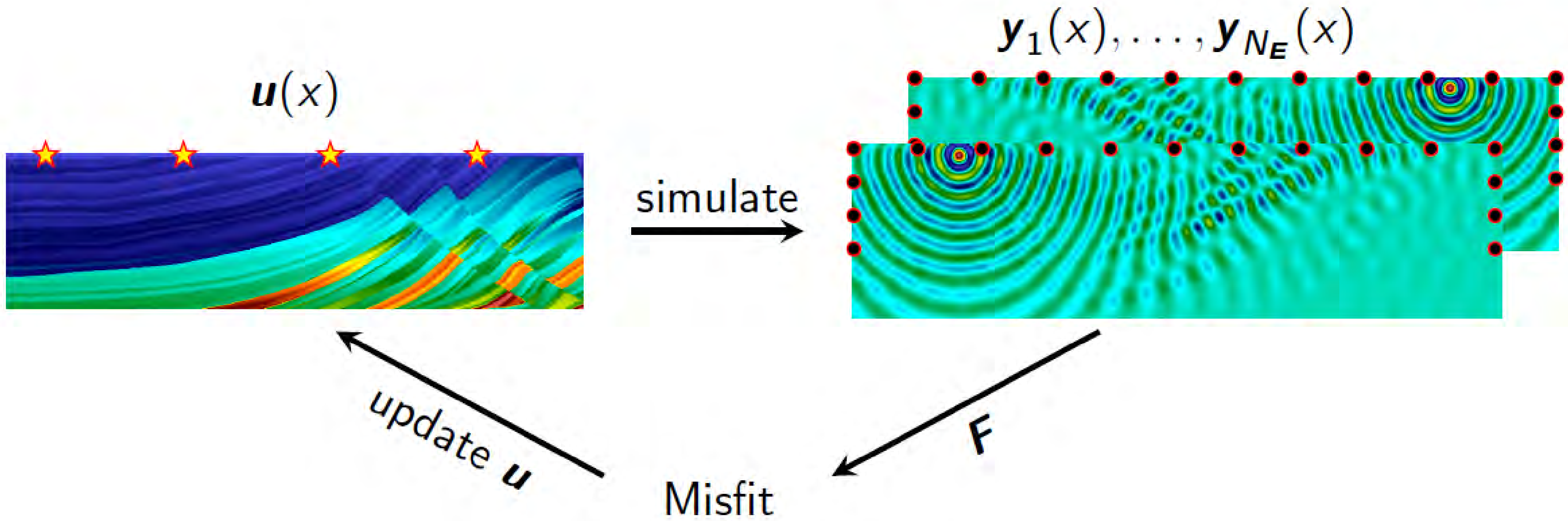




$u(x)$ $y(x)$

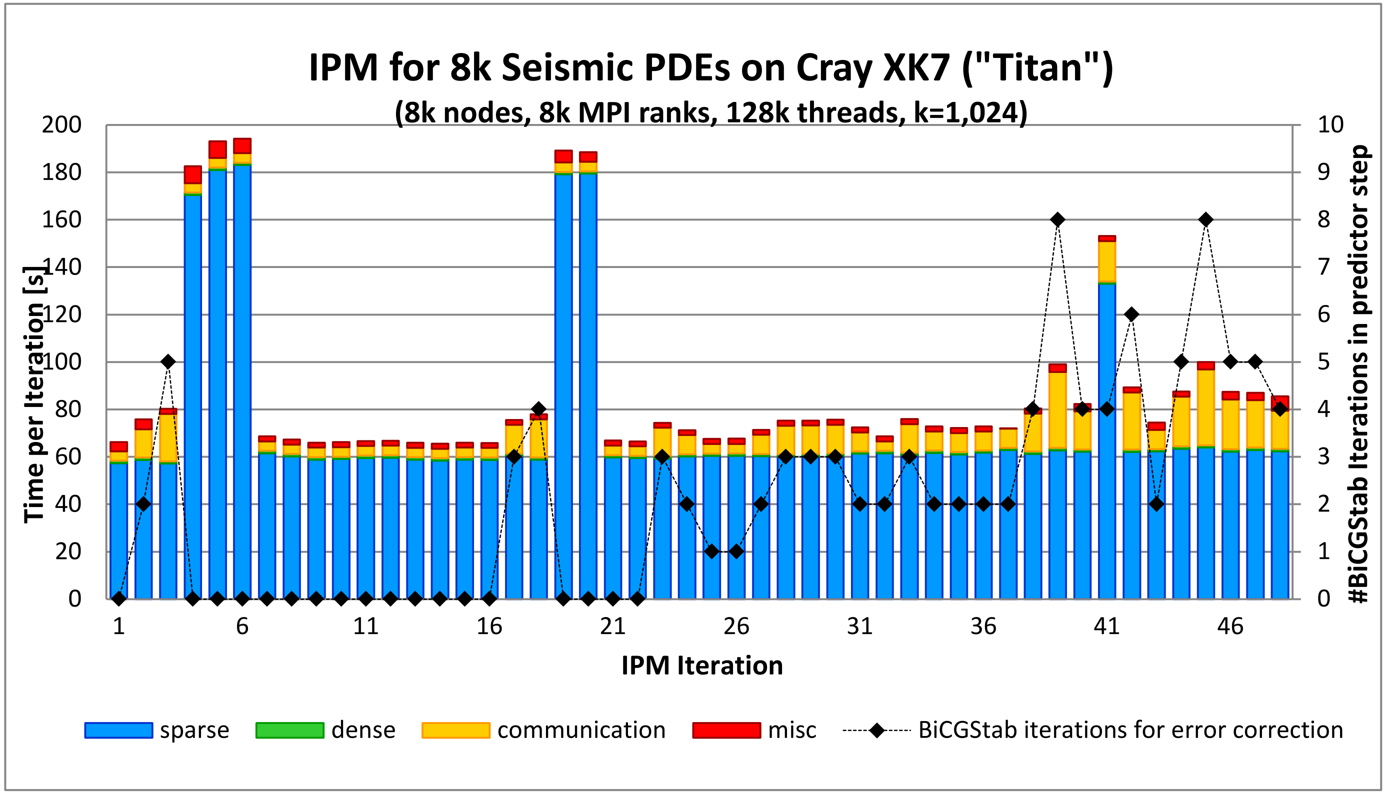
Application 3D Seismic imaging (ETH, USI, SCEC)

- Given multiple sources ☆
- Measurements at location •
- Objective function $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{u})$ measures misfit of simulation and measurements
- Find $\boldsymbol{u}$ with best match of simulation an measurements: $\boldsymbol{F} \rightarrow min$



Time per IPM iteration up to 131,072 cores of “Titan”

- 3D seismic imaging problem (200^3), 8’192 Sensors (“Seismograms”)
- 10,000s of control material variables -> NLP with several billions variables.



- Breakdown of the execution time for each IPM iteration when solving a seismic PDE-constrained optimization problem (Helmholtz) on 8’192 nodes.

M. Christen, O. Schenk, Y. Cui, *Parallel Auto-Tuned Stencils For Scalable Earthquake Simulation Codes*, ACM/IEEE Supercomputing 2013.

Publications

- D. Kourounis, O. Schenk, *Constraint Handling for Gradient-Based Optimization of Compositional Reservoir Flow*, **Journal of Computational Geosciences**, **2015**.
- M. Rietmann, O. Schenk, et.al., *Load-balanced Local Time Stepping for Large-Scale Wave Propagation*, IEEE International Parallel & Distributed Processing Symposium, **IPDPS'15.**
- C. Petra, O. Schenk, M. Anitescu, *Real-time Stochastic Optimization of Complex Energy Systems on High Performance Computers*, **IEEE Computing in Science & Engineering - Leadership Computing,** Volume: 16 pp. 32–42, **2014**.
- M. J. Grote, J. Huber, D. Kourounis, O. Schenk, *Inexact Interior-Point Method for PDE-Constrained Nonlinear Optimization*, **SIAM J. Sci. Comput**. 36-3, pp. A1251-A1276, **2014.**
- C. Petra, O. Schenk, M. Lubin, K. Gärtner, *An augmented incomplete factorization approach for computing the Schur complement in stochastic optimization*, **SIAM J. Sci. Comput, 2014**.
- Rietmann, O. Schenk, et.al, *Forward and Adjoint Simulations of Seismic Wave Propagation on Emerging Large-Scale GPU Architectures*, **ACM / IEEE Supercomputing 2013.**
- M. Christen, O. Schenk, Y. Cui, *PATUS: Parallel Auto-Tuned Stencils For Scalable Earthquake Simulation Codes*, **ACM / IEEE Supercomputing 2013**.
- F. Curtis, J. Huber, O. Schenk, A. Wächter, A Note on the Implementation of an Interior-Point Algorithm for Nonlinear Optimization with Inexact Step Computations, **Mathematical Programming Series** B, 32(6): 3447–3475, **2012**.
- F. Curtis, O. Schenk, and W. Wächter, *An Interior-Point Algorithm for Large-Scale Nonlinear Optimization with Inexact Step Computations*, **SIAM J. Sci. Comput**. Volume 32, Issue 6, pp. 3447-3475, **2010**.

U.S. DEPARTMENT OF ENERGY INCITE LEADERSHIP COMPUTING

PASC Platform for Advanced Scientific Computing

HP2C High Performance and High Productivity Computing

Conclusion

- Swiss Platform for Advanced Scientific Computing (PASC)
  - Long-term initiative on supercomputing and computational science.
- Supercomputing and Exascale Computing
  - Re-design algorithms to increase arithmetic intensity on manycores.
- [Un-] Structured Grid Simulations on Many-Core Architectures
  - High-Performance & High-Productivity Stencil Compiler Project PATUS
  - Integrated into anelastic wave propation code from Southern California Earthquake Center.
  - Spectral element solver for wave propagations: SPECFEM
- Code and algorithmic optimization necessary to tackle global seismic tomography (stencil codes, nonlinear optimization, etc.)

Thanks for your attention.

Università della Svizzera italiana

Università della Svizzera italiana | Faculty of Informatics | Institute of Computational Science ICS

Publications

- D. Kourounis, O. Schenk, *Constraint Handling for Gradient-Based Optimization of Compositional Reservoir Flow*, **Journal of Computational Geosciences**, **2015**.
- M. Rietmann, O. Schenk, et.al., *Load-balanced Local Time Stepping for Large-Scale Wave Propagation*, IEEE International Parallel & Distributed Processing Symposium**, IPDPS'15.**
- C. Petra, O. Schenk, M. Anitescu, *Real-time Stochastic Optimization of Complex Energy Systems on High Performance Computers*, **IEEE Computing in Science & Engineering - Leadership Computing,** Volume: 16 pp. 32–42, **2014**.
- M. J. Grote, J. Huber, D. Kourounis, O. Schenk, *Inexact Interior-Point Method for PDE-Constrained Nonlinear Optimization*, **SIAM J. Sci. Comput**. 36-3, pp. A1251-A1276, **2014.**
- C. Petra, O. Schenk, M. Lubin, K. Gärtner, *An augmented incomplete factorization approach for computing the Schur complement in stochastic optimization***, SIAM J. Sci. Comput, 2014**.
- Rietmann, O. Schenk, et.al, *Forward and Adjoint Simulations of Seismic Wave Propagation on Emerging Large-Scale GPU Architectures*, **ACM / IEEE Supercomputing 2013.**
- M. Christen, O. Schenk, Y. Cui, *PATUS: Parallel Auto-Tuned Stencils For Scalable Earthquake Simulation Codes*, **ACM / IEEE Supercomputing 2013**.
- F. Curtis, J. Huber, O. Schenk, A. Wächter, A Note on the Implementation of an Interior-Point Algorithm for Nonlinear Optimization with Inexact Step Computations, **Mathematical Programming Series** B, 32(6): 3447–3475, **2012**.
- F. Curtis, O. Schenk, and W. Wächter, *An Interior-Point Algorithm for Large-Scale Nonlinear Optimization with Inexact Step Computations*, **SIAM J. Sci. Comput**. Volume 32, Issue 6, pp. 3447-3475, **2010**.

U.S. DEPARTMENT OF ENERGY INCITE LEADERSHIP COMPUTING

PASC Platform for Advanced Scientific Computing

HPZC High Performance and High Productivity Computing