Optimization of finite-difference kernels on multi-core architectures for seismic applications

V. Etienne[1], T. Tonellot[1], K. Akbudak[2], H. Ltaief[2], S. Kortas[3], T. Malas[4], P. Thierry[4], D. Keyes[2]

[1] Saudi Aramco, EXPEC ARC

[2] King Abdullah University of Science and Technology, ECRC

[3] King Abdullah University of Science and Technology, KSL

[4] Intel

April, 2018

where energy is opportunity™

Agenda

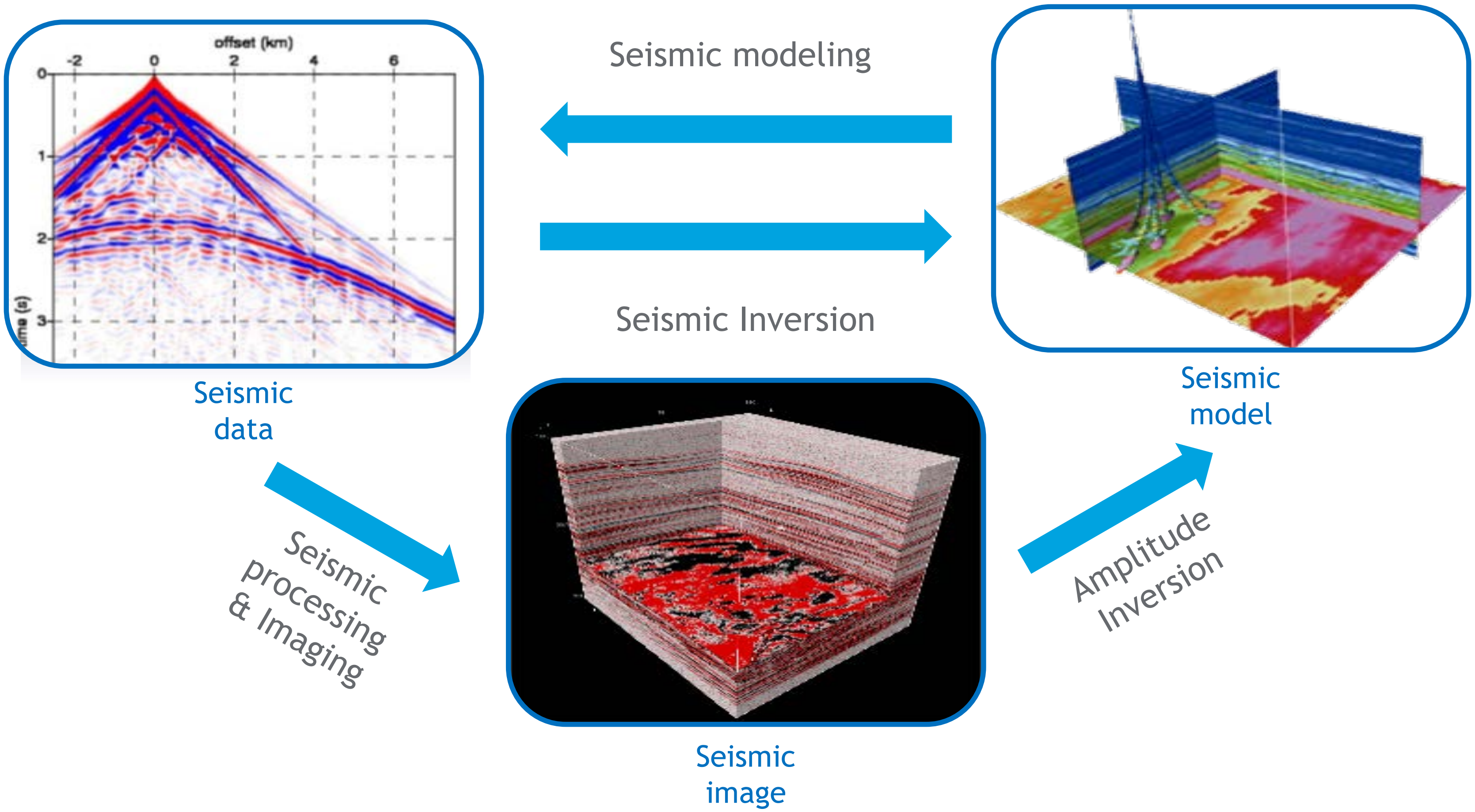
1. Introduction
2. Seismic modeling with the finite-difference method
3. Spatial blocking
4. Temporal blocking
5. Application to seismic modeling and imaging
6. Conclusions

Introduction

1

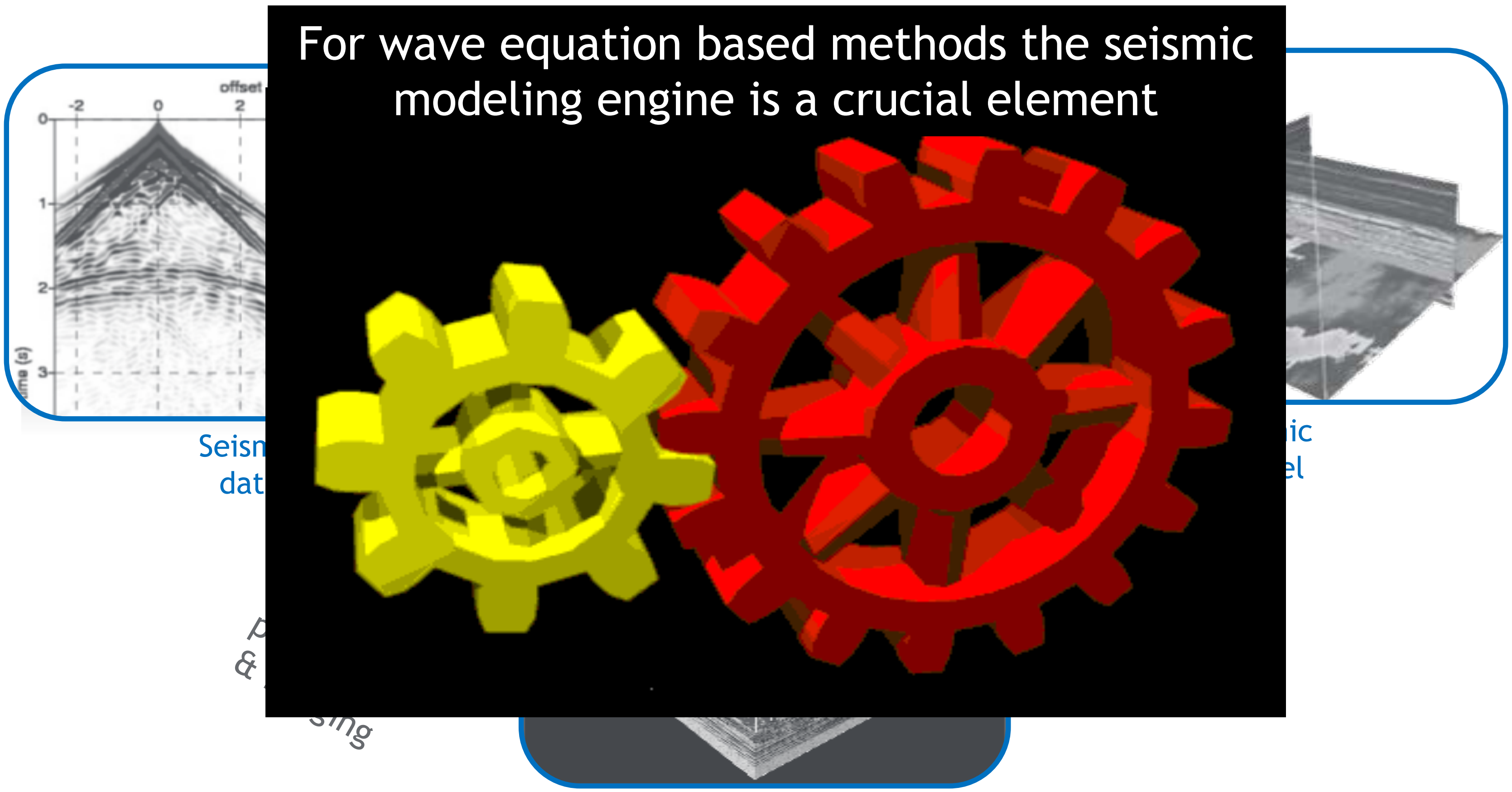
Introduction

Seismic modeling / Imaging / Inversion



Introduction

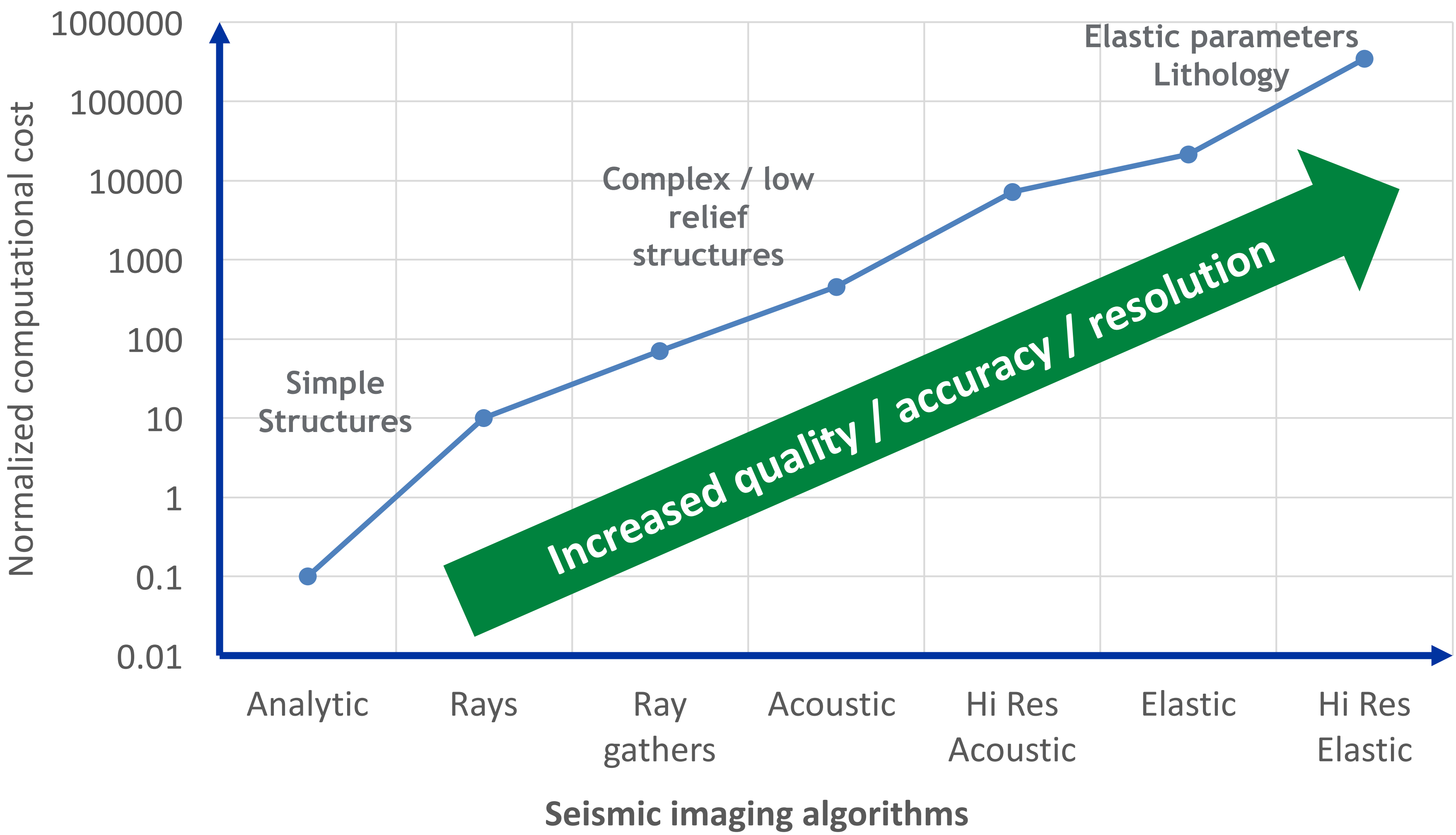
Seismic modeling / Imaging / Inversion



أرامكو السعودية
saudi aramco

Introduction

Why seismic modeling is so important?



أرامكو السعودية
saudi aramco

Introduction

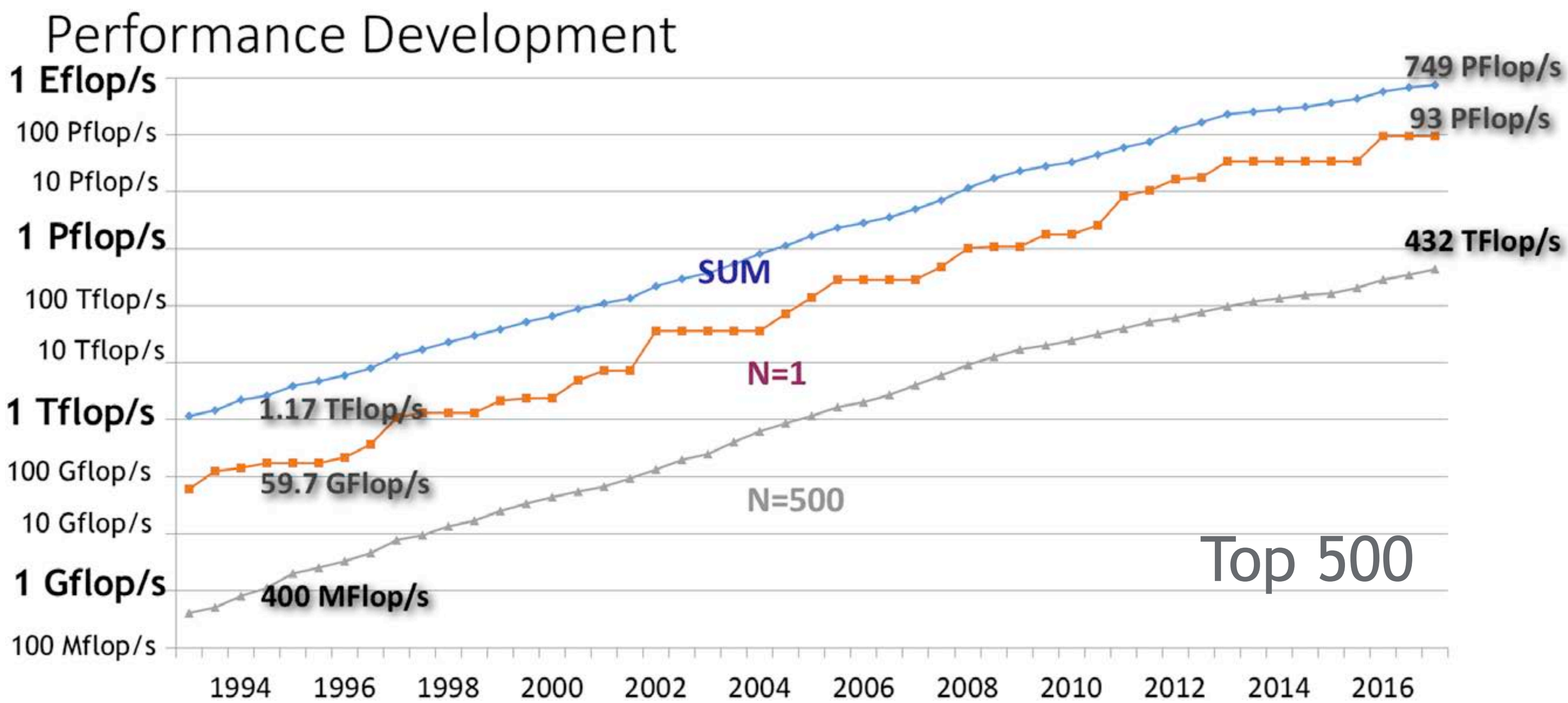
Why seismic modeling is so important?

Growth of acquired data

Typical Acquisition	
Recv/Shot	207,360
Trace/Shot	23040
Source density	640 VPs/km2
Rec Time	6s @ 2ms
Area	10,000 km2
Num Shots	6.4 MM
Size per Shot	260 MB
Size per km2	166 GB
Total Size	1.6 PB

Introduction

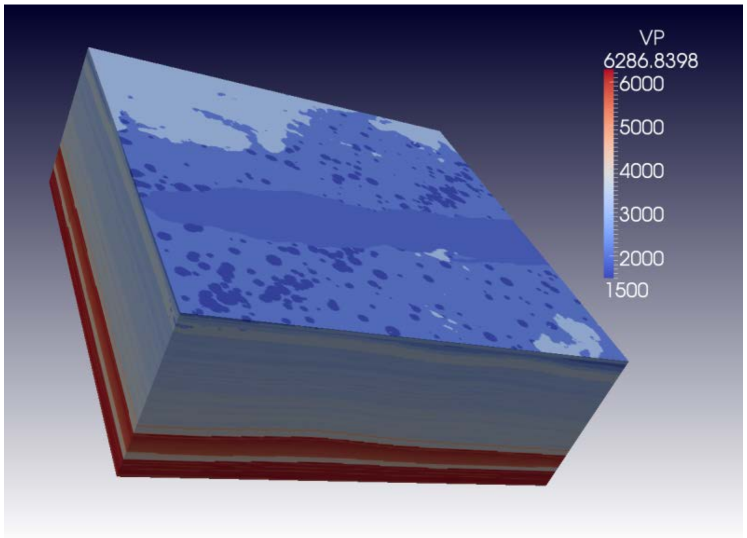
Why seismic modeling is so important?



We do need to develop seismic modeling engines that can scale accordingly to the growth in computer resource

Introduction

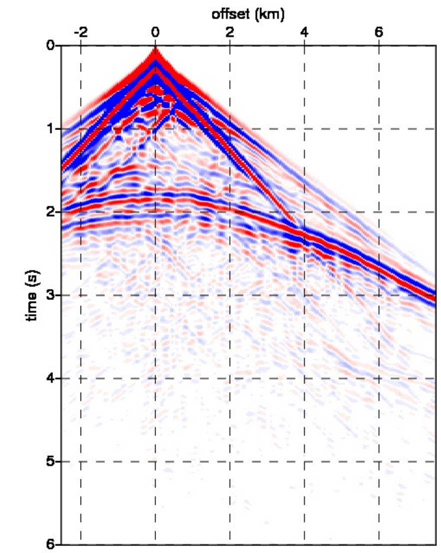
Our goal: design a versatile platform suitable for seismic modeling or inversion algorithms on High Performance Computing platforms



Model

Seismic modeling
Forward problem

Seismic inversion
Inverse problem



Data

HPC

Introduction

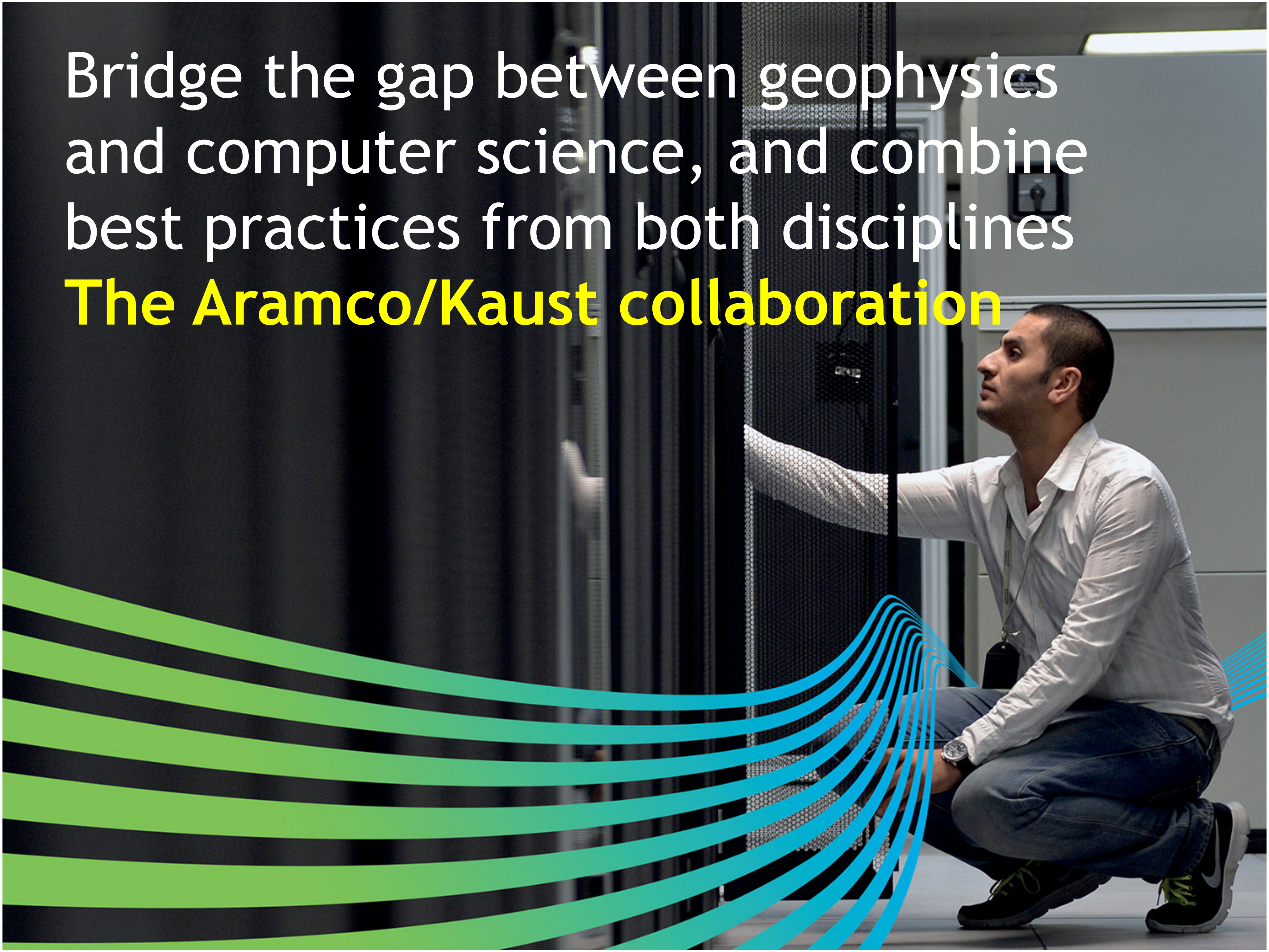
Requirements and technical choices

For our seismic modeling needs:

- We want to cover a wide range of applications
- Using well established numerical methods
- Accurate and stable
- Efficient on modern and future architectures

This leads us to:

- Finite-difference modeling operators
- Staggered grid & time explicit scheme
- CPML absorbing boundaries
- OpenMP/MPI design
- Cache blocking techniques


Bridge the gap between geophysics
and computer science, and combine
best practices from both disciplines
The Aramco/Kaust collaboration

Seismic modeling with the finite-difference method

2

Seismic modeling with finite-difference

Spatial operators

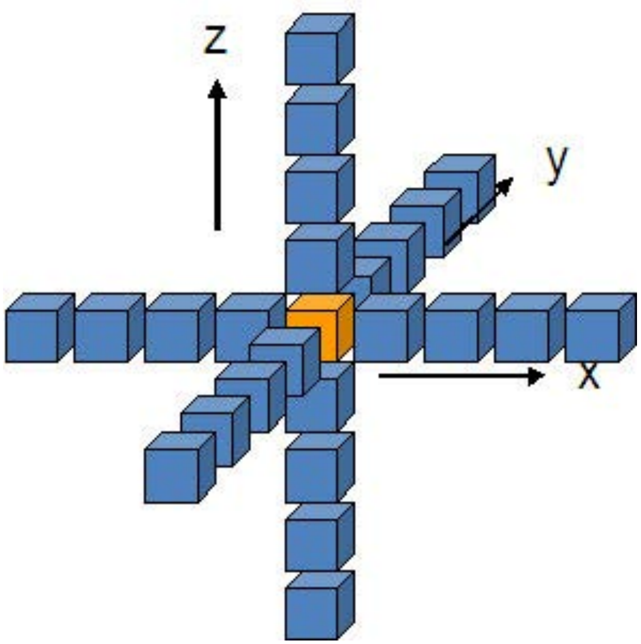
The 3D wave acoustic wave equation

$$\partial_t^2 p = c^2\left(\partial_x^2 p + \partial_y^2 p + \partial_z^2 p\right)$$

is discretized as follows

$$\frac{P_{ijk}^{n+1} - 2P_{ijk}^{n} + P_{ijk}^{n-1}}{\Delta t^2} = c_{ijk}^2 \left(O_{ii}^{N}(P_{ijk}^{n}) + O_{jj}^{N}(P_{ijk}^{n}) + O_{kk}^{N}(P_{ijk}^{n}) \right)$$

where O_{ii}^{N} is the Nth-order spatial operator to evaluate second derivative along index i



Order 8 spatial operator

Each grid point requires 4 manipulations:

- The computation of derivatives along x, y, and z
- The update of pressure

Seismic modeling with finite-difference

Simple 2D acoustic examples

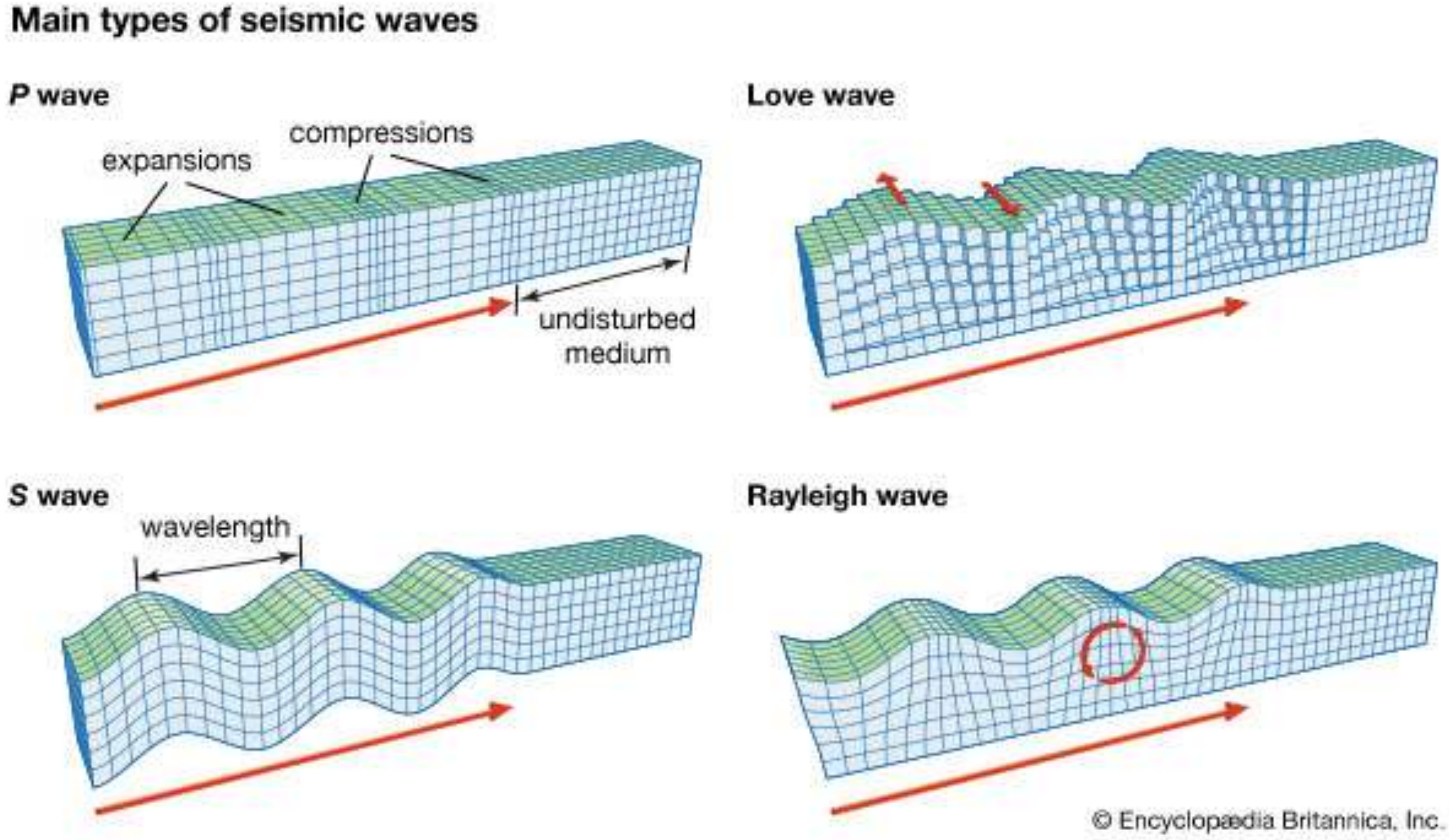
Movie: pure time-domain

Movie: hybrid time/frequency-domain

Complete set-up = model + kernel + boundary conditions + source + receivers

Seismic modeling with finite-difference

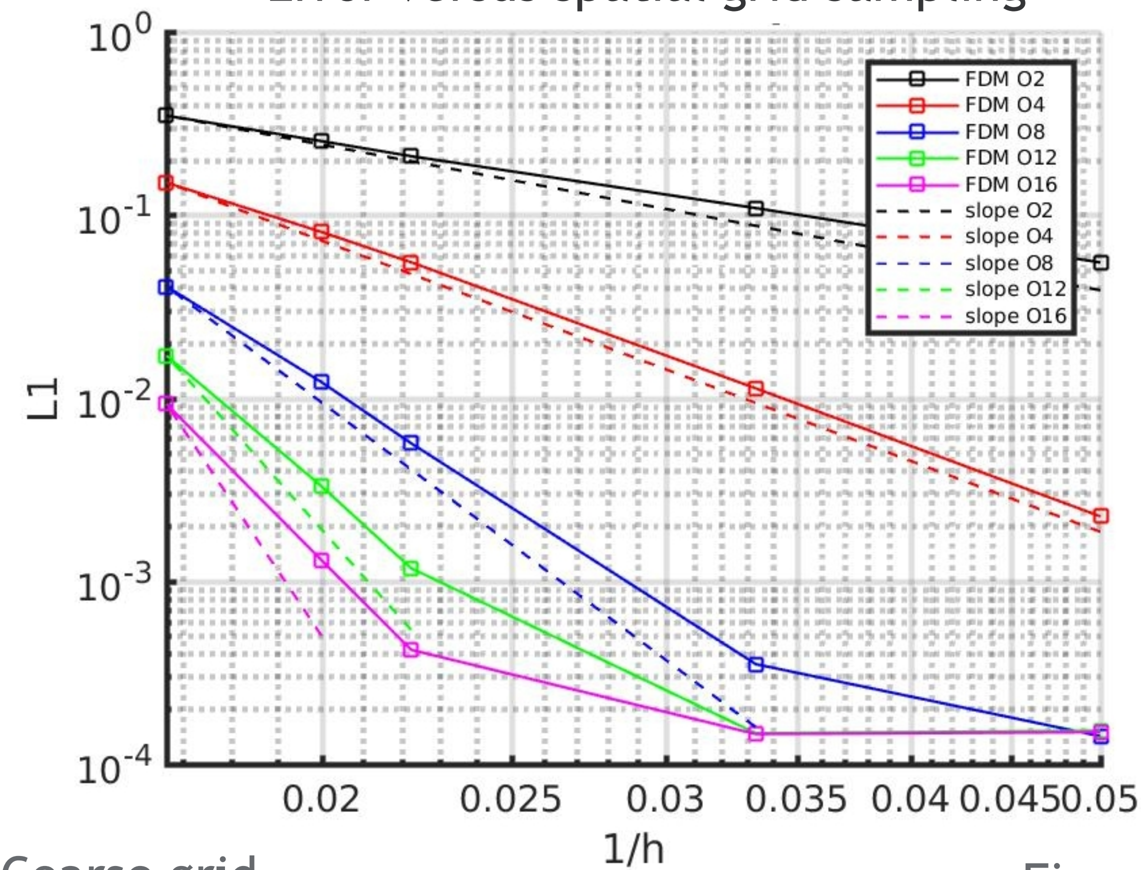
3D elastic – a complex mix of waves



Seismic modeling with finite-difference

Adaptive accuracy

Error versus spatial grid sampling



Coarse grid → Fine grid

Typical discretization rules for optimal accuracy

O2 → 10 pt/λ

O4 → 5 pt/λ

O8 → 4 pt/λ

O12 → 3.5 pt/λ

O16 → 3.2 pt/λ

FWI

RTM

No universal scheme
Application dependent

Seismic modeling with finite-difference

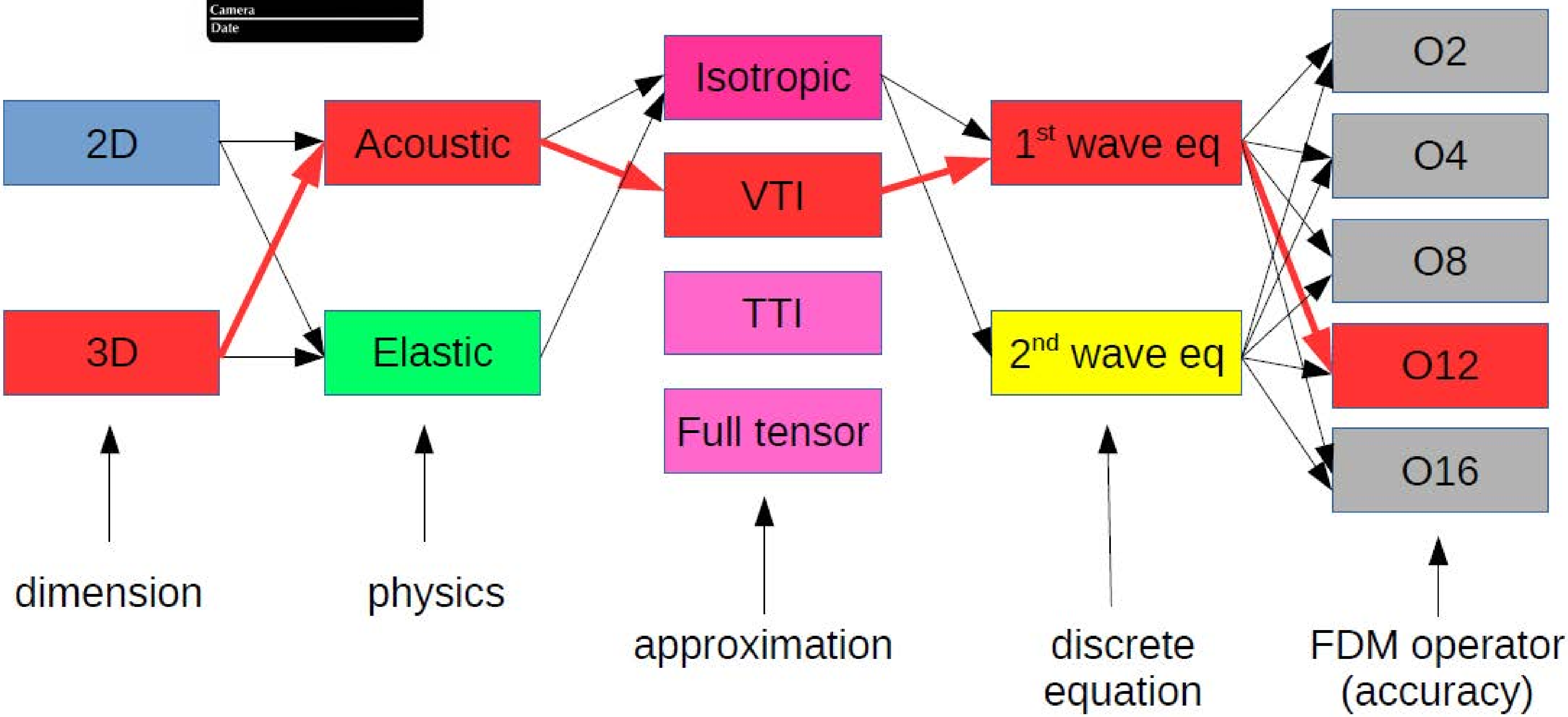
Flexible implementation

Follow an object oriented design
Recast seismic modeling into the objects framework

- dimension: 2D, 3D
- physics: Acoustic, Elastic
- approximation: Isotropic, VTI, TTI, Full tensor
- discrete equation: 1st wave eq, 2nd wave eq
- FDM stencil: O2, O4, O8, O12, O16

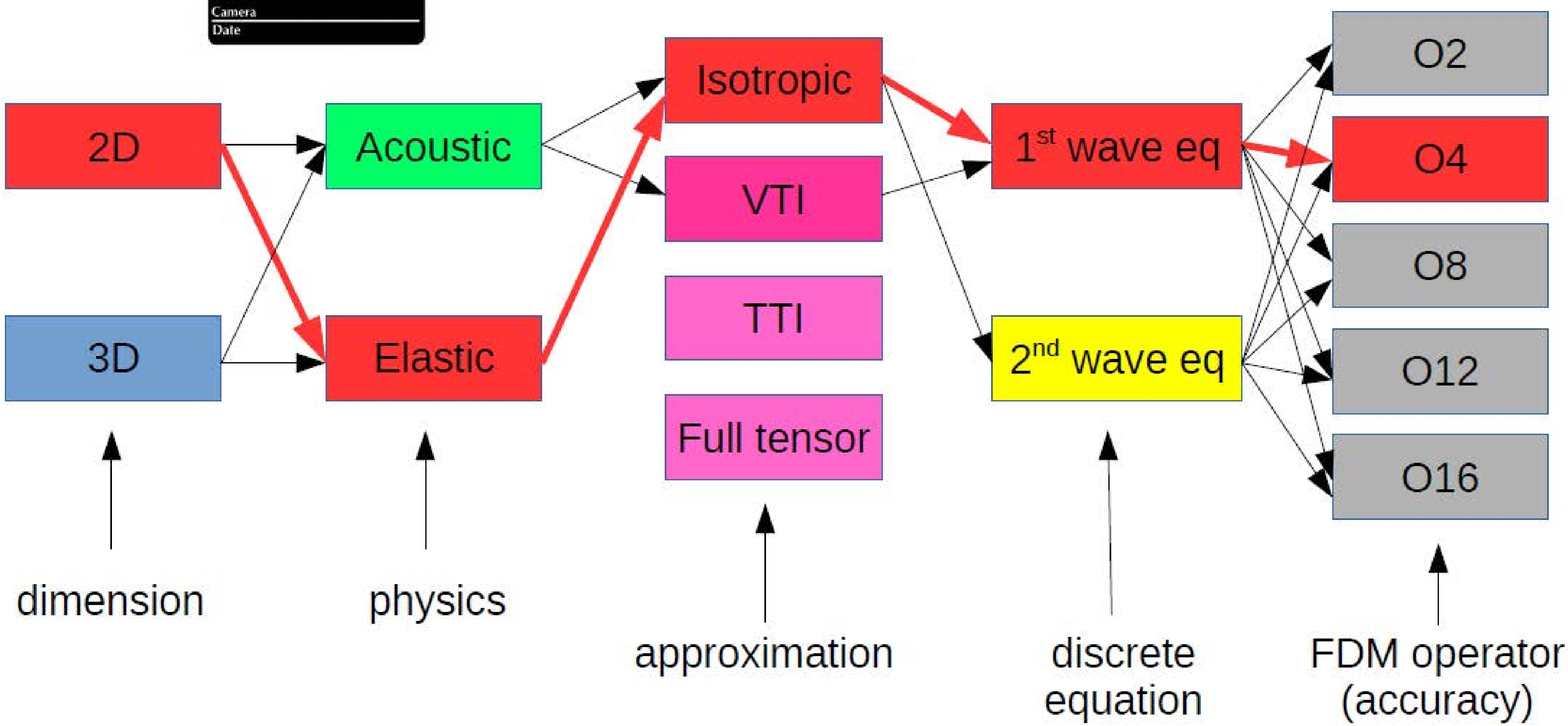
Seismic modeling with finite-difference

<u>Scenario 1</u>
RTM with marine acquisition



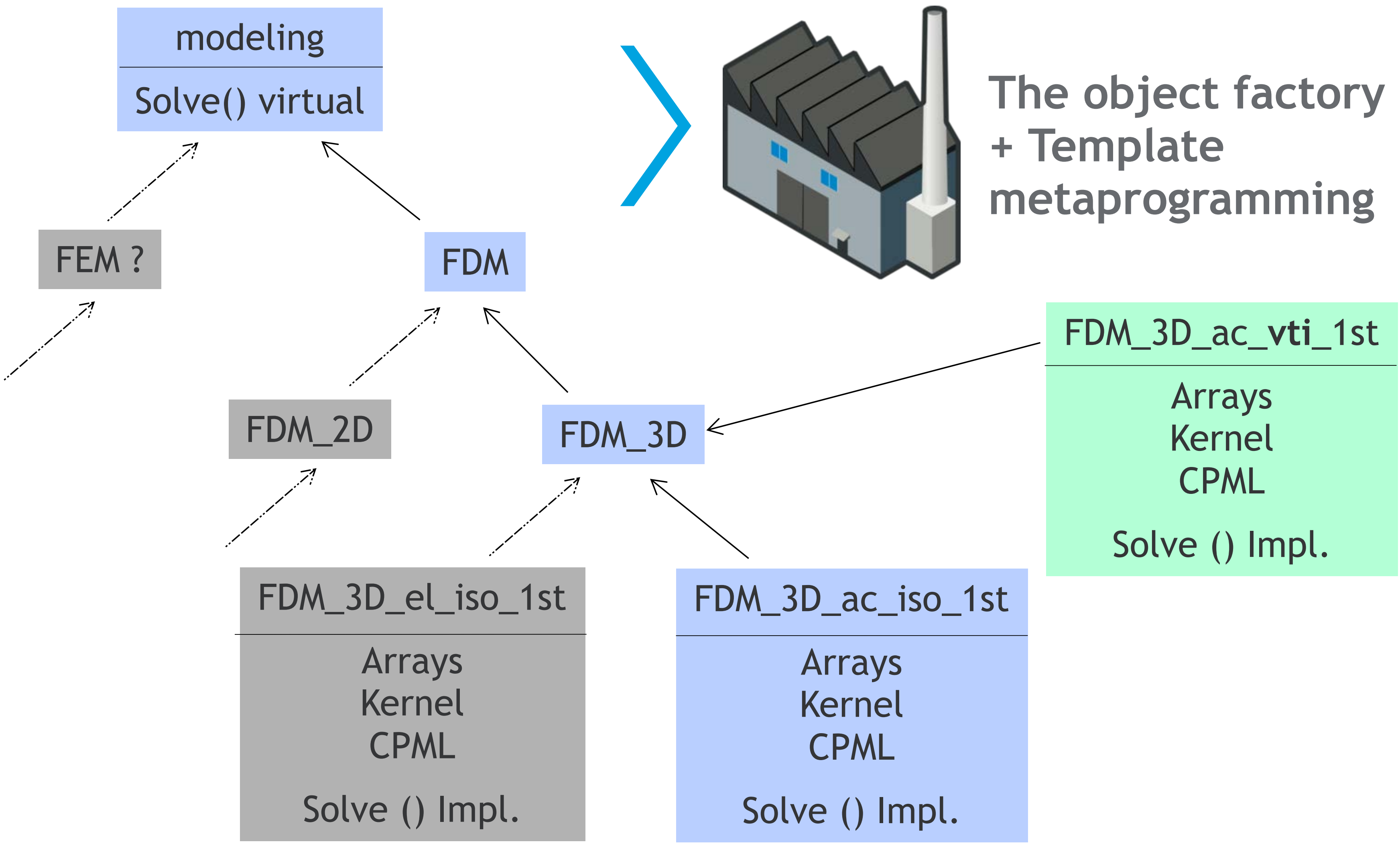
Seismic modeling with finite-difference

Scenario 2
FWI with land acquisition



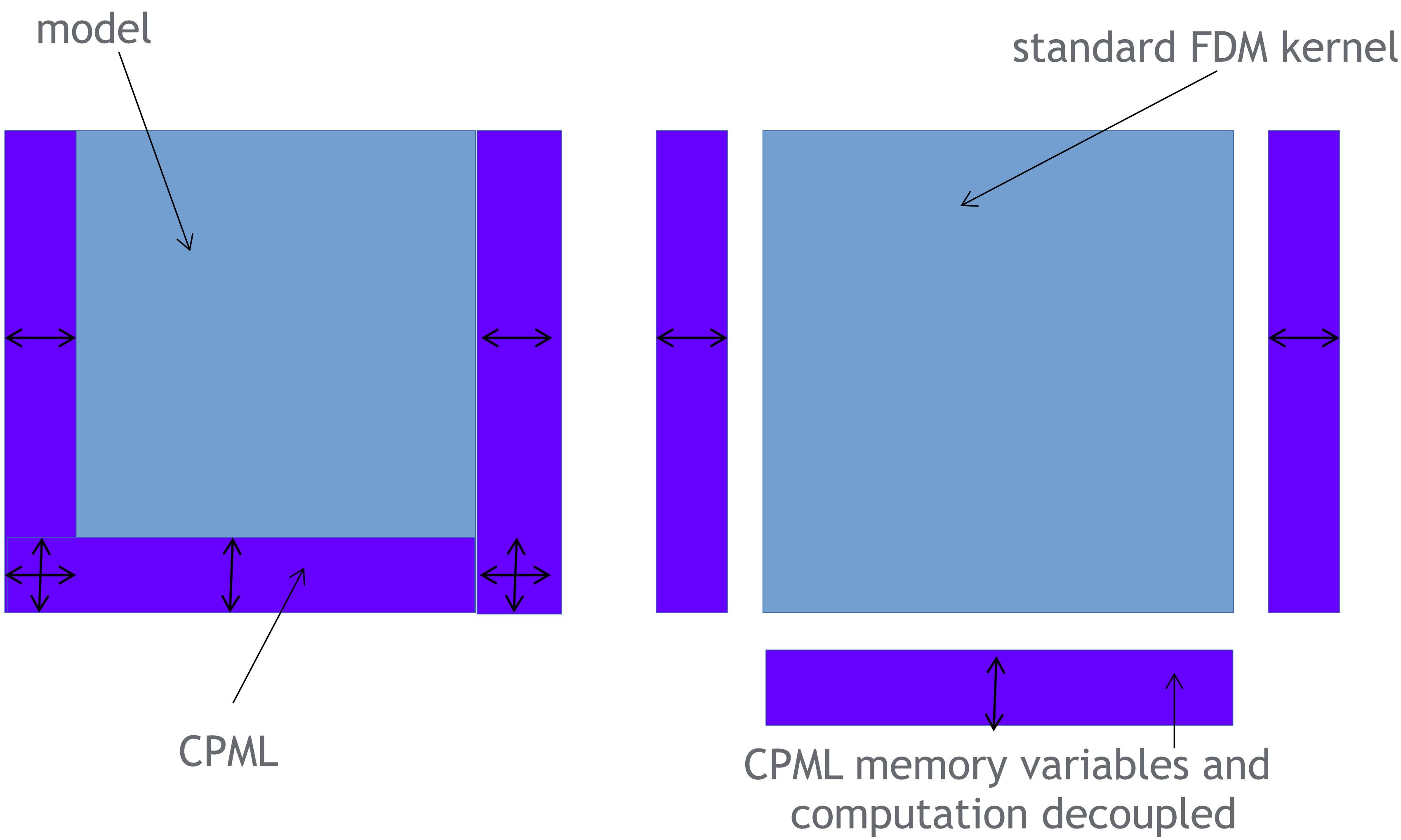
Seismic modeling with finite-difference

Flexible implementation



Seismic modeling with finite-difference

Efficient boundary implementation



Spatial blocking

3

Spatial blocking

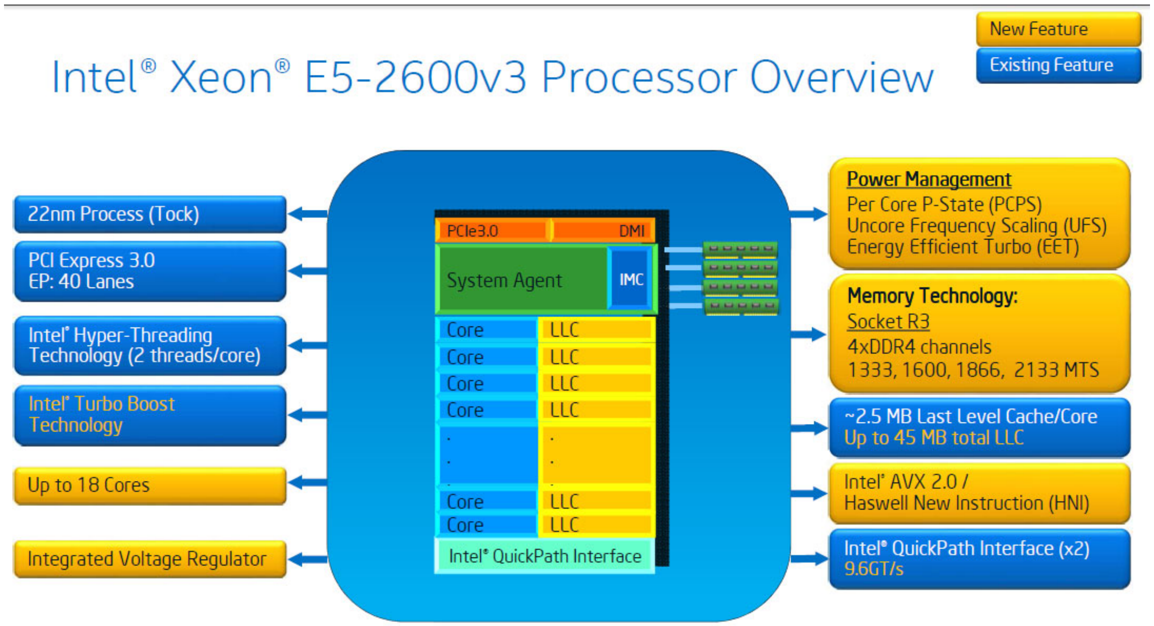
Intel Xeon E5-2600 "Haswell" specifications

SHAHEEN II at KAUST

Computing nodes
6174 Haswell nodes
- 2 socket/node
- QPI x2 between sockets
- 16 cores/socket

Computing core
Core freq. 2.3 GHz
AVX2 16 SP float/vector
2.36 Tflop/s per node

Memory
L1 cache/core 32 KB
L2 cache/core 256 KB
L3 cache shared 40 MB
RAM 128 GB

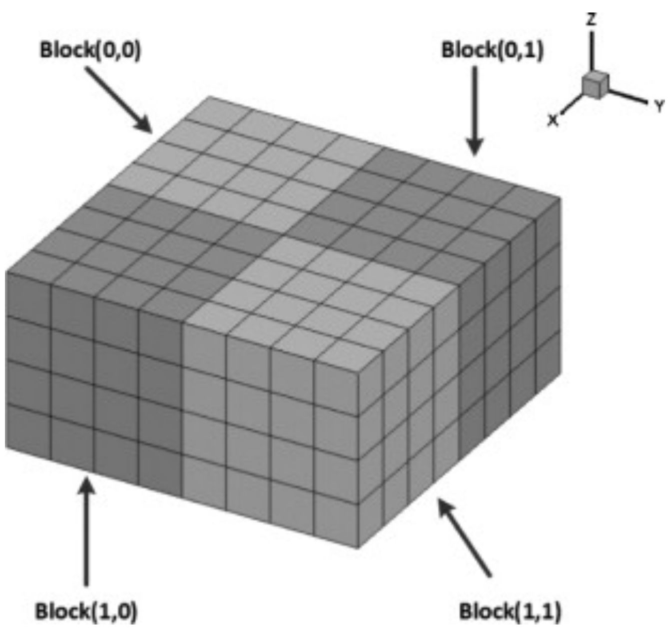


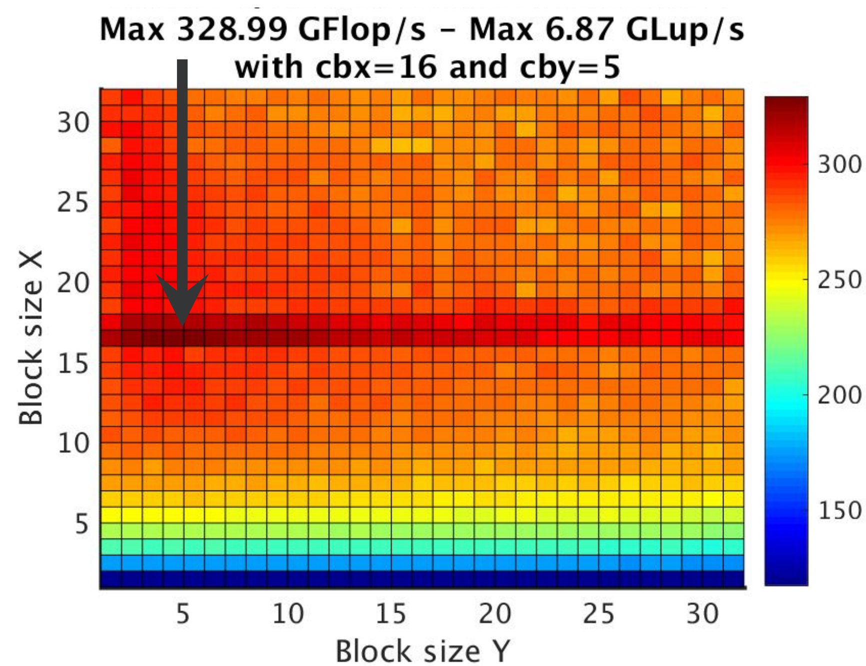
FD kernels are typically memory bounded algorithms
Computations are faster than getting data from RAM
If data reside in cache, computation speed can increase
Typical grid size: 1000x1000x500 (2 GB) can not fit into L3…

How to proceed?

Spatial blocking

General concept





Concept of Cache Blocking

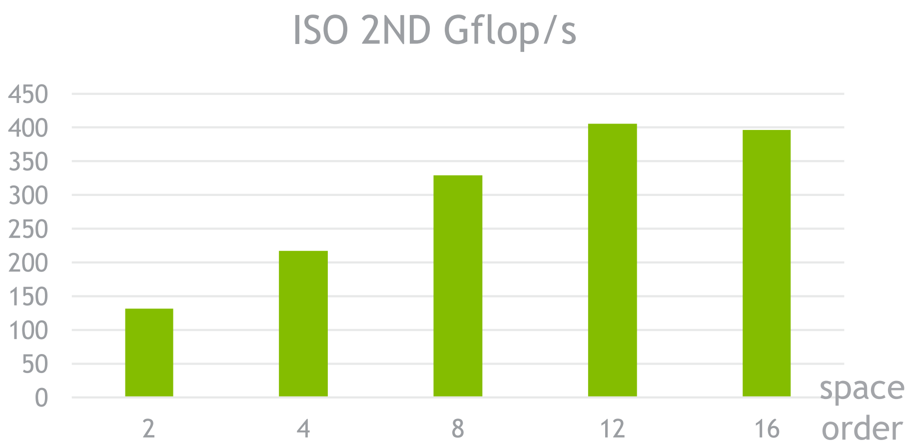
- Divide grid into blocks that fit in CPU cache
- Enhance data reuse in cache
- Crucial on multi-core architectures

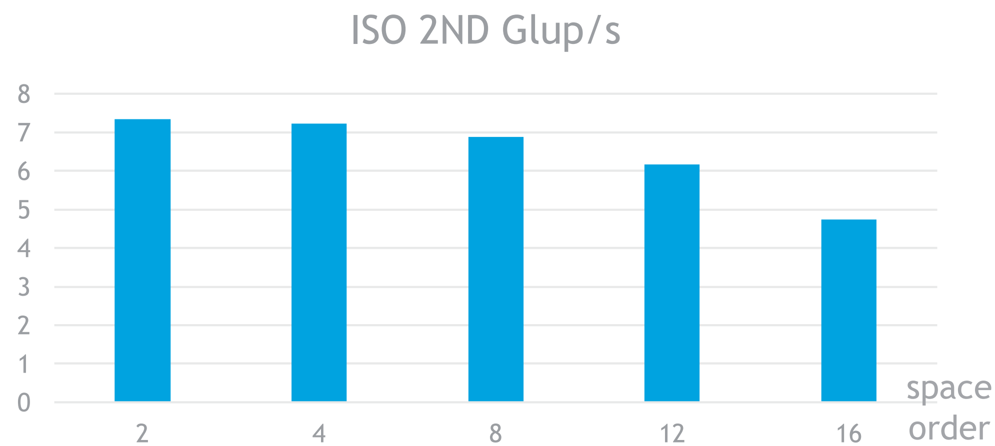
Cache blocking tuning on Shaheen II

- No cache blocking in z (AVX2 vectorization)
- Exploration in x and y (1 to 32 points)
- 1024 configurations evaluated

Spatial blocking

Impact of the spatial order

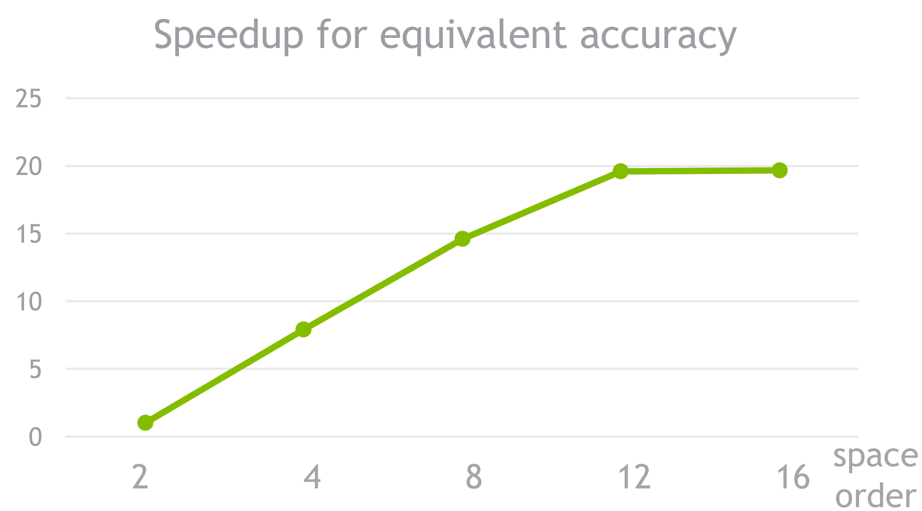




Grid size 512x512x512 (for all tests) - Intel Haswell 2 sockets x 16 cores (32 threads)

Increase spatial order
higher performance (Gflop/s)
lower grid point update/sec (Glup/s)
larger spacing can be used (slide 9)

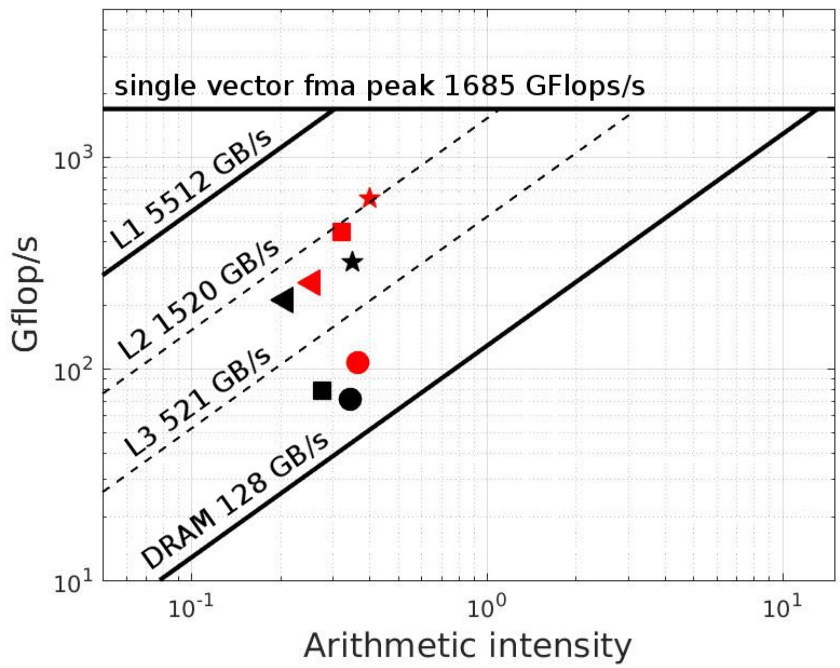
→ **reduced computation time for same accuracy** (with larger spatial sampling)



Spatial blocking

Fine performance analysis with cache-aware roofline model

Cache-aware roofline model



FDM O2 132 Gflop/s ❯ **FDM O8** 329 Gflop/s

derivatives of P[ix][iy][iz]

n3
▼ 211 ▼ 253
L3 - perf. stable

n1 (fast index)
★317 ★637
L2 - highest perf.

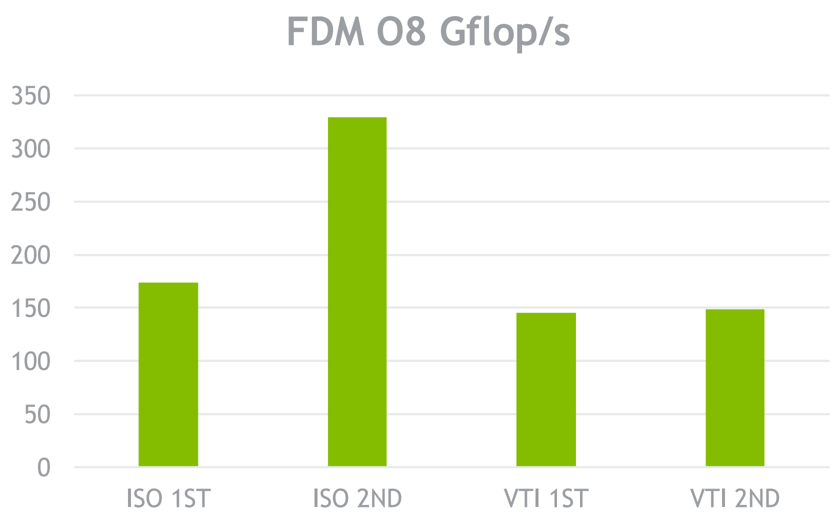
n2 ■79 ■440
L2 - highest speedup × 5.6

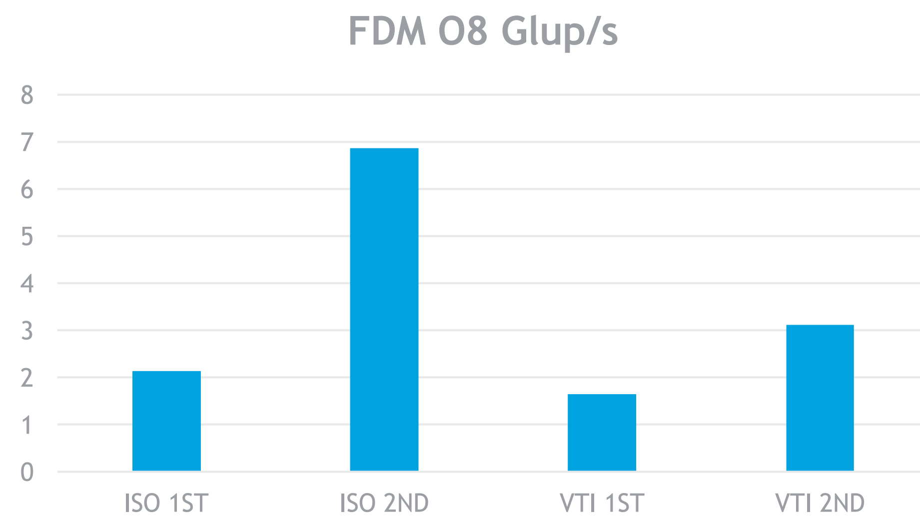
+ pressure update ●72 ●106
DRAM - perf. stable

High speed and good cache-reuse observed for n1 and n2 derivatives
Lower speed and lower cache-reuse for n3 derivative and pressure update
➡ **There is still room for improvement**

Spatial blocking

Impact of the equation





Isotropic 2nd vs 1st order wave equation

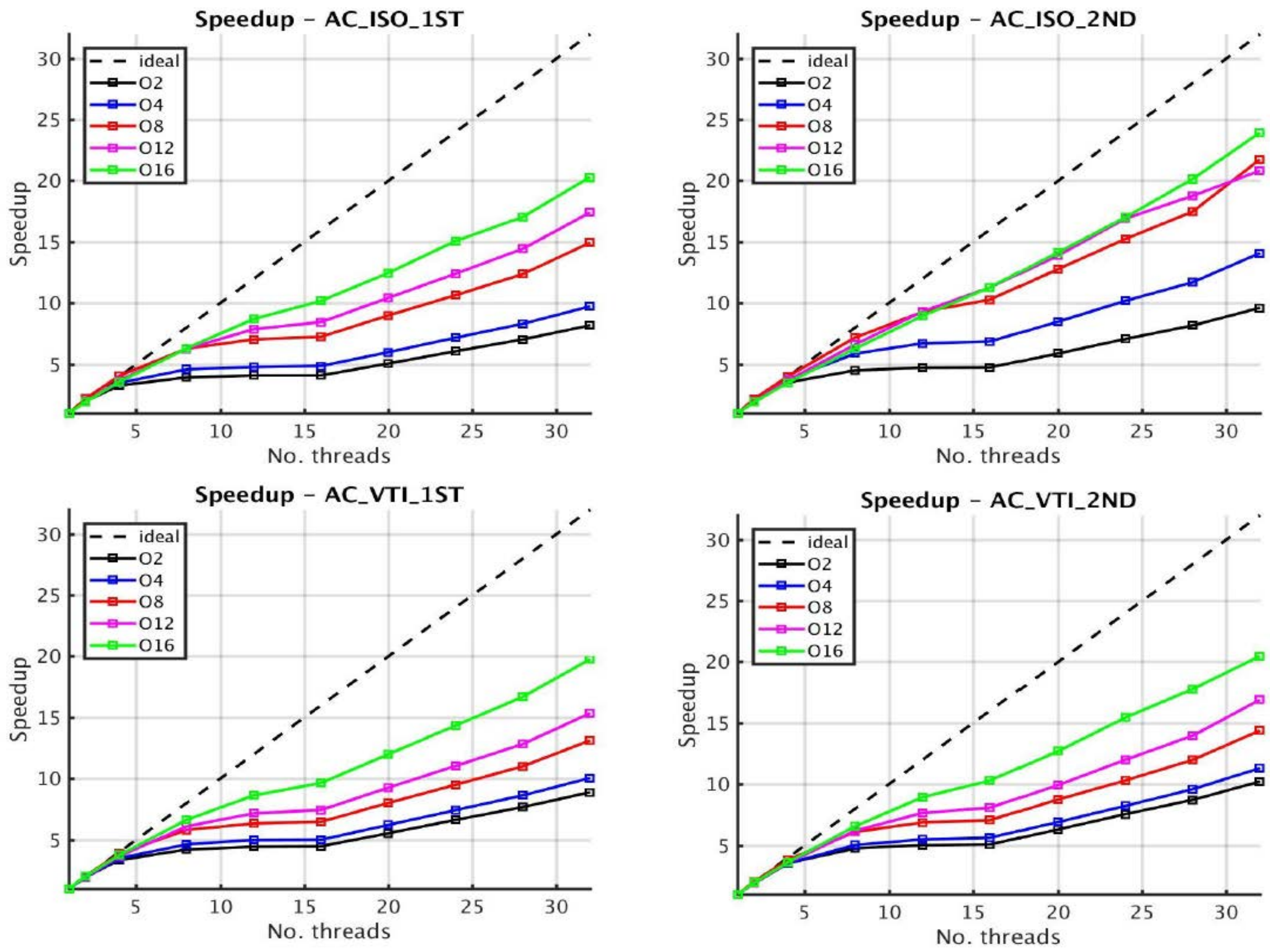
- reduced memory access and math. operations
- higher performance × 1.9 Gflop/s and × 3.2 Glup/s

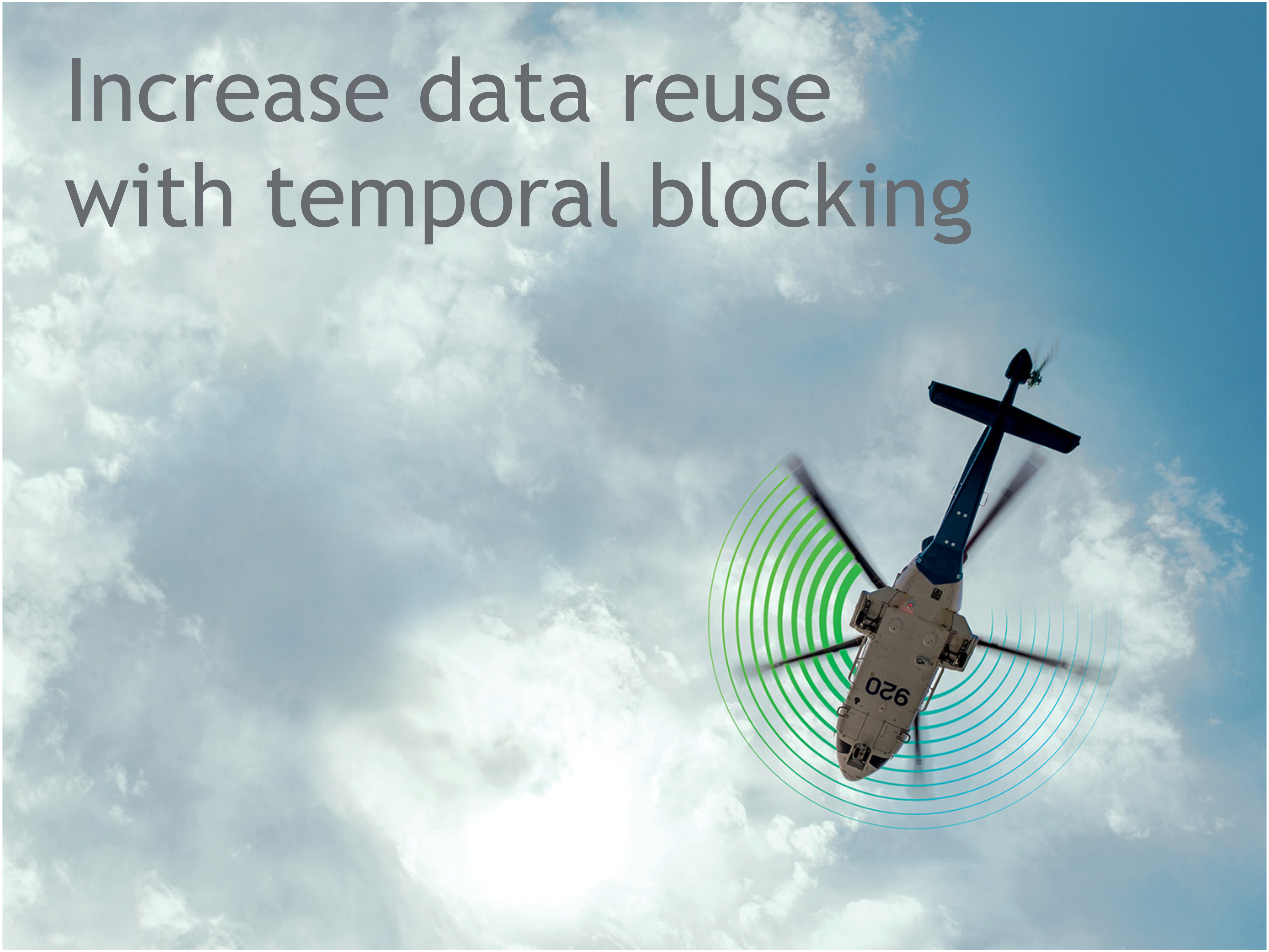
Anisotropic VTI vs Isotropic

- increased memory access and math. operations
- lower performance ÷ 2.2 Gflop/s and ÷ 2.2 Glup/s

Spatial blocking

Impact of the equation




Increase data reuse
with temporal blocking
920

Temporal blocking

4

Temporal blocking

Multicore wavefront + Diamond tiling (MWD)

Key concepts of MWD

Maximize date reuse: perform several time step updates before evicting data to main memory

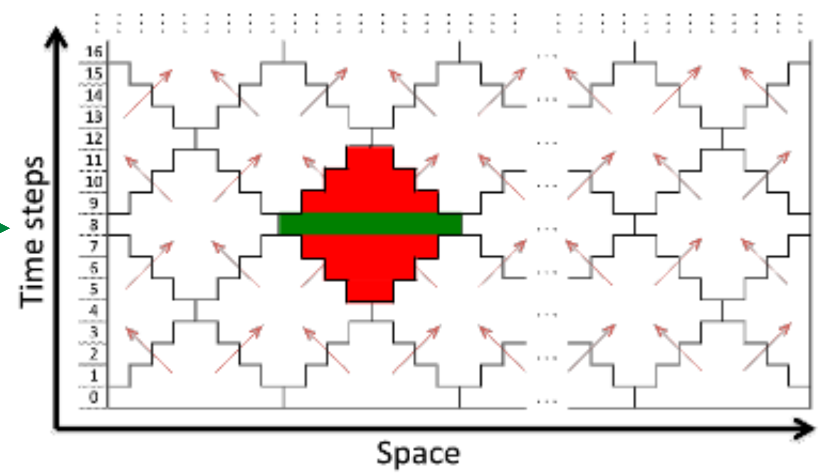
Space-time domain divided into diamonds

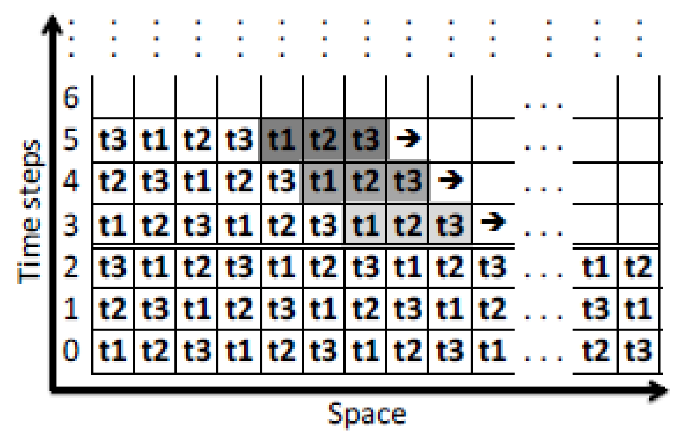
- Diamond slope S = 1/R (stencil radius)
- Low synchronization requirements
- Allow concurrent start
- High concurrency in transient state
- Unified shape for easier implementation

Diamond tiling can be combined with multi-thread wavefront update

Adjust concurrency and intra-diamond parallelism for optimal work balance

MWD for 1D FDM O2 (S=1)





Temporal blocking

Multicore wavefront + Diamond tiling (MWD)

3D implementation of MWD

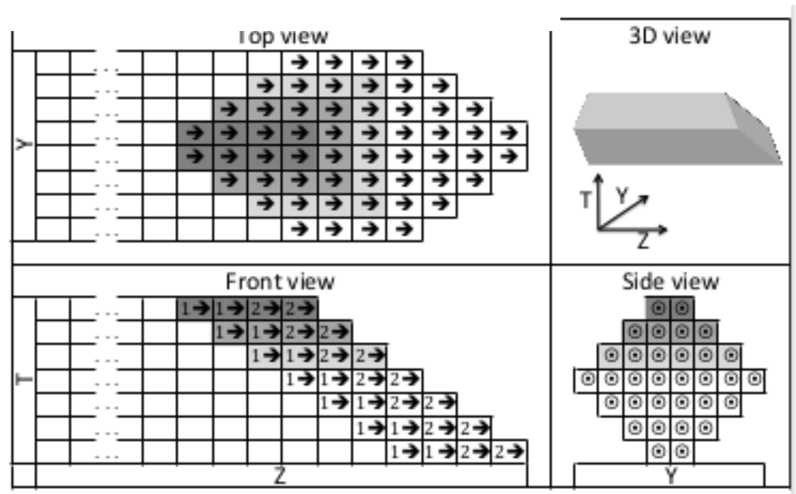
Efficient combination

- x-axis (n1/fast index) left unchanged for efficient vectorization
- Diamond tiling along y-axis (n2)
- Multi-thread wavefront along z-axis (n3)

Synchronization between diamonds

- FIFO queue with completed diamonds
- Critical OpenMP section for queue update

Optimal diamond size and number of threads for the wavefront update are determined by a auto-tuning procedure



Important notes

- No extra memory needed by MWD
- Wavefront allows local and simultaneous updates at various time steps
- Need to design specific data management when fixed time steps required (snapshots for RTM)

Temporal blocking

MWD vs pure spatial blocking

MWD configuration

- 4 threads per diamond
- 8 concurrent diamonds in parallel
- Diamond width = 32, height = 2

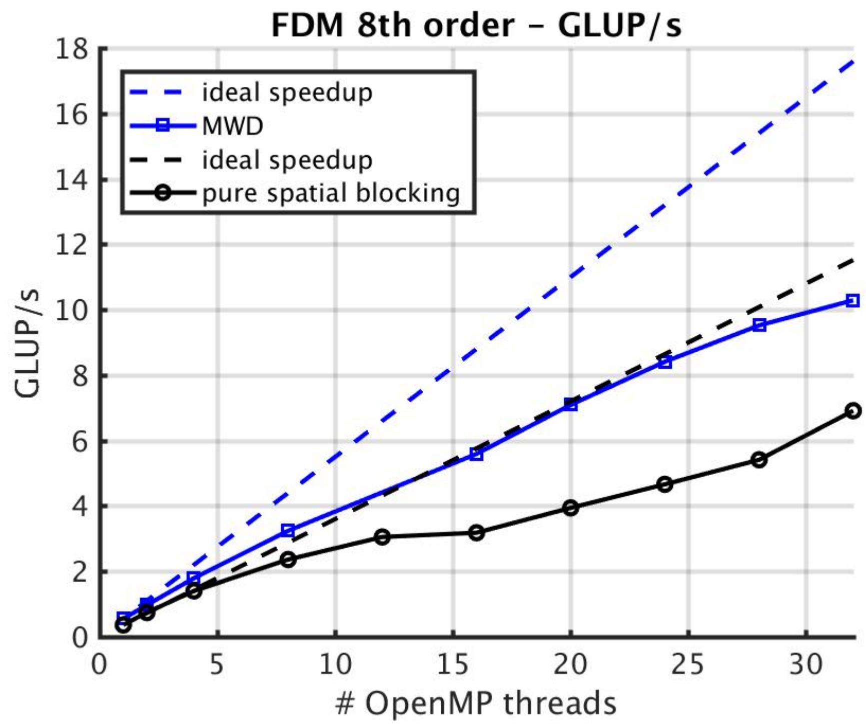
Pure spatial blocking configuration

- Cache blocs size = 16-5 (xy)
- No cache blocking in z

x1.5 speedup obtained with MWD

- Max 10.29 Glup/s with MWD
- Max 6.91 Glup/s with spatial blocking

Parallelism efficiency 60 % for both approaches on 32 OpenMP threads



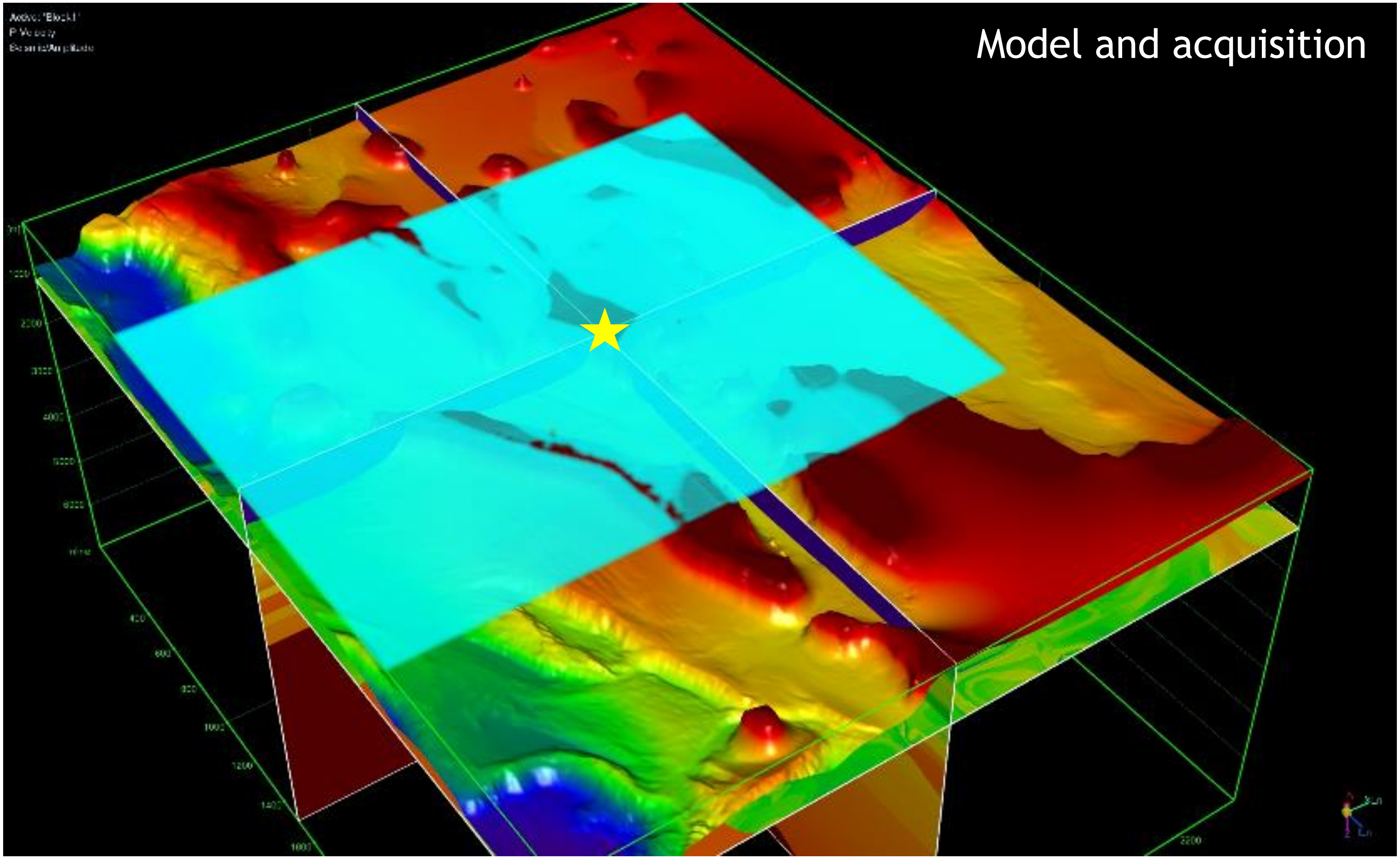
Scalability analysis from 1 to 32 threads

Intel Haswell 2 sockets x 16 cores

Application to seismic modeling and imaging

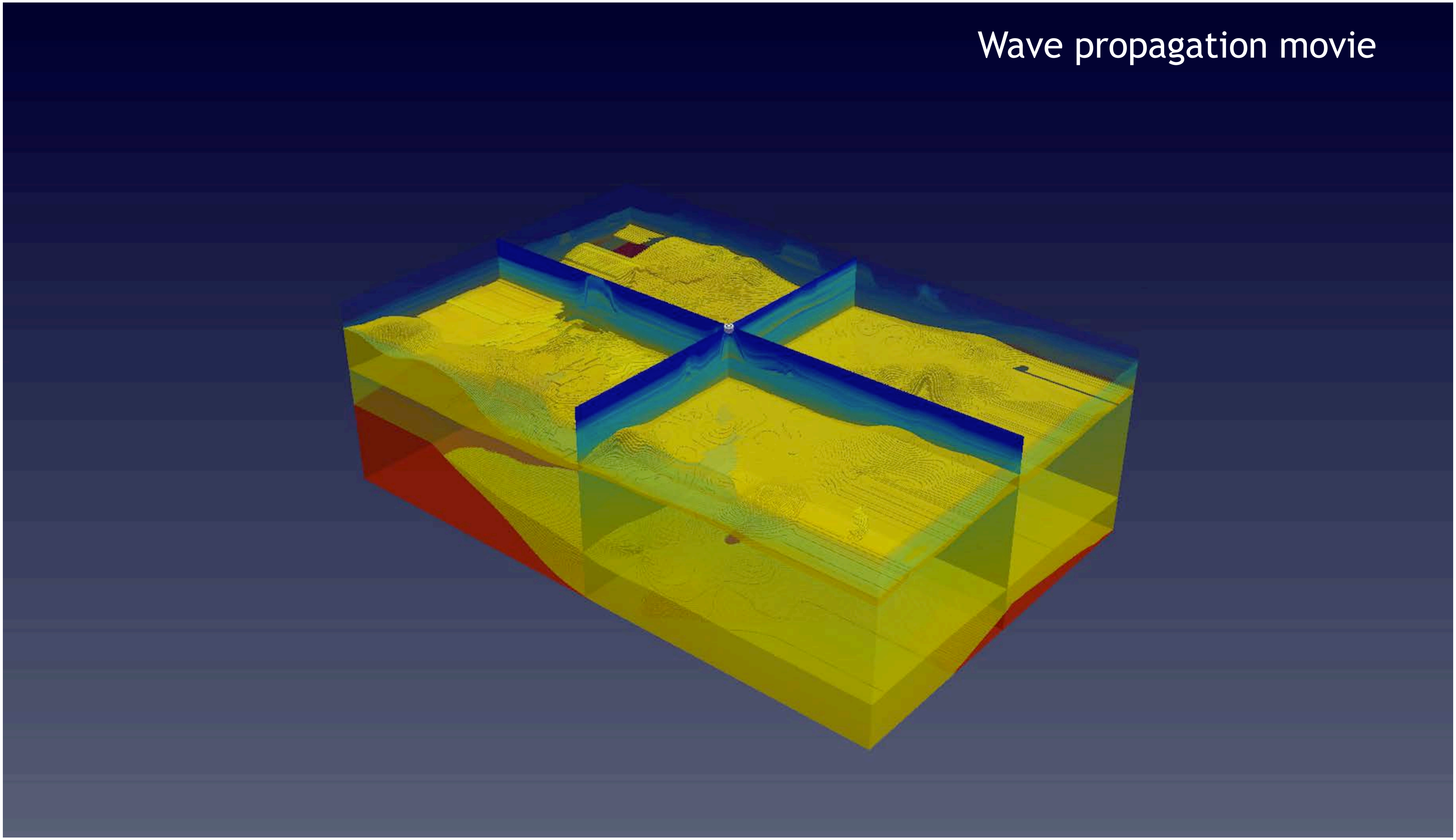
5

Seismic modeling in Offshore Saudi Arabia

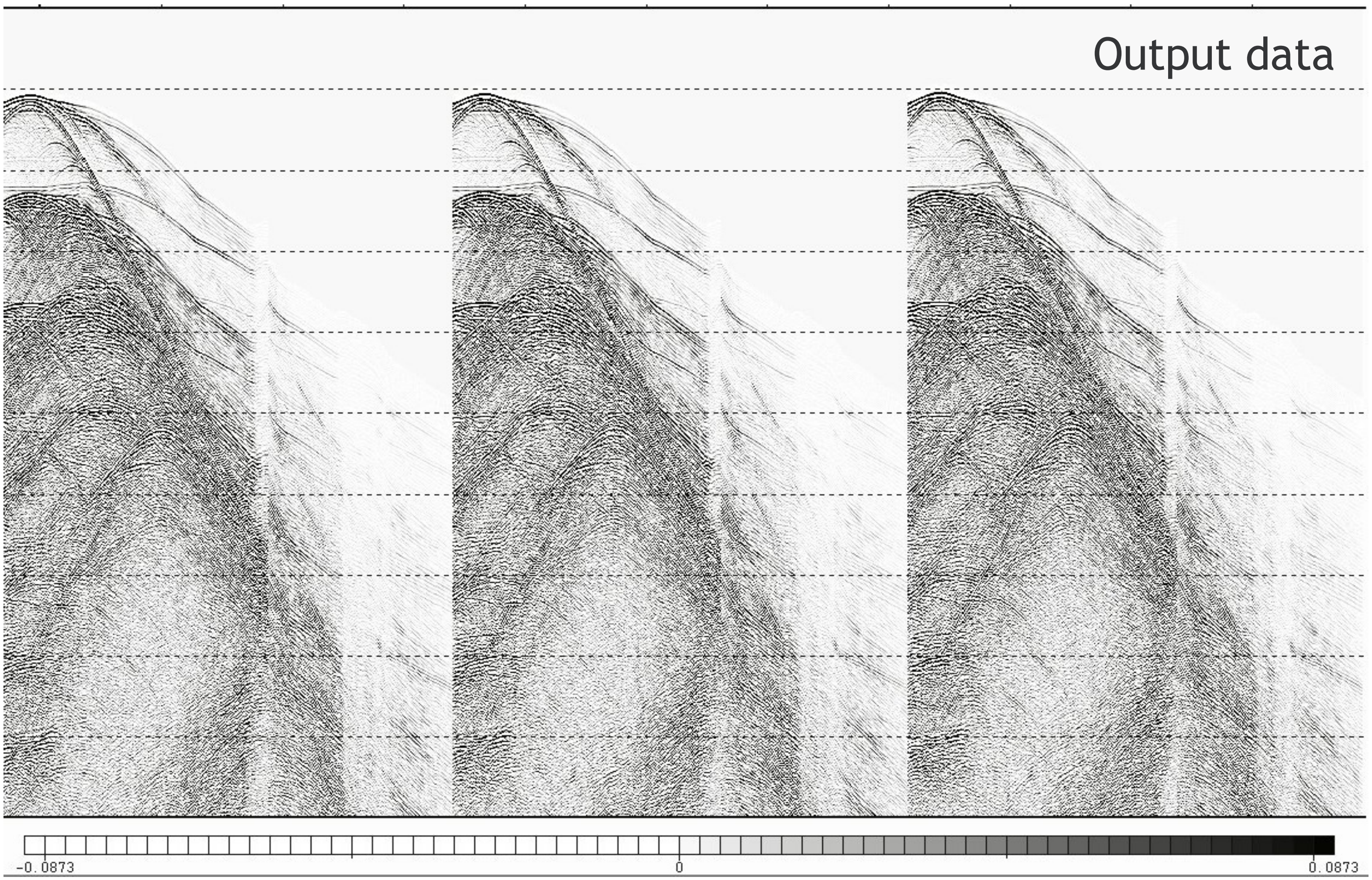


أرامكو السعودية
saudi aramco

Seismic modeling in Offshore Saudi Arabia



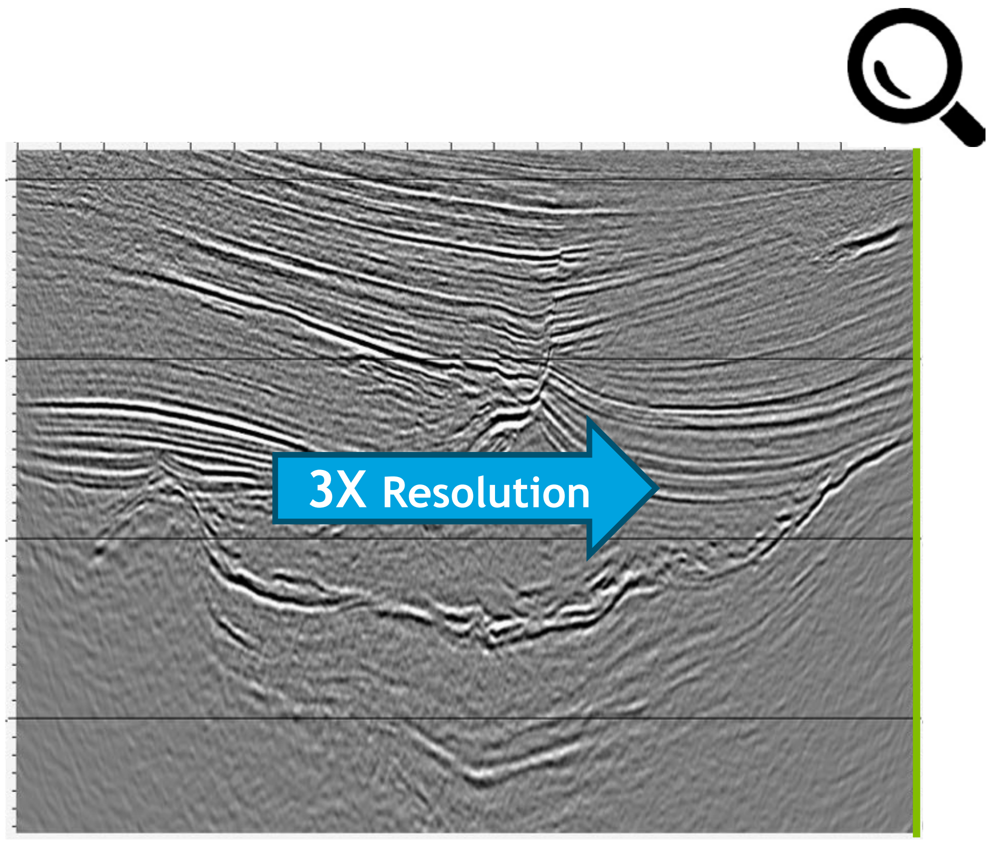
Seismic modeling in Offshore Saudi Arabia



On this application, we reached a peak performance of 1.2 Pflop/s on Shaheen

Seismic migration in Offshore Saudi Arabia

Benefit of supercomputers for seismic imaging



An industry first: 100 Hz reverse time migration

Conclusions

6

Conclusions

Summary

We presented highly optimized finite difference kernels integrated within a versatile platform tailored for seismic applications

The findings of this work concerning cache blocking:

- Pure spatial blocking allow for high performance but some bottlenecks do exist
- Spatial and temporal blocking (MWD) partially alleviate those issues and allow for a x1.5 speedup compared to pure spatial blocking

Achievements

- Application on large scale seismic surveys
- Acoustic modeling
- Acoustic reverse time migration at 100 Hz
- Excellent scalability on Shaheen up to full machine
- Peak performance 1.2 Pflop/s

Conclusions

Future work

Changing the wave equation from acoustic 3D…

$$\partial_t^2 p = c^2\left(\partial_x^2 p + \partial_y^2 p + \partial_z^2 p\right) \Longrightarrow \textbf{7 GLUP/s}$$

Conclusions

Future work

Changing the wave equation from acoustic 3D…

$$\partial_t^2 p = c^2\left(\partial_x^2 p + \partial_y^2 p + \partial_z^2 p\right) \Longrightarrow \textbf{7 GLUP/s}$$

…to elastic 3D

$$\begin{aligned}
\frac{\partial v_x(\mathbf{x},t)}{\partial t} &= \frac{1}{\rho(\mathbf{x})}\{\frac{\partial \sigma_{xx}(\mathbf{x},t)}{\partial x} + \frac{\partial \sigma_{xy}(\mathbf{x},t)}{\partial y} + \frac{\partial \sigma_{xz}(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial v_y(\mathbf{x},t)}{\partial t} &= \frac{1}{\rho(\mathbf{x})}\{\frac{\partial \sigma_{xy}(\mathbf{x},t)}{\partial x} + \frac{\partial \sigma_{yy}(\mathbf{x},t)}{\partial y} + \frac{\partial \sigma_{yz}(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial v_z(\mathbf{x},t)}{\partial t} &= \frac{1}{\rho(\mathbf{x})}\{\frac{\partial \sigma_{xz}(\mathbf{x},t)}{\partial x} + \frac{\partial \sigma_{yz}(\mathbf{x},t)}{\partial y} + \frac{\partial \sigma_{zz}(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial \sigma_{xx}(\mathbf{x},t)}{\partial t} &= (\lambda(\mathbf{x}) + 2\mu(\mathbf{x}))\frac{\partial v_x(\mathbf{x},t)}{\partial x} + \lambda(\mathbf{x})\{\frac{\partial v_y(\mathbf{x},t)}{\partial y} + \frac{\partial v_z(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial \sigma_{yy}(\mathbf{x},t)}{\partial t} &= (\lambda(\mathbf{x}) + 2\mu(\mathbf{x}))\frac{\partial v_y(\mathbf{x},t)}{\partial y} + \lambda(\mathbf{x})\{\frac{\partial v_x(\mathbf{x},t)}{\partial x} + \frac{\partial v_z(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial \sigma_{zz}(\mathbf{x},t)}{\partial t} &= (\lambda(\mathbf{x}) + 2\mu(\mathbf{x}))\frac{\partial v_z(\mathbf{x},t)}{\partial z} + \lambda(\mathbf{x})\{\frac{\partial v_x(\mathbf{x},t)}{\partial x} + \frac{\partial v_y(\mathbf{x},t)}{\partial y}\} \\
\frac{\partial \sigma_{xy}(\mathbf{x},t)}{\partial t} &= \mu(\mathbf{x})\{\frac{\partial v_x(\mathbf{x},t)}{\partial y} + \frac{\partial v_y(\mathbf{x},t)}{\partial x}\} \\
\frac{\partial \sigma_{xz}(\mathbf{x},t)}{\partial t} &= \mu(\mathbf{x})\{\frac{\partial v_x(\mathbf{x},t)}{\partial z} + \frac{\partial v_z(\mathbf{x},t)}{\partial x}\} \\
\frac{\partial \sigma_{yz}(\mathbf{x},t)}{\partial t} &= \mu(\mathbf{x})\{\frac{\partial v_y(\mathbf{x},t)}{\partial z} + \frac{\partial v_z(\mathbf{x},t)}{\partial y}\},
\end{aligned}
\Longrightarrow \textbf{0.5 GLUP/s}$$

Conclusions

Future work

Changing the wave equation from acoustic 3D…

$$\partial_t^2 p = c^2\left(\partial_x^2 p + \partial_y^2 p + \partial_z^2 p\right)$$

…to elastic 3D

$$
\begin{aligned}
\frac{\partial v_x(\mathbf{x},t)}{\partial t} &= \frac{1}{\rho(\mathbf{x})}\{\frac{\partial \sigma_{xx}(\mathbf{x},t)}{\partial x} + \frac{\partial \sigma_{xy}(\mathbf{x},t)}{\partial y} + \frac{\partial \sigma_{xz}(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial v_y(\mathbf{x},t)}{\partial t} &= \frac{1}{\rho(\mathbf{x})}\{\frac{\partial \sigma_{xy}(\mathbf{x},t)}{\partial x} + \frac{\partial \sigma_{yy}(\mathbf{x},t)}{\partial y} + \frac{\partial \sigma_{yz}(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial v_z(\mathbf{x},t)}{\partial t} &= \frac{1}{\rho(\mathbf{x})}\{\frac{\partial \sigma_{xz}(\mathbf{x},t)}{\partial x} + \frac{\partial \sigma_{yz}(\mathbf{x},t)}{\partial y} + \frac{\partial \sigma_{zz}(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial \sigma_{xx}(\mathbf{x},t)}{\partial t} &= (\lambda(\mathbf{x}) + 2\mu(\mathbf{x}))\frac{\partial v_x(\mathbf{x},t)}{\partial x} + \lambda(\mathbf{x})\{\frac{\partial v_y(\mathbf{x},t)}{\partial y} + \frac{\partial v_z(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial \sigma_{yy}(\mathbf{x},t)}{\partial t} &= (\lambda(\mathbf{x}) + 2\mu(\mathbf{x}))\frac{\partial v_y(\mathbf{x},t)}{\partial y} + \lambda(\mathbf{x})\{\frac{\partial v_x(\mathbf{x},t)}{\partial x} + \frac{\partial v_z(\mathbf{x},t)}{\partial z}\} \\
\frac{\partial \sigma_{zz}(\mathbf{x},t)}{\partial t} &= (\lambda(\mathbf{x}) + 2\mu(\mathbf{x}))\frac{\partial v_z(\mathbf{x},t)}{\partial z} + \lambda(\mathbf{x})\{\frac{\partial v_x(\mathbf{x},t)}{\partial x} + \frac{\partial v_y(\mathbf{x},t)}{\partial y}\} \\
\frac{\partial \sigma_{xy}(\mathbf{x},t)}{\partial t} &= \mu(\mathbf{x})\{\frac{\partial v_x(\mathbf{x},t)}{\partial y} + \frac{\partial v_y(\mathbf{x},t)}{\partial x}\} \\
\frac{\partial \sigma_{xz}(\mathbf{x},t)}{\partial t} &= \mu(\mathbf{x})\{\frac{\partial v_x(\mathbf{x},t)}{\partial z} + \frac{\partial v_z(\mathbf{x},t)}{\partial x}\} \\
\frac{\partial \sigma_{yz}(\mathbf{x},t)}{\partial t} &= \mu(\mathbf{x})\{\frac{\partial v_y(\mathbf{x},t)}{\partial z} + \frac{\partial v_z(\mathbf{x},t)}{\partial y}\},
\end{aligned}
$$

7 GLUP/s

The benefit of cache blocking technics is crucial to increase efficiency

0.5 GLUP/s

Acknowledgment and references

We would like to thank Saudi Aramco and KAUST for permission to present this work

Computations were done on KAUST's Shaheen II supercomputer

REFERENCES

Browne, S., Deane, C., Ho, G. and Mucci, P. [1999] PAPI: A Portable Interface to Hardware Performance Counters. *Proceedings of Department of Defense HPCMP Users Group Conference*.

Etienne, V., Tonellot, T., Thierry, P., Berthoumieux, V. and Andreolli, C. [2014] Optimization of the Seismic Modeling with the Time-Domain Finite-Difference Method. *SEG Annual Meeting, Expanded Abstracts, 3536-3540.*

Imbert, D., Imadoueddine, K., Thierry, P., Chauris, H. and Borges, L. [2011] Tips and tricks for finite difference and I/O-less FWI. *SEG Annual meeting, Expanded Abstracts, 3174-3178.*

Ilic, A., Pratas, F. and Sousa, L. [2014] Cache-aware Roofline model: Upgrading the loft. *IEEE Computer Architecture Letters*, vol. 13, n. 1, pp. 21-24.

Treibig, J., Hager, G. and Wellein, G. [2010]. Likwid: A lightweight performance-oriented tool suite for x86 multi-core environments. *39th International Conference on Parallel Processing Workshops. IEEE.*

Malas, T., Hager, G., Ltaief, H., Stengel, H., Wellein, G. and Keyes, D. [2015]. Multicore-optimized wavefront diamond blocking for optimizing stencil updates. *SIAM Journal on Scientific Computing 37 (4), C439-C464.*

أرامكو السعودية
saudi aramco