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

V. Etienne¹, T. Tonellot¹, K. Akbudak², H. Ltaief², S. Kortas³, T. Malas⁴, P. Thierry⁴, D. Keyes²

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
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
Introduction

Seismic modeling / Imaging / Inversion

Seismic modeling

Seismic Inversion

Seismic data

Seismic model

Seismic image

Amplitude Inversion

Seismic processing & Imaging
Introduction

Seismic modeling / Imaging / Inversion

For wave equation based methods the seismic modeling engine is a crucial element.
Introduction

Why seismic modeling is so important?

![Seismic imaging algorithms graph]

- **Normalized computational cost**
- **Seismic imaging algorithms**
  - Analytic
  - Rays
  - Ray gathers
  - Acoustic
  - Hi Res Acoustic
  - Elastic
  - Hi Res Elastic

- **Complex / low relief structures**
- **Simple Structures**
- **Increased quality / accuracy / resolution**
- **Elastic parameters**
- **Lithology**
Introduction

Why seismic modeling is so important?

Growth of acquired data

<table>
<thead>
<tr>
<th>Typical Acquisition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recv/Shot</td>
</tr>
<tr>
<td>Trace/Shot</td>
</tr>
<tr>
<td>Source density</td>
</tr>
<tr>
<td>Rec Time</td>
</tr>
<tr>
<td>Area</td>
</tr>
<tr>
<td>Num Shots</td>
</tr>
<tr>
<td>Size per Shot</td>
</tr>
<tr>
<td>Size per km²</td>
</tr>
<tr>
<td>Total Size</td>
</tr>
</tbody>
</table>
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.

- Seismic modeling
  - Forward problem
- Seismic inversion
  - Inverse problem

Model → HPC

Data
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
Seismic modeling with finite-difference

The 3D wave acoustic wave equation

\[ \partial_t^2 p = c^2 (\partial_x^2 p + \partial_y^2 p + \partial_z^2 p) \]

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 \( N \)th-order spatial operator to evaluate second derivative along index \( i \)

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

Typical discretization rules for optimal accuracy

- \( O_2 \rightarrow 10 \text{ pt/\( \lambda \)} \)
- \( O_4 \rightarrow 5 \text{ pt/\( \lambda \)} \)
- \( O_8 \rightarrow 4 \text{ pt/\( \lambda \)} \)
- \( O_{12} \rightarrow 3.5 \text{ pt/\( \lambda \)} \)
- \( O_{16} \rightarrow 3.2 \text{ pt/\( \lambda \)} \)

No universal scheme
Application dependent

Coarse grid \( \rightarrow \) Fine grid

FWI

RTM
Seismic modeling with finite-difference

Flexible implementation

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

2D

Acoustic

Isotropic

1st wave eq

02

04

08

012

016

3D

Elastic

VTI

TTI

Full tensor

2nd wave eq

dimension

physics

approximation

discrete equation

FDM stencil

Flexible implementation

Seismic modeling with finite-difference

Follow an object oriented design
Recast seismic modeling into the objects framework
Seismic modeling with finite-difference

Scenario 1
RTM with marine acquisition
Seismic modeling with finite-difference

Scenario 2
FWI with land acquisition

- 2D
- Acoustic
- Isotropic
- VTI
- TTI
- Full tensor
- 1st wave eq
- 2nd wave eq
- O2
- O4
- O8
- O12
- O16

- 3D
- Elastic
- physics
- dimension
- approximation
- discrete equation
- FDM operator (accuracy)
Seismic modeling with finite-difference

Flexible implementation

modeling
Solve() virtual

FEM?
FDM

FDM_2D
FDM_3D

FDM_3D_el_iso_1st
Arrays
Kernel
CPML
Solve() Impl.

FDM_3D_ac_vti_1st
Arrays
Kernel
CPML
Solve() Impl.

The object factory + Template metaprogramming
Seismic modeling with finite-difference

Efficient boundary implementation

model

CPML

standard FDM kernel

CPML memory variables and computation decoupled
Spatial blocking
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**

**ISO 2ND Gflop/s**

- 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)

**ISO 2ND Glup/s**

**Speedup for equivalent accuracy**
Spatial blocking

**Fine performance analysis with cache-aware roofline model**

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

---

**Cache-aware roofline model**

- **L1 5512 GB/s**
- **L2 1520 GB/s**
- **L3 521 GB/s**
- **DRAM 128 GB/s**

**Gflop/s vs. Arithmetic intensity**

**FDM O2**
- 132 Gflop/s

**FDM O8**
- 329 Gflop/s

**Derivatives of P[ix][iy][iz]**
- n3
  - DRAM – perf. stable
  - n1 (fast index)
  - n2
  - L2 - highest speedup $\times 5.6$
  - DRAM - perf. stable
  - pressure update

---

**saudi aramco**
Spatial blocking

Impact of the equation

Isotropic 2\textsuperscript{nd} vs 1\textsuperscript{st} order wave equation
• reduced memory access and math. operations
• higher performance $\times 1.9$ Gflop/s and $\times 3.2$ Glup/s

Anisotropic VTI vs Isotropic
• increased memory access and math. operations
• lower performance $\div 2.2$ Gflop/s and $\div 2.2$ Glup/s
Spatial blocking

Impact of the equation

![Graphs showing speedup versus number of threads for different blocking scenarios: AC_ISO_1ST, AC_ISO_2ND, AC_VTL_1ST, AC_VTL_2ND.](image)
Increase data reuse with temporal blocking
Temporal blocking
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 an 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
Seismic modeling in Offshore Saudi Arabia
Seismic modeling in Offshore Saudi Arabia

Wave propagation movie
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
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...

\[ \partial_t^2 p = c^2 (\partial_x^2 p + \partial_y^2 p + \partial_z^2 p) \]

7 GLUP/s
Conclusions

Future work
Changing the wave equation from acoustic 3D...

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

...to elastic 3D

\[
\begin{align*}
\frac{\partial v_x(x, t)}{\partial t} &= \frac{1}{\rho(x)} \left\{ \frac{\partial \sigma_{xx}(x, t)}{\partial x} + \frac{\partial \sigma_{xy}(x, t)}{\partial y} + \frac{\partial \sigma_{xz}(x, t)}{\partial z} \right\} \\
\frac{\partial v_y(x, t)}{\partial t} &= \frac{1}{\rho(x)} \left\{ \frac{\partial \sigma_{yx}(x, t)}{\partial x} + \frac{\partial \sigma_{yy}(x, t)}{\partial y} + \frac{\partial \sigma_{yz}(x, t)}{\partial z} \right\} \\
\frac{\partial v_z(x, t)}{\partial t} &= \frac{1}{\rho(x)} \left\{ \frac{\partial \sigma_{zx}(x, t)}{\partial x} + \frac{\partial \sigma_{zy}(x, t)}{\partial y} + \frac{\partial \sigma_{zz}(x, t)}{\partial z} \right\} \\
\frac{\partial \sigma_{xx}(x, t)}{\partial t} &= (\lambda(x) + 2\mu(x)) \frac{\partial v_x(x, t)}{\partial x} + \lambda(x) \left\{ \frac{\partial v_y(x, t)}{\partial y} + \frac{\partial v_z(x, t)}{\partial z} \right\} \\
\frac{\partial \sigma_{yy}(x, t)}{\partial t} &= (\lambda(x) + 2\mu(x)) \frac{\partial v_y(x, t)}{\partial y} + \lambda(x) \left\{ \frac{\partial v_x(x, t)}{\partial x} + \frac{\partial v_z(x, t)}{\partial z} \right\} \\
\frac{\partial \sigma_{zz}(x, t)}{\partial t} &= (\lambda(x) + 2\mu(x)) \frac{\partial v_z(x, t)}{\partial z} + \lambda(x) \left\{ \frac{\partial v_x(x, t)}{\partial x} + \frac{\partial v_y(x, t)}{\partial y} \right\} \\
\frac{\partial \sigma_{xy}(x, t)}{\partial t} &= \mu(x) \left\{ \frac{\partial v_x(x, t)}{\partial y} + \frac{\partial v_y(x, t)}{\partial x} \right\} \\
\frac{\partial \sigma_{xz}(x, t)}{\partial t} &= \mu(x) \left\{ \frac{\partial v_x(x, t)}{\partial z} + \frac{\partial v_z(x, t)}{\partial x} \right\} \\
\frac{\partial \sigma_{yz}(x, t)}{\partial t} &= \mu(x) \left\{ \frac{\partial v_y(x, t)}{\partial z} + \frac{\partial v_z(x, t)}{\partial y} \right\} 
\end{align*}
\]

7 GLUP/s

0.5 GLUP/s
Conclusions

Future work
Changing the wave equation from acoustic 3D...

\[
\partial_t^2 p = c^2 (\partial_x^2 p + \partial_y^2 p + \partial_z^2 p)
\]

...to elastic 3D

\[
\begin{align*}
\frac{\partial v_x(x,t)}{\partial t} &= \frac{1}{\rho(x)} \left\{ \frac{\partial \sigma_{xx}(x,t)}{\partial x} + \frac{\partial \sigma_{xy}(x,t)}{\partial y} + \frac{\partial \sigma_{xz}(x,t)}{\partial z} \right\} \\
\frac{\partial v_y(x,t)}{\partial t} &= \frac{1}{\rho(x)} \left\{ \frac{\partial \sigma_{xy}(x,t)}{\partial x} + \frac{\partial \sigma_{yy}(x,t)}{\partial y} + \frac{\partial \sigma_{yz}(x,t)}{\partial z} \right\} \\
\frac{\partial v_z(x,t)}{\partial t} &= \frac{1}{\rho(x)} \left\{ \frac{\partial \sigma_{xz}(x,t)}{\partial x} + \frac{\partial \sigma_{yz}(x,t)}{\partial y} + \frac{\partial \sigma_{zz}(x,t)}{\partial z} \right\} \\
\frac{\partial \sigma_{xx}(x,t)}{\partial t} &= (\lambda(x) + 2\mu(x)) \frac{\partial v_x(x,t)}{\partial x} + \lambda(x) \left\{ \frac{\partial v_y(x,t)}{\partial y} + \frac{\partial v_z(x,t)}{\partial z} \right\} \\
\frac{\partial \sigma_{yy}(x,t)}{\partial t} &= (\lambda(x) + 2\mu(x)) \frac{\partial v_y(x,t)}{\partial y} + \lambda(x) \left\{ \frac{\partial v_x(x,t)}{\partial x} + \frac{\partial v_z(x,t)}{\partial z} \right\} \\
\frac{\partial \sigma_{zz}(x,t)}{\partial t} &= (\lambda(x) + 2\mu(x)) \frac{\partial v_z(x,t)}{\partial z} + \lambda(x) \left\{ \frac{\partial v_x(x,t)}{\partial x} + \frac{\partial v_y(x,t)}{\partial y} \right\} \\
\frac{\partial \sigma_{xy}(x,t)}{\partial t} &= \mu(x) \left\{ \frac{\partial v_x(x,t)}{\partial y} + \frac{\partial v_y(x,t)}{\partial x} \right\} \\
\frac{\partial \sigma_{xz}(x,t)}{\partial t} &= \mu(x) \left\{ \frac{\partial v_x(x,t)}{\partial z} + \frac{\partial v_z(x,t)}{\partial x} \right\} \\
\frac{\partial \sigma_{yz}(x,t)}{\partial t} &= \mu(x) \left\{ \frac{\partial v_y(x,t)}{\partial z} + \frac{\partial v_z(x,t)}{\partial y} \right\},
\end{align*}
\]

The benefit of cache blocking technics is crucial to increase efficiency.

7 GLUP/s

0.5 GLUP/s
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


