Computational Challenges in Global Seismic Tomography

Matthieu Lefebvre[1], Ebru Bozdağ[2], Dimitri Komatitsch[3],
Daniel Peter[4], and Jeroen Tromp[1]

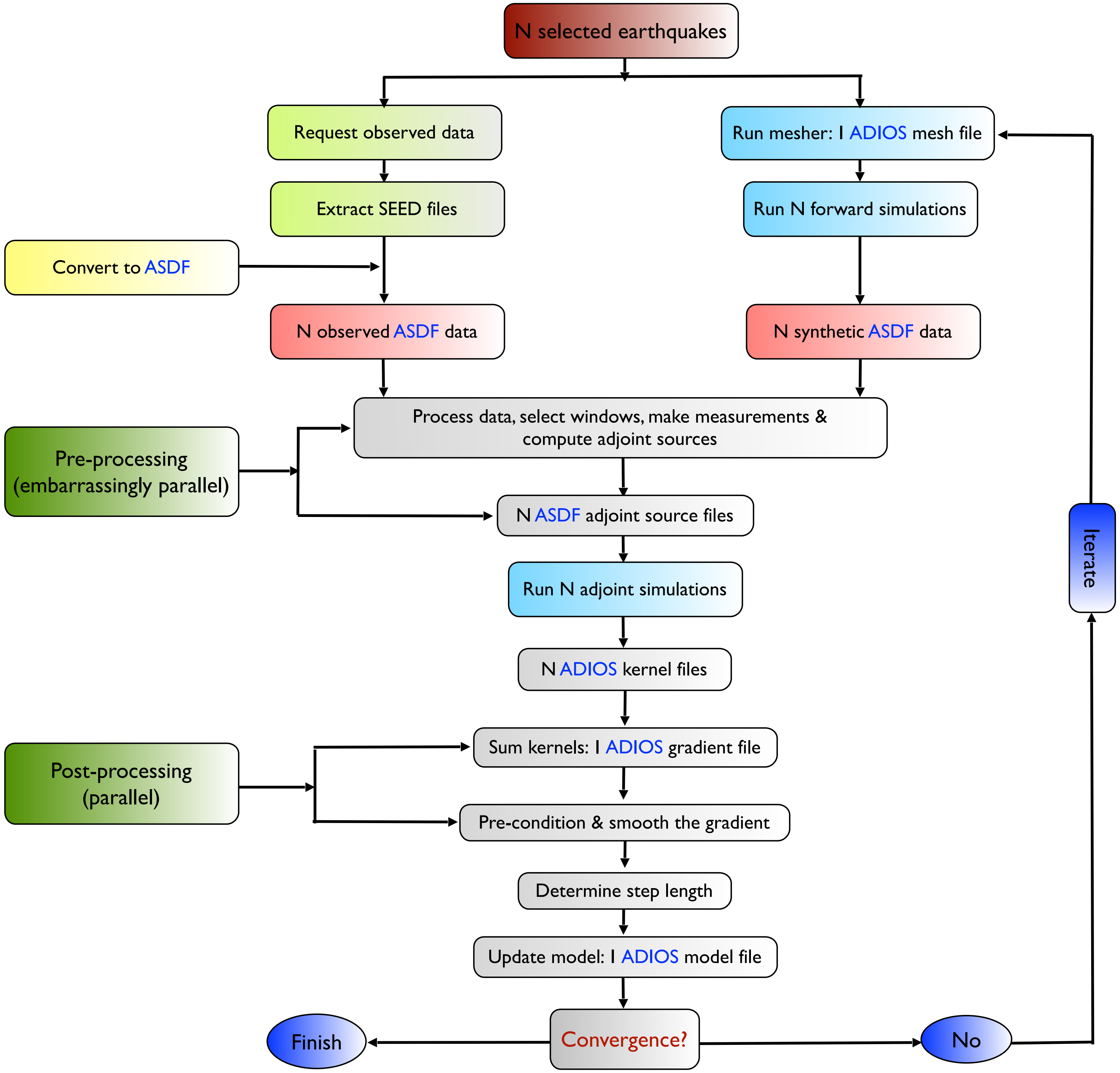
[1]Princeton University
[2]Université de Nice Sophia Antipolis
[3]ETH Zürich
[4]CNRS

2015 OLCF Users Meeting: Reaching for the Summit Together
June 24, 2015

Leadership Computing Projects

- INCITE
    - 2013 - 2014: 100 million core-hours/year on Titan
    - 2015 - 2017: 50 million core-hours/year on Titan
- CAAR
    - 2015 - 2017: Preparation for Summit

Workflow Overview



N selected earthquakes

Request observed data

Extract SEED files

Convert to ASDF

N observed ASDF data

Run mesher: 1 ADIOS mesh file

Run N forward simulations

N synthetic ASDF data

Pre-processing (embarrassingly parallel)

Process data, select windows, make measurements & compute adjoint sources

N ASDF adjoint source files

Run N adjoint simulations

N ADIOS kernel files

Post-processing (parallel)

Sum kernels: 1 ADIOS gradient file

Pre-condition & smooth the gradient

Determine step length

Update model: 1 ADIOS model file

Convergence?

Finish

No

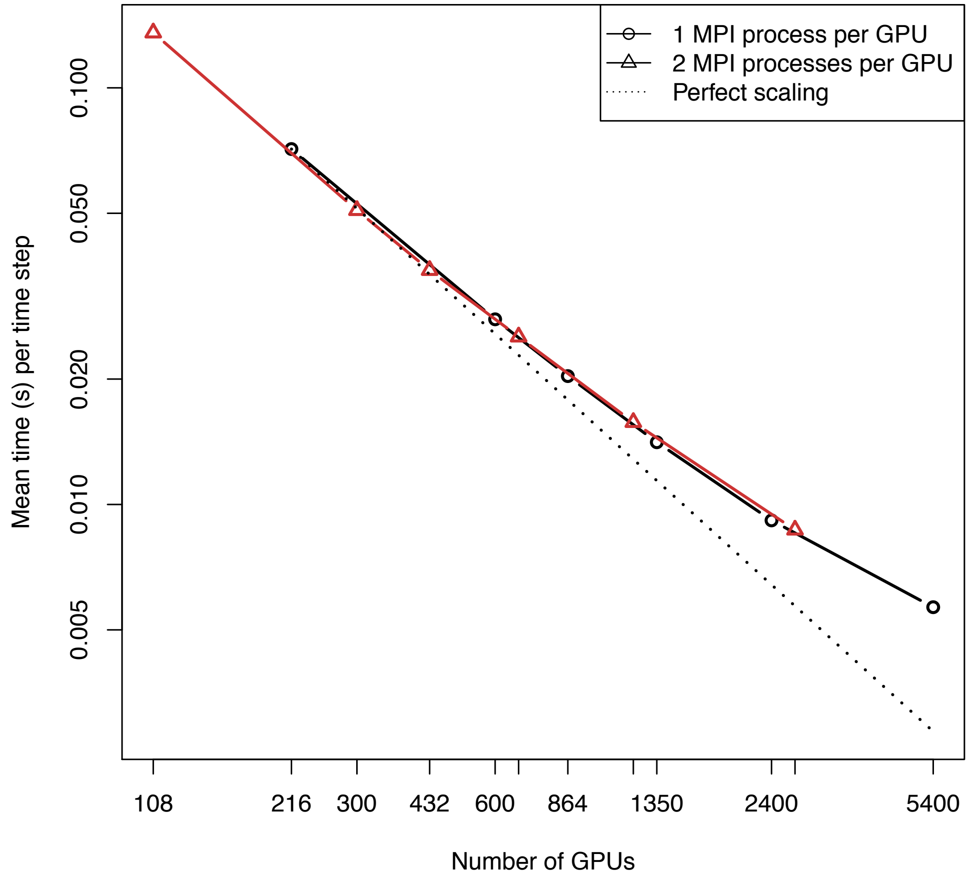
Iterate

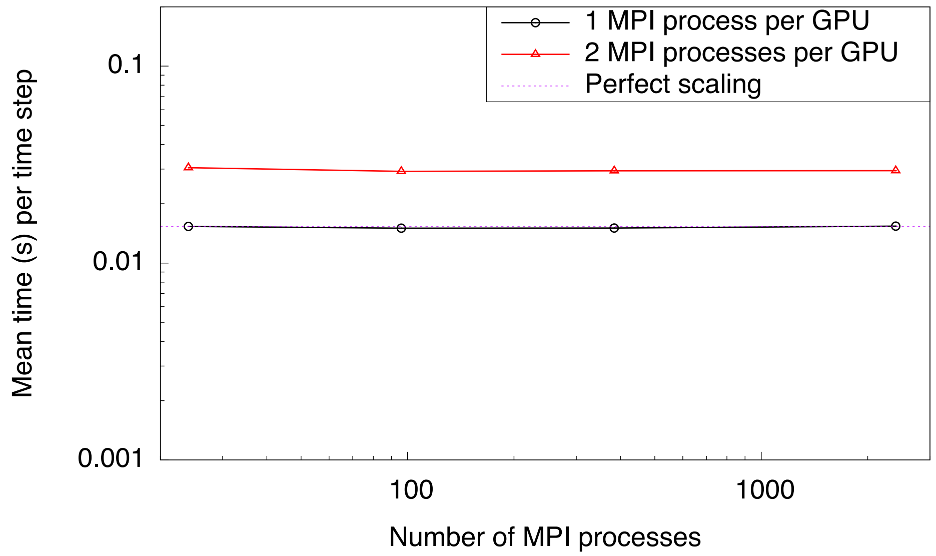
Ebru Bozdağ

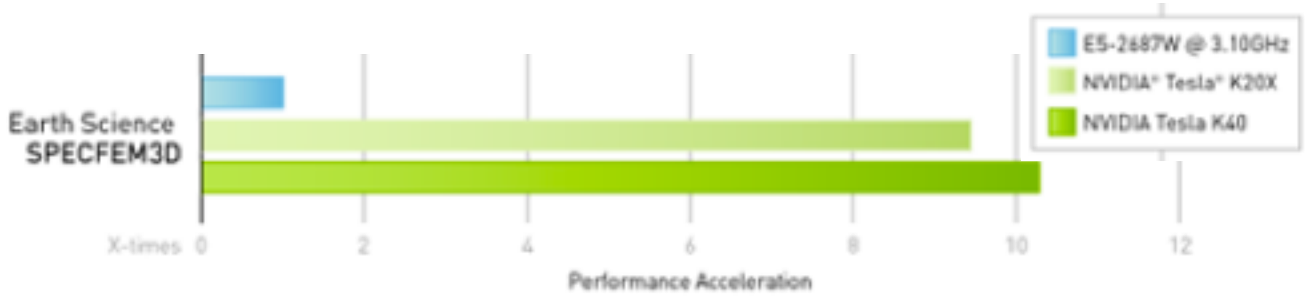
Solver: SPECFEM3D_GLOBE

- Global version of SPECFEM3D
    - Spectral-element method
    - Forward and adjoint capabilities
    - Multiple simulations in one run
    - Open-source software
    - 16 recent developers, overall contributions from ~50 people
    - Programming model:
        - Fortran + MPI
        - GPU: CUDA + OpenCL (via BOAST)
        - Some OpenMP
- Used on a range of supercomputers (Titan, Piz Daint, Curie, Fermi, SuperMUC, …)
- **Accounts for more than 90% of the workflow's computational time**

Solver: Performance







http://www.nvidia.com/object/gpu-test-drive.html

February 05, 2013

Four Applications Sustain One Petaflop on Blue Waters

HPC wire

July 18, 2012

Researchers Squeeze GPU Performance from 11 Big Science Apps

Solver: GPU Portability

- Initial implementation: CUDA
  - In collaboration with NVIDIA (Peter Messmer)
- Current implementation: BOAST
  - *Bringing Optimization through Automatic Source-to-Source Transformations*
  - Kernels written in Ruby
  - Generates CUDA and OpenCL
  - Calls to kernels in C
  - Tuned for Fermi and Kepler architectures

Reaching for the Summit:

Re-profile the code (calculations + transfers)
Prepare the code for future GPUs
CPU-GPU data transfers — Unified memory
Portability: BOAST vs. OpenMP 4

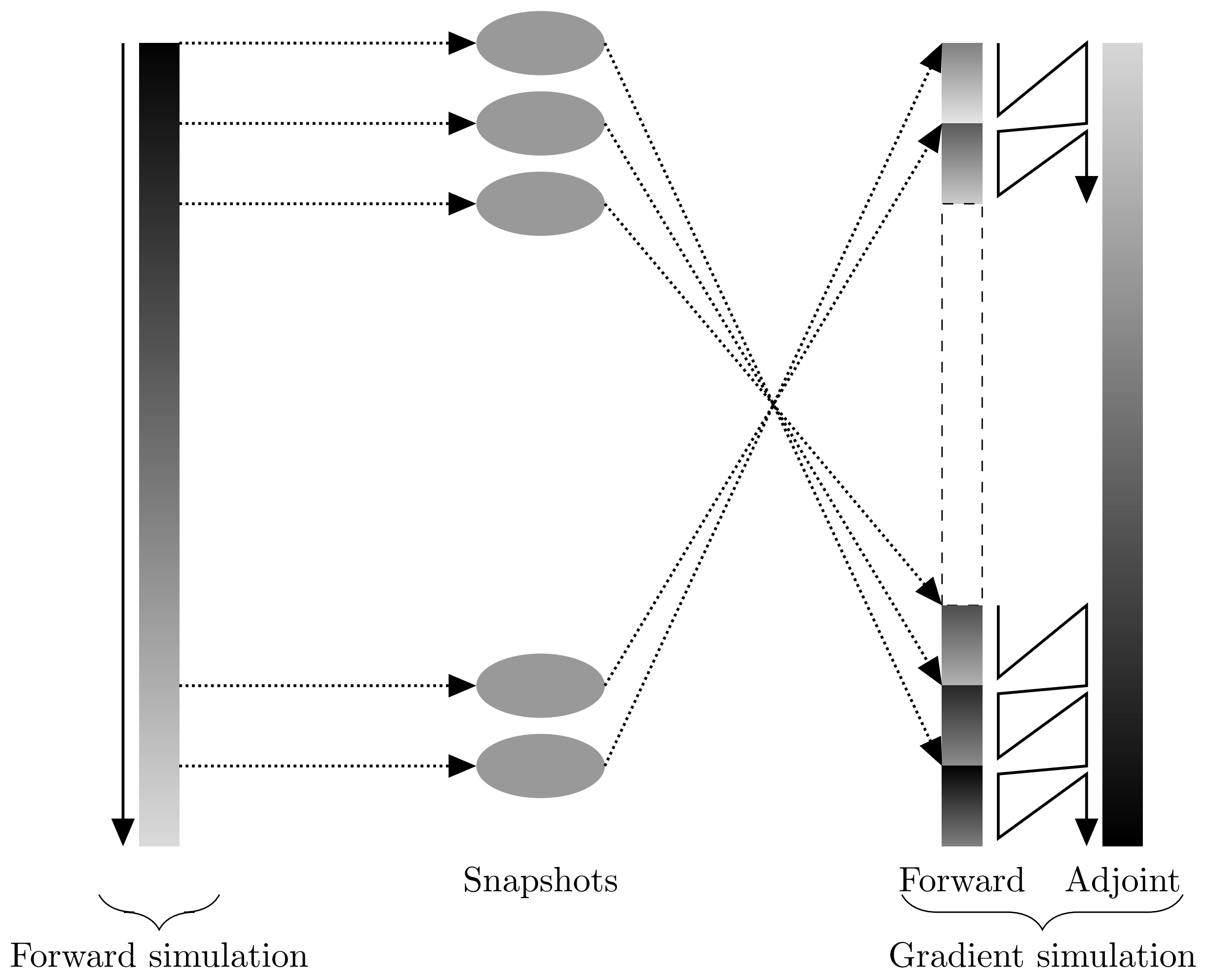
Solver: I/O

- Initial POSIX I/O: 1 file per MPI process and per variable
- ADIOS I/O:
  - Collaboration with Norbert Podhorszki (ORNL)
  - Metadata gives access to data, even when located in the middle of a file
  - Transparent optimization through transport methods
    - Large scale simulations: MPI_AGGREGATE
    - Inversion scale simulations: POSIX

Mesh region	Output Size	Spider (GB/s)	Atlas (GB/s)
Crust-Mantle	2,548	14.3	40.6
Outer core	317	7.4	8.47
Inner core	177	4.8	7.6

Bandwidth for SPECFEM3D_GLOBE output using the ADIOS MPI_AGGREGATE transport method for a 4.3 second resolution simulation using 24,576 MPI tasks. Results are presented both for the old (Spider) and new (Atlas) OLCF filesystems. Numbers for different regions show that large files benefit most from use of the ADIOS library.

Solver: Attenuation Snaphots

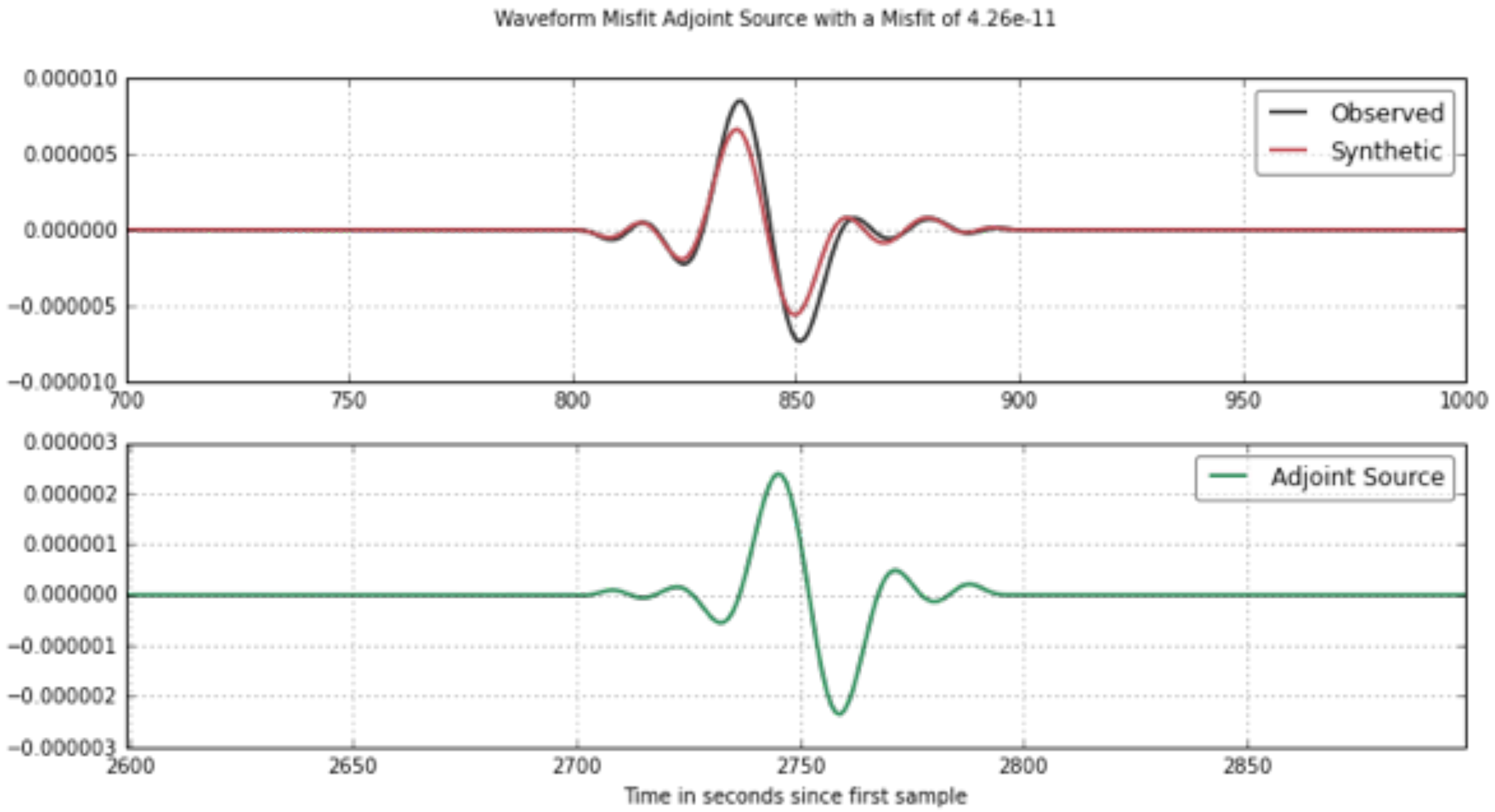


- 50+ GB snapshots (17s resolution)
- Output frequency depends on available memory
- Algorithmic improvements:
  - Coarse-grained memory approach
- Computational improvements:
  - Data-staging
  - Intermediate buffering

Reaching for the Summit:

Rely on ADIOS for the right transport method
Reduce the cost of accessing snapshots
Data pre-fetching, asynchronous writes
Use of an alternate memory area (e.g. burst buffer — additional nodes)

Data assimilation



- Legacy:
    - based on SAC tools
    - bash and Fortran
    - SAC ASCII files

- Current work: pyAdjoint
    - based on Obpsy
    - python library
    - ASDF files

Lion Krischer, Wenjie Lei, James Smith

ASDF: an Adaptable Seismic Data Format

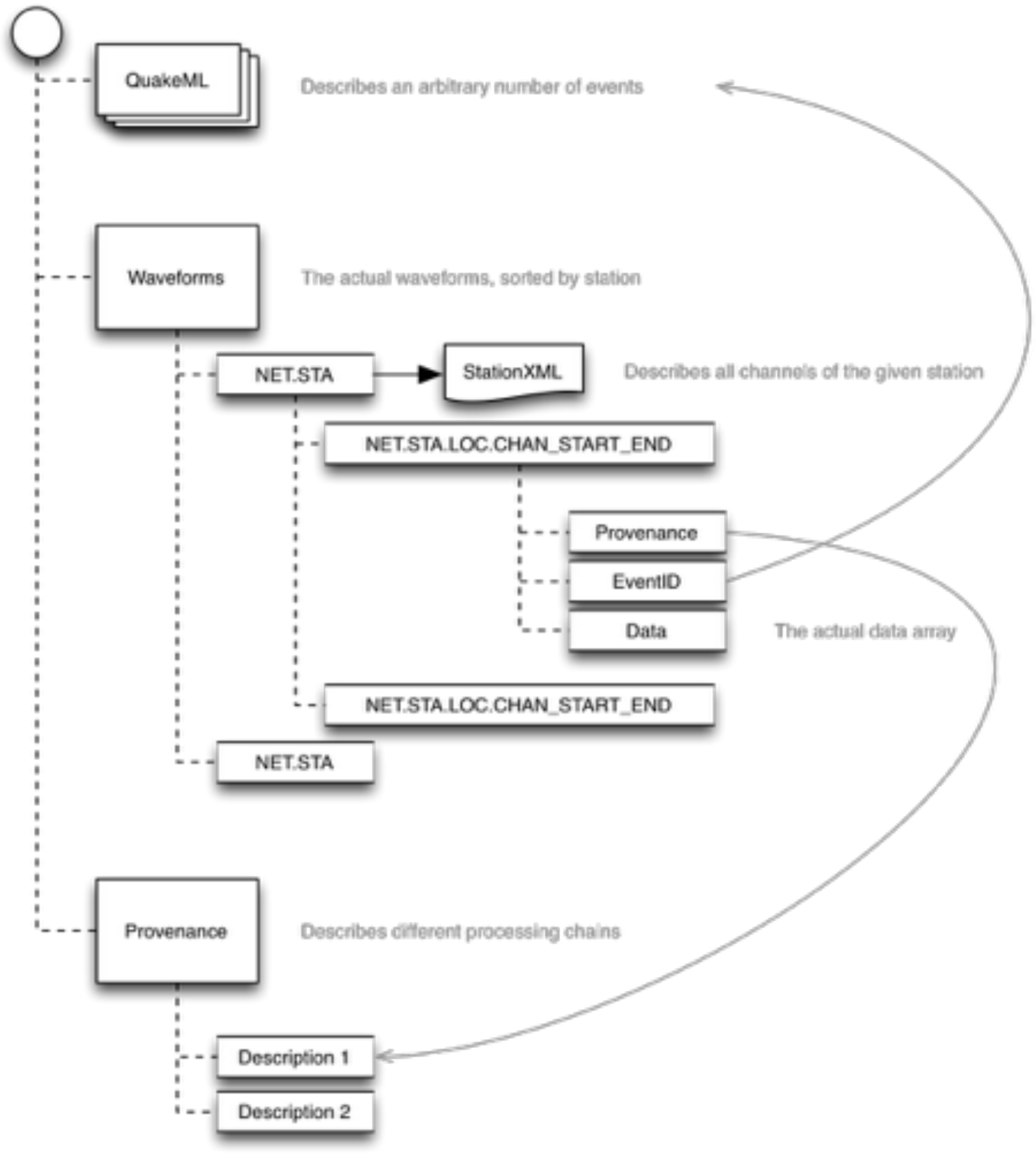
- Collaboration involving Princeton University, Munich University (ObsPy) and Oak Ridge National Laboratory
- Combine all the time series for a single shot or earthquake into one file
- Take advantage of parallel processing
- Use modern file format as container (HDF5)
- Store provenance inside the file for reproducibility
- Use existing standards when possible (e.g., XML)
- Two implementations: Python and C / Fortran

  https://github.com/SeismicData/pyasdf

  https://github.com/SeismicData/asdf-library

James Smith, Lion Krischer

ASDF: Structure



QuakeML

Describes an arbitrary number of events

Waveforms

The actual waveforms, sorted by station

NET.STA

StationXML

Describes all channels of the given station

NET.STA.LOC.CHAN_START_END

Provenance

EventID

Data

The actual data array

NET.STA.LOC.CHAN_START_END

NET.STA

Provenance

Describes different processing chains

Description 1

Description 2

1000 Stations	Number of SAC Files	Number of ADIOS Files
255 Earthquakes	1,275,00	255
6,000 Earthquakes	30,000,000	6,000

James Smith, Lion Krischer

ASDF: Reproducibility

- Current scientific publications provide an explanation of what the experiment does, why it matters, and what the results are
- Encourages collaboration and sharing of data/methods
- SEIS_PROV: domain specific extension for W3C_PROV in the context of seismological data processing and generation
    - A scientists looking at data described by SEIS_PROV should be able to tell what steps were taken to generate this particular piece of data
    - Entities: data (waveform, adjoint source, cross-correlation)
    - Activities: changes entities (filter)
    - Agents: software responsible (specfem3d_globe, obspy)

James Smith, Lion Krischer

Reaching for the Summit:

Tests: stability, scalability
Integration in the inversion workflow

Workflow Management

- Current inversion process steered by user controlled bash scripts
- Automation is critical for reliability and productivity
    - In particular with the twentyfold increase in data to be assimilated
- Requirements:
    - Switch the focus to science
    - Least action
        - Automatically deal with job scheduling, clustering, resilience
        - User interaction only when required (e.g. intermediate visualization)
    - High abstraction level
        - Computational details should be hidden

Workflow: Seisflow

- Super-script rather than a real workflow manager
- Object-oriented approach
  - Defines base class for every step
  - Different approaches are implemented in derived classes
- Implemented in Python
- Sometimes, reinvents the wheel
  - Job generators for PBS, Slurm
- Efficient for toy problems

https://github.com/PrincetonUniversity/seisflows

Ryan Modrak

Workflow: Pegasus

- Taking advantage of work done by workflow management experts
  - Job management
  - Job clustering
  - Data Management
  - Fault resilience
  - Distribute tasks on appropriate systems
    - Simulations, pre-processing, post-processing, visualization require different types of resources

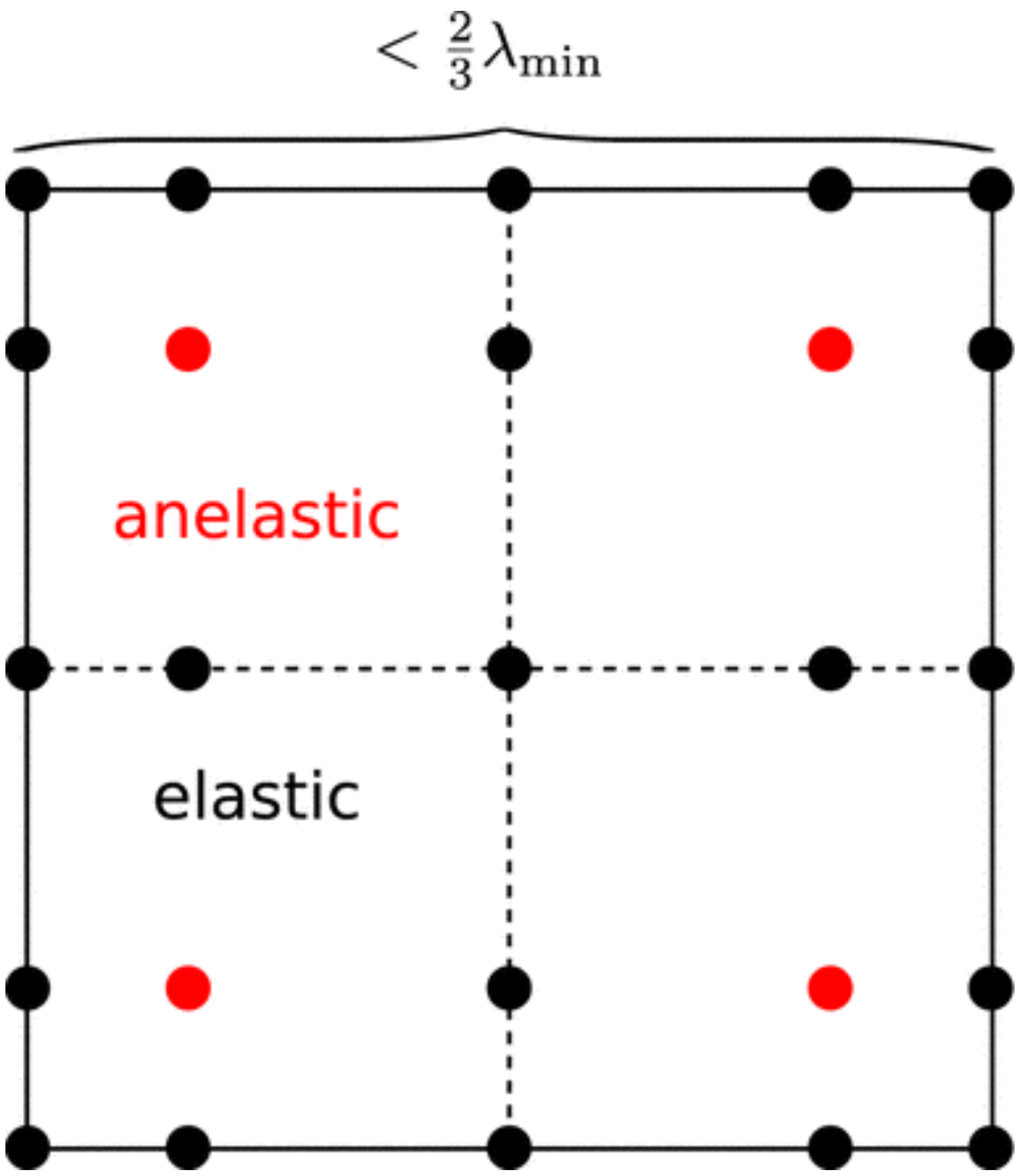
Conclusion

Reaching for the Summit:

Workflow pieces are in place
Reproducibility is an increasing concern in modern seismology
Focus is on data and workflow management

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

Coarse-grain memory



Martin van Driel and Tarje Nissen-Meyer
Optimized viscoelastic wave propagation for weakly dissipative media
Geophys. J. Int. 2014 199: 1078-1093.