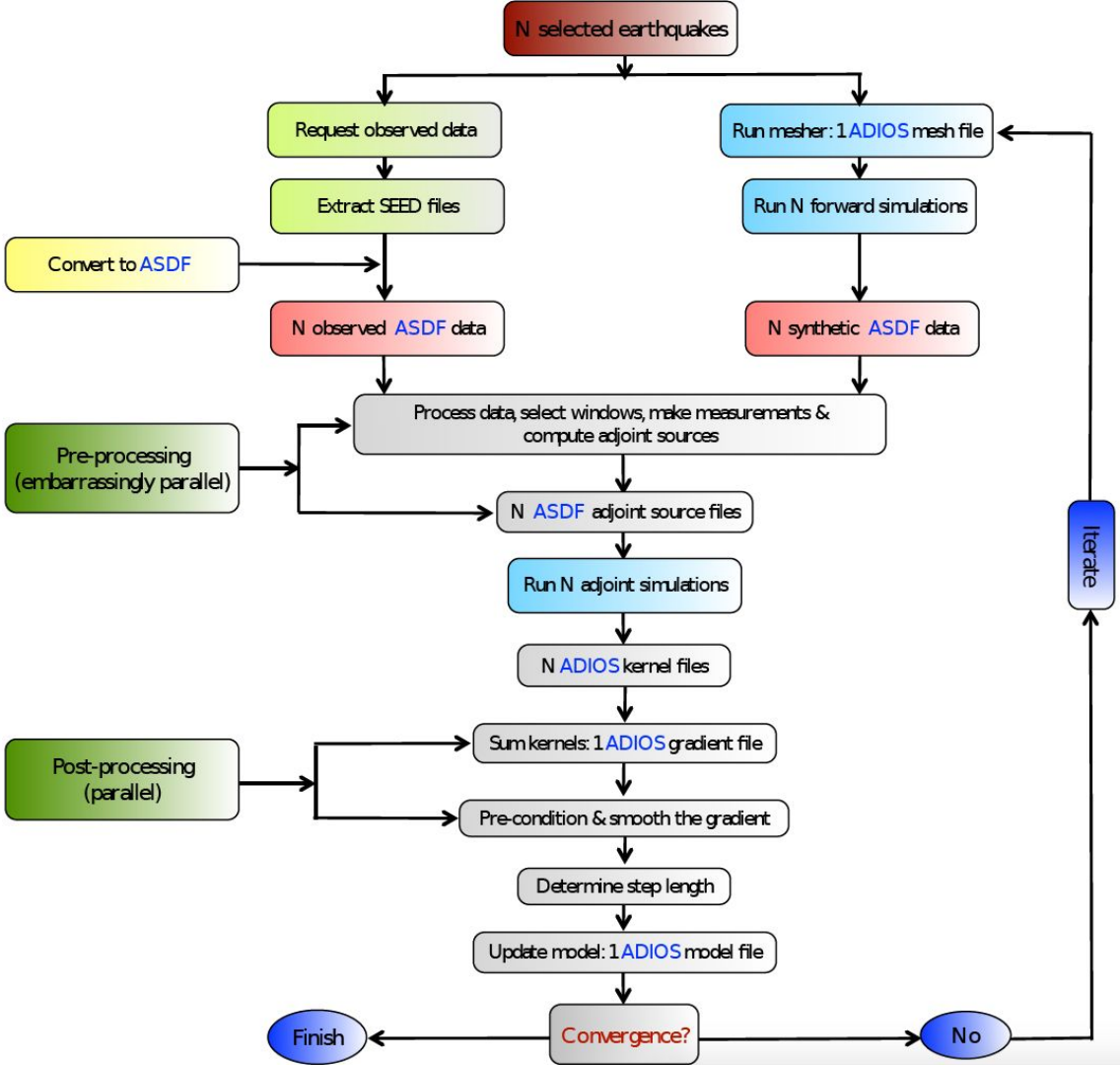
Ensemble Toolkit: Seismic Workflows

Seisflows, Simpy

October 26, 2016


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
Process data, select windows, make measurements & compute adjoint sources
Pre-processing (embarrassingly parallel)
N ASDF adjoint source files
Run N adjoint simulations
N ADIOS kernel files
Sum kernels: 1 ADIOS gradient file
Post-processing (parallel)
Pre-condition & smooth the gradient
Determine step length
Update model: 1 ADIOS model file
Convergence?
Finish
No
Iterate

Two Approaches

- SeisFlows
  - Completely integrated approach
  - Use mostly for research problems*
  - Works reasonably well on stable systems*
  - Computational environment should have little restrictions as for the queuing time
- Simpy
  - Radical-SAGA based approach
  - Sub-workflows for simulation and pre-processing
  - Used to perform the latest global inversion iteration on Titan with a 1000 earthquake

SeisFlows

- End-to-end Seismic imaging, migrations & inversions

- “delivers a complete, customizable waveform inversion workflow”
- “provides a framework for research in regional, global, and exploration seismology”*

http://seisflows.readthedocs.io/en/latest/

SeisFlows: “Modularity & Flexibility”

- “To provide flexibility, SeisFlows is very modular. Users are offered choices in each of the following categories:
  - workflow*
  - system*
  - solver
  - pre-processing
  - post-processing
  - nonlinear optimization”

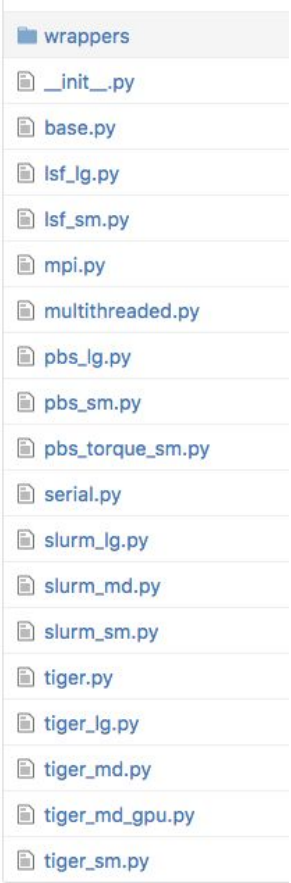
Disagreement:

- Green items: Indeed good to have modularity. Relatively easy to develop and plug different tools
- Red items:
  - Ryan*: Optimization should be an integrated process
  - Matthieu, Wenjie: Separate the science from the engine is better (in Shantenu’s words: separate the “How” from the “What”)

SeisFlows: Issues*

- “System” Modularity:
  - Need to define an interface for each system encountered
  - Akin to RADICAL-SAGA
- Workflow Scheduling
  - Has its own “Pilot’ (~ workflow job)
  - Get a number of nodes through an initial submission
  - The workflow job is submitted and steers the actual jobs
    - (simulation, processing, …)

Retro-fitting SeisFlows in an Ensemble approach will require to break the scheduling logic.



Simpy[1]

- Origin: Running ~2000 reference forward simulation on Titan
  - Submit as many simulations in one batch job (simultaneous + consecutive)
  - Keep track of successes and failures
- Adapted to the Global Tomography workflow later on
  - Managing an iteration with 253 events by hand has proven to be error prone
    - Human time, computer cycles wasted
    - QC?
  - Moving to 1000 events makes it even worse
    - In addition, Titan has been quite unreliable
      - Node failure, slow nodes
    - Needs to accommodate OLCF's policies
      - "Short" walltime
      - Large number of nodes
      - Different system for data processing (Rhea, ~~Eos~~)

1: Fairly unlucky name pick

Simpy

- Events (Earthquakes) are stored in a DB
- Workflows is steered by a “Shaker”
  - Polls the database
  - Feed the job manager
    - Format data, executables
  - The workflow engine is third party
    - Now: SAGA
    - Can be swapped for Pilot
- Build around sub-workflows
  - Now: Solver, pre-processing
  - Should be augmented with post-processing, optimization, ...