Leveraging Madagascar for Reproducible Large-scale Cluster and Cloud Computing

Toby Potter and Jeffrey Shragge
The University of Western Australia

toby.potter@uwa.edu.au

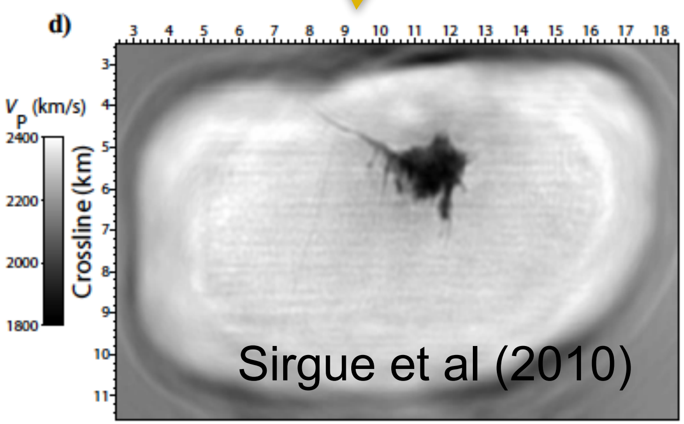
HPC in Exploration Geophysics

THE UNIVERSITY OF WESTERN AUSTRALIA

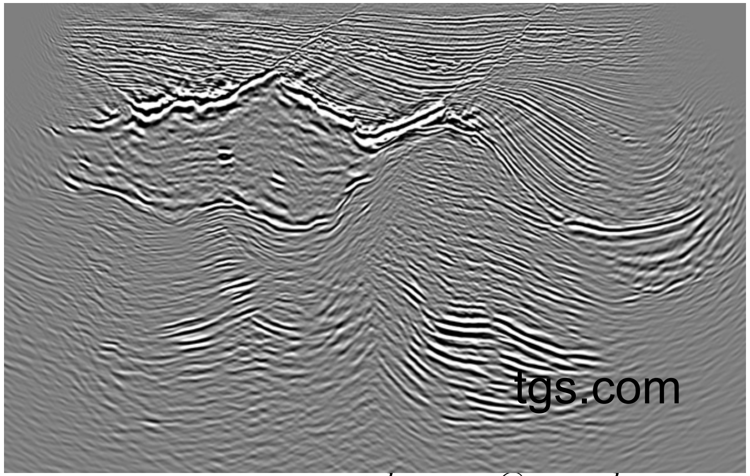
Reverse Time Migration

Full Waveform Inversion

- Up to PB of data
- Millions of core-hours
- Short turnaround



Sirgue et al (2010)



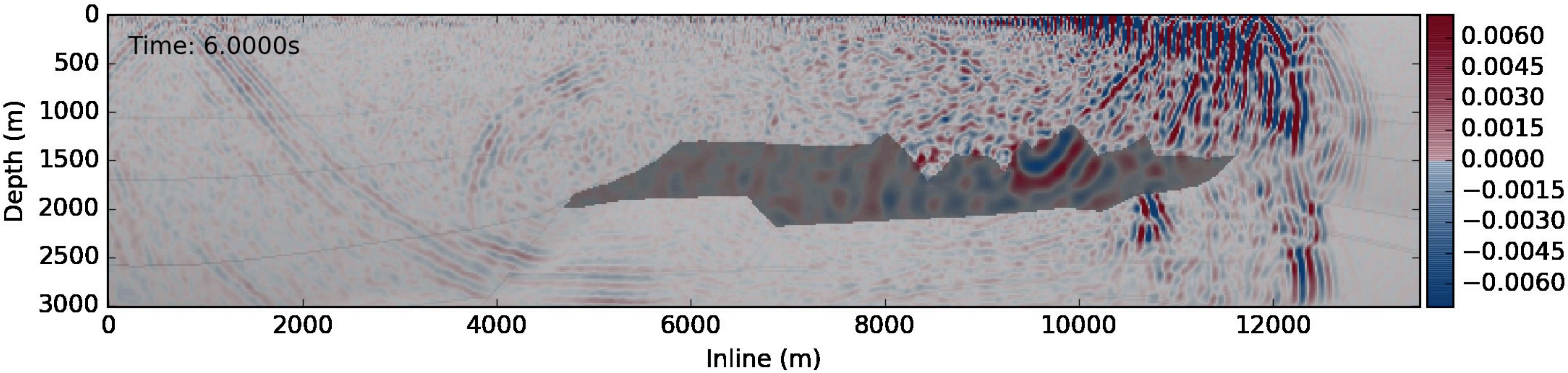
tgs.com

Potter et al., 2015

toby.potter@uwa.edu.au

3D Wavefield propagation

- Finite Difference Time Domain approximations for wavefield modelling
  - Can approximate to high order accuracy
  - Computationally efficient and simple to implement



toby.potter@uwa.edu.au

The future of seismic processing?

toby.potter@uwa.edu.au

Cloud computing options

- Top tier cloud computing providers
- All offer pricing models with dynamic resource allocation
  - Approximately 70% saving compared to on-demand secured resources
- Need 3D wave modelling codes that can adapt to highly variable resource allocations
  - Cloud generally not as fast as bare metal
  - Scalable applications is key to performance
  - Fault tolerance

toby.potter@uwa.edu.au

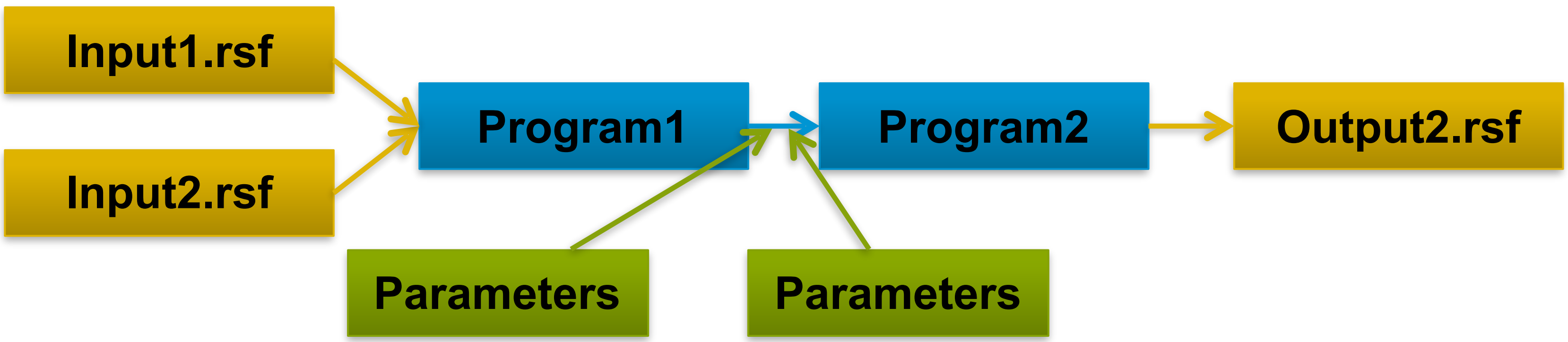
An ideal cloud processing framework

- Rapid development and easy modification
  - Researchers don’t want to spend weeks finding memory errors
- Maps to many job engines
  - Generic launcher
- Resistant to node loss
  - Fast storage and retrieval of state
  - Fault tolerant
- Platform Agnostic
  - Not wedded to any one platform
- Adapts to dynamic resource allocation
- Fast
  - Fortran-like performance

http://www.pc-os.org

What is Madagascar (M8R)?

- Data files used in processing flows with I/O linked by common API
  - Interchangeable Flow commands linked by Unix-style pipes: **|**



- Processing flows written as *SConstruct* scripts
  - Declarative Flow specification, you specify dependencies, not order of execution
  - Python syntax with Madagascar project extensions
- Use software construction (*SCons*) package to run *SConstruct* flows

toby.potter@uwa.edu.au

SConstruct Example – Parallel Looping

```
from rsf.cluster import *    # . . Import Madagascar project rules for your cluster
Cluster(name='my_queue',time=60,ppn=24)
sline = range(0,1000,1)      # . . Set up integer array

# . . Loop over array of 1000 objects with 50 jobs on each of 20 nodes
Fork(time=10,ipn=50,nodes=20)
for iss in sline:
    stag = '04%d' %iss
    Flow('image'+stag,'data'+stag,'my_migration_code par1=… ')
    Iterate()
Join()

# . . Add together object
Flow('image',map(lambda x: 'image-%04d','add ${SOURCES[1:1000]}')
End()                        # . . Additional Madagascar framework commands
```


toby.potter@uwa.edu.au

M8R *scons* Extensions – *mycluster.py**

Object	Description
Flow()	Processing flow command linking input/output files, parameters and programs
Plot()	Generate an intermediate plot files
Result()	Generate a final plot file (i.e. for LaTeX manuscript)
Fetch()	Retrieve data file from remote server (ssh)
Cluster()	**Provide information on cluster resource requirements Queue name, processors per node, walltime (serial)**
Fork()	**Demarcate parallel section; indicate # of nodes, tasks / node, walltime (parallel)**
Iterate()	**Indicate limit of parallel region**
Join()	**End of Fork() section**

***With acknowledgment to Jeff Godwin, Tongning Yang**

toby.potter@uwa.edu.au

Towards the ideal cloud processing framework

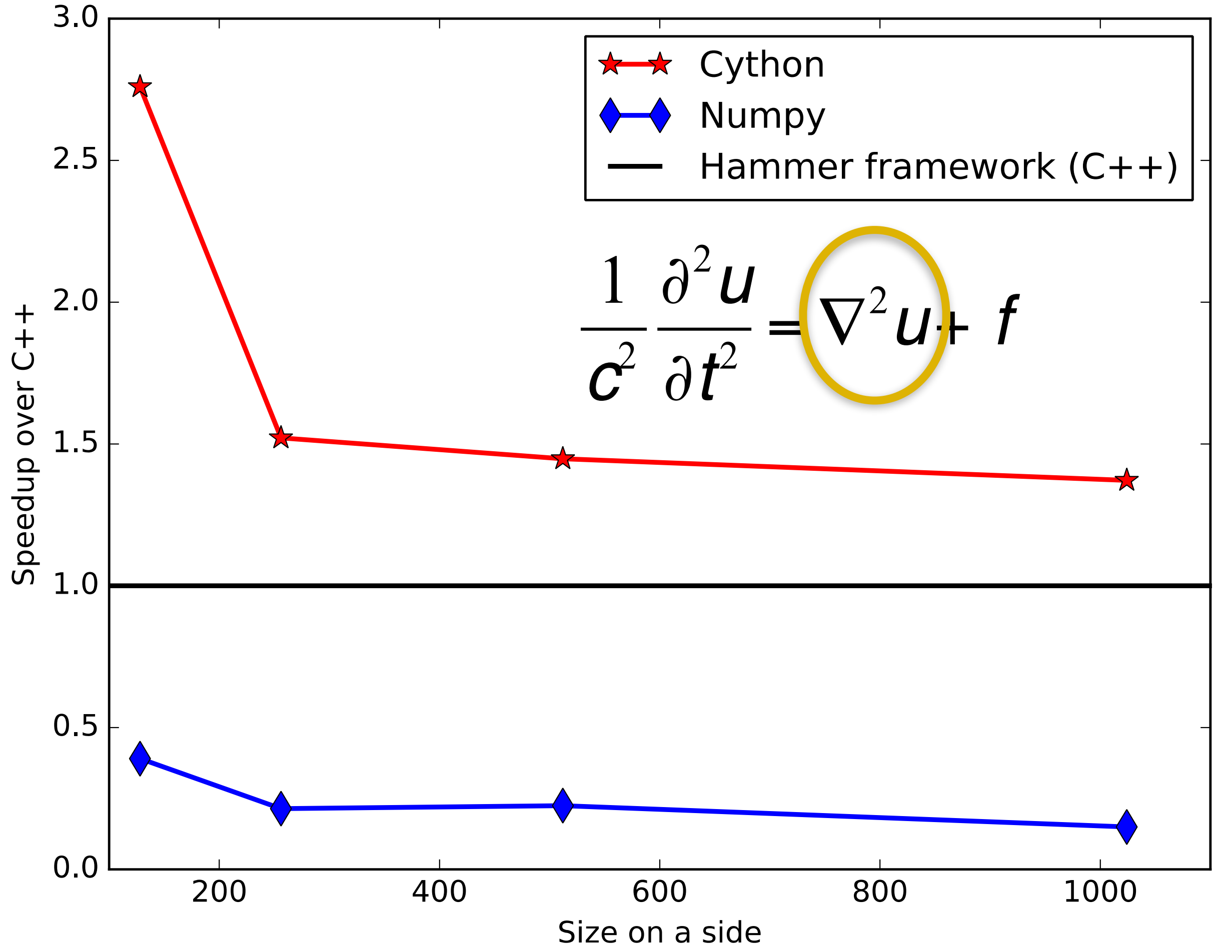
THE UNIVERSITY OF WESTERN AUSTRALIA

- Object-oriented actor model in a high level language
  - Using Python with *ipyparallel* framework
  - Julia?
- Maintaining and modifying state (speed)
  - Python Numpy arrays – 64 bit addressing
  - Fast Cython solvers
- Message passing
  - ZeroMQ routers for fault-tolerant, fast and robust networking
- Data servers for distributed IO
  - Flexible IO backends, e.g HDF5, RSF
  - Enables streaming IO from actors
  - Storage and replay of state from file or memory

toby.potter@uwa.edu.au

Isn't Python slow?

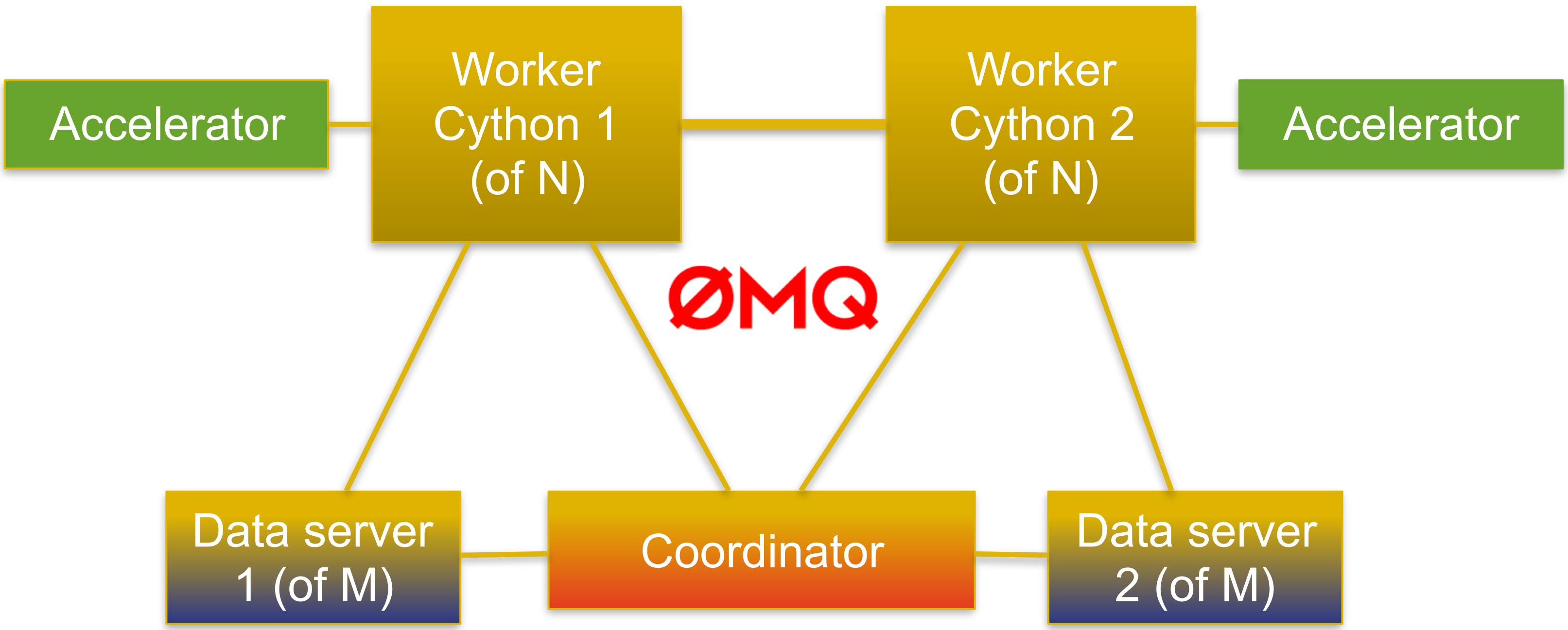
- Numpy arrays: Just contiguous allocations of memory
- Many Numpy operations call optimised C libraries
- Most compute time is spent doing derivatives
- Cython compiles operations to C



toby.potter@uwa.edu.au

Design topology

- 1 Coordinator
- For any collection of machines
  - Any N number of workers
  - Any M number of data servers

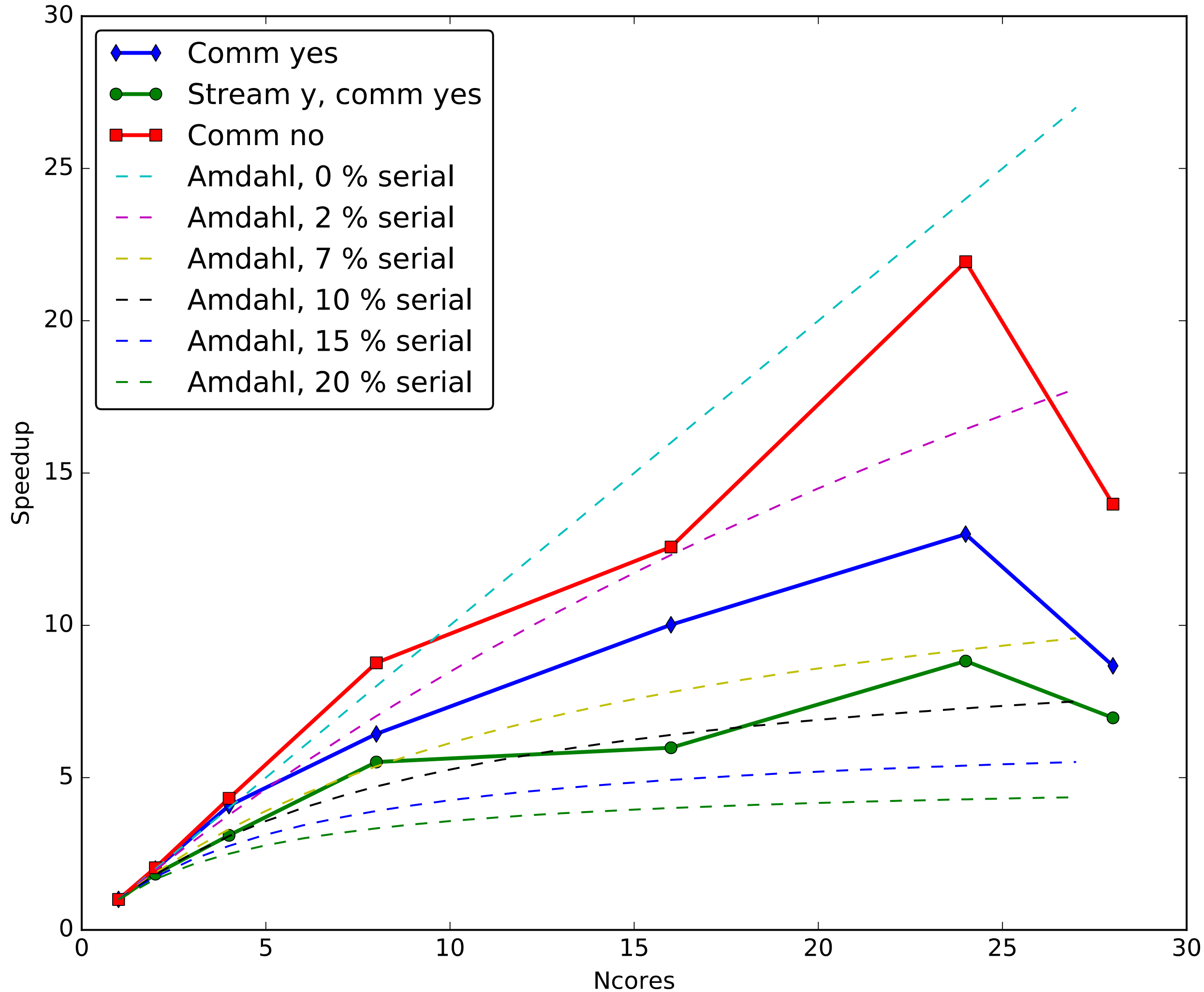


toby.potter@uwa.edu.au

Strong scaling with ZeroMQ in the cloud

- Finite-difference time-domain
- Physics: acoustic wave equation
- Grid size: 512^3
- Platform: Openstack

www.nectar.org.au



Potter et al., 2015

toby.potter@uwa.edu.au

M8R *scons* Extensions – *mycloud.py* *(in progress)*

Object	Description
Flow()	Processing flow command linking input/output files, parameters and programs
Plot()	Generate an intermediate plot files
Result()	Generate a final plot file (i.e. for LaTeX manuscript)
Fetch()	Retrieve data file from remote server (ssh)
Throw()	**Send data to remote server (ssh)**
Cloud()	**Pass information on cloud resource request: disk image, node configuration, queue name, walltime, …**
Fork()	Demarcate parallel section; indicate # of nodes, tasks / node, walltime (parallel)
Iterate()	Indicate limit of parallel region
Join()	End of Fork() section

toby.potter@uwa.edu.au

Concluding Remarks

- Python scripting easily extends M8R to cluster-scale computing
  - Straightforward mark up with little user overhead

- Goal: Extend M8R framework to easily operate on commercial cloud resources with highly variable resource allocations

- Work in extending M8R in the cloud is ongoing:
  - Platform-independent distributed processing engine that leverages ZeroMQ and HDF5.
  - OpenMP-like parallelism observed, with around 5% serial fraction, needs more work
  - Goal: to develop M8R *SCons* wrappers for cloud environments

toby.potter@uwa.edu.au

UWA Geophysics

Geophysics *is the study of the earth, oceans, atmosphere and beyond with the quantitative methods of physics, math and computer science.*