Hello, World!
Design and specifications for SLIM’s software framework

Progress and road map

SLIM group at UBC EOS

February 2008
About

- Our working environment
- Moving towards Python
- From serial to parallel
- Challenges
- Our released software
- SINBAD software release - February 2008
- Future directions
- Summary
Our challenges

- Seismic applications are:
  - extremely large
  - computations intensive
  - IO intensive
- SLIM's technology is driven by multi-d (d>2) non-seperable transforms => need to move out of core
- Academic environment calls for
  - code reuse & reproducibility
  - fast learning curves
Our software

In-core computations:

- Matlab *(Octave)* and Python

Out-off-core seismic tools:

- Seismic Unix, SEP, Delphi, RSF

Scripting tools:

- Unix shells, SLIMpy, SCons

Parallelism

- PETSc *(MPI)*, SLIMpy, pcmd
Moving towards Python

In-core (element-wise operations) for prototyping:

- NumPy
- pyCurveLab

Out-of-core:

- SLIMpy(2) -> high-level algorithm programming (ANAs)
- SCons -> simple scripting and dependency flow
- Madagascar -> only for low-level element-wise operations; not for scripting algorithms
Towards-Python benefits

- Independent of proprietary software licenses
- Reusability
  - Research focused on algorithms not “coding”
- Faster learning curve
  - Easier/faster prototyping and development
  - Easier/faster implementation of new algorithms
From serial to parallel

- Our iterative algorithms require lots of computations
- We need to:
  - accelerate computations
  - optimize utilization of resources in HPC environment
  - distribute memory footprint
  - distribute disposable IO to local filesystems
- To both compute faster and handle bigger problems
From serial ...

- Madagascar takes advantage of SMP systems:
  - through SCons command-line options to support parallel flows
  - recently introduced OMP and MPI utilities on a very trivial level to wrap linear operators by distributing data from a single pipe (over the slow direction), applying an operator to each file, and collecting results onto a single pipe again.
  - SLIMpy have supported parallel flows on SMP systems through option by mechanism similar to SCons

- It turns out that it is nowhere near enough.
... to parallel

We discarded Madagascar approach due to its limitations:

- overhead of scattering/collecting operation for each operator
- limited ability to scatter data in parallel IO fashion
- inflexible scattering topology

We have recently extended SLIMpy with:

- prototype for parallel processing for both CPU and IO
- low-level Madagascar-type utilities that support that prototype
- flexible scattering/collecting operator
- XML metafiles
The parallel-prototype in SLIMpy supports any level of parallelism nearly transparently to user: from serial, through embarrassingly parallel, to complex MPI domain-decomposition by:

- data scattering/gathering operators
- pipe-execution on remote CPUs
- off-loading data from global file systems to CPU’s local /tmp -> effectively providing parallel IO
- parallel extension of RSF files - XML metafiles
- out-off-core equivalent of both SIMD and MIMD computing models
Processing flows

SLIMpy
SIMD/MIMD

Data
Scattering operator
File 1
flow 1
file 1

File 2
flow 2
file 2

...
flow N
file N

Gathering operator
Result

Madagascar
SIMD

Data
scattering operation
File 1
operator
file 1

File 2
operator
file 2

...
gathering operation
...

File N
operator
file N

Result
Challenges

Computational bottlenecks to be optimized:

- long pipes, processing many files at the same time, introduce competitive IO on local file-system and hurt cumulative IO performance
- long pipes can exceed memory resources
- sophisticated algorithm can be IO costly; some suffer from in-core to out-off-core transition and have to be optimized
Challenges

Madagascar limitations to overcome:

- lack of support for distributed file-components
- problem as simple as:
  - limited integer representation of dimensions in RSF files; redundancy of either CurveLab or SurfBox transform can easily exceed the 4-byte integer limit even for relatively small problems
- numerical precision; i.e., 4-byte float representation in Madagascar
Our released software

- To public domain
- Madagascar -> contribution in user/slim
- pyCurveLab
- SINBAD software release ...
Software release – Feb 2008

Highlights:
https://wave.eos.ubc.ca/Software/SINBAD/highlights/

Packages:
https://wave.eos.ubc.ca/Software/SINBAD/2008-02/

Support: a number of mailing lists (see below)
Highlights of SLIM software release to SINBAD sponsors

UBC EOS SLIM (http://slim.eos.ubc.ca)

Date: February 20th, 2008

Summary:
This Web page contains a short description of the software released to SINBAD sponsors.
The list of prerequisites and the tar-balls of the most recently released software packages is located at https://wave.eos.ubc.ca/Software/SINBAD/2008-02/.

1 Wavefield reconstruction
   - 1.1 Curvelet-based focal transform
   - 1.2 Jittered undersampling
   - 1.3 Surfacelet transform
   - 1.4 Seismic denoising with 3-D transform-domain sparsity (NEW)

2 Wavefield separation
   - 2.1 Curvelet-based primary-multiple separation from a Bayesian perspective (NEW)
   - 2.2 3D Curvelet-based primary-multiple separation from a Bayesian perspective (NEW)

3 Imaging
   - 3.1 Curvelet Match Filtering for True Migration Amplitude Recovery (NEW)

4 Solvers
   - 4.1 Iterative solvers
     - 4.1.1 Iterative Soft Thresholding with cooling (ISTc)
     - 4.1.2 SPGL1: A solver for large-scale sparse reconstruction
   - 4.2 LSQR

5 Transforms
   - 5.1 Surfacelet transform with RSF data interface
   - 5.2 Curvelets
     - 5.2.1 2D and 3D Curvelet Transform with RSF data interface
     - 5.2.2 3D MPI Curvelet Transform with RSF data interface

6 Utilities
   - 6.1 Jitter sampling

7 SLIMpy development/programming environment

Bibliography

Install Admin 2008-02-19
Software release for SINBAD members

Last updated: Feb 20, 2008

To submit bug reports, questions, etc., subscribe first to one of the mailing lists:

- SLIM2RSF-user for SLIM2RSF
- SLIM-dev-users for all SLIMpp extensions
- PyCurveLab for PyCurveLab
- SLIMutil for everything else and help with installation of external software.

After you receive the subscription confirmation, send an e-mail to the list.

External software

External software necessary for full installation, and not available from SLIM site:
Requirements specific for each SLIM package are listed with the package (see below).

A. SVN-1.2.1 or newer from http://svnversion.sourceforge.net
   Note: proxy configuration might be needed behind the firewalls)
B. XCom-0.97 or newer from www.xcom.org
C. GCC-4.2 from gcc.gnu.org
D. SWIG-1.3.20 or newer from www.swig.org
E. Python-2.5 from www.python.org
F. NumPy-1.0.4 from www.numpy.org
G. SciPy-0.6.0 from www.scipy.org
H. Maniploth-0.901 or newer from maniploth.sourceforge.net
I. Boost-1.33 or newer from www.boost.org
J. MATLAB-7.1 or newer from www.mathworks.com
K. fftw-2.1.5 from www.fftw.org
L. fftw-3.1.2 from www.fftw.org
M. CurveLab-2.1.1 from www.curvelet.org
   Note: only for license for CurveLab-2.1.1-SLIM
N. MPI compilers based on GCC-4.2
   Note: either vendor implementation or MPICH from www-unix.mcs.anl.gov/mpi
O. Petsc-2.3.2 from http://www-unix.mcs.anl.gov/petsc/petsc
   Note: make sure to compile with the above MPI compilers
P. LaTeX-2.0 or newer (comes with most UNIX-like systems)
Q. SEGTest-0.8.5 from rc.sourceforge.net/Sextest
   Note: use version 188 when downloading directly from Subversion repository
R. MADAGASCAR-3.204 from SVN developer tree at rc.sourceforge.net

Note: compile with API=c++,j90,python,matlab
(Installation instructions can be found at rc.sourceforge.net)

The above software might be installed in the order as listed.

SLIM software packages

The packages are provided here in the compressed tar format.

The installation procedure for each package is explained in the README file included in the package, and also accessible under corresponding "Package README" from this page.

Please, install the packages in the order they are listed on this page and read corresponding "Package README" before unpacking the tarballs.

A. CurveLab-2.1.1-SLIM
   - Info: SLIM extensions to CurveLab-2.1.1
   - Prerequisites:
     1. License for CurveLab-2.1.1
     2. fftw-3.1.2 from www.fftw.org
     3. For MPI part:
        1. fftw-2.1.5
        2. Petsc-2.3.2 or newer
   - Tar-ball: CurveLab-2.1.1-SLIM.tar.gz
   - Package README: CurveLab-2.1.1-SLIM.README.txt
   - Notes: fftw-3.1.2 has to be configured using --with-pic

B. pyCurveLab
   - Info: SLIM python wrapper to CurveLab-2.1.1
   - Prerequisites:
     1. CurveLab-2.1.1-SLIM
     2. Boost-1.33 or newer
     3. NumPy-1.0.4 from http://www.numpy.org/
     4. SciPy-0.6.0 from www.scipy.org
   - Tar-ball: pyCurveLab.tar.gz
   - SLIM extension: pyCurveLab.*
   - Package README: pyCurveLab.README.txt

C. SurfBox.SLIM
   - Info: SLIM version of SurfBox based on code from Yue M. Lyu and MathWorks
   - Prerequisites:
     1. SurfBox-2.1.1-SLIM
     2. Boost-1.33 or newer
     3. NumPy-1.0.4 from http://www.numpy.org/
   - Tar-ball: SurfBox.SLIM.tar.gz
   - SLIM extension: SurfBox.SLIM.*
   - Package README: SurfBox.SLIM.README.txt
   - Notes: None

D. SLIM2RSFext
   - Info: SLIM extensions to RSF
   - Prerequisites:
     1. MADAGASCAR (SVN revision 3204) from either
Software release

I. SLIMpy.apps

- Info: SLIMpy - Applications using SLIMpy's ANAs

<table>
<thead>
<tr>
<th>Prerequisites:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SLIMpy ANAs</td>
<td>None</td>
</tr>
<tr>
<td>2. SLIM2RSFext</td>
<td>None</td>
</tr>
</tbody>
</table>

J. SLIMpy.user_demos

- Info: SLIMpy - Users' demos using SLIMpy's Applications

<table>
<thead>
<tr>
<th>Prerequisites:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SLIMpy ANAs</td>
<td>None</td>
</tr>
<tr>
<td>2. SLIM2RSFext</td>
<td>None</td>
</tr>
</tbody>
</table>

K. SoftReleaseSept07 (updated for February 2008 release)

- Info: Additional collection of stand-alone demos

<table>
<thead>
<tr>
<th>Prerequisites:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SLIMpy ANAs</td>
<td>None</td>
</tr>
<tr>
<td>2. SLIM2RSFext</td>
<td>None</td>
</tr>
</tbody>
</table>

L. SoftReleaseFeb08

- Info: Additional collection of stand-alone demos

<table>
<thead>
<tr>
<th>Prerequisites:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SLIMpy ANAs</td>
<td>None</td>
</tr>
<tr>
<td>2. SLIM2RSFext</td>
<td>None</td>
</tr>
</tbody>
</table>

Prerequisites:
1. MADAGASCAR (SVN revision 3204) from either
   - directly from SVN repository at svn:sourceforge.net using svn's "r 3204" option
   - SLIM's local tarball MADAGASCAR.v3204.tar.gz
2. CurvLab-2.1.1-SLIM
3. Starbox.slim
   - Tar-ball: SLIM2RSFext.tar.gz
   - SLIM revision: SLIM2RSFext.txt
   - Package README: SLIM2RSFextREADME.txt
   - Notes:
     - unpacking SLIM2RSFext.tar.gz will create slimintern directory (not SLIM2RSFext)
     - one can compile RSF together with SLIM2RSFext

E. SLIM2RSFext-MPI

- Info: SLIM MPI extensions to RSF

<table>
<thead>
<tr>
<th>Prerequisites:</th>
<th>Notes:</th>
</tr>
</thead>
</table>
| 1. MADAGASCAR (SVN revision 3204) from either
   - directly from SVN repository at svn:sourceforge.net using svn's "r 3204" option
   - SLIM's local tarball MADAGASCAR.v3204.tar.gz
2. CurvLab-2.1.1-SLIM
3. Starbox.slim
   - Tar-ball: SLIM2RSFext-MPI.tar.gz
   - SLIM revision: SLIM2RSFext-MPI.txt
   - Package README: SLIM2RSFext-MPI README.txt
   - Notes:
     - unpacking SLIM2RSFext.tar.gz will create slimintern-mpi directory (not SLIM2RSFext-MPI)
     - one can compile RSF together with SLIM2RSFext

F. SLIMpy.core

- Info: SLIMpy - a Python interface to ANAs: core packages

<table>
<thead>
<tr>
<th>Prerequisites:</th>
<th>Notes:</th>
</tr>
</thead>
</table>
| 1. MADAGASCAR (SVN revision 3204) from either
   - directly from SVN repository at svn:sourceforge.net using svn's "r 3204" option
   - SLIM's local tarball MADAGASCAR.v3204.tar.gz
2. CurvLab-2.1.1-SLIM
3. Starbox.slim
   - Tar-ball: SLIM2RSFext.tar.gz
   - SLIM revision: SLIM2RSFext.txt
   - Package README: SLIM2RSFext README.txt
   - Notes:
     - unpacking SLIM2RSFext.tar.gz will create slimintern directory (not SLIM2RSFext-MPI)
     - one can compile RSF together with SLIM2RSFext

G. SLIMpy.ANAs

- Info: SLIMpy - Abstract Numerical Algorithms (ANAs)

<table>
<thead>
<tr>
<th>Prerequisites:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SLIMpy core</td>
<td>None</td>
</tr>
<tr>
<td>Tar-ball: SLIMpy.ANAs.tar.gz</td>
<td>None</td>
</tr>
<tr>
<td>SLIM revision: SLIMpy.ANAs.txt</td>
<td>None</td>
</tr>
<tr>
<td>Package README: SLIMpy.ANAs README.txt</td>
<td>None</td>
</tr>
</tbody>
</table>

H. SLIMpy.apps

- Info: SLIMpy - Applications using SLIMpy's ANAs

<table>
<thead>
<tr>
<th>Prerequisites:</th>
<th>Notes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SLIMpy ANAs</td>
<td>None</td>
</tr>
<tr>
<td>2. SLIM2RSFext</td>
<td>None</td>
</tr>
</tbody>
</table>
Software release

- MADAGASCAR.v3204 - SLIM contr. in user/slim (public)
- CurveLab-2.1.1-SLIM - our extensions to CurveLab
- pyCurveLab - Python interface to CurveLab (public)
- SurfBox.slim - our extensions to SurfBox
- SLIM2RSFext - RSF interface to CurveLab and SurfBox
- SLIM2RSFext-MPI - support for MPI
  - 3D-FDCT
  - MPI domain decomposition
  - parallel IO through XML metafiles
Software release

- SLIMpy packages
  - SLIMpy.core  - core utilities
  - SLIMpy.ANAs  - abstract numerical algorithms
  - SLIMpy.apps  - applications
  - SLIMpy.demos  - developer demos (probably to retire)
  - SLIMpy.user_demos  - user demos (future mainstream)
- Demo collections:
  - SoftReleaseSept07  - MATLAB / Python / reproducible research
  - SoftReleaseFeb08  - Python / reproducible research
Software support

Mailing lists:

- SLIM2RSF-user:
  - [http://slim.eos.ubc.ca/mailman/listinfo/slim2rsf-user](http://slim.eos.ubc.ca/mailman/listinfo/slim2rsf-user)
- SLIMpy-user:
  - [http://slim.eos.ubc.ca/mailman/listinfo/slimpy-user](http://slim.eos.ubc.ca/mailman/listinfo/slimpy-user)
- pyCurveLab:
  - [http://slim.eos.ubc.ca/mailman/listinfo/pycurvelab](http://slim.eos.ubc.ca/mailman/listinfo/pycurvelab)
- SLIMsoft:
  - [http://slim.eos.ubc.ca/mailman/listinfo/slimsoft](http://slim.eos.ubc.ca/mailman/listinfo/slimsoft)
Future directions

- new operators through
  - command-line based seismic tools
  - SLIMpy operators
- new ANA algorithms in SLIMpy
- contributing to Madagascar
  - XML metafiles for parallel IO
  - named pipes to support pipes in distributed parallel IO
- releasing SLIMpy core to community (support other seismic packages)
- programing moving near-entirely to SLIMpy (except for fast prototyping)
Summary

- We have built a large software framework for seismic-imaging applications.
- We have implemented new algorithms in this framework.
- We have build a number of low-level tools to support those algorithms.
- ... and the work goes on ...

We make a huge effort in streamlining/accelerating algorithm development and allowing researchers to focus on research rather than software development, and that efforts starts to pay off.
Acknowledgments

This presentation was carried out as part of the SINBAD project with financial support, secured through ITF, from the following organizations: BG, BP, Chevron, ExxonMobil, and Shell. SINBAD is part of the collaborative research & development (CRD) grant number 334810-05 funded by the Natural Science and Engineering Research Council (NSERC).
Good bye, World!