THE NEW SEISPP MODULES IN ANTELOPE CONTRIB: A SCALABLE, HIGH PERFORMANCE, GENERIC OBJECT-ORIENTED PROCESSING FRAMEWORK WITH AN API FOR MORTALS

Prof. Gary L. Pavlis
Department of Earth and Atmospheric Sciences
Indiana University

Antelope User's Group Meeting: Victoria, BC, Aug. 30, 2018

BACKGROUND

- Started writing C++ applications with Antelope libraries in early 2000s
- Datascope database
  - Long term fan BUT
  - Learned the lesson that a relational database is not always the right tool (when your only tool is a hammer everything looks like a nail)
- Now a brief monologue on what I think I've learned

WHERE A RELATIONAL DB IS THE RIGHT CHOICE

- Transactional algorithms
  - dbloc2
  - dbpick
  - dbxcor
- Raw waveform management – especially if large
- When data is naturally relational (i.e. related tables)
  - Bulletin and catalog data
  - What we manage in dbmaster tables.

PROBLEMS THAT DROVE THIS DEVELOPMENT

- High Performance Computing Issues driven by work on plane wave migration code (PWMIG – Pavlis 2011, *Comp & Geosci*)
  - dB programs always IO bound
  - Antelope support issues on HPC clusters
  - HPC file system issues
- Schema issues
  - CSS3.0 designed for raw data, not intermediate processing results
  - Proliferation of nonstandard (often one up) tables
  - Clashing concepts for processed data (e.g. pseudostation stacks)
- Handling intermediate results in complex workflows
- BIGGEST: Every new problem requires a new program

Influences on SEISPP UNIX filter programs

- Old UNIX guy: 40+ yrs of writing shells scripts
- Experience in seismic reflection processing in general and seismic unix in particular
- Development of libseispp library over the past decade
- Lessons learned in 16 yrs of C++ coding
- Witnessing failures in community using bleeding edge (research) technology for an engineering problem

CAVEATS

- This is a prototype
  - Skewed by projects I have ongoing that needed new software tools
  - No attempt to optimize performance
  - Aim is prototype framework for research computing NOT production
  - Some things are good, some need to be redesigned
- Lots of obvious holes – especially cleaner metadata management approach
- Incomplete documentation

SO WHAT IS THIS STUFF?

- First, a set of unix filters
- Let's look a couple of examples

SIMPLE EXAMPLE: FILTER AND DISPLAY AN "ENSEMBLE"

```
#!/bin/csh
set ffid=$1
filter3c "BW 500 5 1500 5" < 4850/${ffid}.dat | display_ensemble
```

A MORE COMPLICATED EXAMPLE

```
#!/bin/csh
#PBS -l nodes=1:ppn
#PBS -l walltime=130:00:00
cd /gpfs/projects/GEOL/GeophysicsLab/Homestake/particle_motion_analysis
set elfile=localevids.list
set outdir=mwpm_output_files
set indir=ensembles
set sta=BEAM
foreach evid (`cat $elfile`)
  foreach sub (deep shallow surface)
     set infile=${indir}/tag_evid_${evid}_subarray_${sub}
     echo $infile
     set outfile=${outdir}/mwpm_${evid}_${sub}.dat
     tcecut -40 120 < $infile | dismember \
        | subset_streamfile sta:string -eq $sta -objt ThreeComponentSeismogram \
        | add_arrivals -t ThreeComponentSeismogram | mwpm > $outfile
  end
end
```

Note: 130 serial processing hours requested – not a lightweight calculation. Driver for parallel development discussed later

A LIVE EXAMPLE

- If the network works we'll see a screen on a cluster at Indiana
- I'll run a variant of the display script with much longer chain of processing modules

WHAT IT IS #2

- Stream Processing Model
- Working approach used in all reflection processing systems since the 1960s

Relevant History: Early Computers

- Huge machines serviced by high priests
- Earliest machines stored most data on magnetic tape

http://www.columbia.edu/acis/history/tapes.html

THE MAGNETIC TAPE LEGACY:

Assertion: all conventional seismic reflection data processing systems have concepts inherited from mag tape data

http://www.columbia.edu/acis/history/tapes.html

The Magnetic Tape Legacy:

- Linear data model
- Streams processing
  - Linear data files
  - Header-based metadata
  - Railroad car processing model

THE LINEAR DATA MODEL:

SEG Y rev 1 May 2002

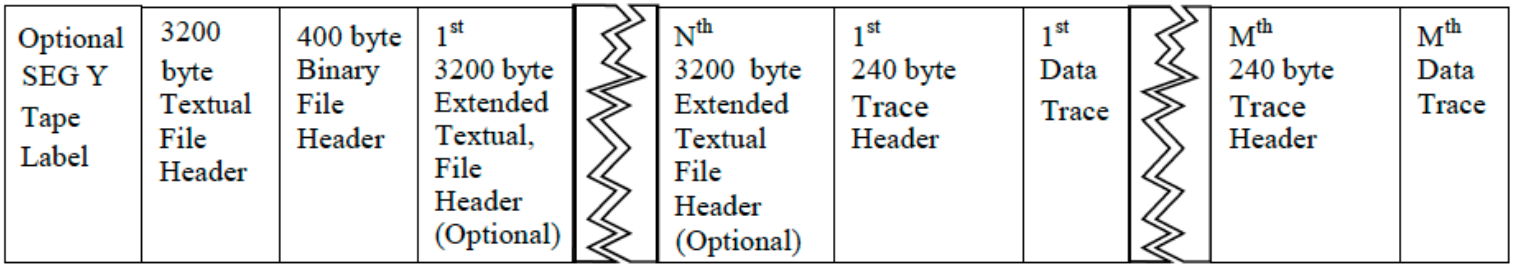
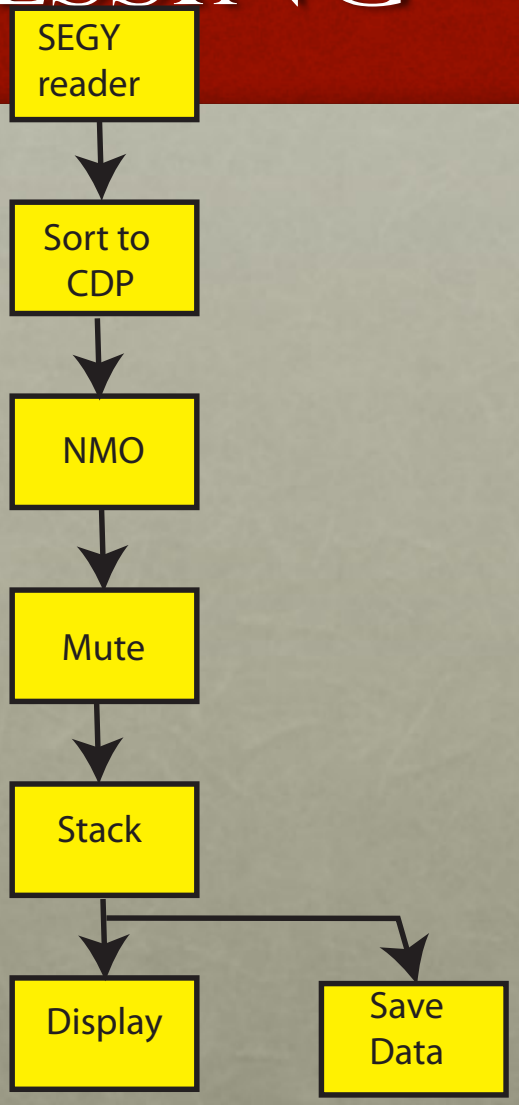


Figure 1 Byte stream structure of a SEGY file with N Extended Textual File Header records and M traces records

2002 SEGY standard (http://www.seg.org)

STREAMS PROCESSING

- Typical flow shown
- Boxes are processes
- Arrows are data flow
- Traditional
  - custom command interpreter (e.g. Disco)
  - SU – unix shell
- Modern=GUI



Generic Concepts in Reflection Style Processing

- The data is not a random collection of stuff but a complete data set with a limited set of organizations (gathers)
- Data flows though a “system” from raw form to final result
- The “system” manages this data flow

RAILROAD SWITCHYARD ANALOG

- Railroad cars are like one seismogram
- Input train is data file
- Side cars are processing modules

Note: I first learned this from David Okaya

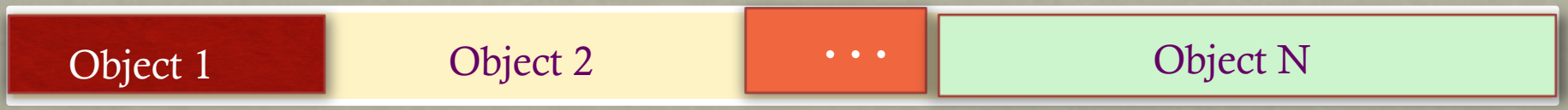
http://images.google.com

WHAT IT IS #3: GENERIC OBJECT ORIENTED STREAM PROCESSING

- Data Object types currently supported
- Railroad car analogs
  - TimeSeries
  - ThreeComponentSeismogram
  - PMTimeSeries
- Switchyard analogs (generic gather)
  - TimeSeriesEnsemble
  - ThreeComponentEnsemble
  - Coming: template class <T> ensemble
- StreamObjectFile Concept

STREAMOBJECTFILE CONCEPT

The Familiar Linear File Model:



Note:

- Normally read from 1 to N
- Conceptually like classic data formats like SEGY
- File slots are not constant size like SEGY
- The format is not the data – linear (stream) file is the abstract concept

A VERY important idea few in our community understand

DATA AND METADATA ABSTRACTION

- Data
  - The core components that define a thing
  - Usually requires multiple required data objects
  - Example: ThreeComponentSeismogram
- Metadata
  - Auxiliary parameters that define something about the data
  - Traditionally stored as “header” data
  - All supported data objects inherit the Metadata object
  - Generic,open ended header

SUPPORTED DATA TYPES

- Seismograms
  - TimeSeries
  - ThreeComponentSeismogram
  - ComplexTimeSeries
- Ensembles (gathers)
  - TimeSeriesEnsemble
  - ThreeComponentEnsemble
- Novel abstraction
  - PMTimeSeries

METADATA INTERFACE

- get specified type
  - get_string(string key)
  - get_double(string key)
  - get_int(string key)
- Put specified type – depend on C++ overloading
  - put(string key, double val)
  - put(string key, string val)
  - put(string key, int val)
- Templates
  - get<T>(string key)

Quick overview of available modules

- Incomplete html documentation
- Will do quick overview in three categories
  - Data import
  - Data export
  - Waveform processing
  - Graphics
  - Metadata manipulators
  - Particle motion analysis

PROGRAM OVERVIEW: DATA IMPORT

- db2seispp – Datascope import
- SU3CEnsembleConverter – Converts seismic unix data organized as triplets to 3C objects
- dbactive_reader – build shot files from continuous data (comparable to db2segy)
- dbxcor_import - used to build extended beams with time shifts defined by dbxcor

PROGRAM OVERVIEW: DATA EXPORT

- export_to_matlab – exports TimeSeriesEnsemble to matrix that can be read by matlab import function
- export_to_matlab_3C – exports ThreeComponentEnsemble to matrices that can be read by matlab import function
- export_to_su – export to seismic unix (TimeSeries data only)

PROGRAM OVERVIEW: WAVEFORM PROCESSING

- agc – 3C automated gain control
- apply_statics – apply simple static time shifts
- filter3c – 3C data filter (uses BRTT filter library)
- linearmoveout – applies plane wave (linear) time shifts
- peak_scaling –scale data by peak 3C value (useful for plotting)
- tcecut/window_streamfile – time windowing (tcecut for 3C ensembles only, window_streamfile more generic
- sphdiv – power law geometric spreading divergence correction
- topmute – mute data relative to zero relative time
- zeropad – extend the front of a waveform segment with zeros

PROGRAMS OVERVIEW: GRAPHICS

- display_ensemble – plot data in a ThreeComponentEnsemble object
- Alternatives
  - Scalar data can be passed through export_to_su and plotted with suxwigb or suximage
  - Use export_to_matlab programs and plot with matlab

PROGRAMS OVERVIEW: METADATA MANIPULATORS 1

- add_arrivals – adds arrival times from travel time tables and source/receiver data
- alias_metadata – rename one or more attributes
- BasicTimeSeriesAttributes/listhdr – Used to dump header (Metadata) contents.
- clrhdrattr – housecleaning program to clear one or more attributes in dataset
- csv_join – build an extended table to use as input to set_metadata that acts like dbjoin

PROGRAMS OVERVIEW: METADATA MANIPULATORS 2

- dbload_hdr – load attributes from antelope db
- dbrevise_source_data – update source data in a dataset (used to update headers after a relocation)
- rename_attributes – change name tags for one or more attributes stored with data (optionally clear old)
- set_metadata – set one or more attributes driven by a pf (fairly flexible but tables have to be built externally)
- set_offset – compute and set a distance attribute (mostly for active source data)

PROGRAMS OVERVIEW: UTILITIES

- ator/rota – convert between absolute and relative time
- build_index – build index for random access (required for sort and parallel reader (discussed later))
- cat_seispp – concatenate multiple data files
- dismember/gather – take an ensemble apart (dismember) or build ensembles from seismograms (gather)
- fragment – take a file apart and write into a directory with one object per file
- seispp_b2t/seispp_t2b – switch between binary and text format
- sort1 – sort data by one key value (warning – primitive pure memory sort intended for simple sort within a pipeline)
- subset_seispp – subset data (can be done inline)

PROGRAMS OVERVIEW: PARTICLE MOTION ANALYSIS

- mwpm – multiwavlet particle motion estimator (Lorie Bear papers in 1990s)
- mask_pm_snr – used to automatically discard pm results for preevent noise
- mwpmavg – array average of pm estimates
- pm2wulff – converts pm estimates to spherical coordinate values for plotting with Wulff net projection in matlab
- pmto3c – extract major axes as 3C seismograms

Note: Requires an auxiliary package available on github called ParticleMotionTools

ADDING A NEW PROGRAM

- Procedure:
  - cd to $ANTELOPE/src/contrib/bin/seispp
  - Run script: new_seispp_module progname
  - cd to progname, edit, debug, man page, install
- Available templates
  - template_instruction.cc – simple with embedded comments to process ThreeComponentEnsemble objects
  - template_plain.cc – instruction with comments removed
  - template_multiobject.cc – contains code example for program supporting multiple data types

THE "API FOR MEER MORTALS" ISSUE IN MY TITLE

- Two examples that are impenetrable by meer mortals
  - Vtk
  - Qt
- Reason: they are large generic packages
- Need: higher level abstraction for domain specific, single concept
- Next slide is an example (not true of everything in my libraries)

StreamObjectFile API

- Writer:
- Reader:

CLUSTER EXTENSIONS

- IndexedObjectReader – random access data reader
- DataSetReader – reads a virtual data set assembled from multiple indexed files defined by pf
- PipelineProcessor – writer that sends output through a fixed unix pipeline chain
- ParallelReader – under development. Aimed as reader for PipelineProcessor chains or one up compute intensive algorithms

DISCUSSION?