ParselSwift

HILADO, WP4: *Bringing it to the user*

Mark Kettenis, JIVE

Swift

- A (functional) parallel scripting language
- Based on Java CoG Kit
- http://www.swift-lang.org
- Not the Apple language!

Swift & AIPS data

- Created an AIPS "mapper"
  Single Java class + two-line diff

- Usage:

```
aipsdata uvdata <aips_mapper;name="N03L1",
class="UVDATA", seq=1>;
aipstable sntable <aips_mapper;name="SN",
version=1>;
```

Swift & ParselTongue

- Execute AIPS tasks from Swift
- Implemented as Swift "app" procedures that call little ParselTongue scripts

```
import "aips";
uvdata = fitld(datain=n03l1, digicor=-1.0);
```

- Automatically generated

VLBI Pipeline

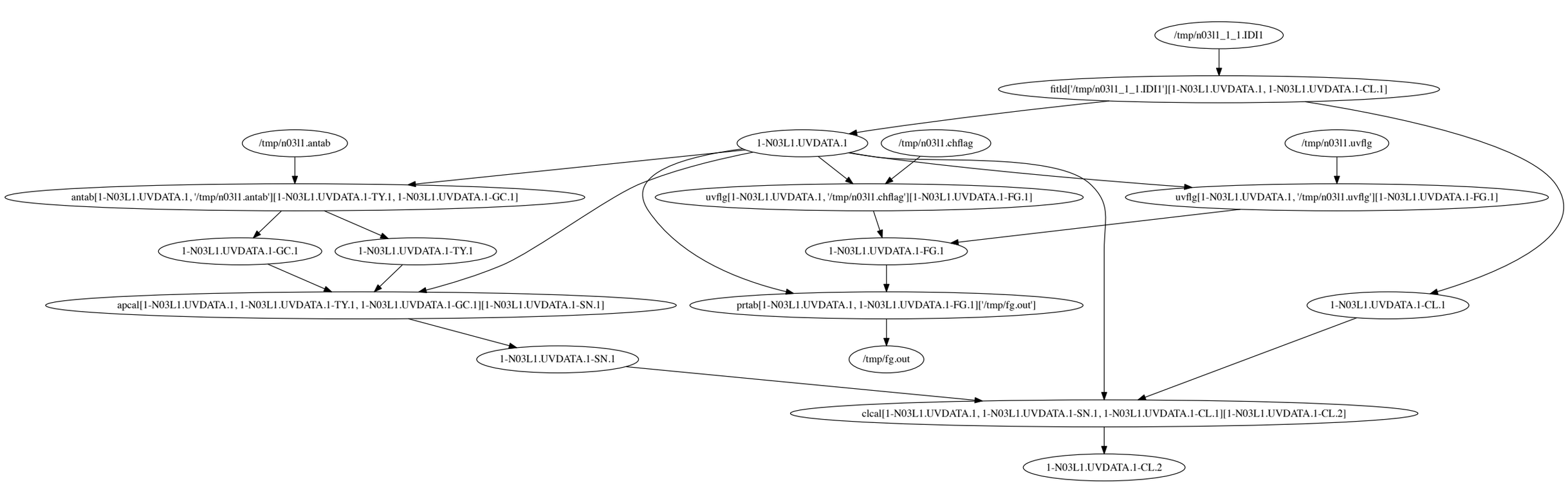
```
import "aips";

aipsfile n03l1 <"/tmp/n03l1_1_1.IDI1">;
aipsfile n03l1_uvflg <"/tmp/n03l1.uvflg">;
aipsfile n03l1_fgout <"/tmp/n03l1.fgout">;
aipsfile n03l1_antab <"/tmp/n03l1.antab">;

aipsdata uvdata <aips_mapper;name="N03L1", class="UVDATA", seq=1>;

aipstable fgtable <aips_mapper;name="FG", version=1>;
aipstable tytable <aips_mapper;name="TY", version=1>;
aipstable gctable <aips_mapper;name="GC", version=1>;
aipstable sntable <aips_mapper;name="SN", version=1>;
aipstable basecal <aips_mapper;name="CL", version=1>;
aipstable ampcal <aips_mapper;name="CL", version=2>;
aipstable fringsol <aips_mapper;name="SN", version=2>;

aipsdata tgtsplit <aips_mapper;name="3C84", class="SPLIT", seq=1>;
aipsdata tgtmulti <aips_mapper;name="3C84", class="MULTI", seq=1>;
aipsdata tgtdata <aips_mapper;name="3C84", class="UVDATA", seq=1>;

aipsdata tgtimage <aips_mapper;name="3C84", class="ICL001", seq=1>;

uvdata = fitld(datain=n03l1, digicor=-1.0);
basecal = cltable(uvdata);

fgtable = uvflg(indata=uvdata, intext=n03l1_uvflg);

n03l1_fgout = prtab(indata=uvdata, intable=fgtable);

(tytable, gctable) = antab(indata=uvdata, calin=n03l1_antab, offset=0.3);

(sntable) = apcal(indata=uvdata, freqid=1.0, tyver=tytable, gcver=gctable, prtlev=1.0);

float ant = 2.0;
float solint = 3.0;
string sources[] = ["3C84", "DA193"];
string selfcal_sources[] = ["3C84"];
float fring_snr = 11.0;

(ampcal) = clcal(indata=uvdata, subarray=1.0, opcode="CALI", gainver=basecal, snver=sntable, refant=ant, interpol="SELF", doblank=1.0, dobtween=1.0,
samptype="BOX");

(fringsol) = fring(indata=uvdata, freqid=1.0, gainuse=ampcal, flagver=fgtable, docalib=2.0, subarray=1.0, refant=ant, search=[1.0], solint=solint,
calsour=selfcal_sources, aparm=[3.0, 0.0, 0.0, 0.0, 0.0, 2.0, fring_snr, 0.0, 1.0], dparm=[2.0]);

tgtsplit = split(indata=uvdata);
tgtmulti = multi(indata=tgtsplit, sources=["3C84"], aparm=[0.067]);
tgtdata = uvprm(indata=tgtmulti, docalib=1.0, channel=0.0, bif=0.0, stokes="I");

tgtimage = imagr(indata=tgtdata, sources=["3C84"], freqid=1.0, subarray=1.0, docalib=2.0, cellsize=[0.005, 0.005], niter=100.0, uvwtfn="U", chinc=16.0,
nchav=16.0, dotv=-1.0, stokes="I", imsize=[512.0, 512.0], dopol=0.0, robust=0.0, xtype=5.0, ytype=5.0, allokay=-1.0, gain=0.05, minpatch=256.0, imagrprm=
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1e-5, 1.0]);
```

Benefits of a Functional Language

- Enables minimal recomputation framework (see Des' talk)
- Parallelization; comes for free with Swift (as does remote execution)
- Sandboxing; observatory based computing

ParselGraph

- Generate dataflow graphs from executing ParselTongue scripts



Bringing it to the User

- ParselTongue requires Obit;
  building Obit has always been hard
- Mac OS X: Homebrew

```
brew install
  http://www.jive.nl/parseltongue/releases/obit.rb
brew install
  http://www.jive.nl/parseltongue/releases/parseltongue.rb
```

- Ubuntu: PPA

```
sudo add-apt-repository ppa:kettenis-w/parseltongue
sudo apt-get update
sudo apt-get install parseltongue
```

Bringing it to the User

- Other Linux: openSUSE Build Service

  `yum localinstall http://download.opensuse.org/repositories/home:/kettenis/Fedora_21/x86_64/obit-22JUN10i-6.1.x86_64.rpm`

- Currently builds Obit packages for:
  - Fedora 19, 20, 21
  - RedHat Enterprise Linux 6, 7
  - CentOS 6, 7
  - Scientific Linux 6, 7
  - openSUSE 13.1, 13.2
  - SUSE Linux Enterprise 12

"The research leading to these results has received funding from the European Commission Seventh Framework Programme (FP/2007-2013) under grant agreement No 283393 (RadioNet3)"