CRUSH

LABOCA 2007

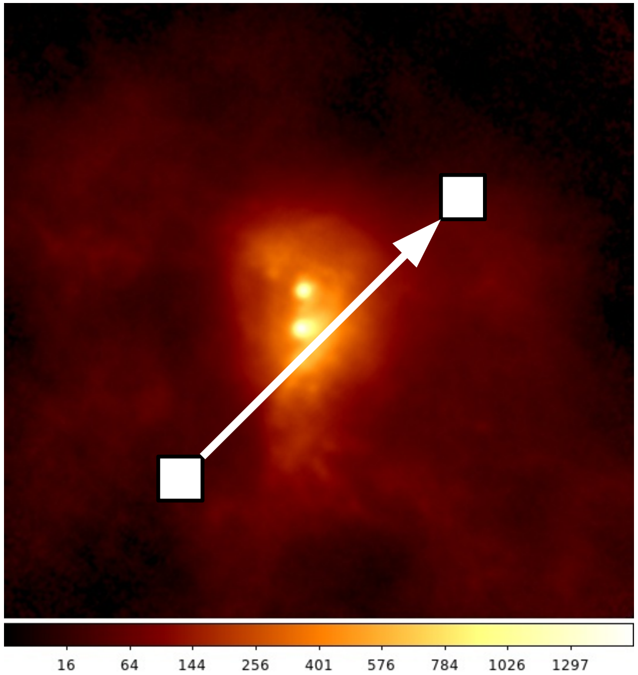
Data reduction and imaging for future (sub)millimeter arrays

Attila Kovács
Caltech

SPIE 2014, Montreal

www.submm.caltech.edu/~sharc/crush

Ground-based imaging in the (sub)millimeter

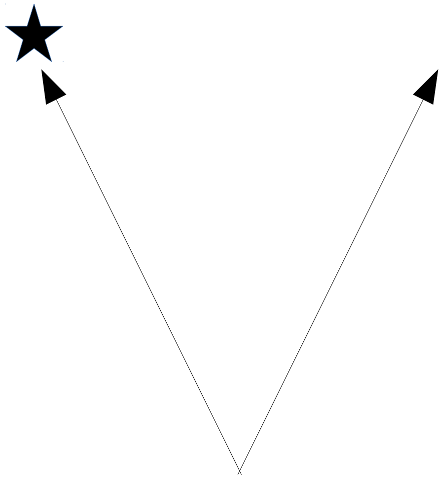


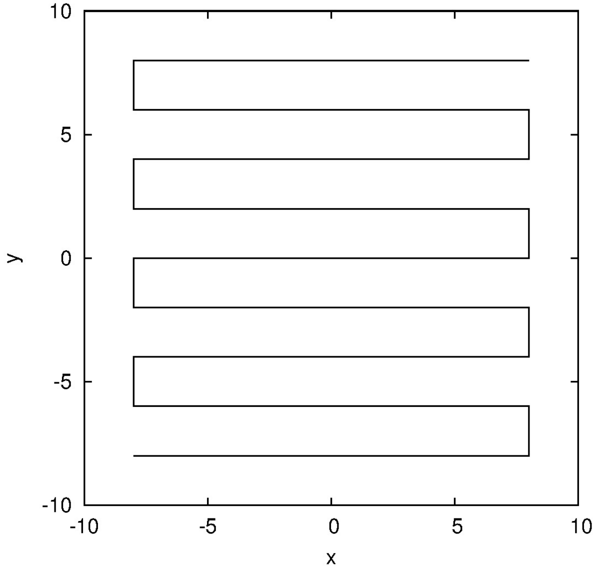
MAKO 2013

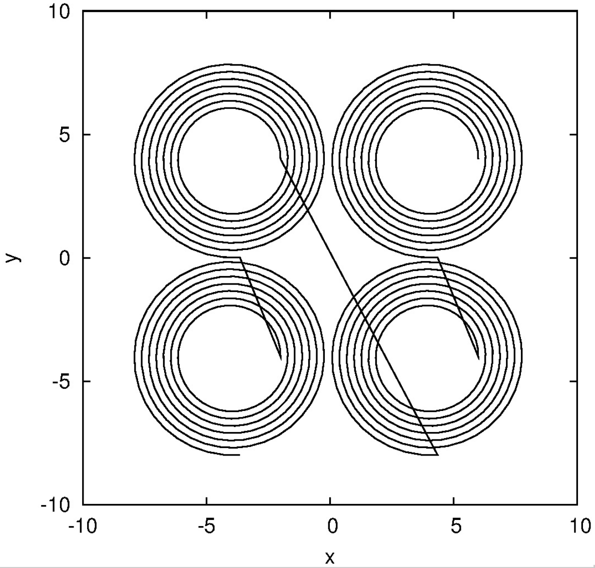
Highly variable atmosphere that is a million times brighter than what we look for...

Move Fast!

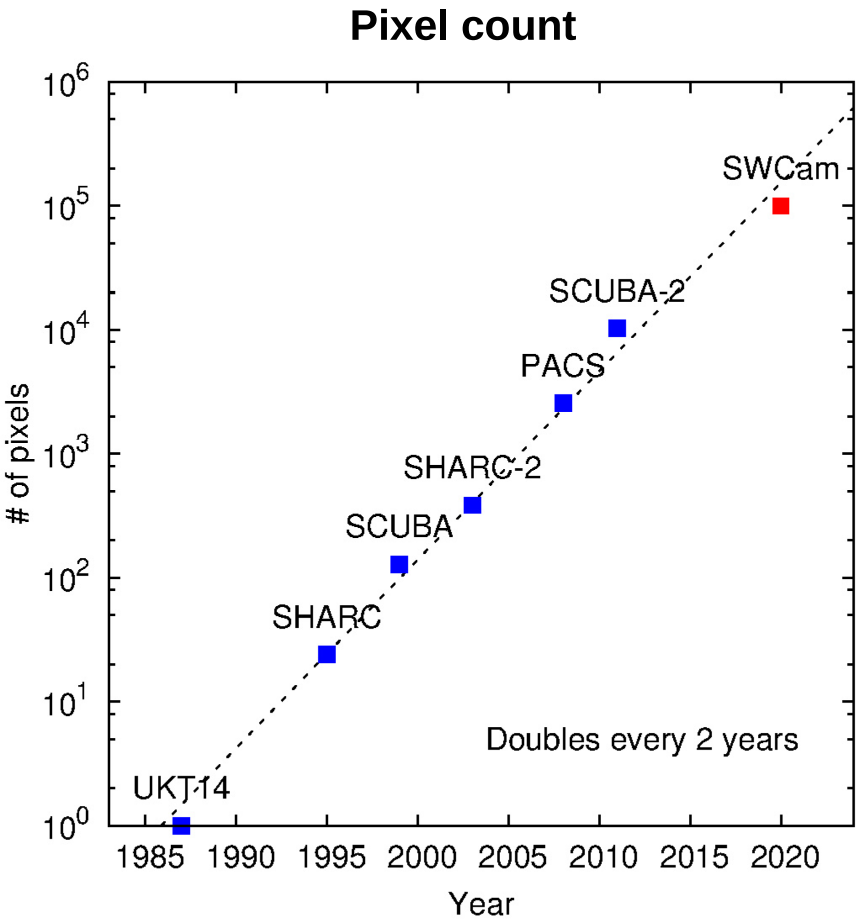
for improved sensitivity
for recovery of large scales...

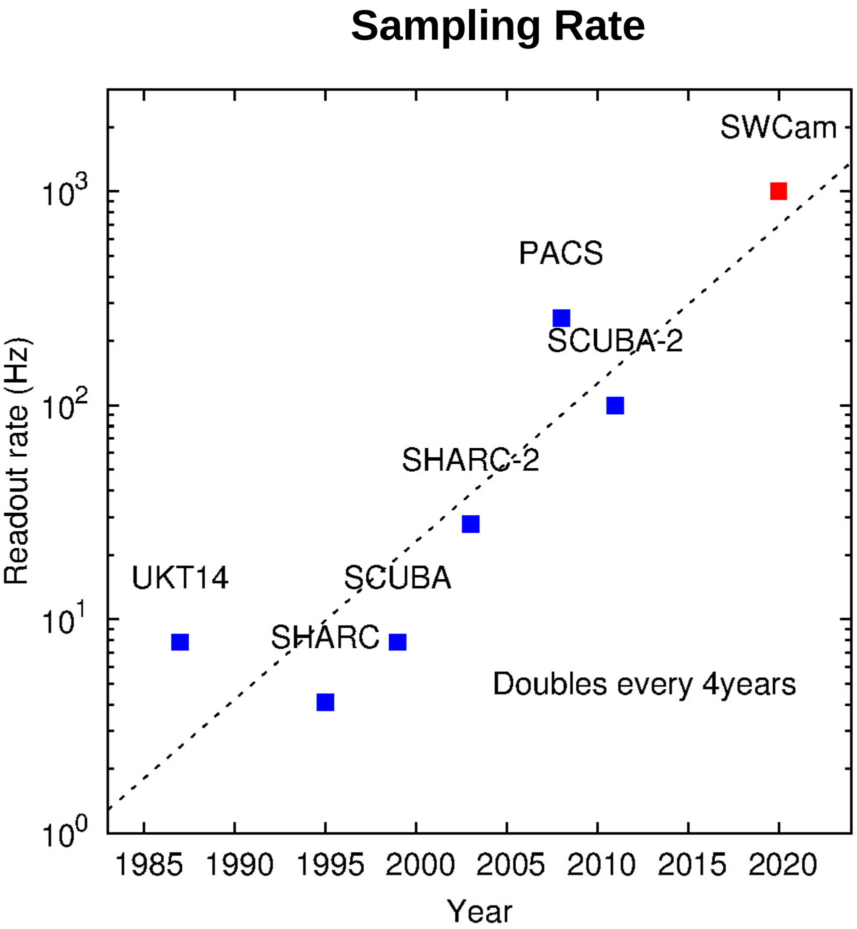




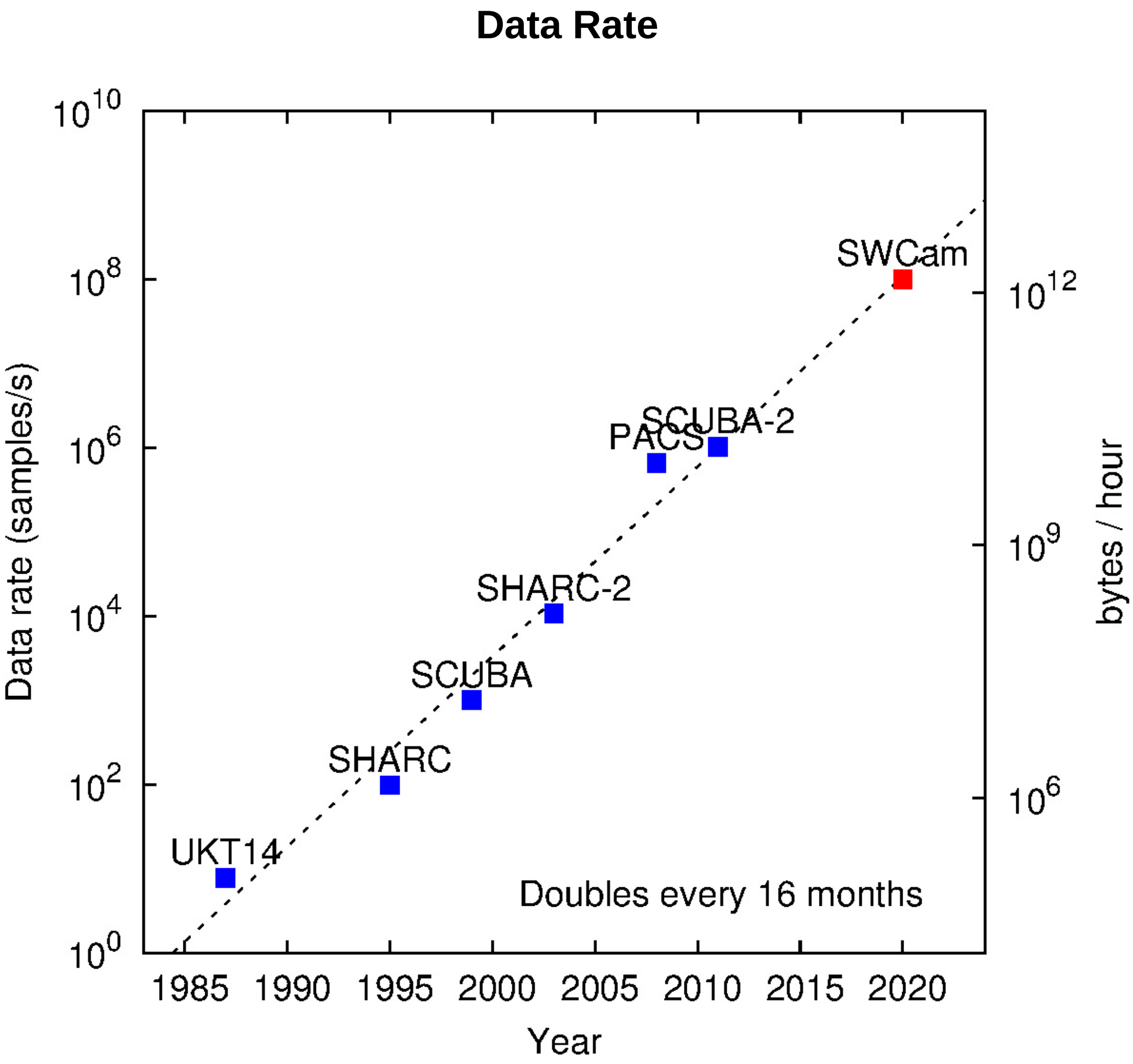


A Data Rate Challenge

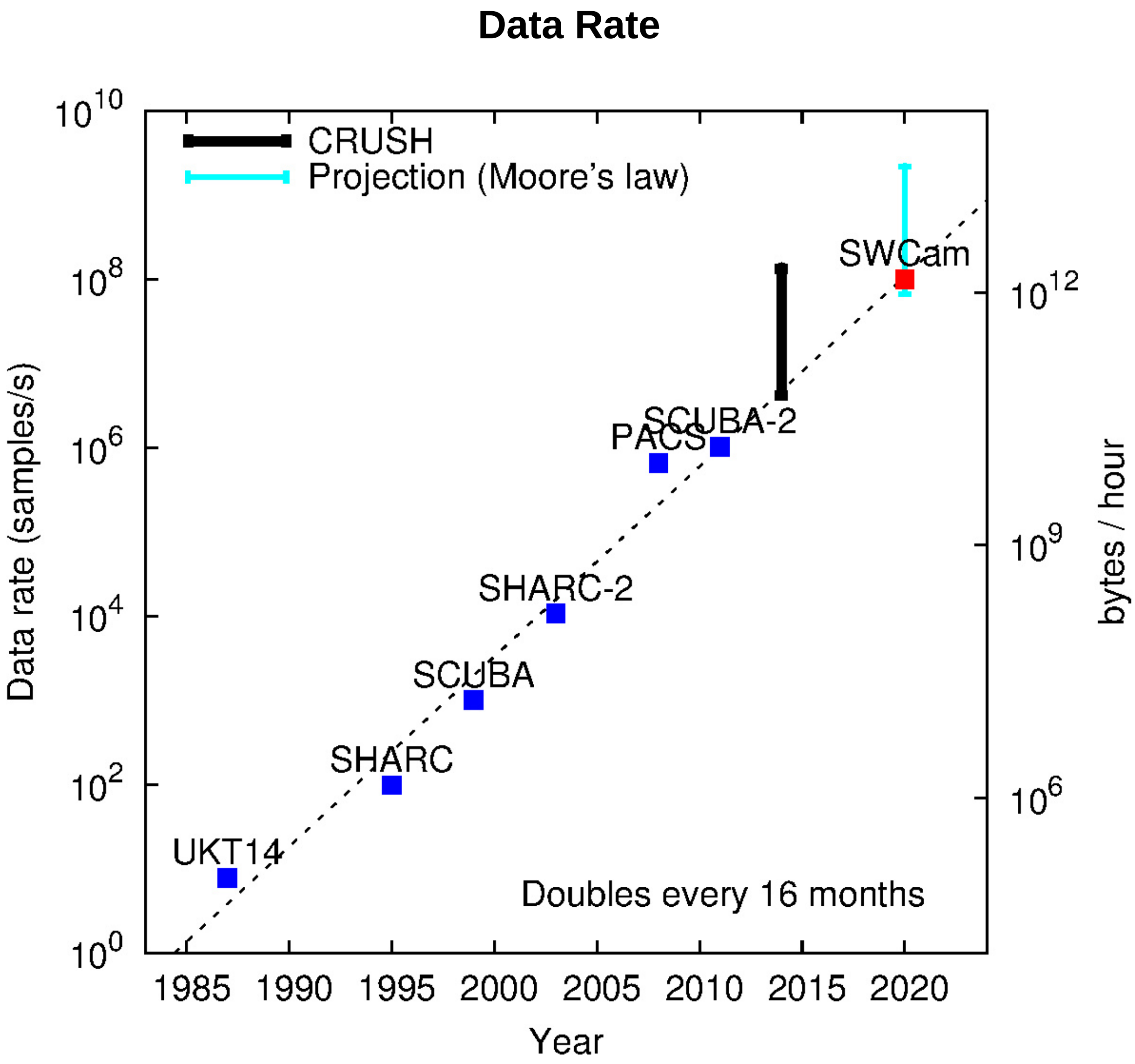




A Data Rate Challenge



A Data Rate Challenge



Programming Language(s)

What language(s) would you use for high performance computing?

CUDA / **OpenCL**

C / **C++** / **Fortran**

Java

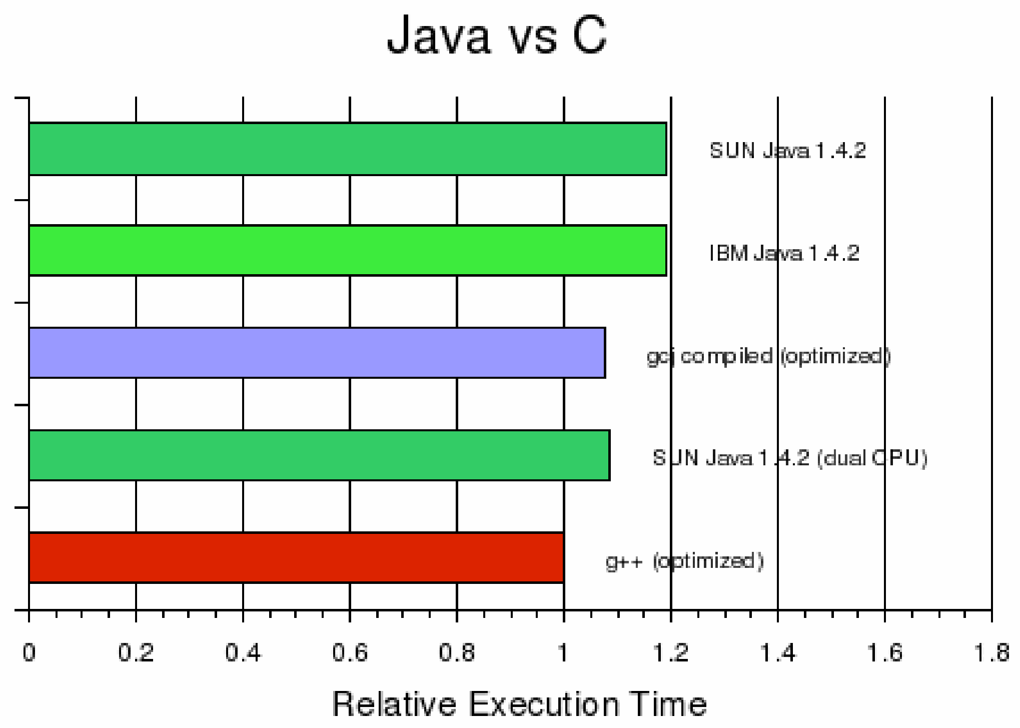
Python

Postscript / **LOGO**...

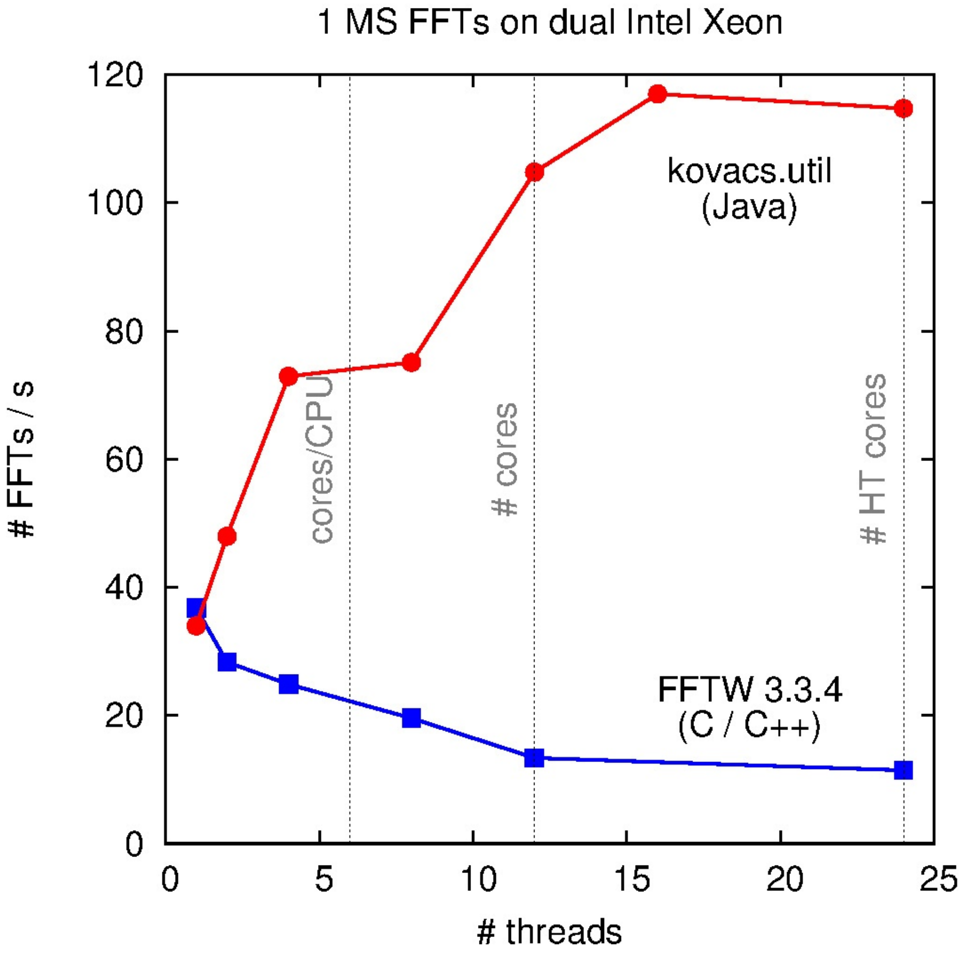
Java vs C/C++

Java is 5-10% slower than the most brutally optimized C/C++...

and can be faster...



in 2005...

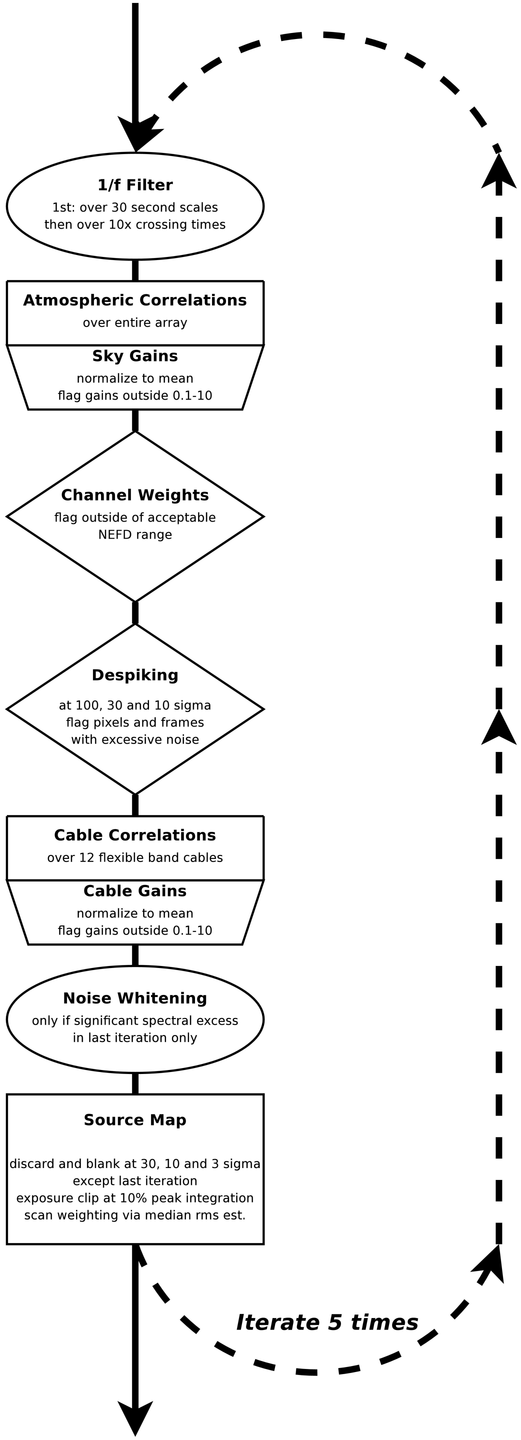


vs. FFTW

CRUSH: An introduction

Pioneering a new paradigm for scanning mode data...

2002	**CRUSH** (SHARC-2, LABOCA, SABOCA, APEX-SZ, p-ArTeMiS, GISMO, SCUBA-2, MAKO, SHARC...)
2002	**sharcsolve** (SHARC-2)
2006	**BoA** (LABOCA, SABOCA, APEX-SZ, ArTeMiS)
2011	**SMURF** (SCUBA-2)
2013	**MOPSIC** (GISMO, NIKA?)



1/f Filter
1st: over 30 second scales
then over 10x crossing times

Atmospheric Correlations
over entire array

Sky Gains
normalize to mean
flag gains outside 0.1-10

Channel Weights
flag outside of acceptable
NEFD range

Despiking
at 100, 30 and 10 sigma
flag pixels and frames
with excessive noise

Cable Correlations
over 12 flexible band cables

Cable Gains
normalize to mean
flag gains outside 0.1-10

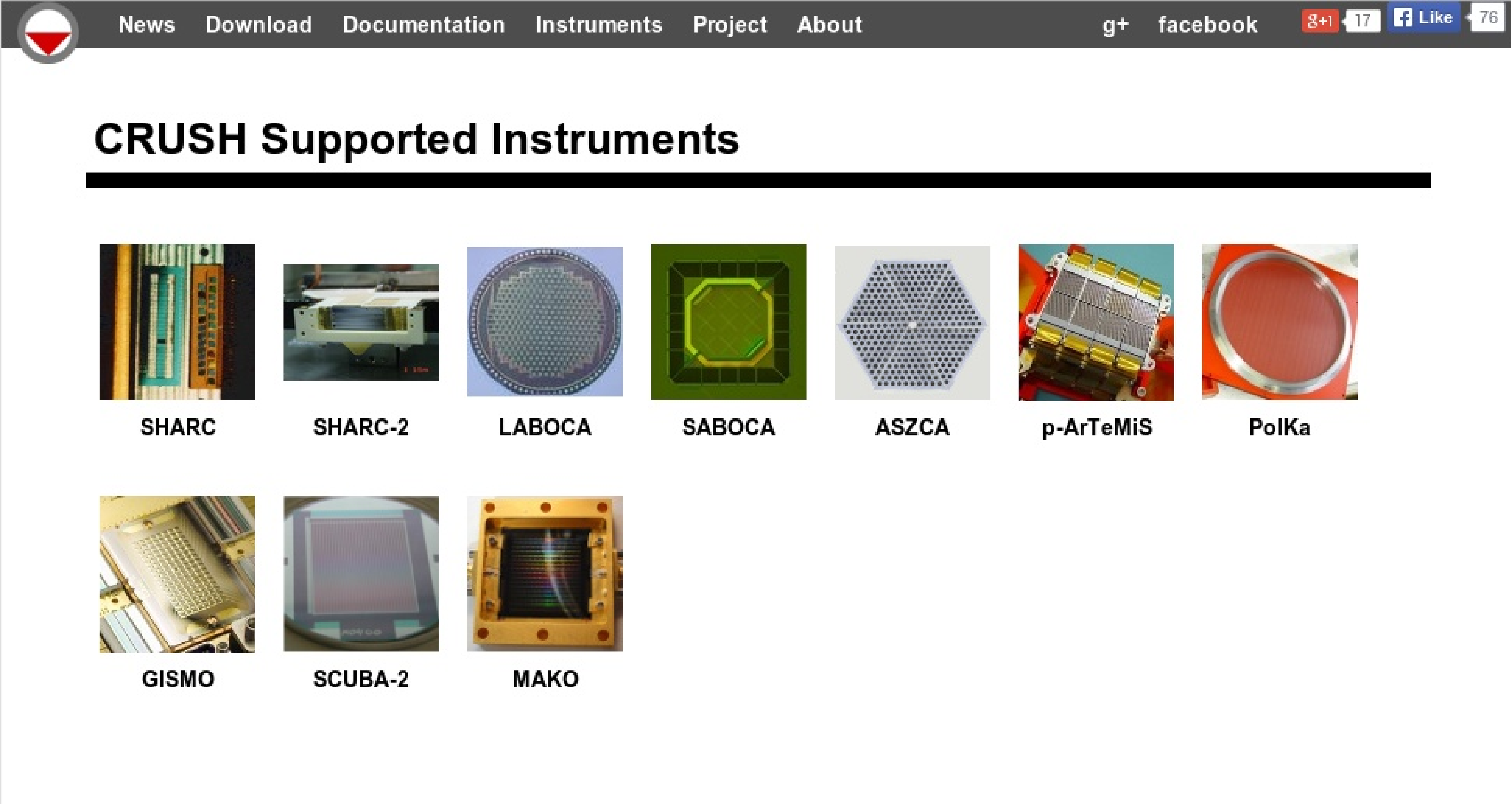
Noise Whitening
only if significant spectral excess
in last iteration only

Source Map
discard and blank at 30, 10 and 3 sigma
except last iteration
exposure clip at 10% peak integration
scan weighting via median rms est.

Iterate 5 times

100% Pure Java

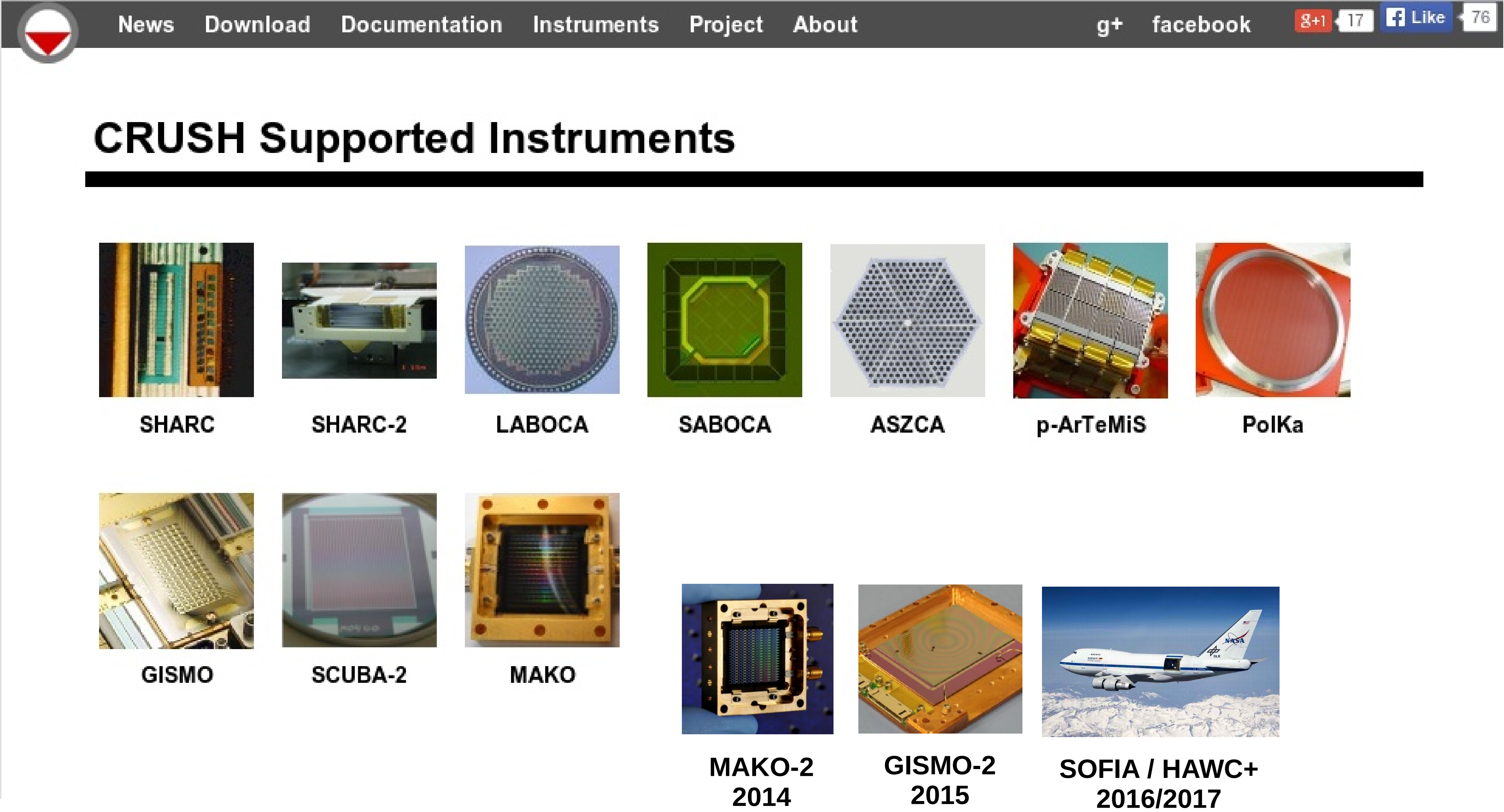
Tarball / ZIP
RPM & **Debian** packages



CRUSH Supported Instruments

SHARC
SHARC-2
LABOCA
SABOCA
ASZCA
p-ArTeMiS
PolKa

GISMO
SCUBA-2
MAKO



CRUSH Supported Instruments

SHARC

SHARC-2

LABOCA

SABOCA

ASZCA

p-ArTeMiS

PolKa

GISMO

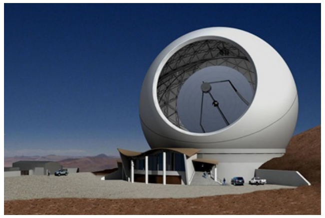
SCUBA-2

MAKO

MAKO-2
2014

GISMO-2
2015

SOFIA / HAWC+
2016/2017



CCAT / SWCam?
2020?

MAKO-2	9153-5	Tue.	11:50
GISMO-2	9153-18	Wed.	10:50
SWCam	9153-21	Wed.	11:50

CRUSH: A Pipeline

Not interactive but highly configurable...

Correlated noise removal

Pixel / channel gain estimation

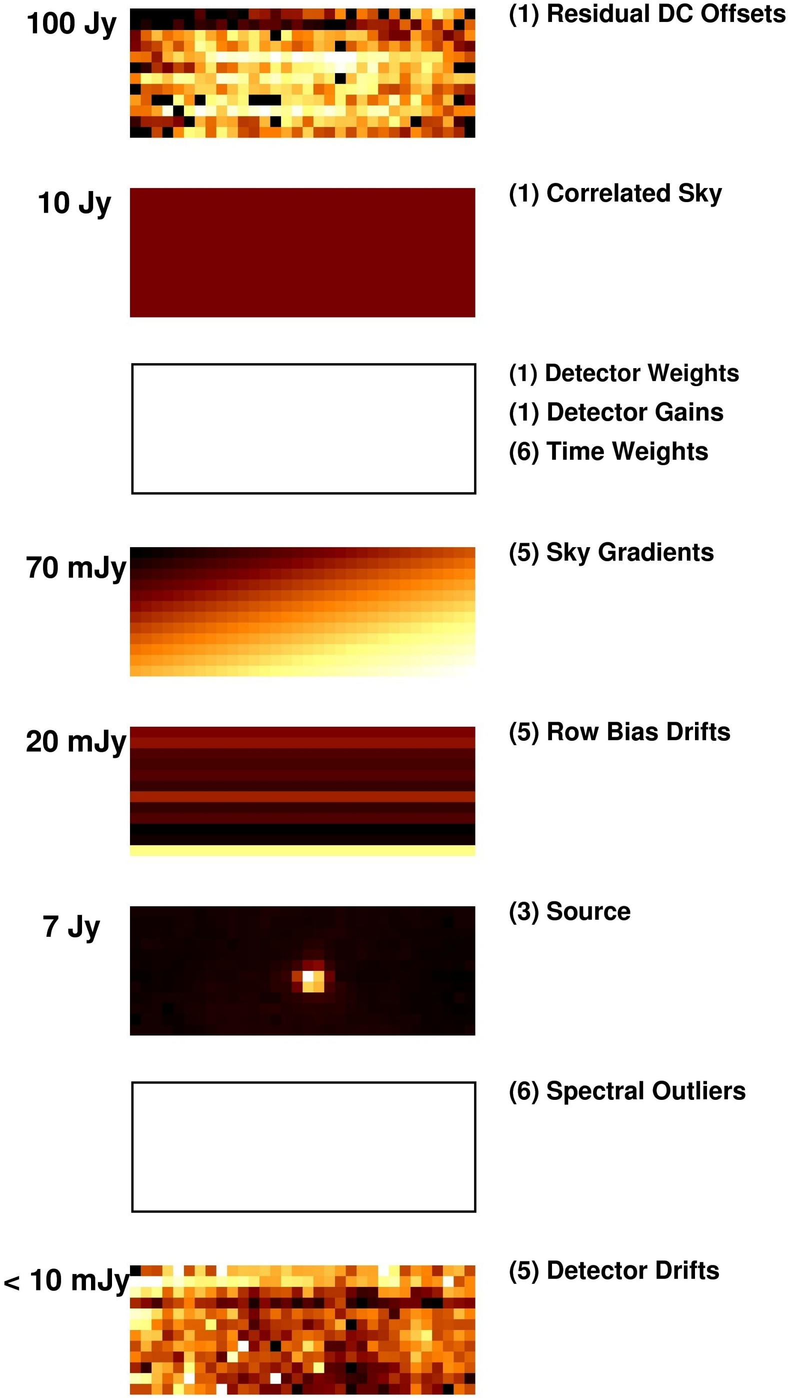
Noise weighting (by channel and/or time)

with rigorous accounting of lost degrees of freedom!!!

Consistency checking

Spectral Filtering

Source Model



CRUSH: Notable Features

Point-source corrections

White noise maps

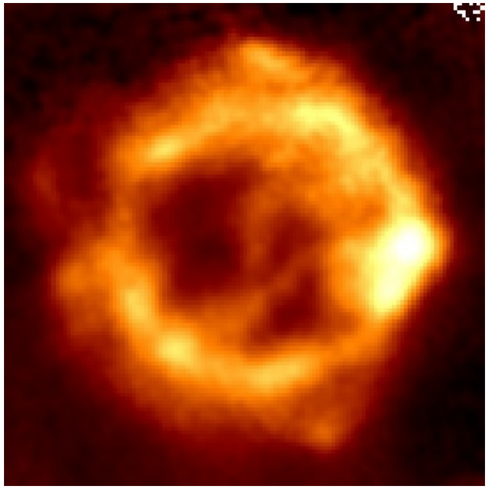
Rich FITS output

Jackknifing / Scrambling

Input models & test sources

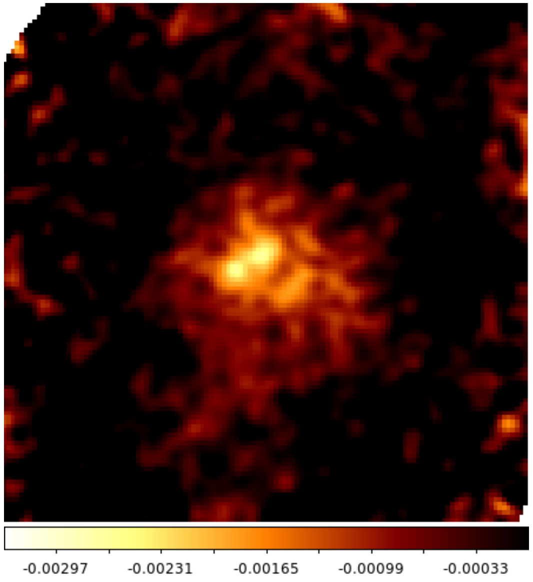
Instant focus, calibration, and pointing

More data products and logging support



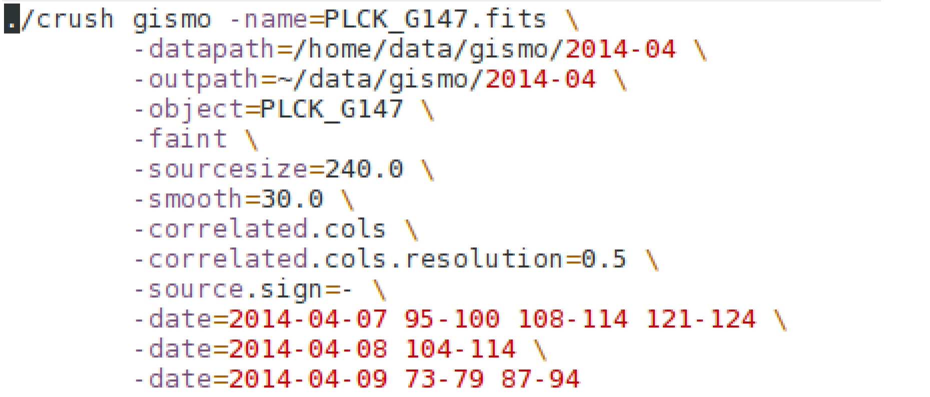
GISMO 2010

CRUSH: Configuration



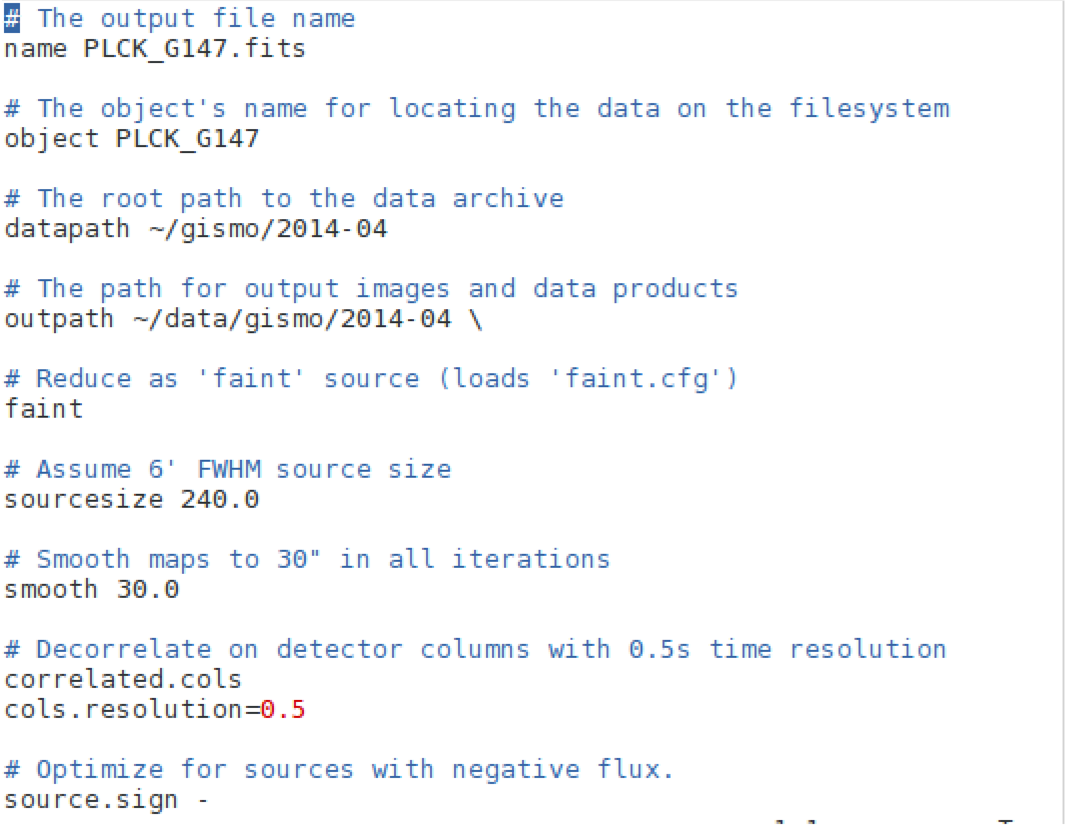
GISMO 2014

PLCK_G147.sh



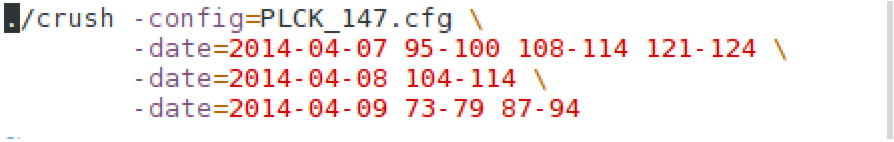
```
./crush gismo -name=PLCK_G147.fits \
        -datapath=/home/data/gismo/2014-04 \
        -outpath=~/data/gismo/2014-04 \
        -object=PLCK_G147 \
        -faint \
        -sourcesize=240.0 \
        -smooth=30.0 \
        -correlated.cols \
        -correlated.cols.resolution=0.5 \
        -source.sign=- \
        -date=2014-04-07 95-100 108-114 121-124 \
        -date=2014-04-08 104-114 \
        -date=2014-04-09 73-79 87-94
```

PLCK_G147.cfg



```
# The output file name
name PLCK_G147.fits

# The object's name for locating the data on the filesystem
object PLCK_G147

# The root path to the data archive
datapath ~/gismo/2014-04

# The path for output images and data products
outpath ~/data/gismo/2014-04 \

# Reduce as 'faint' source (loads 'faint.cfg')
faint

# Assume 6' FWHM source size
sourcesize 240.0

# Smooth maps to 30" in all iterations
smooth 30.0

# Decorrelate on detector columns with 0.5s time resolution
correlated.cols
cols.resolution=0.5

# Optimize for sources with negative flux.
source.sign -
```

PLCK_G147.short.sh



```
./crush -config=PLCK_147.cfg \
        -date=2014-04-07 95-100 108-114 121-124 \
        -date=2014-04-08 104-114 \
        -date=2014-04-09 73-79 87-94
```

CRUSH: Conditional Configuration

1. Simple conditions based on other settings

Set 1/f stability timescale to 15 seconds when the 'extended' option is set.

```
[extended] stability 15
```

Set FITS output name when 'system=horizontal' (reducing in horizontal coordinates).

```
[system?horizontal] name {?object}-altaz.fits
```

1. Interpreted conditions

Turn off spatial filtering of the source for the last 3 iterations

```
iteration.[last-2] forget source.filter
```

Load a configuration file for scans taken between the specified dates

```
date.[2014.03.31-2014.04.14] config run10.cfg
```

Set the calibration constant (i.e. conversion to jansky) based on serial number

```
serial.[*-41086] jansky=1.96e-6
```

Specify the pixel positions (RCP) for a given MJD range

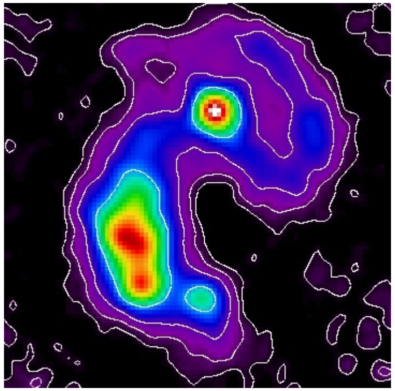
```
mjd.[55086.58-55112.44] rcp {@CRUSH}/laboca/2012-09.rcp
```

Automatically invoke 'bright' settings for Jupiter

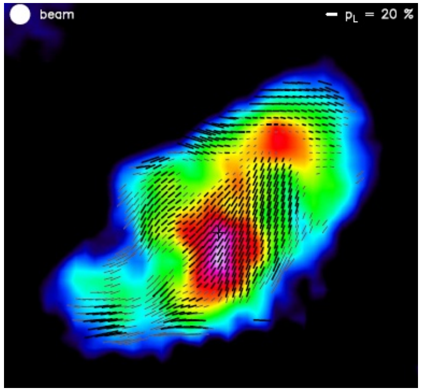
```
object.[Jupiter] bright
```

[....]

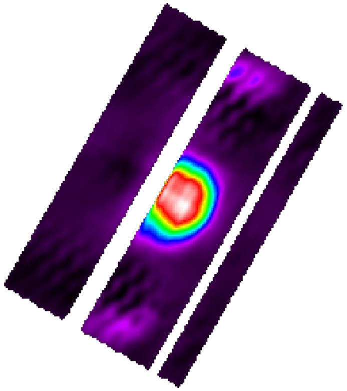
CRUSH: Source Models



Scalar Map



Polarization
I, Q, U, (P, F, A)

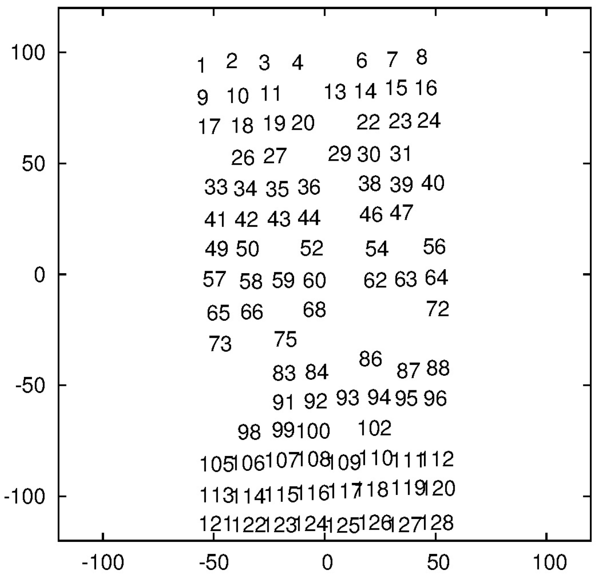


Deconvolved
Dual-beam Map

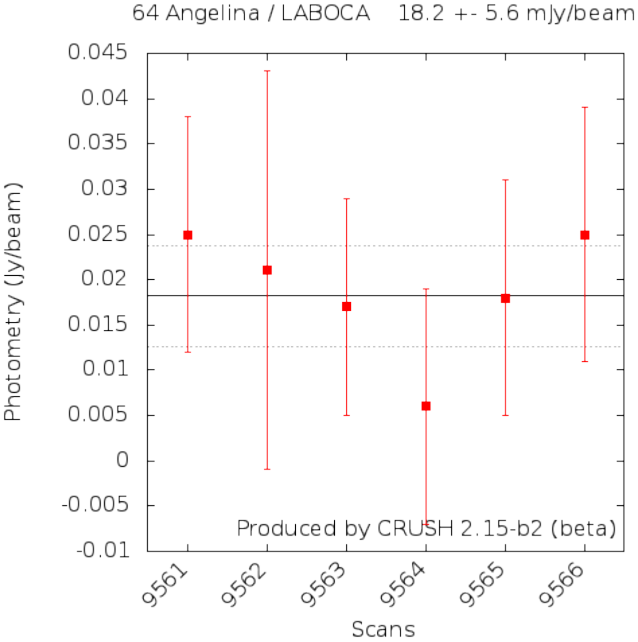
Point-source flux corrected

Independent map pixels

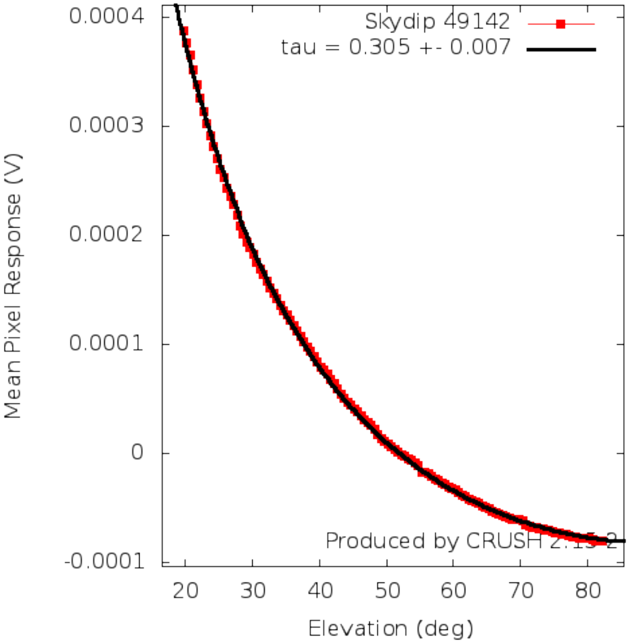
Noise & Integration Time



Beam Map(s)



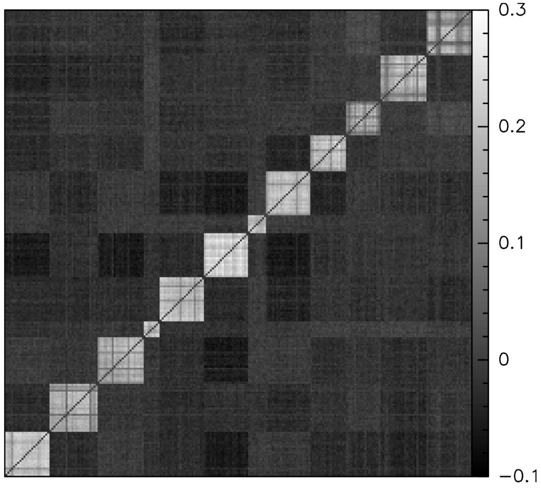
Photometry



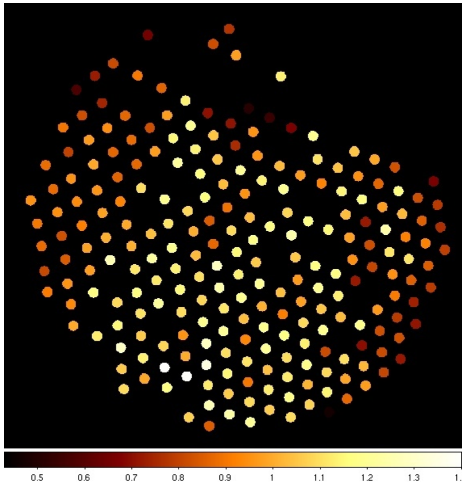
SkyDip

CRUSH: Output Products and Logging

- EPS figures (skydip, photometry)
- PNG thumbnails
- Residual timestreams
- Residual spectra
- Correlated signals
- Covariance Matrices
- Pixel characterization (gains, weights...)
- Pixel positions (beammap)
- ASCII log tables of user-specified columns....



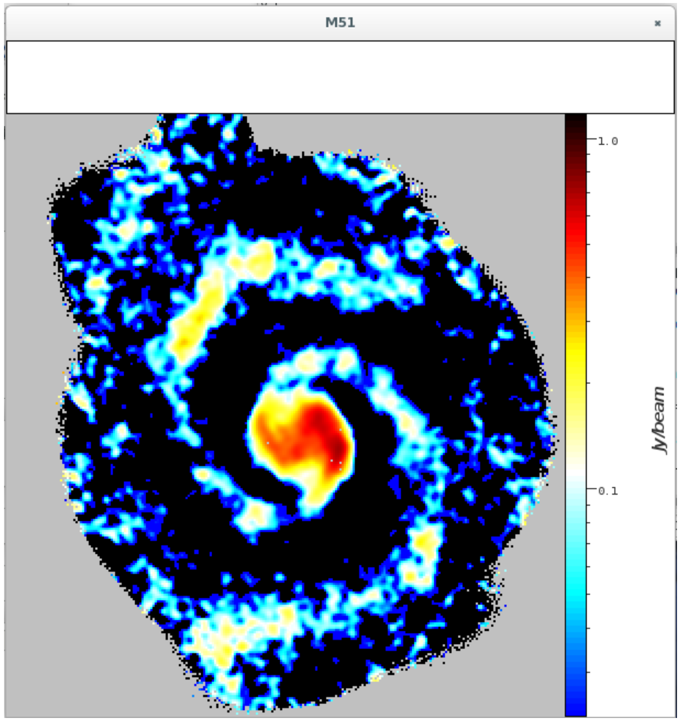
pixel-to-pixel covariance matrix



sky-noise gains vs. pixel positions (LABOCA)

CRUSH: Tools

`crush`	reduction pipeline
`imagetool`	manipulate FITS post reduction
`show`	Image display
`histogram`	Generate map histograms
`detect`	Point source extraction tool
`coadd`	Combine FITS images *Only if co-reducing is not an option!...*
`difference`	Look for differences in two images



show

SHARC-2 2004

I want CRUSH for my instrument...

Step 1. **Reading the data**

100 – 200 lines of Java

Step 2. **Instrument-specific extensions**

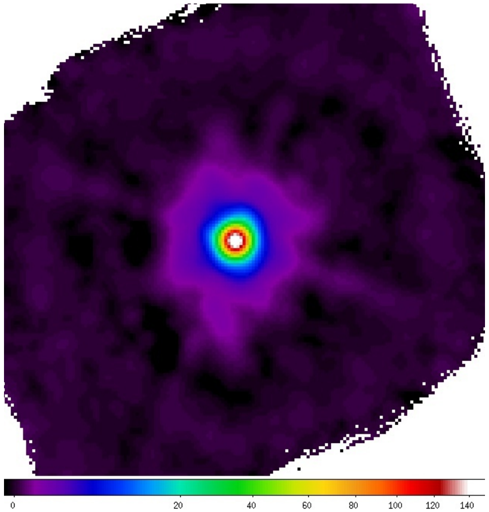
0 – 500 lines of Java

Step 3. **Configuration file**

10 – 100 lines of ASCII key/value definitions

Step 4. **Characterization data**

(E.g. pixel positions, initial weights/gains, wiring, bad pixels pointing model, pointing table, tau lookup, calibration table)

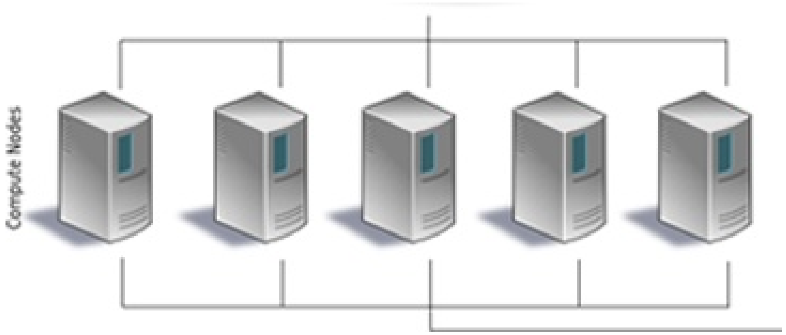


GISMO 2012

Into the Future...

Moore's Law

8-fold increase by 2020...



Further Parallelization

Computing cluster / nodes

GPU

Improved algorithms

perhaps another factor of 2...

CRUSH: The Highlights

The one that started it all...

The fastest of all...

runs on any platform

1-minute installation

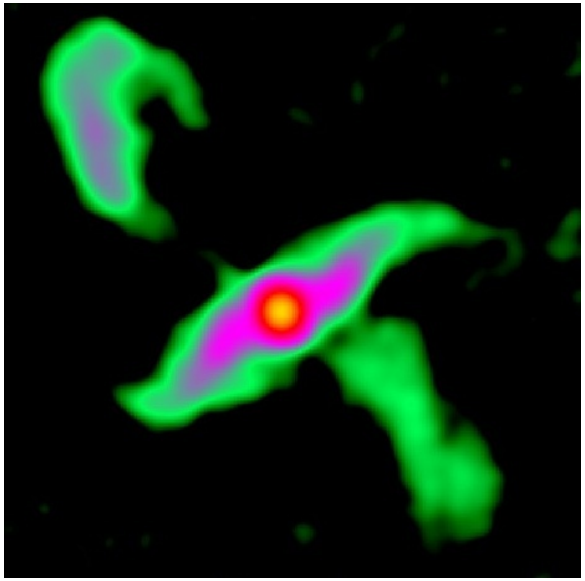
easy to use

poweful configurability

point-source corrected fluxes

best recovery of extended emission

adapt for any instrument...



LABOCA 2008

Acknowledgements
Tom McGlynn for
nom.tam.fits packages

Attila Kovács
attila@caltech.edu
www.submm.caltech.edu/~attila

CRUSH: A Programmer's Library

Do what you want...

Manipulate data with ease

Interactive frontend....

Also **kovacs.util** (on *SourceForge.net*) for Numerical Java

2D vectors, complex numbers & functions, weighted data...

2D image manipilation (coordinate grids)

Astronomical coordinates & conversions

Spherical projections (*Calabretta & Greisen 2002*)

FFTs

Special functions (Bessel, gamma, zeta, error function...)

Matrix inversion, SVD...