LOFAR Data Preparation using Default Pre-Processing Pipeline

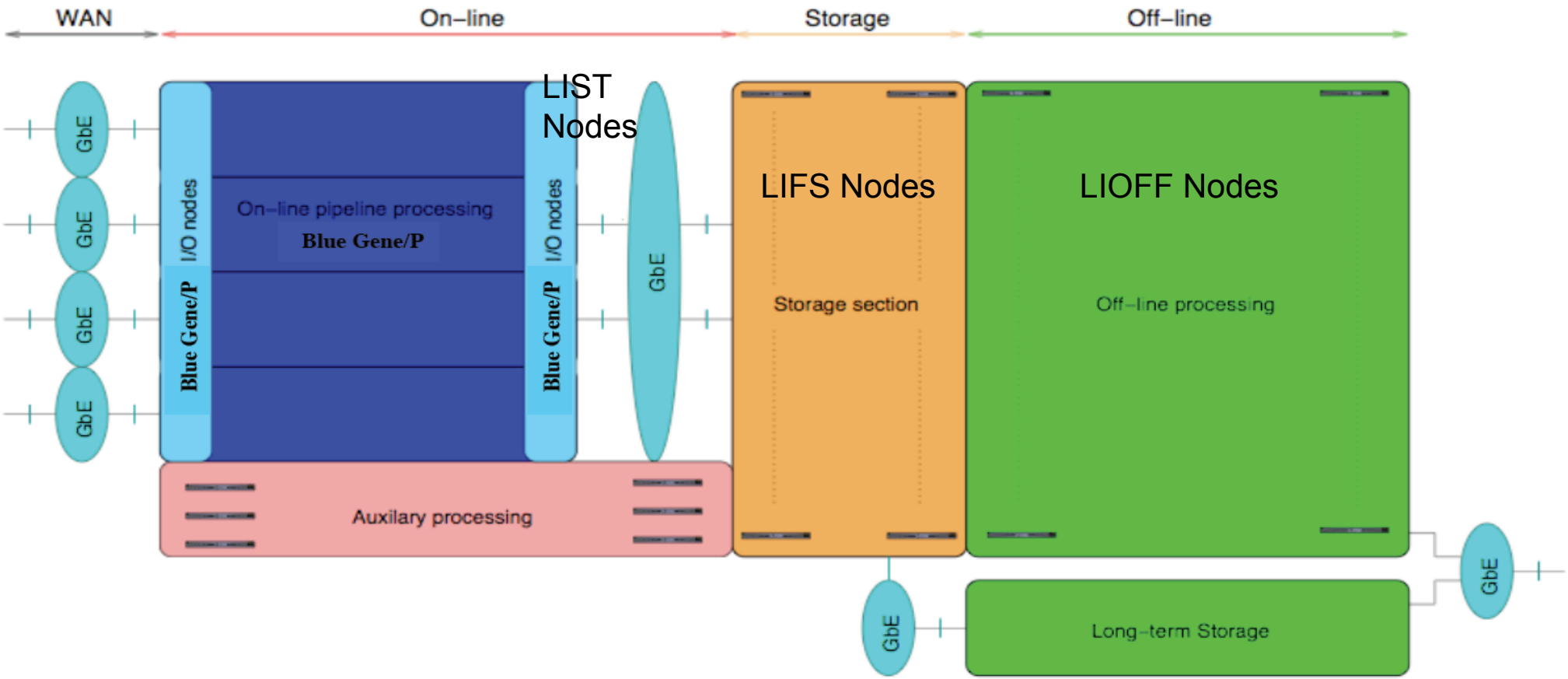
V.N.Pandey, Adriaan Renting, Joris Van zWieten
Marcel Loose, and Ger de Bruyn
Kapteyn Institute/ASTRON

February 10th, 2009
LOFAR Data Processing School, Dwingeloo

Outline

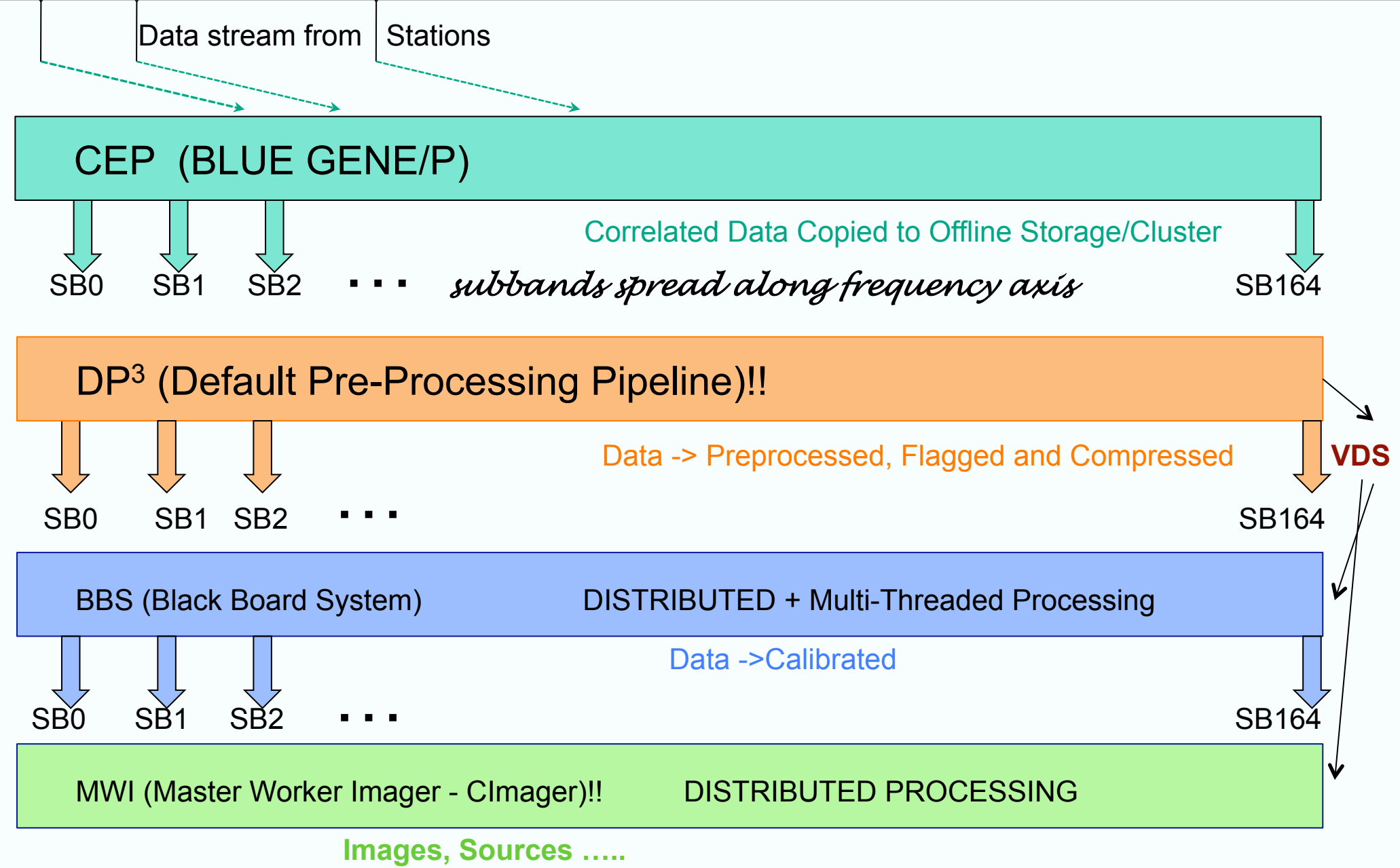
1. LOFAR processing chain – Standard Imaging Pipeline
2. Role of DPPP (or DP^3)
3. DPPP - Capabilities
4. DPPP – More aspects, What we do today
5. Summary

Processing Schematic - LOFAR



- BG/P *Data reception, transpose, correlation, beam-forming, de-dispersion*
  - Storage system *Short term storage of data, ~1 PByte, >100Gbps I/O*
    - Offline cluster *Calibration, data products, off-line analysis, ~1000 nodes*

Offline Automatic Processing – DP^3 Role



DP^3 Capabilities

Default Pre-Processing Pipeline (DP^3)

1. - What does it do? – Aims
2. - How does it work? - Implementation

DPPP- AIM/Objective

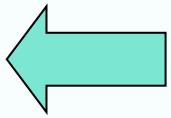
Carry out all Pre-Calibration common default steps in an efficient way.

		INTEGRATED	DISTRIBUTED
1.	Phase Correction due to diff clocks at stns		
2.	Sub-Band Pass correction		
3.	Global Band Pass correction		
4.	Flagging – Pre-Flags and Algorithm based		
5.	Compression along Frequency Axis		
6.	Compression along Time Axis		
7.	Combining Different Sub Bands into a single Mset.		

DPPP - Implementation

Software

Source Code (C++)
Run Script (Python)



Parameter Set File, Cluster Description files

- Runs in a Distributed way on different compute nodes (Not to be tried today)
  Each Measurement Set processed on one processor
- Reads the Measurement set only once for all steps

DP^3 – Capabilities

1. **Phase Correction due to clock drifts**

- Correct for phase differences introduced due to different clocks at stations
- Predetermined Table of corrections to be applied
- At this stage exact algorithm is not clear - TBD

2. **Sub-Band Bandpass correction**

- Bandpass shape within each Sub-band due to Polyphase filter bank
- Predetermined table as a function of frequency
- Already implemented at CEP (Blue Gene) so no more required in DP^3
- In case we need improvements may be implemented.

3. **Global Band Pass correction**

- Global Bandpass correction due to Antennae response to different Frequency
- Predetermined table as a function of frequency
- Once estimated accurately using BBS global solver would be implemented.

Generic Implementation to have multiple tables of corrections in one step

DP^3 – Capabilities

4. Flagging

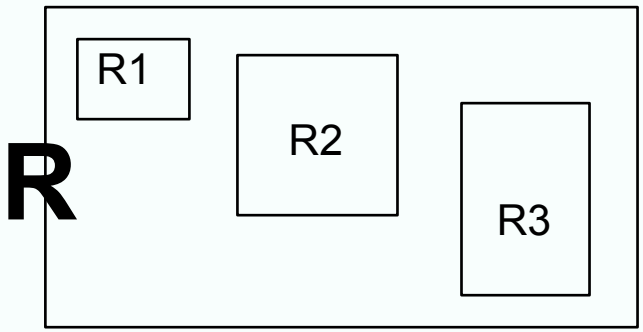
APRIOR Knowledge Based

REGION CUM CRITERIA APPROACH (under progress!!)
REGION: Time, Freq, baseline, Correlation, Direction
CRITERIA: Any condition a pixel in the region satisfies

- Build a mask using
  - existing flags
  - nan (0/0), infinite
  - channel selection
  - antenna
  - baseline length

  More options to be decided

Advantage: Provides region specific solution
Ex Threshold a function of correlation, baseline etc.



R1- C1, C2, ….
R2 – C1, C6…

ALGORITHM BASED

- Generic code to accommodate
  - different flagging algorithms
  - multiple flaggers
- Currently uses
  - Amplitude thresholding
  - MAD flagger (Hampel filter)

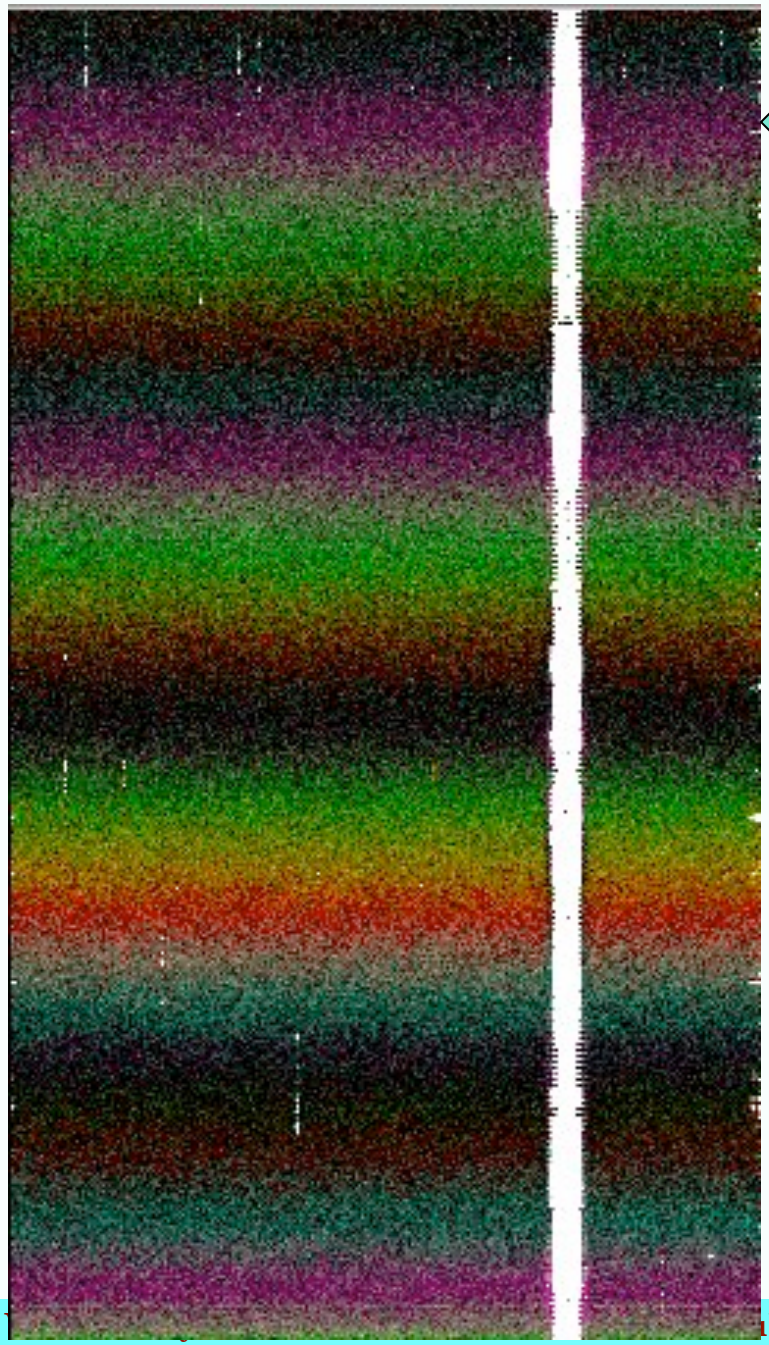
  (Two dimensional – freq & time)

-Mirroring on the ends- avoids edge effects
- Can run on data column desired
- Multiple flaggers implemented

- Satisfactorily results on actual data

DP^3 – current status: Flagger

MS9315 (40MHzOct 24,2008)

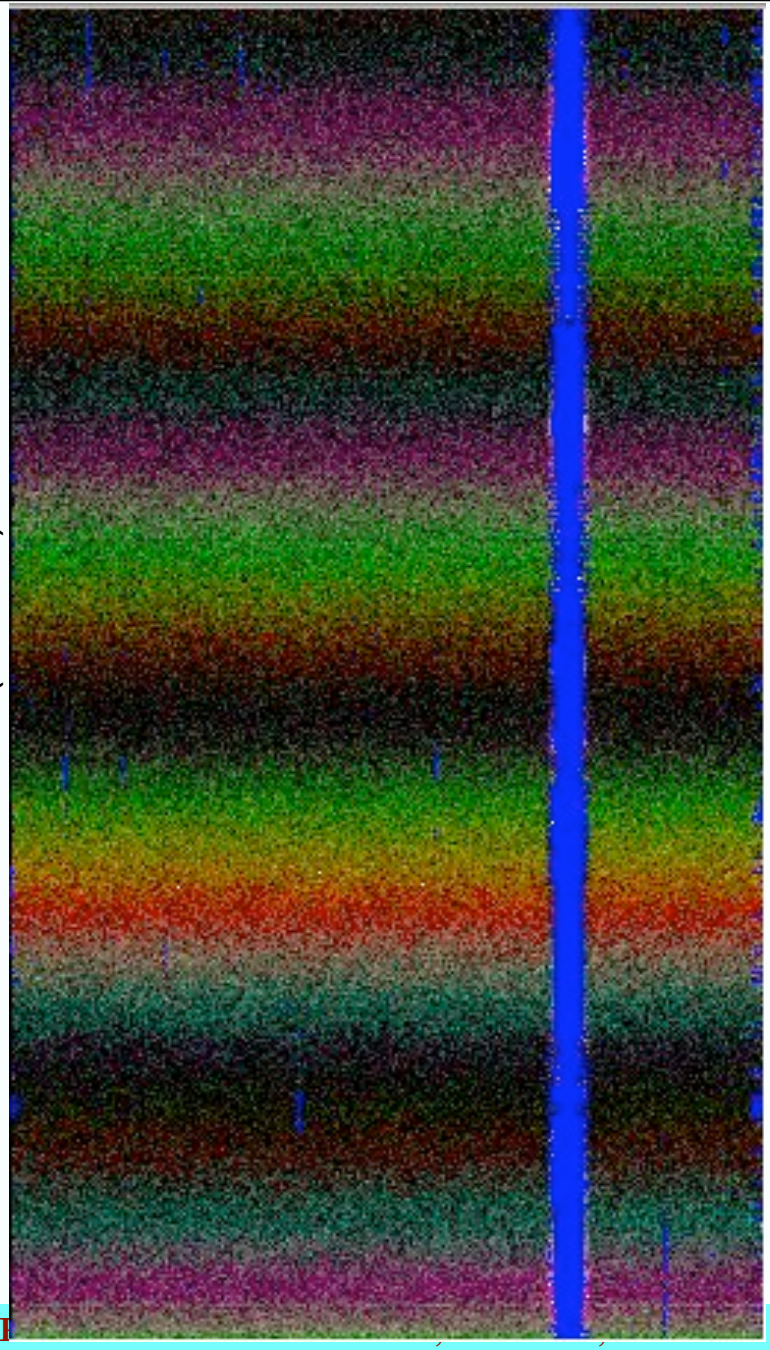


Before Flag

After Flag

Integ 30s

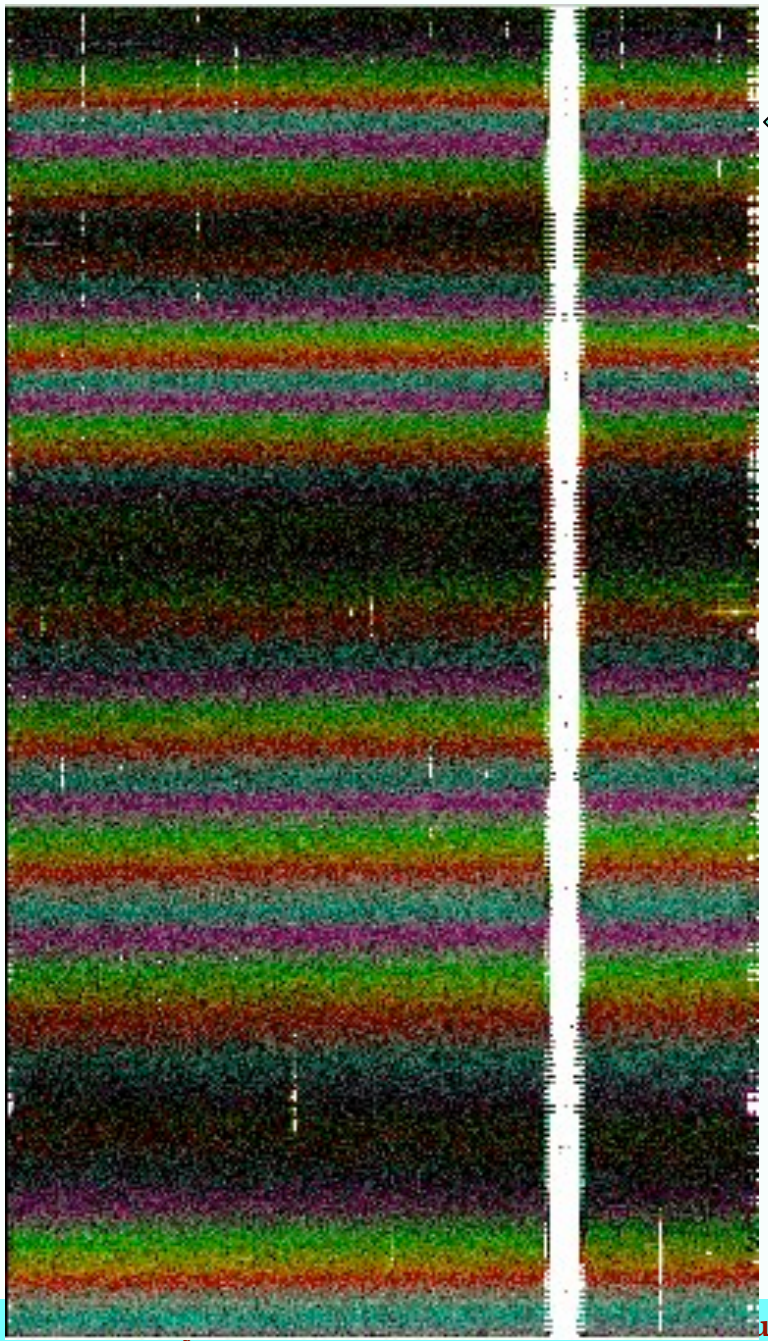
CS1_us0 and
CS1_us1
(XX corrleation)



No absolute Threshold Flagging

DP^3 – current status: Flagger

MS9315 (40MHzOct 25,2008)



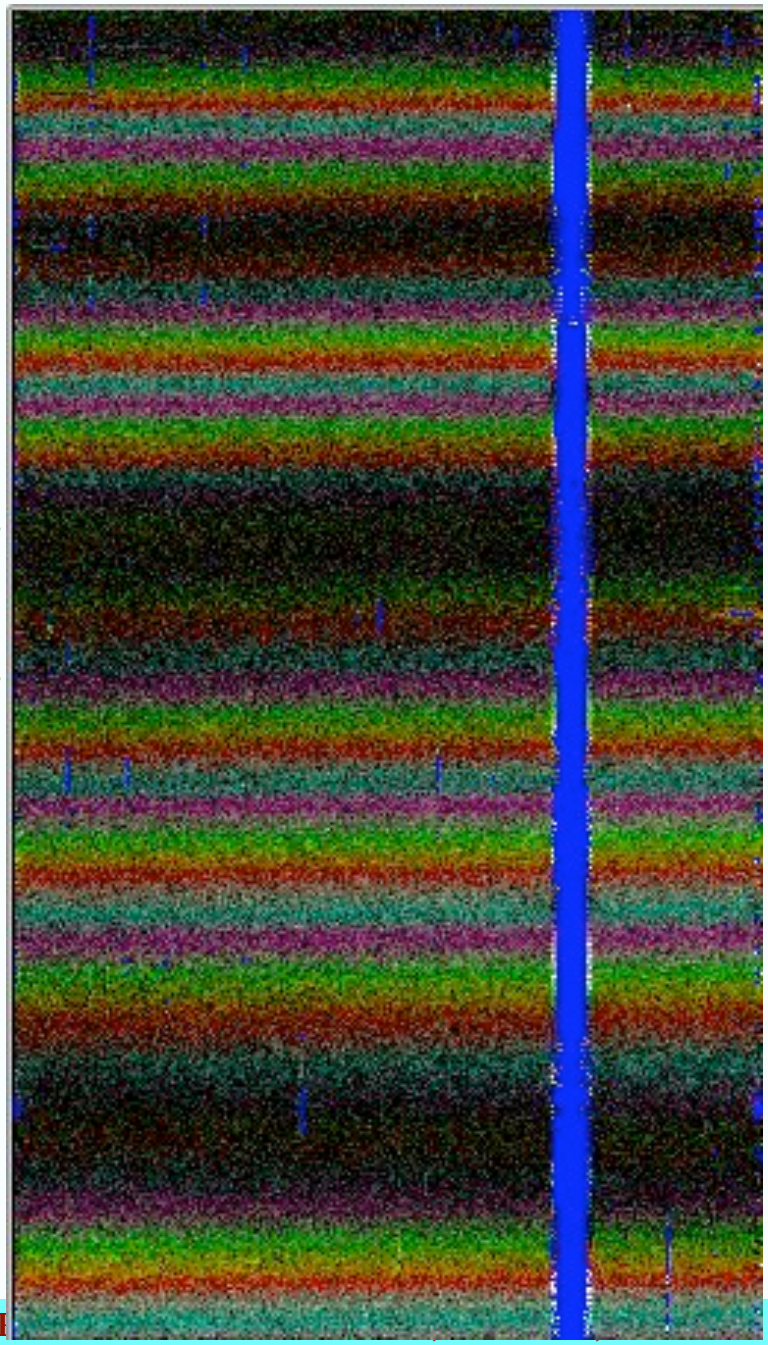
Before Flag

After Flag

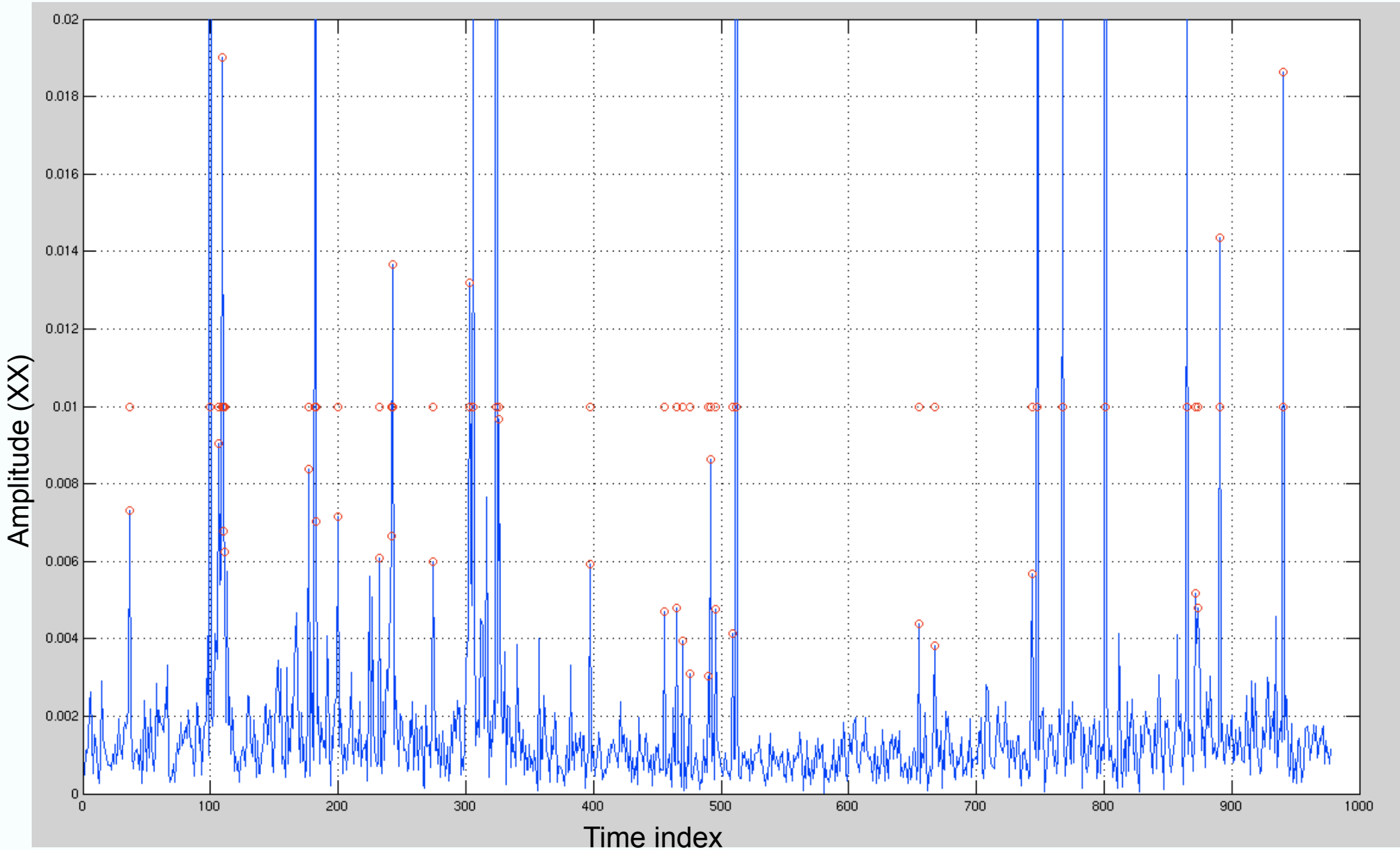
Integ 30s

CS1_us1 and
CS10_us2
(YY corrleation)

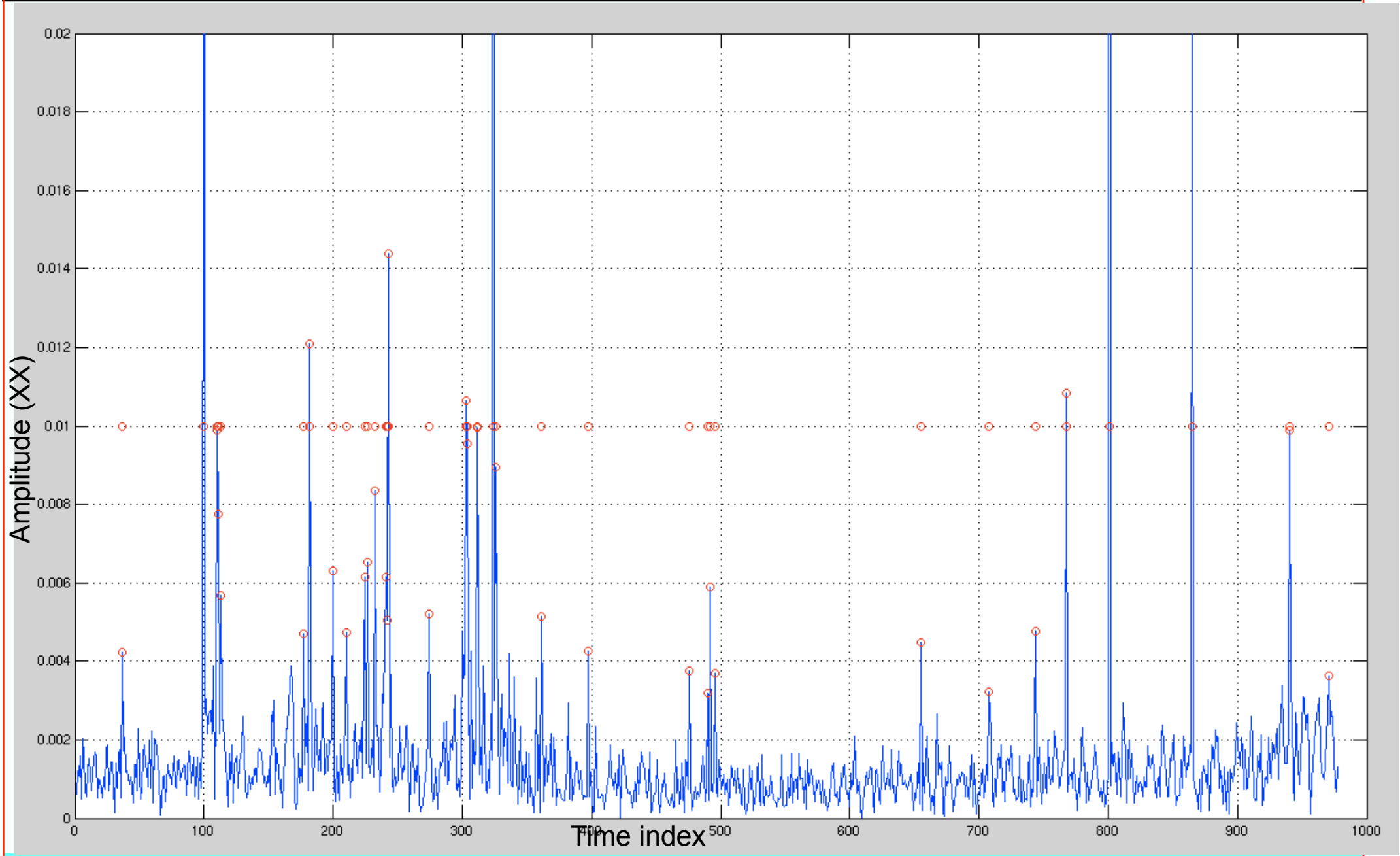
No absolute Threshold Flagging



DP^3 – Flagging: Example 1



DP^3 Flagging Example 2



DP3: Data Compression

5. Compression along frequency axis
6. Compression in time

- Uncompressed Data - >> 100 Terabytes/Day
- Implemented together, any one may be selected or switched off
- Compressed pixel = mean (unflagged values)
- Weight column appropriately modified depending on number of pixels flagged
- Time and Frequency of compressed (Averaged pixel) **
- Multiple stage compression allowed

Performance of DP^3

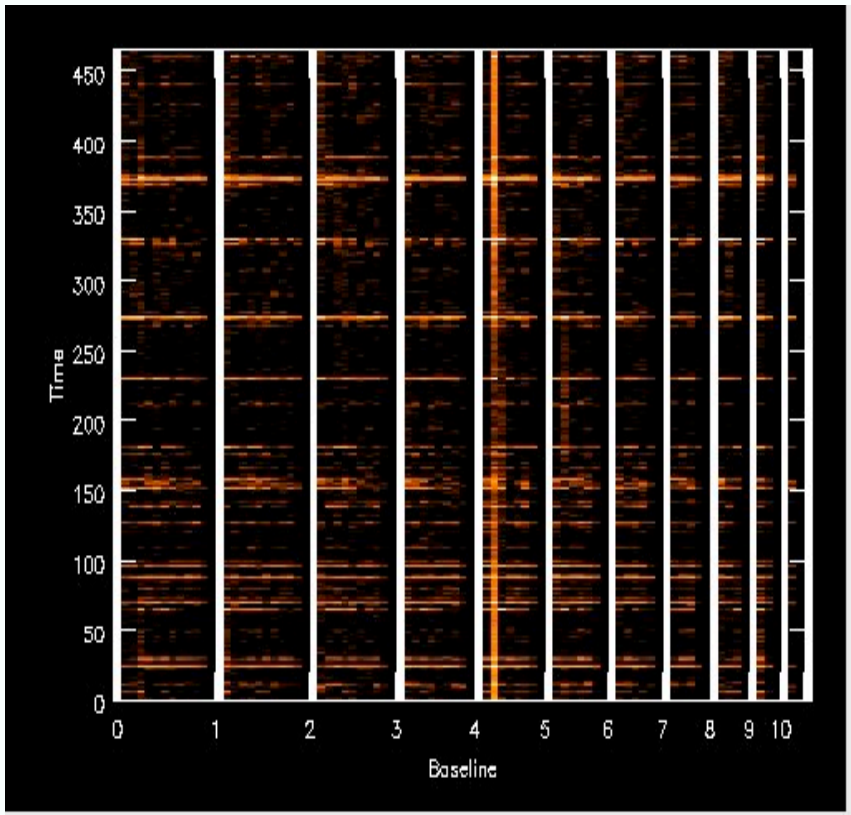
- >99% CPU usage most of the time

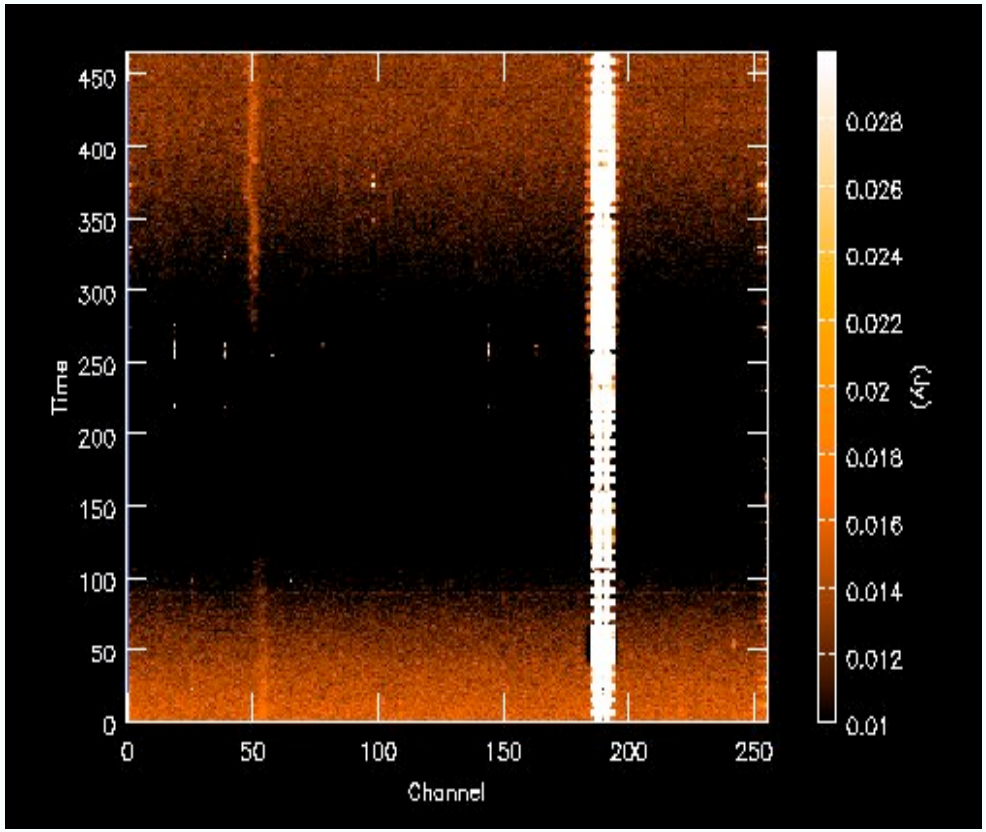
On 24hr data set, 30s integration, 256 channels
15x15 window : 4hours (2.0GHz processor)
Matlab : ~40 hours (uncompiled run)

Hands on DP^3 today

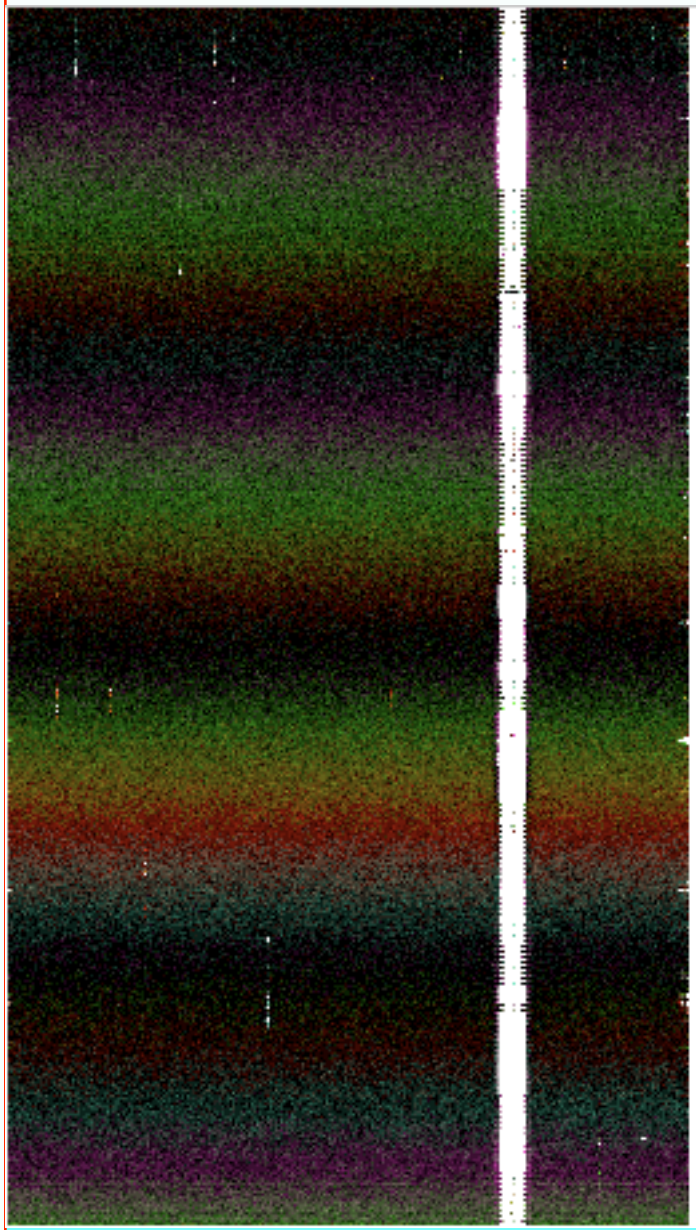
Exercise-05-DPPP

1. Pick up an Actual observation with LOFAR CS1
2. Run DP^3 for RFI Detection/Flagging
3. Look at how effective is the flagging
4. Do compression along time, frequency
5. Inspect the data





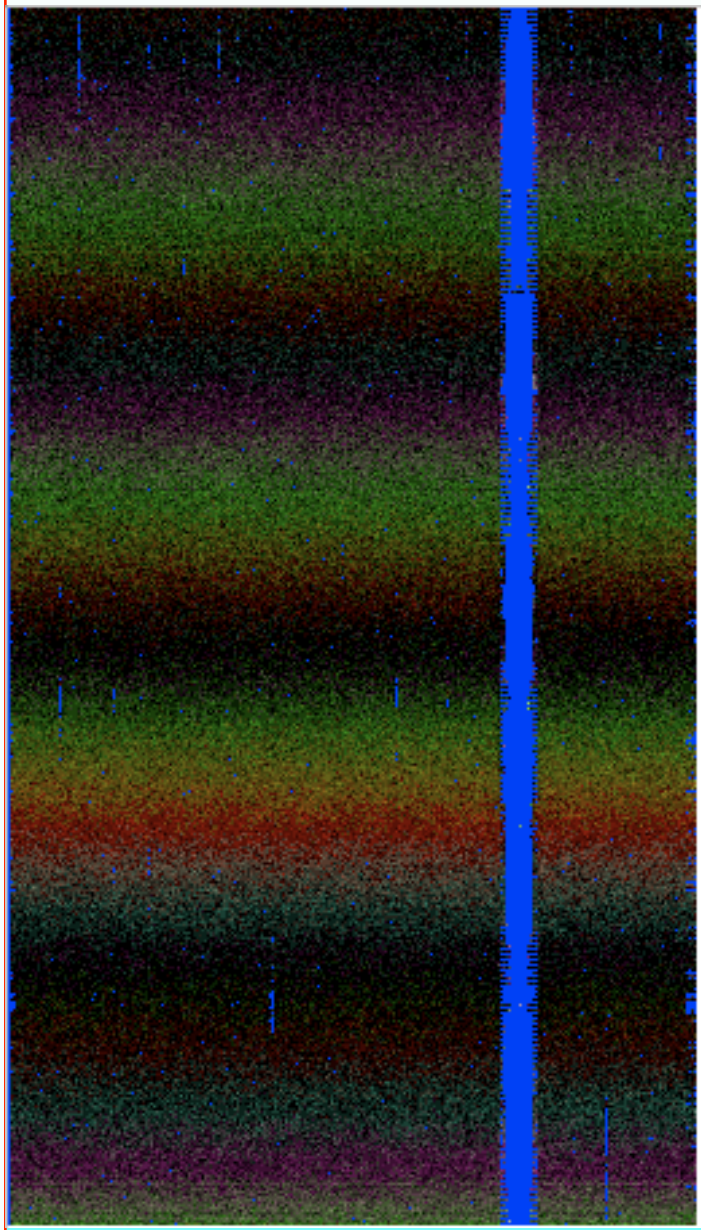
Channels



T
I
M
E

L2008_9315_SB12.MS
39.45MHz, 195KHz wide
256 channels, each channel 0.76KHz

Channels



TIME

Conclusions -Open issues/ Next steps

- Complete DP^3 Pipeline has been tested, works successfully.
- Compressed sub-bands passed successfully via MW Imager.
- Computational Speed improved – steps integrated in one.
- Integrated in the offline Pipeline.
- Code Generic to support different and multiple flaggers
- and Multiple levels of compression.
- Time and frequency of Averaged pixels in compressed MS under discussed.
- More computational bench mark tests.
- Efficient flagging criteria for mask to be discussed.
- Regularly used from ?
- Hands on Session ☺

Discussions??

!! Thank you !!

LOFAR User Software Group

PROJECT
- News & Changes (Archive)
- Key Science Projects
- LOFAR Network
- Glossary
- USG personnel

DEVELOPMENT
- Getting started
- General information
- Code repository
- Environment & Tools
- Supported platforms
- Testing
- Sample data sets

SOFTWARE
- Packages and Tools
- Libraries
- Visualization
- Downloads
- User contributions

DOCUMENTATION
- Workshops
- LOFAR Data products
- Minutes from meetings
- Work packages
- Presentations
- Pictures
- Wiki help

RELATED PAGES
USG CODE
USG DOXYGEN
USG WEBSTATS
USG FORUM
LOFAR WEBSITE
LOFAR WIKI

LOFAR Data Processing School :: Hands-on exercises

The workshop will mainly utilize some of the LOFAR offline cluster nodes to run the exercises. These machines are named "lioff0XX" where X goes from 21-29. To reach these machines, you have to first login to listfen. So ssh into listfen and then ssh into one of the working nodes.

On these machines, there is an area set aside named "/data/lofarschool" where all the exercises will be stored. Within that directory there will be two sub-directories, one called "users" which the students will use as working areas and one called "data" which will contain subdirectories for each of the exercises:

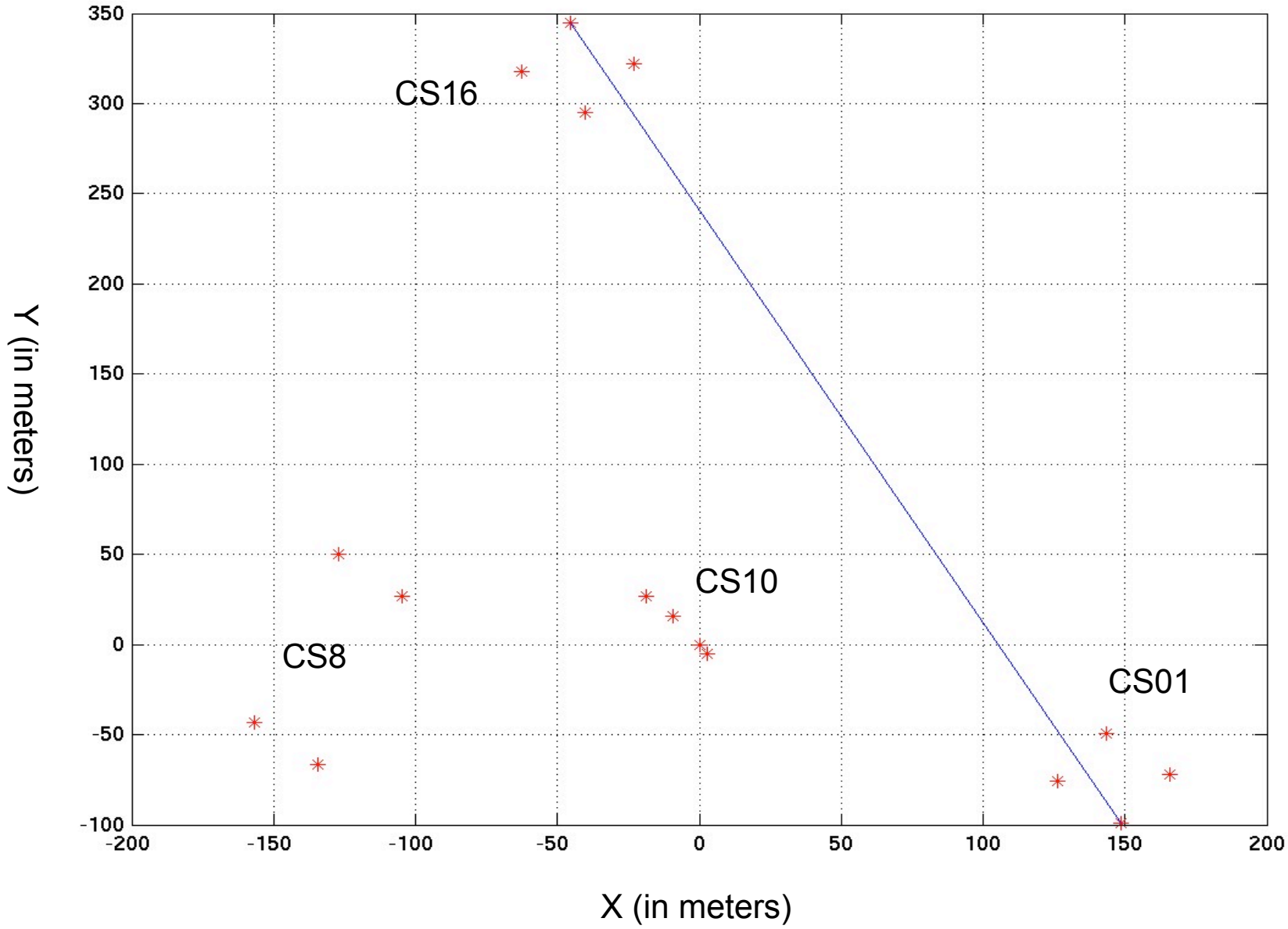
```
/data/lofarschool
      |-- data
      |   |-- Exercise_01
      |   |-- Exercise_02
      |   |-- Exercise_03
      |   `
      `-- users
```

Each directory subdirectory will contain all the inputs, expected outputs, scripts, logfiles, instructions, etc associated with a given exercise, i.e. :

```
lioff023:/data/lofarschool/data/Exercise_01> ll
total 8
-rw-r--r-- 1 wise wise    0 2009-02-04 19:26 example_01.log
-rw-r--r-- 1 wise wise    0 2009-02-04 19:26 example_01.py
drwxr-xr-x 2 wise wise 4096 2009-02-04 19:25 inputs/
-rw-r--r-- 1 wise wise    0 2009-02-04 19:27 instructions.txt
drwxr-xr-x 2 wise wise 4096 2009-02-04 19:25 outputs/
-rw-r--r-- 1 wise wise    0 2009-02-04 19:26 README
lioff023:/data/lofarschool/data/Exercise_01>
```

Everything in "/data/lofarschool/data" will be write protected. If users want to modify any of the input files or scripts, they can copy them to them working areas.

1. Exercise 01A - Station Data
2. Exercise 02B - Station Data
3. Exercise 03
4. Exercise 04
5. Exercise 05
6. Exercise 06


CS16
CS10
CS8
CS01
Y (in meters)
X (in meters)
350
300
250
200
150
100
50
0
-50
-100
-200
-150
-100
-50
0
50
100
150
200