

areaDetector: A module for EPICS area detector support

Slides (2018): Mark Rivers

GeoSoilEnviroCARS, Advanced Photon Source

University of Chicago

EPICS meeting, ITER, June 2019

Presenter: Ulrik Kofoed Pedersen, Head of Beamline Controls

Demo: Gary Yendell

Diamond Light Source

areaDetector Talk Outline

- Motivation & goals for areaDetector module
- Overview of architecture
- Drivers for detectors & cameras
- Plugins for real-time processing
- Viewers and other clients
- Demo with simDetector

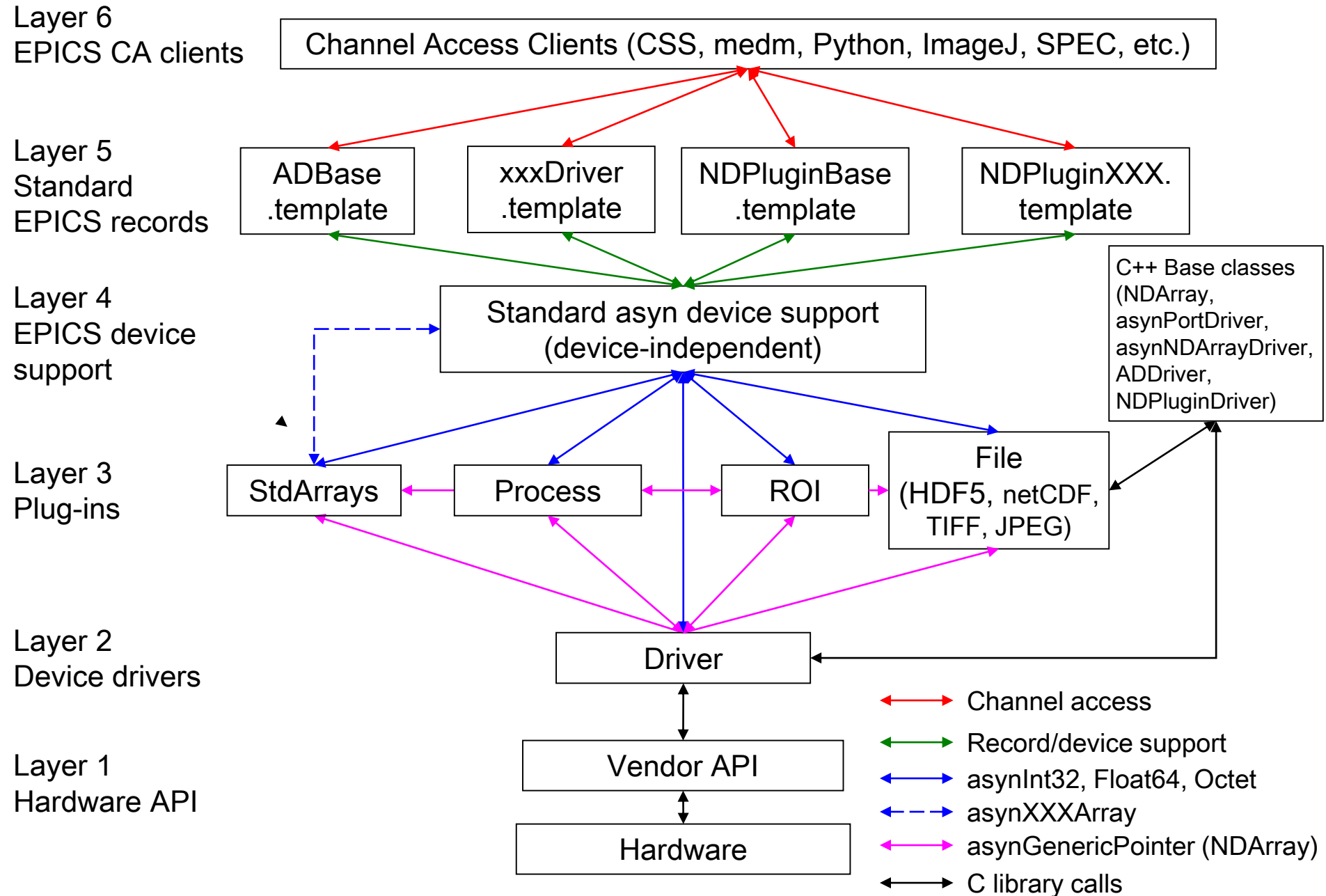
areaDetector - Goals

- Drivers for many detectors popular at synchrotron beamlines
 - Handle detectors ranging from >500 frames/second to <1 frame/second
- Basic parameters for all detectors
 - E.g. exposure time, start acquisition, etc.
 - Allows generic clients to be used for many applications
- Easy to implement new detector
 - Single device-driver C++ file to write. EPICS independent.
- Easy to implement detector-specific features
 - Driver understands additional parameters beyond those in the basic set
- Middle-level plug-ins to add capability like regions-of-interest calculation, file saving, etc.
 - Device independent, work with all drivers
 - Below the EPICS layer for highest performance

areaDetector – Data structures

- **NDArray**
 - N-Dimensional array.
 - Everything is done in N-dimensions (up to 10), rather than 2. This is needed even for 2-D detectors to support color.
 - Other types of devices (Xspress3 and xMAP x-ray spectrometers, quad electrometers also use NDArrays and areaDetector plugins.
 - This is what plug-ins callbacks receive from device drivers.
- **NDArrtribute**
 - Each NDArray has a list of associated attributes (metadata) that travel with the array through the processing pipeline. Attributes can come from driver parameters, any EPICS PV, or any user-written function.
 - Can store motor positions, temperature, ring current, amplifier gains, etc. with each frame.
 - Written to disk files for TIFF, netCDF, and HDF5 file formats.
- **NDArrayPool**
 - Allocates NDArray objects from a freelist
 - Plugins access in readonly mode, increment reference count
 - Eliminates need to copy data when sending it to callbacks.

EPICS areaDetector Architecture



Detector Drivers

Currently 33 detector drivers covering a wide variety of detectors.

- Simulation driver
- GigE cameras (Prosilica/AVT, Point Grey/FLIR, any GigEVision camera via aravis library)
- Point Grey USB-3.x cameras
- Dectris Pilatus and Eiger pixel array detectors
- Princeton Instruments and Photometrics detectors and spectrometers
- Andor CCD and CMOS cameras
- Perkin Elmer and Dexela flat panel detectors
- Web cameras and Axis video servers
- Many more (Bruker, Pixirad, Photonic Sciences, etc.)



ADBase.adl – Generic control screen

- Works with any detector
- Normally write custom control for each detector type to hide unimplemented features and expose driver-specific features

ADBase.adl

Area Detector Control - 13SIM1:cam1:

Setup

asyn port SIM1
EPICS name 13SIM1:cam1:
Manufacturer Simulated detector
Model Basic simulator
Connected
Connection
More

Shutter

Shutter mode
Status: Det. EPICS
Open/Close
Delay: Open Close
EPICS shutter setup

Collect

Exposure time 0.010
Acquire period 0.000
Images 10
Images complete 703
Exp./image 1
Image mode Continuous
Trigger mode Internal
Acquire
Detector state
Time remaining 0.000
Image counter 703
Image rate 67.0
Array callbacks Enable

Readout

	X	Y
Sensor size	640	480
Binning	<input type="text" value="1"/>	<input type="text" value="1"/>
Region start	<input type="text" value="0"/>	<input type="text" value="0"/>
Region size	<input type="text" value="640"/>	<input type="text" value="480"/>
Reverse	<input type="button" value="No"/>	<input type="button" value="No"/>
Image size	640	480
Image size (bytes)		307200
Gain	<input type="text" value="1.000"/>	1.000
Data type	<input type="button" value="UInt8"/>	UInt8
Color mode	<input type="button" value="Mono"/>	Mono

File

Driver file I/O

simDetector: Detector-specific screen

- Example
- 1024x1024 pixels
- 16-bit integer images
- 485 frames/s
- ~1GB/s

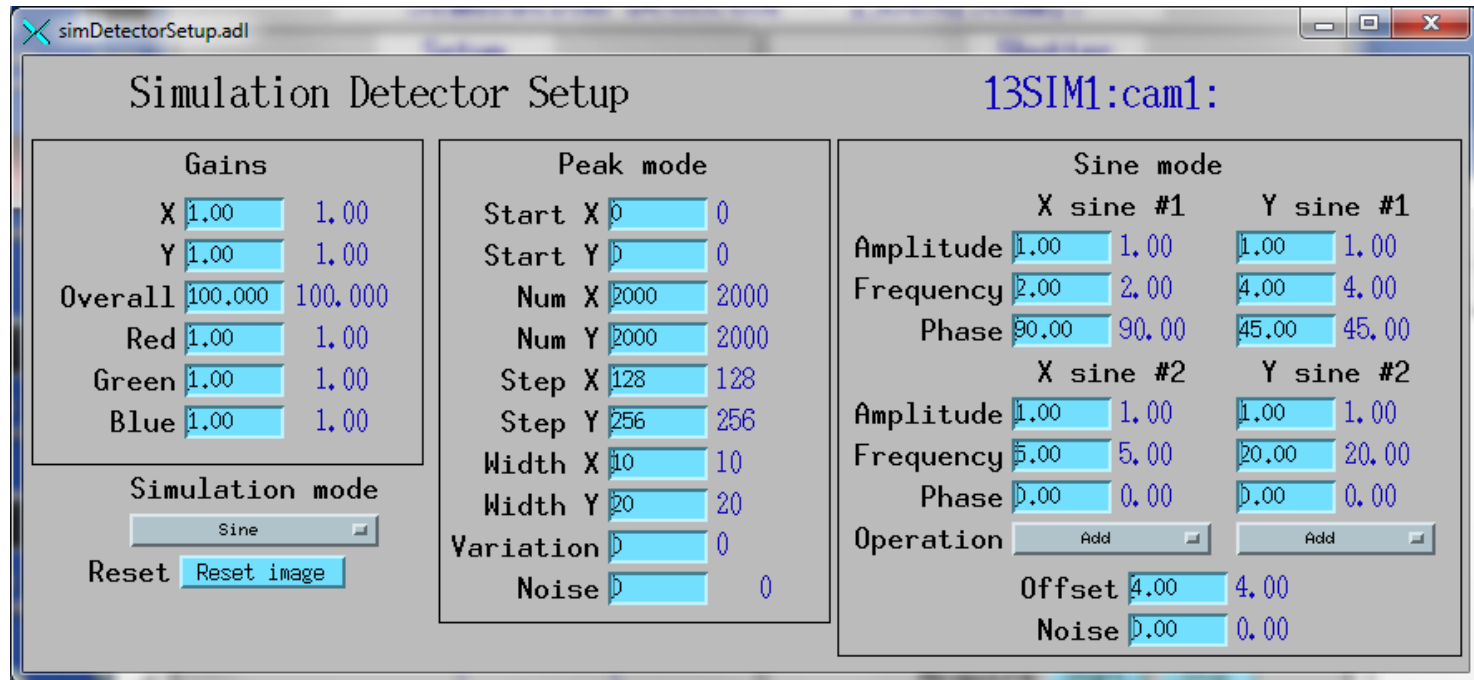
The screenshot displays the 'Simulation Detector - 13SIM1:cam1' window. It is divided into several functional panels:

- Setup:** Displays detector metadata including 'asyn port SIM1', 'EPICS name 13SIM1:cam1', 'Manufacturer Simulated detector', 'Model Basic simulator', 'Serial number No serial number', 'Firmware version No firmware', 'SDK version 2.4.0', 'Driver version 2.4.0', and 'ADCore version 3.0.0'. It also shows a 'Connected' status and buttons for 'Connect', 'Disconnect', and 'Debugging'.
- Shutter:** Controls the detector's shutter. It shows 'Shutter mode' set to 'None', 'Status: Det. Closed' and 'EPICS Closed', and buttons for 'Open/Close'. Delay settings for 'Open' and 'Close' are both set to 0.000. An 'EPICS shutter setup' button is also present.
- Collect:** Manages data acquisition. It shows 'Exposure time' (0.001), 'Acquire period' (0.002), and '# Images' (20). It reports '# Images complete' as 8774. 'Image mode' is set to 'Continuous' and 'Trigger mode' to 'Internal'. 'Acquire' buttons for 'Start' and 'Stop' are shown, with the state 'Collecting'. 'Detector state' is 'Waiting', 'Time remaining' is 0.000, 'Image counter' is 8774, and 'Image rate' is 485.00. 'Array callbacks' are 'Enable'.
- Plugins:** A row of buttons for 'All', 'File', 'ROI', 'Stats', and 'Other #1' and 'Other #2'.
- Readout:** A table showing sensor and acquisition parameters for X and Y axes.
- Attributes:** Shows configuration files and macros. 'File' is 'simDetectorAttributesMacros.xml', 'Macros' is 'CAMERA=13SIM1:cam1;, ID=ID13us:', and 'Status' is 'Attributes file OK'. A 'Simulation setup' button is at the bottom.

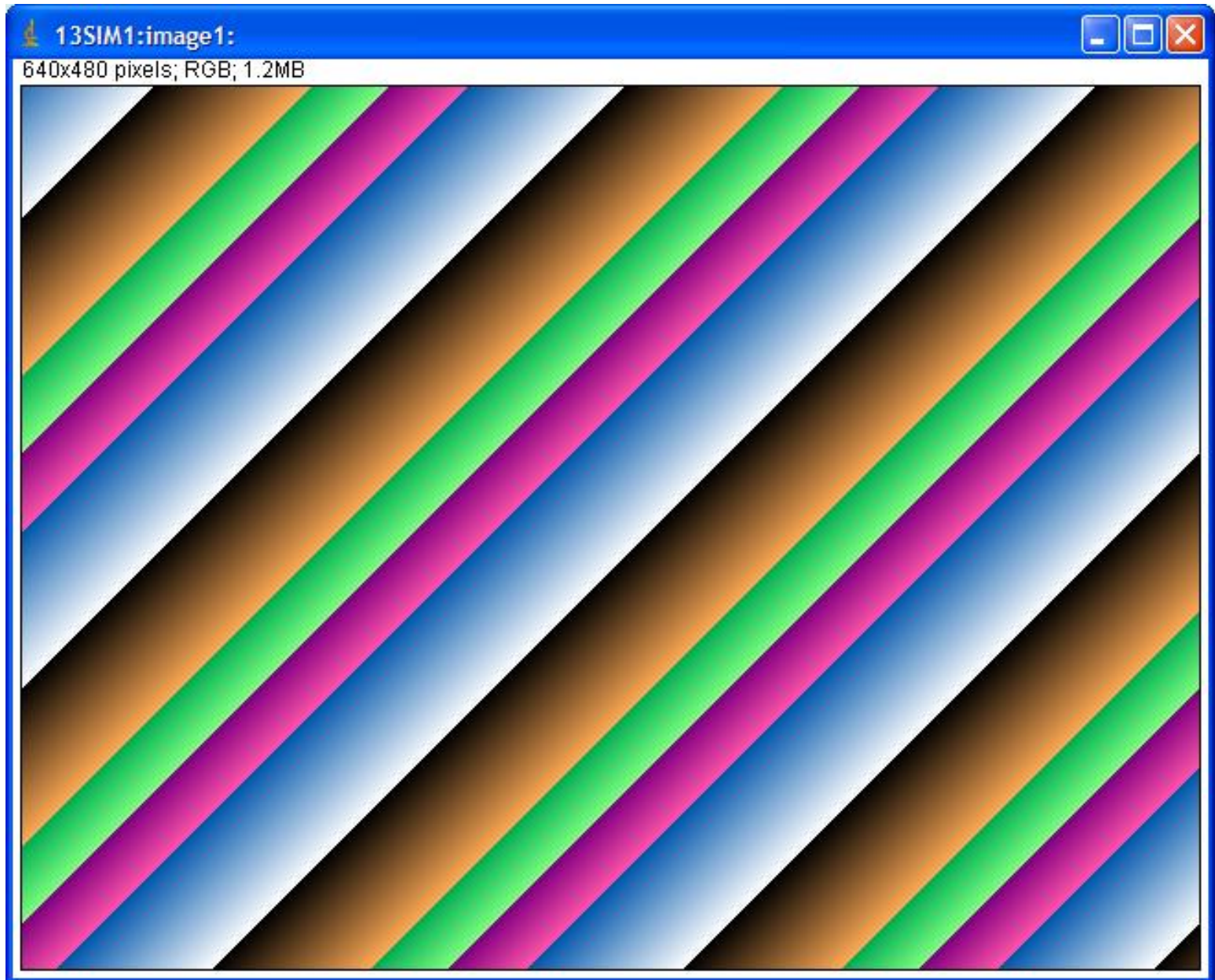
	X	Y
Sensor size	1024	1024
Binning	1	1
Region start	0	0
Region size	1024	1024
Reverse	No	No
Image size	1024	1024
Image size (bytes)	2097152	
Gain	100.000	100.000
Data type	Int16	Int16
Color mode	Mono	Mono

simDetector Driver

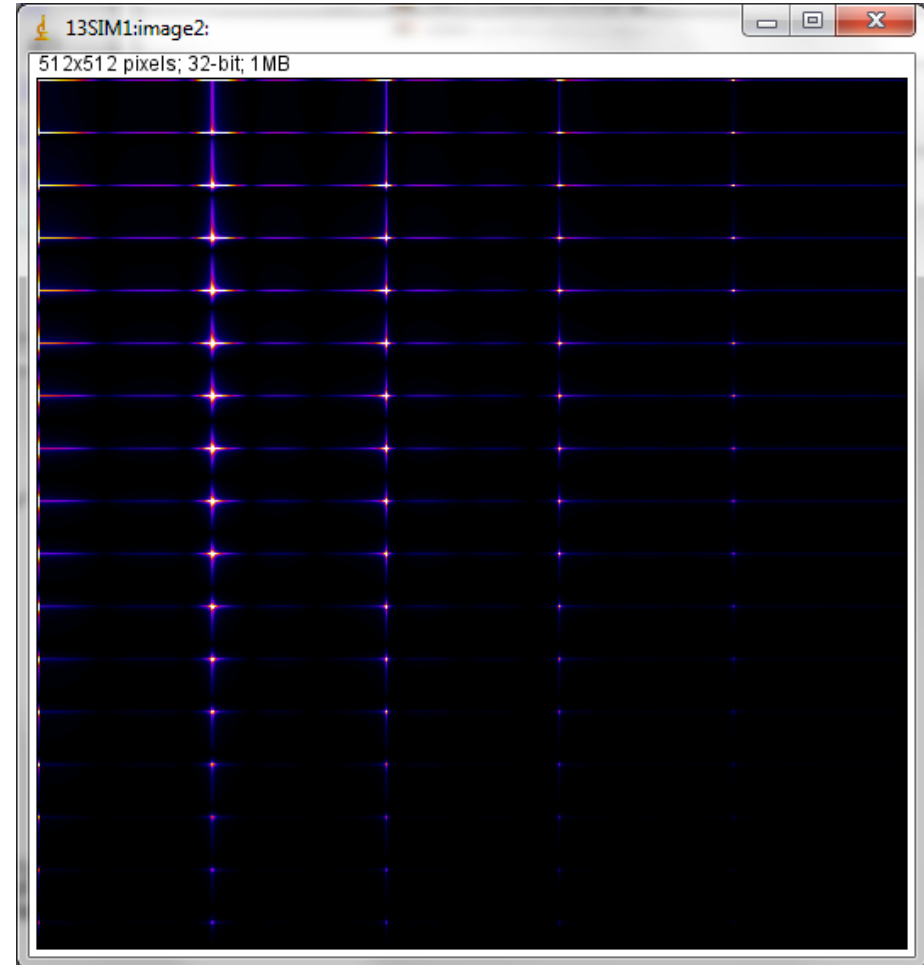
- 3 simulation modes, LinearRamp, Peaks, Sine Waves.
- Independent gains for X, Y, Overall, Red, Green, Blue
- Linear ramp has X and Y linear ramp with array index, each cycle just adds value to each pixel. Very fast.
- Peak mode is array of Gaussian peaks plus noise. Slower.
- Sine mode is 2 sine waves in each of X and Y, summed or multiplied. Slower.



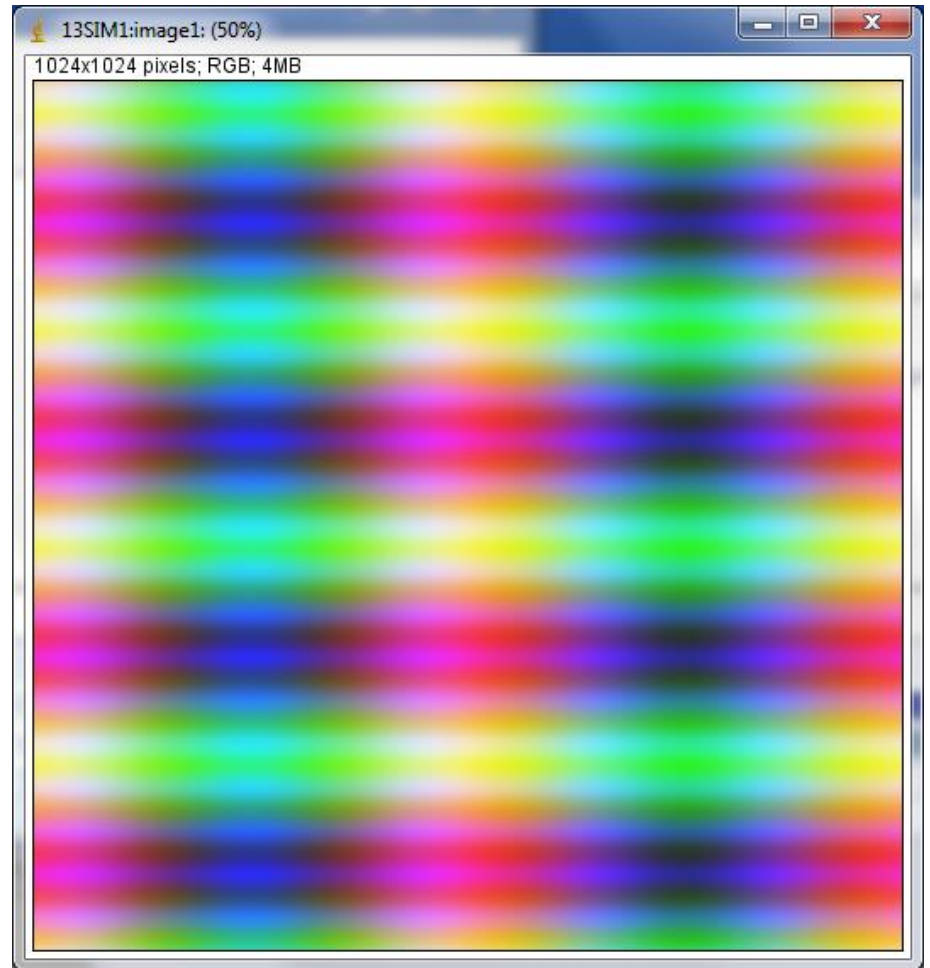
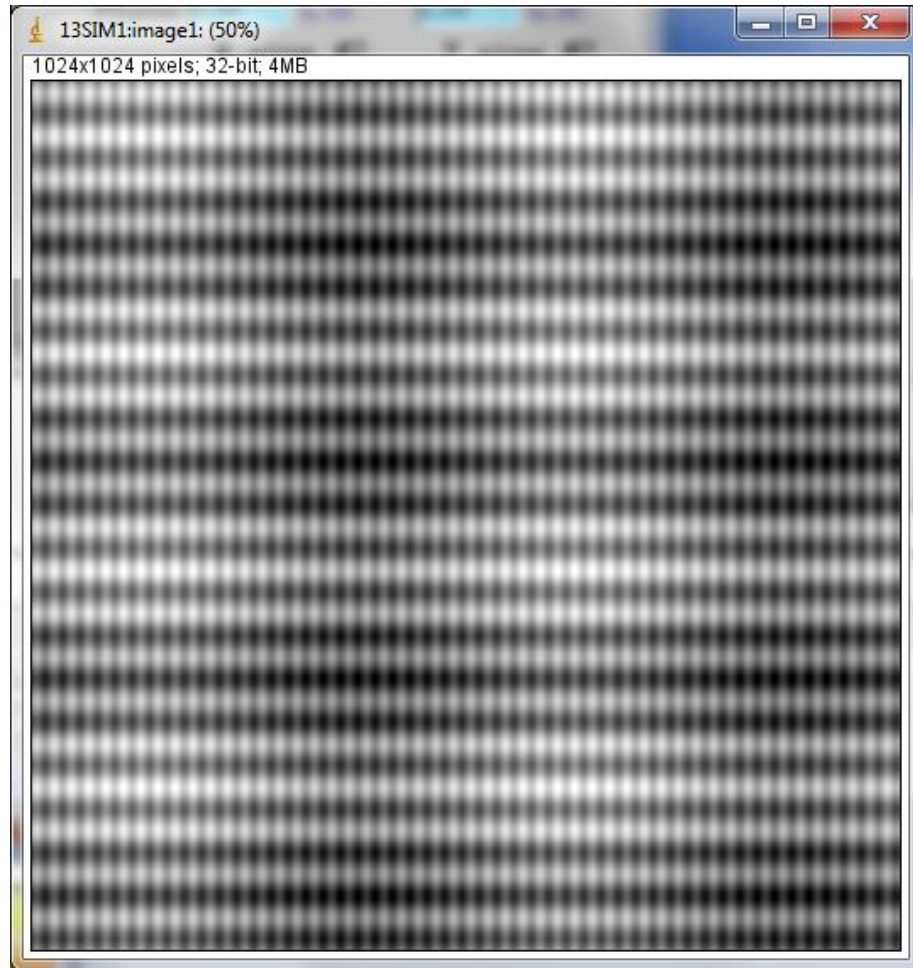
simDetector: Linear Ramp Mode



simDetector Peaks mode with FFT



simDetector: Sine mode



Pilatus specific control screen

pilatusDetector.adl@corvette

Pilatus Detector Control - 13PIL300K:cam1:

Setup

asyn port **PIL**
 EPICS name **13PIL300K:cam1:**
 Manufacturer **Dectris**
 Model **Pilatus**
 Serial number **Unknown**
 Firmware version **Unknown**
 SDK version **7.9.0**
 Driver version **2.5.0**
 ADCore version **3.1.0**

Connected

Connection
 Debugging

Collect

Exposure time 1.000
 Acquire period 1.001
 # Images 10
 Delay time 0.000000
 # Exp./image 1
 Trigger mode Internal

Collecting

Acquire
 Armed **Armed**
 Image counter 3
 Image rate
 Array callbacks Enable

Shutter

Shutter mode
 Status: Det. Closed EPICS Open
 Open/Close
 Delay: Open Close
 EPICS shutter setup

Attributes

File
 Macros
 Status Attributes file OK

Plugins

Stats Other #1 Other #2

Detector

Detector Size
 Threshold (keV)
 Energy (keV)
 En./Thr. apply
 Shaping time/Gain
 Pixel cutoff
 Reset module power
 Reset power delay
 Read file timeout
 Gap fill
 Temperature
 Humidity
 Status rate

File

File Exists: Yes
 File path
 pilatus300k
 File name
 Next file # 9
 Auto increment Yes Ancillary information
 Filename format File format TIFF
 Last filename

Status

Status: Reading image file
 To camserver: Exposure
 From camserver: 15 OK Starting 1.0000000 second background: 2018-01-21T1

Data corrections

Bad pixel file:
 # Bad pixels:
 Flat field file:
 Flat field valid: Min. flat field:

URL Driver

- Driver that can read images from any URL.
- Can be used with Web cameras and Axis video servers.
- Uses GraphicsMagick to read the images, and can thus handle a large number of image formats (JPEG, TIFF, PNG, etc.).

The screenshot shows the 'URLDriver.adl' window titled 'Area Detector Control - 13URL1:cam1:'. It features several control panels:

- Setup:** Includes fields for 'asyn port' (URL1), 'EPICS name' (13URL1:cam1), 'Manufacturer' (URL Driver), and 'Model' (GraphicsMagick). It shows a 'Connected' status and buttons for 'Connect' and 'Disconnect'.
- Shutter:** Shows 'Shutter mode' set to 'None', 'Status: Det.' as 'Closed', and 'EPICS' as 'Closed'. It has 'Open/Close' buttons and 'Delay: Open' and 'Close' fields set to 0.000.
- Collect:** Shows 'Acquire period' (0.100), '# Images' (1), and '# Images complete' (1096). 'Image mode' is set to 'Continuous'. It has 'Start' and 'Stop' buttons for 'Acquire' and a 'Detector state' of 'Acquire'.
- Readout:** Displays image parameters: X=704, Y=480, Image size (bytes)=1013760, Data type=UInt8, and Color mode=RGB1.
- Attributes:** A 'File' field at the bottom.

The screenshot shows the 'URLDriverSetup.adl' window titled 'URL Setup - 13URL1:cam1:'. It contains a table with 10 rows of URL configurations:

Description	URL
1 BMC Hutch (Axis)	http://164.54.160.141/jpg/1/hugesize.jpg
2 BMC Sample (Axis)	http://164.54.160.141/jpg/2/hugesize.jpg
3 The Sun!	images/sun.jpg
4 marCCD	images/marCCD.tif
5 MultiTIFF	images/MultiTIFF.tiff
6 URL6	
7 URL7	
8 URL8	
9 URL9	
10 URL10	

Point Grey drivers

- Driver for all cameras from Point Grey using their FlyCap2 SDK.
- GigE, USB 3.0, and 10 GigE camera
- High performance, low cost
- ADSpinnaker driver which uses their new Spinnaker SDK. Required for their newest camera models like BlackFlyS and Oryx 10-GBit Ethernet.

Model	Interface	Resolution	Price	Measured Speed (frames/s)	Measured Speed (MB/s)
BlackflyS BF3-PGE-13Y3M	1-Gbit Ethernet	1280x1024	\$325	83 frames/s	104 MB/s
Grasshopper 3 GS3-U3-23S6M	USB-3	1920x1200	\$995	156 frames/s	343 MB/s
Oryx ORX-10G-51S5M-C	10-Gbit Ethernet	2448x2048	\$1,875	163 frames/s	779 MB/s

Point Grey GigE Camera BlackFly PGE-20E4C

- e2v EV76C570 CMOS sensor
- Global shutter
- 29 x 29 x 30 mm
- Power Over Ethernet
- 4.5 micron pixels
- 1600 x 1200 pixels, color or mono
- 50 frames/s
- \$525
 - 5X cheaper than comparable Prosilica cameras we bought in the past



Point Grey USB-3.0 Camera Grasshopper3 GS3-U3-23S6M

- 1920 x 1200 global shutter CMOS
- Sony IMX174 1/1.2
- Dynamic range of 73 dB
- Peak QE of 76%
- Read noise of 7e-
- 12-bit or 8-bit data
- Max frame rate of 162 fps
 - ~356 MB/S, >3X faster than GigE
- USB 3.0 interface
- Now used for tomography at 3 APS beamlines, replaced Andor Neo and PCO Edge
- \$995

Mono Sensor 2.3 MP USB VISION



Point Grey 10-Gbit Ethernet Camera

Oryx ORX-10G-51S5C-C

- 2448 x 2048 global shutter CMOS
- Sony IMX250 2/3”
- Dynamic range of 72 dB
- Peak QE of 62%
- Read noise of 2.2e-
- 12-bit, 10-bit, or 8-bit data
- Max frame rate of 162 fps
 - 779 MB/S, >8X faster than GigE
- \$1,875



Model	Resolution	Price	Speed (frames/s)	Speed (MB/s)
ORX-10G-123S6M-C	4096x3000	\$3,950	68 frames/s	797 MB/s
ORX-10G-123S6M-C	4096 x 2160	\$3,650	93 frames/s	785 MB/s
ORX-10G-51S5M-C	2448x2048	\$1,875	163 frames/s	779 MB/s

Point Grey Driver

pointGrey.adl

Point Grey Area Detector Control - 13PG1:cam1:

Setup

asyn port PG1
 EPICS name 13PG1:cam1:
 Manufacturer Point Grey Research
 Model Blackfly BFLY-PGE-20
 Serial Number 13481965
 Firmware Vers. 1.27.3.0
 Software Vers. 2.6.3
 Debugging

Shutter

Shutter mode
 Status: Det. Closed EPICS
 Open/Close
 Delay: Open Close
 EPICS shutter setup

Status

Status rate
 Dropped frames 0
 Corrupt frames 0
 Driver dropped 0
 Transmit failed 0
 Temperature 42.8

Attributes

File

Trigger

Trigger mode
 Trigger source
 Trigger polarity
 Trigger delay
 Skip frames
 Software trigger

Strobe

Strobe source
 Strobe enable
 Strobe polarity
 Strobe delay
 Strobe duration

Bandwidth Control

Max packet size 9000
 Packet size
 Packet size 1440
 GigE packet delay
 Bandwidth (MB/s) 54.9

Readout

	X	Y
Sensor size	1600	1200
Region start	<input type="text" value="0"/>	<input type="text" value="0"/>
Region size	<input type="text" value="1600"/>	<input type="text" value="1200"/>
GigE binning	<input type="text" value="1x1"/>	<input type="text" value="1x1"/>
Image size	1600	1200
Image size (bytes)	1920000	
Gain	<input type="text" value="0.000"/>	<input type="text" value="0.000"/>
Data type	UInt8	
Color mode	Mono	
Video mode	<input type="text" value="Format7"/> <input type="text" value="Format7"/>	
Format7 mode	<input type="text" value="0 (1600x1200)"/>	
Properties	<input type="button" value="⊞"/>	
Frame rate	<input type="button" value="⊞More"/> <input type="text" value="Undefined1"/>	
Pixel format	<input type="button" value="⊞More"/> <input type="text" value="Raw8"/>	
Convert raw	<input type="text" value="None"/> <input type="text" value="None"/>	
Timestamp	<input type="text" value="Camera"/> <input type="text" value="Camera"/>	

Buffers

Buffers max/used	0	1
Buffers alloc/free	2	1
Memory max/used (MB)	0.0	3.7
Buffer & memory polling	<input type="text" value="1 second"/>	

Plugins

All File ROI
 Stats Other

Collect

Exposure time
 Acquire period
 Frame rate
 # Images
 # Images complete 189
 # Exp./image
 Image mode
Collecting
 Acquire
 Detector state Waiting
 Status
 Image counter
 Image rate 30.0
 Array callbacks

Point Grey Driver (Grasshopper3 camera)

pointGreyProperties.adl

13PG1:cam1: Point Grey Properties

Property	Device Unit Control			Absolute Control			
	On/Off	One push	Auto/Manual	Set	Readback	Min	Max
Brightness	On			0	0	0	511
Auto exposure	On	Push	Manual	468	468	1	1023
Sharpness	Off		Manual	0	0	0	4095
White bal. red	Off						
White bal. blue							
Hue	Off						
Saturation	Off						
Gamma	On			512	512	512	4095
Shutter	On	Push	Manual	1181	1242	1	1242
Gain	On	Push	Manual	8	240	0	240
Iris	Off						
Focus	Off						
Temperature	Off						
Trigger mode	On		Manual	5	0	2844	1
Trigger delay	On			0	0	0	4095
Frame rate	Off		Manual	752	407	407	1629
Zoom	Off						
Pan	Off						
Tilt	Off						

Plugins

- Designed to perform real-time processing of data, running in the EPICS IOC (not over EPICS Channel Access)
- Receive NDAarray data over callbacks from drivers or other plugins
- Plug-ins can execute in their own threads (non-blocking) or in callback thread (blocking)
 - If non-blocking then NDAarray data is queued
 - If executing in callback thread, no queuing, but slows driver
- Allows
 - Enabling/disabling
 - Throttling rate (no more than 0.5 seconds, etc)
 - Changing data source for NDAarray callbacks to another driver or plugin
- Plugins can be *sources* of NDAarray callbacks, as well as *consumers*
 - Allows creating a data processing pipeline running at very high speed, each in a different thread, and hence in multiple cores on modern CPUs.

NDPluginDriver medm Screens

NDSStdArrays.adl@corvette.cars.aps.anl.gov

13SIM1:image1:

asyn port	Image1
Plugin type	NDPluginStdArrays
ADCore version	2.6.0
Plugin version	2.6.0
Array port	SIM1 SIM1
Array address	0 0
Enable	Enable Enable
Min. time	0.000 0.000
Callbacks block	No No
Array counter	Reset to 0 22044
Array rate	250.00
Execution time	9.922 msec
Dropped arrays	Reset to 0 19513
# dimensions	2
Array Size	1024 1024 0
Data type	Float32
Color mode	Mono
Unique ID	520875
Time stamp	858383304.571
Attributes file	.
Array callbacks	Disable Disable
Process plugin	Process
More	

NDPluginBaseFull.adl@corvette.cars.aps.anl.gov

13SIM1:image1:

asyn port	Image1
Plugin type	NDPluginStdArrays
ADCore version	2.6.0
Plugin version	2.6.0
Array port	SIM1 SIM1
Array address	0 0
Enable	Enable Enable
Min. time	0.000 0.000
Callbacks block	No No
# threads	1 1
Max # threads	5
Queue size/free	200 0
Sort mode	Unsorted Unsorted
Sort time	0.100 0.100
Sort size/free	0 0
# disordered	Reset to 0 0
Array counter	Reset to 0 42764
Array rate	250.00
Execution time	2.942 msec
Dropped arrays	Reset to 0 38194
Dropped outputs	Reset to 0 0
# dimensions	2
Array Size	1024 1024 0
Data type	Float32
Color mode	Mono
Bayer pattern	RGBB
Unique ID	560283
Time stamp	858383388.180
Attributes file	.
Array callbacks	Disable Disable
Process plugin	Process
asyn record	

Plugins (continued)

- Currently 20 plugins that perform wide variety of operations
- NDPlugInStdArrays
 - Receives arrays (images) from device drivers, converts to standard arrays, e.g. waveform records.
 - This plugin is what EPICS channel access viewers normally talk to.
- NDPluginPVA
 - Converts NDArrays to EPICS V4 NTNDArrays
 - Exports the NtNDArrays over PVAccess with internal V4 server
 - Can be used to send structured data to EPICS V4 clients
 - When used with the PVAccess driver then areaDetector plugins can be run on different machine from the detector driver
- NDPluginROI
 - Performs region-of-interest calculations
 - Select a subregion. Optionally bin, reverse in either direction, convert data type.
 - Divide the array by a scale factor, which is useful for avoiding overflow when binning.
- NDPluginTransform
 - Performs geometric operations (rotate, mirror in X or Y, etc.)

Plugins (continued)

- NDPluginStats
 - Calculates basic statistics on an array (min, max, sigma)
 - Optionally computes centroid position, width and tilt.
 - Optionally Computes X and Y profiles, including average profiles, profiles at the centroid position, and profiles at a user-defined cursor position.
 - Optionally computes the image histogram and entropy
- NDPluginROIStat
 - Multiple ROIs with simple statistics in a single plugin
 - More efficient when many ROIs are needed, e.g. for peaks in a 1-D energy spectrum
 - Min, max, total, net, mean
 - Time-series of each of these statistics

Plugins (continued)

- NDPluginProcess
 - Does arithmetic processing on arrays
 - Background subtraction.
 - Flat field normalization.
 - Offset and scale.
 - Low and high clipping.
 - Recursive filtering in the time domain.
 - Conversion to a different output data type.
- NDPluginOverlay
 - Adds graphic overlays to an image.
 - Can be used to display ROIs, multiple cursors, user-defined boxes, text, etc.
- ffmpegServer
 - MJPEG server that allows viewing images in a Web browser. From DLS.

Plugins (continued)

- **NDPluginAttribute**
 - Extracts NDAttributes from NDArrays and publishes their values as ai records
 - Can collect time-series arrays of the attribute values
- **NDPluginCircularBuff**
 - Buffers NDArrays in a circular buffer
 - Computes a trigger expression using up to 2 NDAttribute values
 - When trigger condition is met then outputs NDArrays
 - User-specified number of pre-trigger and post-trigger arrays to output
- **NDPluginTimeSeries**
 - Accepts 1-D NDArrays[NumSignals] or 2-D [NumSignals,NewTimePoints] and appends to time-series buffer
 - Operates in fixed length (stop when full) or circular buffer modes
 - Optional time-averaging of input data

Plugins (continued)

- NDPluginFFT
 - Computes FFT of 1-D or 2-D NDArrays
 - Exports NDArrays containing the absolute value (power spectrum) of the FFT
 - Exports 1-D arrays of the FFT real, imaginary, absolute values, and time and frequency data.
- NDPluginColorConvert
 - Convert from one color model to another (Mono, RGB1 (pixel), RGB2 (row) or RGB3 (planar) interleave)
 - Bayer conversion removed from this plugin, now part of Prosilica and Point Grey drivers.

commonPlugins.adl All plugins at a glance

Plugin name	Plugin type	Port	Enable	Blocking	Dropped	Free	Rate		
Image1	NDPluginStdArrays	SIM1	Enable <input type="checkbox"/>	Enable	No <input type="checkbox"/>	0	3	89.0	More
PROC1	NDPluginProcess	SIM1	Enable <input type="checkbox"/>	Enable	No <input type="checkbox"/>	0	20	89.0	More
TRANS1	NDPluginTransform	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
CC1	NDPluginColorConvert	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
CC2	NDPluginColorConvert	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
OVER1	NDPluginOverlay	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
ROI1	NDPluginROI	SIM1	Enable <input type="checkbox"/>	Enable	No <input type="checkbox"/>	0	19	89.0	More
ROI2	NDPluginROI	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
ROI3	NDPluginROI	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
ROI4	NDPluginROI	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
STATS1	NDPluginStats	ROI1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
STATS2	NDPluginStats	ROI2	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
STATS3	NDPluginStats	ROI3	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
STATS4	NDPluginStats	ROI4	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
STATS5	NDPluginStats	SIM1	Enable <input type="checkbox"/>	Enable	No <input type="checkbox"/>	885	0	21.0	More
FileNetCDF1	NDFileNetCDF	SIM1	Enable <input type="checkbox"/>	Enable	No <input type="checkbox"/>	0	20	0.0	More
FileTIFF1	NDFileTIFF	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
FileJPEG1	NDFileJPEG	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
FileNexus1	NDPluginFile	SIM1	Enable <input type="checkbox"/>	Enable	No <input type="checkbox"/>	0	20	0.0	More
FileMagick1	NDFileMagick	SIM1	Disable <input type="checkbox"/>	Disable	No <input type="checkbox"/>	0	20	0.0	More
FileHDF1	NDFileHDF5 ver1.8.7	SIM1	Enable <input type="checkbox"/>	Enable	No <input type="checkbox"/>	0	20	0.0	More

ROI plugin

NDROI.adl

13SIM1:ROI1:

asyn port	ROI1		
Plugin type	NDPluginROI		
Array port	SIM1	SIM1	
Array address	0	0	
Enable	Enable	Enable	
Min. time	0.000	0.000	
Callbacks block	No	No	
Queue size/free	20	20	
Array counter	0	834794	
Array rate	48.0		
Dropped arrays	0	0	
# dimensions	2		
Array Size	1024	1024	0
Data type	Int8		
Color mode	Mono		
Bayer pattern	RGGB		
Unique ID	834794		
Time stamp	717905394.895		
Attributes file	i		
asyn record			

Definition

Name: Upper left

Data type: Automatic

Enable scaling: Enable

Scale divisor: 2

	X	Y	Z
Input Size	1024	1024	0
Enable	Enable	Enable	Disable
Binning	1	1	1
ROI start	0	0	0
ROI size	512	512	1
Reverse	No	No	No
ROI Size	512	512	0

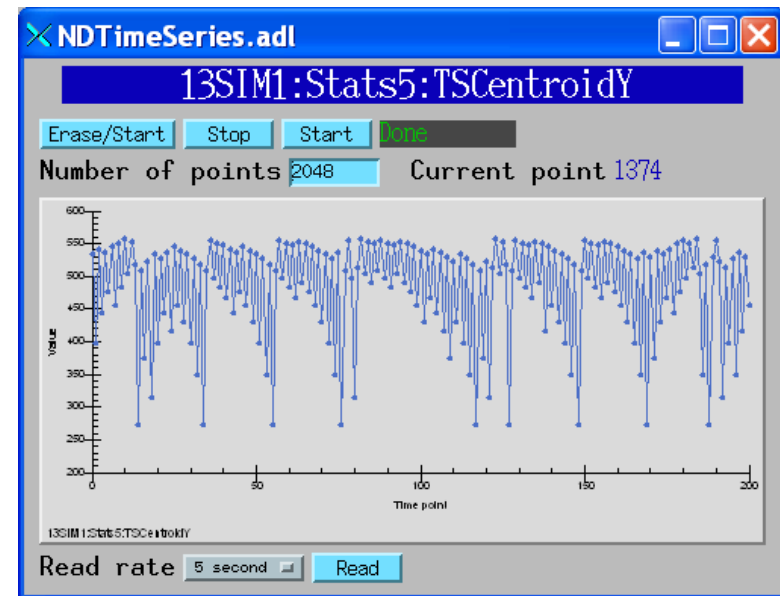
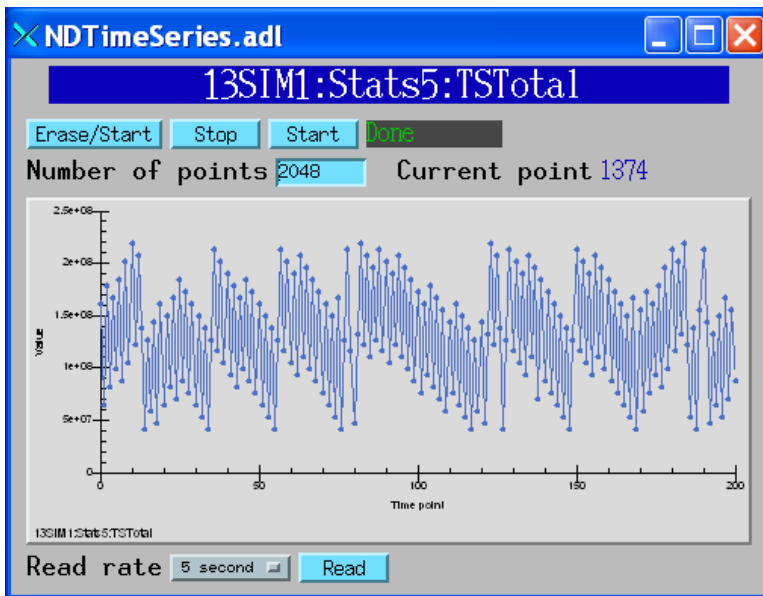
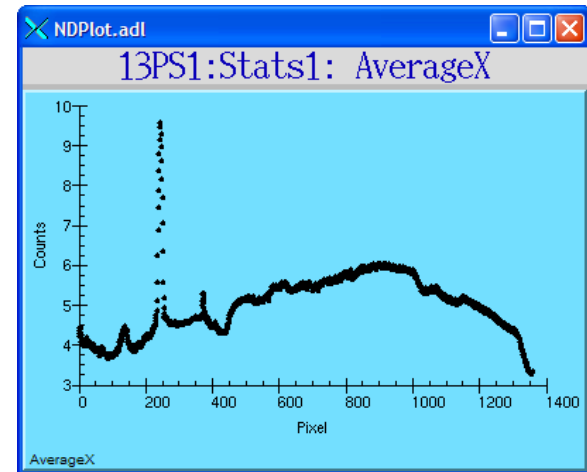
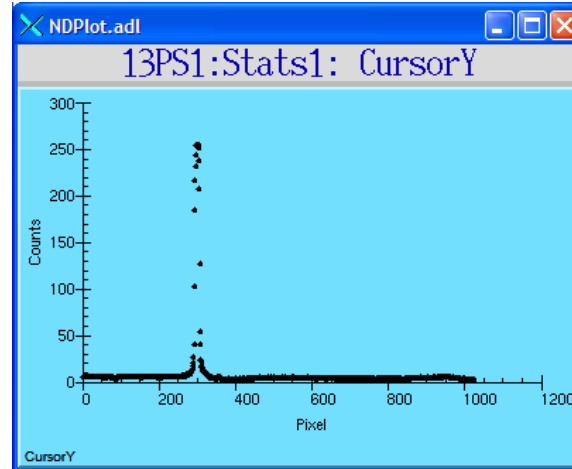
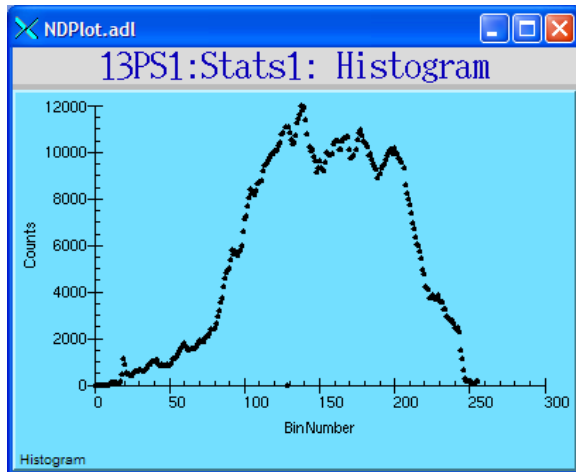
Statistics plugin

NDStats.adl

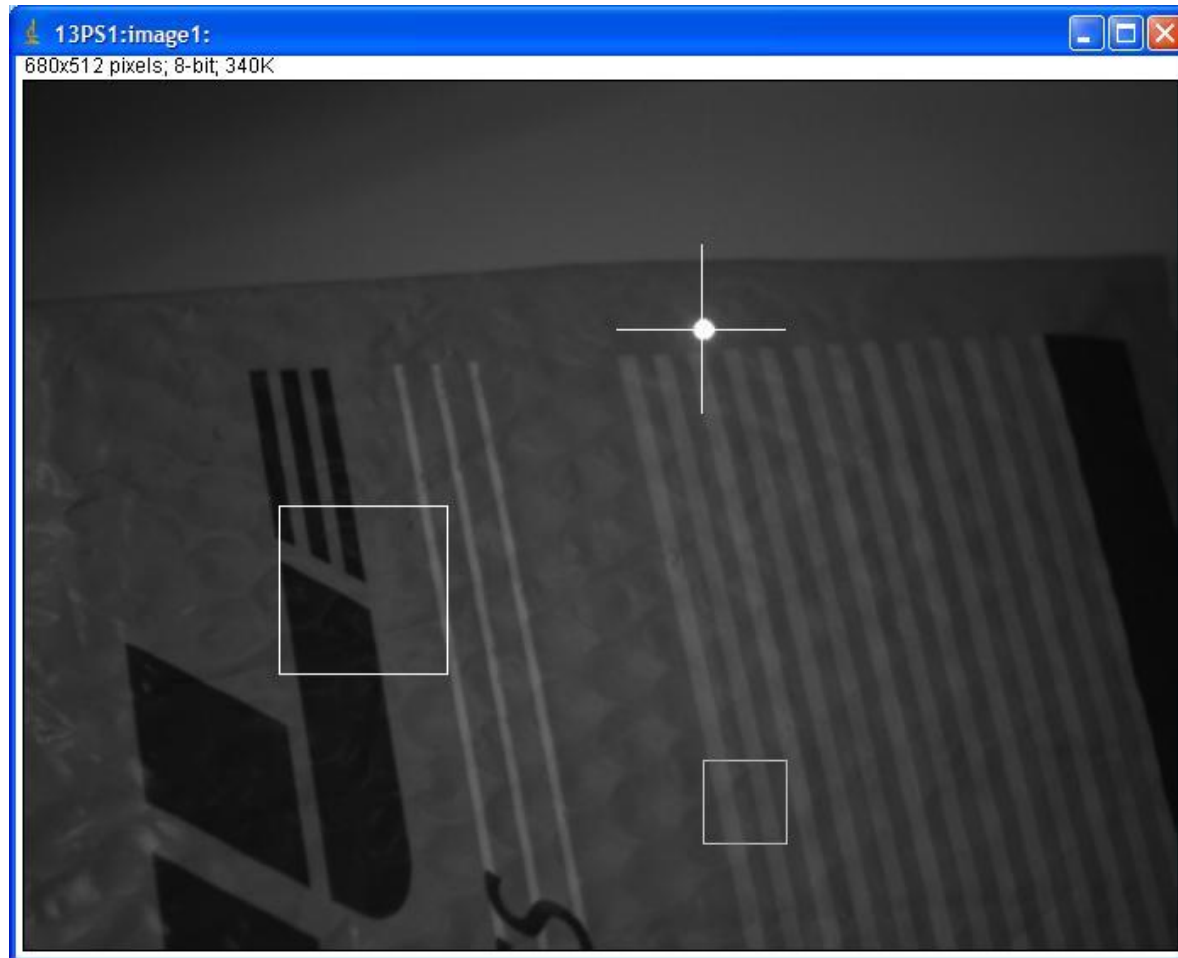
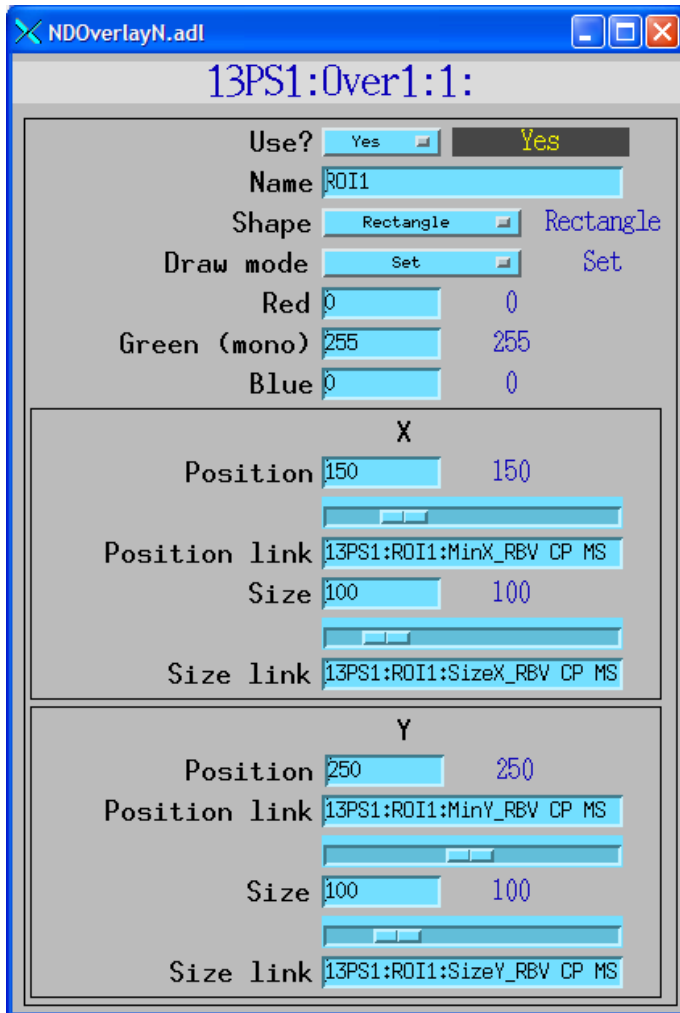
13SIM1:Stats5:

<p>asyn port STATS5</p> <p>Plugin type NDPluginStats</p> <p>Array port SIM1 SIM1</p> <p>Array address 0 0</p> <p>Enable <input type="checkbox"/> Enable Enable</p> <p>Min. time 0.000 0.000</p> <p>Callbacks block <input type="checkbox"/> No No</p> <p>Queue size/free 20 12</p> <p>Array counter 0 4056</p> <p>Array rate 25.0</p> <p>Dropped arrays 0 395</p> <p># dimensions 2</p> <p>Array Size 1024 1024 0</p> <p>Data type Int8</p> <p>Color mode Mono</p> <p>Bayer pattern RGGB</p> <p>Unique ID 4451</p> <p>Time stamp 717886862.801</p> <p>Attributes file .</p> <p>asyn record <input type="checkbox"/></p>	<p>Statistics</p> <p>Compute statistics <input type="checkbox"/> Yes Yes</p> <p>Background width 1 1</p> <p>Minimum 0 Maximum 6</p> <p>Min. X 0 Max. X 200</p> <p>Min. Y 0 Max. Y 148</p> <p>Total 622 Net 622</p> <p>Mean 0 Sigma 0.0</p> <p>Time series plots <input type="checkbox"/></p>	<p>Profiles</p> <p>Compute profiles <input type="checkbox"/> Yes Yes</p> <p>Size X 1024 Y 1024</p> <p>256 256</p> <p>Cursor X <input type="checkbox"/></p> <p>256 256</p> <p>Cursor Y <input type="checkbox"/></p> <p>Plot <input type="checkbox"/></p>
	<p>Centroid</p> <p>Compute centroid <input type="checkbox"/> Yes Yes</p> <p>Centroid threshold 1 1</p> <p>Centroid X 200.0 Y 150.1</p> <p>Sigma X 3.9 Y 3.9</p> <p>Sigma XY -0.024</p> <p>Time series plots <input type="checkbox"/></p>	<p>Histogram</p> <p>Compute histogram? <input type="checkbox"/> Yes Yes</p> <p>Size 256 256</p> <p>Minimum 0 0</p> <p>Maximum 255 255</p> <p>Entropy -13.860</p> <p>Plot <input type="checkbox"/></p>
	<p>Time Series</p> <p>Erase/Start <input type="checkbox"/> Stop <input type="checkbox"/> Start <input type="checkbox"/> Acquiring</p> <p>Number of points 2048</p> <p>Current point 82</p> <p>Read rate <input type="checkbox"/> 5 second <input type="checkbox"/> Read</p>	

Statistics plugin (continued)



Overlay plugin



Centroid of laser pointer calculated by statistics plugin
Cursor overlay X, Y position linked to centroid

Processing plugin

NDProcess.adl

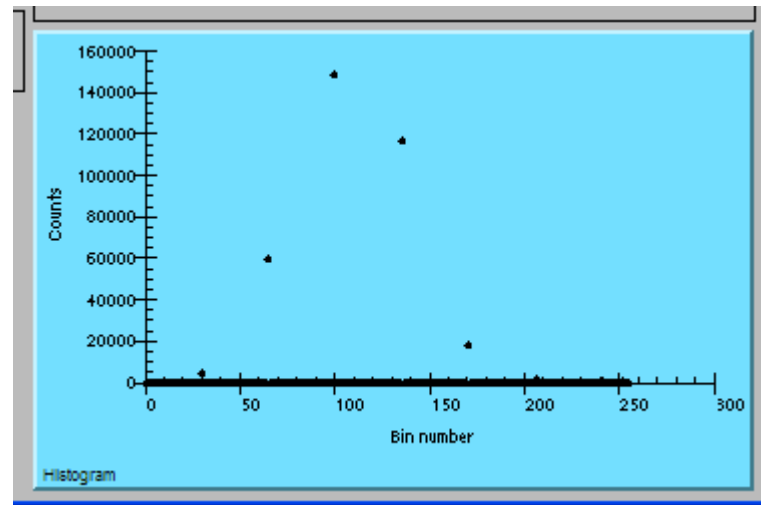
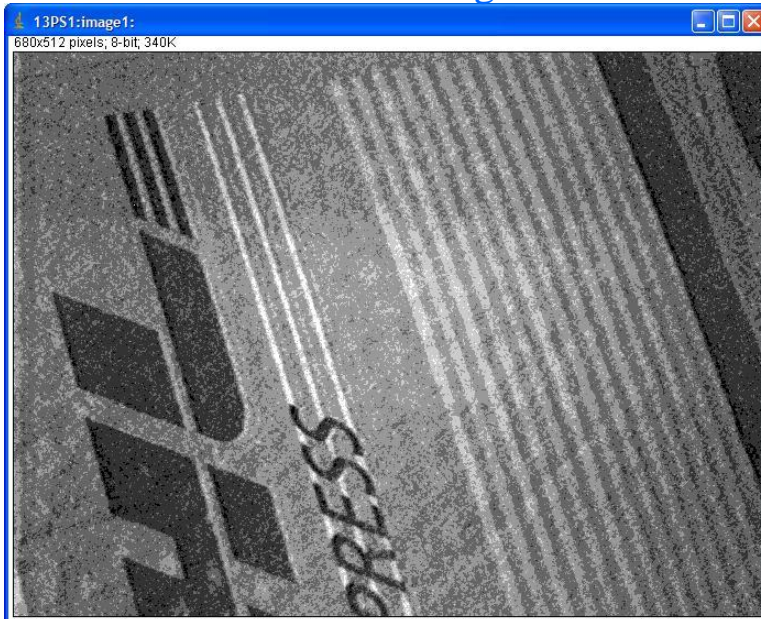
13SIM1:Proc1:

asyn port PROC1 Plugin type NDPluginProcess Array port SIM1 SIM1 Array address 0 0 Enable Enable Enable Min. time 0.000 0.000 Callbacks block No No Queue size/free 20 20 Array counter 0 11572 Array rate 47.0 Dropped arrays 0 0 # dimensions 2 Array Size 1024 1024 0 Data type Int8 Color mode Mono Bayer pattern RGGB Unique ID 12032 Time stamp 717887092.888 Attributes file asyn record	<h3>Background subtraction</h3> Save background Save Invalid Enable background Disable Disable <h3>Flat field normalization</h3> Save flat field Save Invalid Enable flat field Disable Disable Scale flat field 255 255 <h3>Scale and Offset</h3> Enable scale/off. Disable Enable Auto scale/off. Auto calc Scale value 0.10 42.50 Offset value 0.00 0.00 <h3>Low/High Clipping</h3> Enable low clip Disable Enable Low clip value 100 0 Enable high clip Disable Enable High clip value 150 255 <h3>Output data type</h3> Data type Automatic Automatic	<h3>Recursive filter</h3> Enable filter Disable Disable N filter 100 100 N filtered 0 Filter type RecursiveAve Reset filter Reset Auto reset filter Yes Filter callbacks Every array OOffset 0.00 0.00 OScale 1.00 1.00 OC1 1.00 1.00 OC2 -1.00 -1.00 OC3 0.00 0.00 OC4 1.00 1.00 FOffset 0.00 0.00 FScale 1.00 1.00 FC1 1.00 1.00 FC2 -1.00 -1.00 FC3 0.00 0.00 FC4 1.00 1.00 ROffset 0.00 0.00 RC1 0.00 0.00 RC2 1.00 1.00 $O[n] = Ooffset + OScale * ((OC1+OC2/N) * F[n-1] + (OC3+OC4/N) * I[n])$ $F[n] = Foffset + FScale * ((FC1+FC2/N) * F[n-1] + (FC3+FC4/N) * I[n])$ <p>On filter reset: $F[0] = Roffset + RC1 * F[n] + RC2 * I[0]$ I = Input array in callback F = Stored filter (double precision) N = value of NumFiltered O = Output array passed to clients</p>
--	---	---

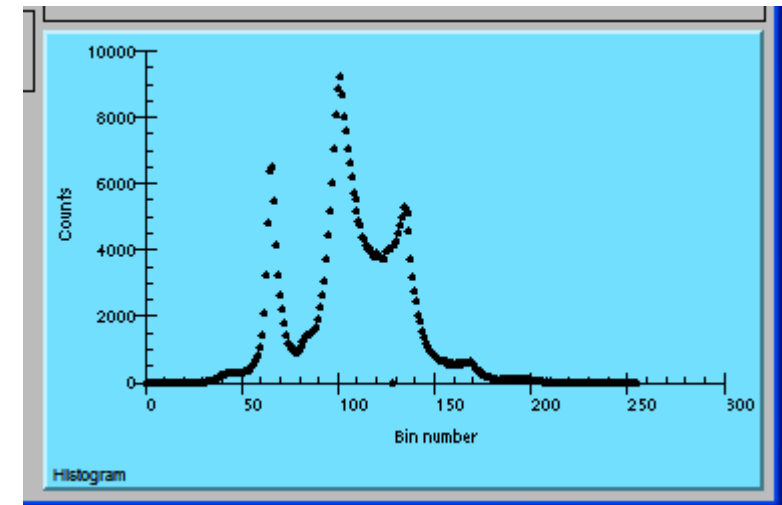
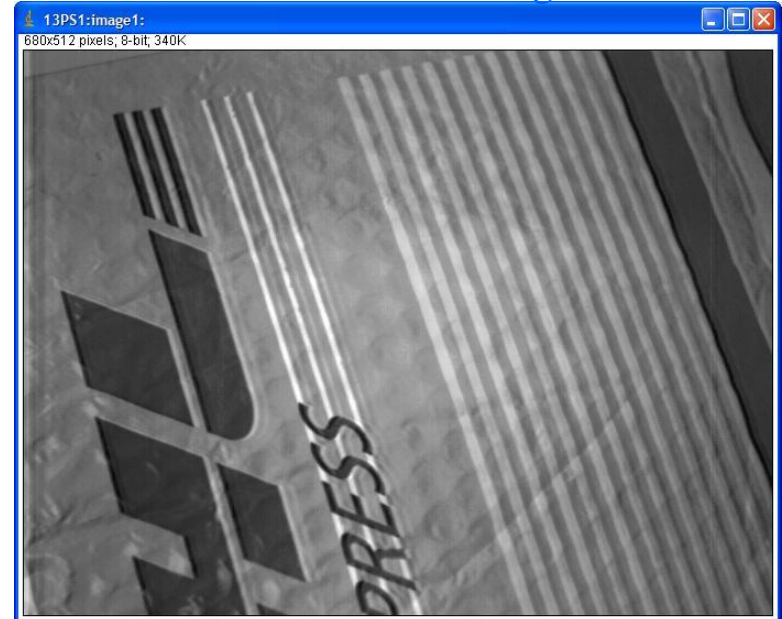
Processing plugin

30 microsec exposure time

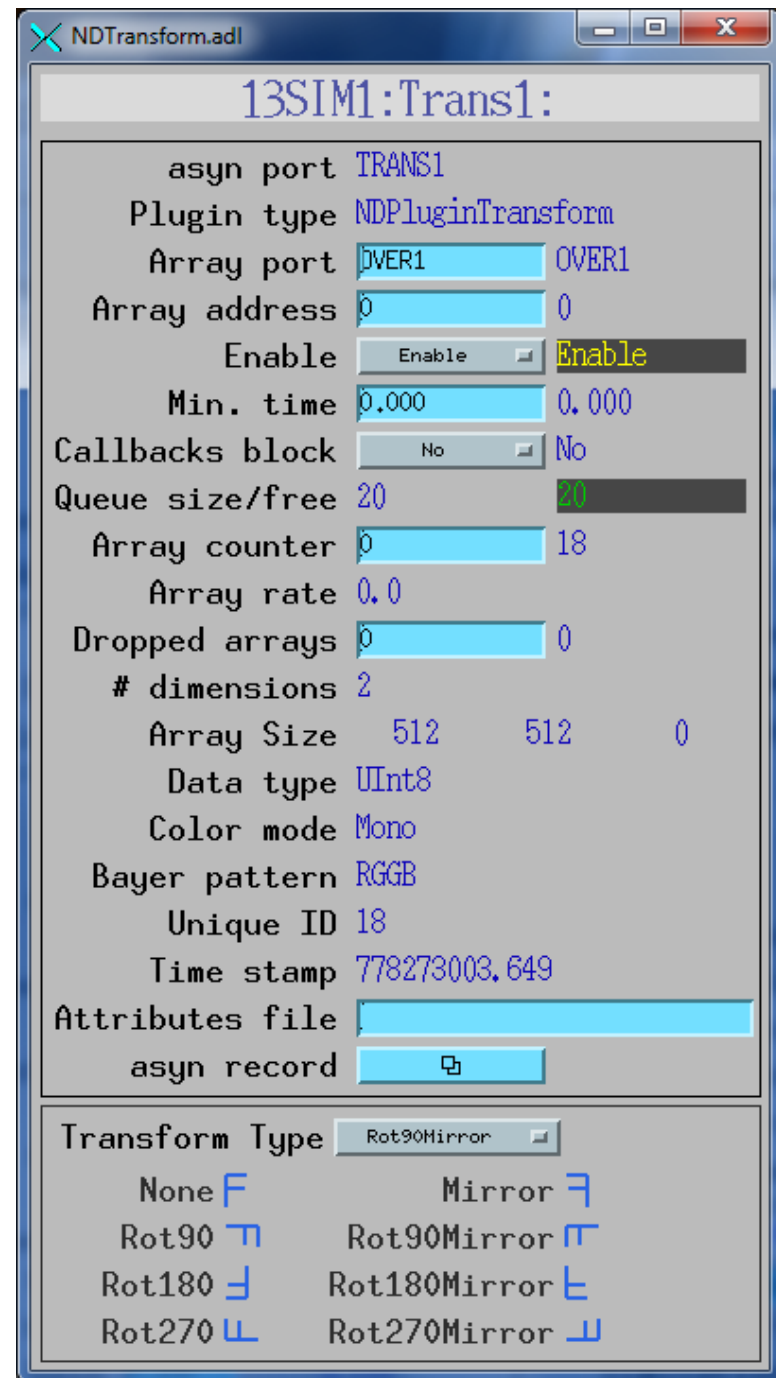
No filtering



N=100 recursive average filter

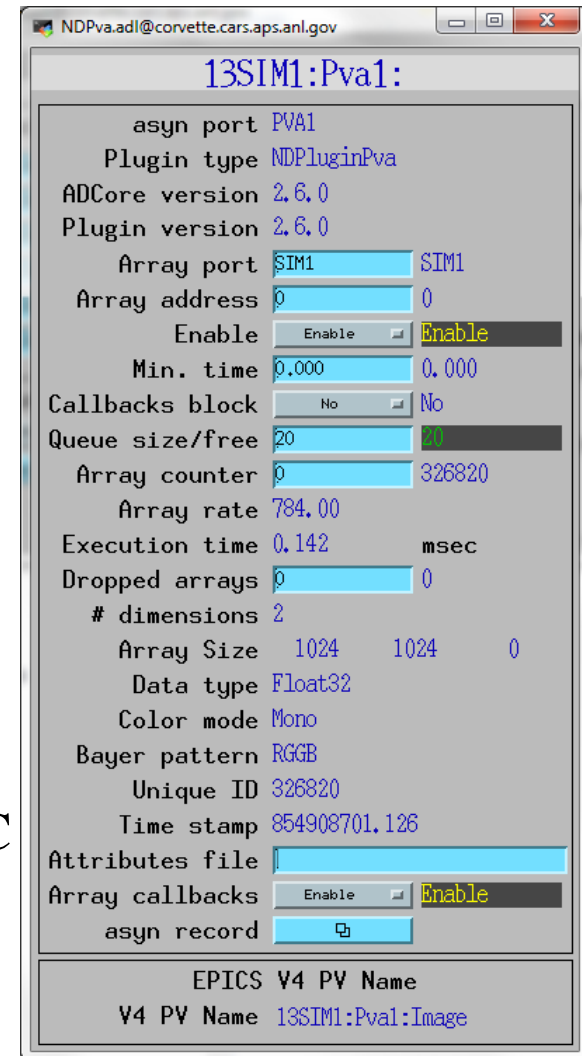


Transform plugin



NDPluginPva (EPICS V4/7)

- New plugin that converts NDArrays into the EPICS v4 normative type NTNDArray
- Embedded EPICSv4 server serves the new NTNDArray structure as an EPICSv4 PV
- High performance, ~3.2GB/s shown here
- Can be received by any EPICS v4 client
 - Java, Python, C++ versions of pvAccess
 - CSS has a widget that can display NTNDArrays
 - New ImageJ plugin
 - Can include an NTNDArray receiver in another IOC
- From Bruno Martins



pvAccess Driver (EPICS V4)

- Logical inverse of NDPluginPva
- Receives NTNDArrays over the network, converts to NDArrays and calls plugins
- Can be used to run areaDetector IOC and plugins on another machine or in another process
- High performance:
 - ~1.2 GB/s shown here with interprocess communication
 - Saturating 10 Gb Ethernet links has been demonstrated
- From Bruno Martins

The screenshot shows the 'Area Detector Control - 13PVA1:cam1' window. It is divided into several sections:

- Setup:** asyn port PVA, EPICS name 13PVA1:cam1, Manufacturer PVAcess driver, Model Basic PVAcess driver, Serial number No serial number, Firmware version No firmware, SDK version 5.0.0, Driver version 1.1.0, ADCore version 2.6.0. Connection is 'Connected' with 'Connect' and 'Disconnect' buttons. Debugging is checked.
- Collect:** # Images 100, # Images complete 105801, Image mode Continuous. Acquire buttons 'Start' and 'Stop' are present. Detector state is 'Idle'. Status shows Image counter 105801 and Image rate 303.00. Array callbacks are 'Enable'.
- Buffers:** Buffers max/used 20 1, Buffers alloc/free 6 5, Memory max/used (MB) 0.0 24.0, Buffer & memory polling 1 second.
- Plugins:** All, File, ROI, Stats, Other buttons.
- Readout:** A table showing sensor parameters:

	X	Y
Sensor size	1024	1024
Binning	1	1
Region start	0	0
Region size	1024	1024
Reverse	No	No
Image size	1024	1024
Image size (bytes)		4194304
Data type	Float32	
Color mode	Mono	
- Attributes:** File input field.
- pvaDriver:** Overrun counter 42308, PV Name 13SIM1:Pva1:Image, PV Connection Up.

Demo

- ADSimDetector
- ImageJ viewer
- Plugins
 - Transform
 - ROI
 - Proc
 - Stats

Plugins: NDPluginFile

- Saves NDArrays to disk
- 3 modes:
 - Single array per disk file
 - Capture N arrays in memory, write to disk either multiple files or as a single large file (for file formats that support this.)
 - Stream arrays to a single large disk file
- For file formats that support it, stores not just NDArray data but also NDAttributes

Plugins: NDPluginFile

- File formats currently supported
 - NDFileTIFF
 - Supports any NDArray data type
 - Stores NDAttributes as ASCII user tags
 - NDFileJPEG
 - With compression control
 - NDFileNetCDF
 - Popular self-describing binary format, supported by Unidata at UCAR
 - NDFileHDF5
 - Writes HDF5 files with the native HDF5 API, unlike the NeXus plugin which uses the NeXus API. Supports 3 types of compression.
 - Supports using an XML file to define the layout and placement of NDArrays and NDAttributes in the HDF5 file
 - Support Single Writer Multiple Reader (SWMR). Only supported on local file systems, GPFS, and Lustre (not NFS or SMB)

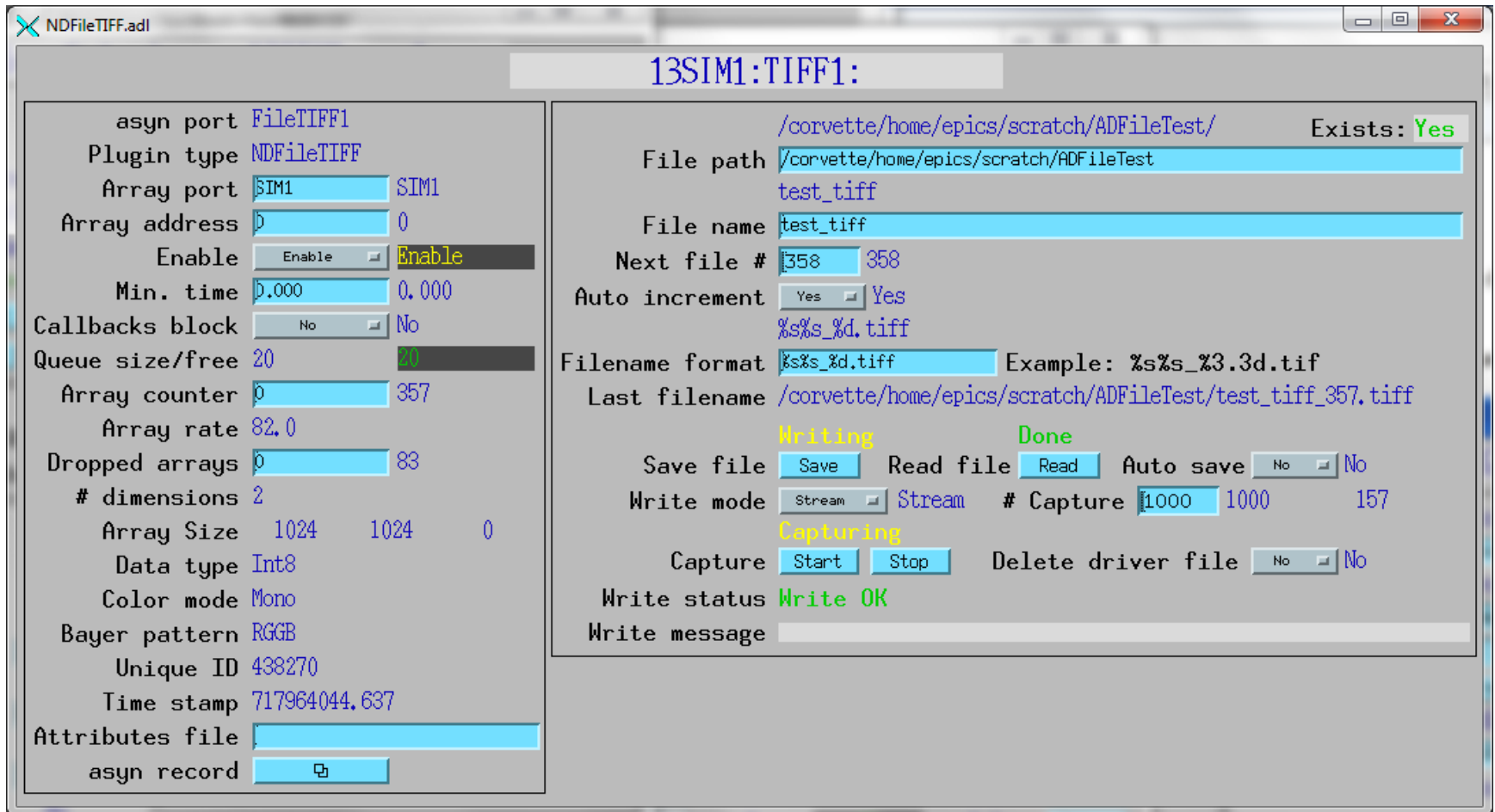
Plugins: NDPluginFile

- File formats currently supported
 - NDFileNeXus
 - Standard file format for neutron and x-ray communities, based on HDF5, which is another popular self-describing binary format; richer than netCDF
 - May be deprecated in a future release since NeXus files can now be produced with the NDFileHDF5 plugin using an appropriate XML layout file
 - NDFileMagick
 - Uses GraphicsMagick to write files, and can write in dozens of file formats, including JPEG, TIFF, PNG, PDF, etc.
 - NDFileNull
 - Used only to delete original driver files when no other file plugin is running

File saving with driver

- In addition to file saving plugins, many vendor libraries also support saving files (e.g. marCCD, mar345, Pilatus, etc.) and this is supported at the driver level.
- File saving plugin can be used instead of or in addition to vendor file saving
 - Can add additional metadata vendor does not support
 - Could write JPEGs for Web display every minute, etc.

NDPluginFile display: TIFF



Example: saving 82 frames/second of 1024x1024 video to TIFF files, a few dropped frames.

NDFileHDF5

NDFileHDF5.adl

13SIM1:HDF1:

asyn port	FileHDF1
Plugin type	NDFileHDF5 ver1.8.7
Array port	SIM1 SIM1
Array address	0 0
Enable	<input type="checkbox"/> Enable <input checked="" type="checkbox"/> Enable
Min. time	0.000 0.000
Callbacks block	<input type="checkbox"/> No <input checked="" type="checkbox"/> No
Queue size/free	20 0
Array counter	0 611
Array rate	10.0
Dropped arrays	0 0
# dimensions	2
Array Size	1024 1024 0
Data type	UInt8
Color mode	Mono
Bayer pattern	RGGB
Unique ID	3461
Time stamp	779563295.068
Attributes file	
asyn record	<input type="checkbox"/>

File path	/home/epics/scratch/	Exists: Yes
File name	test_mono	
Next file #	220 220	
Auto increment	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Yes	
Filename format	%%s_%.3d.h5	Example: %%s_%.3d.h5
Last filename	/home/epics/scratch/test_mono_219.h5	
Lazy open	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Yes	
Save file	<input type="button" value="Save"/> Read file <input type="button" value="Read"/> Auto save <input type="checkbox"/> No <input checked="" type="checkbox"/> No	
Write mode	<input type="checkbox"/> Stream <input checked="" type="checkbox"/> Stream # Capture 100 100 28	
Capture	<input type="button" value="Start"/> <input type="button" value="Stop"/> Delete driver file <input type="checkbox"/> No <input checked="" type="checkbox"/> No	
Write status	Write OK	
Write message		

Compression	<input type="checkbox"/> None <input checked="" type="checkbox"/> None	Extra dimensions
# data bits	8	# (0-2) 0 0
Data bits offset	0	Size N 1 1
SZip # pixels	16 16	Name N frame number n
Zlib level	6	Size X 1 1
Store performance	<input type="checkbox"/> No <input checked="" type="checkbox"/> Yes	Name X scan dimension X
Store attributes	<input type="checkbox"/> No <input checked="" type="checkbox"/> Yes	Size Y 1 1
Run time	9.913	Name Y scan dimension Y
I/O speed	80.7	

Exists: Yes

hdf5_layout_demo.xml

XML File name hdf5_layout_demo.xml

NDFileHDF5

XML file to define file layout

```
<xml>
  <group name="entry">
    <attribute name="NX_class" source="constant" value="NXentry" type="string"></attribute>
    <group name="instrument">
      <attribute name="NX_class" source="constant" value="NXinstrument" type="string"></attribute>
      <group name="detector">
        <attribute name="NX_class" source="constant" value="NXdetector" type="string"></attribute>
        <dataset name="data" source="detector" det_default="true">
          <attribute name="NX_class" source="constant" value="SDS" type="string"></attribute>
          <attribute name="signal" source="constant" value="1" type="int"></attribute>
          <attribute name="target" source="constant" value="/entry/instrument/detector/data"
            type="string"></attribute>
        </dataset>
        <group name="NDAttributes">
          <attribute name="NX_class" source="constant" value="NXcollection" type="string"></attribute>
          <dataset name="ColorMode" source="ndattribute" ndattribute="ColorMode">
            </dataset>
          </group>
          <!-- end group NDAttribute -->
        </group>
        <!-- end group detector -->
      <group name="NDAttributes" ndattr_default="true">
        <attribute name="NX_class" source="constant" value="NXcollection" type="string"></attribute>
      </group>
      <!-- end group NDAttribute (default) -->
      <group name="performance">
        <dataset name="timestamp" source="ndattribute"></dataset>
      </group>
      <!-- end group performance -->
    </group>
    <!-- end group instrument -->
    <group name="data">
      <attribute name="NX_class" source="constant" value="NXdata" type="string"></attribute>
      <hardlink name="data" target="/entry/instrument/detector/data"></hardlink>
      <!-- The "target" attribute in /entry/instrument/detector/data is used to
        tell Nexus utilities that this is a hardlink -->
    </group>
    <!-- end group data -->
  </group>
  <!-- end group entry -->
</xml>
```

Multiple Threads per Plugin

- Support for multiple threads running the processCallbacks() function in a single plugin.
- Can improve the performance of the plugin by a large factor.
- Linear scaling with up to 5 threads (the largest value tested) observed for most of the plugins that now support multiple threads.
- Maximum number of threads that can be used for the plugin is set in constructor and in IOC startup script.
- Actual number of threads to use controlled via an EPICS PV at run time, up to the maximum value.
- Optional sorting of NDArrays by uniqueId to attempt to output them in the correct order.
 - Several new parameters to control this option
- Plugins needed minor modifications to be thread-safe for multiple threads running in a single plugin object.
- Most compute-intensive plugins now support multiple threads.

Multiple Threads per Plugin 1 Thread

NDPluginBaseFull.adl@corvette.cars.aps.anl.gov

13SIM1:Stats5:

asyn port **STATS5**
Plugin type **NDPluginStats**
ADCore version **2.6.0**
Plugin version **2.6.0**
Array port **SIM1** SIM1
Array address **0** 0
Enable **Enable** Enable
Min. time **0.000** 0.000
Callbacks block **No** No
threads **1** 1
Max # threads **5**
Queue size/free **200**
Sort mode **Sorted** Sorted
Sort time **0.050** 0.050
Sort size/free **50**
disordered **Reset to 0** 2501
Array counter **Reset to 0** 2323
Array rate **120.00**
Execution time **8.151** msec
Dropped arrays **Reset to 0** 26491
Dropped outputs **Reset to 0** 0
dimensions **2**
Array Size **1024** 1024 0
Data type **Float32**
Color mode **Mono**
Bayer pattern **RGBB**
Unique ID **319525**
Time stamp **858340043.780**
Attributes file **StatsAttributes.xml**
Array callbacks **Enable** Enable
Process plugin **Process**
asyn record **On**

corvette (epics)

Terminal Sessions View X server Tools Games Settings Macros Help

Quick connect...

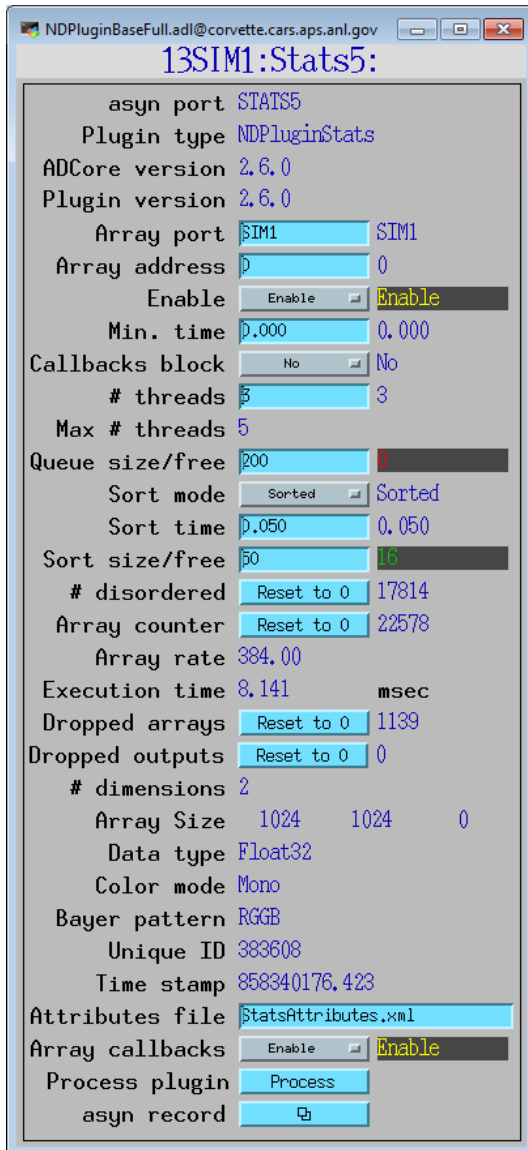
```
top - 06:48:51 up 56 days, 19:07, 15 users, load average: 1.76, 1.75, 1.18
Threads: 2251 total, 5 running, 2246 sleeping, 0 stopped, 0 zombie
%Cpu(s):  8.9 us,  1.4 sy,  0.0 ni, 89.6 id,  0.0 wa,  0.0 hi,  0.1 si,  0.0 st
KiB Mem : 65693432 total, 1427156 free, 3147908 used, 61118368 buff/cache
KiB Swap: 62500860 total, 61232724 free, 1268136 used, 61595632 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
44219	epics	20	0	6890364	1.260g	5876	R	99.9	2.0	6:53.98	STATS5_plugin_1
93793	epics	20	0	6890364	1.260g	5876	R	57.7	2.0	6:33.77	SimDetTask
93993	epics	20	0	470608	17040	5812	R	8.9	0.0	1:04.03	medm
93995	epics	20	0	6890364	1.260g	5876	S	6.9	2.0	0:55.83	CAS-event
89627	epics	20	0	470608	17040	5812	S	6.6	0.0	189:31.91	medm
93880	epics	20	0	6890364	1.260g	5876	S	5.2	2.0	1:32.56	cbLow
3254	epics	20	0	447296	3520	824	S	2.0	0.0	626:21.20	medm
44315	epics	20	0	66644	4544	1564	R	1.6	0.0	0:00.52	top
111915	epics	20	0	379296	10280	5432	S	1.0	0.0	10:12.91	medm
112333	epics	20	0	378916	2808	704	S	1.0	0.0	526:57.51	medm
147095	epics	20	0	379176	10044	5392	S	1.0	0.0	44:50.84	medm
8046	gpd_user	20	0	123592	5952	1032	S	0.7	0.0	355:03.32	motorPoller
8058	gpd_user	20	0	123592	5952	1032	S	0.7	0.0	305:43.29	164.54.160.56:5
17270	epics	20	0	3444656	4500	1124	S	0.7	0.0	87:32.98	164.54.160.190:

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

Multiple Threads per Plugin

3 Threads



NDPluginBaseFull.adl@corvette.cars.aps.anl.gov

13SIM1:Stats5:

asyn port **STATS5**

Plugin type **NDPluginStats**

ADCore version **2.6.0**

Plugin version **2.6.0**

Array port **SIM1** SIM1

Array address **0** 0

Enable **Enable**

Min. time **0.000** 0.000

Callbacks block **No**

threads **3**

Max # threads **5**

Queue size/free **200**

Sort mode **Sorted**

Sort time **0.050** 0.050

Sort size/free **50**

disordered **17814**

Array counter **22578**

Array rate **384.00**

Execution time **8.141** msec

Dropped arrays **1139**

Dropped outputs **0**

dimensions **2**

Array Size **1024 1024 0**

Data type **Float32**

Color mode **Mono**

Bayer pattern **RGGB**

Unique ID **383608**

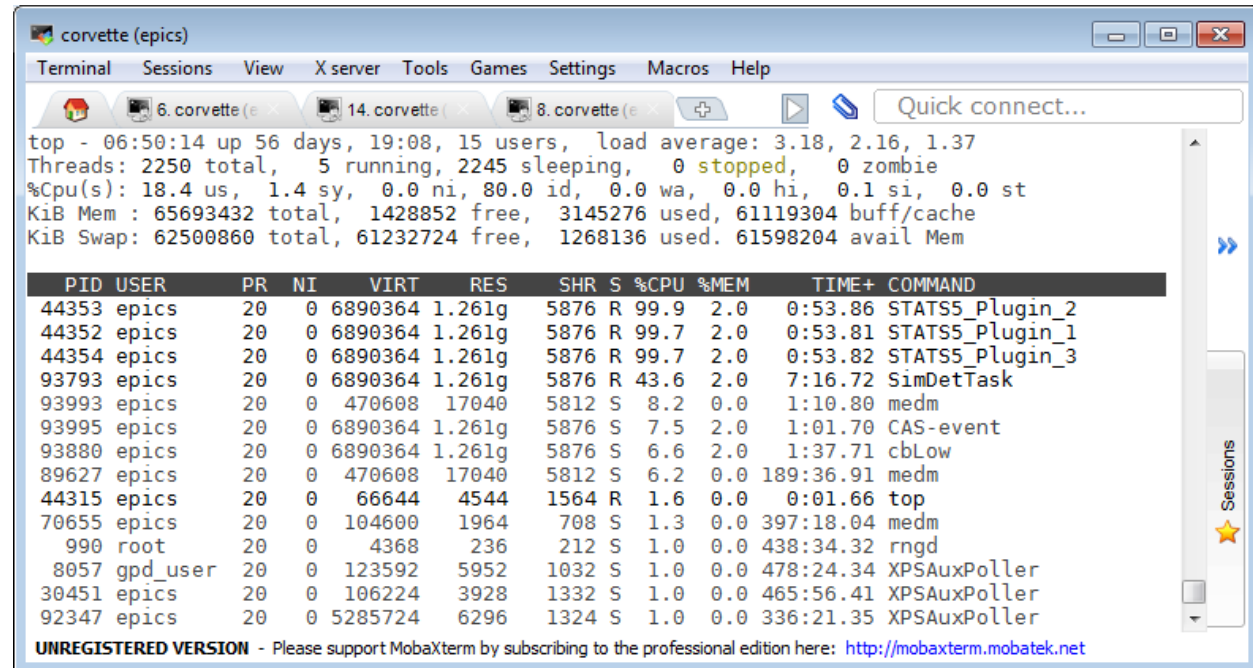
Time stamp **858340176.423**

Attributes file **StatsAttributes.xml**

Array callbacks **Enable**

Process plugin **Process**

asyn record



corvette (epics)

Terminal Sessions View X server Tools Games Settings Macros Help

6. corvette (e) 14. corvette (e) 8. corvette (e) Quick connect...

top - 06:50:14 up 56 days, 19:08, 15 users, load average: 3.18, 2.16, 1.37

Threads: 2250 total, 5 running, 2245 sleeping, 0 stopped, 0 zombie

%Cpu(s): 18.4 us, 1.4 sy, 0.0 ni, 80.0 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

KiB Mem : 65693432 total, 1428852 free, 3145276 used, 61119304 buff/cache

KiB Swap: 62500860 total, 61232724 free, 1268136 used, 61598204 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
44353	epics	20	0	6890364	1.261g	5876	R	99.9	2.0	0:53.86	STATS5_plugin_2
44352	epics	20	0	6890364	1.261g	5876	R	99.7	2.0	0:53.81	STATS5_plugin_1
44354	epics	20	0	6890364	1.261g	5876	R	99.7	2.0	0:53.82	STATS5_plugin_3
93793	epics	20	0	6890364	1.261g	5876	R	43.6	2.0	7:16.72	SimDetTask
93993	epics	20	0	470608	17040	5812	S	8.2	0.0	1:10.80	medm
93995	epics	20	0	6890364	1.261g	5876	S	7.5	2.0	1:01.70	CAS-event
93880	epics	20	0	6890364	1.261g	5876	S	6.6	2.0	1:37.71	cbLow
89627	epics	20	0	470608	17040	5812	S	6.2	0.0	189:36.91	medm
44315	epics	20	0	66644	4544	1564	R	1.6	0.0	0:01.66	top
70655	epics	20	0	104600	1964	708	S	1.3	0.0	397:18.04	medm
990	root	20	0	4368	236	212	S	1.0	0.0	438:34.32	rngd
8057	gpd_user	20	0	123592	5952	1032	S	1.0	0.0	478:24.34	XPSAuxPoller
30451	epics	20	0	106224	3928	1332	S	1.0	0.0	465:56.41	XPSAuxPoller
92347	epics	20	0	5285724	6296	1324	S	1.0	0.0	336:21.35	XPSAuxPoller

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

Multiple Threads per Plugin

5 Threads

NDPluginBaseFull.adl@corvette.cars.aps.anl.gov

13SIM1:Stats5:

asyn port **STATS5**

Plugin type **NDPluginStats**

ADCore version **2.6.0**

Plugin version **2.6.0**

Array port **SIM1** SIM1

Array address **0** 0

Enable **Enable** Enable

Min. time **0.000** 0.000

Callbacks block **No** No

threads **5** 5

Max # threads **5**

Queue size/free **200** 200

Sort mode **Sorted** Sorted

Sort time **0.050** 0.050

Sort size/free **50** 50

disordered **Reset to 0** 0

Array counter **Reset to 0** 2712

Array rate **482.00**

Execution time **7.644** msec

Dropped arrays **Reset to 0** 0

Dropped outputs **Reset to 0** 0

dimensions **2**

Array Size **1024 1024 0**

Data type **Float32**

Color mode **Mono**

Bayer pattern **RRGB**

Unique ID **446681**

Time stamp **858340306.866**

Attributes file **StatsAttributes.xml**

Array callbacks **Enable** Enable

Process plugin **Process**

asyn record **On**

corvette (epics)

Terminal Sessions View X server Tools Games Settings Macros Help

6. corvette (e) 14. corvette (e) 8. corvette (e) Quick connect...

```
top - 06:52:12 up 56 days, 19:10, 15 users, load average: 4.33, 2.90, 1.74
Threads: 2253 total, 5 running, 2248 sleeping, 0 stopped, 0 zombie
%Cpu(s): 24.8 us, 1.2 sy, 0.0 ni, 73.9 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
KiB Mem : 65693432 total, 1423484 free, 3149844 used, 61120104 buff/cache
KiB Swap: 62500860 total, 61232724 free, 1268136 used, 61593804 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
44393	epics	20	0	6890364	1.262g	5876	R	87.9	2.0	0:49.56	STATS5_plugin_5
44392	epics	20	0	6890364	1.262g	5876	R	86.9	2.0	0:49.90	STATS5_plugin_4
44389	epics	20	0	6890364	1.262g	5876	R	86.2	2.0	0:49.73	STATS5_plugin_1
44390	epics	20	0	6890364	1.262g	5876	S	86.2	2.0	0:49.84	STATS5_plugin_2
44391	epics	20	0	6890364	1.262g	5876	R	86.2	2.0	0:49.56	STATS5_plugin_3
93793	epics	20	0	6890364	1.262g	5876	S	36.1	2.0	8:12.23	SimDetTask
93993	epics	20	0	470608	17040	5812	S	8.2	0.0	1:19.96	medm
93880	epics	20	0	6890364	1.262g	5876	S	6.2	2.0	1:45.38	cbLow
93995	epics	20	0	6890364	1.262g	5876	S	6.2	2.0	1:09.41	CAS-event
89627	epics	20	0	470608	17040	5812	S	5.9	0.0	189:43.61	medm
44315	epics	20	0	66644	4544	1564	R	1.3	0.0	0:03.17	top
93985	epics	20	0	6890364	1.262g	5876	S	1.3	2.0	0:05.43	CAC-event
3254	epics	20	0	447296	3520	824	S	1.0	0.0	626:22.91	medm
8053	gpd_user	20	0	123592	5952	1032	S	1.0	0.0	459:24.68	motorPoller

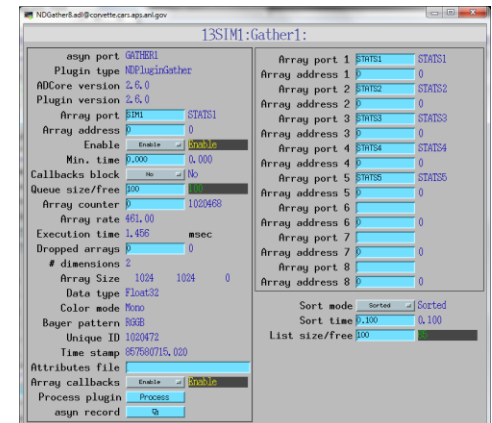
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

NDPluginScatter

- Used to distribute (scatter) the processing of NDArrays to multiple downstream plugins
 - Allows multiple instances of a plugin to process NDArrays in parallel, utilizing multiple cores to increase throughput.
 - Utilizes modified round-robin for choosing next output plugin
- More complex than multiple threads in a single plugin, but allows the plugins running in parallel to have different configurations or even be different plugins

NDPluginGather

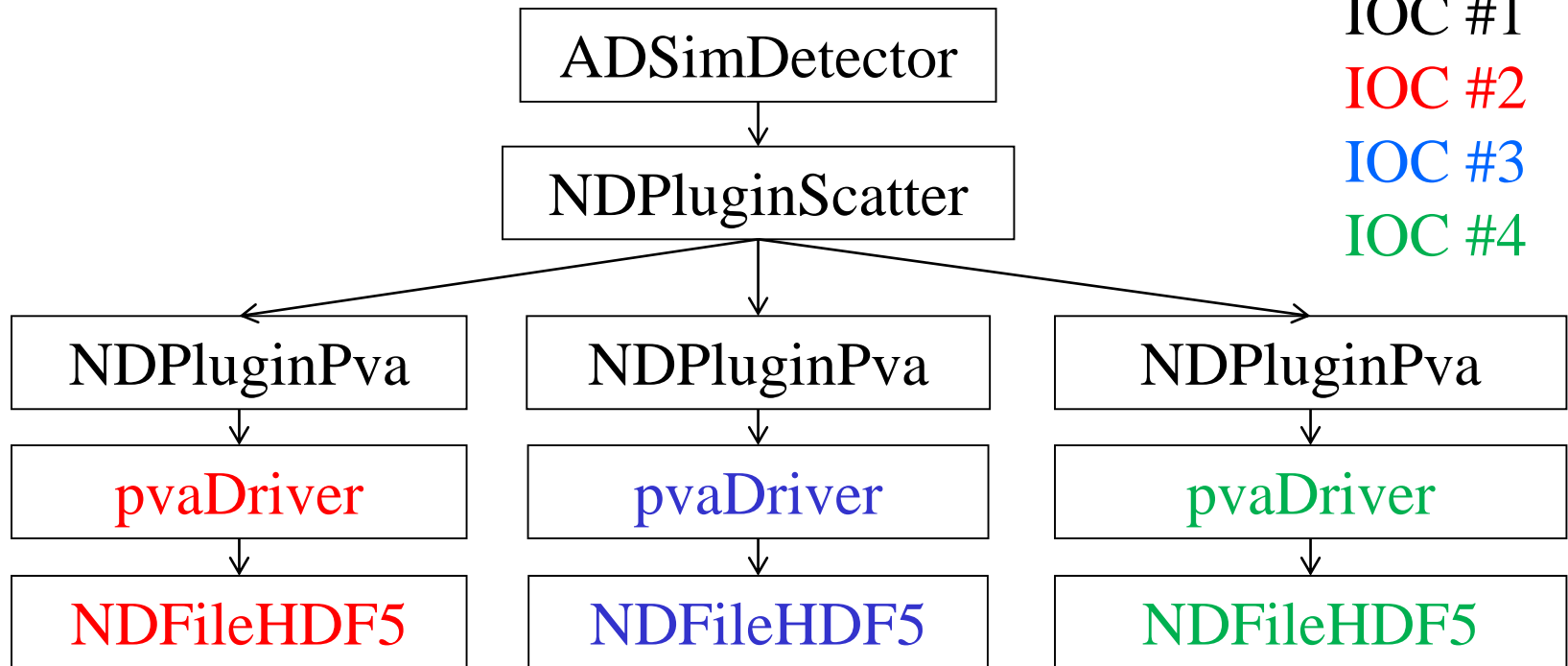
- Merges NDArrays from multiple upstream plugins into a single output stream.
- Designed to work with NDPluginScatter
- Optional sorting by uniqueId



Distributed Processing with NDPluginScatter + EPICS V4

Distribute HDF5 file writing to multiple IOCs (4096 x 3078 8-bit)

IOC #1
IOC #2
IOC #3
IOC #4



# IOCs	Files/sec	GB/sec
1	101.0	1.19
2	195.2	2.29
3	217.5	2.55

Demo

- Multi threaded plugins

Viewers

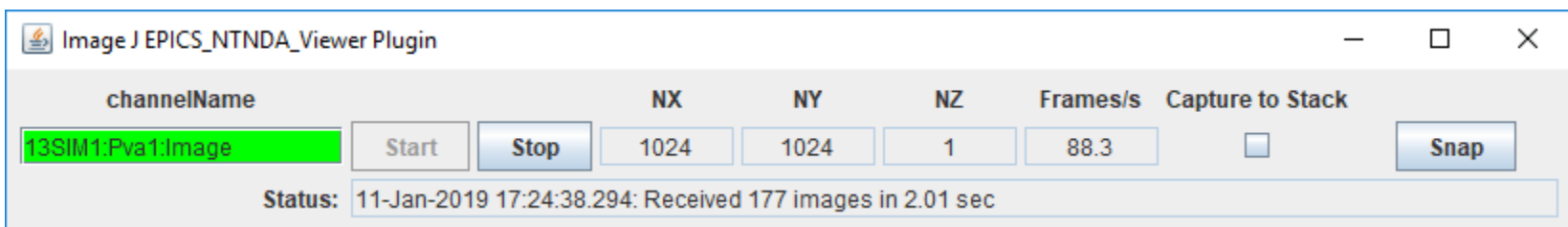
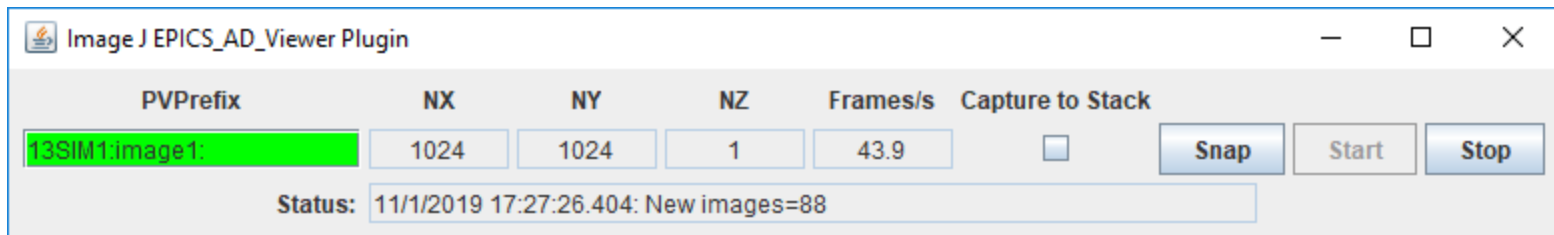
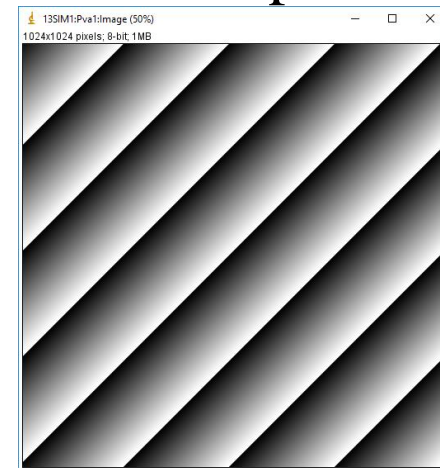
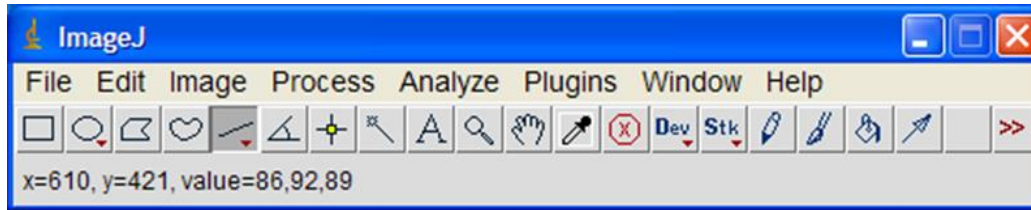
- areaDetector allows generic viewers to be written that receive images as EPICS waveform records over Channel Access
- Current viewers include:
 - ImageJ plugin EPICS_AD_Display. ImageJ is a very popular image analysis program, written in Java, derived from NIH Image.
 - EPICS_NTNDA_Viewer. Same as above but uses pvAccess rather than Channel Access.
 - ffmpegServer allows image display in any Web browser
 - ffmpegViewer high-performance Qt-based viewer for MJPEG stream

Viewers - NDPluginPva Advantages

- NTNDArray data transmitted "atomically" over the network
 - Channel Access requires separate PVs for the image data and the metadata (image dimensions, color mode, etc.)
- With Channel Access data type of waveform record is fixed at `iocInit`, cannot be changed at runtime.
 - If the user wants to view both 8-bit images, 16-bit images, and 64-bit double FFT images then waveform record needs to be 64-bit double, adding a factor of 8 network overhead when viewing 8-bit images.
 - `pvAccess` changes the data type of the NTNDArrays dynamically at run-time, removing this restriction.
- Channel Access requires setting `EPICS_CA_MAX_ARRAY_BYTES`
 - Source of considerable confusion and frustration for users.
 - `pvAccess` does not use `EPICS_CA_MAX_ARRAY_BYTES` and there is no restriction on the size of the NTNDArrays.

Viewers

- ImageJ plugins for displaying Images from EPICS Channel Access and pvAccess
- pvAccess plugin now supports decompression so compressed data can be transmitted across the network



Viewers - advanced

- EPICS_AD_Controller. Allows using the ImageJ ROI tools (rectangle and oval) to graphically define the following:
 - The readout region of the detector/camera
 - The position and size of an ROI (NDPluginROI)
 - The position and size of an overlay (NDPluginOverlay)
 - The plugin chain can include an NDPluginTransform plugin which changes the image orientation and an NDPluginROI plugin that changes the binning, size, and X/Y axes directions. The plugin corrects for these transformations when defining the target object.
 - Chris Roehrig wrote an earlier version of this plugin.

Other Drivers that use ADCore

- NDArrays are not limited to 2-D detectors
 - File, ROI, and statistics plugs are useful for other types of detectors
- Used for spectra arrays [NumMCAChannels, NumDetectors, NumPixels] for:
 - Xspress3 from Quantum Detectors
 - xMAP, Mercury and new FalconX from XIA
- Used for time-series data [NumTimePoints, NumInputs] for the quadEM quad electrometer software
 - AH401, AH501, TetrAMM from CaenEls
 - Two types of electrometers from BNL Instrumentation group (Peter Siddons)

Conclusions

- Architecture works well, easily extended to new detector drivers, new plugins and new clients
- Widely adopted
 - APS, SLAC, NSLS-II, CHESS, DLS, PSI, ESS, Australian Synchrotron, many others
- Base classes, `asynPortDriver`, `asynNDArrayDriver`, `NDPluginDriver` actually are generic, nothing “areaDetector” specific about them.
 - Used to implement other N-dimension detectors, e.g. the XIA xMAP (16 detectors x 2048 channels x 512 scan points) and quadEM (electrometers with 4 detectors x N time samples)
- Collaborative effort
 - Major contributions from Diamond, NSLS-II, SLAC, PSI, many others
- Code available on Github: <https://github.com/areaDetector>
- Thanks for your attention