

areaDetector: A module for EPICS area detector support

Mark Rivers

GSECARS, Advanced Photon Source

University of Chicago



areaDetector Introductory 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 FLIR USB-3 camera

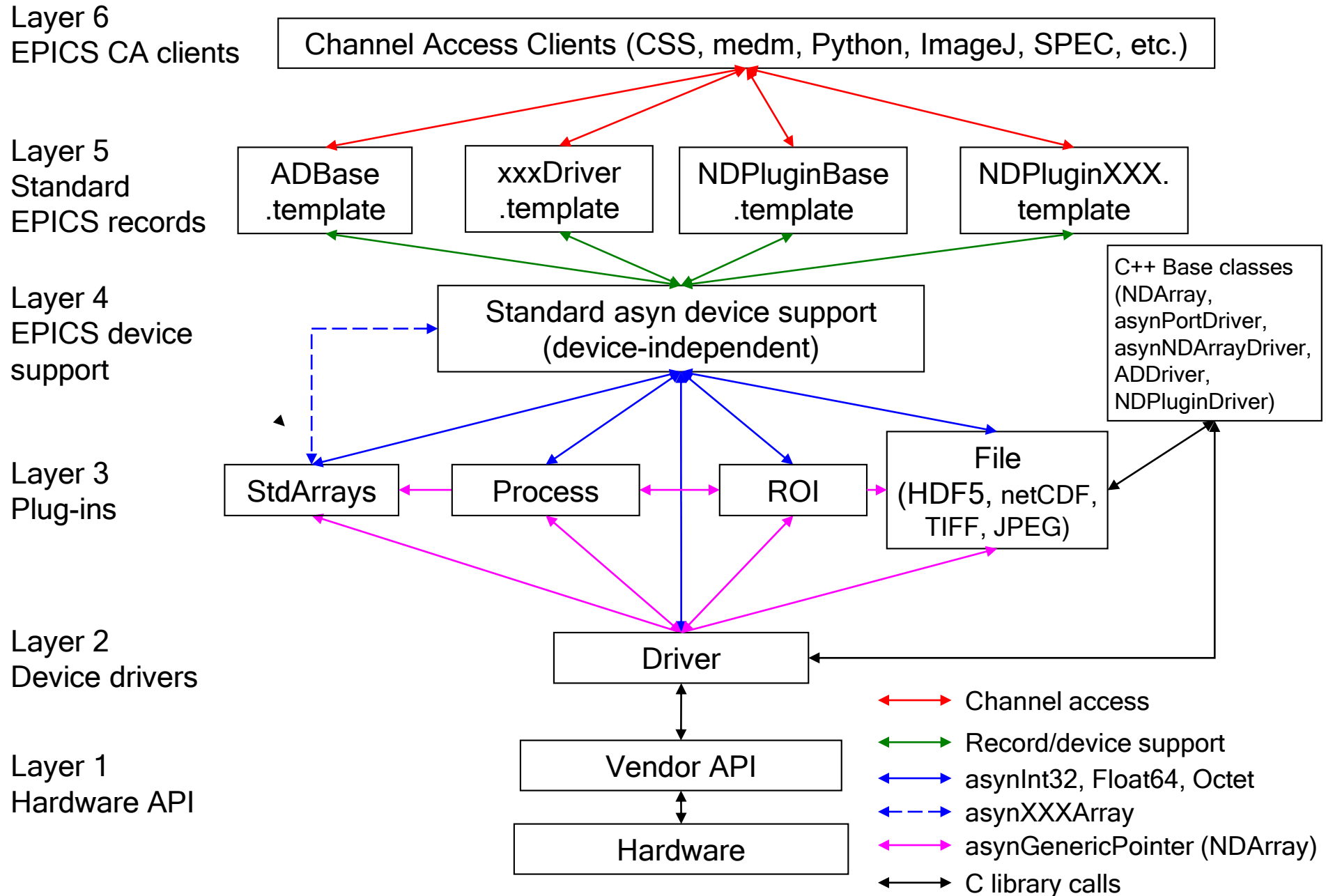
areaDetector - Goals

- Drivers for many detectors, especially those used 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 database 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.
- NDAttribute
 - 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 ~35 detector drivers covering a wide variety of detectors.

- Simulation driver
- GigE cameras (Prosilica/AVT, Point Grey/FLIR, any GigEVision camera via aravis library)
- FLIR and AVT 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 (Basler, Pixirad, Photonic Sciences, etc.)



ADBase.adl – Generic control screen

- Works with any detector
- Normally write custom control for each detector type
 - Hide unimplemented features
 - Expose driver-specific features

The screenshot displays the 'ADBase.adl' window titled 'Area Detector Control - 13SIM1:cam1:'. The interface is divided into several sections:

- Setup:** Contains fields for 'asyn port' (SIM1), 'EPICS name' (13SIM1:cam1), 'Manufacturer' (Simulated detector), and 'Model' (Basic simulator). It shows a 'Connected' status and buttons for 'Connect' and 'Disconnect'. A 'More' button is also present.
- Shutter:** Includes 'Shutter mode' (None), 'Status: Det.' (Closed), and 'EPICS' (Closed). It has 'Open/Close' buttons and 'Delay' fields for 'Open' (0.000) and 'Close' (0.000). An 'EPICS shutter setup' button is also shown.
- Readout:** A table-like section showing sensor and region parameters for X and Y axes.

	X	Y
Sensor size	640	480
Binning	1	1
Region start	0	0
Region size	640	480
Reverse	No	No
Image size	640	480
Image size (bytes)		307200
Gain	1.000	1.000
Data type	UInt8	UInt8
Color mode	Mono	Mono
- Collect:** Includes 'Exposure time' (0.010), 'Acquire period' (0.000), '# Images' (10), '# Images complete' (703), and '# Exp./image' (1). It has 'Image mode' (Continuous) and 'Trigger mode' (Internal) dropdowns. A 'Collecting' status is shown with 'Acquire' (Start/Stop) buttons. 'Detector state' is Readout, 'Time remaining' is 0.000, 'Image counter' is 703, 'Image rate' is 67.0, and 'Array callbacks' are Enabled.
- File:** Contains a 'Driver file I/O' button.

simDetector: Detector-specific screen

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

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

Simulation Detector - 13SIM1:cam1:

Setup

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**
ADCore version **3.0.0**
Connection **Connected**
Connection
Debugging

Shutter

Shutter mode **None**
Status: Det. **Closed** EPICS **Closed**
Open/Close
Delay: Open **0.000** Close **0.000**
EPICS shutter setup

Collect

Exposure time **0.001** **0.001**
Acquire period **0.002** **0.002**
Images **20** **20**
Images complete **8774**
Image mode **Continuous** **Continuous**
Trigger mode **Internal** **Internal**
Collecting
Acquire
Detector state **Waiting**
Time remaining **0.000**
Image counter **0** **8774**
Image rate **485.00**
Array callbacks **Enable**

Plugins

Readout

	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

Attributes

File **simDetectorAttributesMacros.xml**
Macros **CAMERA=13SIM1:cam1:,ID=ID13us:**
Status **Attributes file OK**
Simulation setup

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.

The screenshot shows the 'simDetectorSetup.adl' window with the title 'Simulation Detector Setup' and a subtitle '13SIM1:cam1:'. The window is divided into three main sections: Gains, Peak mode, and Sine mode.

Gains

Parameter	Value 1	Value 2
X	1.00	1.00
Y	1.00	1.00
Overall	100.000	100.000
Red	1.00	1.00
Green	1.00	1.00
Blue	1.00	1.00

Simulation mode

Sine

Reset Reset image

Peak mode

Start X	0	0
Start Y	0	0
Num X	2000	2000
Num Y	2000	2000
Step X	128	128
Step Y	256	256
Width X	10	10
Width Y	20	20
Variation	0	0
Noise	0	0

Sine mode

	X sine #1		Y sine #1	
Amplitude	1.00	1.00	1.00	1.00
Frequency	2.00	2.00	4.00	4.00
Phase	90.00	90.00	45.00	45.00
	X sine #2		Y sine #2	
Amplitude	1.00	1.00	1.00	1.00
Frequency	5.00	5.00	20.00	20.00
Phase	0.00	0.00	0.00	0.00

Operation

Add Add

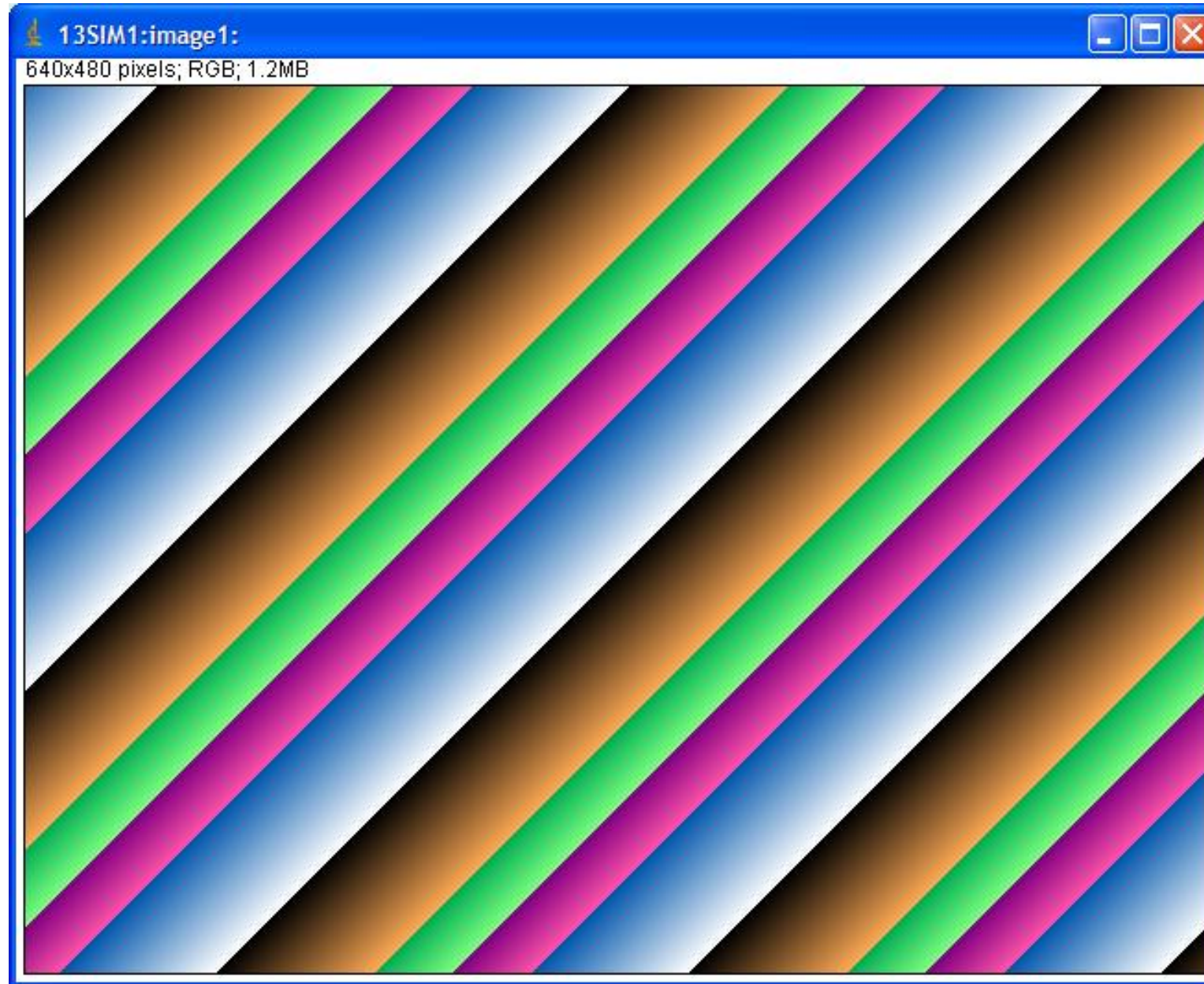
Offset

X	4.00	4.00
---	------	------

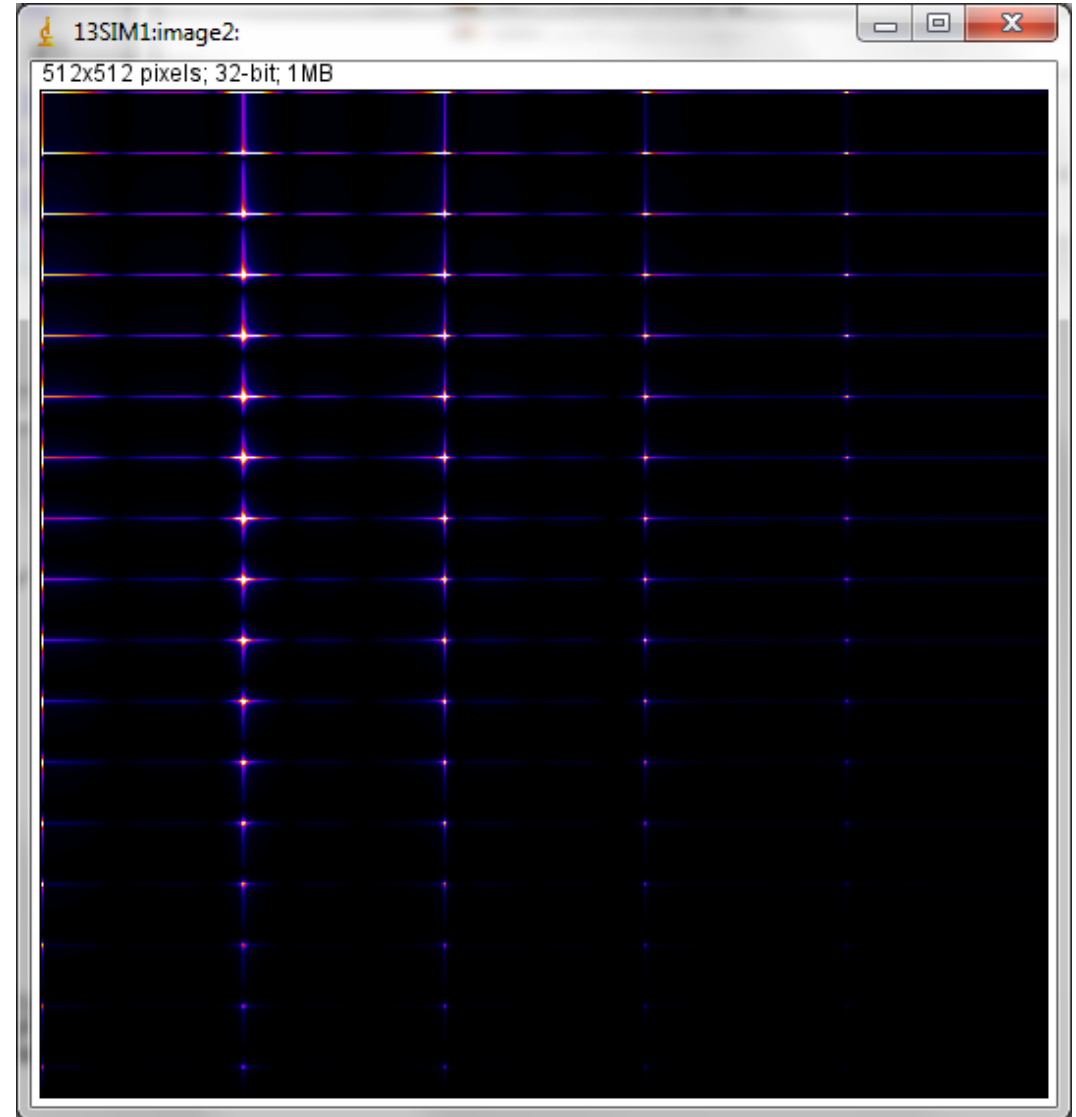
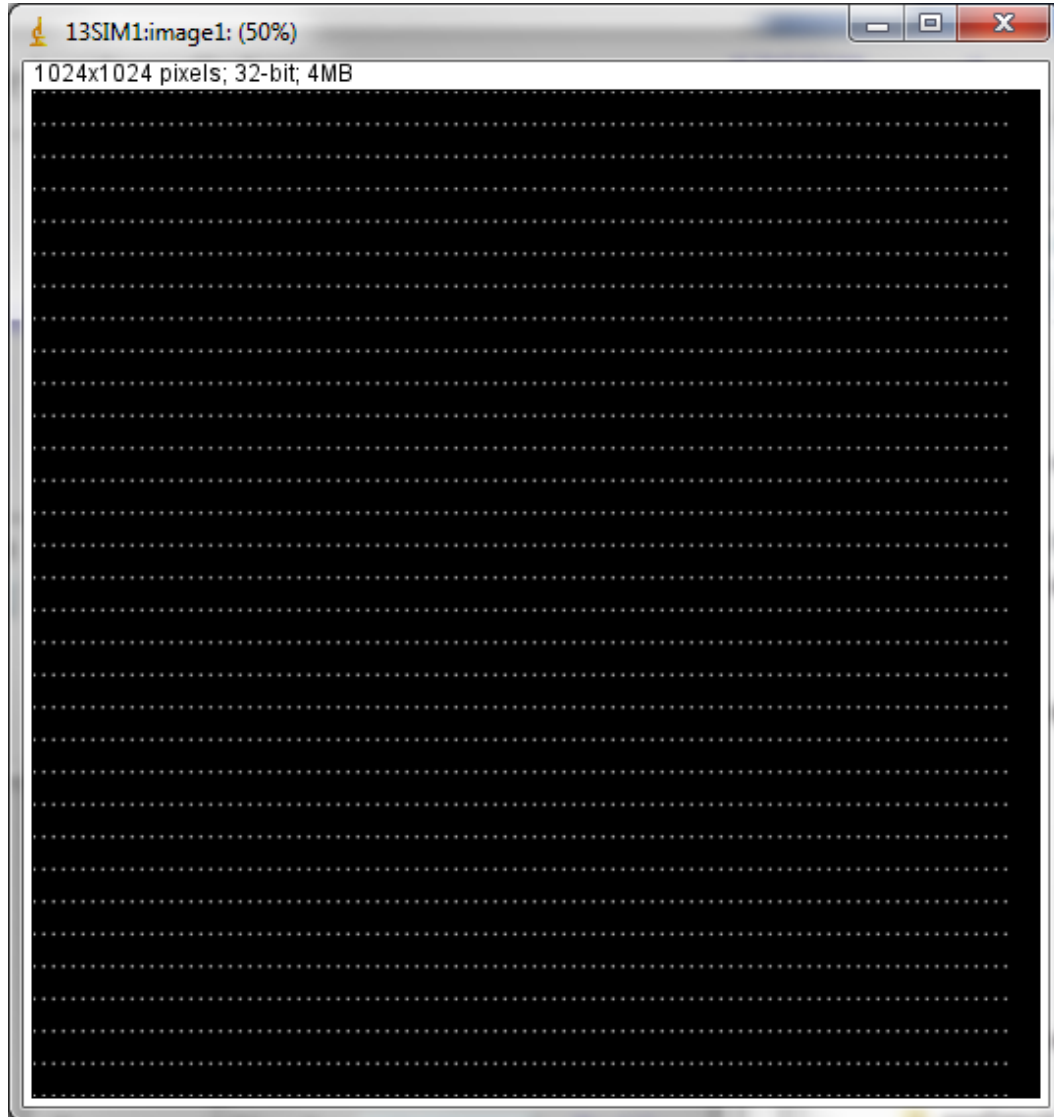
Noise

X	0.00	0.00
---	------	------

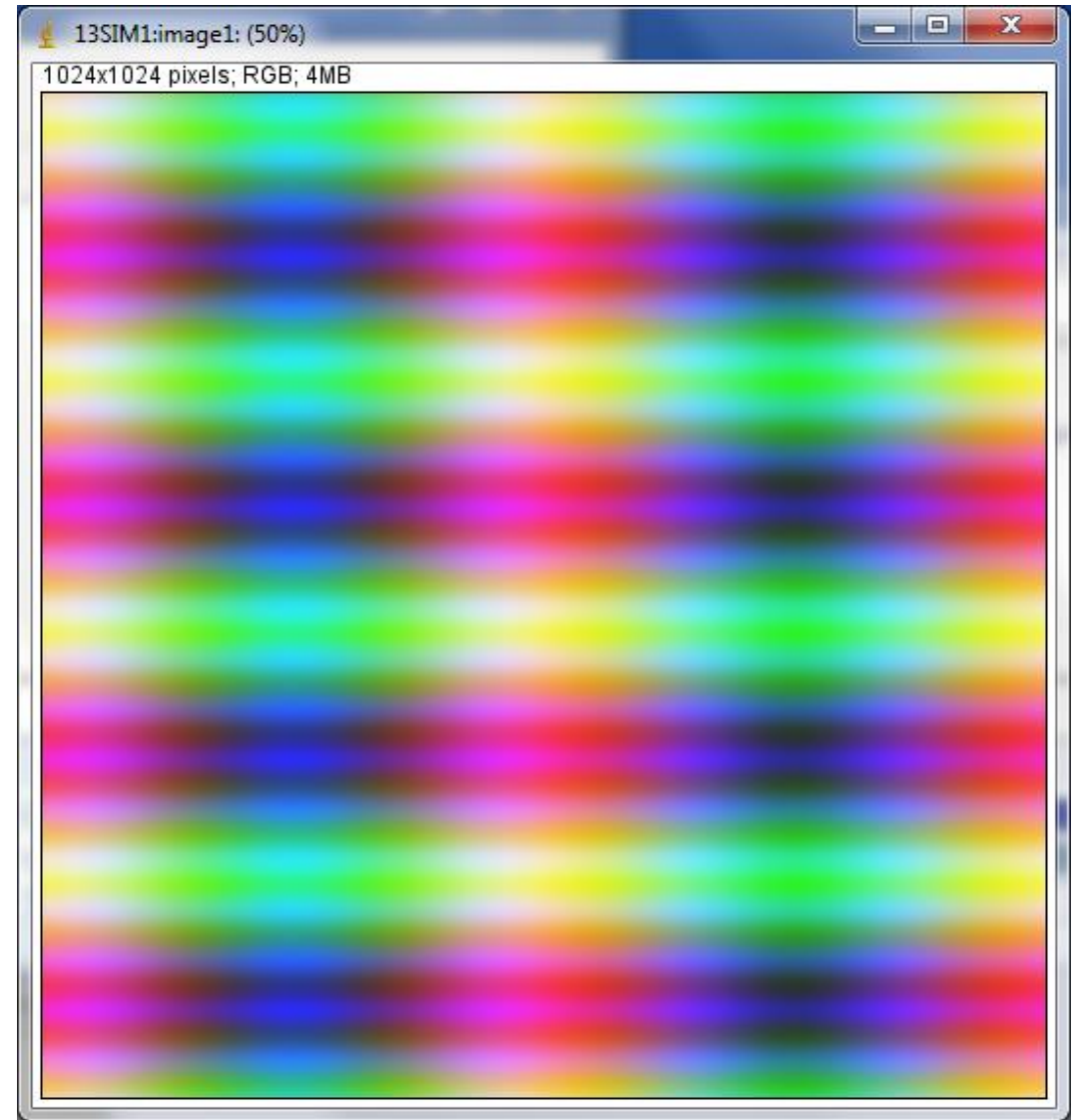
simDetector: Linear Ramp Mode



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 **Connect** **Disconnect**
Debugging **☐**

Collect

Exposure time **1.000** 1.000
Acquire period **1.001** 1.001
Images **10** 10
Delay time **0.000000** 0.000000
Exp./image **1** 1
Trigger mode **Internal** **Internal**
Collecting
Acquire **Start** **Stop**
Armed **Armed**
Image counter **0** 3
Image rate **1.00**
Array callbacks **Enable** **Enable**

Shutter

Shutter mode **EPICS PV**
Status: Det. **Closed** EPICS **Open**
Open/Close **Open** **Close**
Delay: Open **0.000** Close **0.000**
EPICS shutter setup **☐**

Attributes

File **DAC_P3.xml**
Macros
Status **Attributes file OK**

Plugins

All **File** **ROI**
Stats **☐** **☐Other #1** **☐Other #2**

Detector

Detector Size **1475** 195
Threshold (keV) **11.000** 11.000
Energy (keV) **22.000** 22.000
En./Thr. apply **Yes** **Apply**
Shaping time/Gain **7-30KeV/Fast/LowG**
Pixel cutoff **0**
Reset module power **Reset**
Reset power delay **1** 1
Read file timeout **20.000**
Gap fill **-1** -1
Temperature **0.0** 24.5 24.5
Humidity **0.0** 3.5 7.8
Status rate **Passive**

File

File **/cars5/Data/dac_user/2018/IDD-2018-1/Pilatus300K/** Exists: **Yes**
File path **/cars5/Data/dac_user/2018/IDD-2018-1/Pilatus300K**
pilatus300k
File name **pilatus300k**
Next file # **9** 9
Auto increment **Yes** **Yes** Ancillary information **☐**
Filename format **%s_%4.4d.tif** File format **TIFF** **TIFF**
Last filename **/cars5/Data/dac_user/2018/IDD-2018-1/Pilatus300K/pilatus300k_000**

Status

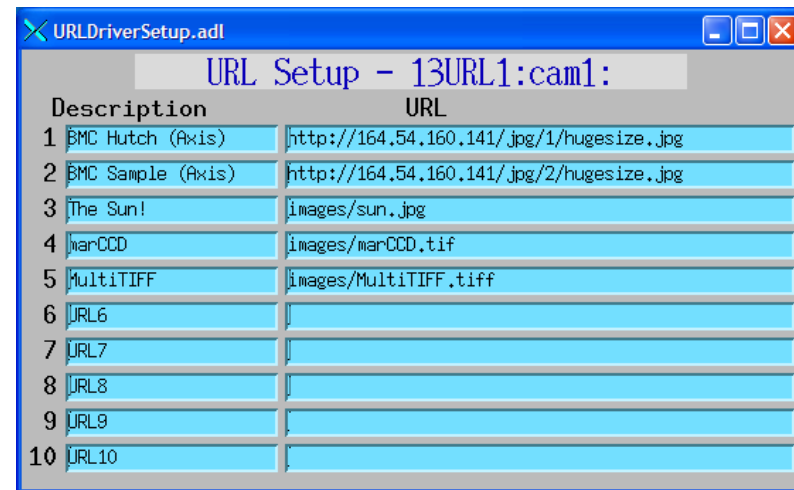
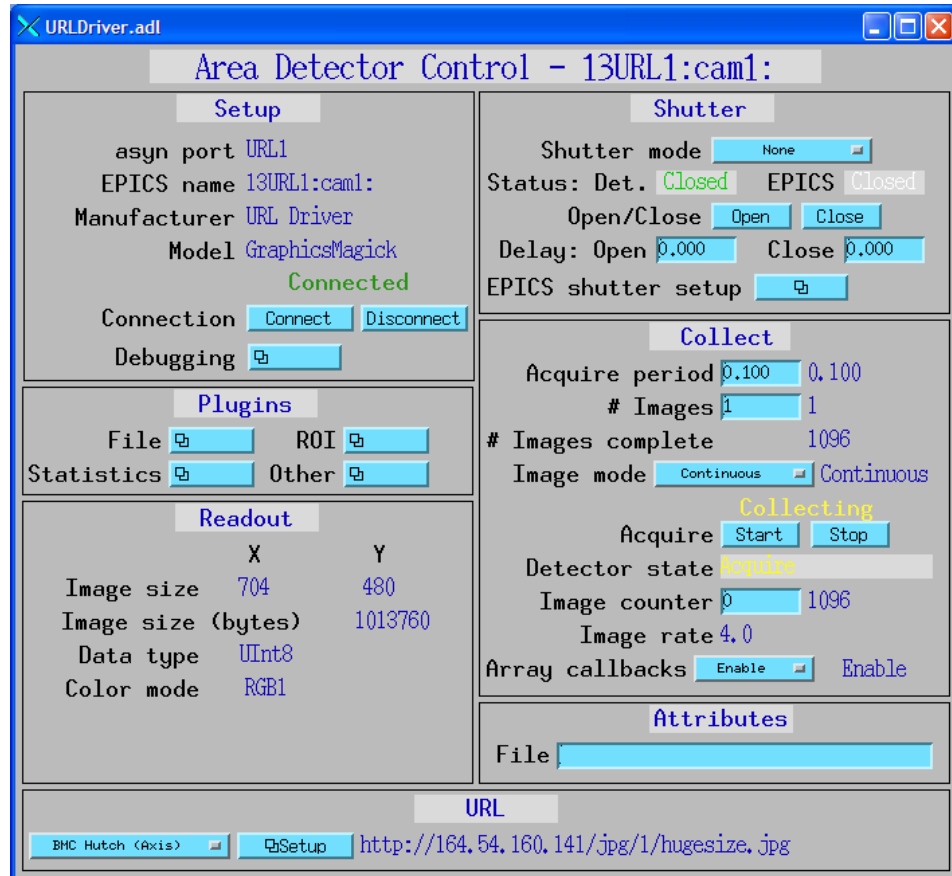
Status: Reading image file **/cars5/Data/dac_user/2018/IDD-2018-1/**
To camserver: Exposure **/cars5/Data/dac_user/2018/IDD-2018-1/Pilatus300**
From camserver: **15 OK** Starting **1.00000000** second background: **2018-01-21T1**

Data corrections

Bad pixel file: **/home/det/p2_det/config/bad_pixel.txt**
Bad pixels: **0**
Flat field file:
Flat field valid: **No** Min. flat field: **100**

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.).



GenICam

- **Generic Interface for Cameras** standard is the base for plug & play handling of cameras and devices.

“The goal of GenICam™ (Generic Interface for Cameras) is to provide a generic programming interface for all kinds of devices (mainly cameras), no matter what interface technology (GigE Vision, USB3 Vision, CoaXPress, Camera Link HS, Camera Link etc.) they are using or what features they are implementing.

- **GenApi**
 - Defines the mechanism used to provide the generic API via a self-describing XML file in the device. Part of GenApi is the Schema, which defines the format of the XML file.
- **SFNC (Standard Features Naming Convention)**
 - Standardizes the name, type, meaning and use of device features, so that devices from different vendors always use the same names for the same functionality.
- **GenTL (Transport Layer)**
 - Standardizes the transport layer programming interface. It is a low-level API to provide a standard interface to a device regardless of the transport layer. It allows enumerating devices, accessing device registers, streaming data and delivering asynchronous events. GenTL also has its own SFNC.

GenICam XML Files

- Every GenICam camera has an XML file inside it that can be accessed to determine the cameras available “features”.
- By reading and parsing this XML file one can automatically generate the EPICS database and OPI screens based on the specific features available in that camera. Small snippet for PixelFormat:

```
<Enumeration Name="PixelFormat" Namespace="Standard">
  <ToolTip>Format of the pixel data.</ToolTip>
  <Description>Format of the pixel data.</Description>
  <DisplayName>Pixel Format</DisplayName>
  <Visibility> Beginner</Visibility>
  <pIsLocked>TLParamsLocked</pIsLocked>
  <ImposedAccessMode>RW</ImposedAccessMode>
  <EnumEntry Name="Mono8" Namespace="Standard">
    <ToolTip>Pixel format set to Mono 8.</ToolTip>
    <Description>Pixel format set to Mono 8.</Description>
    <DisplayName>Mono 8</DisplayName>
    <pIsImplemented>Mono8Inq_Reg</pIsImplemented>
    <Value>0x01080001</Value>
  </EnumEntry>
```


GenICam and aravis

- GenICam is not open source
- There is a reference implementation of the GenICam API, but it cannot be released publicly
 - This is a pain, but we need to live with it
- aravis is an open-source toolkit based on reverse-engineering the GenICam specifications and protocols.
 - Based on glib. Until recently was Linux-only, now also supports Windows.
 - Previously supported only GigE, but now also supports USB

ADGenICam

- Vendor APIs for GenICam cameras are very similar, and programming straight to them results in lots of redundant code.
- Instead create a new areaDetector base class to support any GenICam camera
- Much of the generic code is in this layer
 - Implements code to get and set GenICam features
- Creates the areaDetector parameter library dynamically at iocInit from the drvUser fields passed by each record.

ADGenICam

- Derived classes for real drivers
 - Implement the code to read and write features to the device
 - Implement the code to stream the images from the device
- ADAravis
 - Uses aravis library
 - Linux only (most versions, e.g. RHEL 7). Could in principle now make it work with Windows.
- ADSpinnaker
 - Driver for FLIR/Point Grey cameras using their Spinnaker SDK
 - Windows and new versions of Linux (e.g. Ubuntu 18, not RHEL 7)
- ADVimba
 - Driver for AVT/Prosilica cameras using their Vimba SDK
 - Windows and Linux (most versions, e.g. RHEL 7)
- ADVimba
 - Driver for AVT/Prosilica cameras using their Vimba SDK
 - Windows and Linux (most versions, e.g. RHEL 7)

arv-tool (from aravis package)

- Find the cameras

```
corvette: ADAravis>bin/linux-x86_64/arv-tool-0.6
Allied Vision Technologies-02-2142A-06178 (164.54.160.58)
Allied Vision Technologies-02-2604A-07008 (164.54.160.104)
Allied Vision Technologies-50-0503317598 (164.54.160.62)
Allied Vision Technologies-50-0503419258 (164.54.160.21)
PointGrey-13481965 (164.54.160.114)
Prosilica-02-2142A-06110 (164.54.160.57)
```

- Extract the XML file

```
>arv-tool-0.6 -n PointGrey-13481965 genicam > BFly-20E4C.xml
```

- Look at the XML file

```
>more BFly-20E4C.xml
PointGrey-13481965 (164.54.160.114)
<?xml version="1.0" encoding="UTF-8"?>
<!--
<RegisterDescription
xmlns:xi="http://www.w3.org/2003/XInclude"
...
```

- Must run this once on Linux for each new camera type to create XML file.

Python tool to create template file from XML file

- Create the database with makeDb.py

```
>scripts/makeDb.py BFly-20E4C.xml BFly-20E4C.template
```

- Snippet of the template file

```
record(ai, "$(P)$(R)GC_ExposureTime_RBV") {  
    field(DTYP, "asynFloat64")  
    field(INP, "@asyn($(PORT),$(ADDR=0),$(TIMEOUT=1))GC_D_ExposureTime")  
    field(PREC, "3")  
    field(SCAN, "I/O Intr")  
    field(DISA, "0")  
}  
record(ao, "$(P)$(R)GC_ExposureTime") {  
    field(DTYP, "asynFloat64")  
    field(OUT, "@asyn($(PORT),$(ADDR=0),$(TIMEOUT=1))GC_D_ExposureTime")  
    field(PREC, "3")  
    field(DISA, "0")  
}
```

- Creates output and readback record for each feature that the camera supports.

Python tool to create medm files from XML file

- Create the medm files with makeAdl.py

```
>scripts/makeAdl.py BFly-20E4C.xml BFly-20E4C
```

```
>ls -l *.adl
```

```
-rw-rw-r-- 1 epics domain users 54593 Apr 17 16:53 BFly-20E4C-features_1.adl  
-rw-rw-r-- 1 epics domain users 48936 Apr 17 16:53 BFly-20E4C-features_2.adl  
-rw-rw-r-- 1 epics domain users 21322 Apr 17 16:53 BFly-20E4C-features_3.adl
```

- Snippet of the adl file

```
# (Menu Button)
```

```
menu {  
    object {  
        x=235  
        y=520  
        width=150  
        height=20  
    }  
    control {  
        chan="$ (P) $ (R) GC_SaturationAuto"  
        clr=14  
        bclr=51  
    }  
}
```

Main medm screen for ADGenICam

ADGenICam.adl@corvette

GenICam Camera - 13ARV1:cam1:

Setup	Shutter	Trigger
asyn port ARV1	Shutter mode None	Mode OFF Off
EPICS name 13ARV1:cam1:	Status: Det. Closed EPICS Closed	Source Software Software
Manufacturer Point Grey Research	Open/Close Open Close	Overlap Off Off
Model Blackfly S BFS-PGE-13	Delay: Open 0.000 Close 0.000	Exp. mode Timed Timed
Serial number 17165235	EPICS shutter setup	Software Trigger
Firmware version 1702.1.1.0		
SDK version 0.6.3		
Driver version 1.1.0		
ADCore version 3.7.0		
Connected		
Connection Connect Disconnect		
Debugging		

Plugins	Collect	Readout
All File ROI	Exposure time 0.001 0.001	X Y
Stats Other #1 Other #2	Exposure auto Off Off	Sensor size 1280 1024
	Acquire period 0.050 0.050	Region start 0 0
	Frame rate 83.072 19.998	Region size 1280 1024
	Frame rate enable Yes Yes	Binning 1 1
	# Images 100 100	Image size 1280 1024
	# Images complete 299	Image size (bytes) 1310720
	# Exp./image 1 1	Gain 1.000 0.000
	Image mode Continuous Continuous	Gain auto Off Off
	Collecting	Data type UInt8
	Acquire Start Stop	Color mode Mono
	# Queued arrays 0	Pixel format Mono8 Mono8
	Wait for plugins Yes	
	Acquire busy Acquiring	
	Detector state Acquire	
	Status	
	Image counter 0 409	
	Image rate 20.00	
	Array callbacks Enable Enable	

Buffers	Attributes
Buffers used 0	File
Buffers alloc/free 0 0	Macros
Memory max/used (MB) 0.0 0.0	Status File not found
Buffer & memory polling Passive	
Empty free list Empty	
Camera-specific features	

Identical screens are provided for CSS/BOY, Phoebus, caQtDM and EDM via auto-convert

Auto-generated medm screens for PGR BlackflyS 13Y3M

Screen #1

PGR_BlackflyS_13Y3M-features_1.adl@corvette

PGR_BlackflyS_13Y3M Features Screen #1 - 13ARV1:cam1:

SerialPortControl		AcquisitionControl		AnalogControl	
SerialPortSelector	SerialPort0	AcquisitionMode	Continuous	GainSelector	All
SerialPortSource	Off	AcquisitionStart	AcquisitionStart	Gain	0.000 0.000
SerialPortBaudRate	Baud57600	AcquisitionStop	AcquisitionStop	GainAuto	Continuous Off
SerialPortDataBits	8	AcquisitionFrameCount	2 100	BlackLevelSelector	All All
SerialPortStopBits	Bits1	AcquisitionBurstFrameCount	1 1	BlackLevelRaw	0 0
SerialPortParity	None	ExposureMode	Timed	BlackLevel	0.000 0.000
SerialTransmitQueueMaxCharacterCount	4096 4096	ExposureTime	0.001 0.001	BlackLevelClampingEnable	No No
SerialTransmitQueueCurrentCharacterCount	0 0	ExposureAuto	Continuous Off	BalanceRatioSelector	0 0
SerialReceiveQueueMaxCharacterCount	4096 4096	AcquisitionFrameRate	83.072 19.998	BalanceRatio	0.000 0.000
SerialReceiveQueueCurrentCharacterCount	0 0	AcquisitionResultingFrameRate	83.072 19.998	BalanceWhiteAuto	N.A. N.A.
SerialReceiveQueueClear	SerialReceiveQueueClear	AcquisitionFrameRateEnable	Yes Yes	Gamma	0.800 0.800
SerialReceiveFramingErrorCount	0 0	AcquisitionLineRate	89734.38 89734.386	GammaEnable	Yes Yes
SerialReceiveParityErrorCount	0 0	TriggerSelector	FrameStart	SharpeningEnable	No No
		TriggerMode	Off	SharpeningAuto	No No
		TriggerSoftware	TriggerSoftware	Sharpening	0.000 0.000
		TriggerSource	Software	SharpeningThreshold	0.000 0.000
		TriggerActivation	N.A.		
		TriggerOverlap	Off		
		TriggerDelay	0.000 0.000		
		SensorShutterMode	Global		

FLIR/Point Grey drivers

- ADSpinnaker driver for all cameras from FLIR/Point Grey using their Spinnaker SDK.
- GigE, USB 3.0, and 10 GigE camera
- High performance, low cost
- I will demonstrate USB-3.0 camera today
- Example models:

Model	Interface	Resolution	Price	Measured Speed (frames/s)	Measured Speed (MB/s)
BlackFlyS BF2-PGE-13Y3M	1-Gbit Ethernet	1280x1024	\$415	83 frames/s	104 MB/s
Grasshopper3 GS3-U3-23S6M	USB-3	1920x1200	\$1,179	156 frames/s	343 MB/s
Oryx ORX-10G-51S5M	10-Bbit Ethernet	2448x2048	\$2,258	163 frames/s	779 MB/s

FLIR/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



FLIR/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
- Used for tomography at 3 APS beamlines, replaced Andor Neo and PCO Edge
- \$1,179



Point Grey 10-Gbit Ethernet Camera

Oryx ORX-10G-310S9M

- 6464 x 4852 global shutter CMOS (31 MP)
- 10-Gbit Ethernet interface
- Sony IMX342 3.45 micron pixels
- Dynamic range of 66 dB
- Peak QE of 65%
- Read noise of 5.2e-
- 12-bit, 10-bit, or 8-bit data
- Max frame rate of 26 fps
 - 778 MB/S, >8X faster than GigE
- Used for tomography on 2 APS beamlines



Model	Resolution	Price	Speed (frames/s)	Speed (MB/s)
ORX-10G-310S9M	6464x4852	\$7,089	26 frames/s	778 MB/s
ORX-10G-123S6M	4096x3000	\$4,666	68 frames/s	797 MB/s
ORX-10G-51S5M	2448x2048	\$2,258	162 frames/s	774 MB/s
ORX-10GS-34S4M	2048x1536	\$1,774	216 frames/s	648 MB/s

ADSpinnaker Driver

ADSpinnaker.adl@corvette

FLIR/Spinnaker - 13SP1:cam1:

Setup
asyn port SP1
EPICS name 13SP1:cam1:
Manufacturer FLIR
Model Oryx ORX-10G-51S5M
Serial number 18011754
Firmware version 1710.0.0.0
SDK version 1.20.0.14
Driver version 1.0.0
ADCore version 3.7.0

Connected
Connection
Debugging

Plugins

Trigger
Mode Off
Source Software
Overlap Off
Exp. mode Timed
Software

Buffers
Buffers used 0
Buffers alloc/free 1 1
Memory max/used (MB) 0.0 4.8
Buffer & memory polling 1 second
Empty free list

Shutter
Shutter mode
Status: Det. Closed EPICS Closed
Open/Close
Delay: Open 0.000 Close 0.000
EPICS shutter setup

Collect
Exposure time 0.001 0.001
Exposure auto Off
Acquire period 0.006 0.006
Frame rate 0.000 162.266
Frame rate enable No
Images 10 10
Images complete 11577
Exp./image 1 1
Image mode Continuous

Collecting
Acquire
Queued arrays 0
Wait for plugins
Acquire busy Acquiring
Detector state Waiting
Status
Image counter 0 6562094
Image rate 163.00
Array callbacks Enable
Camera-specific features

Status
Status rate
Buffer underrun 0
Failed buffer 0
Failed packet 0
Temperature 57.5

Readout

	X	Y
Sensor size	2448	2048
Region start	0	0
	2448	2048
Region size	2448	2048
	1	1
Binning	1	1
Image size	2448	2048
Image size (bytes)	5013504	
Gain	1.000	1.000
Gain auto	<input type="button" value="Off"/>	Off
Data type	UInt8	
Color mode	Mono	
Pixel format	<input type="button" value="Mono8"/>	Mono8
Convert format	<input type="button" value="None"/>	None
Timestamp mode	<input type="button" value="Camera"/>	Camera
UniqueId mode	<input type="button" value="Camera"/>	Camera

Attributes
File
Macros
Status File not found

Plugins

- Designed to perform real-time processing of data, running in the EPICS IOC (not over EPICS Channel Access)
- Receive NDArray 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 NDArray 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 NDArray callbacks to another driver or plugin
- Plugins can be *sources* of NDArray 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

NStdArrays.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 RGGB

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

commonPlugins.adl

13SIM1: Common Plugins

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 Automatic

Enable scaling Enable Enable

Scale divisor 2 2

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

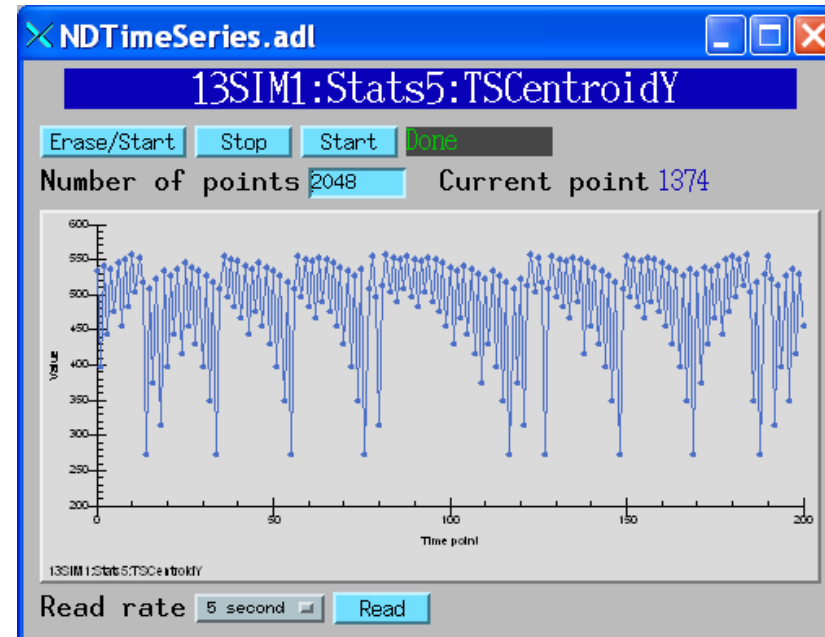
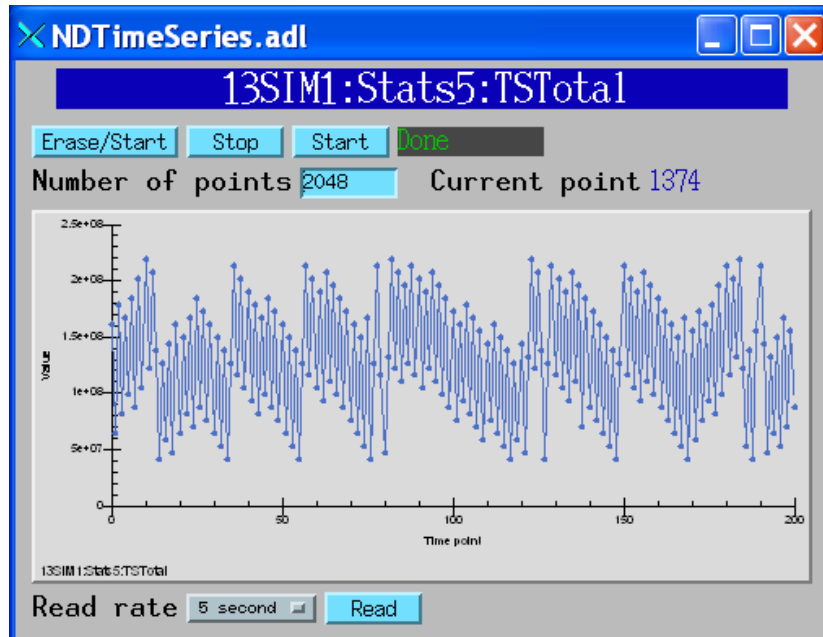
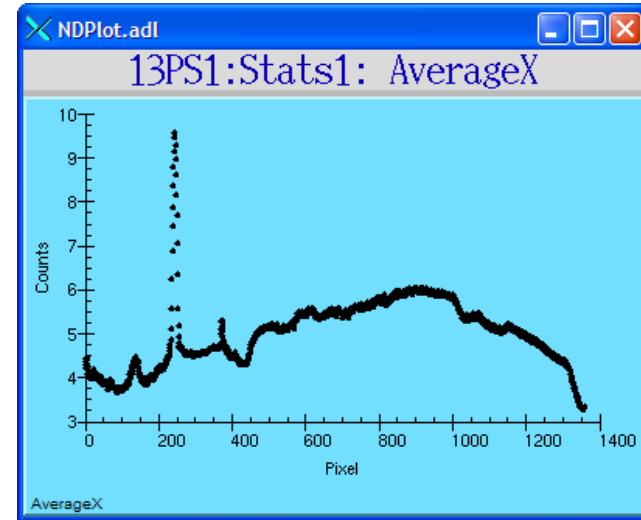
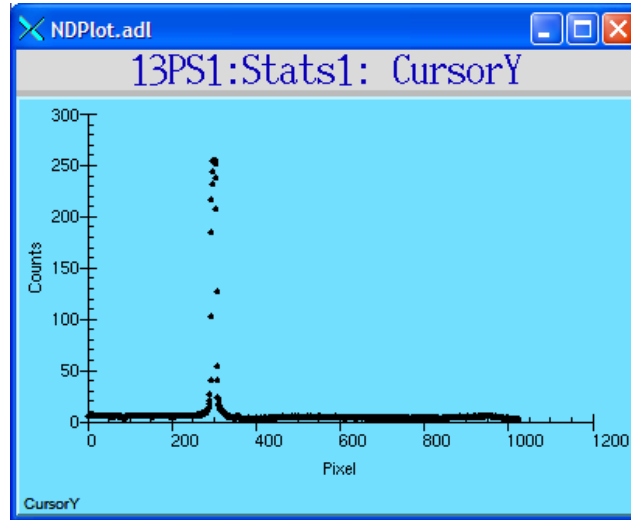
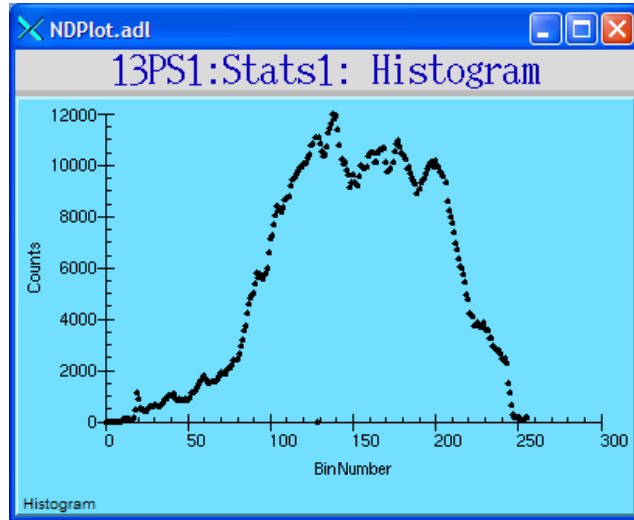
Statistics plugin

NDStats.adl

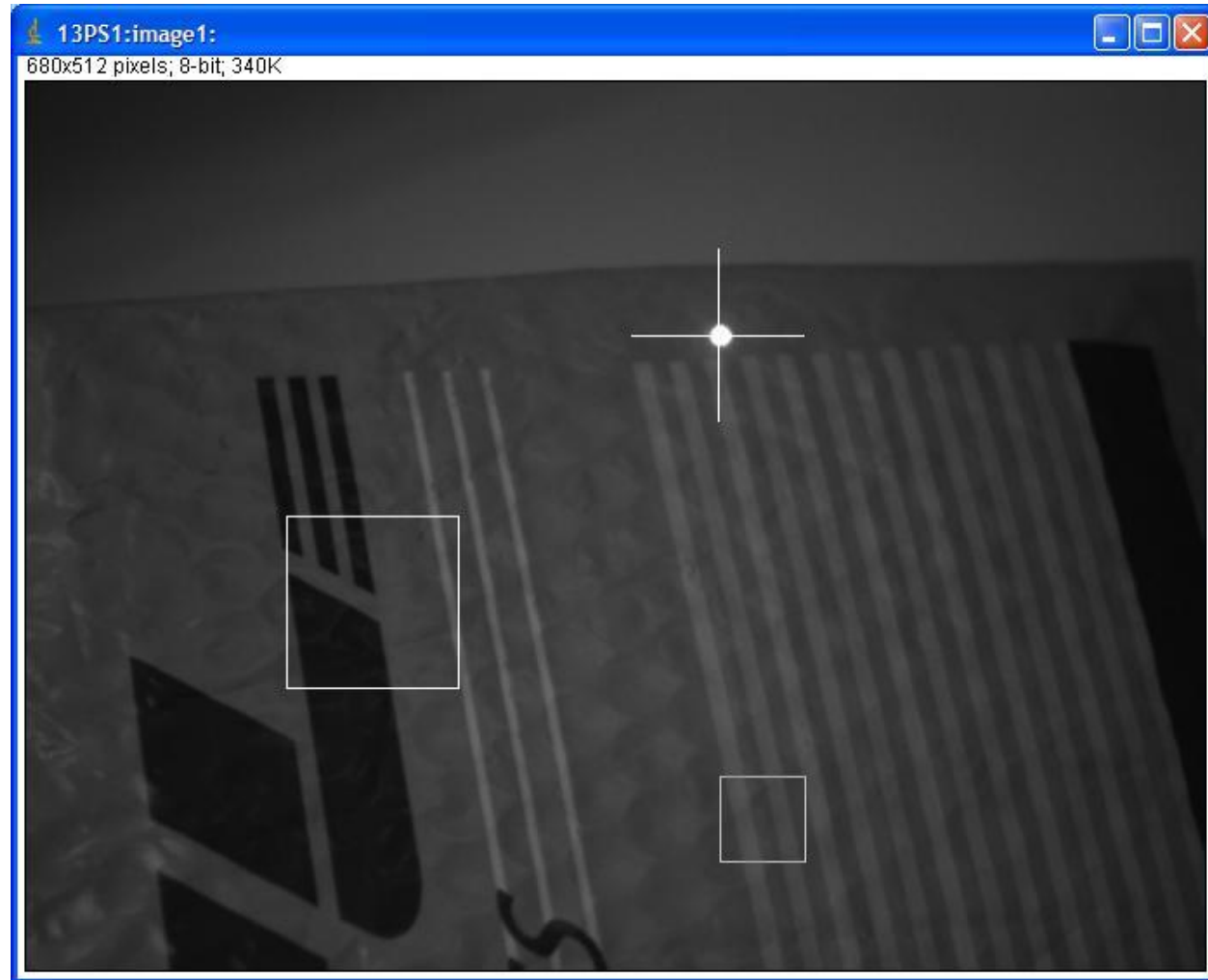
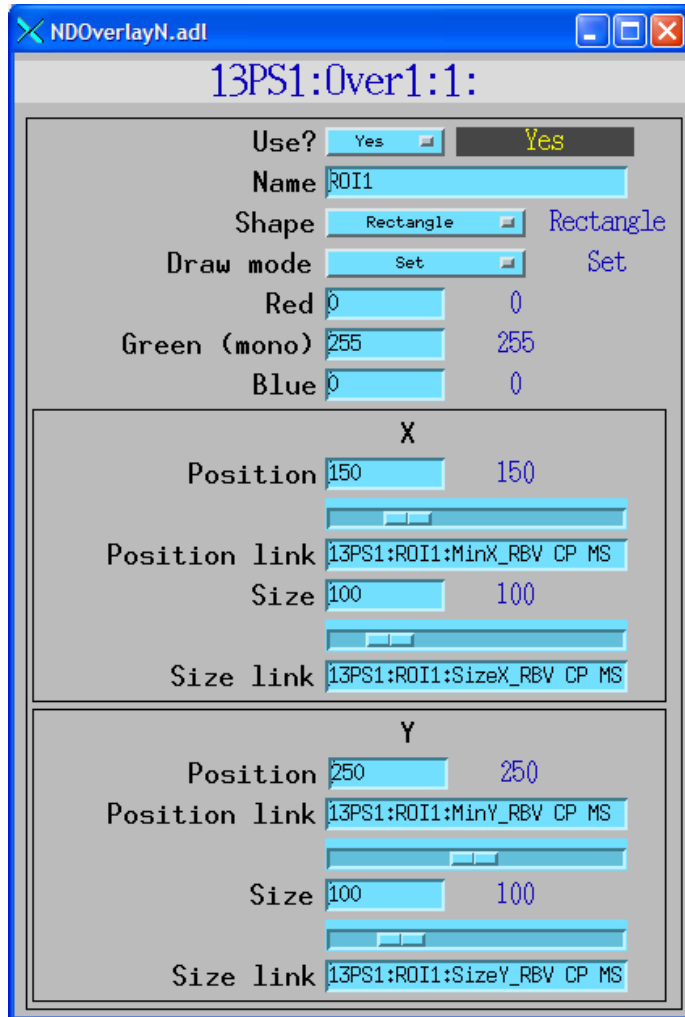
13SIM1:Stats5:

<p>asyn port STATS5</p> <p>Plugin type NDPluginStats</p> <p>Array port SIM1 SIM1</p> <p>Array address 0 0</p> <p>Enable Enable Enable</p> <p>Min. time 0.000 0.000</p> <p>Callbacks block 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 </p>	<p>Statistics</p> <p>Compute statistics 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 </p> <p>Centroid</p> <p>Compute centroid 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 </p> <p>Time Series</p> <p>Erase/Start Stop Start Acquiring</p> <p>Number of points 2048</p> <p>Current point 82</p> <p>Read rate 5 second Read</p>	<p>Profiles</p> <p>Compute profiles Yes Yes</p> <p>Size X 1024 Y 1024</p> <p>256 256</p> <p>Cursor X </p> <p>256 256</p> <p>Cursor Y </p> <p>Plot </p> <p>Histogram</p> <p>Compute histogram? 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 </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

SIM1SIM1

Array address

00

Enable

EnableEnable

Min. time

0.0000.000

Callbacks block

NoNo

Queue size/free

2020

Array counter

011572

Array rate

47.0

Dropped arrays

00

dimensions

2

Array Size

102410240

Data type

Int8

Color mode

Mono

Bayer pattern

RGGB

Unique ID

12032

Time stamp

717887092.888

Attributes file

asyn record

Background subtraction

Save background

SaveInvalid

Enable background

DisableDisable

Flat field normalization

Save flat field

SaveInvalid

Enable flat field

DisableDisable

Scale flat field

255255

Scale and Offset

Enable scale/off.

DisableEnable

Auto scale/off.

Auto calc

Scale value

0.1042.50

Offset value

0.000.00

Low/High Clipping

Enable low clip

DisableEnable

Low clip value

1000

Enable high clip

DisableEnable

High clip value

150255

Output data type

Data type

AutomaticAutomatic

Recursive filter

Enable filter

DisableDisable

N filter

100100

N filtered

0

Filter type

RecursiveAve

Reset filter

Reset

Auto reset filter

Yes

Filter callbacks

Every array

00ffset

0.000.00

0Scale

1.001.00

0C1

1.001.00

0C2

-1.00-1.00

0C3

0.000.00

0C4

1.001.00

F0ffset

0.000.00

FScale

1.001.00

FC1

1.001.00

FC2

-1.00-1.00

FC3

0.000.00

FC4

1.001.00

R0ffset

0.000.00

RC1

0.000.00

RC2

1.001.00

0[n]

$$0[n] = 00ffset + 0Scale * ((0C1 + 0C2 / N) * F[n-1] + (0C3 + 0C4 / N) * I[n])$$

F[n]

$$F[n] = F0ffset + FScale * ((FC1 + FC2 / N) * F[n-1] + (FC3 + FC4 / N) * I[n])$$

On filter reset:

$$F[0] = R0ffset + RC1 * F[n] + RC2 * I[0]$$

I = Input array in callback

F = Stored filter (double precision)

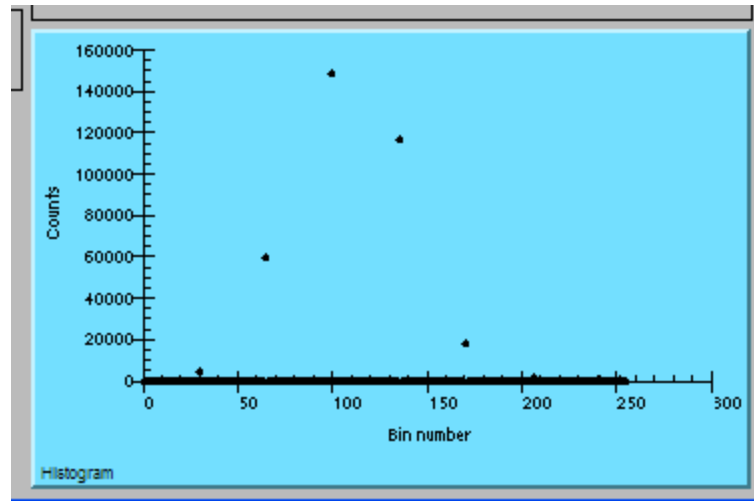
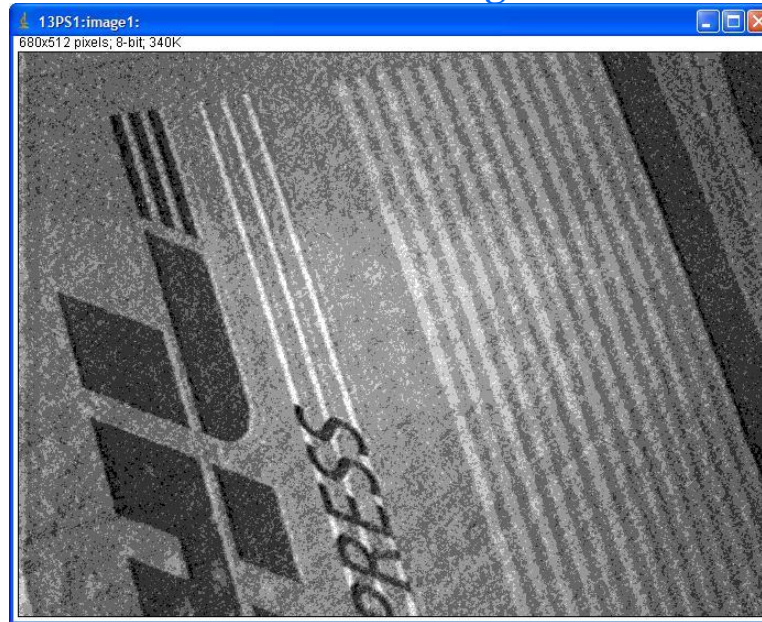
N = value of NumFiltered

0 = Output array passed to clients

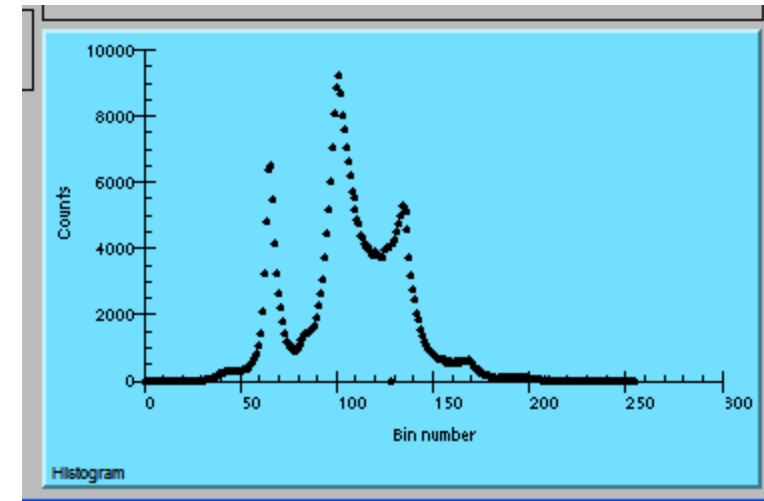
Processing plugin

30 microsec exposure time

No filtering



N=100 recursive average filter



Transform plugin

NDTransform.adl

13SIM1:Trans1:

asyn port TRANS1

Plugin type NDPluginTransform

Array port OVER1 OVER1

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 18

Array rate 0.0

Dropped arrays 0 0

dimensions 2

Array Size 512 512 0

Data type UInt8

Color mode Mono

Bayer pattern RGGB

Unique ID 18

Time stamp 778273003.649

Attributes file

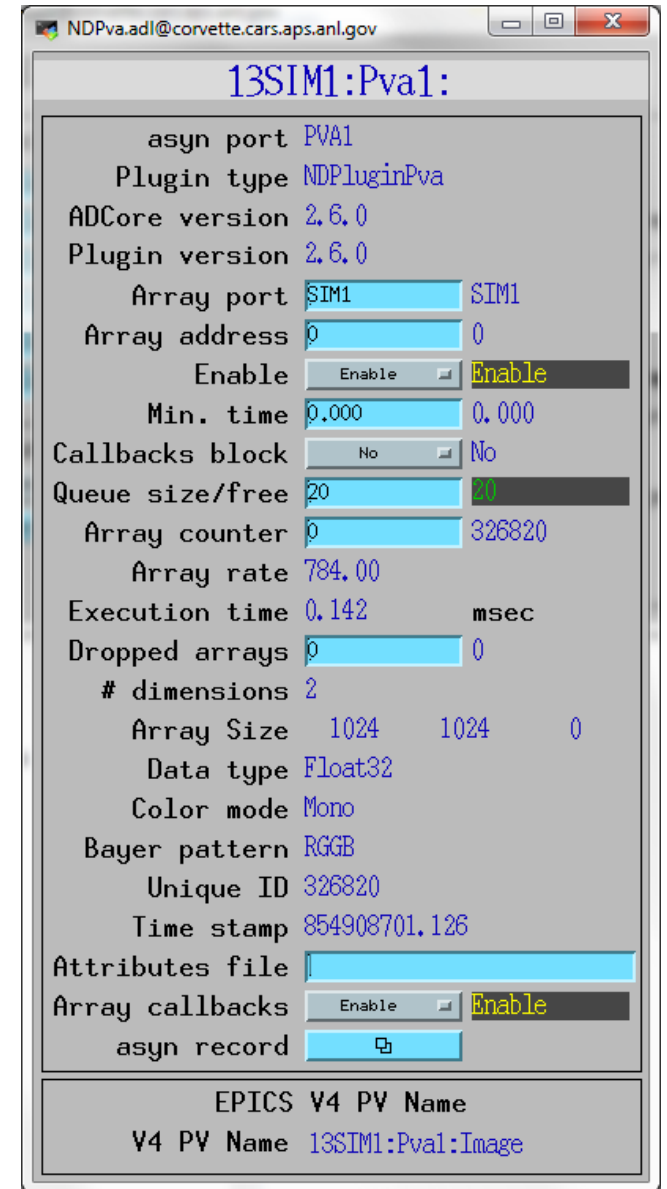
asyn record

Transform Type Rot90Mirror

None	F	Mirror	F
Rot90	F	Rot90Mirror	F
Rot180	F	Rot180Mirror	F
Rot270	F	Rot270Mirror	F

NDPluginPva (EPICS 7)

- Converts NDArrays into the EPICS PVA normative type NTNDArray
- Embedded EPICSv4 server serves the new NTNDArray structure over PVAccess
- 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
 - ImageJ plugin
 - Can include an NTNDArray receiver in another IOC



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

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

Area Detector Control - 13PVA1:cam1:

Setup
asyn port PVA
EPICS name 13PVA1:cam1:
Manufacturer PVAccess driver
Model Basic PVAccess 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
Connected
Connection
Debugging ☐

Collect
Images 100 100
Images complete 105801
Image mode Continuous
Collecting
Acquire
Detector state Idle
Status
Image counter 105801
Image rate 303.00
Array callbacks 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

Readout

	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

pvaDriver
Overrun counter 42308
13SIM1:Pva1:Image
PV Name 13SIM1:Pva1:Image
PV Connection Up

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

- NDFileTIFF
 - Supports any NDArray data type
 - Stores NDAttributes as ASCII user tags
- NDFileJPEG
 - With compression control
 - No NDAttributes
- NDFileNetCDF
 - Popular self-describing binary format, supported by Unidata at UCAR
 - NDAttributes are written
- 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
 - Supports Single Writer Multiple Reader (SWMR). Only supported on local file systems, GPFS, and Lustre (not NFS or SMB)

Plugins: NDPluginFile

- 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

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**

Array address **0**

Enable **Enable**

Min. time **0.000**

Callbacks block **No**

threads **1**

Max # threads **5**

Queue size/free **200**

Sort mode **Sorted**

Sort time **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 **RGGB**

Unique ID **319525**

Time stamp **858340043.780**

Attributes file **StatsAttributes.xml**

Array callbacks **Enable**

Process plugin **Process**

asyn record

corvette (epics)

Terminal Sessions View Xserver Tools Games Settings Macros Help

6. corvette (e x) 14. corvette (x) 8. corvette (e x) 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

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**

Enable **Enable**

Min. time **0.000** **0.000**

Callbacks block **No** **No**

threads **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 **RGGB**

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

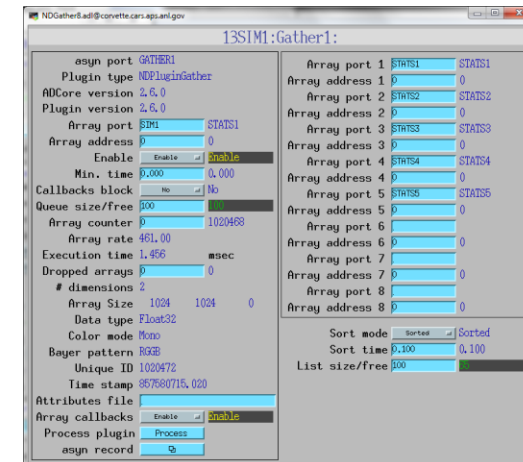
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

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

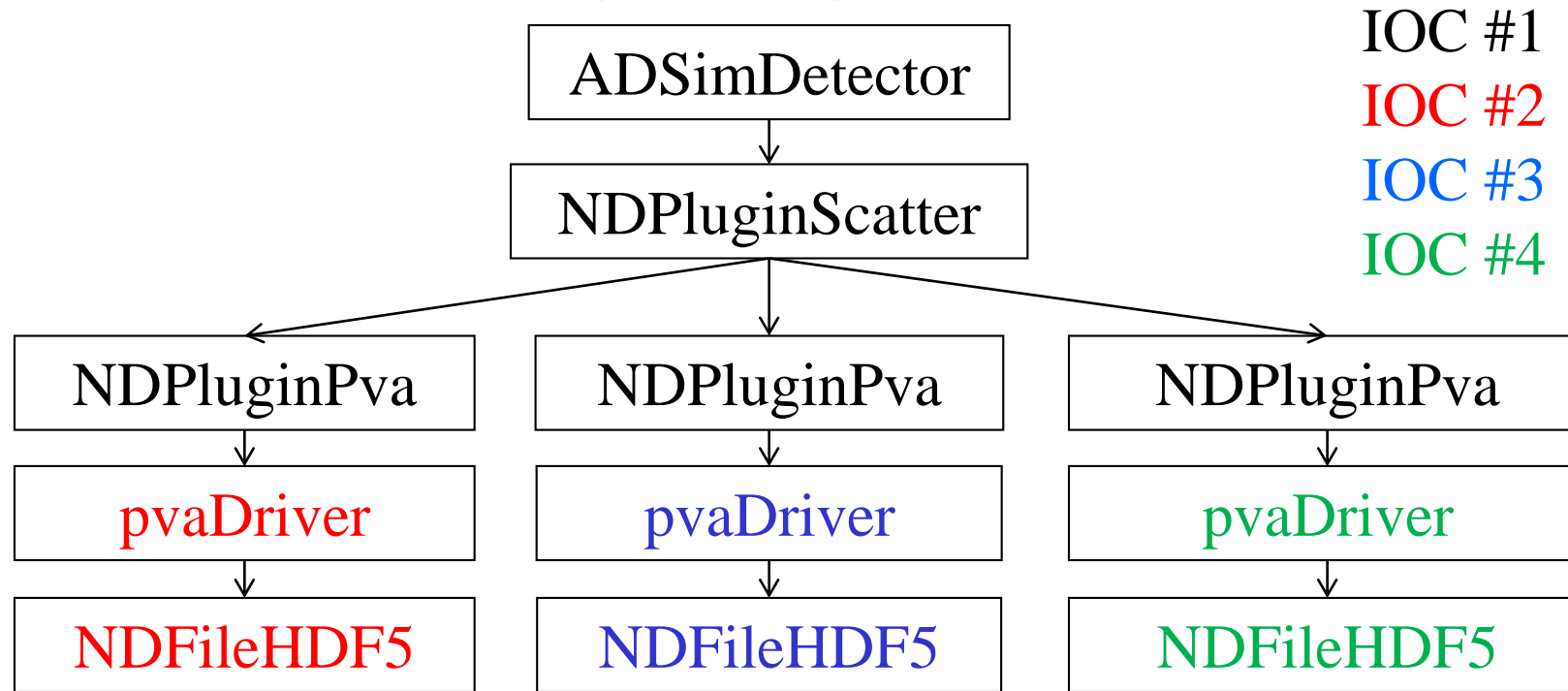
NDPluginGather

- 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)



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

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_NTNDATA_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

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.

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
 - Dante from XGLab
- 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
- Base classes, `asynPortDriver`, `asynNDArrayDriver`, `asynPluginDriver` actually are generic, nothing “areaDetector” specific about them.
- They can be used to implement any N-dimension detector, e.g. the XIA xMAP (16 detectors x 2048 channels x 512 points in a scan line)
- Documentation
 - <https://areaDetector.github.io>
- Can get code from github
 - <https://github.com/areaDetector>