

# Channel Access, CaSnooper, and CASW

Kenneth Evans, Jr.

Presented

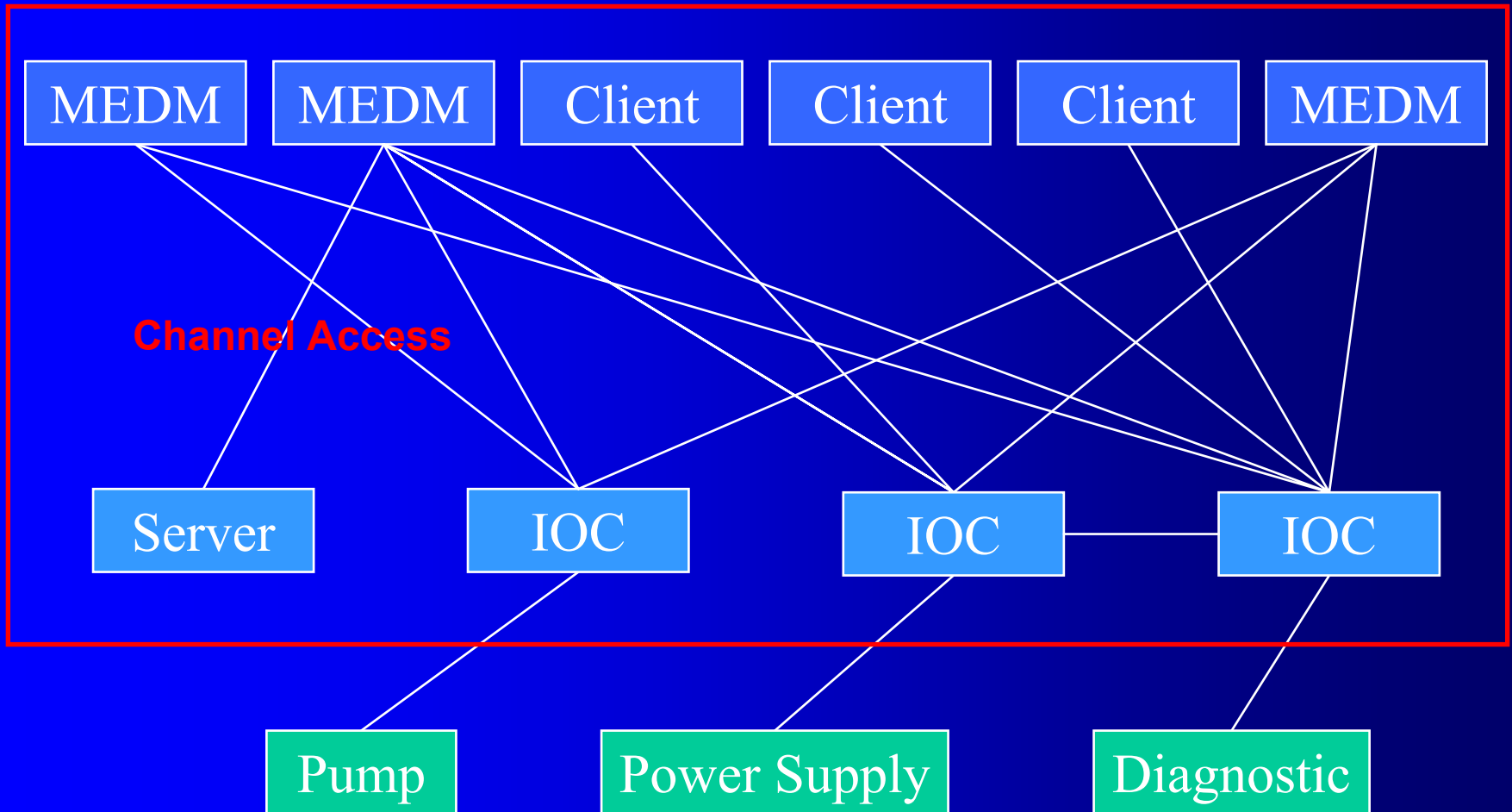
November 7, 2003

Argonne National Laboratory

# Topics Covered

- Channel Access
  - The means by which EPICS Clients and Servers talk to each other
- CaSnooper
  - An application to monitor Search Requests for Process variables
- CASW
  - An application to monitor Beacon Anomalies
- RunCaSnooper
  - An application that provides an interface to CaSnooper and CASW with an associated MEDM and StripTool

# EPICS Overview



# Concepts

- Network Protocols
- Process Variable Connection Process
- Search Request
- Exist Test
- Beacons
- Beacon Anomaly
- CaRepeater


# Network Protocols

- Channel Access uses two Network Protocols, UDP and TCP
- UDP (User Datagram Protocol)
  - One way, unreliable
  - Send out packets, no guarantee they reach their destination
  - Can be broadcast or directed
    - Broadcasts: To all IP addresses, e.g. 164.54.11.255
    - Directed: To a specific IP address, e.g. 164.54.8.167
  - Broadcasts do not leave subnets for security reasons
- TCP (Transmission Control Protocol)
  - Two way, reliable, persistent
  - Socket at each end
  - Acknowledgements, timeouts, retransmissions, etc. guarantee reliability

# Connection Process

- Client, like MEDM, wanting a PV sends a UDP search request
  - Sent to EPICS\_CA\_ADDR\_LIST
  - (Or its default -- broadcast to all interfaces on the host machine)
  - Sent on EPICS\_CA\_SERVER\_PORT [5064]
  - Do you have this PV?
- Each Server that gets a packet does an exist test
  - Do I have this PV?
- Server with the PV sends a directed UDP reply to the Client
  - I have this PV.
- A TCP connection is established between the Server and the Client (or an existing one is used)
  - One per Client-Server pair, no matter how many PVs
  - Let's talk.

# Search Request


- A Client makes a search request when it wants to find out what Server has the PV
  - Happens when a PV is first created in the Client
  - On a beacon anomaly (still unresolved PVs only)
  - When another PV is created (still unresolved PVs only)
- A search request consists of a sequence of UDP packets
  - Starts with a small interval (30 ms), that doubles each time
  - Until it gets larger than 5 s, then it stays at 5 s
  - Stops after 100 packets or when it gets a response
  - Never tries again until it sees a beacon anomaly or creates a new PV
  - Total time is about 8 minutes to do all 100
  - The sequence may be different owing to fine tuning
- Usually connects on the first packet or the first few


# Exist Test

- Every time a Server receives a search request packet, its `pvExistTest` routine is called
- The Server has to check if it has the PV
  - Returns `ExistsHere` or `DoesNotExistHere`
- Normally a search request sequence ends after a few packets
  - Because one Server soon returns `ExistsHere`
- For PVs that do not exist
  - There are 100 tests per search request sequence for that PV
  - This happens every time a Client initiates a search request sequence
    - Each time the Client searches for a new PV
    - At each beacon anomaly, perceived or real



# Beacons

- A Beacon is a UDP broadcast packet sent by a Server
- When it is healthy, each Server broadcasts a UDP beacon at regular intervals (like a heartbeat)
  - EPICS\_CA\_BEACON\_PERIOD, 15 s by defaultA horizontal line with five yellow dots spaced evenly along it, representing regular beacon intervals.

—●—●—●—●—●—
- When it is coming up, each Server broadcasts a startup sequence of UDP beacons
  - Starts with a small interval (25 ms, 75 ms for VxWorks)
  - Interval doubles each time
  - Until it gets larger than 15 s, then it stays at 15 s
    - Takes about 10 beacons and 40 s to get to steady stateA horizontal line with seven yellow dots. The first three dots are very close together, and the remaining four dots are spaced evenly along the line, representing a startup sequence that transitions to a steady state.

●●●—●—●—●—●—●—
- Beacons go to CaRepeater, which forwards them to the Clients
  - On EPICS\_CA\_BEACON\_PORT [5065]
- Clients monitor the beacons for frequency changes (Beacon Anomalies)

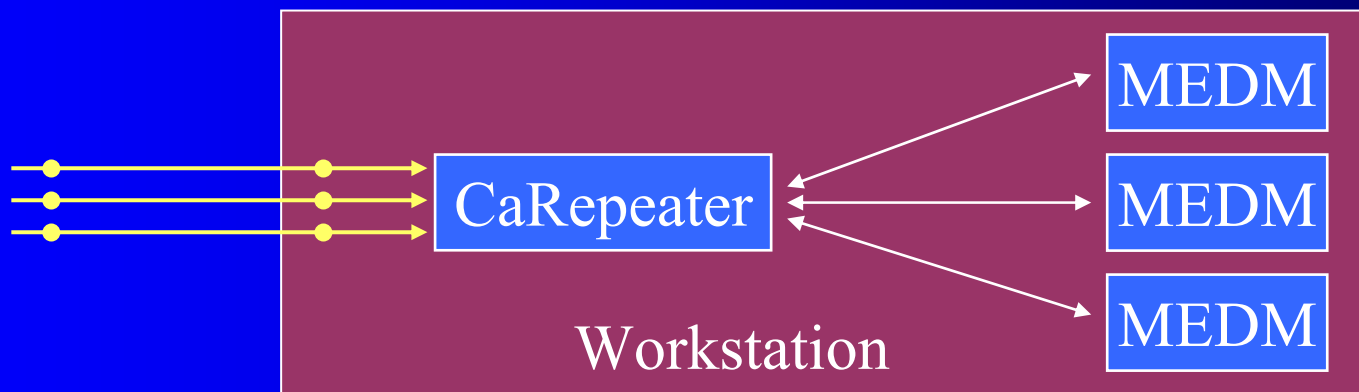
# Beacon Anomaly

- A Beacon Anomaly is any change from the normal beacon interval (15 s)
- No beacons:
  - After 30 sec the Client sends message over TCP connection
  - If no beacons and no reply, connection is down
  - That is when MEDM screens go white
- Abnormal interval:
  - Short: IOC has come up
  - Long: IOC was disconnected
- All cause Clients to reissue outstanding search requests
- Network problems can look like beacon anomalies



# CaRepeater

- UDP broadcasts are not guaranteed to go to every process on a workstation
- CaRepeater solves this problem
  - There is one CaRepeater process per workstation
  - Clients make a TCP connection to it when they start up
  - CaRepeater receives the beacons
    - EPICS\_CA\_REPEATER\_PORT [usually 5065]
  - CaRepeater forwards the beacons to the Client



# Summary

- Servers send beacons at regular intervals and with a faster pattern when they come up
- A Beacon anomaly is any pattern that is not a regular beacon
- Beacon anomalies cause Clients to send search requests
- Search request sequences end early for found PVs but not for non-existent PVs
- Search requests put a load on the Servers and add to network traffic
- This can cause problems
- Consequently, undesirable beacon anomalies and search requests should be minimized or eliminated

# CaSnooper

- CaSnooper is a Server whose ExistTest routine keeps track of Search Requests rather than seeing if it has the PV
- It can print the names of all PVs being searched for and related statistics using several report formats
- It can also check if these PVs are connected (C) or not (NC)
- It has internal PVs if started with the -n option
  - Overall rates that can be monitored
  - Others that allow it to be controlled from an MEDM screen
  - The PV prefix [default CaSnoop] can be changed to prevent collisions
- Running CaSnooper:
  - Run at the command line to get one report
  - Run with PVs for monitoring, say with SDDSmonitor or StripTool
  - Run with PVs and control with MEDM for continuous operation
- To run CaSnooper you need the full path
  - /usr/local/epics/extensions/bin/solaris-sparc/caSnooper
  - It is not installed for 3.13 (3.13 Servers have less capability)

# CaSnooper Options

```
XTerm
46 krypton:EVANS>/usr/local/epics/extensions/bin/solaris-sparc/caSnooper -h
CaSnooper 2.1.0.1 (8-27-2003) EPICS 3.14.3
Usage: caSnooper [options]
Options:
  -c<integer> Check validity of top n requests (0 means all)
  -d<integer> Set debug level to n
  -h          Help (This message)
  -i<string>  Specify a PV name to watch individually
  -l<decimal> Print all requests over n Hz
  -p<integer> Print top n requests (0 means all)
  -n[<string>] Make internal PV names available
                Use string as prefix for internal PV names
                (10 chars max length) Default string is: CaSnoop
  -s<integer> Print all requests over n sigma
  -t<decimal> Run n seconds, then print report
  -w<decimal> Wait n sec before collecting data

47 krypton:EVANS>█
```

# Sample CaSnooper Output

Two lines from RunCaSnooper →

Print top 10 (-p10) →

Check top 10 (-c10) →

```
CaSnooper
Starting CcSnooper...
Type Ctrl-C to stop it
Starting CaSnooper 2.1.0.1 (8-27-2003) at Nov 04 13:50:08
EPICS 3.14.3
Individual Name is CaSnoop.test
PV name prefix is CaSnoop

Nov 04 13:50:42:
There were 17508 requests to check for PV existence for 2307 different PVs.
Max(Hz): 11.17
Mean(Hz): 0.22
StDev(Hz): 0.61

PVs with top 10 requests:
1 willow:52275 FEL:$(M8).VAL 11.17
2 willow:52275 FEL:$(M8).RBV 8.94
3 gateway433:33790 FE:09:ID:SR:HPOS:CC.VAL 4.15
4 gateway433:33790 FE:09:ID:SR:VPOS:CC.VAL 4.15
5 willow:52275 FEL:VUV4f2_able.VAL 3.35
6 gateway435:33270 s17id:scan1.NPTS 3.06
7 gateway435:33270 s17id:scan1.MPTS 3.06
8 gateway435:33270 s17id:scan1.PASM 3.06
9 gateway435:33270 s17id:scan1.P1EP 3.06
10 gateway435:33270 s17id:scan1.P1AR 3.06

Nov 04 13:51:01:
There were 23868 requests to check for PV existence for 2329 different PVs.
Max(Hz): 10.45
Mean(Hz): 0.19
StDev(Hz): 0.56

Connection status for top 10 PVs after 10.00 sec:
1 willow:52275 FEL:$(M8).VAL NC 10.45
2 willow:52275 FEL:$(M8).RBV NC 8.36
3 gateway433:33790 FE:09:ID:SR:HPOS:CC.VAL NC 4.01
4 gateway433:33790 FE:09:ID:SR:VPOS:CC.VAL NC 4.01
5 willow:52275 FEL:VUV4f2_able.VAL NC 3.13
6 gateway435:33270 s17id:scan1.NPTS NC 3.04
7 gateway435:33270 s17id:scan1.P1PV NC 3.04
8 gateway435:33270 s17id:scan1.R1PV NC 3.04
9 gateway435:33270 s17id:scan1.CMND NC 3.04
10 gateway435:33270 s17id:scan1.P1EP NC 3.04
```

individual name, prefix

statistics

machine:port, (can be used to identify source)

name

search rate in Hz

Not connected, will be C for connected (hardly ever the case)

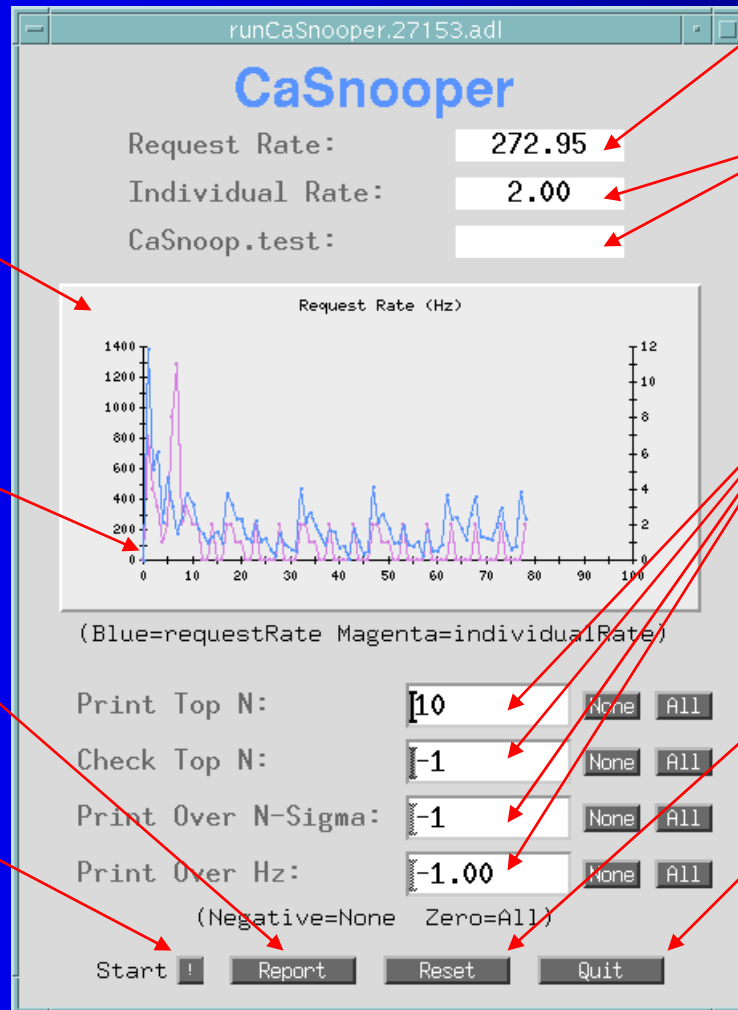
# Control CaSnooper via MEDM

Cartesian plot of requestRate and individualRate

CaSnooper was started here (with EPICS\_CA\_REPEATER\_PORT = default=5065)

Execute selected reports in the CaSnooper stdout

Shell command to start CaSnooper, MEDM, StripTool, etc.



Request rate

Individual rate for CaSnoop.test, which doesn't exist

Use these to set what will happen when you press Report. Case illustrated will print the top 10.

Reset the counters in CaSnooper

Stop CaSnooper



# CASW

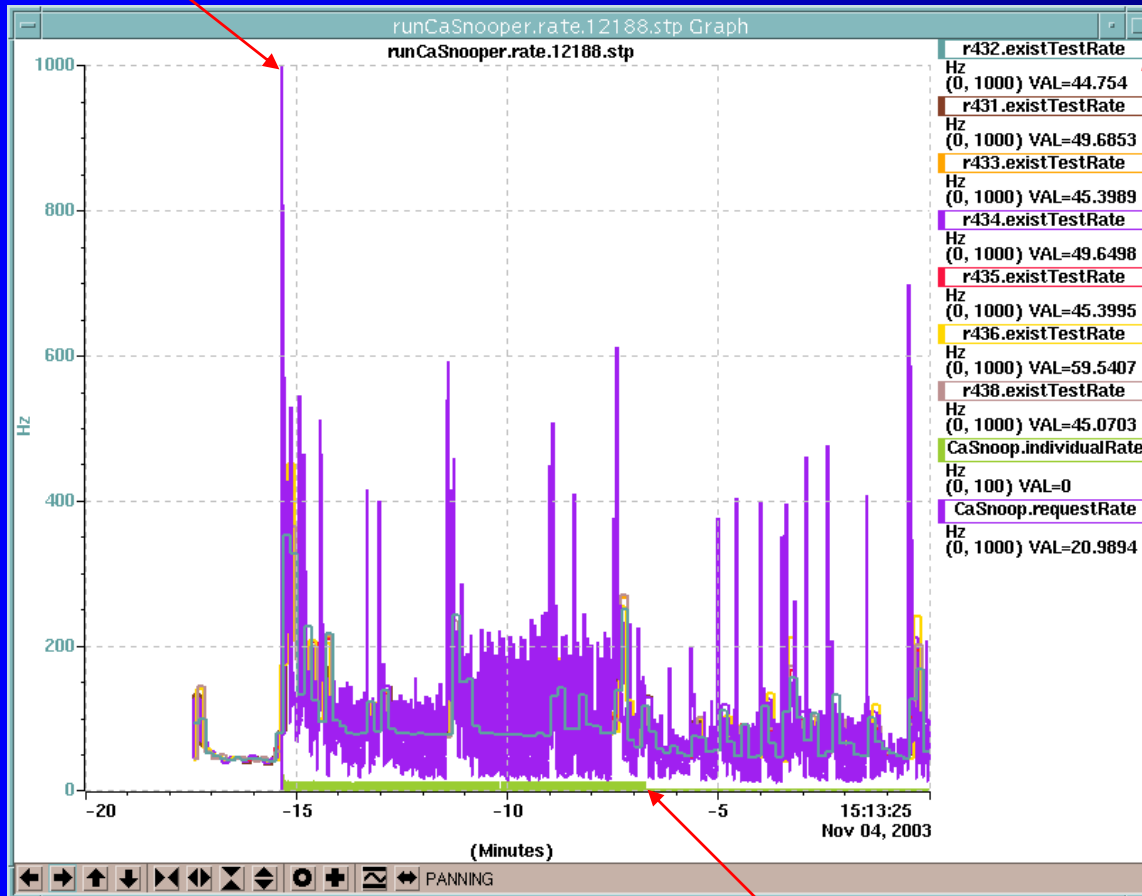
- CASW (Channel Access Server Watcher) monitors Beacon Anomalies
- Is a simple command-line utility
- Part of EPICS Base
- Need a full path to the version of base desired
  - 3.14 (Recommended)  
/usr/local/epics/base3.14.3/bin/solaris-sparc/casw
  - 3.13:  
/usr/local/epics/base/bin/solaris/casw
- Prints a line when it sees a beacon anomaly

CaSnooper Starting

```
Starting CASW...
Type Ctrl-C to stop it
There will no output until a beacon anomaly occurs
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:50:39.322522701
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:50:44.331146567
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:50:44.331729436
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:50:49.337082255
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:50:49.338022943
chiron:5064                    2003-11-04 11:50:52.285177497
chiron:5064                    2003-11-04 11:50:52.299845797
chiron:5064                    2003-11-04 11:50:52.320118219
chiron:5064                    2003-11-04 11:50:52.348091798
chiron:5064                    2003-11-04 11:50:52.408653298
chiron:5064                    2003-11-04 11:50:52.536166793
chiron:5064                    2003-11-04 11:50:52.788118789
chiron:5064                    2003-11-04 11:50:53.299866476
chiron:5064                    2003-11-04 11:50:54.321393444
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:50:54.344781889
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:50:54.345192506
chiron:5064                    2003-11-04 11:50:56.369893513
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:50:59.352993929
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:50:59.353456538
chiron:5064                    2003-11-04 11:51:00.466606322
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:04.357752360
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:04.359234565
chiron:5064                    2003-11-04 11:51:08.660414353
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:09.366160712
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:14.373282812
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:19.380480042
chiron:5064                    2003-11-04 11:51:23.660740900
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:24.387719304
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:29.397511716
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:34.401504995
chiron:5064                    2003-11-04 11:51:38.661120641
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:39.410394708
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:44.416328586
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:49.424073457
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:54.432394468
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:51:59.438268469
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:52:04.445217601
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:52:09.457422947
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:52:14.460862062
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:52:19.467448738
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:52:24.471852165
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:52:29.473472890
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:52:34.478712317
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:52:39.484049642
pcdiag6.aps4.anl.gov:5064      2003-11-04 11:52:44.489705934
```

# StripTool

(CaSnooper was started here)



Reverse Gateways  
(1 min update)

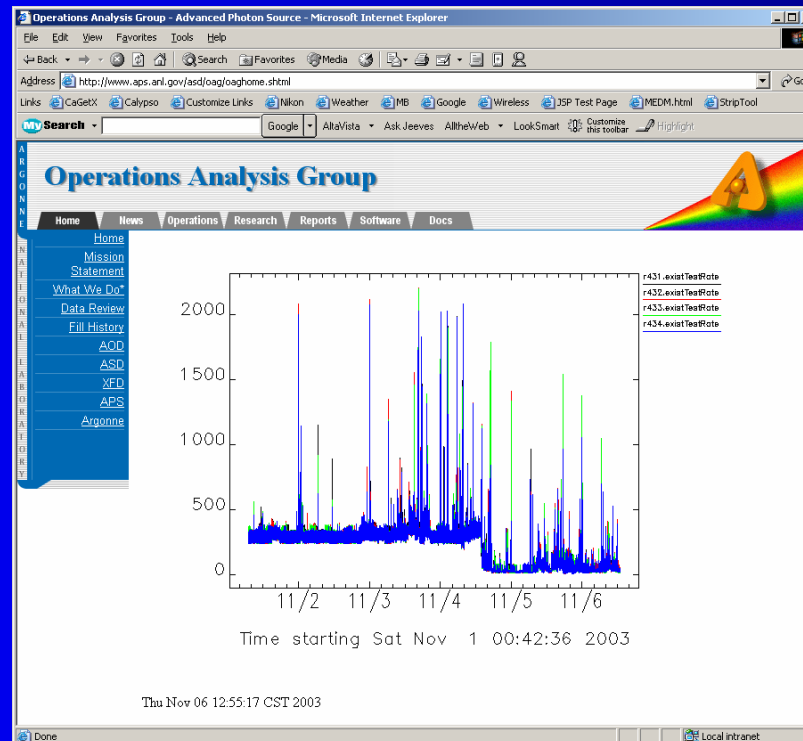
CaSnooper  
(1 sec update)

(Reverse Gateways,  
CaSnooper, and all IOCs  
see the same Search  
Requests)

Search for individual PV  
ended after about 8  
minutes

# OAG Monitoring

- The search request rates from the Reverse Gateways on the machine subnet are being continuously monitored
- You can access the history from <http://www.aps.anl.gov/asd/oag/logging/MonitorDataReview.html>



# RunCaSnooper

- RunCaSnooper provides an interface to CaSnooper and CASW with an associated MEDM and StripTool
- Is part of IocappsTools (Type iocHelp for a list)
- By default it brings up an MEDM and a StripTool and does not start CaSnooper
  - If CaSnooper is running, the MEDM screen will not be white
  - If it is white, you can start CaSnooper from the MEDM screen
  - CaSnooper uses EPICS\_CA\_REPEATER\_PORT=9876 by default, not 5065
  - You can start also start CASW and StripTool from the MEDM screen
- Everything is generated on the fly and stored in /tmp
- Look in /tmp for:
  - Logs of the CaSnooper reports and CASW output
  - MEDM ADL file and StripTool configuration files
- Start it (type runCaSnooper) with no options:
  - Displays extensive directions
  - Then optionally allows you to start it (type y to continue)

# RunCaSnooper

The screenshot displays the RunCaSnooper software interface with several windows open:

- CaSnooper**: Shows startup logs including "Starting CaSnooper 2.1.0.1 (8-27-2003) at Nov 04 16:36:03" and "Individual Name is CaSnoop.test".
- CASW**: Shows logs for "Starting CASW..." and lists PVs with top 10 values, such as "chiron:5064" and "gateway:433".
- runCaSnooper.rate.12188.stp**: A graph showing request rates in Hz over time (minutes). The y-axis ranges from 0 to 1000 Hz, and the x-axis ranges from -20 to 16:47:39 on Nov 04, 2003. Multiple colored lines represent different test rates.
- runCaSnooper.12188.adl**: A summary window titled "CaSnooper" showing:
  - Request Rate: 54.01
  - Individual Rate: 0.00
  - CaSnoop.test: [empty]It also includes a smaller graph of "Request Rate (Hz)" and control options like "Check Top N: 1", "Print Top N: 10", and "Print Over N-Sigma: 1".

# RunCaSnooper Options

```
XTerm
48 krypton:EVANS>runCaSnooper -h

RunCaSnooper: Provides an interface to CaSnooper and CASW with an
associated MEDM and StripTool

Usage:
runCaSnooper [Options]
Options:
  -h          Help
  -d          Use no defaults, only the switches you enter
  +d         Use default setup without printing help
  -m         Do not start MEDM
  +m         Start MEDM [Default]
  -s         Do not start StripTool
  +s         Start StripTool [Default]
  -c         Do not start CaSnooper [Default]
  +c         Start CaSnooper
  -crp <int> Set EPICS_CA_REPEATER_PORT for CaSnooper [Default is 5064]
              Use a number greater than 5000, for example 6666
  -w         Do not start CASW [Default]
  +w         Start CASW
  -p <string> Specify a prefix for CaSnooper process variables
              [Default is CaSnoop]
  -i <string> Specify an individual name for CaSnooper
              [Default is CaSnoop.test]
  -clean     Remove caSnooper./* files created by runCaSnooper in /tmp
              (This will also remove any log files created !)
```

# References

- Jeffery O. Hill, *EPICS R3.14 Channel Access Reference Manual*, (EPICS Documentation, 2003 or latest).  
<http://www.aps.anl.gov/epics/modules/base/R3-14/index.php>
- W. Richard Stevens, *UNIX Network Programming*, (Prentice-Hall, Upper Saddle River, NJ, 1998) Vol. 1.
- K. Evans, Jr., *CaSnooper Reference Manual*, (EPICS Documentation, 2003 or latest).  
<http://www.aps.anl.gov/epics/extensions/caSnooper/index.php>

# Thank You

This has been an  
APS Controls Presentation



# Thank You

This has been an  
APS Controls Presentation