



Online Monitoring for the CDF Run II Experiment

T.Arisawa, D.Hirschbuehl, K.Ikado, K.Maeshima,
H.Stadie, G.Veramendi, W.Wagner, H.Wenzel,
M.Worcester

MAR 24-28, 2003

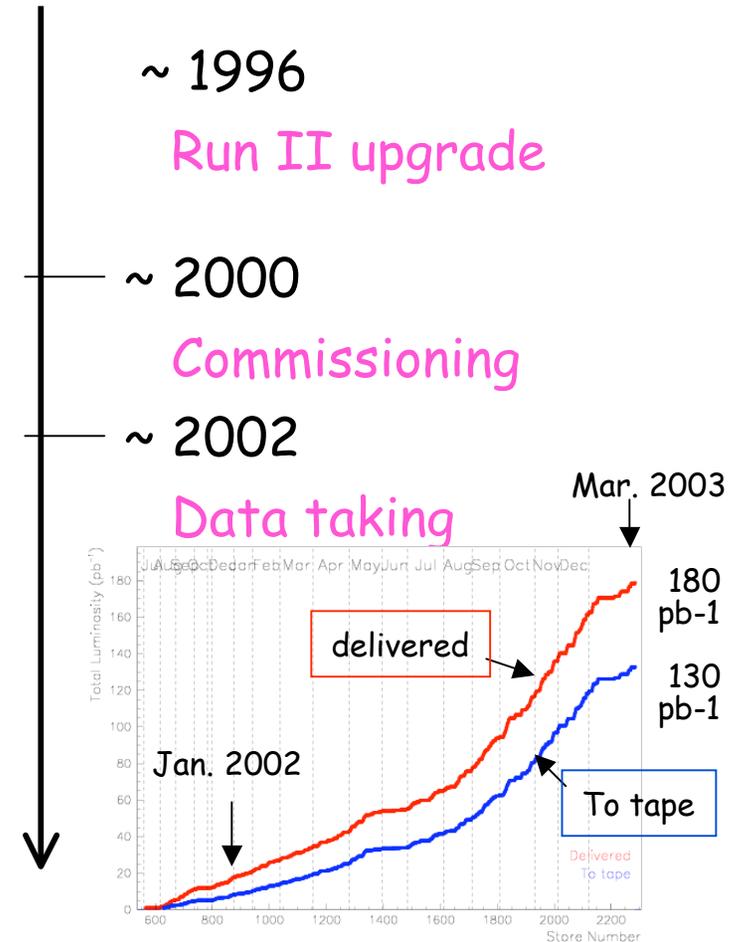
CHEP03



Brief History of Online Monitor

1. **Design Stage**
(May 1998 -)
Worked closely with Root development team.
2. **Implementation/Commissioning**
(summer 1998 -)
Monitors were used extensively to commission trigger/detector.
3. **Operation/Improvements**
(2000 -)
Integral part of the CDF DAQ to keep taking high quality data.

CDF Run II Time Line



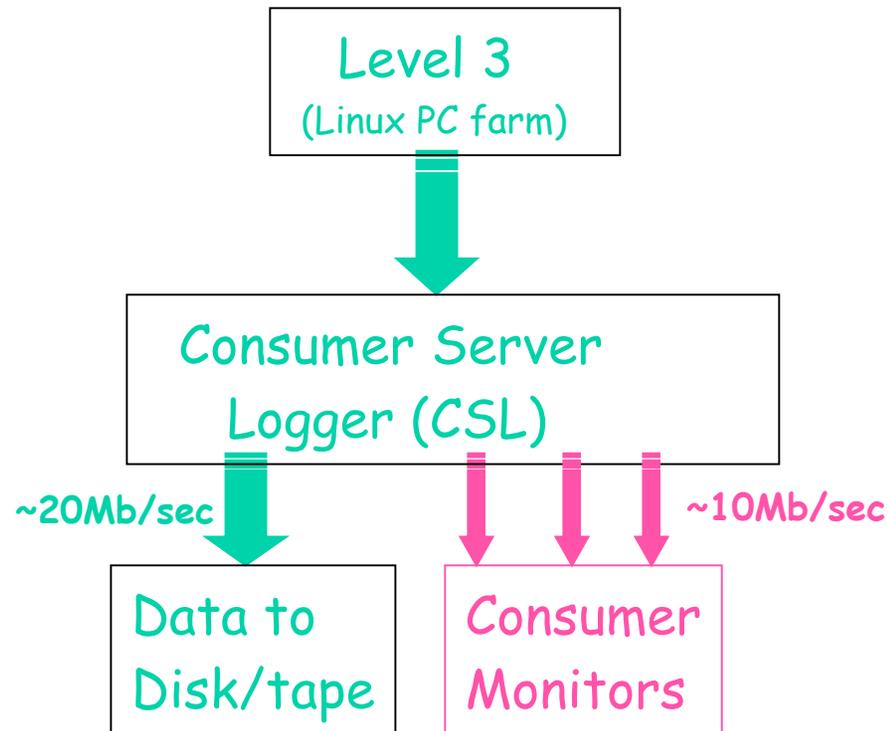


Design Goals

- Monitor the DAQ/trigger/detector performance without interfering with the data taking.
- Monitored results can be interpreted by the shift crew fast and clear in order to maintain high quality data taking.
- Different consumer processes can run on different machines (expandability).
- Each consumer receives only the data it needs (choice of triggers)
- The monitoring and the display processes are separated. The number of displays is limited only by network traffic and bandwidth.
- Different consumers can be combined to one executable.
- Common interface and maintainability.
- Framework is decoupled from CDF software specifics. (SPYMon, MINUS, etc...)



Data Flow to the Monitors



1. Level 3

- Final level of event filtering

2. Consumer Server/Logger

- Writes events passing Level 3 to disk.
- Serves events to consumers.

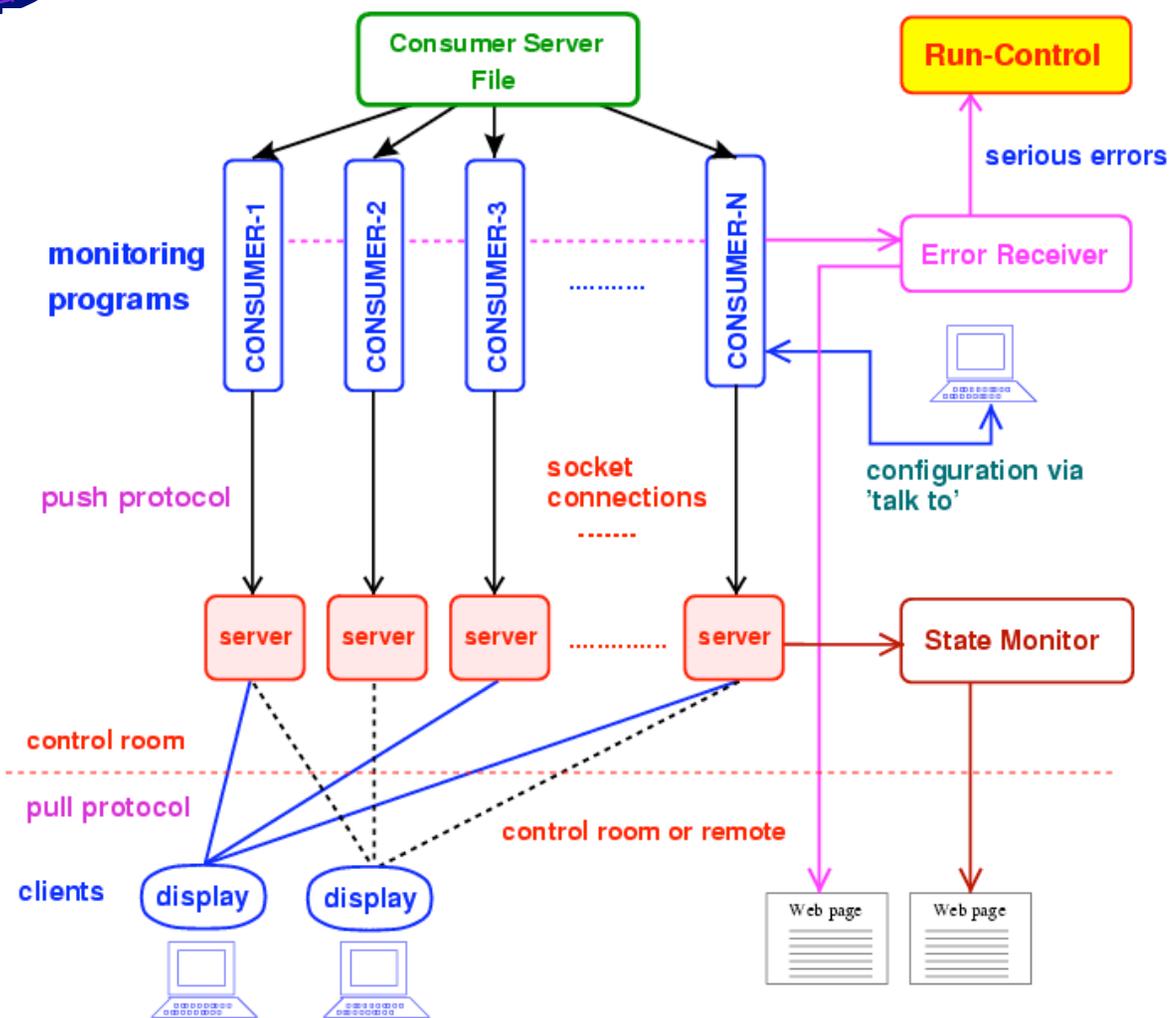
3. Rates/Sizes

- ~ 250 kb/events
- Logging rates: ~ 20 Mb/sec
- ~10Mb/sec to consumers.
- Typically 11 monitors running

Consumer: a process which receives event data from CSL in real time.



Consumer Framework Components



- Consumers
- Display Server
- Display Client
- Error Receiver
- State Monitor

C++ using ROOT



Framework Components

I. The Consumers

- Analyze and monitor the event data
- Use CDF Run II offline framework (AC++) to look at data*
- Consists of different AC++ modules:
 - APPConsumerInputModule
 - ConsumerErrorModule (adds special destination to the ZOOM error logger to send errors to the ConsumerErrorReceiver)
 - Module inheriting from ConsumerFrameworkModule consists of different monitors that are written by the experts. All these monitors inherit from BaseMonitor2 which starts server process at the beginning of a job.

* In order to look at CDF event data, a decision was made very early on to use Cdf offline framework for the consumers. However, the consumer framework is basically free from Cdf offline specifics and the entire package can be used easily in different settings.



Framework Components (cont.)

BaseMonitor2

- Base class for all monitors
- Provides "framework" functions and functions to be overwritten by the monitor writers.

TConsumerInfo

- is sent from the consumer to server
- is sent from the server to the display
- contains information about the consumer
 - name
 - run number
 - number of events processed
 - list of all ROOT objects that are available



Framework Components (cont.)

II. The Server

- receives ROOT objects from the consumer via socket
- deals with requests from the displays
- reports the status of the consumer to the state manager

III. The Display

- ROOT-based GUI
- can connect to the server via socket or ROOT files, can browse Root objects from the consumers
- at first requests TConsumerInfo and creates a list tree
- only updates objects on canvas, does not redraw whole canvas
- useful features like; auto update, slide show, pop-up warning window, etc...



Framework Components (cont.)

The StateManager

Collects consumer status/info and via Display Server and displays the info. on a web page.

List of Servers

Status of Consumerserver Processes:

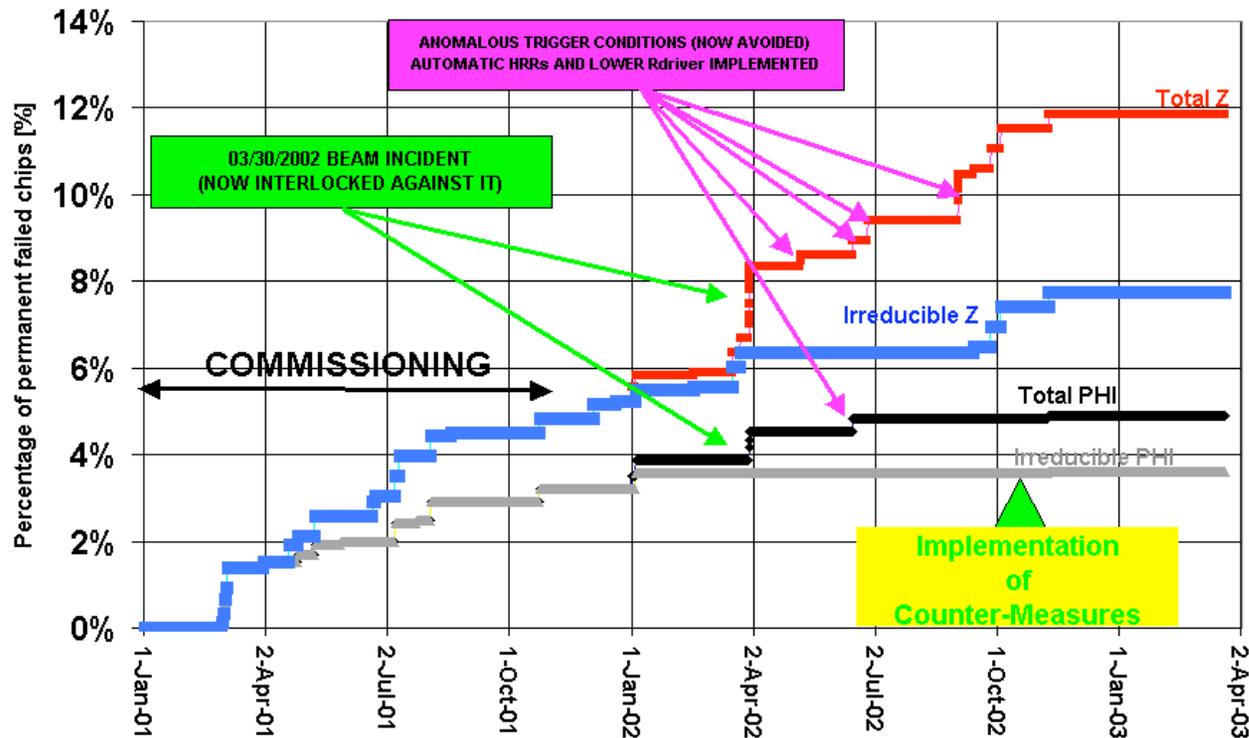
Consumer	Hostname	Port	Run number	# of Events processed	Server Status	Updated	Time
BeamMon	b0dap66.fnal.gov	9091	160488	115350	Running	Y	Sat Mar 22 03:22:14 2003
DAQMon	b0dap76.fnal.gov	9091	160488	164890	Running	Y	Sat Mar 22 03:21:59 2003
LumMon	b0dap67.fnal.gov	9091	160488	127490	Running	Y	Sat Mar 22 03:22:30 2003
ObjectMon	b0dap56.fnal.gov	9091	160488	116600	Running	Y	Sat Mar 22 03:22:35 2003
SVTSPYMON	b0dap75.fnal.gov	9091	160488	439293	Running	Y	Sat Mar 22 03:21:36 2003
SVXMon	b0dap66.fnal.gov	9092	160488	1485	Running	Y	Sat Mar 22 03:21:59 2003
SilliMon	b0dap65.fnal.gov	9091	160488	9020	Running	Y	Sat Mar 22 03:22:20 2003
Stage0_	b0dap77.fnal.gov	9092	160488	129720	Running	Y	Sat Mar 22 03:22:15 2003
TrigMon	b0dap67.fnal.gov	9092	160488	2630	Running	Y	Sat Mar 22 03:22:46 2003
XMon	b0dap77.fnal.gov	9091	160488	19250	Running	Y	Sat Mar 22 03:22:14 2003
YMon	b0dap76.fnal.gov	9092	160488	10050	Running	Y	Sat Mar 22 03:22:36 2003

Last Update: Sat Mar 22 03:22:46 2003



Fast Feedback on Error Condition

SVXII: time evolution of unrecoverable failures



In beginning of run II, unexpected unrecoverable failures were observed in silicon detector. Implementation of Counter-measures including "auto-reset" from run_control triggered by SVXMon (a consumer monitor) detecting the "danger state" helped to reduce the failures substantially.



In the CDF Control Room



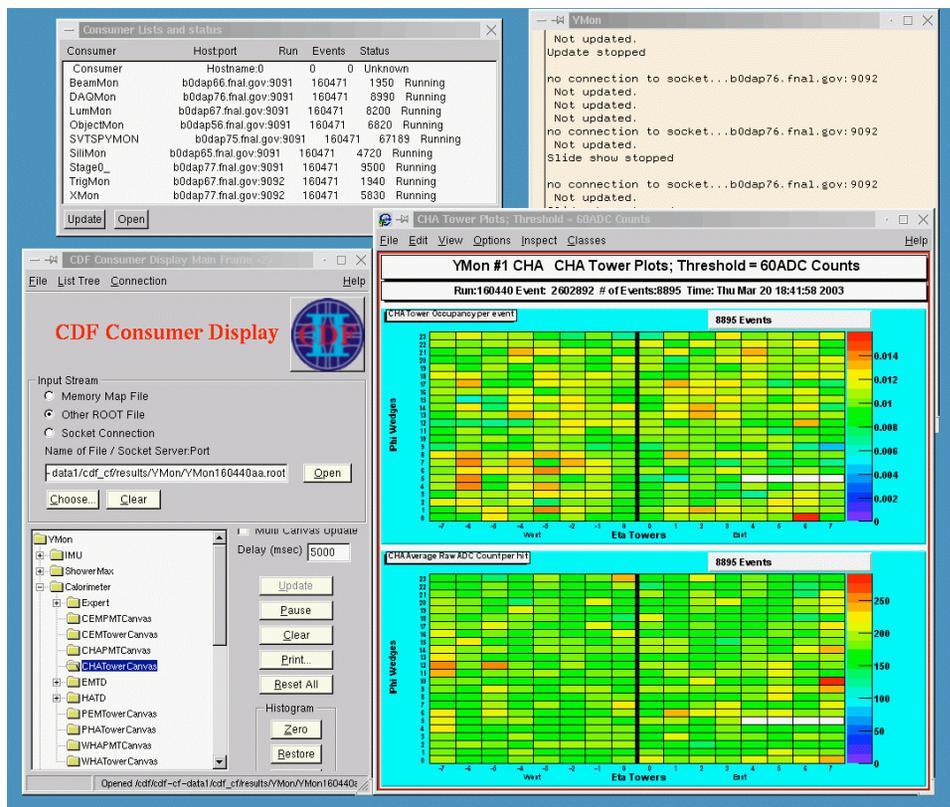
Consumer Framework people



Running Consumers

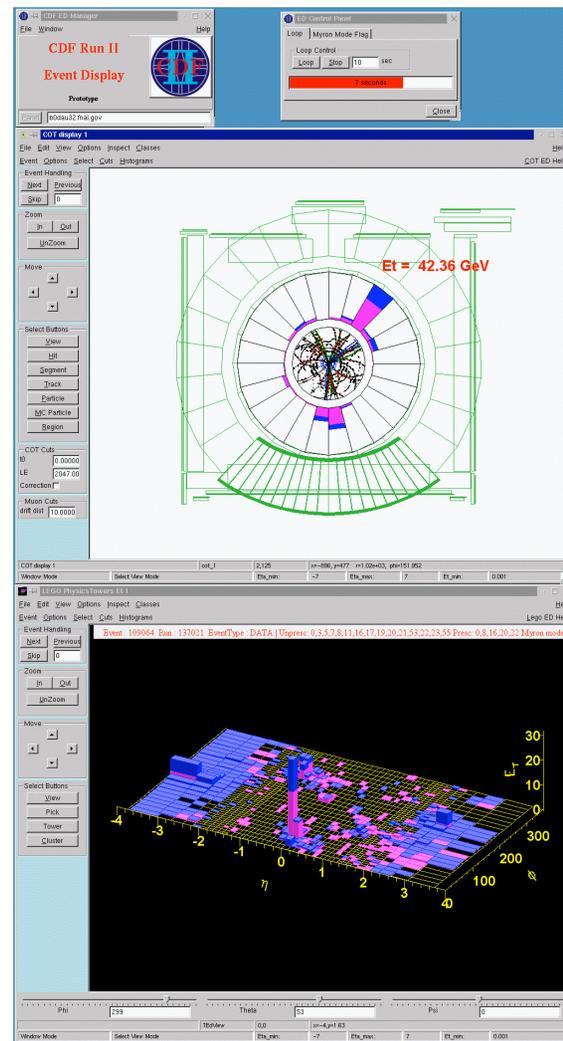


Examples of Monitor Display



Occupancy Monitor

Event Display





Conclusions

- Online consumer monitoring is an integral and essential part of the CDF data taking operation.
- The monitoring output is coherent, and one person on shift (not an expert for a specific detector) monitors the entire detector performance from event data, efficiently.
- The output is available to view remotely.
- We continue to fine-tune and improve each monitor, however, the monitoring framework has been very stable for the past year.