narval

Contents

1 Introduction **1**

2 Presentations **1**
2.1 Advanced Gamma Tracking Array 1
2.2 Orsay Tandem 1
2.3 Scanning tabble for detector characterisation 1
2.4 StratusLab project 2

3 GRU: Ganil Root Utility **2**
3.1 Ganil DAQ 2
3.2 Root 2
3.3 GRU Core 2
3.4 GRU client 2

4 Open hardware project **2**

5 Understanding Narval **2**
5.1 Vovabulary 2
5.2 Configuration files 3
5.3 Ada Actor's process 3
5.4 A three layers framework 3
5.5 Buffer API 4

1 Introduction

Historicaly from OASIS. Under GPL. Unique data link replaced by Data Flow concept (TCP, shared memory...)

```
$ svn co http://csngwinfo.in2p3.fr:2401/narval # login=passwd=narval
```

2 Presentations

2.1 Advanced Gamma Tracking Array

6660 electronic channels photon tracking. 2 kind of detectors (main + ancillary).

- Acquisition layer
- Configuration layer
- Permanant layer

2.2 Orsay Tandem

Located at paris-sud campus. VME crate:

- 5 aquisition cards with 6 channels
- 1 ressource manager card

Event building from:

- Tandem detector
- X-ray camera

Architecture:

- Shared libraries compiled outside Narval environment.
- XML configuration file
- CVISU: spectra visualisation (coeficients, thresholds, not based on root)

Carte cometh: 40Mb.s-1

2.3 Scanning tabble for detector characterisation

Scanning table for Agata (ENX+Narval). ViGru for spectrum monitoring

- Start/Stop
- Set and get somes parameters

2.4 StratusLab project

cloud we use for the formation.

3 GRU: Ganil Root Utility

GRU = ROOT + Ganil C++ libraries

3.1 Ganil DAQ

- Gloal Run Control Core
- Electronic Control Core

3.2 Root

Developped by René Brun. Root embedded CINT (C interpreter like shell).

3.3 GRU Core

http://wiki.ganil.fr/Documentation/Gru Online + Offline.

- Input device get buffer from diferents devices.
- Output spectra (histograms+graphs+data vectors) to TList Root)
- Soap server to get commands

3.4 GRU client

- Many sources (DB, HD, Soap: use as an osciloscop to debug electronic)
- Configuration via xml.
- Can fit curves
- Cutting curves into a new pad (new histogram store into the server updated online)

4 Open hardware project

CERN Open Hardware Licence. CERN, GSI, Soleil.

5 Understanding Narval

5.1 Vovabulary

Process:

- Producer
- Intermediary (like a switch n->m)
- Consumer
- Narval naming services (register sub systems)
- aws_shell / soap protocol / werewolf (GUI)
- central log
- coordinator (state machine)

5.2 Configuration files

Narval use polyorb to connect server following the topology.

- polyorb *narval.conf*
- topology *.xml*
- script (parralized orders for an automatic run)

5.3 Ada Actor's process

Ada actors (linked with narval sources) gives access to full Ada API:

- Inherit from Active_Actor_Type
- Override Buffer_handling (get/put)
- Override On_Start/On_Stop/... methods

Ada actor as shared library (only one generic actor type):

- Easy to compile
- Access to partial API
- Inherit from Actor_Interface
- Need to overide all the API

5.4 A three layers framework

```
$ narval_launch
$ ps -ef

nroche    1983     1  0 15:27 ?        00:00:00 SCREEN -m -d po_cos_naming --polyorb-iiop-polyorb-protocols-iio
nroche    1985     1  0 15:27 ?        00:00:00 SCREEN -m -d -c /etc/narval/screen_commands_for_narval
nroche    1986  1983  0 15:27 pts/2    00:00:00 po_cos_naming --polyorb-iiop-polyorb-protocols-iiop-default_por
nroche    1987  1985  0 15:27 pts/3    00:00:00 narval_naming_service
nroche    1988  1985  0 15:27 pts/4    00:00:00 central_log --log_level info
nroche    1989  1985  0 15:27 pts/5    00:00:00 aws_shell --aws_server_port 6080 --automatic-script-loading
```

- Permanent layer: naming services, aws shell, central log
  `Narval_Naming_Services` make Fat pointers for remote call interface. Exception are spread over the network.
  `aws_shell` is a bridge offering SOAP services (2 commands: send and send with arg).
  `central log` based on log4Ada and can be connected throuht log4j (java).
- Coordination layer: sub system 1,2...
  `sub_system_coordinator` embed the state machine and manage actors (launch, send orders...)
- Acquisition layer: actors
  `generic_...` to load C++/Java/... shared libraries.
  `data_rate_...` to test rates.
  `generic_ada_actors` to load Ada shared libraries.

5.5 Buffer API

It is a new buffer for each transfert: cannot read while still writing in.

- Common interface
- `Buffer_Handle_Type` embed all needed information about the buffer
- Host interface (remote buffer are still managed by an host)
- Produce interface
- Consume interface

Buffer pool:

- cf man 7 mq_overview
- IPC++

Infiniband ?