EPICS: Experimental Physics and Industrial Control System

Control Architecture
Reading Group
Overview

- What, Why and Who?
- The Subsystems
- Performance
- Conclusions
What is EPICS and Why?

- Scaleable “real-time” remote control
  - distributed systems
  - small test stands
- Client / Server Model
  - Server: low-level hardware
  - Client: user interface
- Control: supervisory, closed-loop and sequential
- “Configuration tools in place of programming”
- Large installed base of tested software
- Modular design that supports incremental upgrades
- Well defined interfaces for extensions at every level
Who is Using EPICS?

- Over 90 independent projects in North America, Europe and Asia
  - Los Alamos National Laboratory
  - Argonne National Laboratory
  - Lawrence Berkeley Laboratory
  - Superconducting Super Collider Laboratory
  - Continuous Electron Beam Accelerator Facility
  - University of Saskatchewan, UBC
  - Duke University, Stanford
  - Scientific Instrument Limited
Accelerators: Think BIG!
EPICS Subsystems (1)

Input-Output (Real-World)

Distributed Run-Time DataBase
Input-Output Controller (IOC)

Custom Programs
C, Java, Matlab, Mathematica, Perl, Python
State Notation Language

Display Manager
"Pretty Pictures"

Alarm Manager
"Danger!"

Archiver
"Store Data"

Sequencer
"State Machines"

June 23, 2004
Control Architecture Reading Group
EPICS Subsystems (2)
Distributed Database (Servers)

- **Database**: local control
  - Highest level on each IOC
  - Above hardware drivers
  - *Simple* config file

- Data Acquisition
- Data Conversion
- Alarm Detection
- Closed Loop Control
- 4-100 kHz PID loops
Display Manager (Client)

- Interface to Operator
- X-Windows
- Strip Charts, etc.
Alarm Manager (Client)

- “Fault Trees”
- Steady State Operation
- Give guidance to operator
Archiver (Client)

- Data to Disk
- Select Channels to Retrieve
- 5000 Channels / sec
- Multiple Archivers at once on network
Sequencer (Client)

- Execute State Machines
- Runs on each IOC
- “State Notation Language”
- Switches Op. Modes
- Handles Exceptions
- C code can be added
Channel Access

- Controls how clients and servers talk to each other
- “Software Bus”
- Over TCP or UDP
- Establish connections
- Get, Put, Monitor Info
Event Synchronization

- “Real Time” across network
  - millisecond time-stamps
- Measure same event across network
- Based on individual local machine clock
- Avoid Ethernet “collisions”
I/O & Network Performance

- 4-100 kHz IOC low-level loops
- < 60 Hz Channel Access Loops
- 10,000 Channel Access monitors per second on 10 MBit Ethernet
- Ethernet load < 30% (for determinism)
- **Signal latency on network: 2ms+**
  - 68040 on 10 Mbit Ethernet
- “Network bandwidth is the primary limiting factor”
- The Ground Test Accelerator (old stats)
  - 2,500 physical connections
  - 10,000 database records in 14 IOCs
  - 8 workstations
  - 5-7% of 10 Mbit Ethernet.
Reliability

- Accidents are Expensive!
- 95% uptime
- Lots of testing, been around for years
- Not a toy or pet project
Portability

- VxWorks, Linux, Windows, RTEMS, Darwin, Solaris
- Control Net, PCI, CAN-Bus, Industry Pack, VME, VXI, PCI, ISA, CAMAC, GPIB, Profibus, Bitbus, Serial, Allen-Bradley, Modbus, Yokogawa, G-3, Ethernet/IP
- 500kB+ Server Executable
- RTEMS vs. VxWorks
  - IOC
  - Critical: Hard real-time
  - RTEMS as fast as VxWorks
- Linux (“all-in-one”)
  - non-critical systems
Conclusion

- Scaleable
- Distributed
- Deterministic & relatively fast
  - (1ms time-stamps)
- Reliable
- Ethernet-based control architecture
- Standard open-source Unix tools
Further Reading

- “Recommended” Documents
- [http://lansce.lanl.gov/lansce8/Epics/epicsX5Farch-1.html](http://lansce.lanl.gov/lansce8/Epics/epicsX5Farch-1.html)
- EPICS Architecture @ ANL
- EPICS: Recent Developments and Future Perspectives
- EPICS on the RTEMS real-time executive for multiprocessor systems