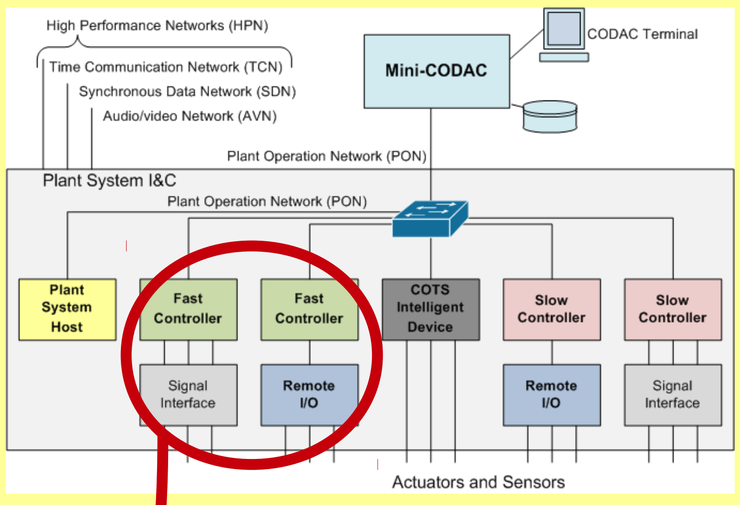
Optimizing EPICS for Multi-Core Architectures

ICALEPCS 2013
14th International Conference on Accelerator and Large Experimental Physics Control Systems
06-11th October 2013
San Francisco, USA

ITER Fast Controller

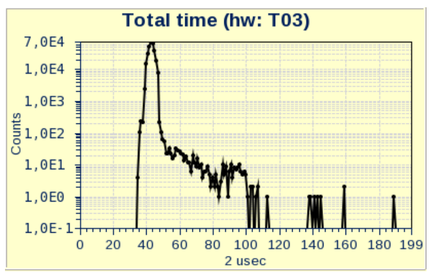


- PICMG 1.3 compliant industrial PC
- PXI/PXIe/cPCI chassis over PCIe link
- Red Hat Enterprise Linux (RHEL)
- Real-time extensions (RHEL MRG-R)

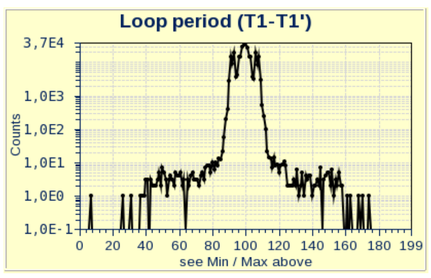
First Results

Improvement of real-time behavior on a test system running a simulated fast control loop, sampling an analog signal from an A/D converter and reading time stamps from a timing card

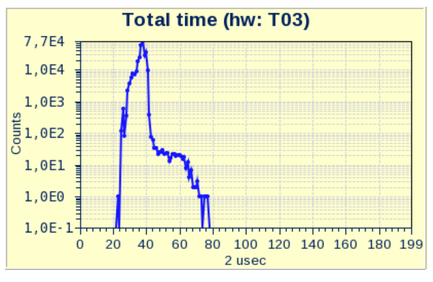
Total loop time

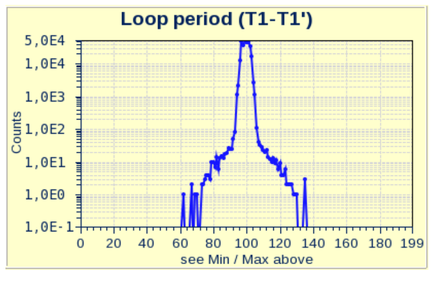


Periodicity



No optimization.





Dedicated core, high priority.

Abstract

EPICS is a widely used software framework for real-time controls in large facilities, accelerators and telescopes. Its multithreaded IOC (Input Output Controller) Core software has been developed on traditional single-core CPUs. The ITER project will use modern multi-core CPUs, running the RHEL Linux operating system in its MRG-R real-time variant. An analysis of the thread handling in IOC Core shows different options for improving the performance and real-time behavior, which are discussed and evaluated. The implementation is split between improvements inside EPICS Base, which have been merged back into the main distribution, and a support module that makes full use of these new features. This paper describes design and implementation aspects, and presents results as well as lessons learned.

Ralph Lange
Helmholtz-Zentrum Berlin für Materialien und Energie / BESSY II, 12489 Berlin, Germany
Franck Di Maio
ITER Organization, Route de Vinon, CS 90 046, 13067 Saint Paul-lez-Durance Cedex, France

Improvements

- **Enhancement of EPICS thread show routines**
  Easier correlation with Linux system level commands

- **Parallelization of callback threads**
  Less latency, lower queue usage, higher processing throughput

- **Rule-based CPU affinity, scheduling policy, and priority settings**
  Fine-tuning to run IOC on subset of CPUs, dedicate CPUs to EPICS or external real-time processing, optimize system's performance

```
top - 08:50:00 up 18 days, 13:49,  7 users,  load average: 1.20, 1.54, 1.20
Threads:  29 total,   1 running,  28 sleeping,   0 stopped,   0 zombie
%Cpu(s): 14.0 us,  0.2 sy,  0.0 ni, 85.4 id,  0.4 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:  12327468 total, 11579136 used,   748332 free,   888180 buffers
KiB Swap: 23437304 total,    33776 used, 23403528 free,  6011940 cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+ COMMAND
20902 lange     20   0 1759m 4976 2668 R 99.9  0.0   0:43.29 cbmScanner
20891 lange     20   0 1759m 4976 2668 S 14.0  0.0   0:06.00 cbLow-2
20892 lange     20   0 1759m 4976 2668 S 14.0  0.0   0:06.02 cbLow-3
20889 lange     20   0 1759m 4976 2668 S 13.6  0.0   0:05.97 cbLow-0
20890 lange     20   0 1759m 4976 2668 S 13.6  0.0   0:05.98 cbLow-1
20884 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:02.37 _main_
20885 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 errlog
20887 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 taskwd
20888 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 timerQueue
20893 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 cbMedium-0
20894 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 cbMedium-1
20895 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 cbMedium-2
20896 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 cbMedium-3
20897 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 cbHigh-0
20898 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 cbHigh-1
20899 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 cbHigh-2
20900 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 cbHigh-3
20901 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 dbCaLink
20903 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 scanOnce
20904 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 scan-10
20905 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 scan-5
20906 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 scan-2
20907 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 scan-1
20908 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 scan-0.5
20909 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 scan-0.2
20910 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 scan-0.1
20911 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 CAS-TCP
20912 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 CAS-beacon
20913 lange     20   0 1759m 4976 2668 S  0.0  0.0   0:00.00 CAS-UDP
```

Output of top command showing EPICS thread names.

```
# Rules for MCoreUtils
#
# Format of each line:       name:policy:priority:affinity:pattern
#
# name      distinguishing tag
# policy    scheduling policy (first letter suffices, case independent, * = don't change)
# priority  scheduling priority (OSI units, + or - defines a relative change, * = don't change)
# affinity  CPU set (use , and - to specify ranges, * = don't change)
# pattern   regular expression to match thread names against

IOC:f:*:0-3:.*
Loop:f:99:4:TestLoop
```

Example rules file:
Run all EPICS threads on CPUs 0-3,
but run TestLoop at priority 99 on CPU 4.

Further Possibilities and Plans

- **Callbacks for Scan-I/O mechanism**
  Add callbacks to the scanIoRequest API, so that drivers know when records have finished processing

- **Driver „private“ callback threads**
  Add additional user-configurable priorities, so that drivers can use dedicated callback threads, with configurable number of parallel threads and queue depth (depth = 0 directly processes records from driver thread)

The authors would like to thank the ITER CODAC team and the EPICS Base Developers for their cooperation, help, and fruitful discussions.