1. Introduction 1
2. A Simple Astrometric Kernel 2
2.1. Servo Cycle 2
2.2. SCHA Module 2
2.3. SCDEC Module 2
2.4. SASZ! Module 2
2.5. CASZ! Module 3
2.6. RDZA! Module 3
2.7. RDAZ! Module 3
2.8. CZ Module 4
2.9. ATAN(Y/X) Module 4
2.10. PTCOR Module 4
2.11. REFRACT Module 4
2.12. DOHPAR Module 5
2.13. Total Floating Point Operations 5
3. Floating Point Load 6
4. Epics Performance 7
4.1. Overview 7
4.2. TRP 7
4.2.1. Record Processing Model 7
4.3. TCA 8
4.3.1. APS STudy 8
4.3.2. SSC STudy 9
4.3.3. Estimator for TCA 9
5. Model TCS Results 10
5.1. Limits On Number Of Channel Access Clients 10
5.2. Discussion 11

GEMINI 8-M TELESCOPES PROJECT

CONTROLS GROUP

PERFORMANCE ESTIMATES OF AN EPICS BASED TELESCOPE CONTROL SYSTEM

DOCUMENT NUMBER: RPT-C-G0024
ISSUE NUMBER: 001
REVISION NUMBER: 1.0
DATE: 26 OCTOBER 1993
PREPARED BY: P.M. MCGEHEE
APPROVED BY: R.J. MCGONEGAL
PROPRIETARY/NON-DISCLOSURE: NO
PROCUREMENT SENSISTIVE: NO
DISCLOSABLE: YES

Abstract

The Gemini 8-m Telescopes Project has chosen the EPICS (Experimental Physics and Industrial Control System) toolkit as the foundation of it's real-time control system. An estimate of the Telescope Control System [TCS] performance is derived using astrometric kernel floating point loads and EPICS timing studies done at the Superconducting Super Collider Laboratory and at Argonne National Laboratory.

This report calculates the maximum number of EPICS channel access clients that can be supported on a Motorola MVME-167 (M68040) VMEbus controller with a CPU loading of no more than 50% for a variety of TCS cycle rates ranging from 10 Hz to 50 Hz. The study of an astrometric kernel processing system shows that an EPICS based TCS should be able to handle the control of it's associated physical subsystems located in remote IOCs at 50Hz with a CPU loading a factor of two less than 50%.

1 INTRODUCTION

The Gemini 8-m Telescopes Project has included the EPICS (Experimental Physics and Industrial Control System) toolkit as the foundation of it's real-time control system. This system will be used on all of the VxWorks machines and it's real-time database access layer ("channel access") will be the standard interface between the real-time work packages.

A critical part of the Gemini control system is the Telescope Control System (TCS) package that will be responsible for performing the complete astrometric kernel and sending commands to the mount, primary mirror, secondary mirror, cassegrain rotator, atmospheric dispersion compensator, and the enclosure.

These systems will be found in the following IOCs (Input-Output Controllers):

SYSTEM	**IOC**
TCS	TCS
Mount	Mount
Primary Mirror	Primary Mirror
Secondary Mirror	Secondary Mirror
Cassegrain Rotator	Cassegrain Rotator
ADC	Primary Mirror
Enclosure	Enclosure

In order for the Gemini telescopes to meet their tracking error budgets it is estimated that the positions (Alt/Az) sent to the mount servo systems must be updated at a rate of 50Hz. This is with the mount servos utilizing an intelligent motor controller like the DeltaTau PMAC card to perform the actual loop closure. This report presents the analysis of the CPU load on an MVME-167 (MC68040-based) VMEbus controller running the TCS under the VxWorks and EPICS environment.

2 A SIMPLE ASTROMETRIC KERNEL

As a baseline design the RA-DEC to ALT-AZ conversion done within the Caltech Submillimeter Observatory's Antenna microcomputer's SERVO module was used to study the amount of floating point computations required. A breakdown of the floating point operations performed is outlined here. The actual FORTH source code is available as a separate document.

2.1 Servo Cycle

The celestial mode servo cycle for the CSO includes calls to the following modules:

- SCHA
- SCDEC
- SASZ!
- CASZ!
- RDZA!
- RDAZ!
- PTCOR

2.2 SCHA Module

This module computes the sin and cos of RA.

Following point operations:

Operation	Count
F+	4
F-	2
F*	2
SIN	1
COS	1

2.3 SCDEC Module

This module computes the sin and cos of DEC.

Following point operations:

Operation	Count
F+	5
SIN	1
COS	1

2.4 SASZ! Module

This module computes sinAZ * sinZA.

Following point operations:

Operation	Count
F*	2

2.5 CASZ! Module

This module computes cosAZ * sin ZA

Operation	Count
F-	1
F*	3

2.6 RDZA! Module

This module converts RA-DEC to Zenith Angle.

Other modules called:

- CZ

Floating point operations:

Operation	Count
F+	8
F-	2
F*	3
SQRT	1
ASIN	1

2.7 RDAZ! Module

This module converts RA-DEC to Azimuth Angle.

Other modules called:

- ATAN(Y/X)

Floating point operations:

Operation	Count
F+	7
F-	4
F*	3

F/	1

2.8 CZ Module

This module converts DEC to cosine ZA.

Floating point operations:

Operation	Count
F+	1
F*	3

2.9 ATAN(Y/X) Module

This module computes the arctan.

Floating point operations:

Operation	Count
F+	1
ATAN2	1

2.10 PTCOR Module

This module computes pointing corrections:

Other modules called:

- REFRACT
- DOHPAR

Floating point operations:

Operation	Count
F+	12
F-	1
F*	13
F/	2
SIN	1
COS	1

2.11 REFRACT Module

This module computes the refraction.

Floating point operations:

Operation	Count
F+	12
F-	2
F*	16
F/	3

2.12 DOHPAR Module

This module computes the horizontal parallax.

Floating point operations:

Operation	Count
F-	1
F/	1

2.13 Total Floating Point Operations

A sum of the operations performed in this simple model is shown.

Operation	Count
F+	50
F-	13
F*	45
F/	7
SIN	3
COS	3
ATAN2	1
ASIN	1
SQRT	1

3 FLOATING POINT LOAD

For a Motorola Floating Point Co-Processor running in Double Precision the execution times are:

Operation	Clock Cycles
F+	78
F-	78
F*	98
F/	130
SIN	416
COS	416
ATAN2	428
ASIN	606
SQRT	132

Combining this table with the one in section 2.13 gives the following total number of clock cycles taken up by the astrometric kernel.

Total clock cycles = 3900 + 1014 + 4410 + 910 + 1248 + 1248 + 428 + 606 + 132

= 13,896.

Total CPU time (@ 25MHz) = 556 microseconds.

An observation by P.Wallace (STARLINK) is that a full-blown astrometric kernel (as in SLALIB) requires that this estimated CPU time based on a simple model be multiplied by a factor of ~4.

Total CPU time (@ 25MHz) = 2224 microseconds.

This CPU time would be for each update cycle run in the TCS.

4 EPICS PERFORMANCE

4.1 Overview

This section will use data from the Superconducting Super Collider Laboratory (SSC), from the Advanced Photon Source at Argonne, and the astrometric kernel results from the previous section to create a CPU loading model.

The Physics Research Division at the SSC conducted an extensive performance test of the EPICS system on MVME-167 processors. Their results showed that the total CPU usage by the EPICS IOC is:

CPU = TRP + TCA + tKernel

where TRP is the sum of the time spent processing all the records in the database

TCA is the sum of the time spent by all tasks that service client's event queues

tKernel is the spent by the VxWorks kernel during task management (assumed small).

4.2 TRP

The total record processing time is the sum of the Record Processing (RP) time spent in each individual process variable. The RP is dependent upon the performance of the various layers. Each process variable is defined by a record type (AI, AO, BI, CALC, etc), a device type (PMAC, SOFTWARE, etc), the number of elements in the record (usually applies to WAVEFORM and COMPRESS records), and the scheduled frequency (10Hz, 5Hz, etc).

EPICS works through four distinct layers: Database, Record, Device Support, and Device Driver.

RP = [tDb(recType) + tRec(rectype, caReq) + tDev(recType, devType) +
tDriv(devType, nElem) + tFetch] * schFreq

caReq is the number of channel access clients requesting this process variable.

tFetch is the time spent by the EPICS task fetching the next process variable
(in the order of 10 to 18 microseconds).

4.2.1 Record Processing Model

If we define the record processing overhead (RPO) as the sum of the time spent in the Database, Record, and Device Support layers, it is clear from the SSC report that the RPO does not vary significantly across the range of record types. The RPO for the Analog Input record (AI) will be used since it has one of the highest values.

The device type (devType) will be pure software to represent the astrometric kernel. The time spend in the device driver layer will then be defined by the kernel calculation time. The number of records will also be defined as 1 (one).

Then:

tDb(AI) = 9.8 microseconds

tRec(AI, caReq) = 43.8 + 50.9caReq microseconds

tDev(AI, AST-KERNEL) = 2 microseconds

tDriv(AST-KERNEL, 1) = 2224 microseconds.

Thus,

TRP = RP = [2298 + 50.9caReq]*schFreq microseconds/second

= 1E-4*[2298+ 50.9caReq]*schFreq %

4.3 TCA

The amount of time spend handling channel access processing as distinct from simple record processing was not studied in depth by the SSC paper. The Advanced Photon Source (APS) at Argonne National Laboratory has done an informal study of the effect of channel access on CPU usage in fast (60Hz) applications with a MVME-167 IOC.

4.3.1 APS Study

The APS test environment consisted of four distinct CPU loads:

fastRP = record processing of 27 custom records at 60 Hz

fastCA(X records @ Y Hz) = channel access of the above records

slowRP = record processing of *some* number of slow records (< 1 Hz)

slowCA = channel access of the above slow records at normal record processing rates

CASE	fastRP	fastCA	slowRP	slowCA	% CPU
I	YES	NO	YES	NO	28
II	YES	NO	YES	YES	34
III	YES	120 records @ 5Hz	YES	YES	41
IV	YES	30 records @ 60Hz	YES	YES	68
V	YES	60 records @ 60HZ	YES	YES	CA dies

A few conclusions can be drawn from this table:

1) The total record processing overhead (**fastRP** + **slowRP**) is 28%.
2) **slowCA** is 6%.
3) **fastCA**(120 records @ 5Hz) is 7% which implies a linear scale of 0.012 %/post-per-sec.

4) **fastCA**(30 records @ 60Hz) is 34% which implies a linear scale of 0.019 %/post-per-sec.

5) Using the linear scale derived in 4), the CPU load in case V is 102%.

The conclusion that the APS has come to is that although EPICS record processing is fast, the overhead associated with channel access can be significant. For their fast applications (27 records driven at 60Hz by I/O events) they have created custom records that have a setable posting rate (usually at 2 Hz). The fast values written into COMPRESS circular buffers and the slow calls to db_post_events() passes to channel access an average value.

4.3.2 SSC Study

The SSC study shows for a MVME-167 system with 100 10Hz AI records that for **fastCA**(10 records @ 10Hz) the CPU load was at 21% and for **fastCA**(100 records @ 10Hz) the CPU load was at 32%. This leads to the conclusion that the **fastCA** linear scale is 0.0122%/post-per-sec in agreement with the APS results.

4.3.3 Estimator For TCA

Using a linear scale of 0.015%/post-per-sec, the TCA is estimated as

TCA = 0.015*schFreq*caReq %

5 MODEL TCS RESULTS

Having obtained an estimated relationship between TCS configuration and CPU loading we will predict the number of channel access connections that can be maintained at a CPU loading of 50% for a range of TCS processing rates in the 10Hz to 50Hz range.

For this example TCS we use the following model:

1) The entire astrometric kernel is done as a monolithic subroutine record at a rate of **schFreq.**
2) The information for the physical subsystems is shipped out via channel access at **schFre**q to **caReq** channel access clients residing in remote IOCs.
3) The VxWorks overhead tKernel is small - we assume that it is effectively zero..

5.1 Limits On Number Of Channel Access Clients

The sum of the TRP and TCA estimators derived in the previous sections gives:

$$CPU = 1E\text{-}4*[2298 + 50.9caReq]*schFreq + 0.015schFreq*caReq \%$$

$$= 0.2298schFreq + 0.0201schFreq*caReq$$

which implies

$$caReq = (CPU/schFreq - 0.2298)/0.0201$$

caReq is then the maximum number of channel access client connections that can be supported at a given CPU loading and TCS processing rate.

The equations for the CPU loading components are:

$$Astrometric\ Load = 0.2224schFreq$$

$$Record\ Processing\ Load = 0.0074schFreq + 0.00509caReq*schFreq$$

$$Channel\ Access\ Load = 0.015caReq*schFreq$$

For CPU = 50%:

schFreq	Astrometric	Record	CA	caReq
10 Hz	2.2%	12%	35%	237
20 Hz	4.4%	11%	33%	112
30 Hz	6.7%	11%	31%	71
40 Hz	8.9%	10%	30%	50
50 Hz	11%	10%	28%	38

5.2 Discussion

The Gemini Telescope Control System must be able to support channel access clients associated with the mount, primary mirror, secondary mirror, cassegrain rotator, and ADC subsystems. These subsystems expect the following information from the TCS:

- Mount - Azimuth position, Azimuth velocity, Altitude position, Altitude velocity
- Primary Mirror - List of Zernicke polynomial coefficients
- Secondary Mirror - List of Zernicke polynomial coefficients
- Cassegrain Rotator - Field rotation angle rate of change
- ADC - ADC angle

If the position and velocity parameters are implemented as Analog Output records and the Zernicke lists are written into array buffers (Compress record in Circular Buffer mode), then the TCS must at a minimum provide fast updates to 8 channel access clients. Given the results in the previous section, this goal can be achieved with a CPU loading of 20% on a MVME-167 for a TCS rate of 50Hz.

Although EPICS channel access can present a major part of the CPU loading in fast applications it does not appear to be a severe bottleneck for the planned VxWorks/EPICS systems although the presence of this overhead may certainly influence design decisions. In the five cases examined in section 5.1 the channel access posting rate associated with the 50% CPU loading was in the range of 1900 to 2300 posts-per-second with the channel access posting accounting for ~60% of the CPU usage.