



Distributed Control Systems at SSRL

Constraints for Software Development Strategies

Timothy M. McPhillips
Stanford Synchrotron Radiation Laboratory

Overview



Computing Environment at our Beam Lines

- Need for cross-platform development.
- Need for distributed architecture.

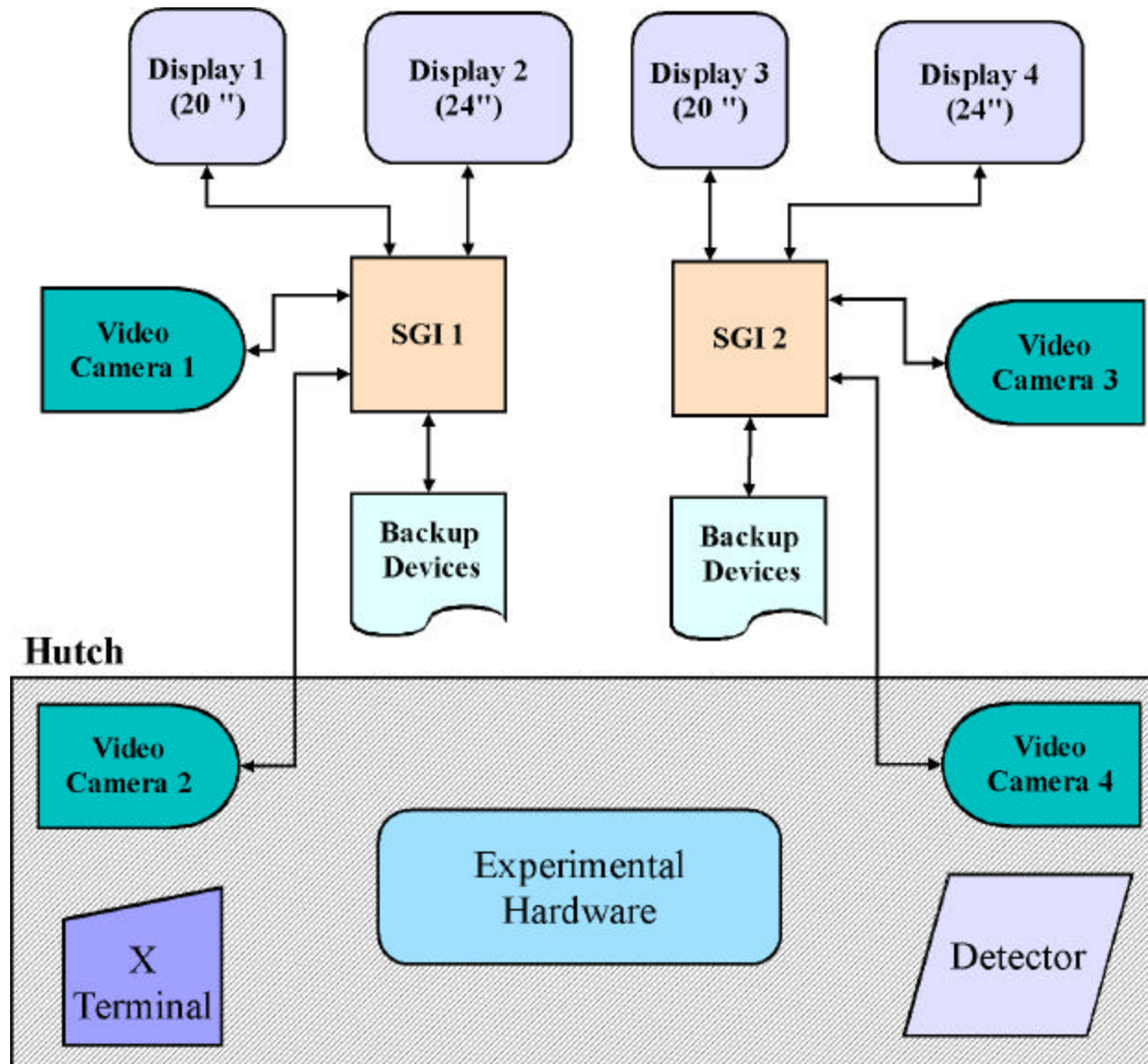
Beam Line Software

- Need for collaborative software.
- Need for high performance software.

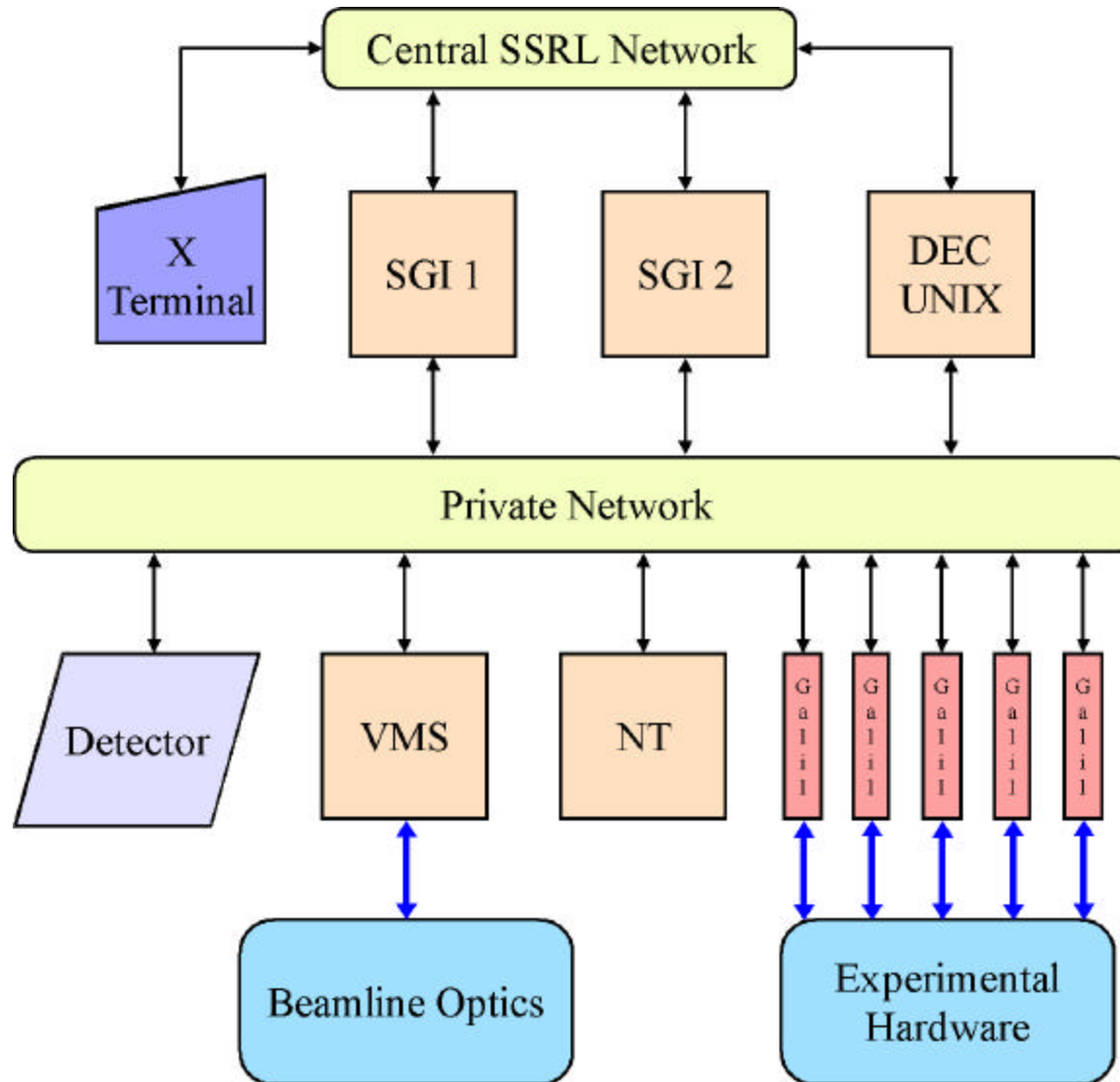
Software Development Strategy

- Constraints on our strategy.
- Our current strategy.

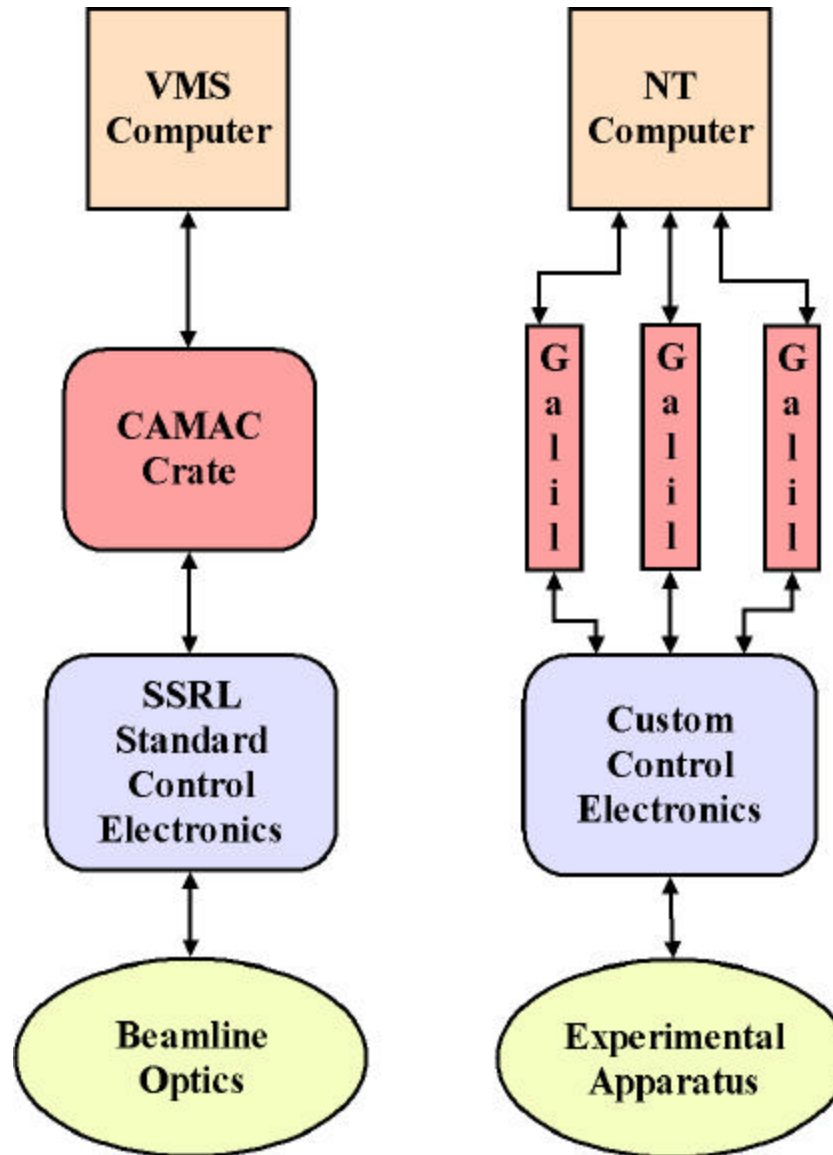
User Environment at Beam Line



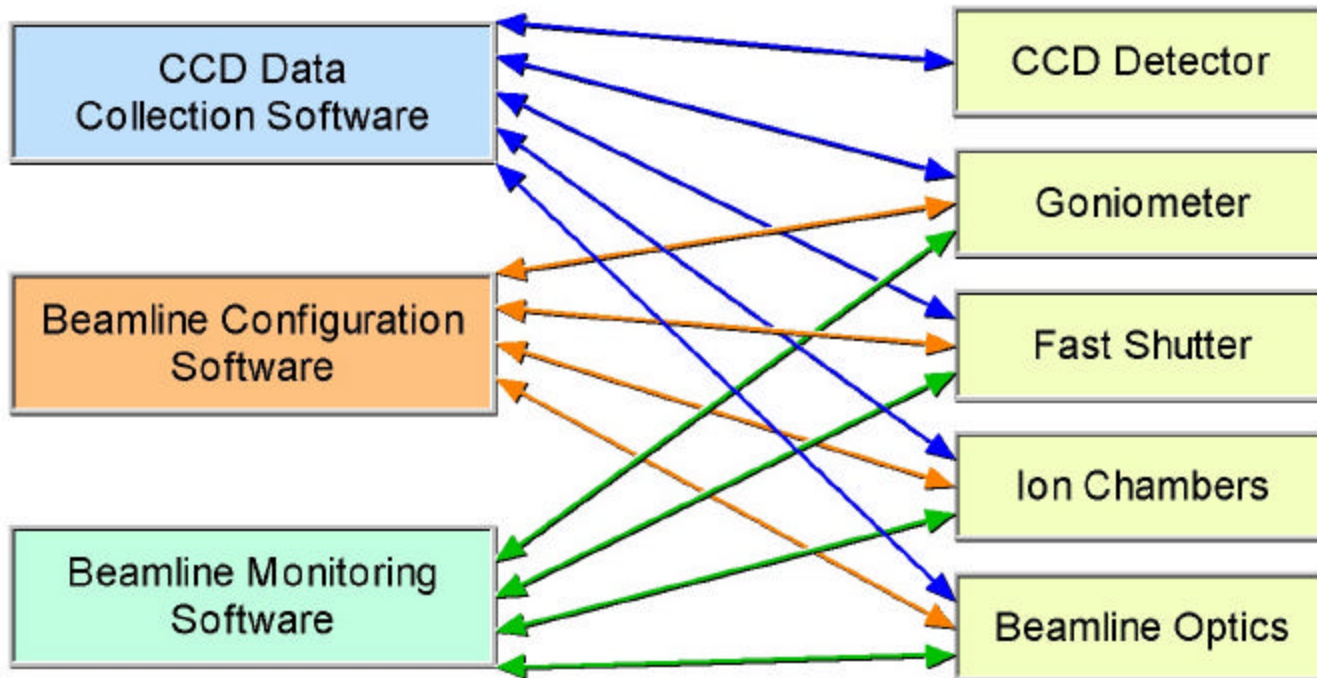
Network Architecture of a Beam Line



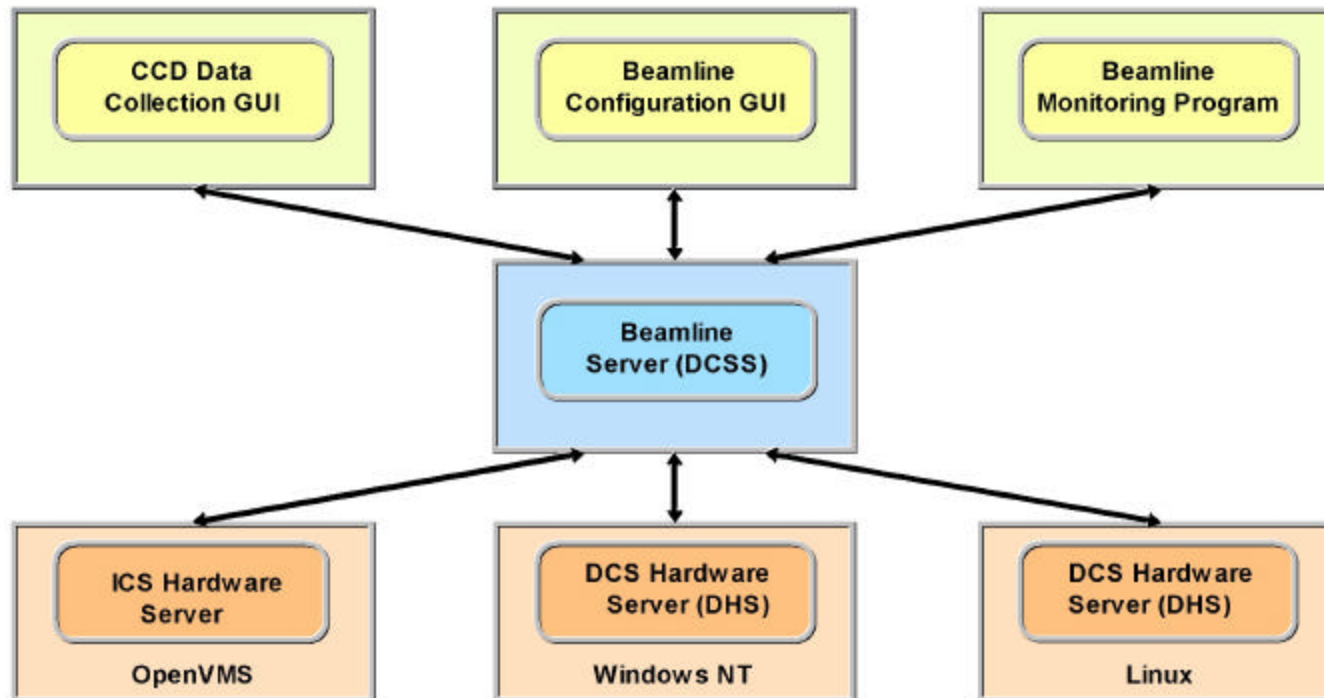
Problem 1: Multiple Hardware Hosts



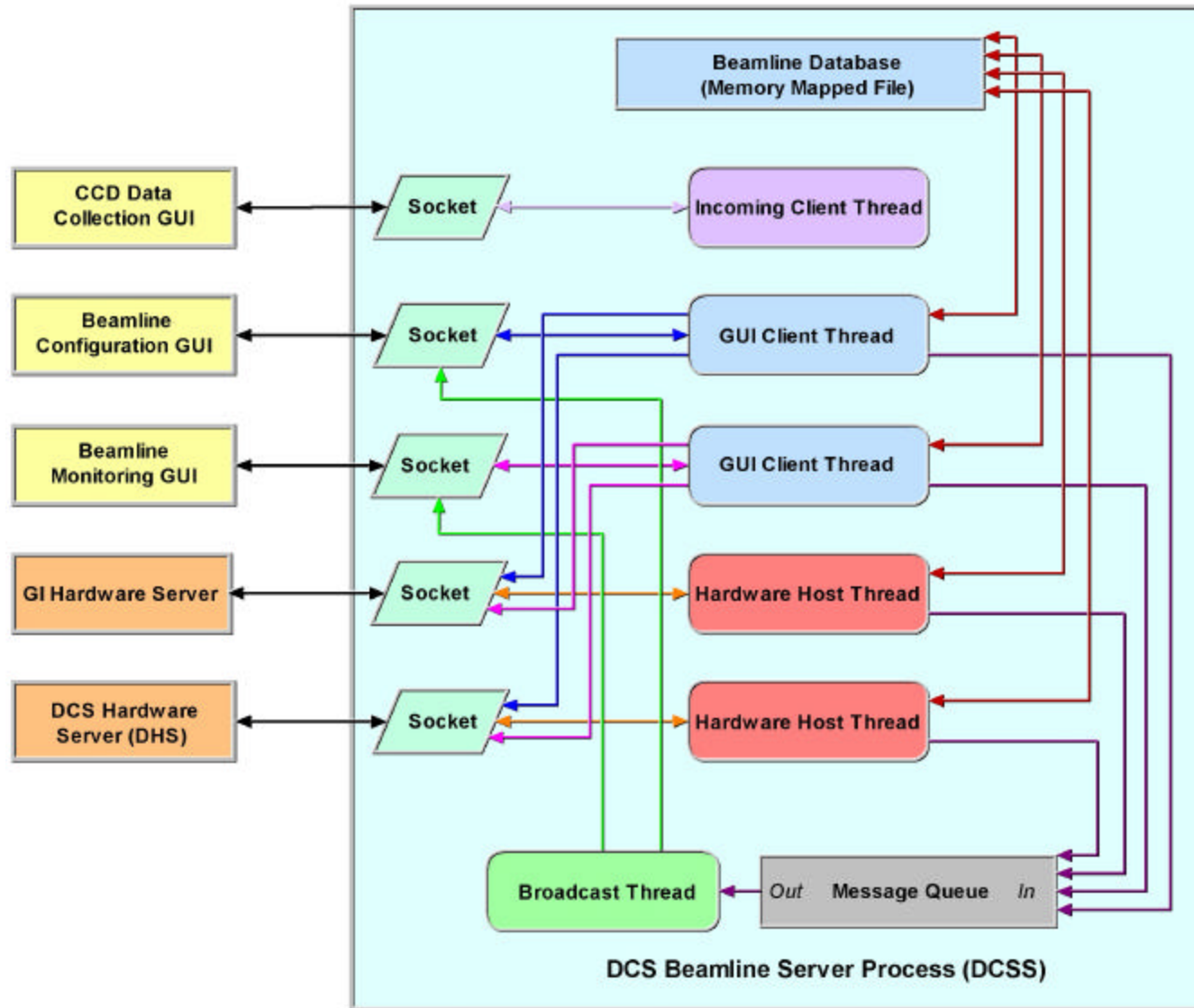
Problem 2: Multiple, Simultaneous User Interfaces



Solution: Distributed Control System (DCS)



DCS Server (DCSS)



Beam Line Universal - Integrated Configuration Environment (BLU-ICE)



The screenshot displays the 'Beamline 9-1 Configuration' software interface. At the top, a menu bar includes 'File', 'Component', 'Shutter', 'Network', 'View', 'Options', and 'Window'. Below the menu, a 'Selected Motor' dropdown is set to 'mirror_slit_upper'. A row of control buttons includes 'Move by', 'Move to', 'Set to', 'Crash Stop', 'Correct', and 'Scan'. A second row of buttons includes 'Abort', 'Undo Move', and 'Configure'. The main workspace is divided into several panels:

- Table:** A 3D schematic of a table with parameters: table_vert_1 (-17.718 mm, -55389 steps), table_vert (-12.703 mm, 7322 steps), table_vert_2 (2.343 mm, 7322 steps), table_yaw (39.858 deg), table_pitch (-78.724 deg), table_horz (-6.204 mm, -7891 steps), and table_horz_2 (6.262 mm, 8065 steps).
- Monochromator:** A 3D schematic of a monochromator with parameters: mono_angle (26.634 deg, 213072 steps), mono_bend (315.932 mm, 351 steps), mono_slit (40.000 mm, 6320 steps), and mono_theta (13.864 deg).
- Mirror:** A 3D schematic of a mirror with parameters: mirror_slit_upper (12.000 mm, 1893 steps), mirror_vert (7.862 mm, 9631 steps), and mirror_pitch (1.170 deg, 907 steps).
- Hutch Frontend:** A schematic showing a series of slits labeled 2, 4, 8, 16, 32, Sc, Pb, with an initial intensity I_0 of 7320.

A log window at the bottom displays the following text:

```
11 Mar 1999 11:30:07 move table_vert by 0.05 mm
11 Mar 1999 11:30:07 NOTE: Motor table_vert is currently at -12.702800 mm.
11 Mar 1999 11:30:07 NOTE: Move of motor table_vert to -12.652800 mm started.
11 Mar 1999 11:30:08 NOTE: Move of motor table_vert completed normally.
11 Mar 1999 11:30:12 move table_vert by -0.05 mm
11 Mar 1999 11:30:12 NOTE: Motor table_vert is currently at -12.652800 mm.
11 Mar 1999 11:30:12 NOTE: Move of motor table_vert to -12.702800 mm started.
11 Mar 1999 11:30:14 NOTE: Move of motor table_vert completed normally.
11 Mar 1999 11:43:07 configure mirror_slit_upper
```

BLU-ICE Motor Scan Windows



Beamline 9-2 Configuration

File Component Shutter Network View Options Window Help

Selected Motor: mirror_pitch

Move by: 0.331 deg

Buttons: Crash Stop, Correct, Scan, Abort, Undo Move, Configure

Define Scan

File Options

Scan Axes

| Axis | Points | Start | End | Step |
|--------------|--------|-------|-------|------|
| mirror_pitch | 11 | 0.255 | 0.355 | 0.01 |
| (none) | | | | |

Detectors

Signal: i0

Reference: (none)

Timing

Integration time: 0.1 sec

Motor settling time: 0.0 sec

Filters

HA Al_16
 Se Al_32
 Al_8 Al_64

Repeat

Number of scans:

Delay between scans:

Files

Filename root: bl92_o

Scan Number: 98

Scan

Overlay plots

Start

Mirror

Diagram showing mirror components and their positions:

- mirror_slit_upper: 2.998 mm, 473 steps
- mirror_slit_lower
- mirror_vert: 3.000 mm, 3675 steps
- mirror_pitch: 0.306 deg, 237 steps

Scan

File Options Mode Cursor1 Cursor2 Show

bl92_optics_035 11 Mar 1999 12:28:24

Cursor 1
x = 0.331
y = 4621

Cursor 2
x = 0.306
y = 62

dx = 0.025
dy = 4559.435

bl92_optics_035

11 Mar 1999 12:27:54 NOTE: Connecting to server bl921 on port 3175...

11 Mar 1999 12:28:00 define_scan

11 Mar 1999 12:29:34 select_motor slit_1_vert_gap

BLU-ICE Motor Configuration Windows



Beamline 9-2 Configuration

File Component Shutter Network View Options Window Help

Selected Motor: **detector_pitch** | Move by: 0.331 | Move to: | Set to: | **Crash Stop** | Correct | Scan | Abort | Undo Move | Configure

Goniometer

- gonio_phi: 271.000 deg / 54200 steps
- gonio_omega: 0.000 deg / 0 steps
- gonio_z: 0.000 mm / 0 steps

Detector

- detector_vert: 176.231 mm / 277528 steps
- detector_pitch: 0.000 deg / 0 steps
- detector_z: 600.000 mm / 472440 steps
- r_horz: 916 mm / 1 steps

detector_pitch configuration

detector_pitch

Position and Limits

| | | |
|--------------|-------------|--------------|
| Upper limit: | 500.000 deg | 750000 steps |
| Position: | 0.000 deg | 0 steps |
| Lower limit: | 0.000 deg | 0 steps |

Enable upper limit
 Enable lower limit
 Lock motor

Stepper Motor

| | |
|---------------|--------------------|
| Scale factor: | 1500.000 steps/deg |
| Speed: | 1000 steps/sec |
| Accel. time: | 1 msec |
| Backlash: | 0 steps |

Enable anti-backlash
 Reverse motor direction

Apply Close

11 Mar 1999 12:32:46 *select_motor detector_pitch*
11 Mar 1999 12:33:01 *configure detector_pitch*

Advantages of Writing BLU-ICE in Tcl/Tk



Command Prompt with Scripting

- Tcl was originally designed to be an embedded scripting language, so it is easy to give the user a command prompt and a full featured programming language for scripting.
- User can script any operation in BLU-ICE using control structures, variables, procedures, and even classes.

Platform Independent GUI

- Tcl/Tk runs on any Unix, VMS, Mac, and 32-bit Windows computer.
- Scripts can be distributed without compilation and run on any computer Tcl/Tk has been installed on.
- Scripts can also be bundled with Tcl/Tk binaries and distributed as a single executable file. In this case, Tcl/Tk does not have to be installed on the target machine.

Rapid Development

- Tcl/Tk GUIs can be written with only a fraction of the code necessary in typical system programming languages such as C, C++, or Java.
- GUIs can be quickly written and are easy to maintain in Tcl.
- This characteristic is critical in the rapidly changing environments of our beam lines.

Object Orientation

- The [Incr Tcl] extension to Tcl provides object-oriented features such as classes.
- The [Incr Widgets] extension provides an object oriented framework for building complex widgets from built-in Tcl widgets.

Extensible in C/C++

- Tcl was designed to be extended readily in C. Extensions can be loaded dynamically.
- High performance code, multiple threads and so on are best implemented in extensions.

Data Collection with the New BLU-ICE

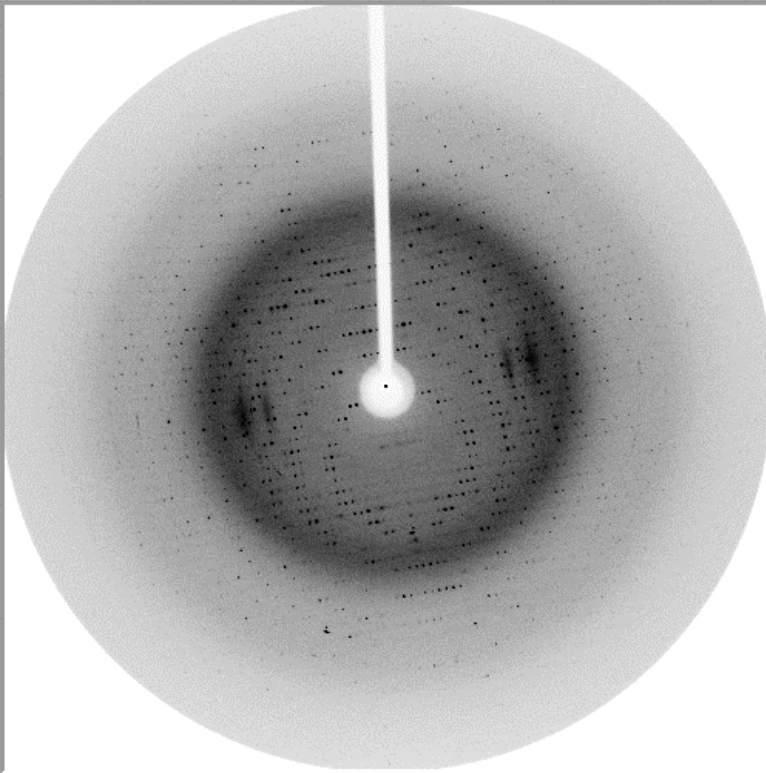


BLU-ICE Beamline 9-2

File Component Shutter Network View Options Window Help

Setup Collect Runs Scan Strategy Monitoring Video

/data/tim/mb_example_079.mar3450



Start
Pause
Abort

Phi: 354.64
Omega: 0.00
Kappa: 0.00
Distance: 505.00

Run 3 (inactive)

Update Delete Reset

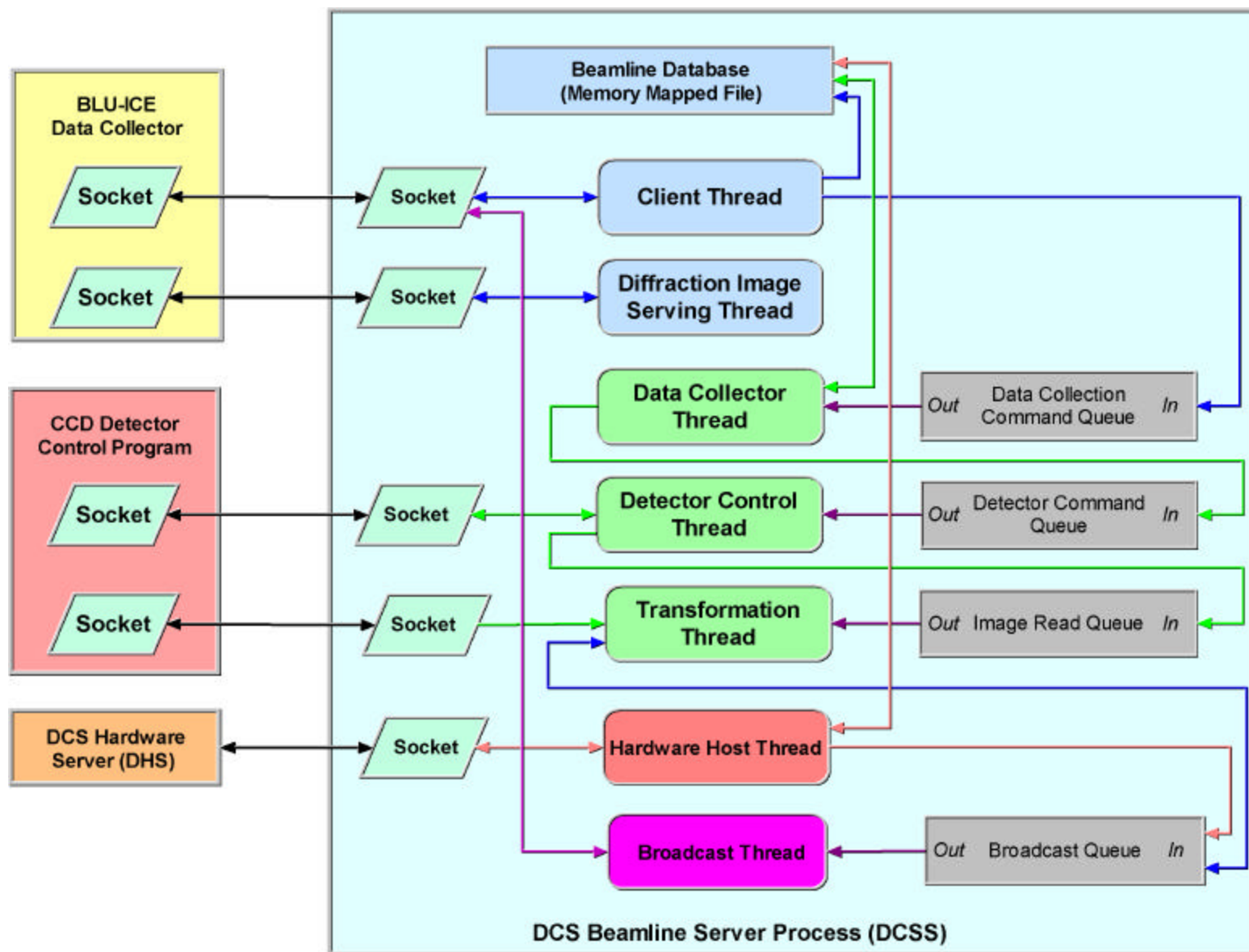
Prefix: test
Directory: /data/tim
CCD Mode: slow
Energy: 16000.03 eV
Distance: 500.000 mm
Axis: phi
Time: 10 sec
Delta: 1.0 deg

| Frame | phi |
|------------|-----------|
| Start: 001 | 15.0 deg |
| End: 090 | 105.0 deg |

Contrast 200 Zoom 1

Idle Energy: 16000.03 eV Network: Master Shutter: Closed 01:21:41 PM

DCSS Performance Now Critical



Cross-Operating System Library (XOS)



Features

- **Supports portable, multithreaded, distributed programs**
 - Network communication using a much simplified socket object.
 - Thread creation and synchronization with mutexes and semaphores.
 - Memory mapped files and hash tables.
 - Interthread communication using message queues and Win32-style messages.
- **Compile-time approach**
 - Header file xos.h loads appropriate, system-dependent include files.
 - Objects hide architectural differences.

Advantages

- **Portability**
 - Compile code on Digital Unix, IRIX, OpenVMS, Windows NT/95.
 - Easy to port to new platforms similar to any of the above.
- **Reliability**
 - Simpler APIs leads to more reliable code.
 - Less need to study different platforms.
- **Performance**
 - Native system calls on each platform for maximum performance
 - No runtime overhead for platform independence.

Constraints for Software Development Strategies



Cross-Platform

- Multiple operating systems needed at beam lines; future needs unknown.
- Remote users of the collaboratory may have many different operating systems.
- Other synchrotron labs and even users' home labs may use our software.
- Must support VMS because other SSRL beam lines use it nearly exclusively.
- Use XOS (Cross-Operating System) Library for low-level software and Tcl/Tk for GUI components when feasible.

Distributed

- Applications must integrate services provided by different computing platforms.
- User interfaces must be kept separate from other components.
- Use TCP/IP socket interfaces between all application components.

High-Performance Server Processes

- Server software must be extremely fast and take advantage of multiple processors.
- Write multithreaded C++ programs with XOS for portability.

Open Source

- Must be able to distribute all software freely without licensing issues.
- Installation distributions, source code and documentation should be nicely packaged.
- Document well enough that other groups can use and extend our solutions on their own.

Low Maintenance Overhead

- Write packages in layers that mix and match.
- Wrap packages in clean APIs that do not require knowledge of underlying code.
- Avoid requiring complex infrastructures. Make it easy for novice programmers.