

Applications for Real-Time Linux

By Steve Brosky

Chief Scientist

Concurrent Computer Corporation

Real-Time Division



2 Approaches to Real-Time Linux

- **RT/Linux or RTAI**
 - Real-Time executive controls the system
 - Linux runs as a process under RT executive
 - RT executive has very fast interrupt response
 - File systems, networking, graphics handled by Linux
- **Real-Time version of Linux**
 - Modifications to Linux itself to support real-time application needs

RT/Linux

- **Advantages**
 - Interrupt response guarantees
 - Minimal intrusion in standard Linux
 - Determinism
- **Disadvantages**
 - multiple programming environments
 - It's not Linux
 - No ability to use file systems, networking or graphics with determinism

Real-Time Version of Linux

- **Advantages**
 - Familiar Linux/POSIX programming environment
 - File systems, networking and graphics are part of the real-time environment
- **Disadvantages**
 - Response time will likely always be slower than RT/Linux
 - Maintainability against evolving Linux baseline

Which Approach is Best?

- **Both approaches have merit**
- **Some applications are better suited by one or the other real-time Linux approach**
- **To support the broadest number of applications, both approaches should be utilized**

Applications that Require a Real-Time Variant of Linux

- **Three application examples will be presented:**
 - **Flight simulator**
 - **Ship-board combat system**
 - **Medical Imaging**

Flight Simulator

- **High-end flight simulators:**
 - **Flight Trainers**
 - e.g. Phase 3 full flight simulator
 - **Motion-based**
 - 6 degrees of freedom
 - **Hardware in the loop**
 - Used to test flight control software
 - **Combat aircraft**
 - Simulates weapons systems
 - Simulates enemy targets



Basic Loop

- **Read inputs**
 - stick / pedal positions
 - flap levers
 - switches / circuit breakers
- **Calculate flight data**
 - angle of wing / position of flaps / velocity
 - air density / weather conditions / foreign objects
 - effect on altitude and attitude of air craft
- **Output**
 - change displays (altimeter, flap display, gauges)
 - update visual system
 - control motion



Real-Time Requirements

- **Loop frequencies vary**
 - 15Hz to 120Hz
- **Calculations to support flight models must be performed deterministically**
- **Time to perform I/O must be predictable**
- **Missed deadlines will cause “jitter”**
 - impact is that the simulation does not feel real

Deterministic I/O

- **Programmed I/O on ARINC or 1553 bus**
- **A/D for cockpit controls can be handled by a specialized box**
 - **Handles individual hard real-time deadlines**
 - **Communicates with host system via point to point Ethernet on a per frame basis**
- **Trainers: must save recorded flight data to disk for later playback**
- **Military: special “target projectors” use UDP/Ethernet in soft real-time**
- **Terrain databases on disk**

Graphics

- **Low-end:**
 - 2 X-tubes in front of the pilot / co-pilot
 - Host system performs graphics in real-time
- **High-end:**
 - Use a dedicated image generator
 - Offloads graphics from the simulation host
 - Provide only x,y,z coordinates and air craft “attitude”



Ship-board Combat System

- **Many inputs**
 - sensors
 - radar
 - sonar
 - gyroscope
 - satellite communication
- **Complex signal processing algorithms**
- **Output**
 - weapons control



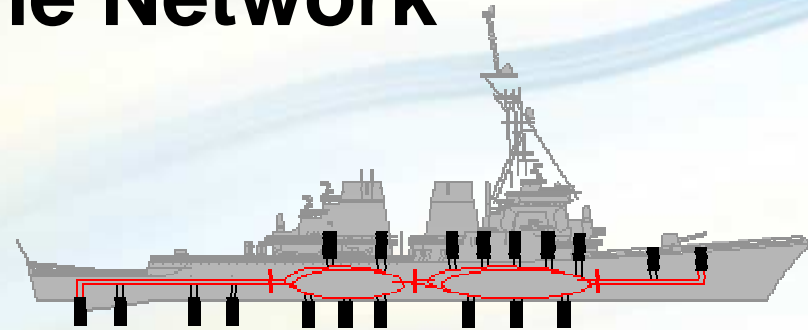
Distributed Compute Environment

- **27 separate computer systems**
- **Intelligent controllers, single board computers and SMP systems**
- **Each Input/Output has an associated processor**
- **Complex processing handled in SMP systems**
- **FDDI, ATM, Fast Ethernet**



Real-Time Network

- **Redundant connections**
- **Switches used to make network communication more deterministic**
- **UDP protocol**
- **Proprietary middleware layer supplies reliability**
- **Determinism of transmissions (must be in the one millisecond range)**



SMP Systems

- **One CPU used for background OS activity**
- **Some CPUs dedicated to signal processing**
 - **hard real-time (10s of microseconds)**
- **Some CPUs dedicated to network I/O**
- **Shared memory communications between dedicated and network CPUs**
 - **low latency**
 - **ease of programming**
- **File system generally not accessed on hard RT processors**
 - **but is available if needed**

Issues

- **Network stack determinism is critical**
- **Rich programming environment is desirable**
 - Ada / C++
 - Middleware is based on POSIX services
 - Toolkits / development environment
- **Partitioning of application**
 - Exclusive access to a CPU for hard RT tasks
 - Willing to waste processor resources to gain determinism

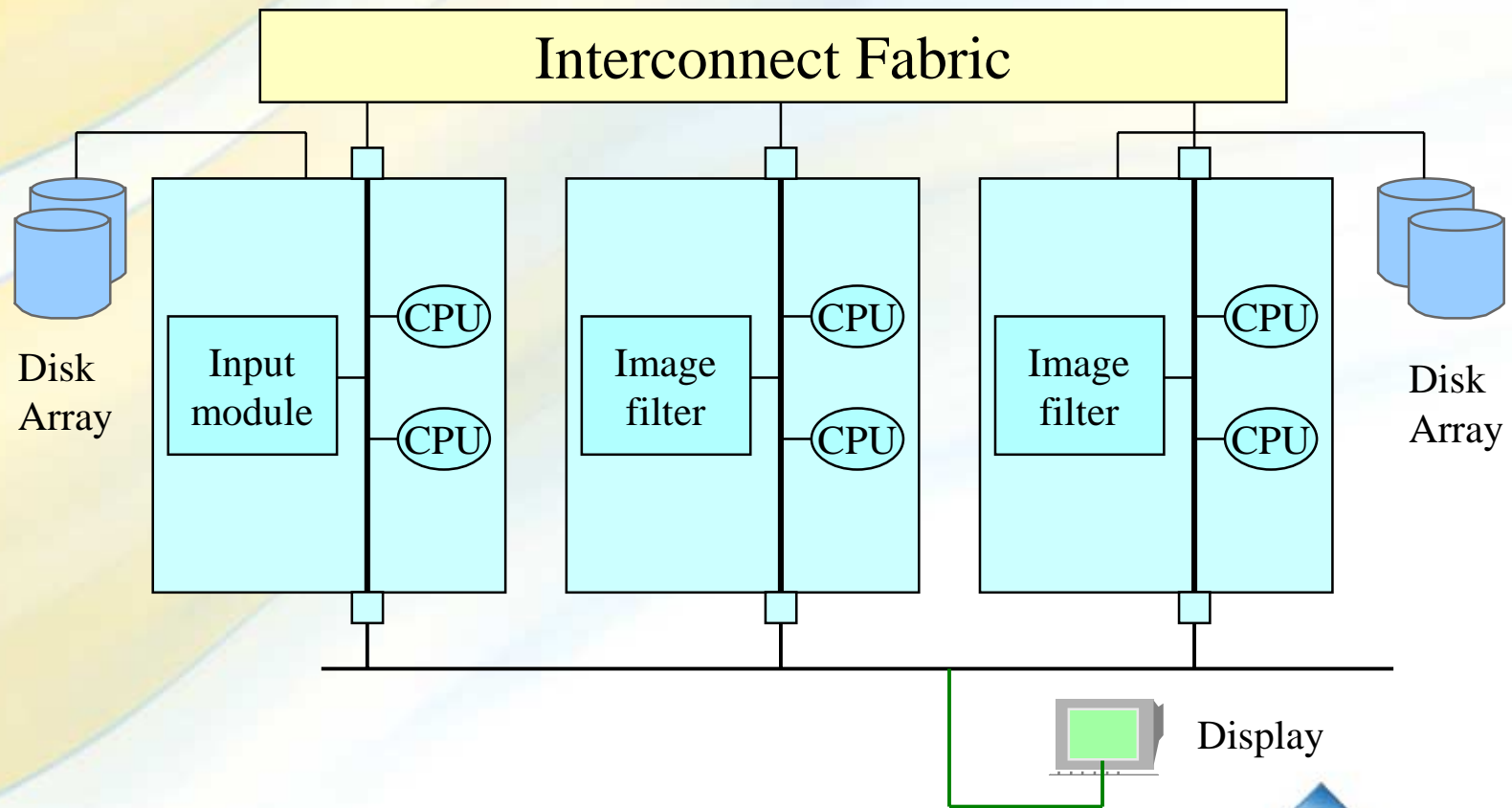
Digital Cardiology

- **Specialized x-ray system**
- **Provides cardiac surgeons with visual feedback and guidance during an operation**
- **X-ray tube mounted on moveable arm**
- **Digital detector plate collects images**
- **Used to guide surgical instruments in a patient's heart**

Data Throughput

- **Continuous sequence of frames acquired**
 - up to 30 frames per sec
- **Images are 1024x1024 pixels**
- **One or two byte pixels**
- **5 seconds of data = 40Mbytes**

Cardiac Intervention System



Requirements

- **Allowed end-to-end latency is in milliseconds**
- **Complex graphics must be performed as part of allowed end-to-end latency**
- **Image processing times must be deterministic**
 - **sub-millisecond range**
- **Large image files must be written to a SCSI disk array in real-time**
- **Large image files must be read from a SCSI disk array in real-time**

Required System Attributes

<i>Requires Deterministic:</i>	<i>File Systems</i>	<i>Networking</i>	<i>Graphics</i>
Flight Simulator	<i>S</i>	<i>S</i>	<i>S</i>
Ship board Combat	-	<i>Y</i>	-
Medical Imaging	<i>Y</i>	<i>S</i>	<i>Y</i>

S = Sometimes Required

Y = Required

- = Not Required

Concerns with Real-Time Linux Variant

- **Linux changes so fast, how can a real-time variant keep up?**
- **A real-time OS requires all developers to be mindful of designing their algorithms so that they block interrupts minimally. How can we maintain this discipline in the open source community?**
- **Linux is too complex to be a real-time operating system.**

How Can We Keep Up With Linux?

- **We can't, so what?**
- **Clearly a real-time Linux offering will be behind the latest development Linux kernels.**
- **Many of these applications are safety-critical - they take years to test and deploy**
- **Their technology is always somewhat behind the cutting-edge**

Can Linux Community Produce a Real-Time OS?

- **The main body of Linux developers will not be concerned with real-time**
- **Producing a real-time Linux variant will involve testing for determinism issues**
- **SMP and exclusive binding to processors helps alleviate this concern**

Stated Goals for Linux 2.5

- **Improved O(1) scheduler (Ingo Molnar)**
- **Improved SMP scaling from 8 or 16 CPUs to 32, 64 and beyond**
 - **Short critical sections are important for massively parallel operation**
- **Better support for high-end NUMA machines**
- **Support for true asynchronous I/O**

Linux is Too Complex

- **The type of applications presented are hosted today on real-time SVR4 variants**
- **Linux is far less complex in its algorithmic design than SVR4**

Towards a Real-Time Linux Variant

- **Can we utilize the open source movement for creating a real-time Linux variant?**
- **Linux is today focused more on general purpose computing**
- **Perhaps some Linux components can be made selectable at configuration time**
 - e.g. the scheduler
- **Perhaps it is time for a real-time fork of Linux?**

Real-Time “Fork”

- A real-time “fork” would have to track the changes in standard Linux
- The “fork” would essentially be a set of patches maintained against standard Linux
- This would work best if sponsored by a group of users that need this technology
- The Trillium Project is an example of companies partnering to create a common open source base

Real-Time Linux Variants

- **There are a few commercial offerings of real-time Linux variants today**
- **More are being developed**
- **This effort will continue, because there are applications that need/want this technology**
- **Perhaps the open source model can be used to create a common baseline to address this application need**