

Embedded Linux is Ready For Real-Time



MONTAVISTA™
S O F T W A R E

Powering the Embedded Revolution

www.mvista.com



MONTAVISTA[™]
S O F T W A R E

Agenda

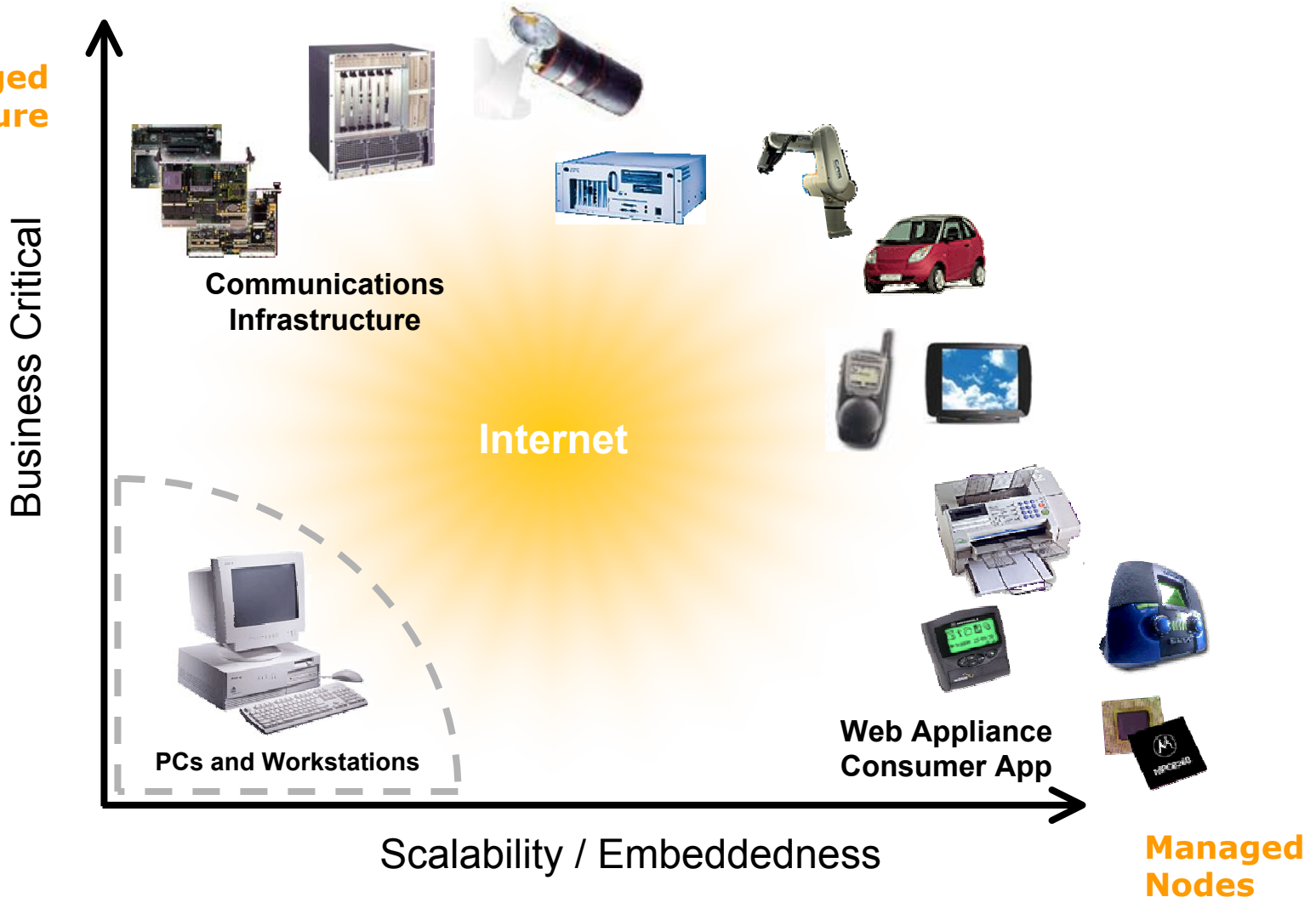
- Intro to Linux and Realtime
- MontaVista Approach to Realtime
- Achievements – Benchmarks
- MontaVista and the Open Source Community
- Conclusions and Future Direction



MONTAVISTA
SOFTWARE

Linux Application Space for the embedded world

**Managed
Infrastructure**





MONTAVISTA[™]
S O F T W A R E

Linux and Realtime

- What is Realtime?
- What is Soft Realtime?
- What is Hard Realtime?



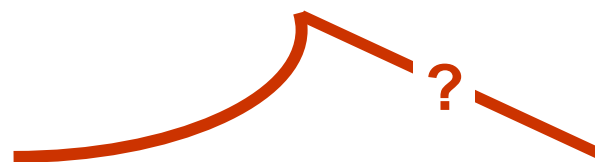
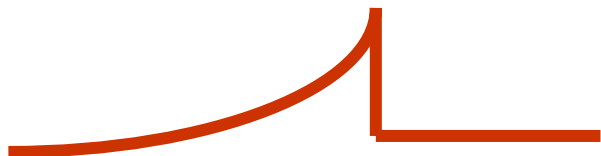
MONTAVISTA
SOFTWARE

Definition of Terms



- In hard real-time system, the value (utility) a computation falls to zero at a given deadline
- In a soft real-time system, that value (utility) declines at some known or unknown rate

-- Dr. R. Callison, Boeing Phantomworks





MONTAVISTA
SOFTWARE

Real-Time and Embedded

■ Perceptions

- From proprietary RTOS platforms
- Traditional association of embedded and hard real-time – questionable?

■ Requirements

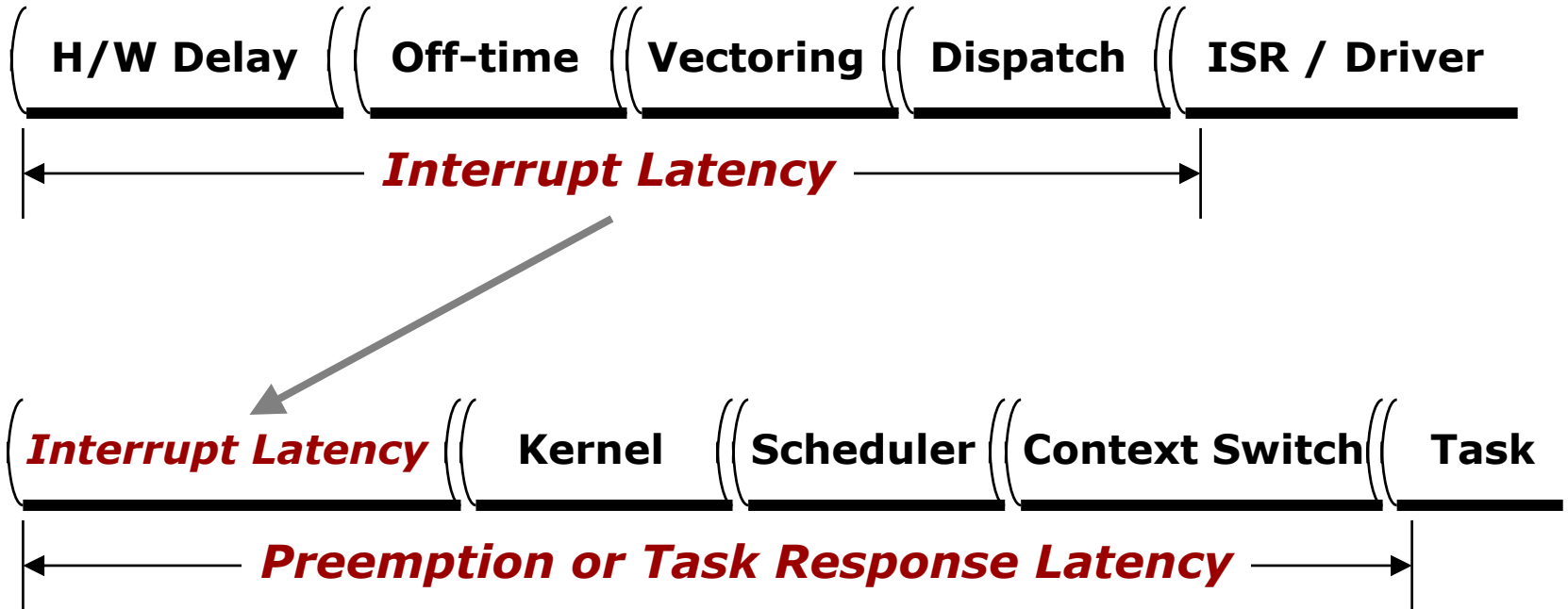
- All embedded apps have a performance requirement
- 30-40% have some real-time need
- < 10% require “hard” real-time





MONTAVISTA
SOFTWARE

Definition of Terms



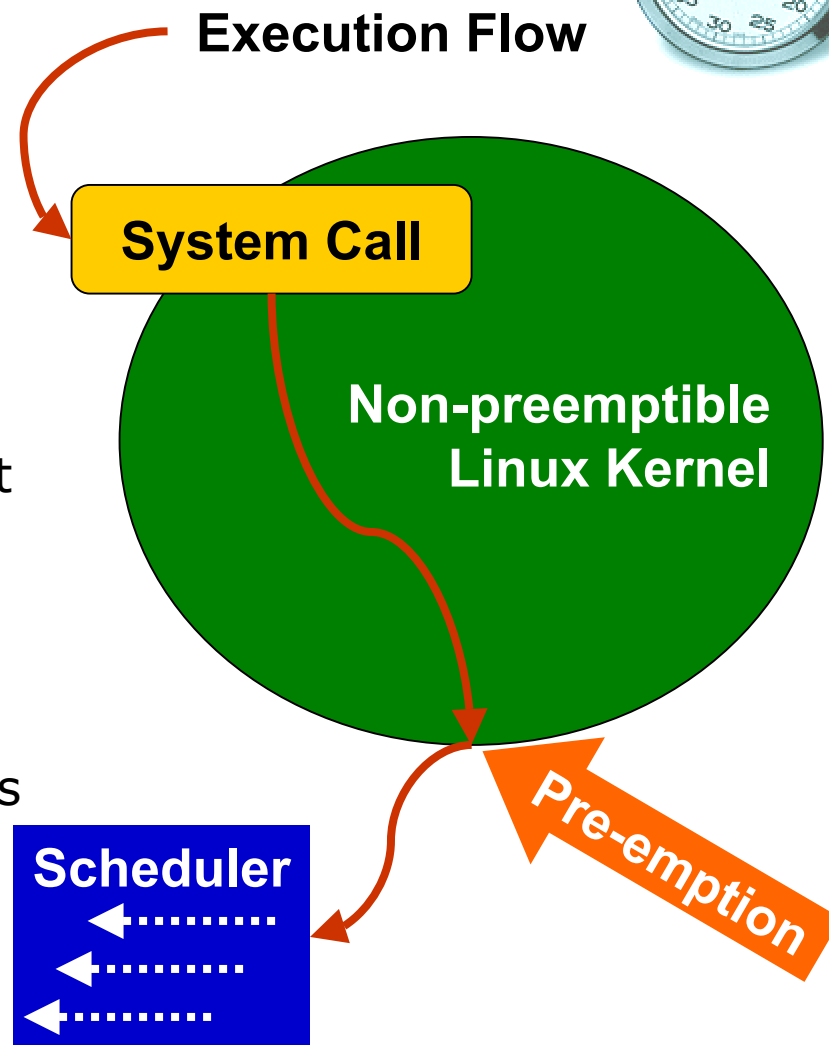


MONTAVISTA
SOFTWARE

Linux and Real-Time Linux Kernel Design

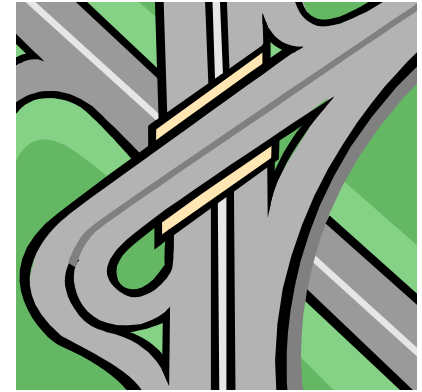


- Time-sharing style scheduling
 - Design intent
- Non-preemptible kernel
 - Optimized for throughput
- Driver-induced latency
 - Generic drivers
 - Back-to-back worst-cases





Linux and Real-Time Paths to Real-Time



- As-is : real-fast
- Performance Tuning
- Alternate Kernels
- Real-time with Native Linux



Linux and Real-Time Performance Tuning



- Traditional approach
 - Used with Solaris, NT, etc.
- Many applications deployed today
- Constrained real-time requirements
 - Small number of devices



Linux and Real-Time Sub-Kernel Approaches



- Use “RT” component or sub-kernel
 - Run Linux in virtual environment
- Options today
 - RTLinux and RTAI
- Problems
 - Lose robust Linux programming model
 - Proprietary APIs
 - Limited IPC with Linux applications



MONTAVISTA
SOFTWARE

Linux and Real-Time Make Linux More Deterministic

- Start by measuring actual performance
 - Maximum blocking times
 - Context switch
- Linux Real-Time Characterization Project
 - Interrupt latency measurement tool
 - See white papers and downloads at <http://www.mvista.com/realtime>

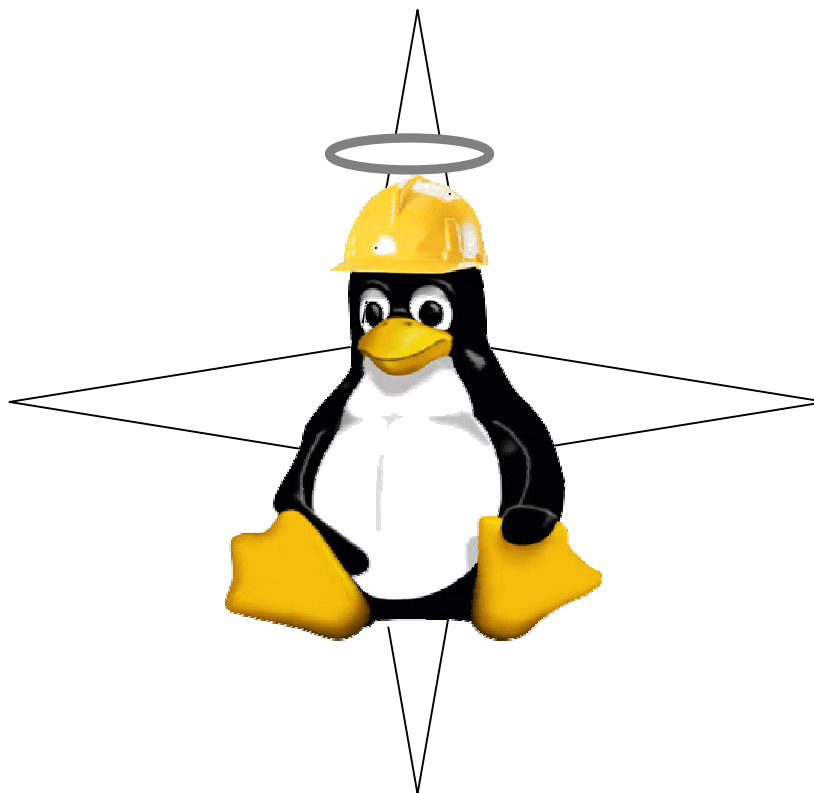




MONTAVISTA
SOFTWARE

Linux and Real-Time MontaVista Approach

Make Linux better. . .





MONTAVISTA
SOFTWARE

Linux and Real-Time Complementary Components

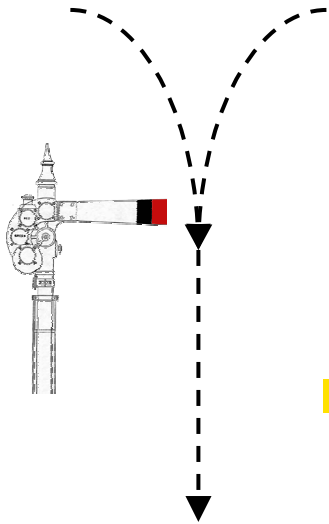
- Fixed priority scheduler
 - Runs in front of standard scheduler
 - Fixed overhead
 - Implements 128+ priorities for pthreads
 - Enforce scheduling policies
 - SCHED_FIFO and SCHED_RR policies
- Offers more stable, soft real-time
- Preserves robust Linux user programming model, APIs





Linux and Real-Time Preemptible Linux Kernel

- Leverages Linux 2.4 SMP kernel
 - Off-the-shelf Linux
 - No proprietary add-ons
 - Offers natural preemptibility
 - Spin locks → non-preempt flag macro
 - Hard Real-time attributes
- Preserves Linux programming model
 - User-level applications
 - Standard APIs

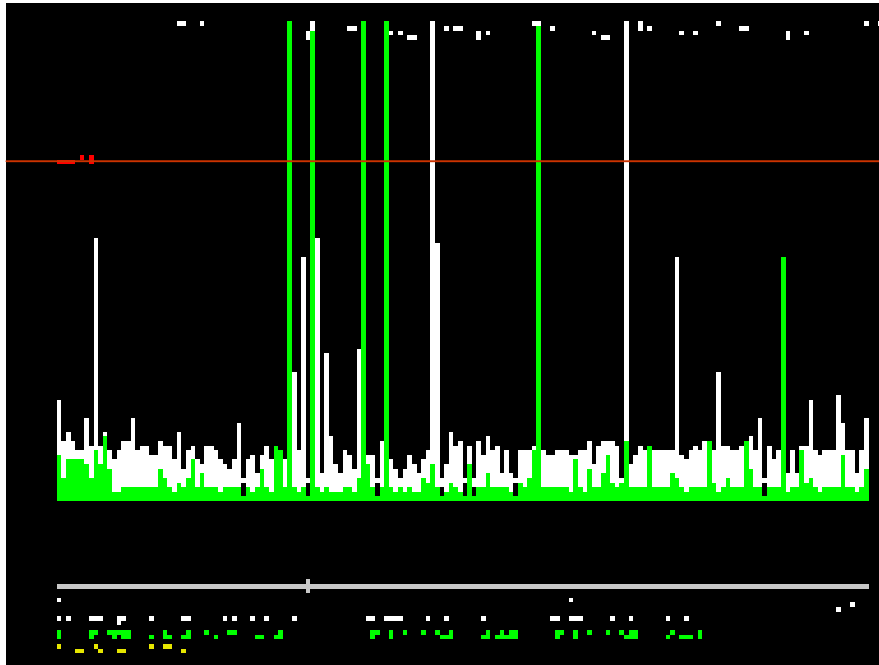




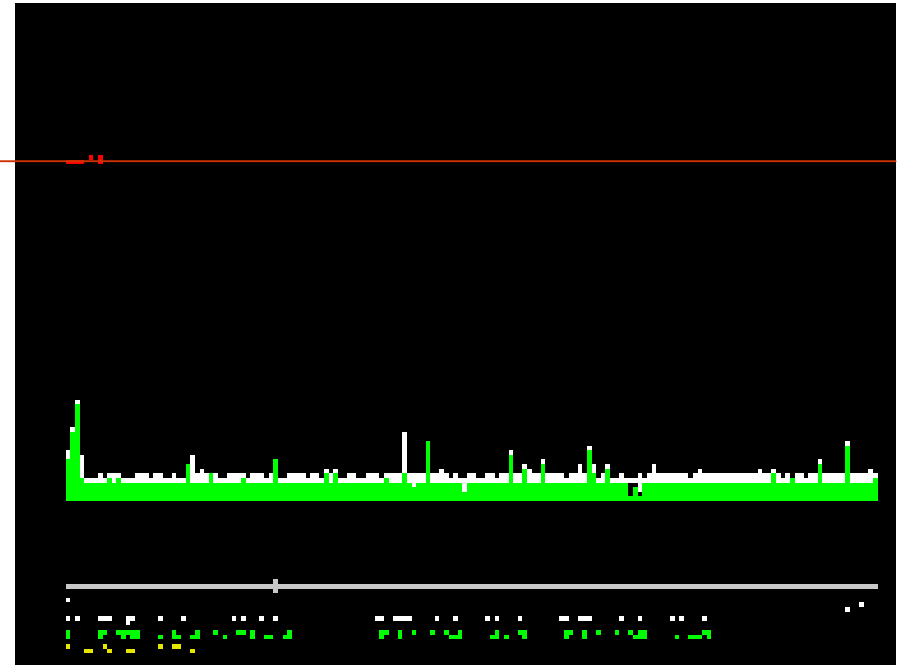
MONTAVISTA
SOFTWARE

Linux and Real-Time Audio Benchmark Example

- Audio Processing Under Load
 - Worst-case response times



Standard Kernel



Preemptible Kernel



MONTAVISTA
SOFTWARE

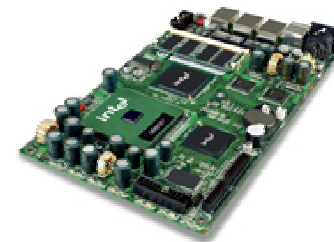
Interrupt Latency Data Collection Methods

- Kernel Instrumentation Patch
 - Architecture-specific
 - Measures ALL interrupts, keeps track of longest paths and module locations
- Methods
 - Load with LMBench and Netperf
 - Collect data for 1 hour

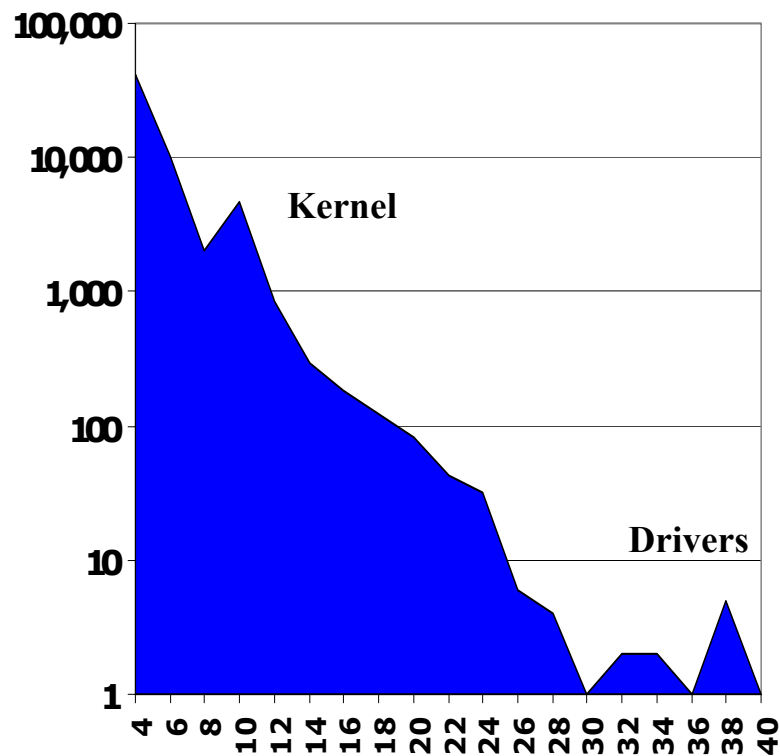


MONTAVISTA
SOFTWARE

Linux Interrupt Latency



- Hardware
 - Intel Pica*
 - 300 MHz Celeron
 - 32 MB w/IDE disk
- Load
 - Lmbench, Netperf
- Data Set
 - 730,575,267 samples
- Worst Case
 - 38 μ secs
- Majority
 - 98.998% < 2 μ secs



Remaining samples with
over 2 μ secs latency

*Entry-Level Communications Appliance Reference Design



MONTAVISTA
SOFTWARE

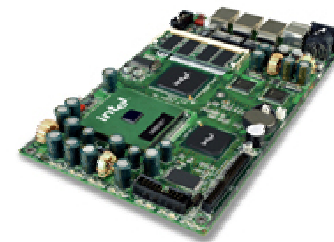
Preemption Latency Data Collection Methods

- Jitter Benchmark Version 1.0
- Jitter Basics
 - Real-Time Clock (RTC) produces 2Khz periodic interrupt
 - Jitter program pends 'read' operations on RTC /dev/rtc
- Jitter Method
 - Receives read
 - Retrieves system time (timestamp)
 - Calculates time from interrupt posted until jitter thread actually scheduled to read data
 - Interrupt can fall in kernel, driver, user code

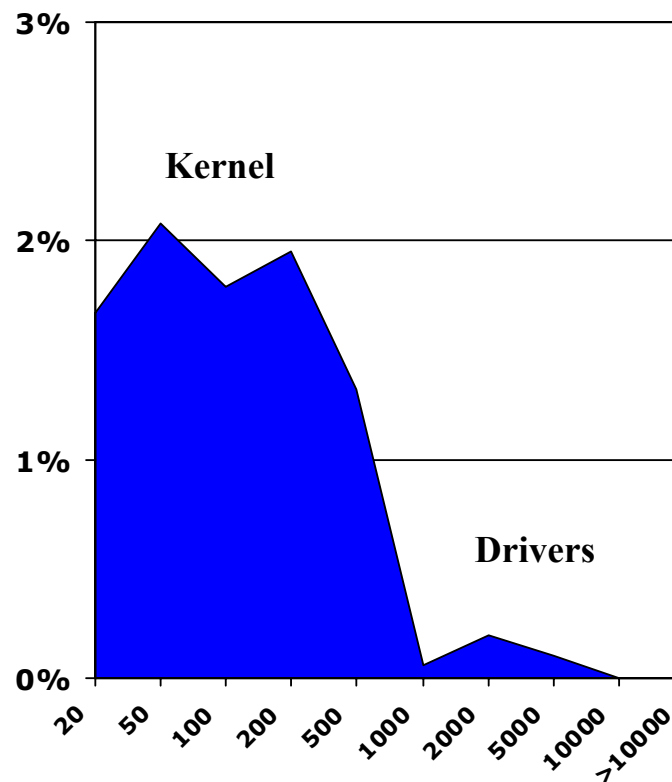


MONTAVISTA
SOFTWARE

Preemption Latency



- Hardware
 - Intel Pica*
 - 300 MHz Celeron
 - 32 MB RAM w/IDE disk
- Load
 - Lmbench, Netperf
- Data Set
 - 730,575,267 samples
- Worst Case
 - Loaded : 14.6 msecs
 - Unloaded : 84 μ secs
- Majority
 - Loaded : 90.84% < 10 μ secs
 - Unloaded : 99.47% < 10 μ secs



Remaining samples with over 10 μ secs latency (loaded)

*Entry-Level Communications Appliance Reference Design



MONTAVISTA™
SOFTWARE

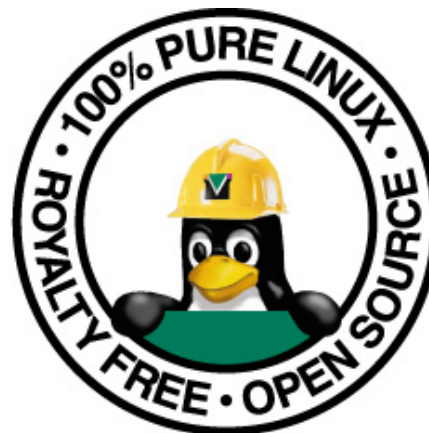
MontaVista and the Open Source Community



MONTAVISTA
SOFTWARE

MontaVista Basics

- **100% Pure Linux**
 - No “fork” – stay in close synch with developments in the Open Source Community
- **Royalty Free**
 - Improved COGS
- **Open Source**
 - No proprietary elements





MONTAVISTA
SOFTWARE

Open Source Community

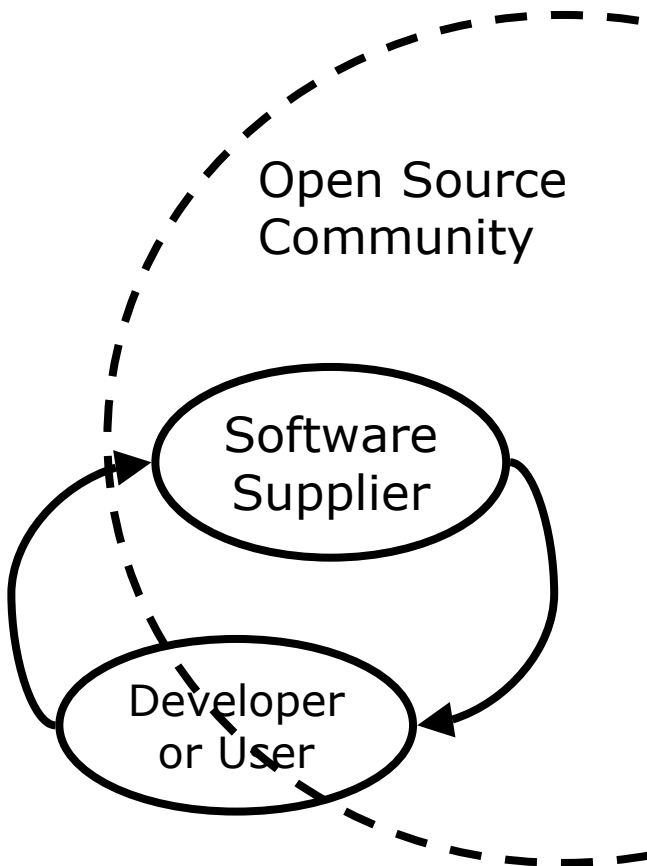
- Encourages open discussion of design & code
- Drives stress & corner testing
- Faster repair of defects
- More architectures & configurations
- Quick evolution of features
- High level of innovation
- Yielding raised expectations

**Active participation in the open source community allows
MontaVista to produce higher quality products faster**



MontaVista Model

MONTAVISTA™
SOFTWARE

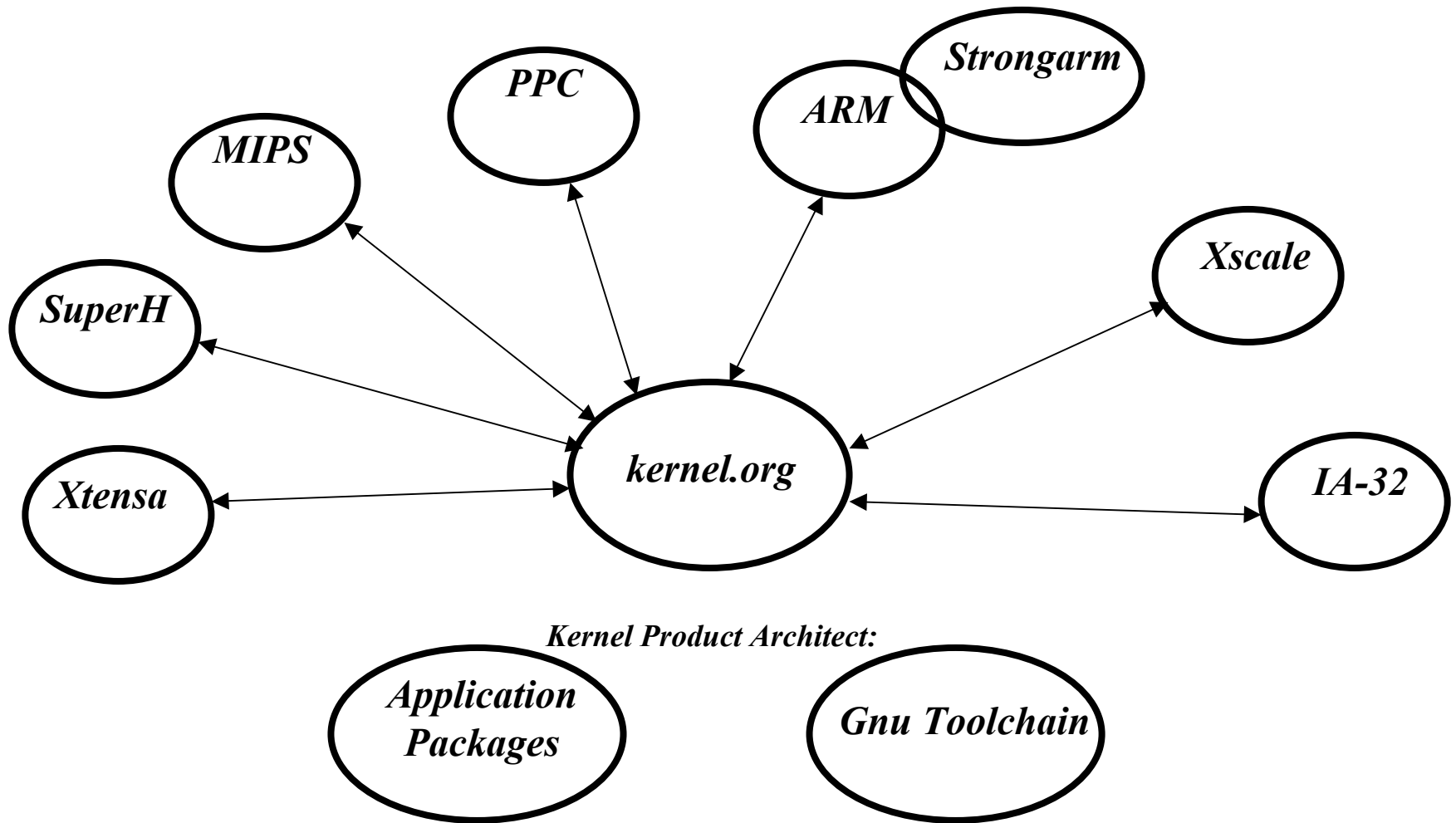


- Supplier doesn't "extract" technology from Open Source
- Supplier is PART of Open Source
 - Uses standard, non-forked sources
 - Contributes to and may even sponsor entire project trees
 - Injects new technology into community



MONTAVISTA
SOFTWARE

Open Source Community Trees and MontaVista Liaisons





MONTAVISTA[™]
SOFTWARE

Open Source Contributions

- Real-time Scheduler
 - Implements RTOS de-facto standard algorithm for realtime scheduling
- Preemptive Kernel Patch
 - Provides dramatic increase in process level responsiveness to events
 - February 2002 : accepted by Linus Thorvalds!
- Library Optimizer Tool
 - Optimizes use of library code to provide small executable footprint
- High Resolution POSIX Timers
- OS Emulators
 - VxWorks and pSOS



MONTAVISTA™
SOFTWARE

Conclusions and Future Direction



MONTAVISTA
SOFTWARE

Linux and Real-Time Future Direction

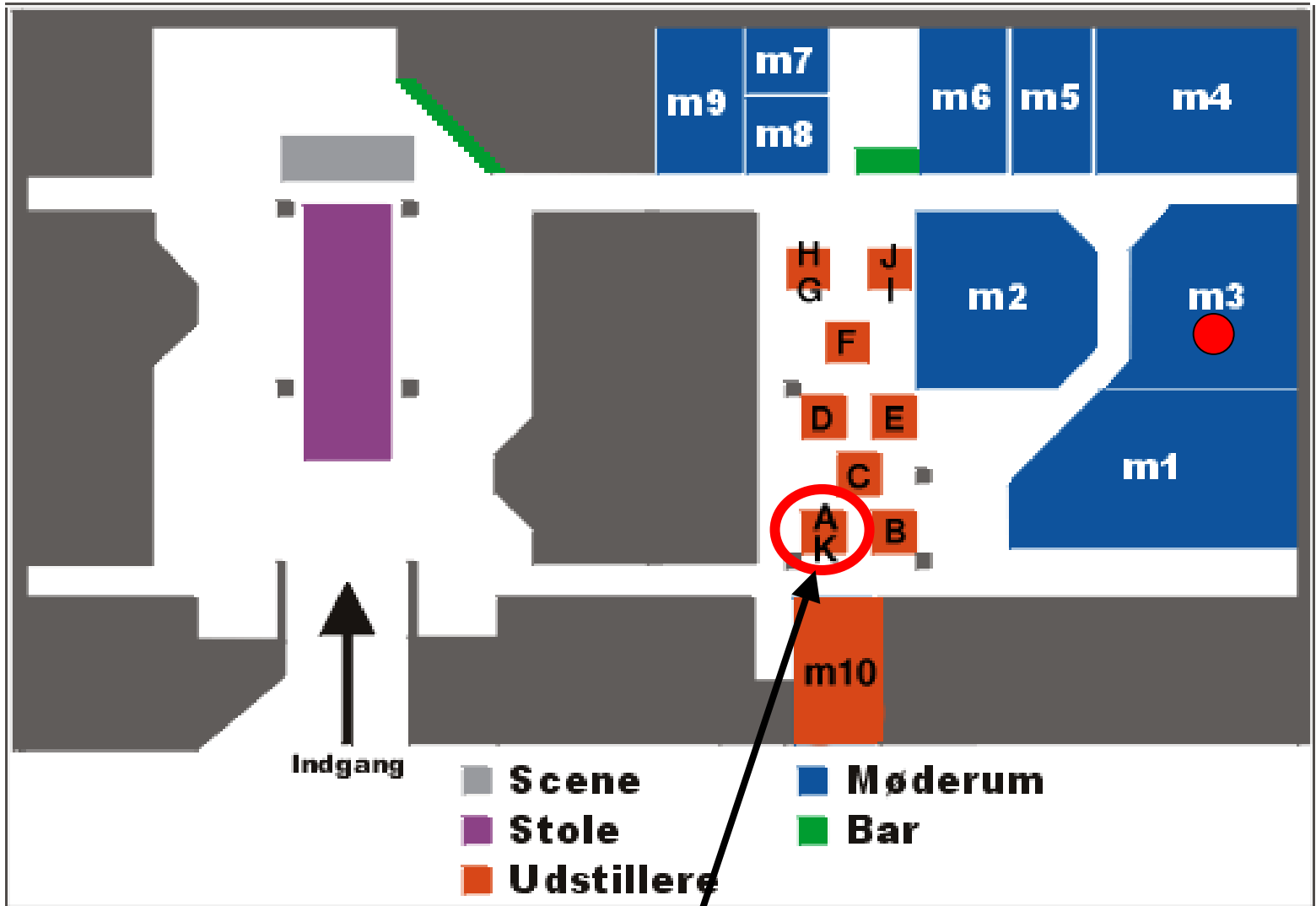


- Preemptive patch now ported across architectures
 - X86, PPC, MIPS, SH, ARM, XScale.
 - Make applicable to SMP hardware on x86
- Open Source Community
 - Preemptive patch already multi-developer project
<http://kpreempt.sourceforge.net/>
 - Merged by Linus Torvalds into main Linux development tree, beginning version 2.5.4-pre6
 - Enhancements
 - Tighter response times
 - Kernel
 - Drivers



MONTAVISTA
SOFTWARE

MontaVista at LinuxForum 2002



MontaVista



MONTAVISTA[™]
SOFTWARE

Embedded Linux is Ready For Real-Time!

