

# Real-Time in Linux



**MONTAVISTA**<sup>TM</sup>  
S O F T W A R E

*Powering the Embedded Revolution*

Xopen Real-Time Forum  
January 23, 2002  
Anaheim, California

Kevin Morgan  
VP of Engineering





**MONTAVISTA**  
SOFTWARE

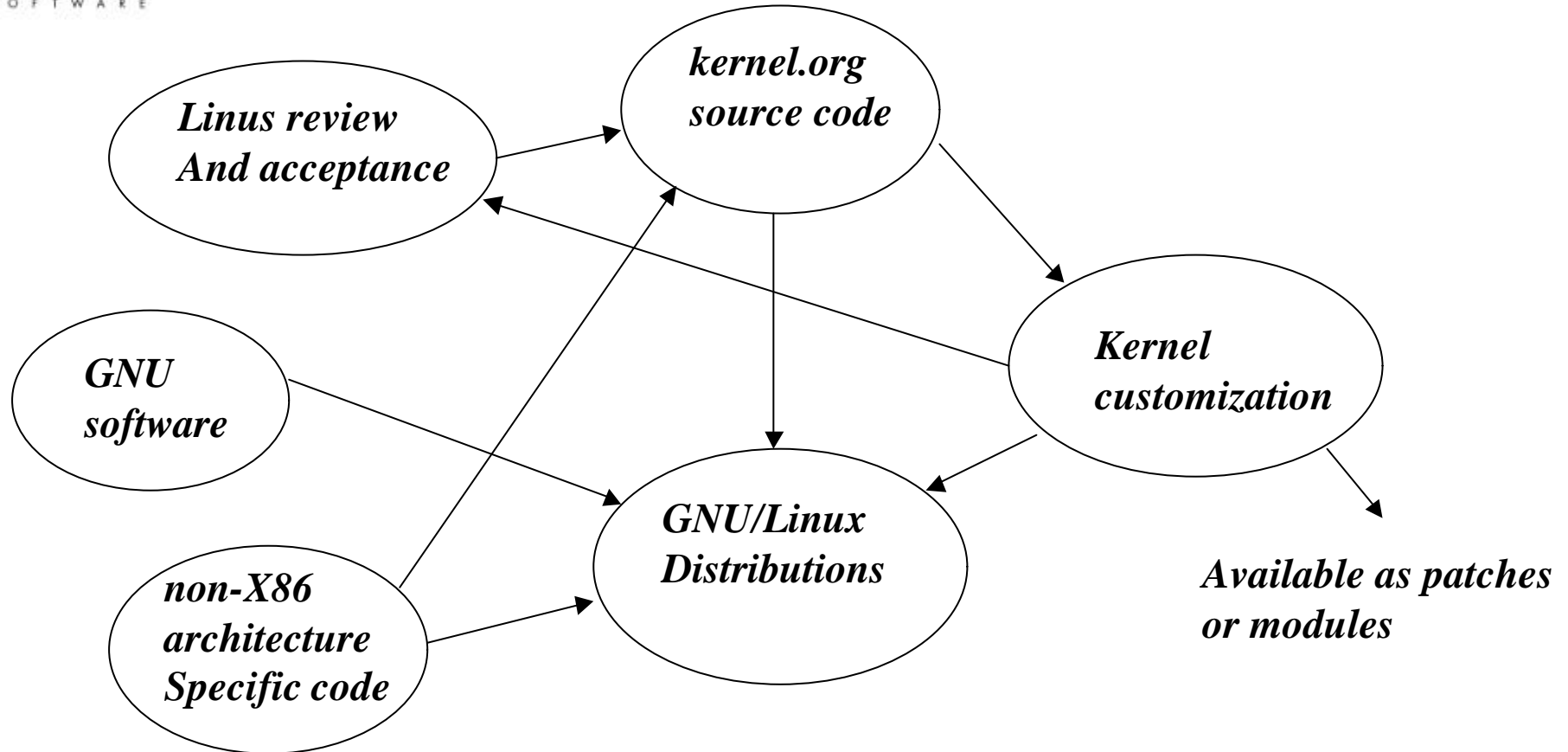
# What is Linux?

- A registered trademark owned by Linus Torvalds
- The kernel distributed by Linus through kernel.org.
- Kernel's derived from the kernel.org kernel.
- A misnomer for distributions of GNU software (utilities, apps and toolchains) bundled with the "Linux" kernel.



# What is Linux?

MONTAVISTA  
SOFTWARE



*So...is kernel.org Linux plus XFS (file system from Silicon Graphics) "Linux"?  
Is kernel.org Linux plus preemption patches "Linux"? YOU DECIDE...*



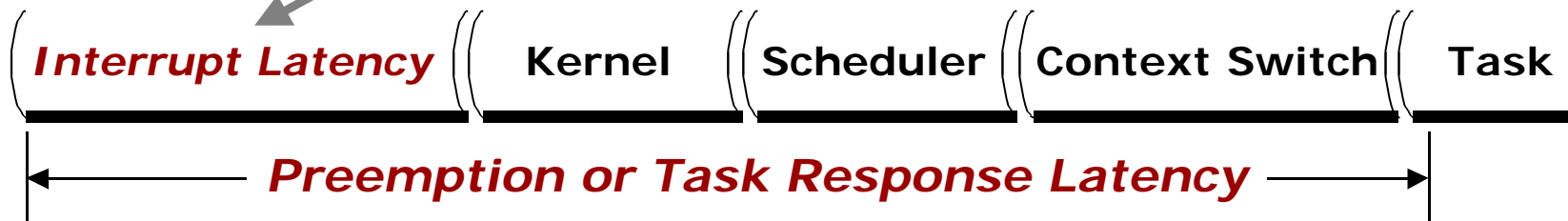
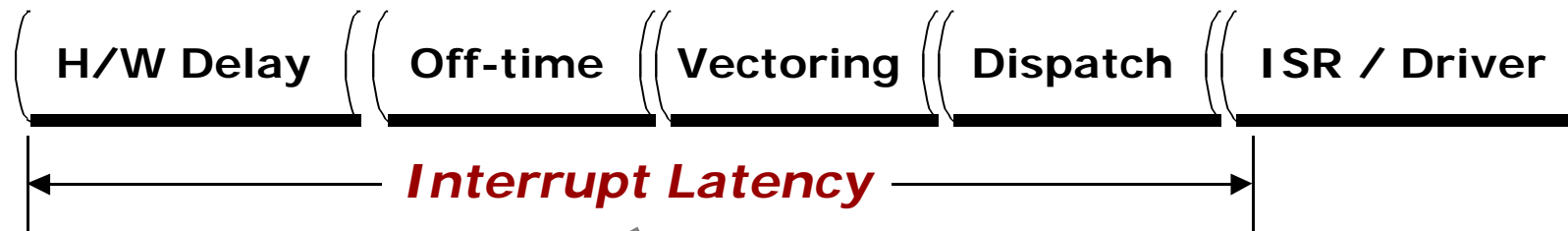
# Styles of Linux Customization

- Open source project program (code always available on line, multiple parties involved, etc.)
- GPL (or similar) licensing and general availability after development.
- GPL (or similar) licensing but availability only to “customers”.
- Dynamically loadable modules that may be proprietary/closed source.
- Submittal for inclusion in the Linux kernel proper.



MONTAVISTA  
SOFTWARE

# Definition of Terms



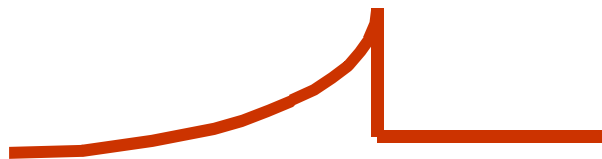


# Linux and Real-Time Definition of Terms



- 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





# Primary Linux Issues Re: Responsiveness

- Interrupt off periods
- Preemption off periods
- Time event resolution
- Bottom half driver structure
- Scheduler overhead
- Pre-allocation of resources
- “Proof” of responsiveness



# General MontaVista Approaches

- Focus on improving Linux (versus providing hybrid solutions).
- System measurement and provision of measurement tools.
- Open source project improvements
- Inclusion in MontaVista GNU/Linux product
- Submission of appropriate technology to Linus Torvalds for consideration of inclusion in kernel.org Linux.



# Specific MontaVista Efforts (to date...)

- Measurement and publishing of interrupt off times and measurement tools (open source)
- Tuning of some interrupt off times
- Fixed overhead real-time process scheduler (open source)
- Kernel preemption (open source)
- Clock event resolution improvement ("high res timers")
- Measurement of process response latency under selected loads



# Technology Status Updates - Scheduler

- Real-time scheduler available for most major embedded architectures for 2.4.x
- 2.5 has incorporated an  $O(1)$  scheduler from Ingo Molnar that replicates RT attributes of MV's scheduler, with better multiprocessor CPU affinity attributes. (Patch available for 2.4 as well.)



# Technology Status Updates - Preemption

- Preemption patch available for most major microprocessor architectures.
- MV rolling out standard Linux product with preemption available on all supported architectures.
- Preemption highly likely to go into 2.5 soon...(Kevin predicts).
- SMP spinlock path length reductions and SMP defect corrections reducing 2.4 worst case latencies.



# Preemption and Red Hat's Perspectives

- Vendors have a responsibility to innovate and provide differentiated value.
- The open source process is all about massive parallel innovation and "best in class" feedback into the "official Linux".
- The open source community and open source users "decide" what is "go forward technology", not a particular vendor.
- Maintenance costs? MontaVista pays zero maintenance cost for preemption technology...
- RTLinux solves a different problem than kernel preemption (= > Steve Brosky's talk).
- Prediction: before the end of 2003, all Linux vendors will deliver preemptible kernels (2.6.x)



# Technology Status Updates – Bottom Halves

- Change at 2.4.7 (or thereabouts) from bottom halves run prior to processes to a dynamic system
- Bottom half execution switched to kernel thread execution (at priority sched\_other) if overload conditions detected.
- More work is needed here to provide flexible prioritization of bottom halves versus processes.



# Technology Status Update

## Hi Res Clocks/Timers

- John Mehaffey will cover this later today.
- Microsecond level precision on time event delivery (with interrupt off skewing).
- Posix real-time timer API's.
- All open source, available today via sourceforge.



# Latency Measurements Software Specification

- Kernel
  - MontaVista HardHat Linux 2.0  
(based on kernel version 2.4.2)
- Compiler
  - GNU gcc version 2.95.3
- Load
  - LMbench and Netperf

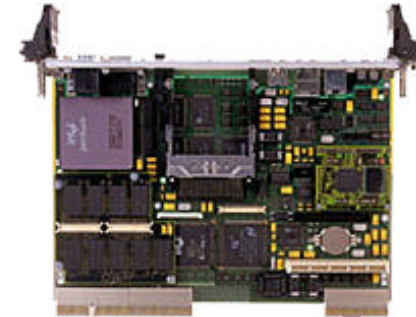


# Performance Test Environment

Host



Target



Hub

- Host and Target Systems
- Private Ethernet subnet connected through a hub (no gateway or switch or other traffic).



# Platform Specifications x86-based Targets

<b>System:</b>	<b>Intel Entry-Level Comm. Appliance Reference Design (Pica)</b>	<b>Ziatech ZT5550 CompactPCI System Slot Controller</b>
<b>CPU:</b>	<b>300 MHz Intel Celeron</b>	<b>266 MHz Pentium III</b>
<b>Cache:</b>	<b>128K L2 On-chip</b>	<b>16K ins / 16K data 512K L2</b>
<b>Storage:</b>	<b>32 MB RAM, IDE disk</b>	<b>128 RAM, IDE disk</b>
<b>Network:</b>	<b>100BaseT Ethernet</b>	<b>100BaseT Ethernet</b>



# Platform Specifications

## PPC and MIPS Targets

<b>System:</b>	<b>IBM PPC 405 (Walnut)</b>	<b>MIPS Malta</b>
<b>CPU:</b>	<b>200 MHz PowerPC 405GP</b>	<b>100 MHz MIPS R5000</b>
<b>Cache:</b>	<b>16K ins / 8K data</b>	<b>32K ins / 32K data</b>
<b>Storage:</b>	<b>32 MB RAM NFS root</b>	<b>32 MB RAM NFS root</b>
<b>Network:</b>	<b>100BaseT Ethernet</b>	<b>100BaseT Ethernet</b>



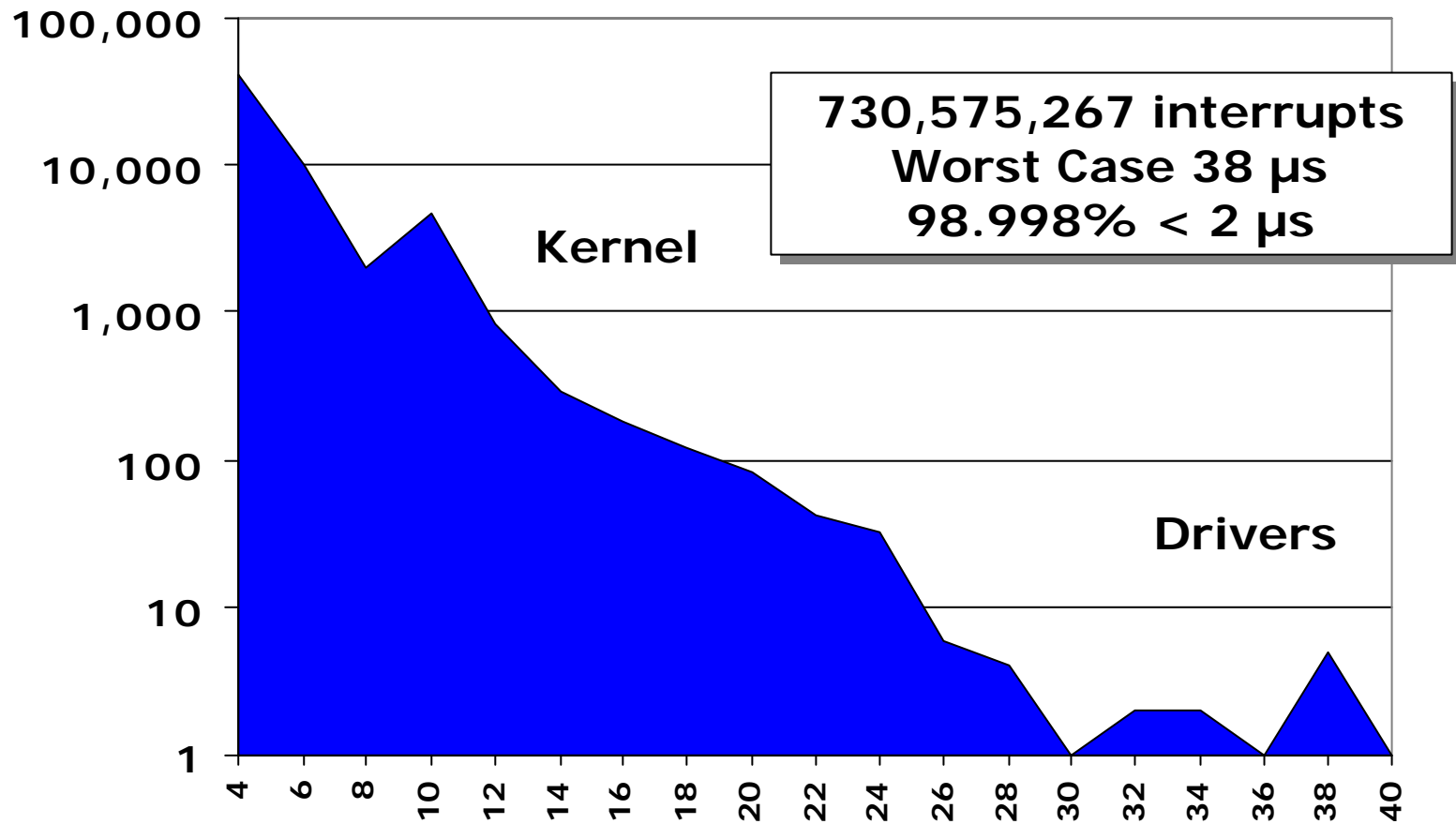
# Interrupt Latency Data Collection Methods

- 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



# Interrupt Latency

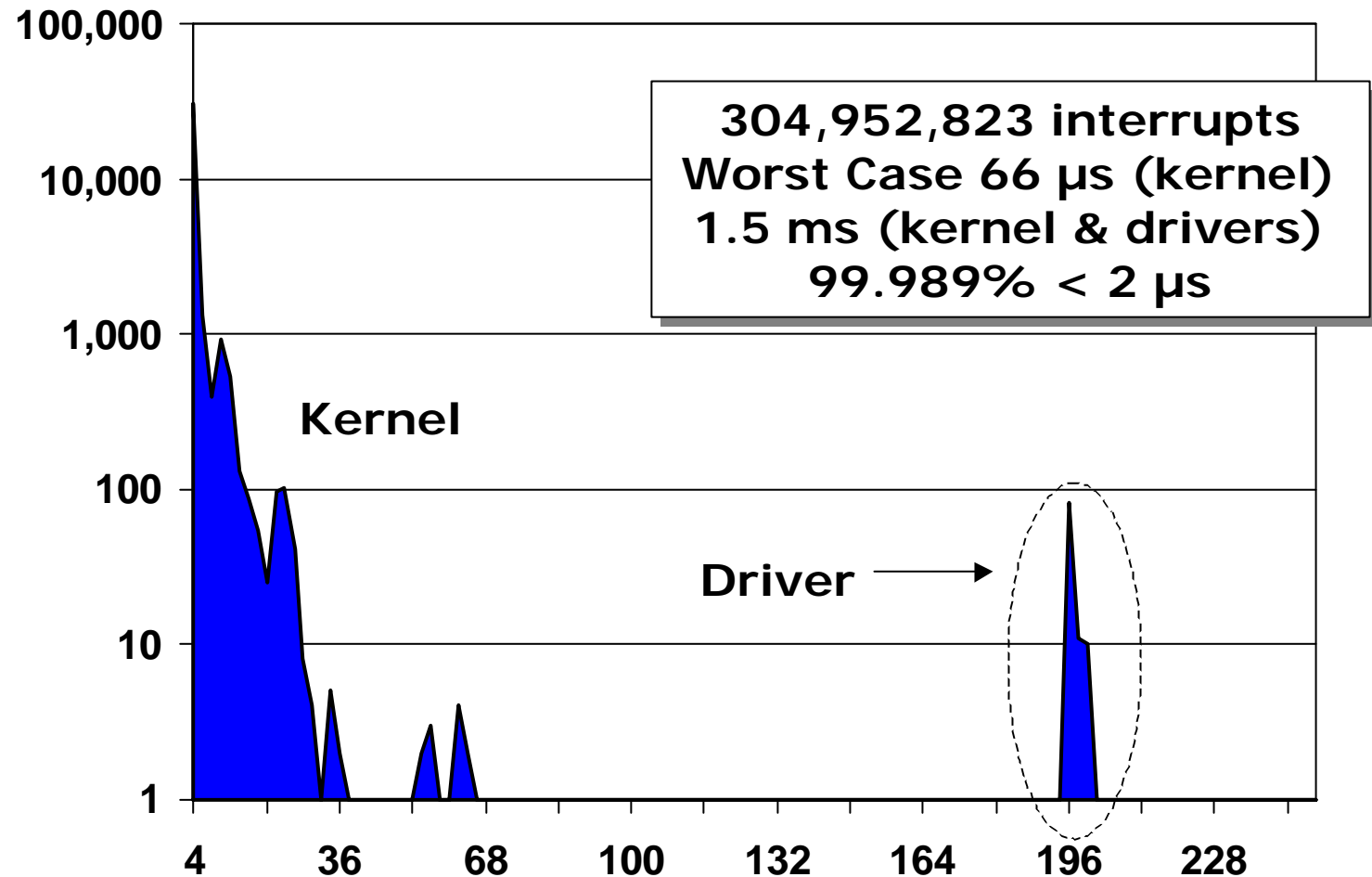
## Intel Pica - 300MHz





# Interrupt Latency

## Ziatech ZT5550 - 266 MHz

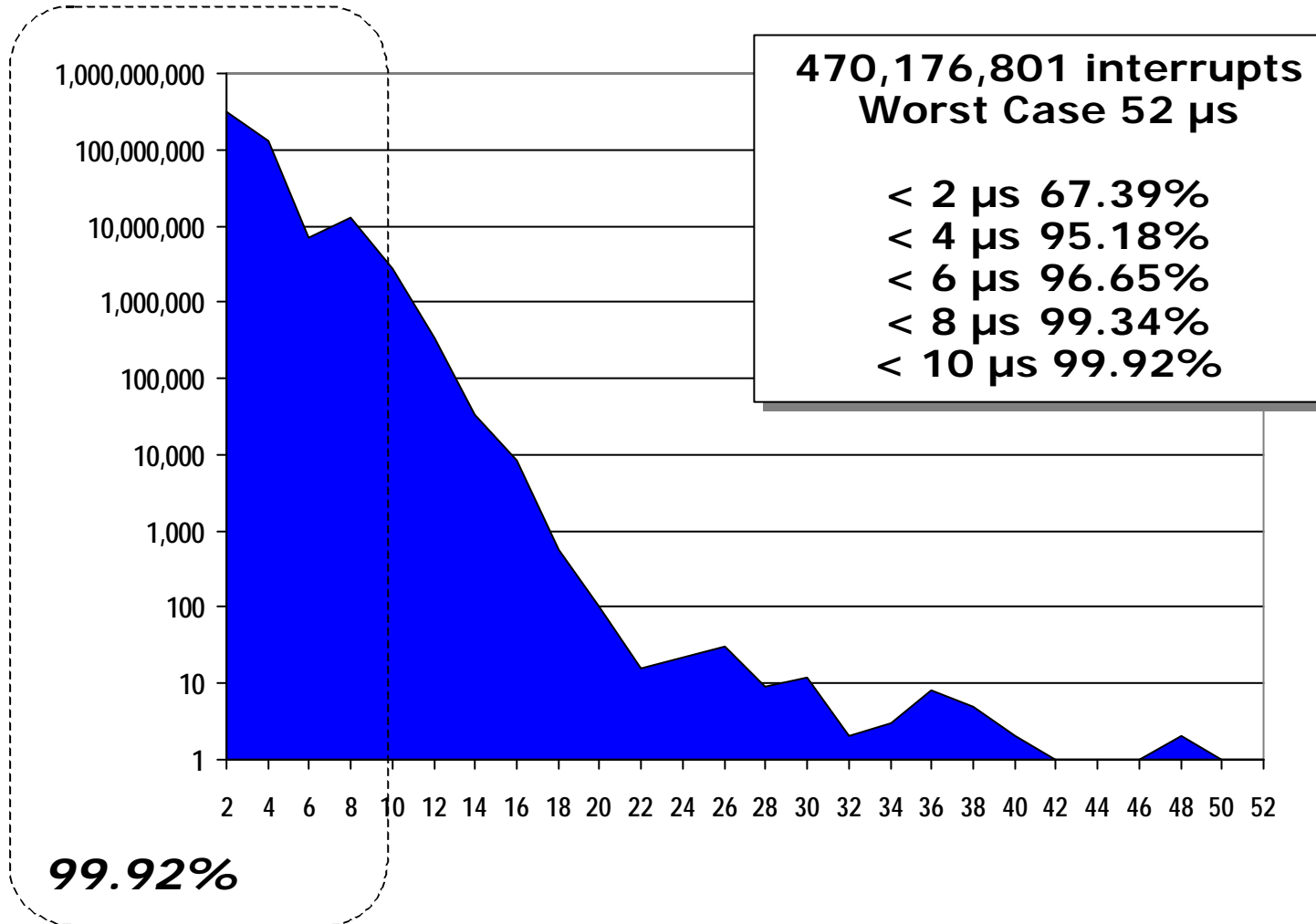




MONTAVISTA  
SOFTWARE

# Interrupt Latency

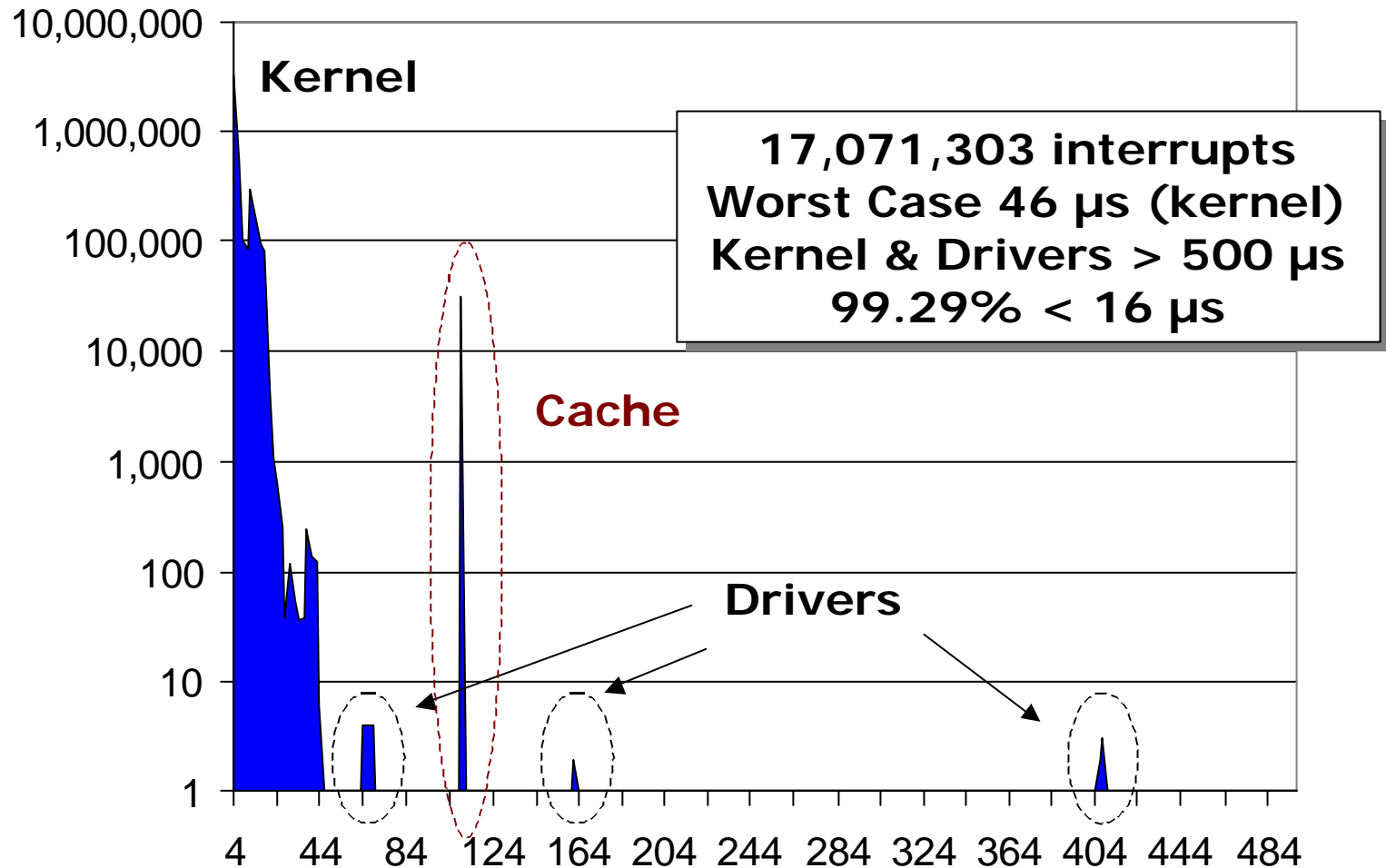
## IBM PPC 405GP - 200 MHz





# Interrupt Latency

## MIPS Malta – 100 MHz





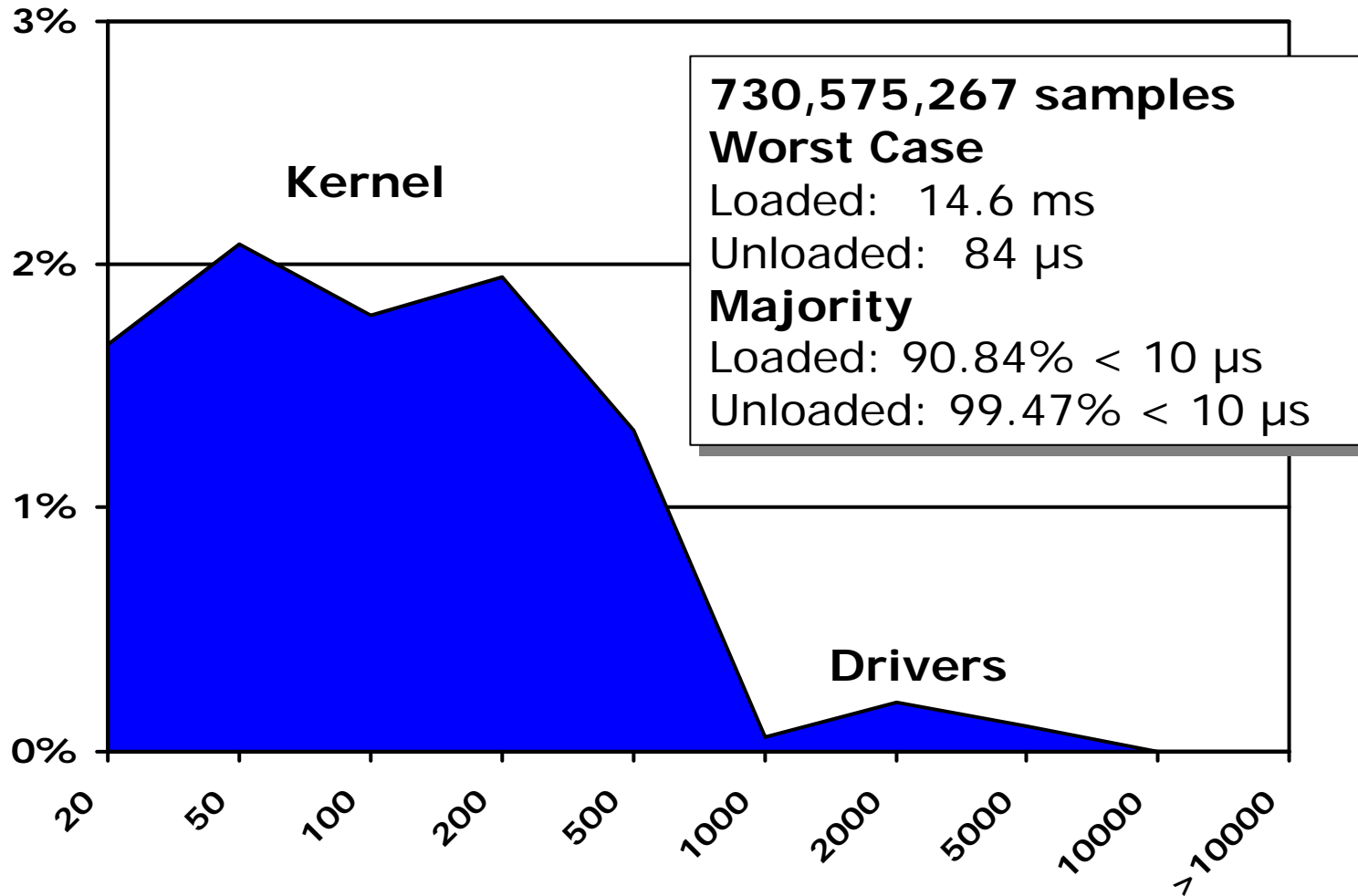
# Preemption Latency Data Collection Methods

- 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



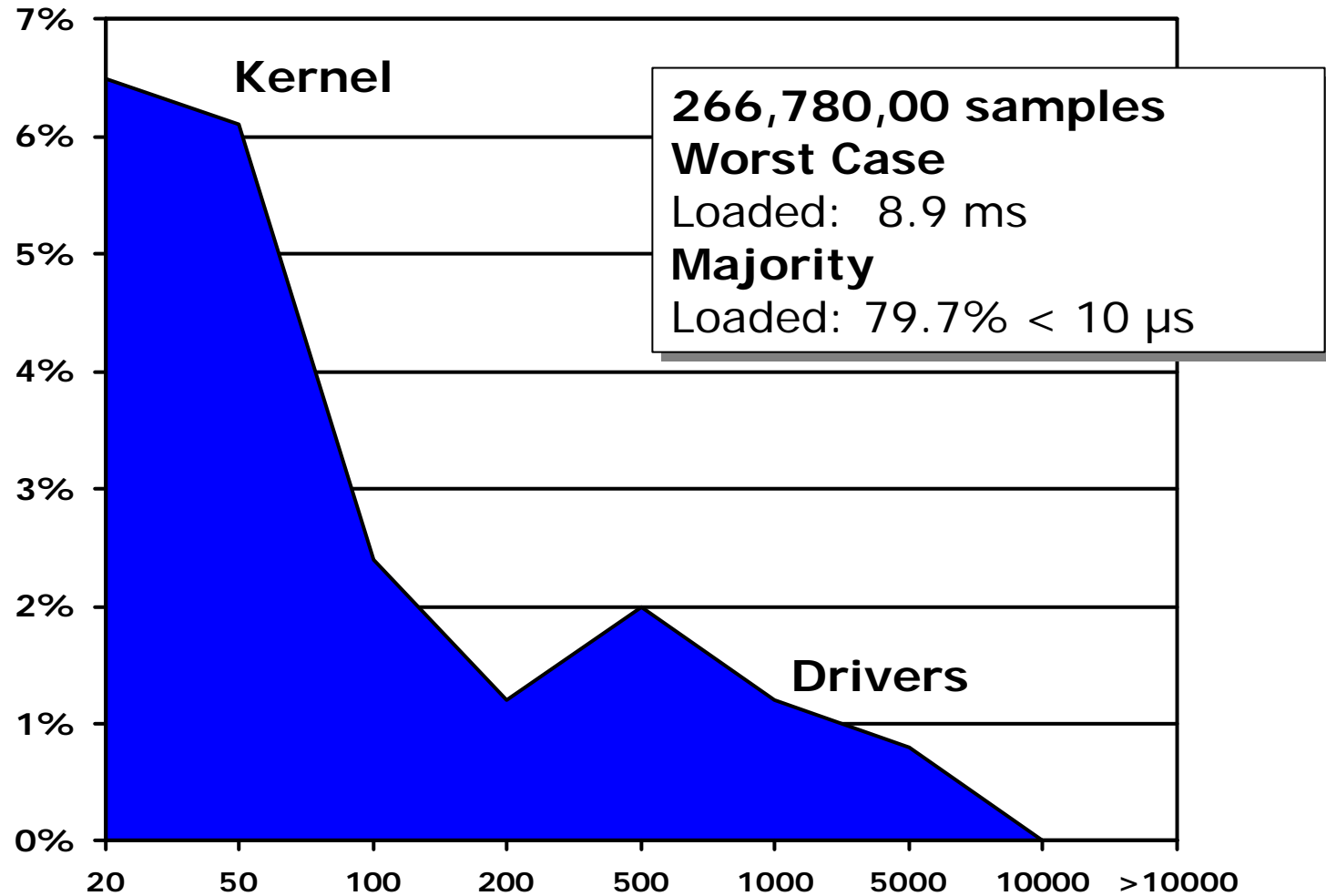
MONTAVISTA  
SOFTWARE

# Preemption Latency Intel Pica





# Preemption Latency Ziatech ZT5550





**MONTAVISTA**  
SOFTWARE

# LMBench

- Kernel Performance Benchmark
- Bandwidth Benchmarks
  - Cached file read, memory copy (bcopy), memory read and write, pipe, and TCP/IP
- Latency Benchmarks
  - Process context switch, network connection establishment, pipe, TCP, UDP, and RPC “hot potato”, file creation/deletion, process creation, signal handling, system call overhead, memory read latency
- Miscellaneous Benchmarks
  - Processor clock rate calculation
- Using LMBench version 2.0



**MONTAVISTA**  
SOFTWARE

# LMBench

## System Call Tests

Test CPU	Null call	Null I/O	Stat	Open/ Close	Select TCP	Sig inst	Sig hndl	Fork proc	Exec proc	Sh proc
Pica	1.0 $\mu$ s	1.6 $\mu$ s	5.9 $\mu$ s	9.0 $\mu$ s	n/a	3.0 $\mu$ s	9.5 $\mu$ s	580 $\mu$ s	3.5 ms	12 ms
ZT5550	1.15 $\mu$ s	1.74 $\mu$ s	7.17 $\mu$ s	14 $\mu$ s	52 $\mu$ s	3.24 $\mu$ s	9.7 $\mu$ s	542 $\mu$ s	3.3 ms	14 ms
EP405	1.56 $\mu$ s	3.88 $\mu$ s	29 $\mu$ s	53 $\mu$ s	n/a	6.06 $\mu$ s	27 $\mu$ s	1.7 ms	16 ms	67 ms



MONTAVISTA  
SOFTWARE

# LMBench

## Context Switch Test

No. Programs	2	2	2	8	8	16	16
Data Seg Size	0 K	16 K	64K	16 K	64K	16 K	64K
Pica	3.0 $\mu$ s	20 $\mu$ s	-	46 $\mu$ s	-	66 $\mu$ s	-
ZT 5550	1.97 $\mu$ s	25 $\mu$ s	77 $\mu$ s	26 $\mu$ s	134 $\mu$ s	28 $\mu$ s	228 $\mu$ s
EP 405	21 $\mu$ s	110 $\mu$ s	307 $\mu$ s	123 $\mu$ s	305 $\mu$ s	122 $\mu$ s	304 $\mu$ s

- LMBench also provides a measure of context switch
- Context switch numbers not comparable to RTOS
  - Switch between Linux processes, including "touches" of all cache lines of the data segment.
  - TLB and cache disruption impacts included in measurement.



**MONTAVISTA**  
SOFTWARE

# Conclusions...

- Responsiveness and general real-time features improving quickly in "Linux"
- These improvements are allowing Linux to address a broader class of applications and provide an alternative to traditional proprietary RTOS solutions.
- The best enhancements will, through the open source process, get adopted into Linus's kernel.
- Not all important features will go into Linus's kernel, but will still be important and widely used.
- There's always more work to do...