SUSE® Linux Enterprise Real Time Extension

Kai Dupke
Senior Product Manager
kdupke@suse.com

Libor Pecháček
Project Manager
lpechacek@suse.com

Topics

SUSE® Linux Enterprise Real Time Extension

- Overview
- Recent Updates
- Customers
- Roadmap
- Feature Details

Challenge

Real Time

- Do you lose money if the system timing is wrong?
- Do you need
  - Deterministic command execution
  - Deterministic timing
  - Low latency response on events
  - Low latency communication

Can you afford delayed execution?

Real Time Extension

Overview

Overview

SUSE Linux Enterprise Real Time Extension

- An extension to SUSE Linux Enterprise Server
- Provides deterministic low latency performance for time-critical applications
- Industry-standard real-time version
  - Kernel preemption
  - CPU shielding
  - Task prioritization
  - Priority inheritance
  - Interrupt threads
  - Open Fabrics Enterprise Distribution

Overview

SUSE Linux Enterprise Real Time Extension

- Latency sensitive Workloads
  - Guaranteed process response
  - Sub-millisecond latency
  - Immediate resource access for time critical processes
- Low jitter
  - Repeated execution in the same time
  - Precision Time Protocol
- Separate Processes
  - Hierarchical priority scheme
  - CPU shielding
  - Task prioritization
- Low Latency Communication
  - 10G ethernet / Infiniband
  - TCP offload
- LTTng 2.1
  - Application tracing

Target Customers

SUSE Linux Enterprise Real Time Extension

Manufacturing, Telecommunication, Finance

Using SUSE for standard workloads

Customized applications

Replace proprietary and embedded Real Time

Test latency impact on workloads

Recent Updates

Service Pack 3

SUSE Linux Enterprise Real Time Extension

- Update to same 3.0 kernel as SUSE Linux Enterprise Server
  - Small patchset
  - Focus on RT capabilities
  - Share hardware enablement
- Precision Time Protocol (PTP)

Customers

Customers

SUSE Linux Enterprise Real Time Extension

Jülich Supercomputer Center

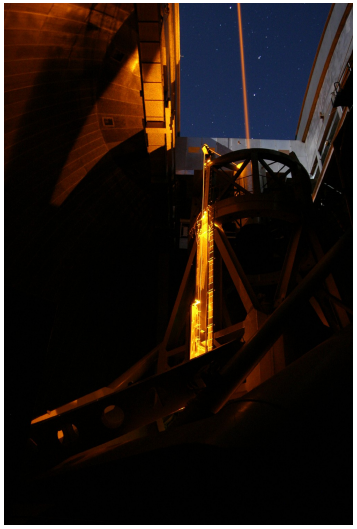
Thyssen Krupp Electric Steel

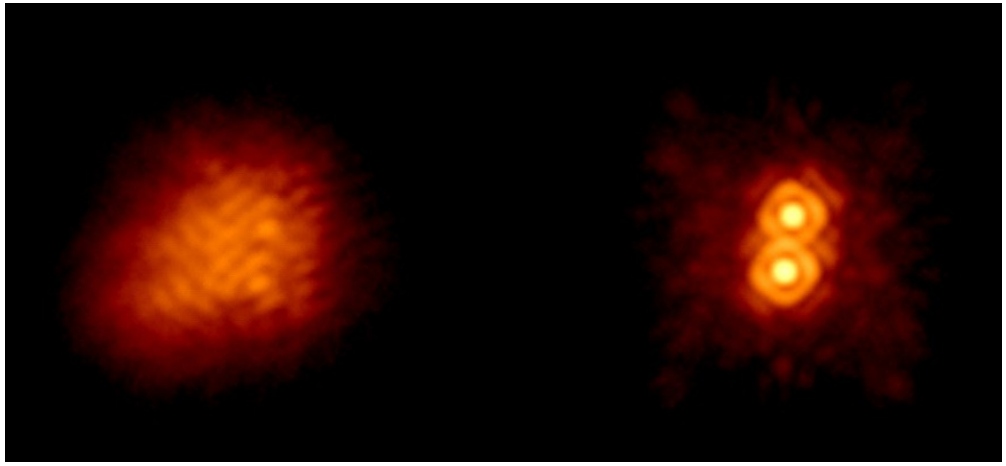
UMB Financial Corp.

NASA JPL Telescope

NASA JPL

SUSE Linux Enterprise Real Time Extension



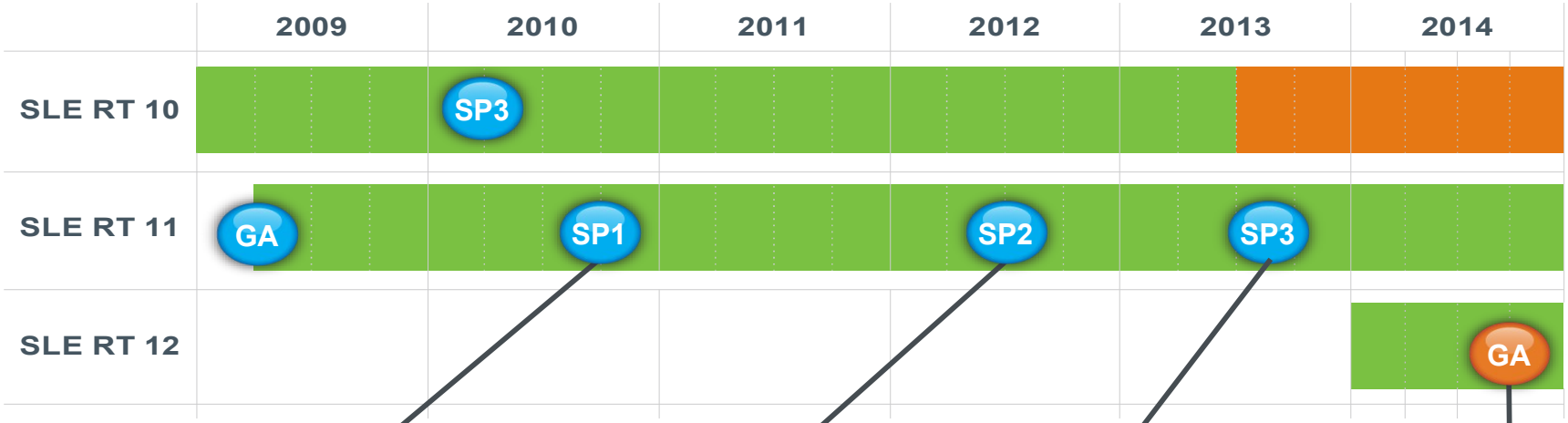


- 200-inch telescope with adaptive optics on Mount Palomar
- Avoid atmospheric blurring in Real Time
- Control more than 3000 mirror segments with a latency <250 msec

Roadmap

Roadmap

SUSE® Linux Enterprise Real Time Extension



SLE RT 11 SP1

- PREEMPT_RT Kernel
- Per device IRQ threads
- Enhanced scheduler
- Tuning tools:
  cset, rt-test,perf

SLE RT 11 SP2

- Same Kernel as SLES
- HW enablement shared with SLES
- Tracing tools:
  - LTTng
  - Eclipse Plug-in
- NVIDIA drivers

SLE RT 11 SP3

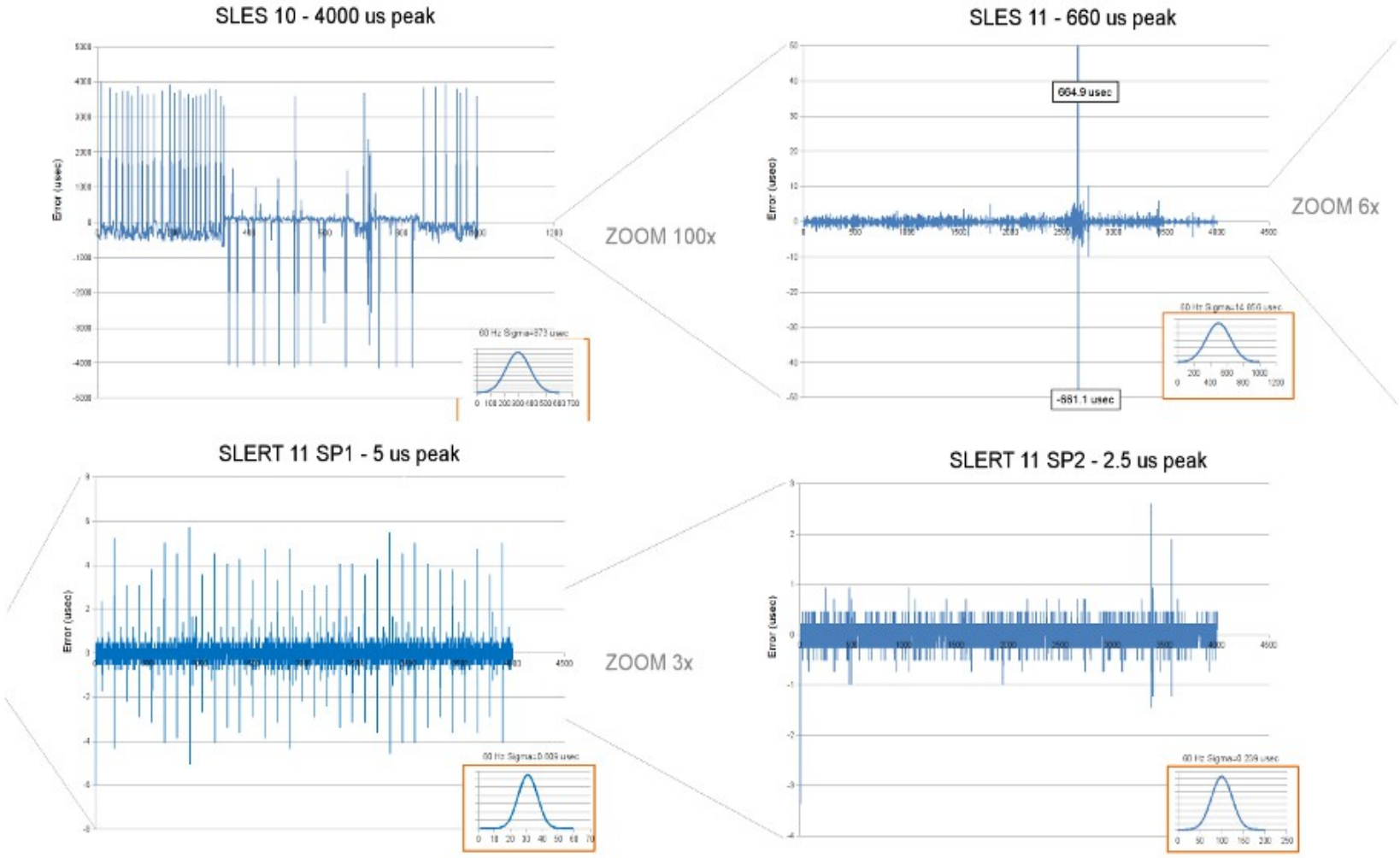
- LTTng update
- Precision Time Protocol

SLE RT 12

- Based on SLES 12
- GPU computing
- Match throughput of non-RT

Customer Timer Benchmark

SUSE Linux Enterprise Real Time Extension



Improvement: Factor 1,800!

Feature Details

Kernel Preemption

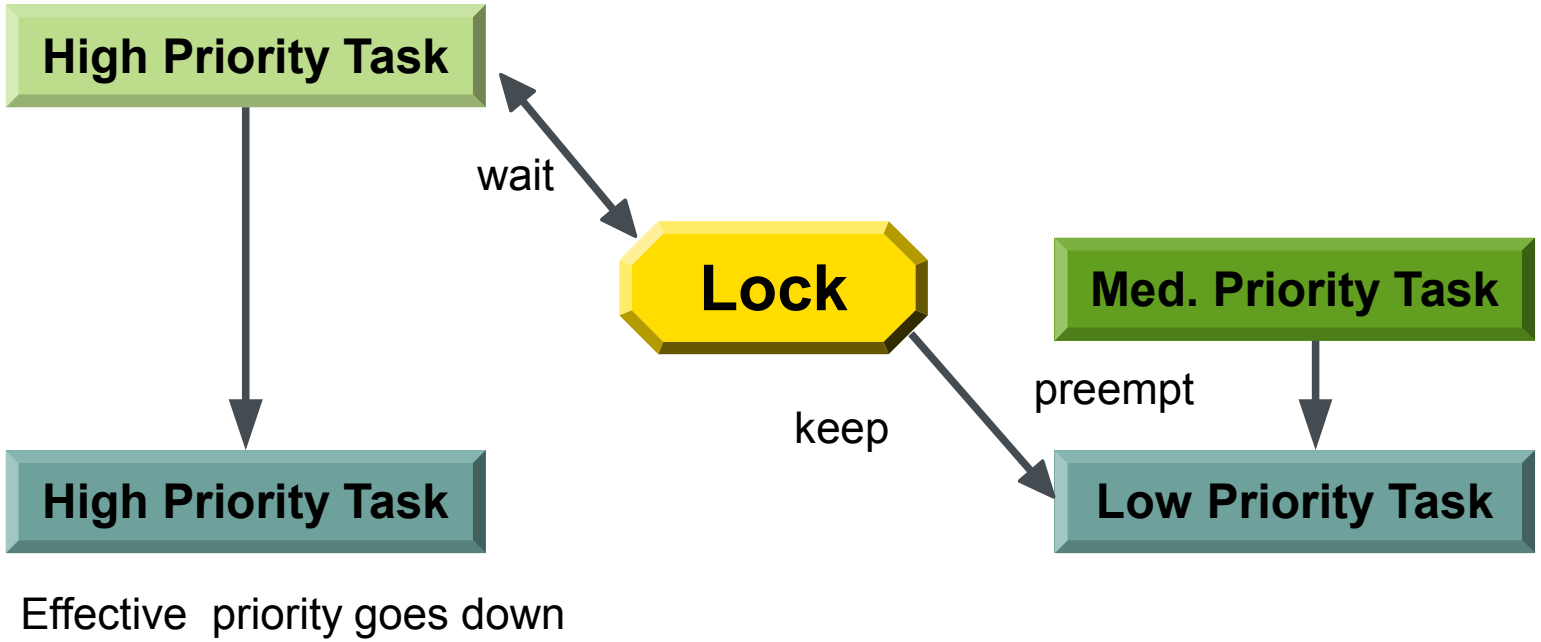
SUSE Linux Enterprise Real Time Extension

- Time critical tasks get immediate access to the CPU
- Real-Time Kernel improvements
  - Priority Inheritance Mutexes substitute non-premptive spinlocks (enables preemption for critical Kernel sections)
  - adaptive locking
  - read-write locks converted
  - preemptive interrupt handlers in thread context

Priority Inversion – Problem

SUSE Linux Enterprise Real Time Extension

- High priority task wait for lock release of low priority task initiated by medium priority task

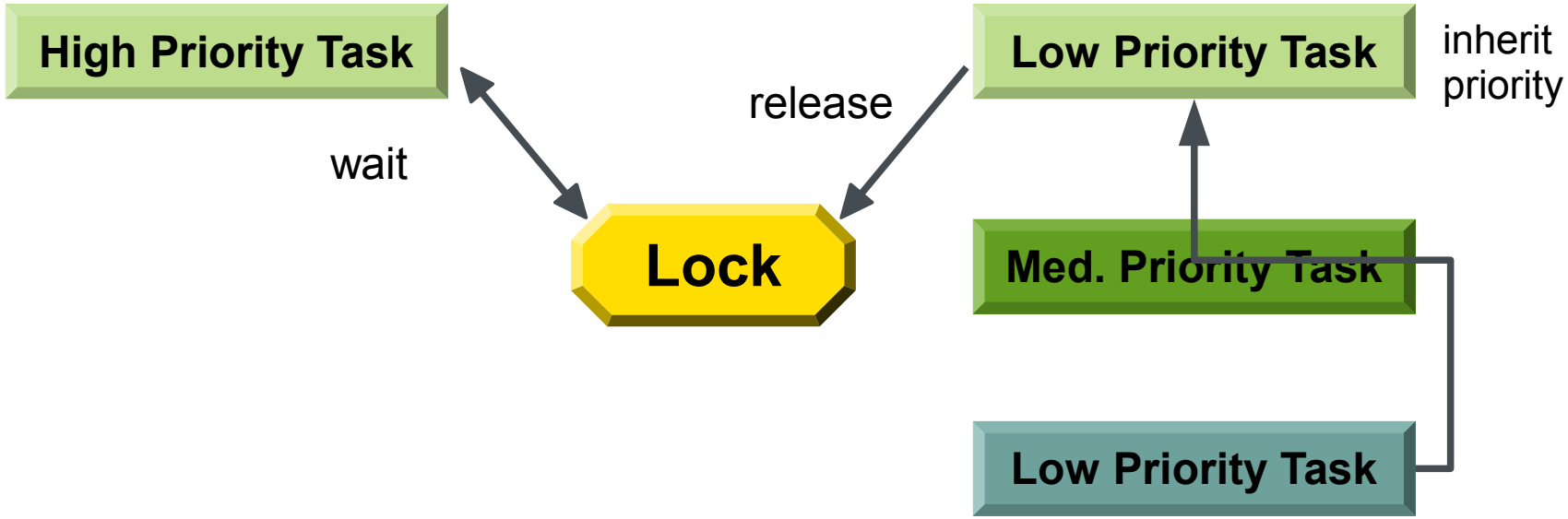


Effective priority goes down

Priority Inversion – Solution

SUSE Linux Enterprise Real Time Extension

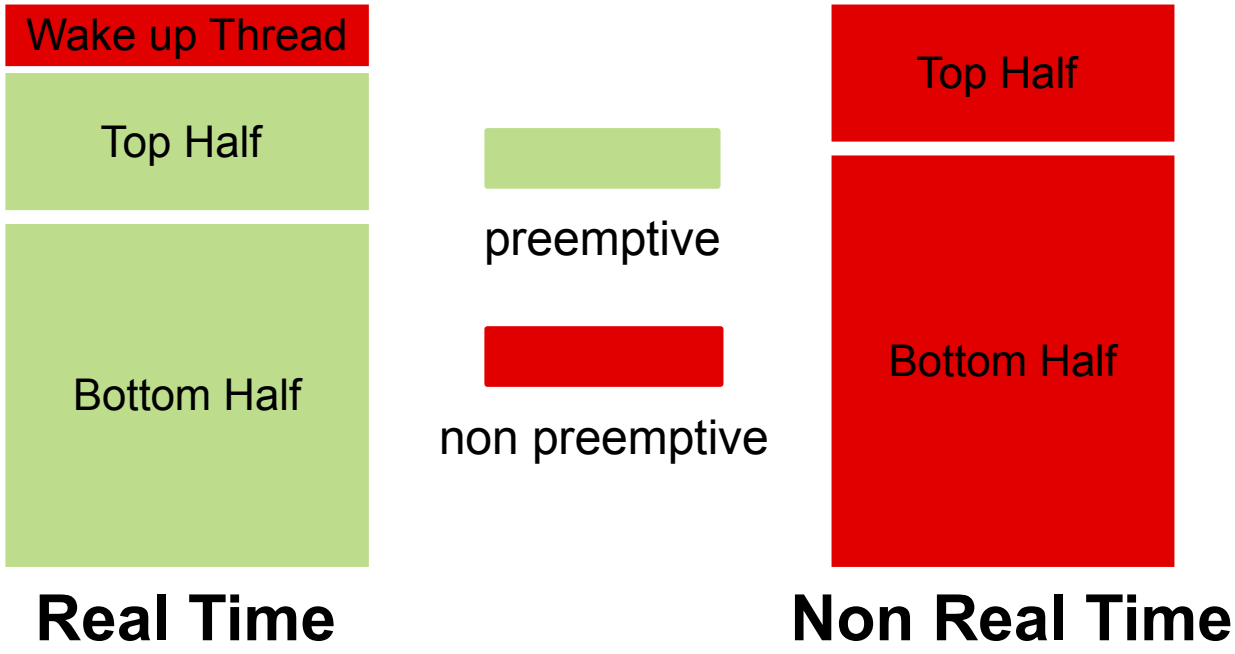
- Priority Inheritance
  - Boost priority of low priority process until the lock is released



Interrupt Threads

SUSE Linux Enterprise Real Time Extension

- Convert interrupts into threads
  - Prioritization possible
  - Preemption possible
  - Leverage CPUSets



Prioritization

SUSE Linux Enterprise Real Time Extension

- Tasks and interrupts can be prioritized
- Optimized schedulers
  - FIFO & ROUND_ROBIN for real time tasks
  - OTHER for non-real time tasks
  - Manipulated by `chrt(1)`
- Threads can be prioritized above interrupts

Profiling, Tracing

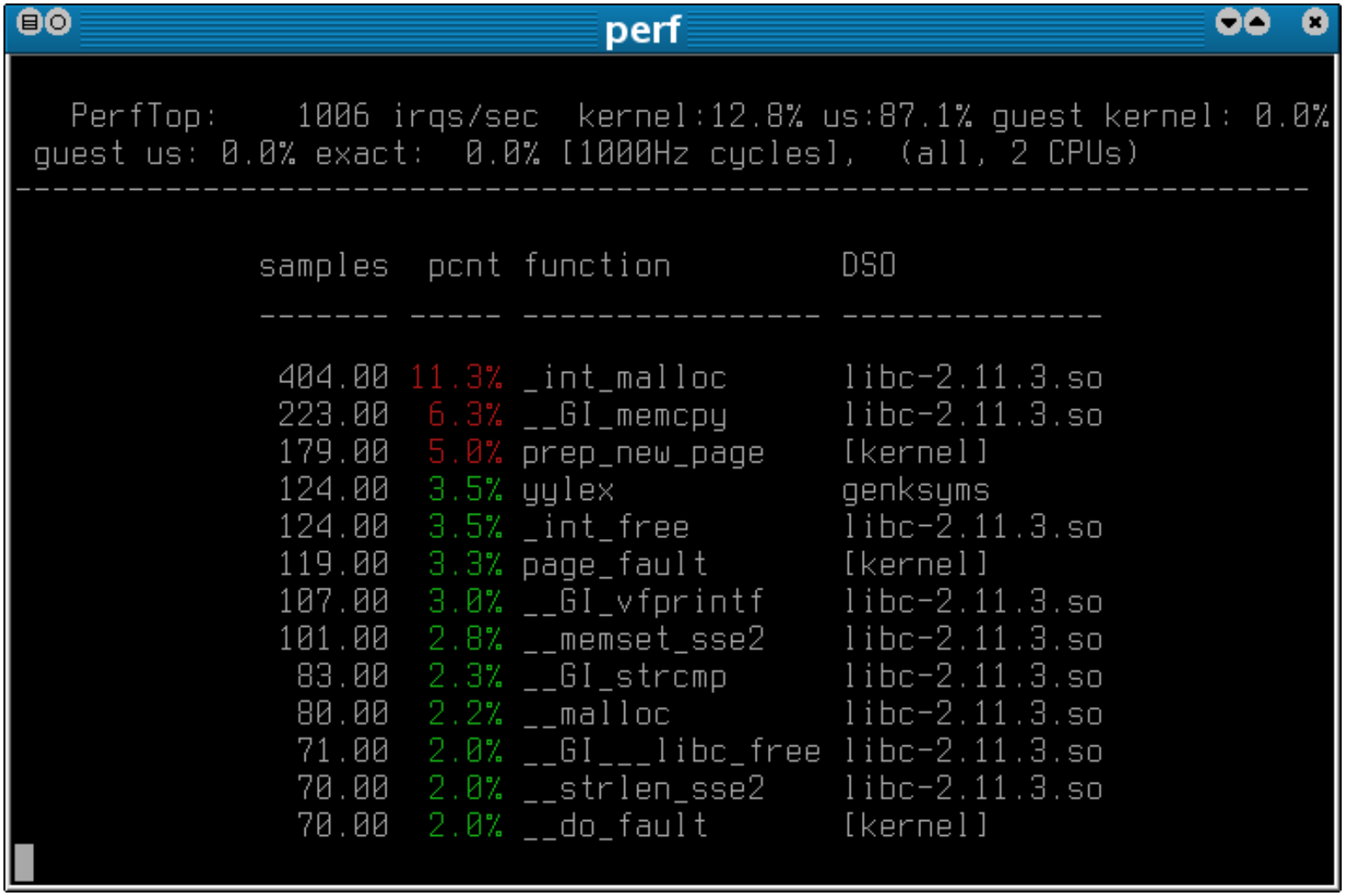
Profiling, Tracing

- Profiling
  - Statistics, interested in big picture
- Tracing
  - Like logging
  - Interested in **detailed** sequence of events

Oprofile and perf

- Stochastic profilers
- Leverage hardware support
- Transparent background operation while collecting information
- Suitable for finding bottlenecks in real-world systems
- `perf top`

perf top



Kernel compilation running

Tracing

ftrace

- In-depth view of kernel function calls
- Function tracer
- Function graph tracer
- Deepest stack, preemption disabled, *latency threshold*
- Capture trace upon oops or panic
- All these features available with `kernel-trace`

trace-cmd – ftrace frontend

- Ftrace is controlled through debugfs by default
- trace-cmd simplifies capturing traces
- … and also viewing them

SystemTap

- Scripted tool for examining live Linux systems
- SystemTap script lists events and associated handlers which execute an action
  - Probe can be inserted inside a function
- Script compiled into kernel modules
- Can access and alter arbitrary structures

LTTng

- Kernel and userspace tracing
- Minimal performance impact
- Kernel instrumentated in key subsystems
  - Scheduler, timers, signal, IRQ, block
  - Dynamic trace points
- Application instrumentation points
  - May be added into signal handlers and libraries

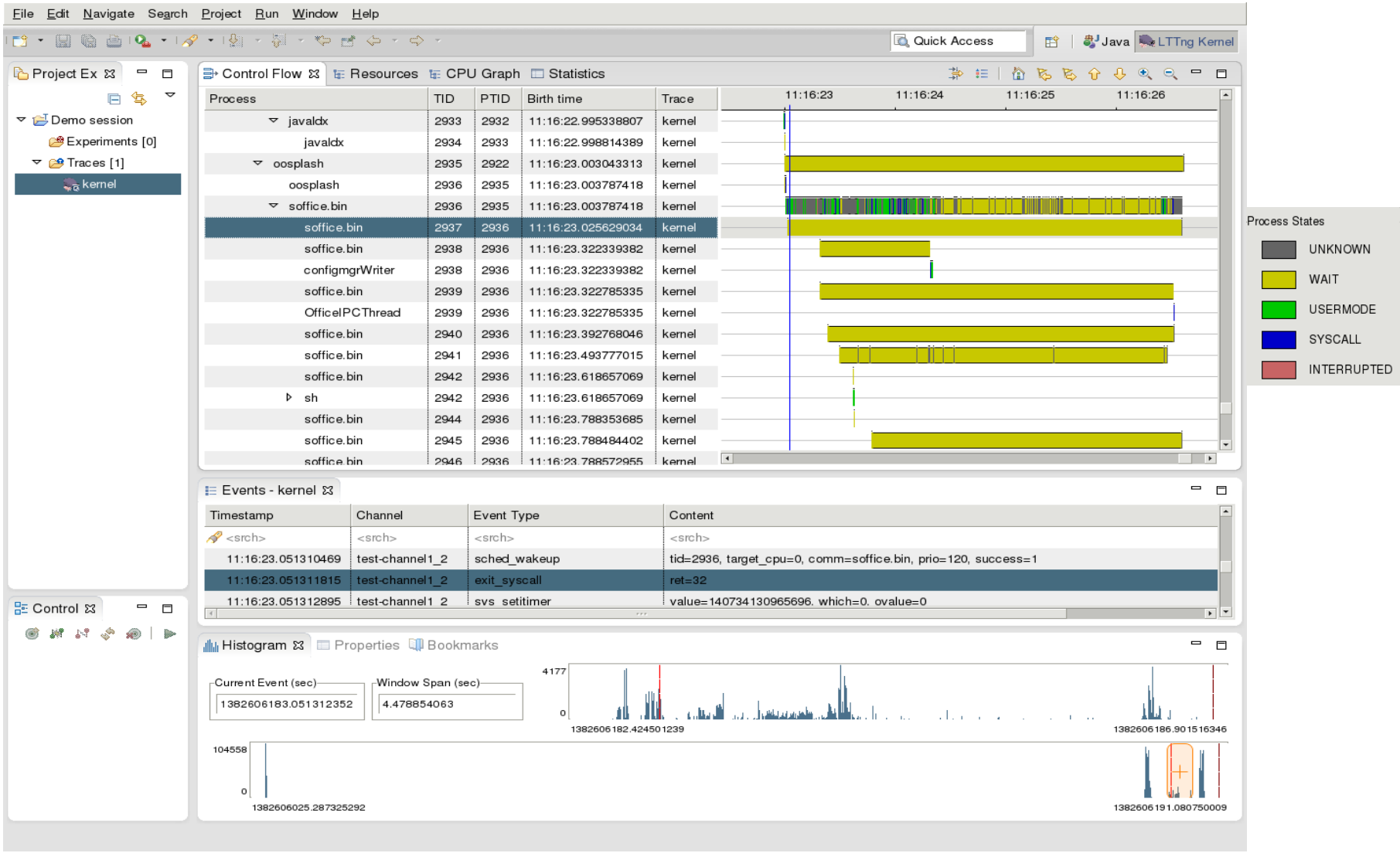
LTTng 2.1 - New features

- Network streaming
  - A new component: `lttng-relayd`
- Userspace tracing (UST)
- Flush pending records to storage upon `lttng stop`
  - Can be overriden on command line

LTTng Eclipse Viewer

- Eclipse plugin available for graphical viewing
- Following kernel-oriented analysers are offered:
  - Control Flow - to visualize processes state transition
  - Resources - to visualize system resources state transitions
  - Statistics - to provide simple statistics on event occurences
- SUSE specific
  - CPU Flow – to visualize what each CPU has been doing

LTTng Eclipse Viewer



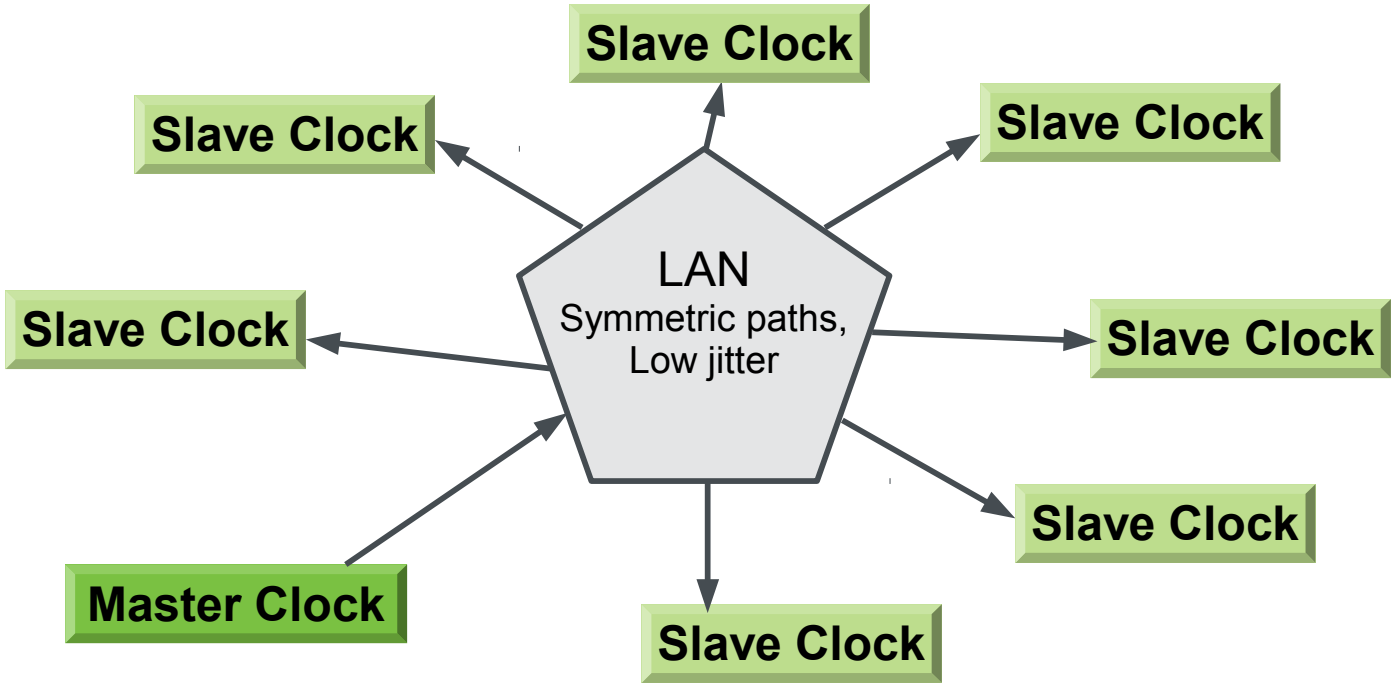
Precision Time Protocol

Precise Timekeeping

- Across a group of hosts (c.f. NTP)
- Leverages hardware support
  - Dedicated network interface card oscillators
- Master – slave mode of operation
  - Master is elected dynamically within the group
- Sub-microsecond synchronization

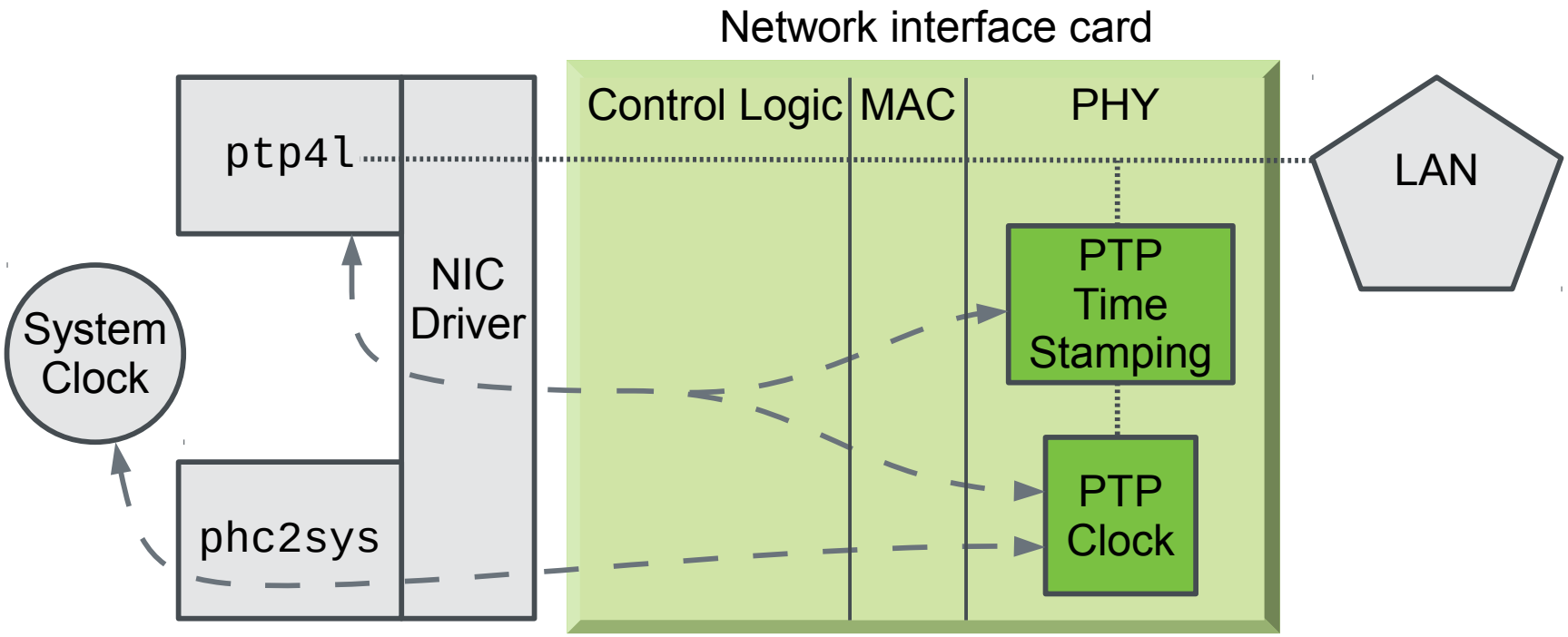
Master-Slave Design

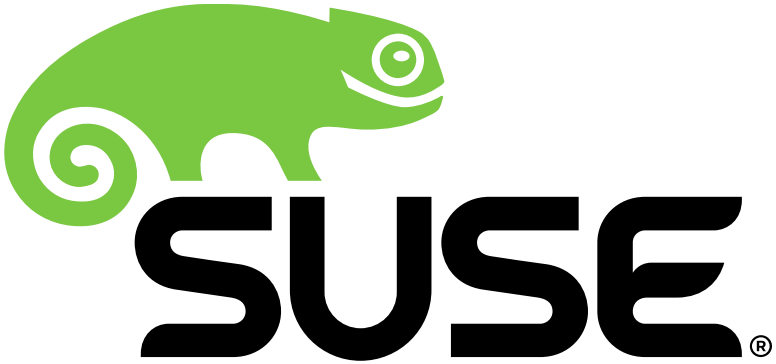
- Master sends out time information into network
  - Often in two steps - “time signal” and time information
- Slave nodes calculate offset and adjust their clocks



Hardware-assisted PTP in Linux

- PTP Hardware Clock (PHC)





Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org