

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.

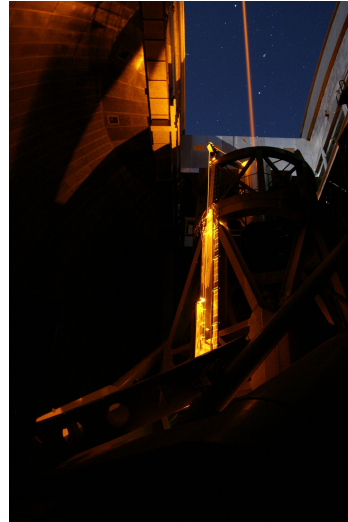


NASA JPL Telescope

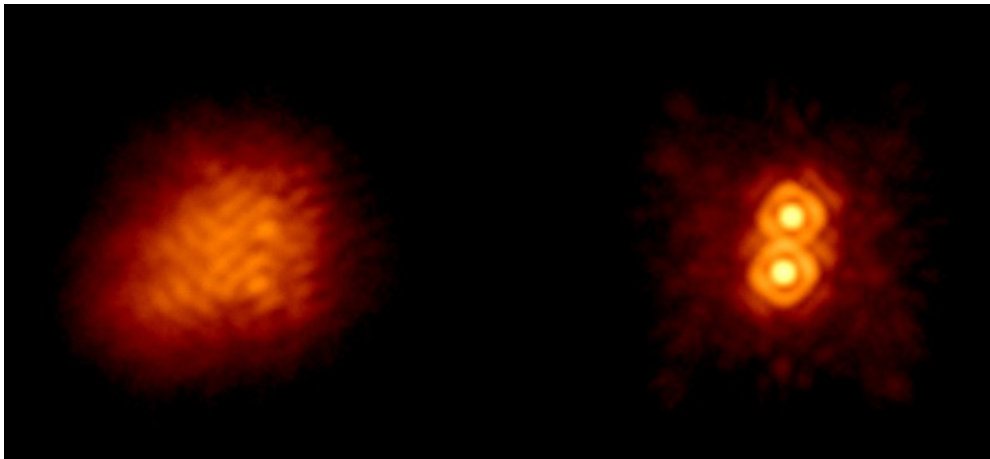


NASA JPL

SUSE Linux Enterprise Real Time Extension



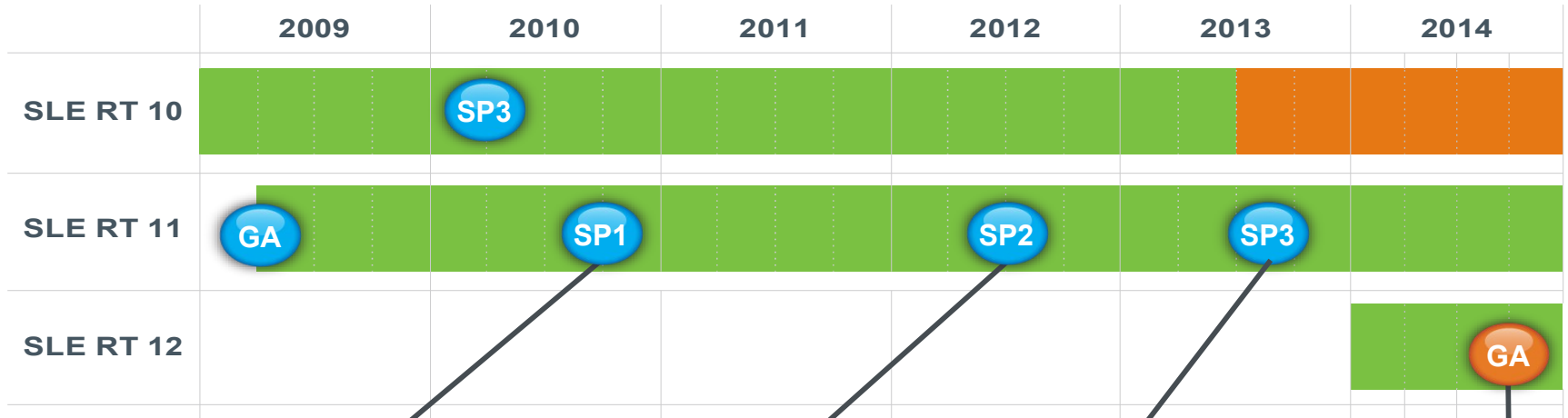
- 200-inch telescope with adaptive optics on Mount Palomar
- Avoid atmospheric blurring in Real Time
- Control more than 3000 mirror segments with a latency <math><250\text{ msec}</math>



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

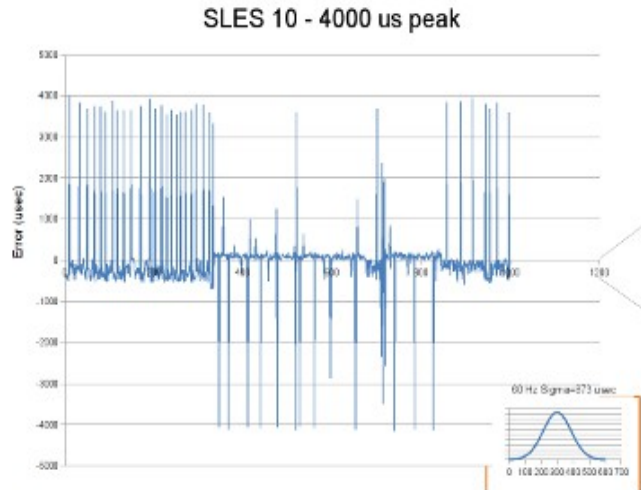
- LTTng update
- Precision Time Protocol

SLE RT 12

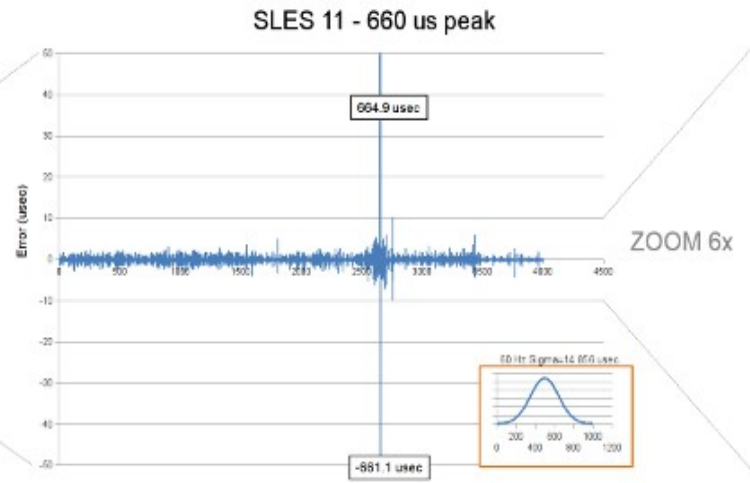
- Based on SLES 12
- GPU computing
- Match throughput of non-RT

Customer Timer Benchmark

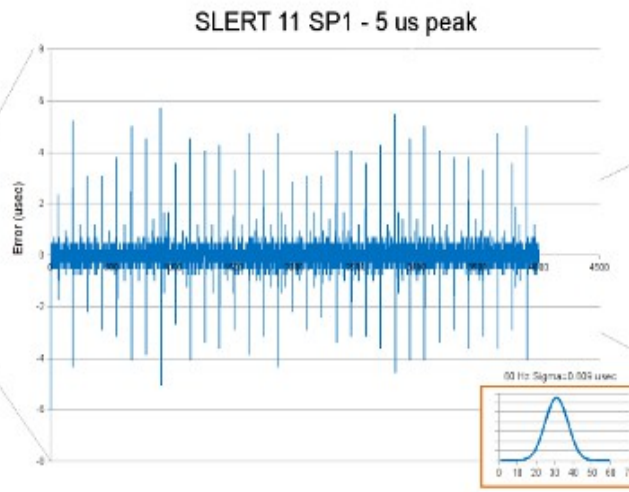
SUSE Linux Enterprise Real Time Extension



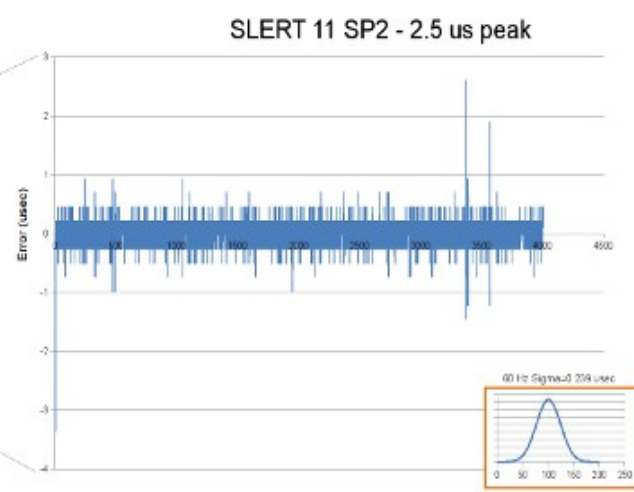
ZOOM 100x



ZOOM 6x



ZOOM 3x



Improvement: Factor 1,800!

Feature Details

Kernel Preemption

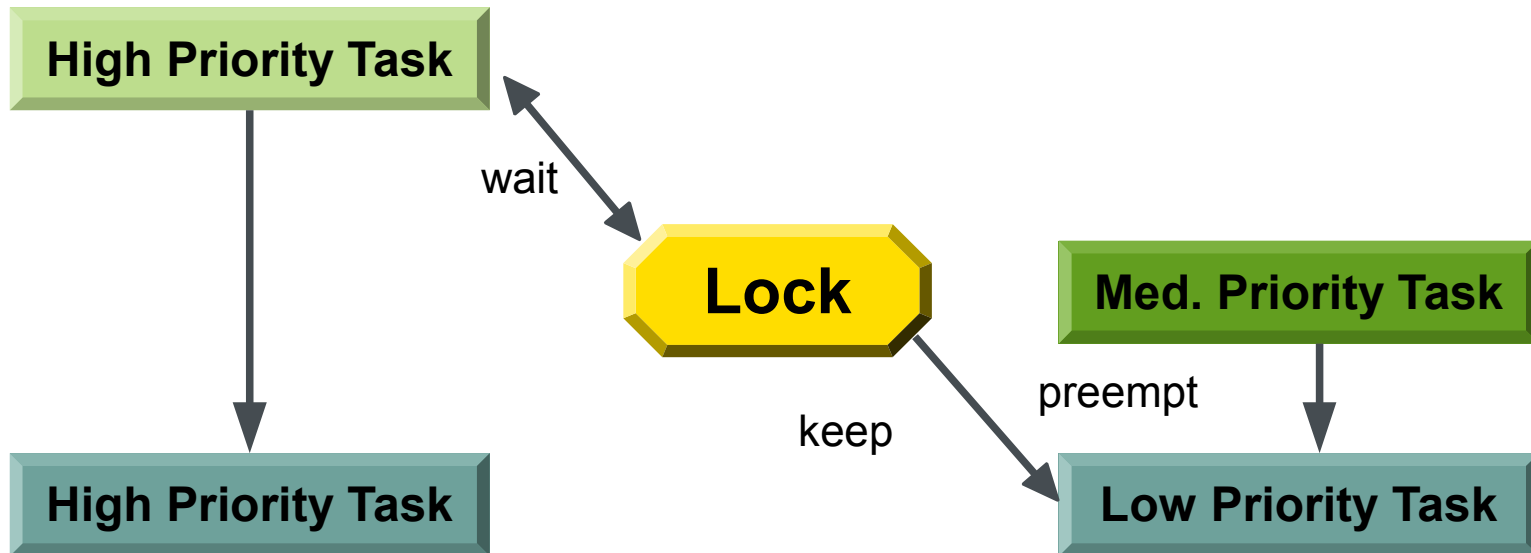
SUSE Linux Enterprise Real Time Extension

- Time critical tasks get immediate access to the CPU
- Real-Time Kernel improvements
 - Priority Inheritance Mutexes substitute non-preemptive 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



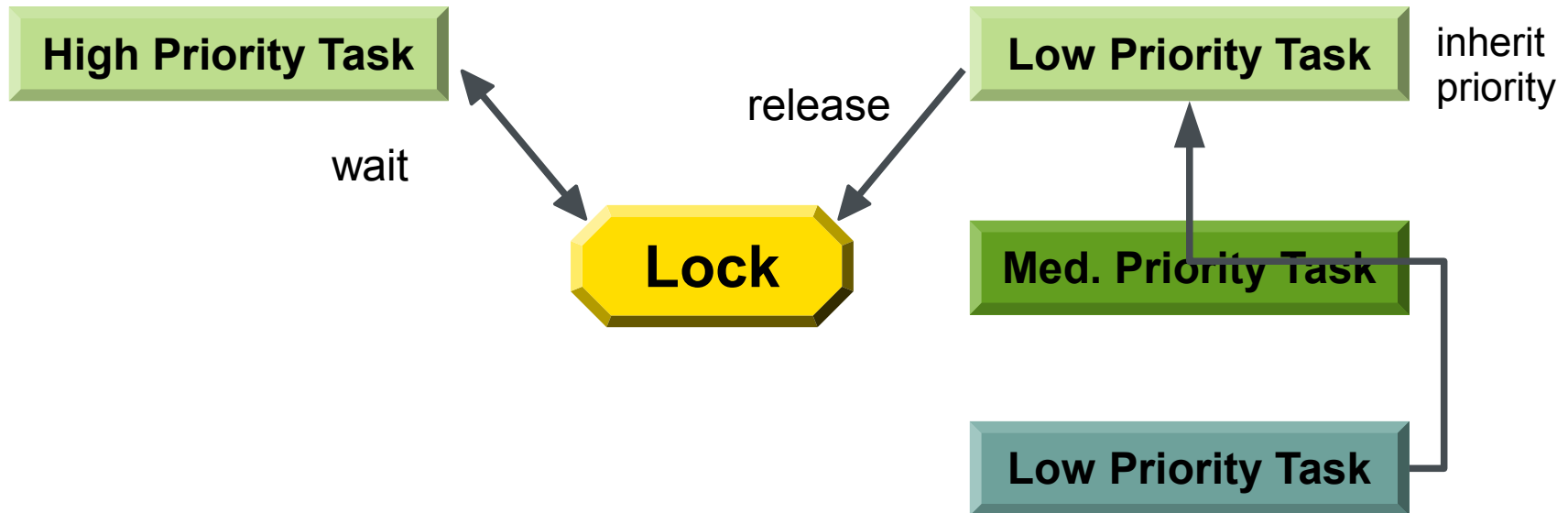
Effective priority goes down

Priority Inversion – Solution

SUSE Linux Enterprise Real Time Extension

- Priority Inheritance

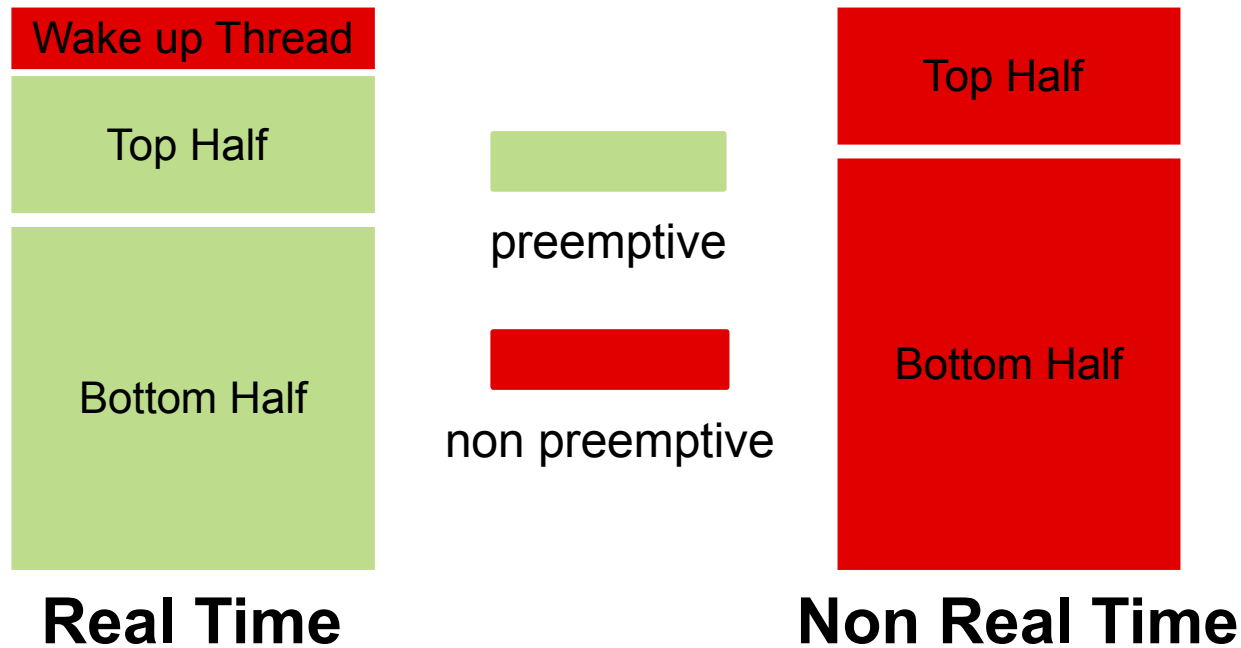
- Boost priority of low priority process until the lock is released



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

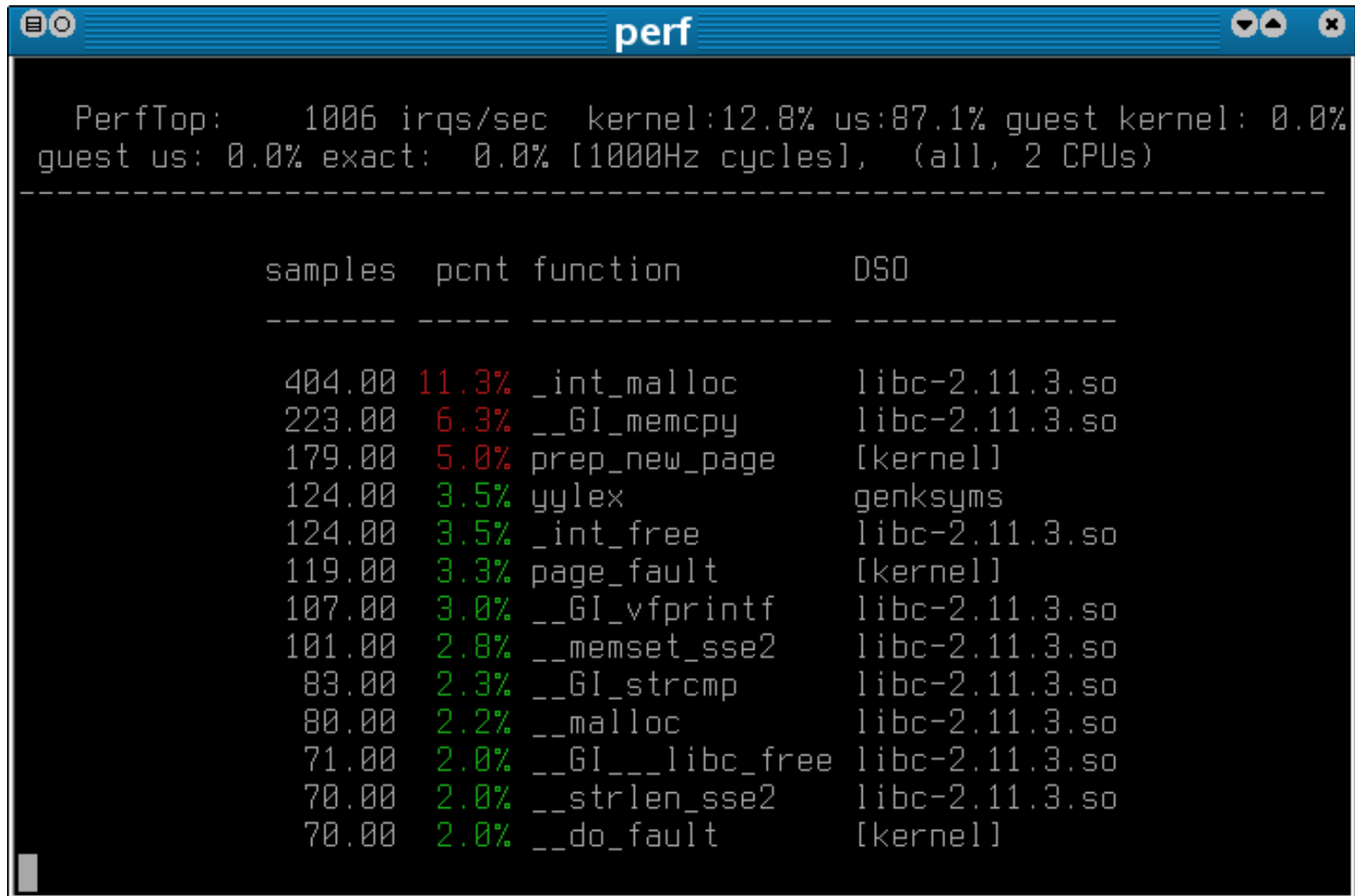
Profiling, Tracing

- 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



```
PerfTop: 1006 irqs/sec kernel:12.8% us:87.1% guest kernel: 0.0%
guest us: 0.0% exact: 0.0% [1000Hz cycles], (all, 2 CPUs)
-----
  samples  pcnt  function      DSO
-----
  404.00  11.3%  _int_malloc   libc-2.11.3.so
  223.00   6.3%  __GI_memcpy   libc-2.11.3.so
  179.00   5.0%  prep_new_page [kernel]
  124.00   3.5%  yylex         genksyms
  124.00   3.5%  _int_free     libc-2.11.3.so
  119.00   3.3%  page_fault    [kernel]
  107.00   3.0%  __GI_vfprintf libc-2.11.3.so
  101.00   2.8%  __memset_sse2 libc-2.11.3.so
   83.00   2.3%  __GI_strcmp   libc-2.11.3.so
   80.00   2.2%  __malloc      libc-2.11.3.so
   71.00   2.0%  __GI___libc_free libc-2.11.3.so
   70.00   2.0%  __strlen_sse2 libc-2.11.3.so
   70.00   2.0%  __do_fault    [kernel]
```

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 overridden 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 occurrences
- SUSE specific
 - CPU Flow – to visualize what each CPU has been doing

LTTng Eclipse Viewer

File Edit Navigate Search Project Run Window Help

Quick Access Java LTTng Kernel

Project Ex Control Flow Resources CPU Graph Statistics

Process	TID	PTID	Birth time	Trace
javaldx	2933	2932	11:16:22.995338807	kernel
javaldx	2934	2933	11:16:22.998814389	kernel
oosplash	2935	2922	11:16:23.003043313	kernel
oosplash	2936	2935	11:16:23.003787418	kernel
soffice.bin	2936	2935	11:16:23.003787418	kernel
soffice.bin	2937	2936	11:16:23.025629034	kernel
soffice.bin	2938	2936	11:16:23.322339382	kernel
configmgrWriter	2938	2936	11:16:23.322339382	kernel
soffice.bin	2939	2936	11:16:23.322785335	kernel
OficalPCThread	2939	2936	11:16:23.322785335	kernel
soffice.bin	2940	2936	11:16:23.392768046	kernel
soffice.bin	2941	2936	11:16:23.493777015	kernel
soffice.bin	2942	2936	11:16:23.618657069	kernel
sh	2942	2936	11:16:23.618657069	kernel
soffice.bin	2944	2936	11:16:23.788353685	kernel
soffice.bin	2945	2936	11:16:23.788484402	kernel
soffice.bin	2946	2936	11:16:23.788572955	kernel

Process States

- UNKNOWN
- WAIT
- USERMODE
- SYSCALL
- INTERRUPTED

Timestamp	Channel	Event Type	Content
<srch>	<srch>	<srch>	<srch>
11:16:23.051310469	test-channel1_2	sched_wakeup	tid=2936, target_cpu=0, comm=soffice.bin, prio=120, success=1
11:16:23.051311815	test-channel1_2	exit_syscall	ret=32
11:16:23.051312895	test-channel1_2	svs_settimer	value=140734130965696, which=0, ovalue=0

Control

Histogram Properties Bookmarks

Current Event (sec): 1382606183.051312352
Window Span (sec): 4.478854063

104558
0
1382606025.287325292
4177
0
1382606182.424501239
1382606186.901516346
1382606191.080750009

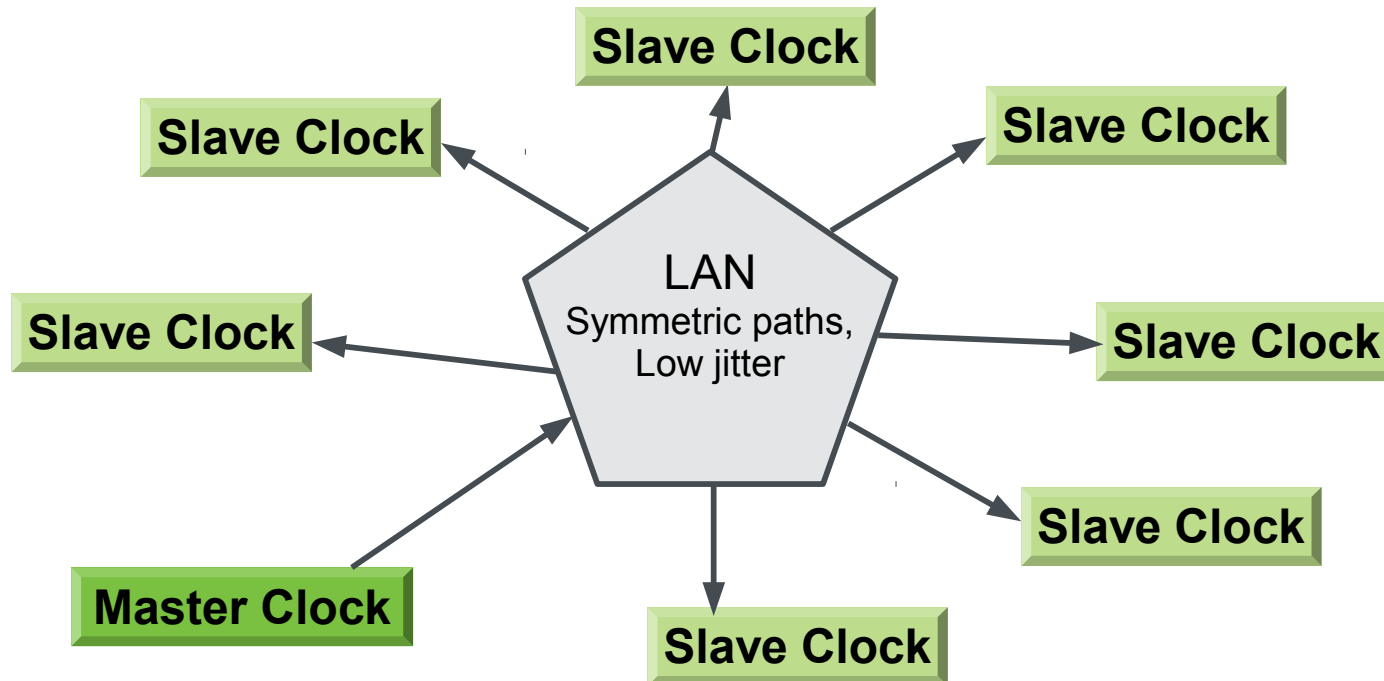
Precision Time Protocol

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

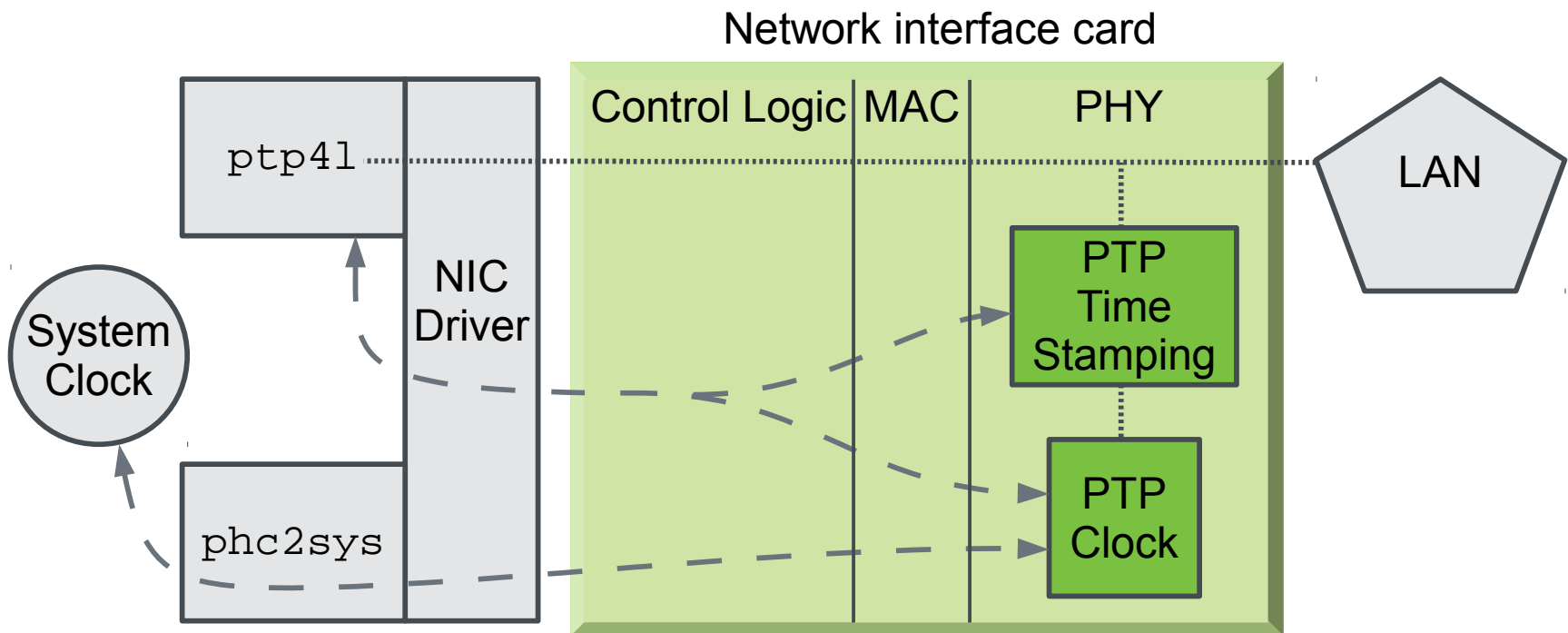
Master-Slave Design

- 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

Unpublished Work of SUSE. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

