Linux-rt: Turning a General Purpose OS into a Real-Time OS

Peter Zijlstra
(peterz@infradead.org)

Red Hat Inc.

What is Linux

- An open source Unix-like Kernel
- Started by Linus Torvalds in 1991
- MINIX
- GNU – RMS - 1983

Linux

- SMP (1-4096 CPUs)
- Preemptive
- 24 Major Architectures
- Countless Boards/Systems
- 1000's of drivers
- 1000000's LOC
- 1000's contributors
- 1 community

linux-rt

- Turn Linux, a GPOS into a RTOS
- Because:
  - People are adding GPOS features to RTOS'
  - Gives the programmer a single framework
  - It's fun

To Preempt or not to Preempt?

- !Preempt
  - Analyze all sections
  - Legacy code
  - Too much
- Preempt
  - Replace non-preemptible constructs with preemptible ones
  - Solve priority inversion

Preempt_RT

- IRQs
- Spinlocks
- RCU
- Per CPU data
- RW locks

- Threaded IRQs
- Mutexes
- Preemptible RCU (*)
- +Locks/Atomic
- Mutexes (*)

Threaded IRQs

- Kernel thread per ISR
- Generic hard-IRQ handler
  - Disables IRQ line
  - Wakes thread
- No generic code in IRQ context
- Memory allocators can be preemptible

Spinlocks/Mutexes

- raw_spinlock_t
  - For the few real sites
- spinlock -> mutex
  - spin_lock_irq*() doesn't alter IRQ state
- Implicit preempt-disable dependencies

Preemptible RCU

- Fun subject to talk about with Paul McKenney
- Implicit dependencies on preempt_disable
  - Lockdep annotation (?)

Per CPU data

- Add a lock
- Migration
- Atomics
- Trades performance for preemptibility

RW locks

- Non-deterministic
  - Waiting for unbounded # of readers
- Complex boost chain
- Map to mutex (*)
  - Sacrifice performance in favour or determinism

Priority Inversion

- Priority Inheritance
  - Needs simplification (?)
- RCU Boost (*)
- Work Queues (deferred work)

Semaphores vs. PI

- No resource owner
  - Convert to Mutex
  - Convert to Completions
- Eradicate semaphores (?)

Trouble

- More preemption
  - Bigger race windows
  - More likely to hit deadlock

Lockdep

- Runtime lock dependencies
- Lock classes
  - Lock initialization site
  - Requires annotations
- Validate DAG
  - Generates warnings before locking up

Lockdep

- Annotating classes
  - I-nodes
    - Class per filesystem
  - Recursion
    - mutex_lock_nested()
  - Trees
    - Balanced trees
      - Limited depth
      - Class for each level
    - Unbalanced
      - (?)
  - RCU (?)

Tracing

- Quickly find problems
  - IRQ-off latency
  - Preempt-off latency
  - OOPS-history
- Uses compiler prologue hooks (mcount)
- Records call trace history
- Catches races with predicates

Lockstat

- Lock usage statistics
  - contentions/acquisitions
  - wait-/hold-time
  - bounces
  - contention points
- Shows bottlenecks
  - files_lock

Really cool stuff

- Lockless (read-side) pagecache
  - RCU
- Concurrent (write-side) pagecache
  - Optimistic locking/RCU
  - Lock-coupling

Current developments

- Hot topics:
  - RT balancer
  - RCU Boost
  - Adaptive spin
  - RW locks
  - Group scheduling (bandwidth limiting)
  - Lockless get_user_pages()

RT balancer

- FIFO/RR
- SMP real-time invariant
- CPUSET root domain aware
- How to handle affinity (?)

RCU boost

- Prio boost all read-side sections on sync_rcu()
- Force grace period using krcupreemptd

Adaptive Spin

- Avoid context switch overhead
- Spin while owner is running

RW-locks

- Multi reader support
  - Reader limit
- Full PI
  - Boosts all the readers
  - Prio-fair
- Horribly complex code (?)

RT group scheduling

- cgroups
  - Task groups
  - Hierarchical
- FIFO/RR
- Bandwidth limits
  - Safe for !root users
- PI issues (?)

Lockless get_user_pages()

- Locklessly walk the page tables
- Avoids mmap_sem (rwlock)
- Improvements for:
  - DIO
  - futexes

Future Developments

- Things we hope will happen:
  - Partitioned EDF scheduler (?)
    - Deadline inheritance (?)
  - Soft-RT scheduler class (?)
  - RT network extensions (?)
  - ...

EDF

- Partitioned EDF scheduler
- Needs to extend the already complex PI framework
  - Deadline inheritance

Soft-RT

- Integrated or its own class?

RT network extensions

- RX memory reserves
  - Overlaps with swap over network effort
- Protocols (RTP?)

...

- We hope people will contribute their ideas
- And code
- Join the Linux(-rt) community
- Help obsolete the -rt patches

Academics vs Linux

- 'Cultural' differences
- Academic credit for work on Linux (?)
- Educate the kernel people
  - Various backgrounds
    - Physics, Math, HW Eng., MD
  - Mental context switches
  - Can't remember yesterday

How to do a kernel project

- Involve from the start
- Release early, release often
- Feedback on LKML
  - Act on it
  - Convince the other he's wrong
- Don't give up!

In theory, there is no difference
between theory and practice.
But, in practice, there is.