‘TICKLESS’ KERNEL
PRACTICAL EXPERIENCES

LinuxCon 2013
New Orleans

Fernando Garcia f@gentwo.org
Christoph Lameter cl@linux.com

Why is Linux ticking.

- Time keeping
- Scheduling
- OS maintenance
- Counter scaling
- Deferred free (lockless operations)
- Process statistics *getrusage()*
- Deferred processing

- Not tick related:
  - User space daemons
  - OS thread spawning
  - IPIs
  - Writeback / Filesystems
  - Compaction/NUMA migration.

Why it should not do that.

- Performance: Cache footprint increases.
- Cpu holdoffs.
- Applications experience seemingly random delays.
- Useless if only one app or none is running and the scheduler has nothing to do.
- Deterministic response (allows accurate Rendezvous for HPC application threads).
- Bare Metal performance (HPC and HFT requirements)

How to configure a tickless kernel

- Dedicate “Sacrificial” or OS processors
  - Minimum one per NUMA node.
- Tickless processors
  - Considering processor cache affinity and I/O affinities.
- Configuring RCU
- NOHZ configuration
- Dealing with write back threads
- Other OS measures

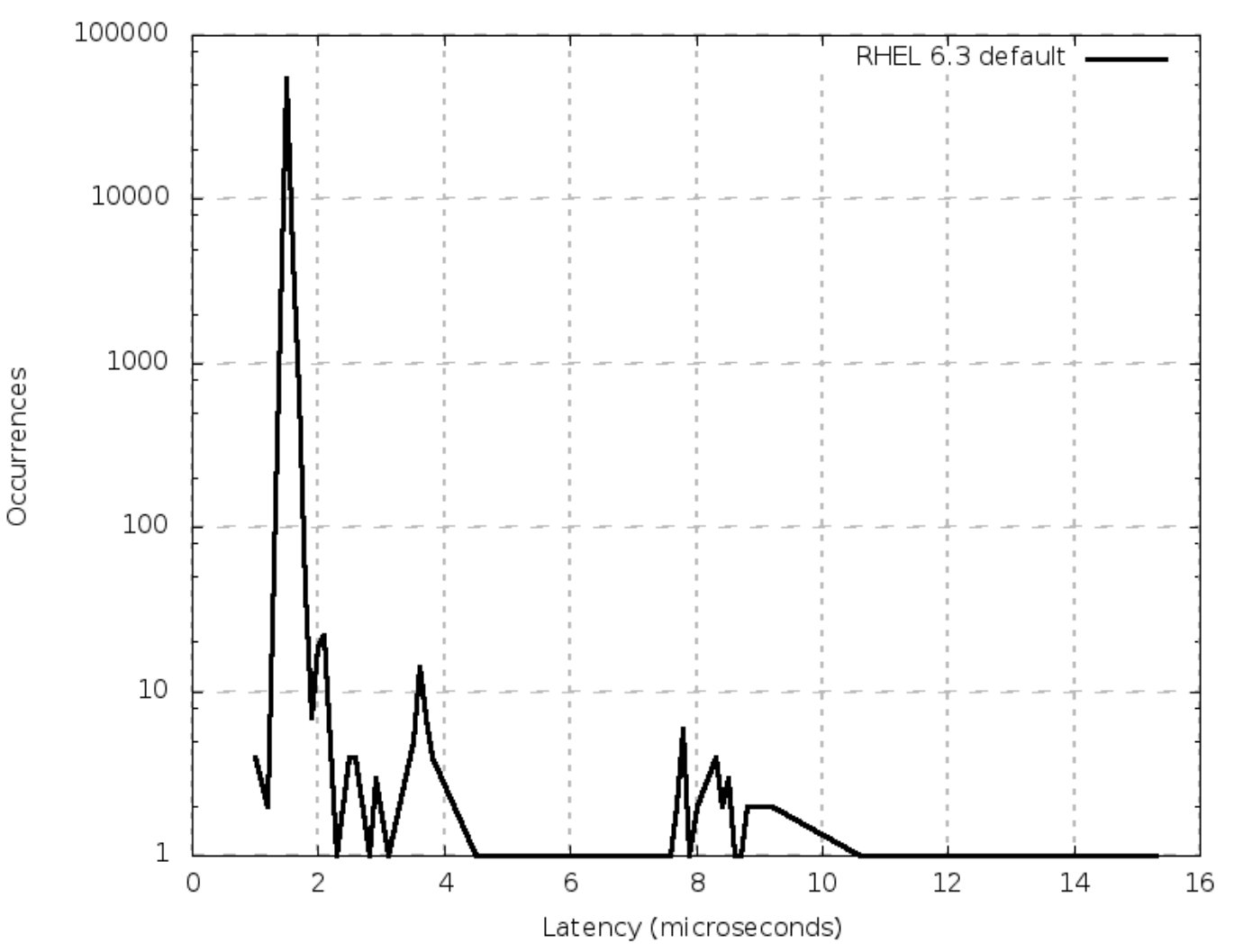
OS Noise in a Linux distribution

- Redhat distro as a reference point.
- Noise on an idle system
- Noise through file system operations
- Noise through operating system services on other processors
- Processor cache contention
- Hyperthreading: Processor units contention.

Tools

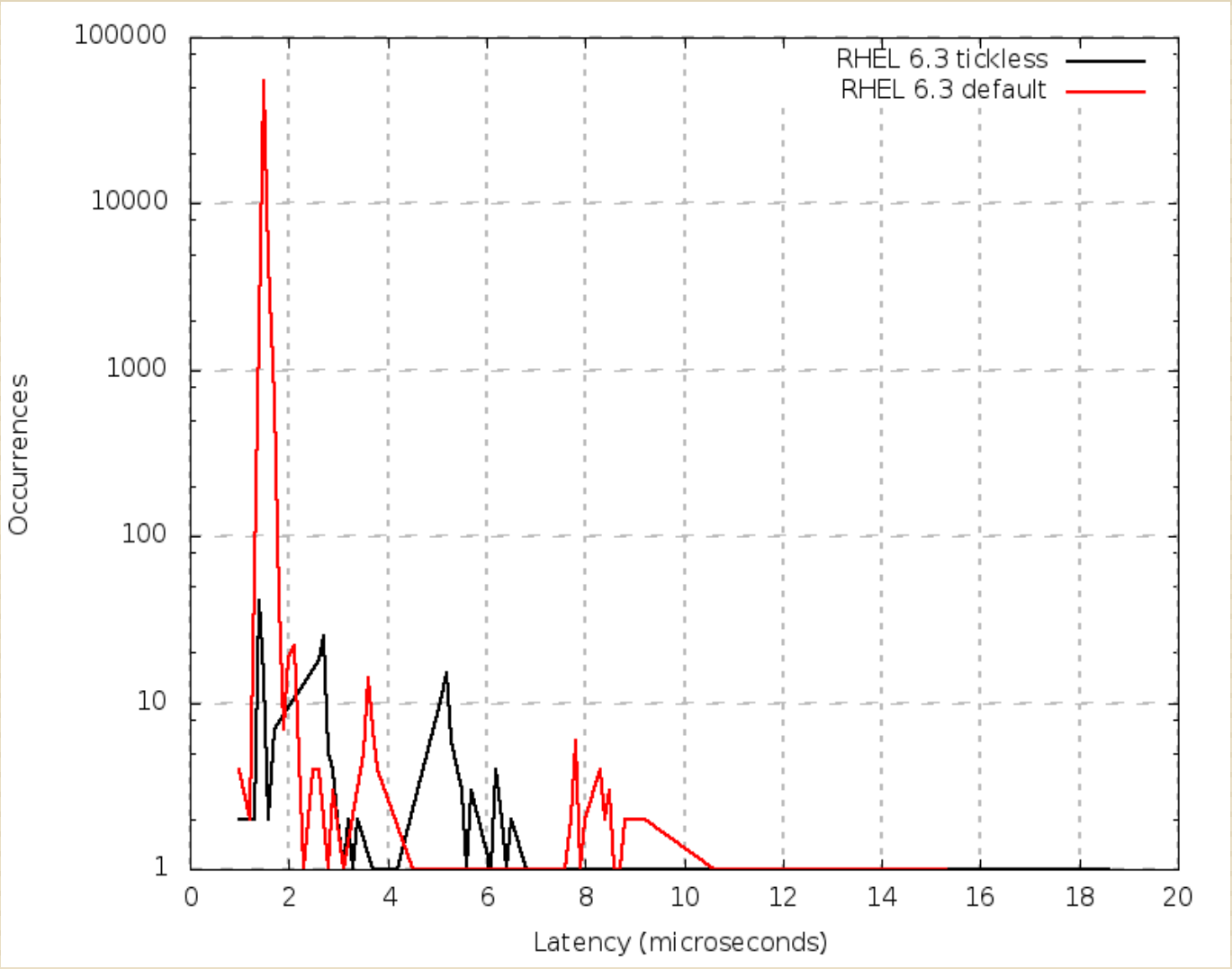
- Latencytest (or other tools)
  - Established base numbers for OS noise.
  - Allows debugging of setup. You should see significantly reduced numbers once the config is right.
- turbostat
  - Debug power state moves and frequency scaling
- perf
  - Investigate reasons for OS activities that cause holdoffs.

Holdoff on standard RHEL 6.3



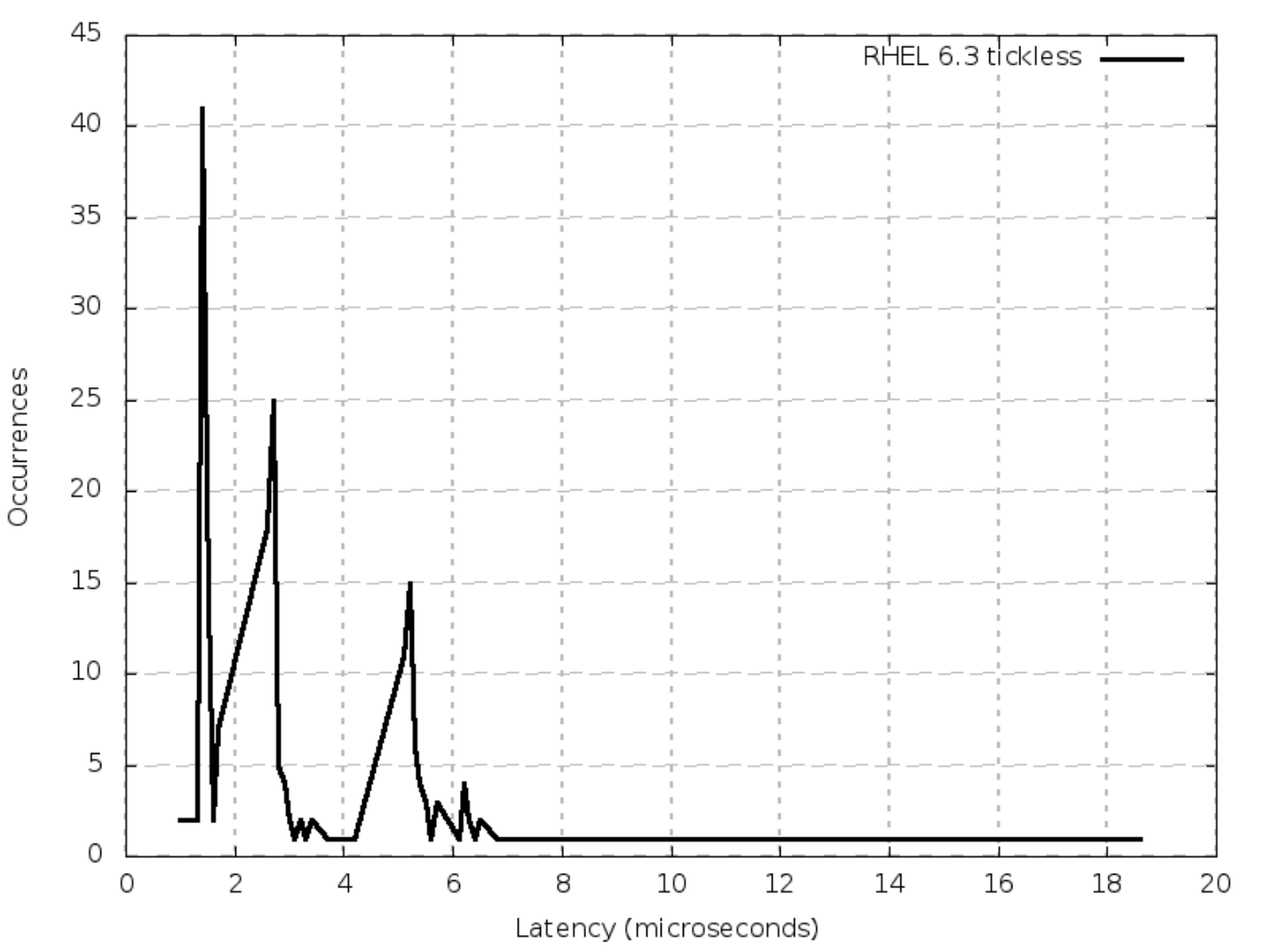
- Defaults to 1000 HZ so > 1000 holdoffs per second.
- High number of short events (likely tick not doing anything.
- Events > 5usec likely related to context switches.

RHEL 6.3 vs Linux 3.11 tickless



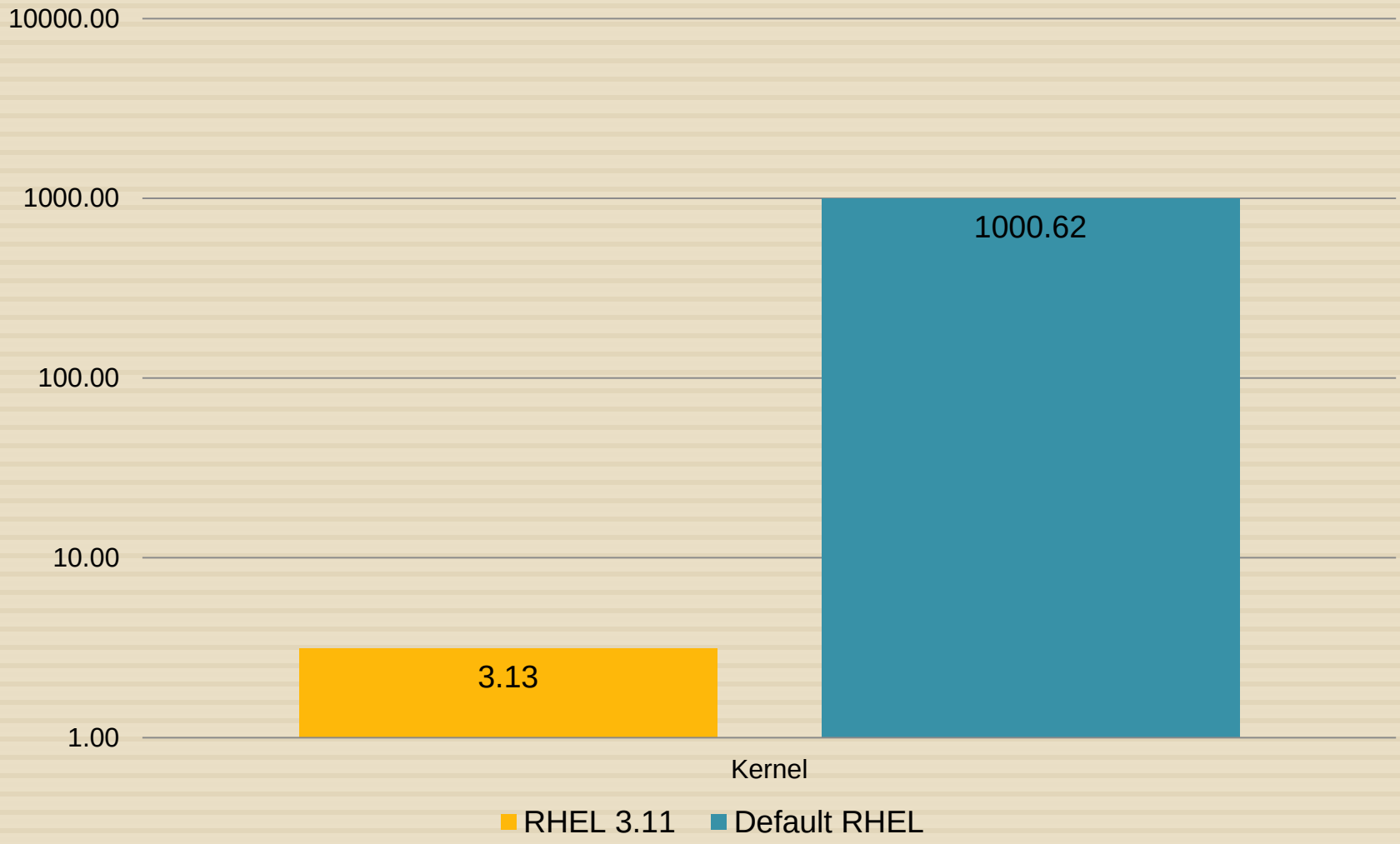
- Reduction of events by 4 orders of magnitude
- Kernel 3.11 has lower latency in general
- Frequency of high latency events is also reduced.
- There are still numerous enhancements pending.

Tickless numbers



- Nicer view of the tickless run

Total Variances per second



Other issues

- Tests on an otherwise idle system. There will be more if OS services are in use.
- OS Spawning of threads
  - Kthreadd
  - Worker threads
  - I/O related threads
  - Usermodehelper (kmod, devices)
- IPIs
  - Scheduling
  - Flushing

Easy configuration

- Bootup
  - System auto configures sacrifical processor
  - All processing moved off other processors
- Active system
  - Only intervene on processors if required.
  - Logging of the reasons why

Work to be done

- Only run vmthreads if necessary
  - Patch by Christoph exists
- Autoconfig of system
  - Setup is weird right now.
- Move other stuff to OS cpus.
  - Requires scripting right now.
- Stop kernel from spawning on tickless cpus.
  - Patch by Christoph exists

Grand Future Vision

- Assymmetric SMP
- Cores dedicated to tasks
- Configuration of cores for particular tasks.

Conclusion

- Questions?
- Answers?
- Opinons?