Chronicle SOFTWARE

Chronicle Tune

DOCUMENTATION

chronicle.software

Chronicle Tune

To maximise application performance on Linux, in particular achieving low and predictable latency with minimal jitter, it is crucial to fine-tune applications to their environment. Leveraging decades of experience in OS tuning, Chronicle has distilled these insights into a product which can help extract best performance for diverse application stacks.

Chronicle Tune is an integrated package to set up and manage the low-level tuning of Linux hosts. Originating as an internal tool aimed at optimising our own low-latency, low-jitter environments, Chronicle Tune is designed to be simple to install and configure, while providing powerful CPU isolation, IRQ control, and power management, together with a complementary range of tuned kernel and system parameters. Importantly the kernel tuning is maintained in lock-step with changes to the CPU isolation configuration.

Simple to install and configure, Chronicle Tune integrates the following:

- CPU isolation, for low-latency, low-jitter tasks
- Transparent control of CPU power management
- Per-thread scheduling using standard calls (eg taskset, sched_setaffinity)
- IRQ management
- Whole-system optimisation, including SSD, disk, memory, and network
- Changes can be applied dynamically, eg according to a schedule or in response to changing load

Correctly tuned systems can not only yield better performance, but also improve efficiencies by more effectively utilising available resources, and matching power demands to load. Depending on the workload and scale, Chronicle Tune can help extract better performance, and by using available resources more efficiently, has the potential to also reduce hardware and running costs.

Key Features

Flexible CPU Management

Chronicle Tune partitions a host's CPU set into two subsets: a general set for standard processes and a reserved set which is shielded from general workflows for exclusive use by latency-sensitive processes and threads. This approach ensures that critical applications are allocated required resources without interference from other processes.

Process placement and control with Chronicle Tune uses standard taskset and sched_setaffinity commands, allowing for precise, granular control over process and thread placement. Of particular note, Chronicle Tune respects range-based CPU placement, avoiding limitations and common pitfalls with other tools such as isolcpus.

Real-Time Configuration

One of the standout features of Chronicle Tune (vs isolcpus especially) is its ability to change configurations without requiring a system reboot. This capability crucially enables highly available production environments to be reconfigured on-the-fly in response to varying loads, market conditions and similar. Reconfiguration is performed as root, ensuring appropriate security and control levels.

Comprehensive Kernel Tuning

Chronicle Tune goes beyond basic CPU isolation by incorporating extended kernel tuning parameters that align with the reserved CPU set. This holistic approach ensures that the entire system is optimised for low-latency and low-jitter performance, addressing potential bottlenecks at various levels of the system architecture, including SSD, disk, memory, and network.

Integrated IRQ Management

Efficient management of Interrupt Requests (IRQs) is essential for maintaining system performance, especially in low-latency environments. Chronicle Tune provides robust and clear IRQ management, helping optimise system performance with low jitter and reduced interrupt-to-application latency.

Power Management

Chronicle Tune enables simple and transparent dynamic management of CPU power controls, again helping maximise performance when needed, and reducing power consumption and associated costs out of hours..

Configuration

Chronicle Tune simplifies the configuration process with a single INI file. This file uses key-value pairs grouped by sections, making it easy to read and modify. The tool's command-line interface further streamlines configuration, allowing administrators to implement changes quickly and efficiently.

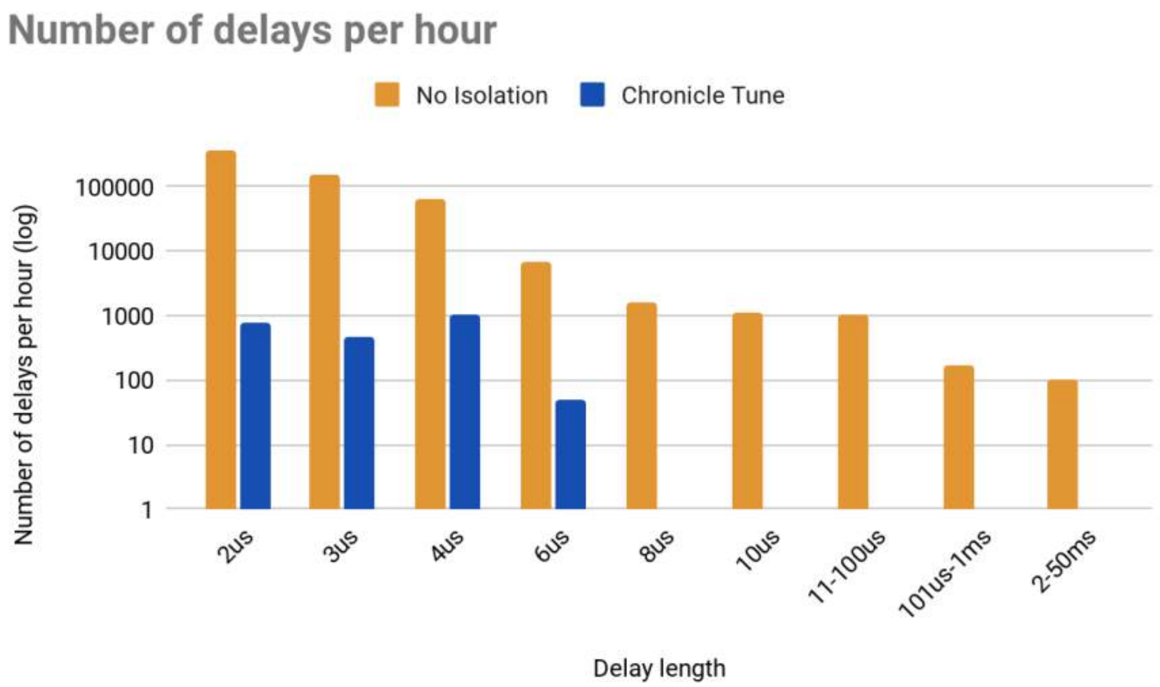
Performance

Jitter

Jitter profiling is a useful metric for gauging the effectiveness of system tuning, with the desired result being fewer and smaller jitter events, translating to fewer interruptions to business critical workloads.

The following plot shows jitter histograms for a typical Enterprise server, both before tuning (orange plots) and after Chronicle Tune is applied (blue plots). The aim is to be tucked into the lower left corner of the plot corresponding to fewer jitter events (y axis), and smaller interruptions (x axis).

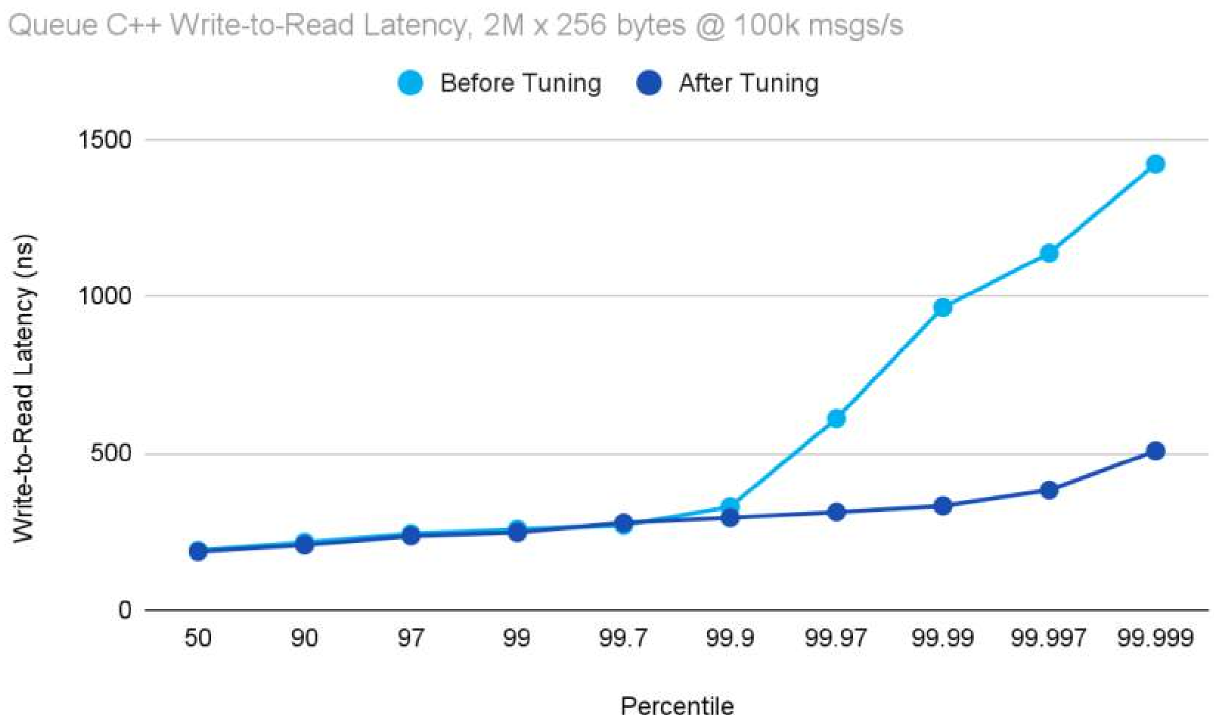
Chronicle Tune effectively eliminates any delays longer than 10us, and reduces the number of interruptions by orders of magnitude. Note that both axes are logarithmic.



Average number of jitter events per hour, grouped by length of the delay. Note that the y-axis is logarithmic, with the right hand half of the x-axis also logarithmic, spanning several orders of magnitude. Tests performed on a 2x12 core Xeon E5-2650.

Chronicle Queue Write-to-Read Latency

The next test shows how reduced jitter from tuning can translate directly to improved application performance. In this case the write-to-read latency of C++ Chronicle Queue exchanging 256-byte messages at a rate of 100k msgs/s is measured. The aim is for a low and flat plot, corresponding to consistent low latencies with minimal jitter.



Write-to-Read latency of C++ Queue exchanging 256 byte messages at 100k msg/s (lower is better)

The tests show Chronicle Tune substantially eliminates jitter, resulting in much reduced outliers even out as far as the 99.999 percentile.

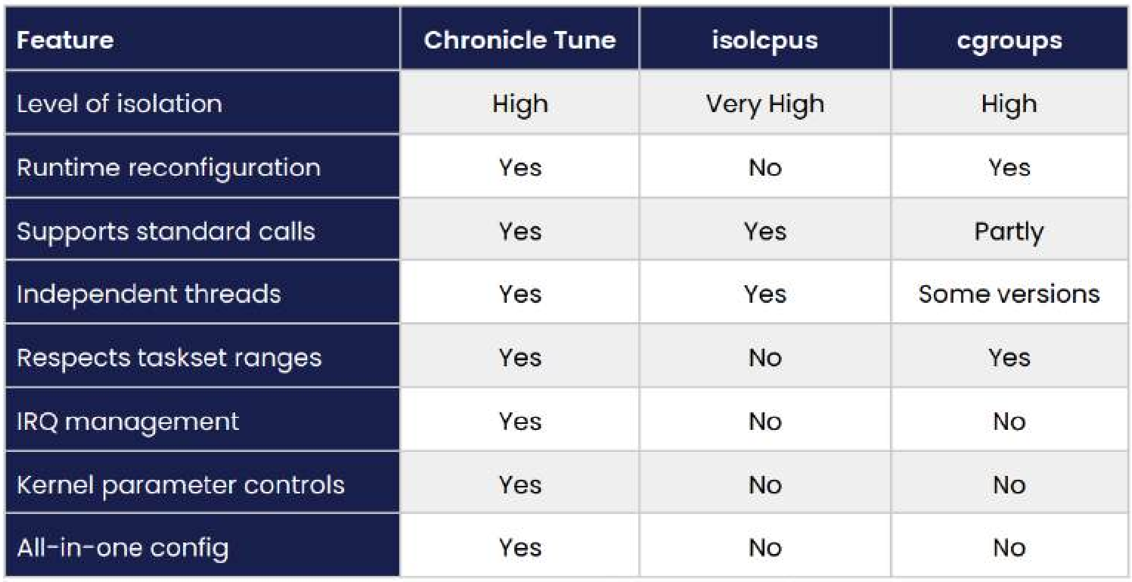
Comparison with isolcpus and cgroups

While Chronicle Tune covers a lot more than just CPU isolation and process/thread management, it is natural to compare this specific aspect alongside two traditional CPU isolation methods in Linux: isolcpus and cgroups/csets.

Chronicle Tune offers important benefits over both isolcpus and cgroups, particularly in its flexibility and integrated approach. While isolcpus achieves very high isolation by removing specific cores from the scheduler's consideration, it requires a system reboot to change configurations, making it impractical for dynamic environments, eg. load-balanced clusters.

Furthermore, isolcpus necessitates meticulous thread management, especially in multi-threaded applications like Java, where auxiliary threads can disrupt performance if not carefully isolated. Chronicle Tune, on the other hand, allows for real-time reconfiguration without the need for rebooting, enabling system settings to be adjusted on-the-fly to maintain optimal performance continuously.

cgroups provides another alternative by enabling the creation of CPU sets for process isolation, with the added benefit of runtime reconfiguration. However, cgroups can be cumbersome in its execution, requiring specialised commands to move processes between groups. Additionally, support for per-thread control is limited in some cgroup versions, which can hinder the fine-grained tuning necessary for low-latency applications, especially for Java given the large number of auxiliary threads spawned by the JVM. Chronicle Tune addresses these limitations by supporting standard system calls like taskset and sched_setaffinity, allowing for precise control of individual threads within processes. This feature is particularly beneficial for applications requiring specific thread placements, ensuring that critical tasks receive the necessary resources without interference.



Feature	Chronicle Tune	isolcpus	cgroups
Level of isolation	High	Very High	High
Runtime reconfiguration	Yes	No	Yes
Supports standard calls	Yes	Yes	Partly
Independent threads	Yes	Yes	Some versions
Respects taskset ranges	Yes	No	Yes
IRQ management	Yes	No	No
Kernel parameter controls	Yes	No	No
All-in-one config	Yes	No	No

Table 1. A complete comparison of the various features of Chronicle Tune, `isolcpus`*, and* `cgroups`*.*

Conclusion

By integrating complementary kernel tuning together with power and IRQ management, Chronicle Tune goes beyond simple CPU isolation and thread placement, providing a holistic approach to system optimization.

Such integration ensures that all aspects of system performance are aligned with the requirements of latency-sensitive applications, something that neither isolcpus nor cgroups fully achieve. By combining the strengths of both methods and adding its unique capabilities, Chronicle Tune offers a superior solution for managing low-level system tuning in Linux environments.

For more information about Chronicle Tune, please visit https://chronicle.software/tune, or contact sales@chronicle.software

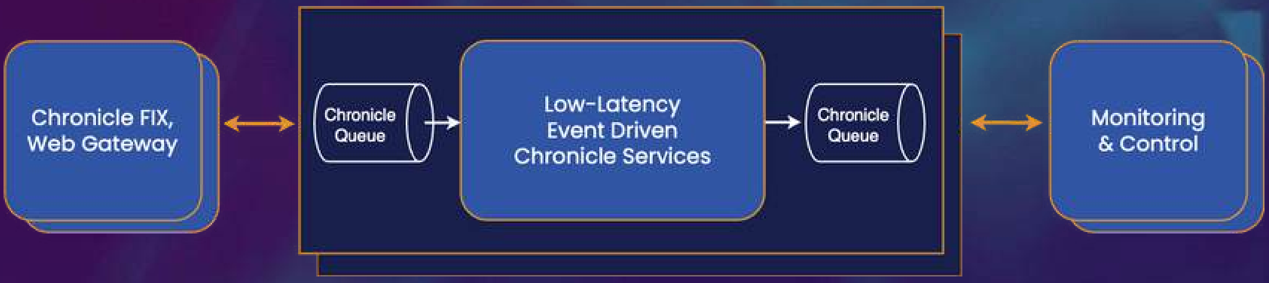
Chronicle SOFTWARE

Enabling Technology for Trading Systems

Billions of dollars a day are handled via Chronicle's technology platform. We are trusted to deliver exceptional performance with minimal time to market, whilst optimising developer efficiency.

Build it Fast, Quickly and Safely.

SOLUTIONS


Chronicle FIX, Web Gateway
Chronicle Queue
Low-Latency Event Driven Chronicle Services
Chronicle Queue
Monitoring & Control

Chronicle FIX

Super fast, super reliable

Chronicle Services

Event driven microservices architecture

Chronicle Queue

Low latency messaging framework

Chronicle Tune

Improve latency and reduce jitter