PTC Perc®

WHEN MILLISECONDS MATTER

The PTC Perc family of virtual machine solutions and development tools for single-processor or multi-core systems are proven, competent performers in complex mission-critical, real-time applications.

Java virtual machines (VMs) are common-place technology today, especially in the ever-expanding worlds of the Internet of Things (IoT) and Industrial Internet of Things (IIoT). But generally, traditional Java VMs are too unpredictable and unreliable to be of use for sophisticated embedded applications. PTC Perc combines predictable and reliable performance with Java Standard Edition compatibility.

In comparison, the benefits of PTC Perc are both clear and compelling: a more productive development cycle, and more efficient and reliable programs. These advantages result in faster time to market, lower costs, and higher customer satisfaction. This is why PTC Perc is the most field-tested VM for real-time Java developers.

More productive development cycle

PTC Perc provides compatibility with off-the-shelf Standard Edition class libraries - rather than limited “personal,” “micro,” or “compact” subsets, giving developers better library support for more complex applications. Powerful desktop-caliber development tools including symbolic debuggers and run-time performance monitors and profilers provide a richer development experience. Management APIs and command shells are included to supply visibility and control over internal operation of the VM, and a broad wealth of native compiler and operating systems ports for key real-time operating systems are available.

More efficient and reliable programs

With an industry leading, patented real-time garbage collector featuring ultra-reliable performance, PTC Perc delivers unprecedented real-time predictability down to the sub-millisecond range. Real-time operation of threads, synchronization, and timers give developers the ultimate platform for managing application execution with deterministic and reliable real-time behavior.

Thanks to its rich set of pre-optimized libraries, PTC Perc allows engineers to focus on optimizing application-specific modules.

These features combine to make PTC Perc the ideal VM environment and toolset for software targeted to complex projects such as intelligent gateways, distributed real-time control, network infrastructure management plane, and industrial automation.

PTC Perc support for modern processors

PTC Perc is available in single processor as well as SMP variants that support multi-core systems. PTC Perc is also available for 32-bit and 64-bit microprocessor families.

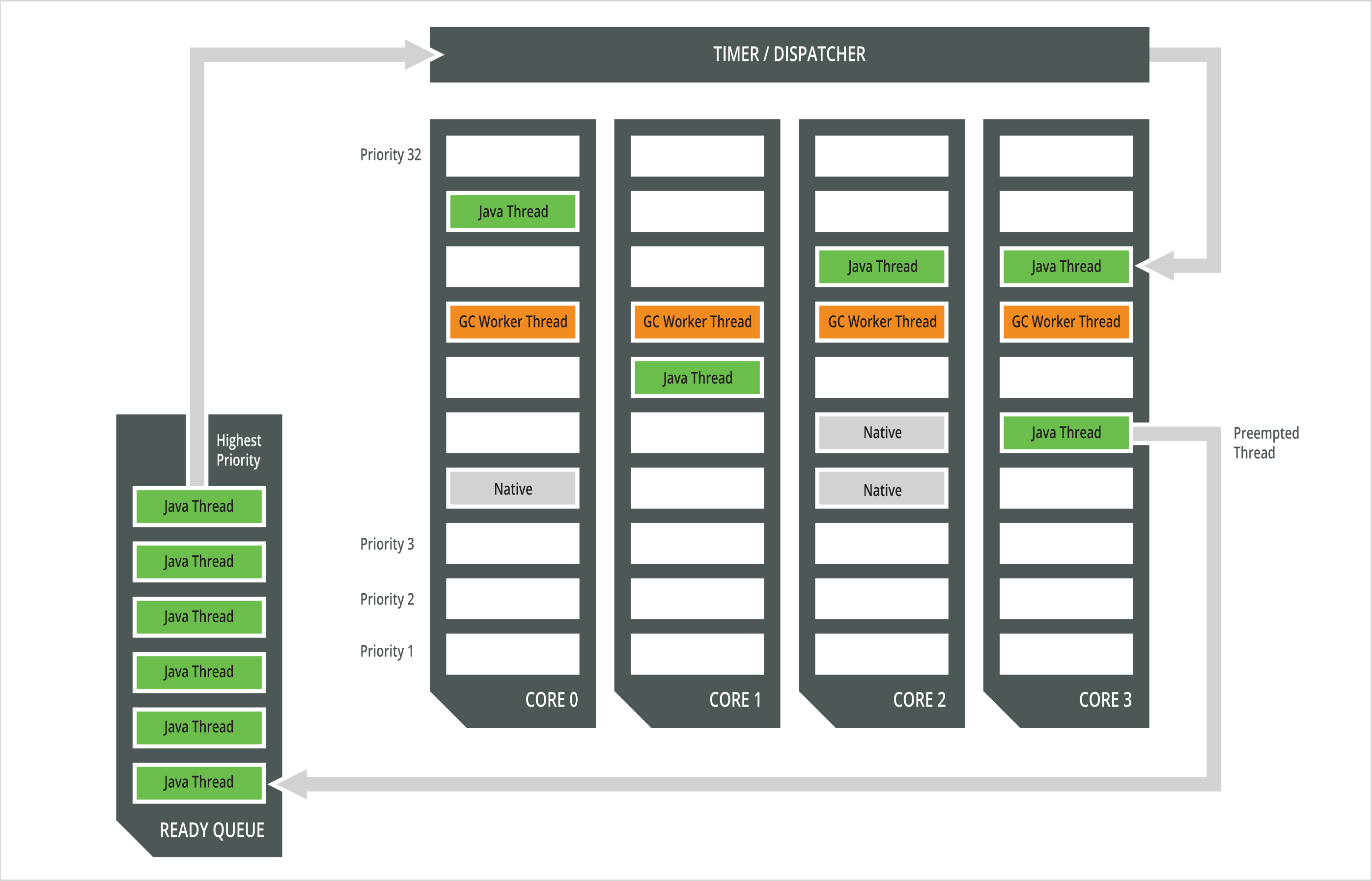
Java, with its built-in support for concurrency, is a development language and execution environment well suited to use with multi-core processors. However, most garbage collection technologies are not structured to take advantage of multiple processors and not capable of running concurrently across multicore systems. To get the most effective use of the horsepower provided by these new processors, a more intelligent GC technology is required.

PTC Perc has been addressing some of the most challenging applications imaginable, combining predictable and reliable performance with Java Standard Edition compatibility. As a result, it has become the most deployed VM in the real-time space.

Today's applications continue to be more complex and generally larger, placing added computing burden on the hardware platforms in which they execute. The industry has responded with multi-processor or multicore systems. PTC Perc meets the needs of these new architectures in addition to providing the breadth of features, functions, and performance embedded and real-time developers require.

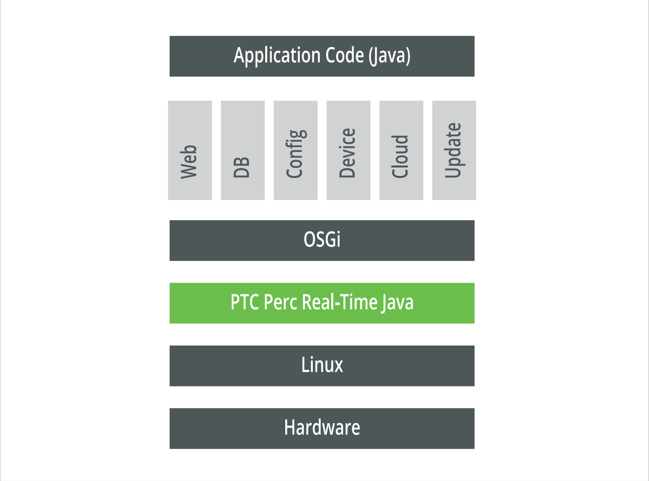
The concurrent GC in PTC Perc performs the collection of unused objects by multiple processors, while Java application threads continue to operate concurrently. This enhances the ability of the GC to pace the garbage collection rate to the application's memory allocation rate.

SMP Scheduling with PTC Perc



PTC Perc's SMP-specific features:

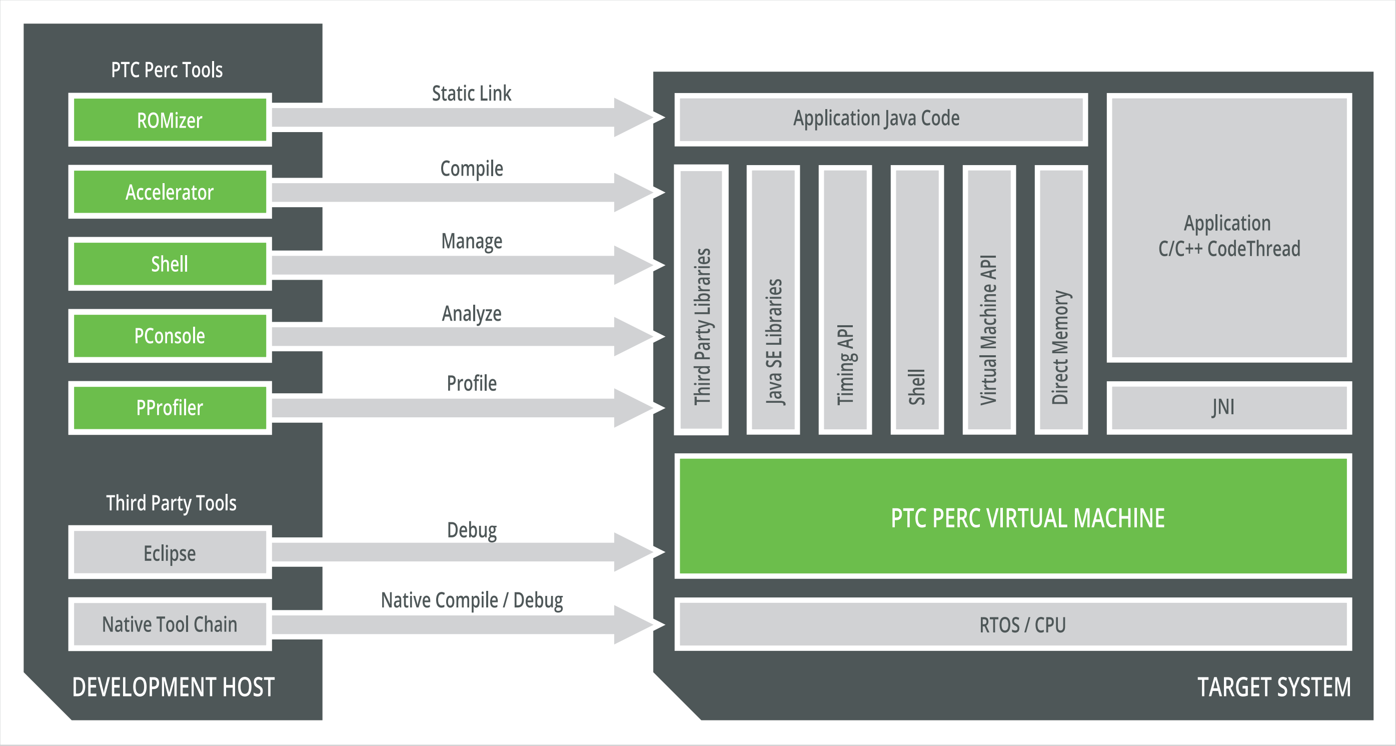
- Runs multiple Java threads concurrently across all available CPUs/Cores
- Supports setting the affinity of CPUs/Cores available to PTC Perc SMP for mixed Java/Native applications
- New, multi-threaded Garbage Collector (GC) runs faster and takes advantage of available CPUs/Cores
- GC is incremental, pre-emptible by higher priority Java threads
- Impressive deterministic real-time behavior with typical response latency of <1ms on x86 1.6GHz+
- Support for Linux®/x86, Linux/PPC, and Linux/ARM SMP kernels and popular multi-core capable Real-Time Operating Systems (RTOSs)



PTC Perc IoT Gateway Stack

IoT/IIoT/M2M intelligent gateways and PTC Perc

With the advent of IoT, a new class of hardware has come upon the scene to supply connectivity of end devices and sensors to the cloud. Many of these devices are being supplied with software stacks including not only networking software, but also Java VMs and IoT platform services and libraries. VDC Research, a leading industry analysts addressing IoT and the embedded market indicates that "Even newer device classes such as gateways require real-time functionality for some of the more "automation/control"-oriented use cases." For those applications needing real-time performance and IoT connectivity, PTC Perc is an ideal solution.



PTC Perc Toolchain

Key features

Standard class libraries

- Enables the most compelling features of Java (including JNI, RMI, JDBC, Collections, Concurrency, XML, etc.,) as part of the standard edition

Embedded Graphic Support

- AWT/Swing on selected platforms
- Direct memory API provides compiler-optimized access to buffers and memory-mapped I/O

VM management API

- Allows engineers to fine-tune on the fly
- Programmatic access to PTC Perc VM threads, monitors, memory, GC, files, and networking

Accelerator

- AOT and JIT deliver superior performance by turbocharging applications up to 20x over interpreted implementations
- Supports dynamic loading of native-compiled Java classes

Shell

- Cuts development time by providing direct access to the running VM via serial or telnet connection
- Optimizes system performance and functionality by allowing remote viewing information and direct control of the VM running classes

Remote debugging

- Saves time, money, and programmer effort
- Allows programmers to seamlessly debug on the target device

PConsole and PProfiler tools

- Graphically displays memory usage, thread activity, and CPU utilization of Java applications
- Speeds time-to-market and optimizes system performance by helping to quickly detect and fix memory leaks and bottlenecks

ROMizer

- Improves execution speed by statically linking VM, libraries, and application object code into a single executable image

Ancillary Tool support

- Leverages RTOS vendor and native development tool chains

For more information, visit:
ptc.com/developer-tools/perc

J07570–PTC Perc–EN–0816