info@concurrent-rt.com
concurrent-rt.com
800.666.4544

NightStar

Real-Time Linux® Debugging and Analysis Tools

Concurrent's NightStar™ is a powerful, integrated tool set for debugging and analyzing time-critical Linux applications. NightStar tools run with minimal intrusion,thus preserving application execution behavior and determinism. Users can quickly and easily debug, monitor, schedule, analyze and tune applications in real-time.
NightStar GUI-based tools reduce test time, increase productivity and lower development costs. Time-critical applications require debugging tools that can handle the complexities of multiple processors and cores, multi-task interaction and multi-threading. NightStar's advanced features enable system builders to solve difficult problems quickly. NightStar tools are available for a range of popular Linux distributions.

- Deterministic multi-core debugging, monitoring, tracing and scheduling
- Ideal for time-critical CPU and GPU applications
- Easy-to-use graphical user interface
- Support for any mix of C/C++, Ada and Fortran® tasks
- Self-hosted or remote target system operation
- Integrated context-sensitive help

NightView™ Source-level Debugger

- Multi-system, multi-process and multi-thread debugging via a single interface
- Hot patches including breakpoints, patchpoints,monitorpoints and watchpoints
- Application speed conditions and ignore counts
- Dynamic memory (heap) debugging
- Modification and display of variables during execution

NightTrace™ Event Analyzer

- Synchronized graphical or text display of all system activity
- OS event logging including system calls, interrupts and exceptions
- Precision time-stamping of all events
- Powerful data analysis API
- Function call and parameter tracing
- User-defined event logging

NightSim™ Application Scheduler

- Cyclic execution of multiple processes
- Major and minor cycles with frame overrun notification and control
- Scheduling via real-time clocks or external interrupt sources
- Ideal for simulation applications

NightProbe™ Data Monitor

- Unintrusive sampling and recording of program data
- Synchronous and asynchronous data capture
- Flexible data display features
- Program data modification
- Sampling, recording and replay APIs

NightTune™ Performance Tuner

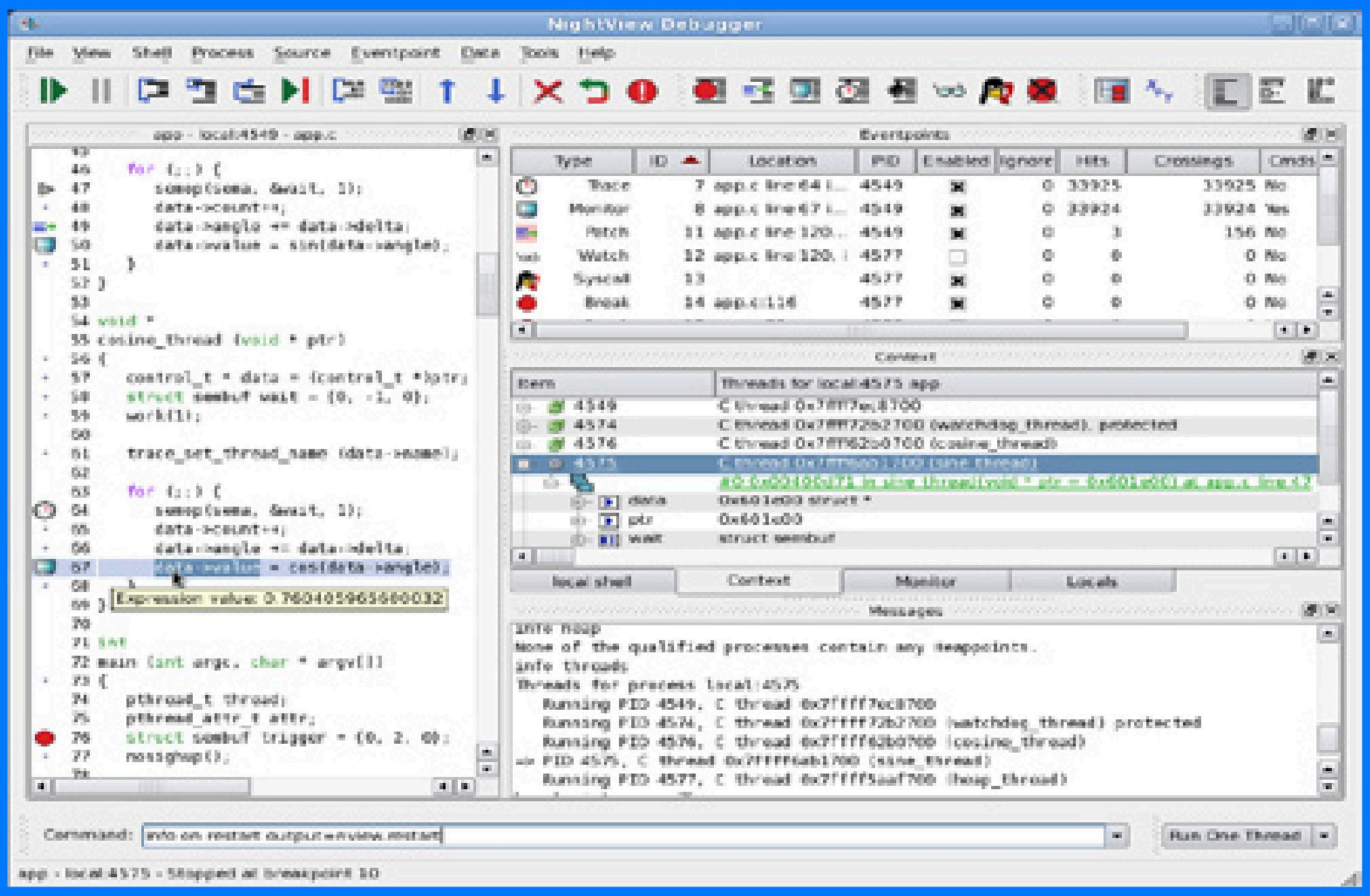
- Monitoring of CPU use, memory utilization and network operation
- Interactive control of CPUs, priorities, policies, shielding and interrupts
- Detailed process and thread information
- Dynamic CPU affinity control for processes, threads and interrupts
- Dynamic display of system and application performance

NightView Source Level Debugger

NightView allows users to simultaneously debug multiple time-critical processes. With NightView, a programmer can change program execution and modify or display data with-out stopping or interrupting the program. Eventpoint conditions such as hit and ignore counts are patched directlyinto an application and executed at full application speed. NightView provides fine-grained control without adversely affecting application deadlines. NightView monitorpoints can display expressions, including function calls, at user-selected locations without stopping a process, thus providing data displays that are synchronized with the application’s algorithms

Language-Sensitive Debugging

NightView supports the debugging of multiple applications written in any combination of C/C++, Ada and Fortran. Allvaria bles and expressions in each program are referenced in its own language syntax. NightView is fully integrated with the NightTrace event analyzer. NightView can insert tracepoints at user-specified locations for live or post-execution analysis with NightTrace



NightView supports multiprocessor, multiprocess debugging in any mix of languages from a single GUI.

More Powerful Than The GNU Debugger

NightView offers many features not available in GNU debuggers (gdb/ddd). NightView advantages include the ability to debug multiple processes from a single session and processes started from scripts. With NightView, patched-in code runs at full speed. While a process is executing, hot patching can modify variables, change execution flow, add eventpoints and add or replace subprograms. Monitorpoints can display expressions and stack variables, and signals can be delivered directly to the process, bypassing the debugger.

Complete Memory Debugging

NightView includes an interactive memory debugger that helps find and eliminate problems during the debug process without code recompilation or relinking. NightView watches for heap memory leaks, monitors the amount of memory an application uses, and tracks how it allocates and frees memory. With its memory debugger enabled, NightView lets users track heap transactions in real-time, thus allowing for more ecient debugging than post-run analysis. Programmers can stop execution, check for problems, test patches and then continue debugging.NightView can detect common heap programming errors, such as uninitialized use of allocated areas, double-frees, use of freed pointers and writing past the end of allocated areas. No recompilation or relinking is required.

GPU Debugging

NightView allows users to simultaneously debug multiple, multi-threaded processes including the execution of CUDA® code on a GPU. The tool provides a variety of ways to display GPU contexts such as GPU threads, blocks, warps and lanes. NightView's powerful data display panels allow a user to filter GPU threads based on user-defined expressions to help isolate massively parallel debugging issues.

NightTrace Event Analyzer

NightTrace is a tool for displaying and analyzing the dynamic behavior of applications, the Linux operating system and the interaction between them.NightTrace can log events from multiple processes executing simultaneously on multiple CPUs or systems. NightTrace can also combine user-defined application events with OS events to present a synchronized view of the entire system. NightTrace allows users to zoom, search, filter, summarize and analyze events. Tracing analysis can be performed live or post-execution.NightTrace was specifically designed to meet the most stringent requirements of time-critical applications.Using synchronized, fast-access hardware clocks and kernel-free primitives, NightTrace tracepoints arelogged with minimal overhead. Tracepoints can be inserted into device drivers, interrupt level code and any user application. Tracepoints can be left in production applications even when not collecting trace data. NightTrace also provides instrumentation for user programs to automatically display function call parameters, entry, return and return values.

Graphical and Interactive

NightTrace graphically displays requested eventsand states along a timeline graph to clearly show the relative timing of events and provide an overall picture of application and operating system activity.NightTrace can locate specific events and zoom in on them with a fine degree of granularity for precise timing observation. The NightTrace graphical display is completely user-configurable and customizable. Configurations can be saved and later recalled, and different configurations can be viewed simultaneously.

Automatic Function Call Tracing

NightTrace's application illumination feature allows programmers to automatically trace application function calls and examine the values of parameters passed and returned. Function call tracing is fully customizable and can provide a complete view of system library activity (e.g. glibc, pthread).

NightProbe Data Monitor

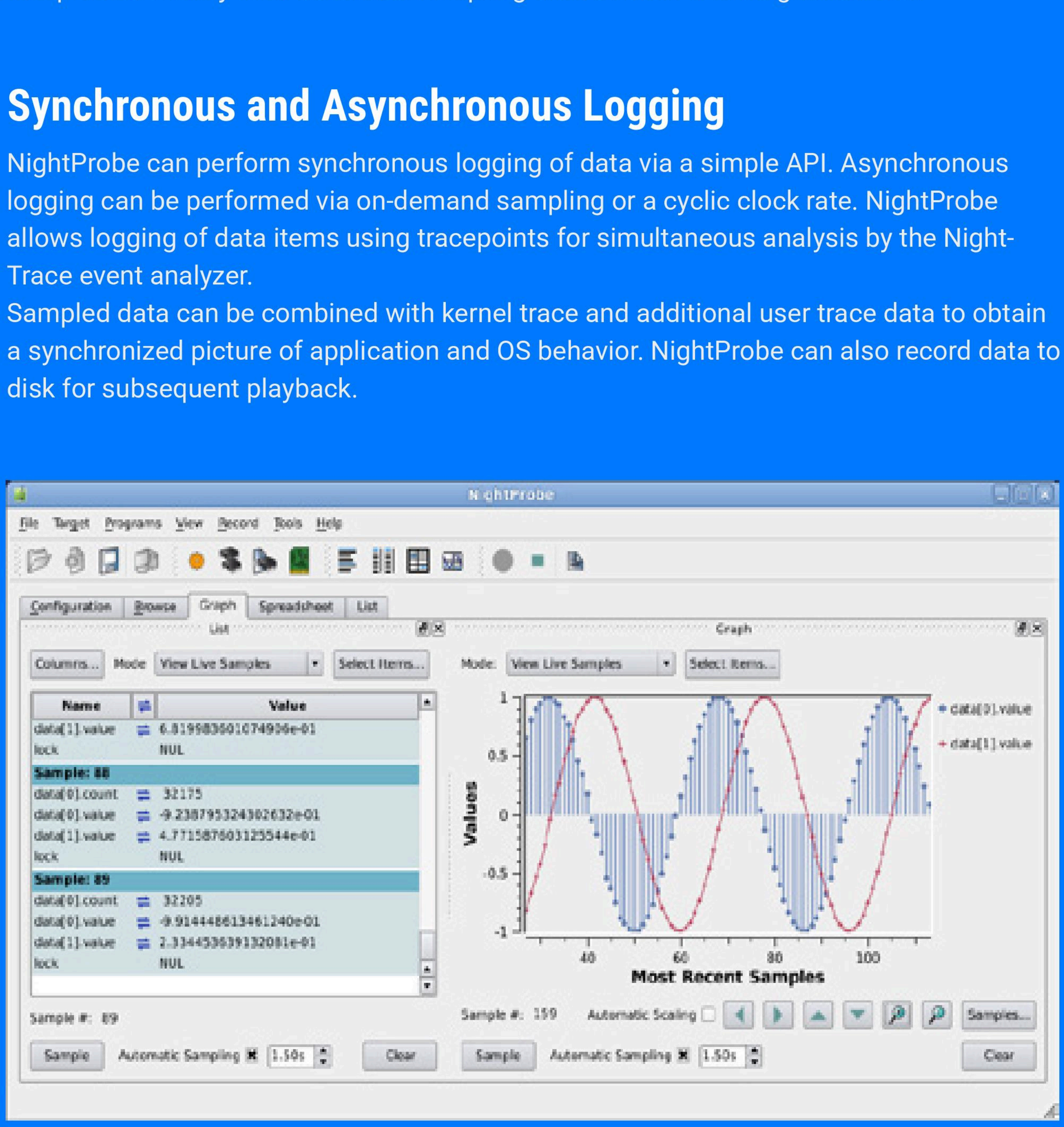
NightProbe is a tool for monitoring, modifying and recording data values from multiple, independent application resources, including programs, shared memory segments and memory devices. NightProbe can be used in development for debugging, analysis, prototyping and fault injection, or in a production environment to create a GUI control panel for program input and output.

NightProbe utilizes a non-intrusive technique of mapping the target resource's address space into its own. Subsequent direct memory reads and writes by NightProbe allow it to sample and modify data without interrupting or otherwise affecting resources.

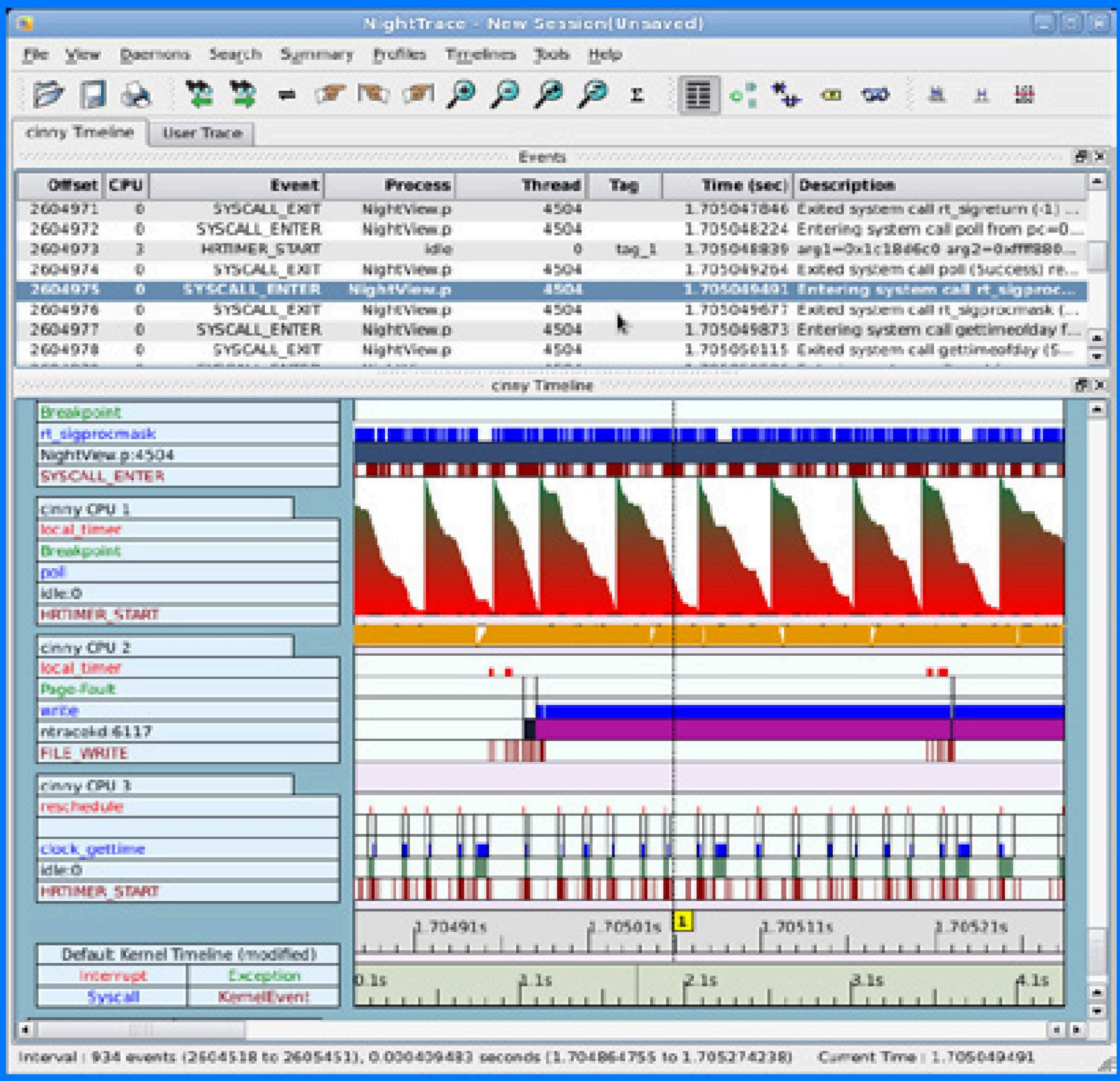
Synchronous and Asynchronous Logging

NightProbe can perform synchronous logging of data via a simple API. Asynchronous logging can be performed via on-demand sampling or a cyclic clock rate. NightProbe allows logging of data items using tracepoints for simultaneous analysis by the NightTrace event analyzer.

Sampled data can be combined with kernel trace and additional user trace data to obtain a synchronized picture of application and OS behavior. NightProbe can also record data to disk for subsequent playback.



NightProbe provides configuration and control for data monitoring and recording



NightTrace's Analysis API allows users to easily create custom programs that monitor or analyze application or system activity.

Kernel Trace Support

By combining system event information such as interrupts, exceptions, context switches, system calls and device accesses together with event information from user applications, NightTrace provides a clear picture of the activities of the kernel at any point during the application's run. NightTrace provides statistical performance data about events and states, including frequency, time of occurrence, duration, gap and minimum and maximum times. Users can create customized state and event conditions by specifying the applicable process, thread, CPU, system and event content. Displays can be customized to yield insight into operating system and application performance and behavior.NightTrace's Analysis API allows users to easily create custom programs that monitor or analyze application or system activity.

Interactive Sampling and Modification

NightProbe offers a variety of data display options including lists, tables, spreadsheets and graphs. Direct modification of user data is accomplished by simply typing new data item values into cells that display the variables. NightProbe allows sampled data to be passed to user applications for live customized display or for post-run analysis.
NightProbe supports scalar and structured data types in C/C++, Ada and Fortranthat have static base addresses. Pointer indirection is also supported. NightProbe scans the symbol table and debug information of user programs allowing the user to browse for data items or specifically enter the names of data items to be monitored. Any application that contains symbol table and debug information may be used with NightProbe.No application source code changes are required.

NightTune System and Application Tuner

NightTune provides a GUI for monitoring and tuning application and system performance. Users can monitor the priority, scheduling policy, CPU assignment and CPU usage of user applications. NightTune also monitors system CPU usage,context switches, interrupts, memory utilization, and disk and network activity.
NightTune can monitor processes individually or in user-specified groups or by CPU. NightTune also displays information about individual threads or tasks within a process. Available information includes library and system call tracking (strace),file descriptor usage and connections, signal states, and detailed memory usage with page-level statistics on residency,locking and NUMA pools. Multiple frames and windows are used to display information allowing users to customize the desired display.

System Tuning by GUI

NightTune's GUI can change the process attributes of an individual thread, task, process or group of processes as a whole. For example, dragging a process icon to a CPU icon binds the process to that processor. The user instantly views the results of the tuning effort both graphically and as text. NightTune provides a hierarchal display of logical CPUs showing multiple cores and hyper-threading. Users can also easily change CPU interrupt assignments.

GPU Device Information

NightTune offers a GUI representation of the parallel computing capabilities of each CUDA device on the system. NightTune provides detailed device info including GPU configuration and capabilities, block configuration, memory capacity and more. A user can also change the configuration settings as needed. NightTune provides dynamic usage data including GPU usage, memory activity and usage, as well as power, fan speed and temperature metrics. Both a topical display and history graphs are available to give insight into GPU usage over time.

NightTune monitors system activity including CPU use, context switches and disk utilization.

NightSim controls all scheduling parameters and displays information about all processes,including executing frequency.

NightSim Application Scheduler

NightSim is a tool for scheduling and monitoring time-critical applications that require predictable, cyclic processexecution. Ideal for simulation applications, NightSim allows developers to dynamically adjust the execution of multiple,coordinated processes, their priorities, scheduling policies and CPU assignments. Users can monitor the performance mof applications by displaying period execution times, minimums and maximums, and optionally pause all application execution when any process overruns its allocated frame.NightSim provides a GUI to the operating system's Frequency-Based Scheduler (FBS), a high-resolution task scheduler that enables processes to run cyclically. NightSim lets users easily configure groups of processes to run on local systems,and save the resulting configurations for reuse. A performance monitor gathers CPU utilization data for processes runningunder the FBS. NightSim may be used during the development, debug and production phases of a simulation model.Configurations can be saved as a script and then re executed to repeat a simulation. NightSim scripts are useful in target environments where GUI processing is prohibited or undesired.

Extensive Performance Statistics

NightSim monitors up to 14 different performance statistics for each process, such as minimum and maximum cycle times.Using these statistics, users can optimize CPU utilization by balancing their load across multiple processors. NightSim displays are customizable, allowing users to select specific statistics and processes to monitor and sort criteria or weighted display.

About Concurrent Real-Time

Concurrent Real-Time is the industry's foremost provider of high-performance real-time computer systems, solutionsand software for commercial and government markets. Its real-time Linux solutions deliver hard real-time performance in support of the world's most sophisticated hardware in-the-loop and man-in-the-loop simulation, high-speed data acquisition, process control and low-latency transaction processing applications. With over 50 years of experience in realtime solutions, Concurrent provides sales and support from offices throughout North America, Europe and Asia.For more information, please visit Concurrent Real-Time at www.concurrent-rt.com

Concurrent Real-Time at www.concurrent-rt.com

Visit our SIMulation Workbench Wiki at wiki.simwb.com

©2017 Concurrent Real-Time, Inc. Information subject to change without notice. Concurrent Real-Time and its logo are registered trademarks of Concurrent. All Concurrent product names are trademarks or registered trademarks of Concurrent, while all other product names are trademarks or registered trademarks of their respective owners. The registered trademark of Linux is used pursuant to a sublicense from the Linux Mark Institute, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. All rights reserved.