KTAU: Kernel Tuning and Analysis Utilities

Performance Research Lab-University of Oregon
http://www.cs.uoregon.edu/research/ktau

KTAU Overview

The University of Oregon is working in partnership with Argonne National Labs on the ZeptoOS project to study flexible operating and runtime systems for petascale architectures with heterogeneous computing fabrics. Oregon is looking specifically at the problem of performance tools and metrics that allow scientists working on HPC OS/Rs to leverage robust measurement infrastructure and shared tool infrastructure for performance assessment and benchmarking

Kernel Tuning and Analysis Utilities (KTAU) is a toolkit for profiling and tracing the Linux kernel. The toolkit's main strength is in analyzing program behavior within the kernel space. It produces Context-of-Execution based (or per-process) profiles/traces. Coupled with TAU, KTAU can produce integrated performance information of program executing in user and kernel spaces. As the name suggests, it also allows performance analysis, tuning and debugging of the kernel as a whole by aggregating the per-process data.

KTAU Design

KTAU provides two performance analysis facilities, profiling and tracing. Profiling is a technique used to measure the performance of a particular code section, which is wrapped by a set of instrumentation points. This technique portraits a summarized view of performance data with the advantage of a compact profile data size. The measurement of each code section is taken and stored with respect to the context of a process. Organizing performance information in this manner allows selective access directly to the kernel performance data of a certain process, while system-wide information can be acquired by collecting the data from all processes.

KTAU provides several frameworks to be used in various environment settings. In case the user-level source code is available, it can be instrumented directly with KTAU user-API, which access the kernel performance data through Linux "proc" file system. Base on this framework, KTAU extends TAU (Tuning and Analysis Utility) to enable the performance measurement of both user and kernel space, and allow them to be merged to provide the integrated view. When the user-level source code is not available, KTAUD can be used to measure the kernel performance of the user-level application. KTAUD is a daemon that periodically accesses the kernel performance data of specified processes.

While profiling provides a summary view of performance data, trace can be use to provide a detail view of process executions with respect to time. KTAU uses the same kernel instrumentation as for profiling to generate an execution trace. The trace can be view in a scope of a process or system-wide. KTAU trace data can be converted into various formats using data-conversion tools provided by TAU. The two main supporting formats are VTF and SLOG2, which can be used with existing trace analyzers such as "Vampir", and "JumpShot" respectively.


Kernel Profile Analysis
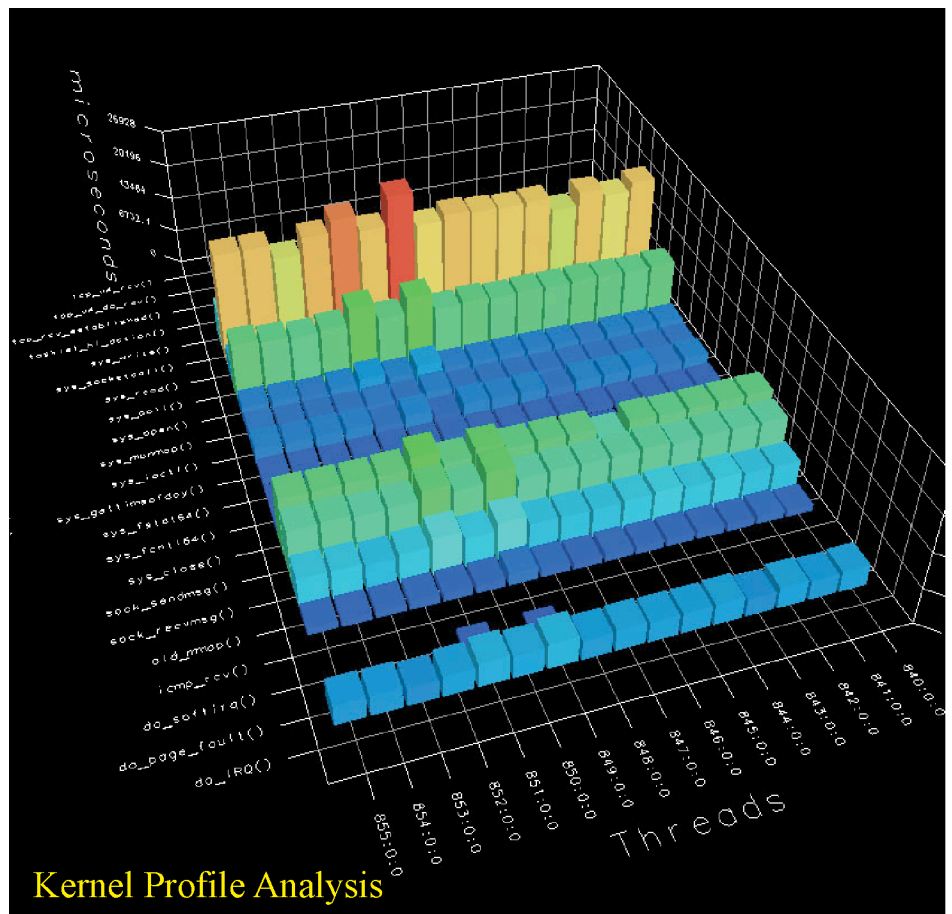
Features List

- Linux kernel performance analysis using profiling and tracing techniques
- Provide a system-wide view of performance that can provide a kernel-as-a-whole performance perspective
- Provide a process-centric view of application performance within the kernel.
- Integrated user/kernel space performance analysis
- Supported by TAU performance measurement system
- Support for *Paraprof* parallel profile analyzer
- Support for *Vampir/Jumpshot* trace analyzers
- Linux 2.4/2.6 ported to ia32 and ppc32
- BG/L ionode kernel with ZeptoOS ported
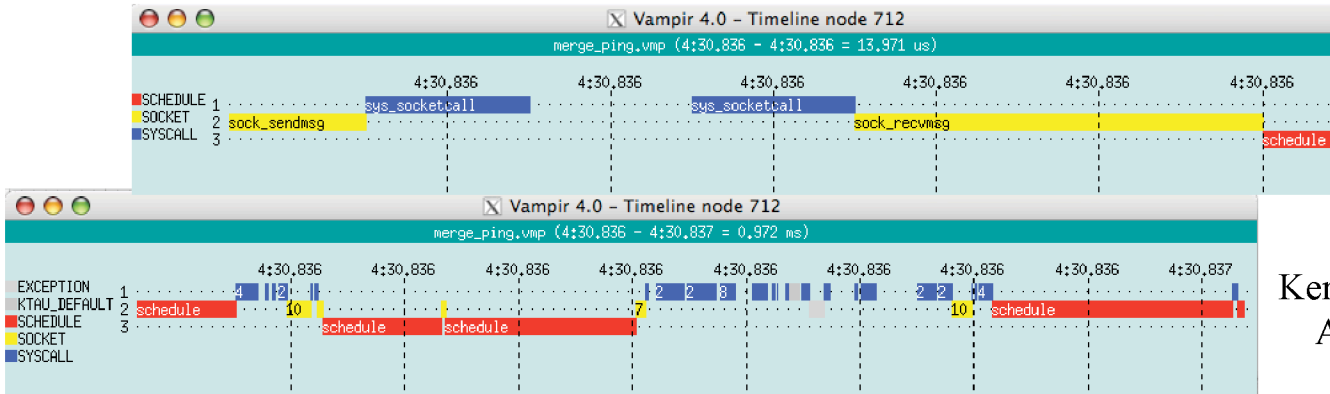
Kernel Trace Analysis

KTAU Instrumentation

Instrumentation is inserted into the Linux kernel source allowing KTAU to intercept kernel execution paths and measure performance data. Several key areas of the Linux kernel are instrumented including interrupt handling, bottomhalves handling, scheduling, system calls, network subsystem, signal handling, exceptions, and traps
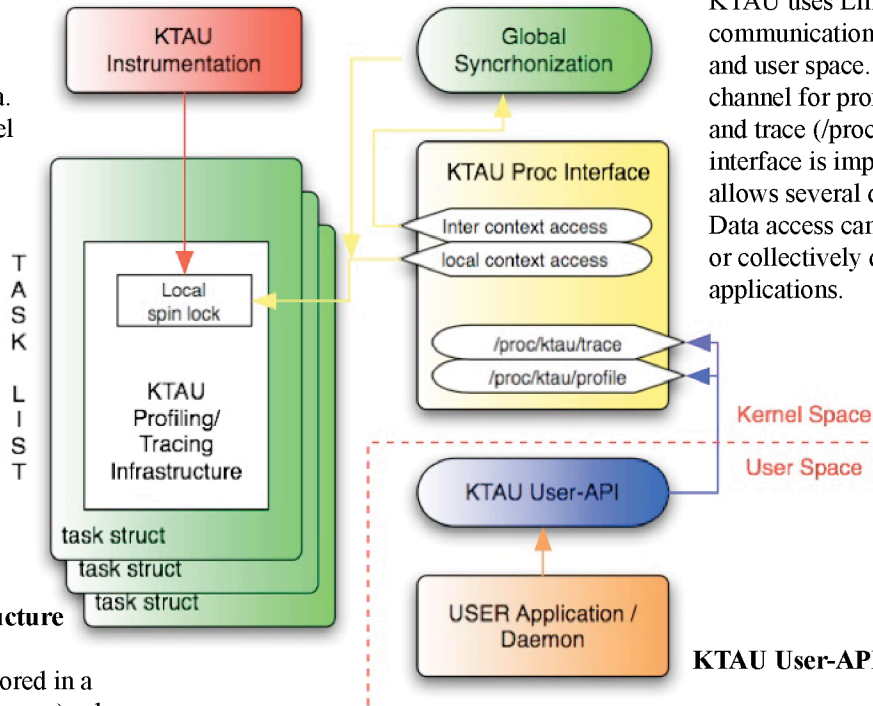
KTAU Proc Interface

KTAU uses Linux proc-filesystem as a communication channel between kernel and user space. It provides a dedicated channel for profile (/proc/ktau/profile) and trace (/proc/ktau/trace). The proc interface is implemented on ioctl, which allows several data-access schemes. Data access can be done selectively or collectively depends on the applications.

KTAU Profiling/Tracing Infrastructure

Performance data is managed and stored in a Context-Of-Execution based (per process) scheme. Each Linux task_struct is extended to store the current performance information of each process including a table containing a profile of kernel routines, and a ring-buffer containing records of kernel trace.

KTAU User-API Library

KTAU provides a programming interface for application to simplify the data access mechanism.