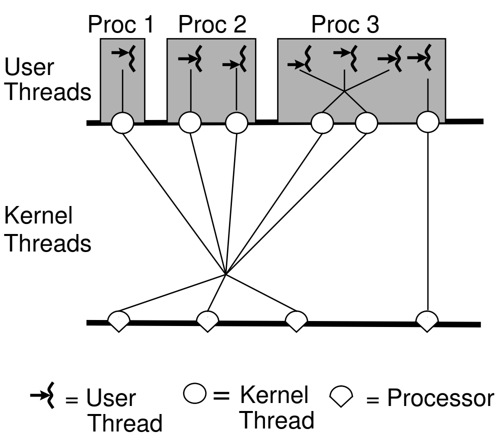
Solaris Multithreading

TECHNOLOGY SPEC

Multithreading (MT) is a modern programming paradigm which enables applications to move from strictly serial processing to concurrent processing. In so doing, programmers can exploit the software benefits of increased concurrency, parallelism, and simpler representation of many tasks. Concurrency increases efficiency on all systems, including uni-processors. Parallelism exploits the performance potential of new, low-cost, high-performance multiprocessors.

Solaris™ has pioneered the development and use of multithreading for application programmers and in the operating system itself. Two facilities are provided:

- Two-level architecture and implementation for greater efficiency and flexibility.
- Application Programming Interfaces (APIs) for both the full Solaris threads functionality and the emerging POSIX 1003.1c *pthreads* interface.



Solaris Multithreading Two-level Architecture

Architecture: The Solaris multithreading architecture is based on a two-level model of application-space threads and underlying kernel threads. This differs from conventional implementations in that it provides far more efficient use of system resources and imposes dramatically fewer limits on the programmers use of threads, since thousands of threads can be utilized with minimal overhead. We also include a new low-overhead synchronization model which allows fast communication between processes.

Application Programming Interfaces: The Solaris *libthread* API provided the basis for much of the draft POSIX *pthread* submissions. POSIX *pthreads* are still being finalized. Programmers using *libthread* will find it very simple to use *pthreads*. An implementation of the current draft POSIX standard is available from SunSoft, and the final standard will be part of future Solaris releases.

Solaris multithreading provides application developers with a number of important advantages:

Performance: Applications can improve performance in two fundamental ways. They can overlap processing of application, kernel, and I/O functions, and they can split functions into multiple threads on multiple processors. Applications can also use the powerful new synchronization architecture between processes.

SunSoft
A Sun Microsystems, Inc. Business

Responsiveness: Many applications appear to be unresponsive to users because they block user interaction while performing other functions, or because they perform one function at a time. Multithreading lets an application create separate threads for these tasks, so that the users always see the application at its greatest responsiveness.

Programming Flexibility: As the most complete multithreading architecture and implementation, Solaris multithreading gives the developer the freedom to chose a design model from a variety of concurrency alternatives which fit the application's needs.

Full UNIX® Compatibility and Standards Compliance: Solaris multithreading preserves the full capabilities of UNIX signals, while providing POSIX standards compliance.

A Complete Platform for Multithreading Development and Deployment: Using multithreading effectively requires much more than a threads library. It also requires kernel support, thread-safe libraries for system functions, thread-aware tools such as debuggers and performance profilers, as well as full documentation and training. Solaris provides a complete multithreading platform for developers, as well as the performance and efficiency for deployment.

Performance Comparisons

Threads allow a program to execute more efficiently than before. On multiprocessor systems, a threaded program can take advantage of the additional CPUs without any additional work on the part of either the user or the programmer. On uniprocessor machines, the exact same program can run faster due to the overlapping of multiple kernel tasks, user tasks, and I/O.

Solaris thread management is efficient and fast. The extensive design and tuning of Solaris provide the most efficient primitives for developing and deploying modern, high-concurrency, multithreaded applications.

	*Solaris**	*Windows NT™*
Create thread	250	3250
Create and exit thread	800	4850

**Measurements in microseconds, on identical x86 platforms.*

Using SunSoft's automatic-threading Fortran compiler (which uses Solaris Threads) nearly doubles the performance of many programs on a two-CPU desktop machine. Due to these impressive results, Solaris is the only OS with published multiple-CPU SPECfp numbers. (June 1994 SPEC Newsletter)

	Tomcatv	*Swm256*	*Su2cor*	*Hydro2d*
SS20 (1 CPU)	93.0	50.6	108.7	101.4
SS20 (2 CPU)	148.0	94.9	191.4	175.4

Status of POSIX Draft Standard

As of August, 1994, POSIX has produced a draft 9. It, or a closely similar successor, is expected to soon become the final standard. Solaris threads today have almost all the functionality of POSIX draft 9, and a very similar API, minimizing the transition path to POSIX. SunSoft will ship a fully POSIX-compliant library after the standard is approved, and will continue to support the current library well into the future.

A developer can safely write to Solaris threads today, enjoying the full advantages of multithreading, and easily move to POSIX threads to get the added feature of multiplatform portability tomorrow.

SunSoft
A Sun Microsystems, Inc. Business

2550 Garcia Avenue, Mountain View, CA 94043-1100 USA ● (415) 960-3200 ● Fax (415) 961-6070

Printed in USA 10/94 94338-002
U.S. Distribution Only