Fast User-Level Inter-thread Communication and Synchronisation

Author: Li Lin Supervisor: Dr Keivn Vella

University of Malta

Abstract

This project's objective is to design and implement several user-level inter-thread communication constructs for SMASH. Since communication and synchronization among threads in a process is heavily used in most multi-threading programs,the efficiency of inter-thread communications affects the efficiency of the whole program. Traditionally, those communication and synchronization constructs were built with locks, but this lock-based approach has some problems such as potential deadlock starvation and priority inversion. In order to solve these problems, lock-free algorithms were invented. Recently there is a lot of interest in lock-free algorithms in the research community. In this project, all the inter-thread communication constructs considered have two implementations: the lock-based implementation and the lock-free implementation.

Inter-thread communication though the kernel

- Involve system calls.
- May block the kernel thread.
- Expensive.

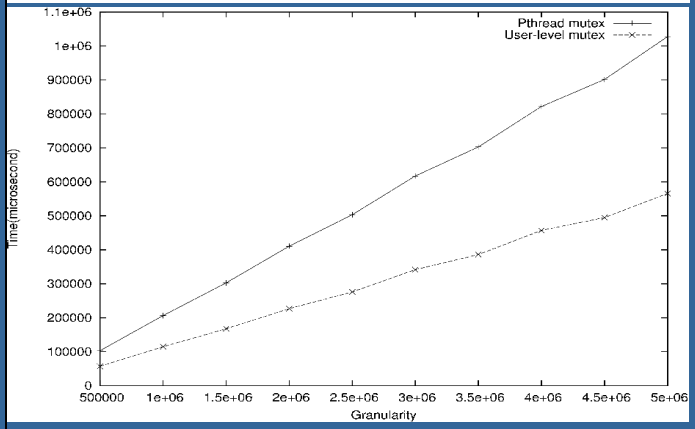
Implemented constructs

- Mutex
- Semaphore
- Message queue
- The CSP communication channel

Results

Several benchmarks were conducted on a machine with an Intel Core2 Duo CPU, 4Mb L2 cache and 2Gb memory. The operating system was Debian unstable.

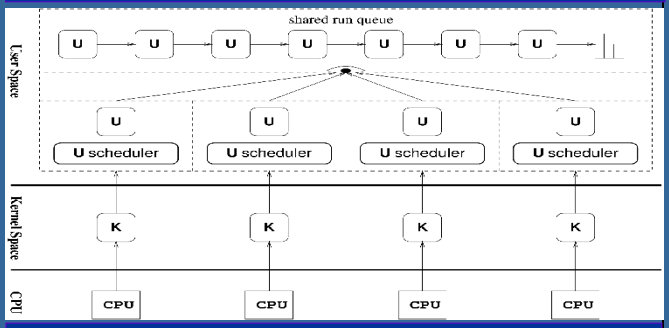
The following graph shows the performances of the same concurrent program when using pthread mutex and our user-level mutex.



SMASH

- A user-level thread system on top of pthreads developed by Kurt.
- Runs on both uniprocessor machine and SMPs.
- Very fast context switch

Architecture of SMASH

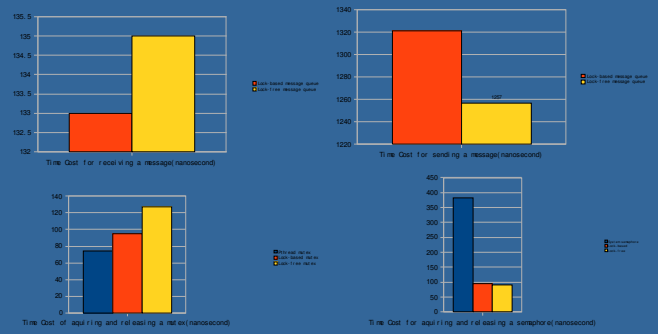


Inter-thread communication in user space

- No vertical switches
- Only blocks user-level threads
- Does not involve the kernel

Lock-based VS Lock-free

Spin locks	Lock-free algorithm
•Easy to design •May have high memory contention •May have live lock •May have priority inversion	•Hard to design •Low memory contention in general •Free of dead/live lock •No priority inversion



Acknowledgements

In this project, several user-level inter-thread communication constructs are developed for SMASH, and some improvements has been shown. Additionally, lock-free algorithms are exploited. Although due to the hardware limit, the advantage of these lock-free implementations can not be shown, we still believe that lock-free algorithms will perform better than lock-based ones when a SMP system is under high contention.