California Institute of Technology
Department of Computer Science
Computer Architecture

CS184b, Spring 2003 Assignment 5: Measure Communication Monday, April 28

Due: Monday, May 12, 9:00AM

Measure and report the latency and throughput of thread-to-thread communications under linux for the following cases:

- thread-to-thread on same processor, no SMT
- thread-to-thread on same processor, with SMT
- thread-to-thread on different processors on an SMP machine
- process-to-process on different machines (same switch)

Please work as a group for this assignment. Divide work. Support each other in developing experiments, analysis, collecting and corraborating results.

You may need to explore more than one mechanism or communication style to find the one which gives the best performance in each case. *e.g.*

- OS sockets?
- OS Interprocess Communication Primitives (IPC)
- direct use of mapped, shared-memory
- schedule/yield/interrupt?? polling?
- UDP?
- TCP?

You will likely to need use a number of tools to collect your data. *e.g.*

- Pentium cycle and event counters
- Read CPU ID and/or affect process scheduling affinity?

You may want to look at:

- SCORE Stream implementation (`/cs/research/ic/software/SCORE/runtime/ScoreStream.cc` – you will at least need to strip out the time synchronization bookkeeping which this has embedded for simulations in order to get a lightweight/efficient version.)
- A compiled SCORE operator, such as `/cs/research/ic/software/SCORE/common/example_app/sub_add.cc`, provides and example of both using IPC `msgsnd` and of spawning a pthreads thread.
- `/cs/research/ic/develop/vpr.cntr/timer/timer.c` example of using Pentinum cycle counter. The command to read the cycle time stamp is `rdtscll` (read time stamp counter long long). `asm/msr.h` is the header for "machine specific registers." You may want to use some of the event counters/modes beyond the time stamp counter.

- In `/home/rafi/src/which_cpu/`, Rafi Rubin has extracted some code from `procps` to identify which processor a process is running on. This version is a bit heavy handed. Maybe it's possible to get processor identification in a more lightweight manner using one of the MSR registers?

One possible division of work is to designate an expert for each communication mechanism. Individuals may also want to take the lead on particular measurement tools and techniques.

You may need to use different machines for the experiments. *e.g.*

- matrix3x for SMT experiments
- department single-processor, grid-engine machines for single-processor experiments
- matrix1x and/or 4-processor department grid-engine machines for non-SMT, SMP experiments

Note differing clock rates and normalize results as appropriate.

Turnin:

1. For each case, report min/avg/max latency and throughput. If there is a latency vs. throughput curve, plot the results.
2. Breakdown cycles and explain where time is going. *e.g.* How many cycles/issue slots on sending thread? on receiving thread? cycles on wire (as appropriate)? context switching? time in cache hierarcy?
3. Detail your division of labor
4. Detail experimental techniques used:
   - how collected data
   - how arranged experiments
   - mechanisms explored (which tried? which gave best results?)
   - benchmark code used
5. For cycle breakdowns and experimental techniques, indicate who wrote which section.