Introduction to Lab 1

Andreas Sandberg <andreas.sandberg@it.uu.se>
Division of Computer Systems
Dept. of Information Technology
Uppsala University

2011-09-12

What is Uppmax?

Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) is Uppsala University’s resource of high-performance computers and know-how of high-performance computing (HPC).¹

The Kalkyl Cluster
Specifications

- Runs Scientific Linux 6.1 (A RedHat Enterprise Linux clone customized for scientific applications)
- 384 nodes with:
  - Dual Intel Xeon 5520 nodes @ 2.26GHz (Nehalem, quad-core)
  - 24 GB RAM (some nodes have up to 72 GB)
  - Gigabit Ethernet & Infiniband

What is lab 1?
...or why are they dragging us to the computer lab at 8am?

The purpose of this assignment is to give insights into:
1. how a cache works
2. how program execution is affected by cache parameters
3. how to tune an application for a specific cache configuration

¹http://www.uppmax.uu.se/
The Kalkyl Cluster

Logging in transferring files

- Use SSH to connect to `kalkyl.uppmax.uu.se`
  - `ssh -Y username@kalkyl.uppmax.uu.se`
  - `-Y` – Enables X-forwarding
- Transfer files using the `scp` command
  - `scp ./foo username@kalkyl.uppmax.uu.se:bar/`
  - Transfers the file `./foo` to the directory `bar` in your home directory on Uppmax

Submitting interactive jobs

- You should not run Simics on the login node. Long running jobs on the login node will be terminated.
- Use `interactive -A g2011132 -t 4:00:00`
  - Starts an interactive shell in a screen terminal
  - `-A g2011132`—Request the course project for CPU time accounting
  - `-t 4:00:00`—Expected runtime
- Jobs running longer than the specified runtime will be terminated

What is Simics?

You already know this, so let's get down to business!

Target and Host

- The `target` is the simulated system
- The `host` is the machine running Simics
- The prompts:
  - `target#` – the target system’s prompt
  - `host$` – the host system’s prompt
  - `simics>` – Simic’s command prompt
Simics commands

- `simics> help`
  - Does exactly what you would expect...

- `simics> run`
  - Starts or continues the simulation
  - `Ctrl-C or simics> stop` breaks the execution if Simics is running

- `simics> run 1000`
  - Runs another 1000 instructions and stops

- `simics> quit`
  - Exits Simics

Hostfs

- Module to mount the `host` machine's file system in the `target` machine.

- `target# mount /host`
  - mounts the host's file system on `/host`

Simics snapshots

... or how to travel in time.

- Allows you to store the complete state of a machine
- You can restart Simics with the data in the snapshot
- Convenient way to “fast forward” through the boring boot processes

- `simics> write-configuration` (stores a snapshot of a machine)
  - E.g.: `simics> write-configuration ./my_snapshot`
Simics snapshots

Loading them

- simics> read-configuration
  - Loads a snapshot.
  - E.g.: simics> read-configuration ./my_snapshot
- host$ ./simics -c ./my_snapshot
  - Starts Simics using a snapshot.

Magic instructions

- Allows the target machine to communicate with Simics
- Uses no-ops in the target architecture
- Simics modules can hook into a hap (callback) to handle magic instructions

A special case is the magic breakpoint
- Causes Simics to stop the simulation
- simics> enable-magic-breakpoint
  - Enables magic breakpoints.
- simics> disable-magic-breakpoint
  - Disables magic breakpoints.

```c
#include <simics/magic-instruction.h>
static void foo ()
{
  MAGIC_BREAKPOINT;
}
```
- ...but you have to make sure that Simic's include directory is included in the compiler's include path!
Simics windows
The host/Simics terminal

Simics and Target windows
The target window

Introduction Uppmax Using Simics Summary

Introduction Uppmax Using Simics Summary

Important dates

Groups:

Prep. Room 1412 & 1549, 13:15–17:00

A 2011-09-14, Room 1412, 13:15–17:00
B 2011-09-15, Room 1412, 08:15–12:00
C 2011-09-15, Room 1412, 13:15–17:00

Deadline: 2011-09-19 15:14

Summary

You will:

- Simulate a 64-bit x86 machine
- Implement a cache simulator extension to Simics
- Optimize a C implementation of matrix-matrix multiplication

Complete lab manual on the course homepage

http://www.it.uu.se/edu/course/homepage/avdark/ht11
Thou shalt not follow the NULL pointer, for chaos and madness await thee at its end.\(^3\)