Computer Architecture 2012/2013

Assignment 3b

Date: November 27th, 2012
Deadline: December 9th 2012, 23:59

Contents

1 Overview

Reminder: The purpose of assignment series 3 is to evaluate the cache behavior of programs using MGSim, using MIPS code as made possible by assignment series 2.

The goal of assignment 3b is to compare the memory performance of various programs with different architectural parameters.

2 Instructions

- For this assignment, you can work in groups of 2.
- Read this entire document before you start.
- Your must submit a compressed tarball[1], named after your last name and student ID, containing:
  - the files that you have produced during the assigment.
  - a file `report.rst` containing your write ups to open questions using reStructured Text. This must also contain your full name and student ID. Ensure that `report.rst` is valid by using `rst2html`.
  - any related diagram or program output.
- Your submission must be sent by e-mail before the deadline, at the e-mail address given by the assistants. Do not send your submission to the mailing list!

3 Prerequisites

For this assignment, MGSim version 3.3 is recommended.

3.1 Using MGSim 3.3 as installed on LIACS

MGSim 3.3 for MIPSel has been preinstalled in `/home/rcposs/opt/bin`.

To set up the environment, use the following commands with bash:

```
export PATH=/home/rcposs/opt/bin:$PATH
export MANPATH=/home/rcposs/opt/share/man:$MANPATH
export LD_LIBRARY_PATH=/home/rcposs/opt/lib:$LD_LIBRARY_PATH
export MGSIM=/home/rcposs/opt/bin/mgsim-mipsel
```

or, with tcsh:

```
set path=(/home/rcposs/opt/bin $path)
setenv LD_LIBRARY_PATH /home/rcposs/opt/lib:$LD_LIBRARY_PATH
setenv MANPATH /home/rcposs/opt/share/man:$MANPATH
setenv MGSIM /home/rcposs/opt/bin/mgsim-mipsel
```

3.2 Using MGSim 3.2 from a previous assignment

(Not directly recommended, unless you would really enjoy using your own MIPS implementation.)

MGSim 3.2 contains a bug: if a program requests termination of the simulator, for example when control reaches the end of `main`, the simulator will not print statistics. You can fix this as follows:

1. modify `compile.sh` to use `minicrt-mips32.s` instead of `minicrt-mips33.s`

[1] A compressed tarball is created with `tar -czf xxxx.tgz ....`

2. in the ISA simulation code, change the implementation of the BREAK instruction to:

   ```
   GetKernel()->Stop();
   ```

   (instead of `ExecDebug...` as previously advertised)

Moreover, you cannot use the script `runall.sh` and the provided `.ini` files with MGSim 3.2. Instead:

3. copy `minisim32.ini` as many times as you need different hardware configurations;

4. modify each copy as desired;

5. make your own script `runall32.sh` using a single execution loop and `COMMAND` set to `debug mem; run; stats; quit` (note the extra `stats` to ensure statistics are printed).

3.3 Using MGSim 3.3 on your own computer

To install your own copy of MGSim 3.3, follow the steps of

http://staff.science.uva.nl/~poss/ca2012/INSTALL.html

using `--target=mipsel` when configuring MGSim.

4 Getting acquainted with the programs

4.1 “Burst” and “Reuse”

1. Look at the *source code* of `burst.c` and `reuse.c`.

2. How many dynamic memory loads do they define? Express this number as a function of the value of `N` in your report.

3. Without running these programs, looking at their code alone, what difference do you expect to observe in their behavior? Explain in one paragraph in your report.

4.2 “Matmul” and “Matmul2”

Same 3 questions as above.

4.3 Compiling the programs

1. Look at the source code of the script `compile.sh` and explain what it does (in your report).

2. Run the script to compile the programs.

5 Getting acquainted with the platform configurations

MGSim 3.3 introduces the ability to specify multiple configuration files on the command line, so they are automatically combined to define a platform.

5.1 Memory topology

1. Use the following commands:

   ```
   mipsel-mgsim -c minisim33.ini \
       -I mem_directddr.ini -I l1_4k_1assoc.ini \
       -T test.dot -n reuse-10
   dot -Tpng -o test.png test.dot
   ```

   then open the resulting file `test.png`. Explain what you see.

2. Try the different `mem_*.ini` configuration files. You can also view them in a text editor. Explain in your report the differences between them, using the generated diagrams and the information contained in the manual page mgsimdoc(7). (Use `man mgsimdoc` to view this manual page.)

5.2 Cache layouts

1. Using the existing `l1_*.ini` and `l2_*.ini` files as examples, create configurations for a 16KiB L1 D-cache and a 32KiB L2 cache. Copy them in your submission directory.

2. Copy the text of the script `runall.sh` in your report, and annotate each line to explain what it does.

6 Behavior analysis across platforms

6.1 Expected vs. observed

1. Run the following command:

   ```
   ./runall.sh reuse-1000
   ```

   This command generates trace files in the `output` subdirectory automatically.

2. Using your analysis tools from assignment 3a, compute the latency distribution and statistics of `reuse-1000`. Place a copy in your submission.

3. You have estimated above the expected number of dynamic memory loads. Does your measurement at step #2 confirm or infirm your prediction? How do you explain the divergence, if any? Use your results from #2 to illustrate your argument.

4. Based on your prediction of dynamic load count, what difference of execution time do you expect between `reuse-10` and `reuse-1000`?

5. Compare experimentally the performance of `reuse-10` against `reuse-1000`, by analyzing `reuse-10` as per steps #1 and #2.

   - Do your observations confirm/infirm your prediction for the memory type "serial"? How do you explain this?
   - Do your observations confirm/infirm your prediction for the memory type "directddr"? How do you explain this?

   Each time, you can illustrate using the metrics extracted by your tools from assignment 3a.

6.2 Locality matters

1. Using the steps above, compare experimentally the performance of `burst-10` against `burst-1000`. Provide the statistics from your tools from assignment 3a to illustrate. Does your observation match your expectation?

2. Based on your understanding so far, rephrase (or repeat) your expectation as to the difference of behavior between `burst-1000` and `reuse-1000`.

3. Compare experimentally the performance of `burst-1000` against `reuse-1000`. Provide the statistics from your tools from assignment 3a to illustrate. Does your observation match your expectation? Why?

6.3 Matrix-matrix multiply

1. Using the techniques discovered so far, extract the execution times of the two `matmul` programs across multiple platforms.

2. Between the L1 overall cache size, L2 overall cache size, L1 or L2 associativity, what are the parameters that have the most impact on the performance? Why?