MemSpyer User's guide

Antonis Danalis
danalis@ics.forth.gr

Institute of Computer Science (ICS)
FORTH Greece

Contents

1 Installation

In order to install this package, you need

- the package itself (ftp://ftp.ics.forth.gr/pub/memspyer.tar.gz)
- SunOS version 4.1.4
- The Shade package from SUN (esp. libshade.a, libshade_p.a, libspix.a, libspix_p.a)
- gdb (ftp://sunsite.unc.edu/pub/gnu/gdb/)
- addr2line (a module of binutils ftp://sunsite.unc.edu/pub/gnu/binutils/)

1.1 The source tree

First unzip and extract the archive.

```
> gunzip memspyer.tar.gz
> tar xvf memspyer.tar
```

The package will extract under the directory memspyer. In the root directory (`memspyer/`) you should find a script called check_requirements and a README file with minimal installation instructions.
The package installs into the following directories:

`memspyer/interface/`	Source code for the GUI
`memspyer/shade/cachesim5`	Source code for the cache simulator
`memspyer/lib/`	Libraries
`memspyer/include/`	Header files
`memspyer/bin/`	Executable (target) files
`memspyer/demo/`	Demonstration programs

1.2 Configuration - Requirements

At **compile time** the package will need libshade.a, libshade_p.a, libspix.a and libspix_p.a libraries. You should put them under `/memspyer/lib/` or set the environment variable **LD_LIBRARY_PATH** to include them.
At **run time** memspyer uses gdb and addr2line so you should install them in `/memspyer/bin` or set the local variable **path** to include them.

2 Starting the program

To use the program you should add `/memspyer/bin` in your local variable **path**

```
> set path=($path /memspyer/bin) (for csh like shells)
> PATH=$PATH:/memspyer/bin (for sh like shells)
```

and then run memspyer

```
> memspyer
```

2.1 Demo programs

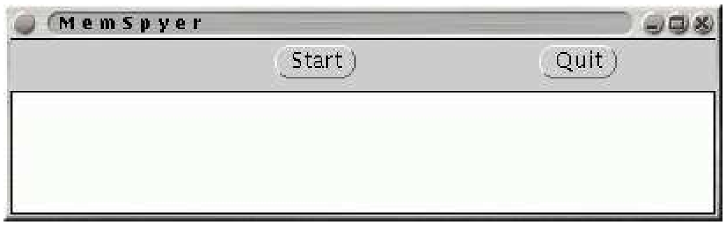
In `/memspyer/demo` you should find the source code of some programs that demonstrate memspyer's functionality.

2.2 Other programs

To profile a program, you should have it in your **path** or give the full path (i.e. `/usr/bin/ls`) in memspyer.

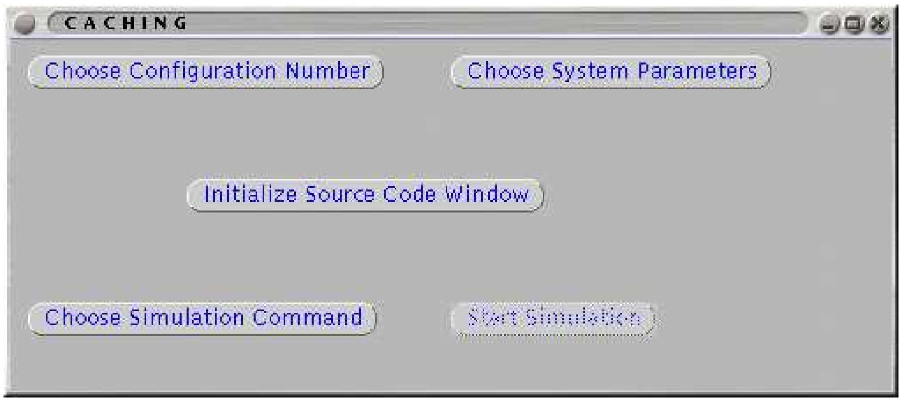
3 Using the program

The first/last frame gives you the choice of starting/leaving the program.

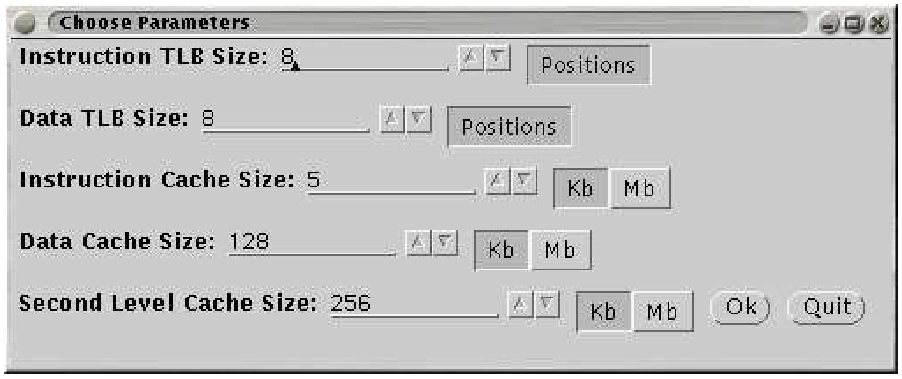


If you select Start the frame disappears and two new frames appear.
The first (Initialization) is the stdout and stderr of memspyer. The second (CACHING) is the "Control Panel" of the program.

3.1 The Control Panel



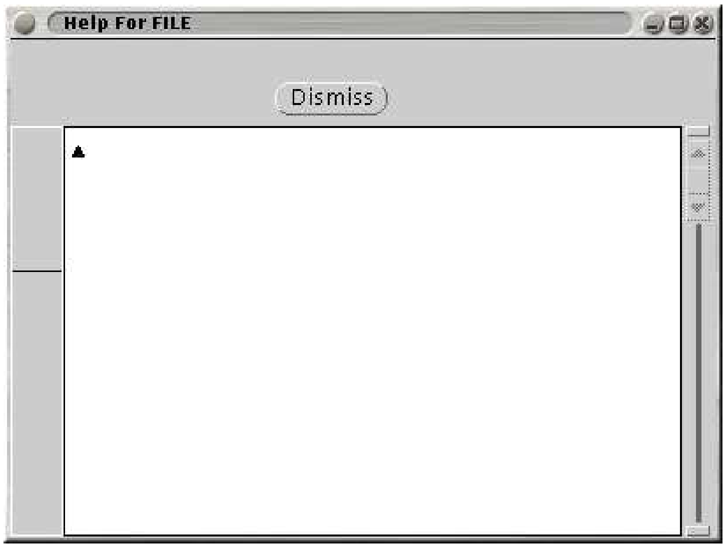
The first button (Choose Configuration Number) provides the ability to choose the number of executions of the program beeing profiled.
The second button (Choose System Parameters) prompts you with a new frame where you can set the attributes of the virtual environment.



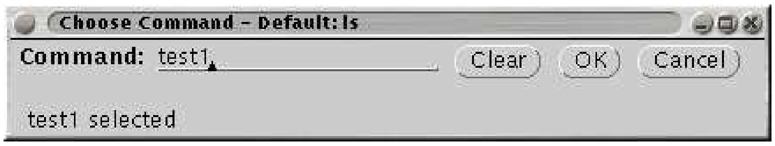
This attributes are:

- Instruction TLB Size
- Data TLB Size
- Instruction Cache Size
- Data Cache Size
- Second Level Cache Size

The third button (Initialize Source Code Window) starts a new window where the source code of the simulated program will be drawn [1].



The fourth button (Choose Simulation Command) enables user to select the program that will be simulated.



When the user chooses a program, the Start Simulation button becomes active. By pressing this button, the Control Panel disappears and three new windows appear:

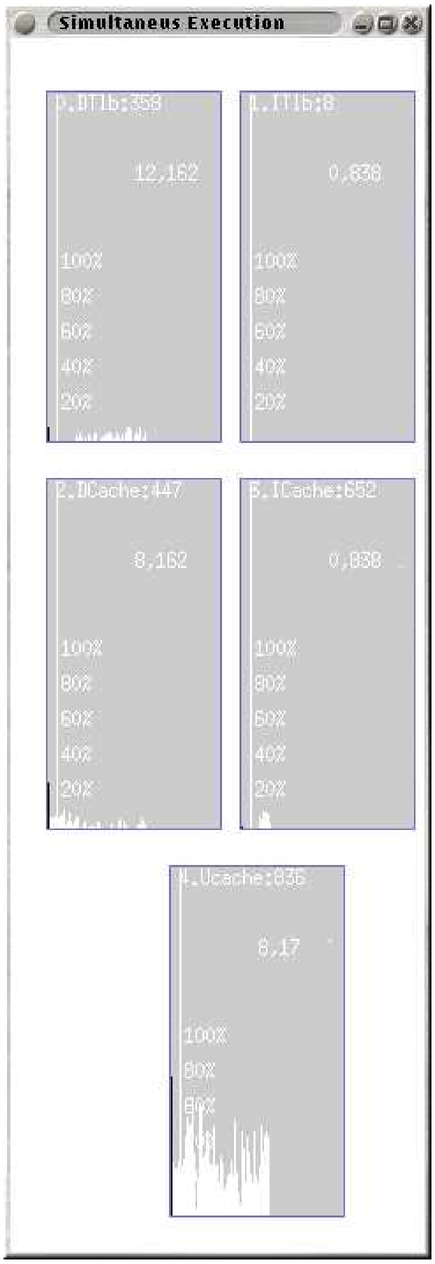
- The charts window
- The execution control window
- The tty

[1]This feature is only for programs compiled with **-g** option

3.2 The Charts window

This window includes five charts corresponding to the misses in:

1. Data TLB (DTlb)
2. Instruction TLB (ITlb)
3. Data Cache (DCache)
4. Instruction Cache (ICache)
5. Secondary Unified Cache (Ucache)



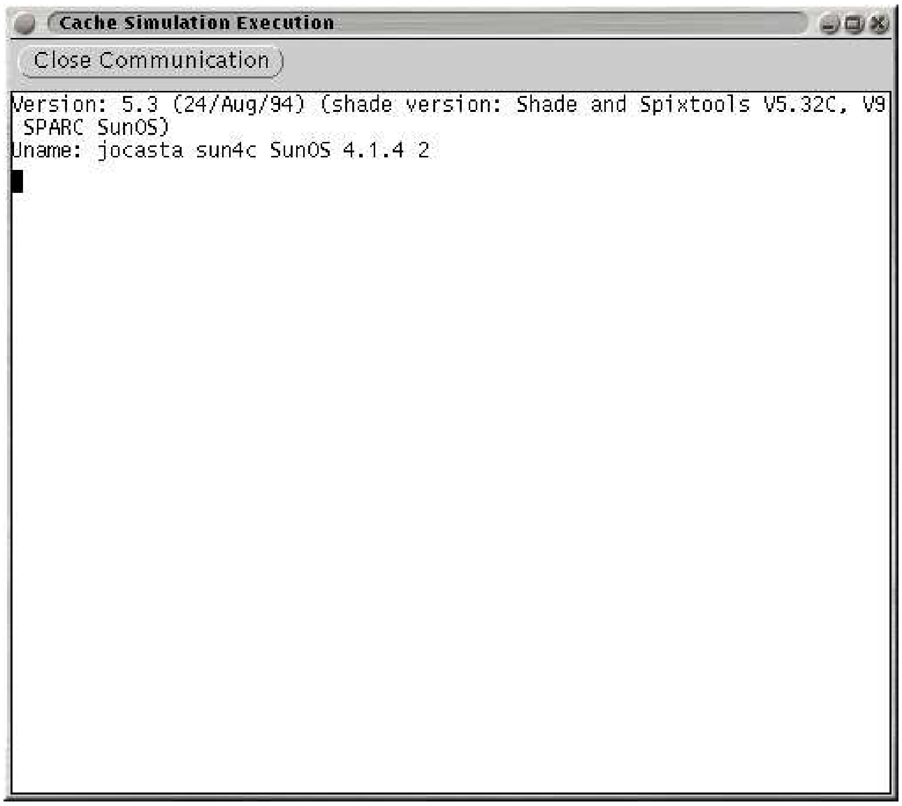
While a program is being profiled, memspyer plots a graph in each chart. Each graph presents the percentage of misses that occured. If the misses are to few (or to many), the y axis automatically zooms in (or out) for better view.

3.3 The Execution control window

This frame provides the user the ability to control the execution of the profiled program. The available buttons start/continue, stop and run step by step the program. Additionally there is a dismiss button that destroys the window.

3.4 The TTY

This window is the tty of the profiled program that displays the stdout and stderr of the program. On this window there is just one button (close communication) that closes the communication with the shade proccess and terminates the profiling.



4 Demo programs

The package includes some demo programs under the directory `memspyer/demo`. The first one (test0) is a very simple program just to verify that everything is working well. The second one (test1) consists of two source files, to demonstrate the source code window useness. The third program (test2) have several stages. In each stage the memory I/O's maximize the misses occuring at one of the three different memory types (DTlb,DCache,Ucache) keeping low the misses at the others.