Web-based training on computer architecture: The case for JCachesim

Irina Branovic[1], Roberto Giorgi[2], and Antonio Prete[3]

[1,2] Dipartimento di Ingegneria dell' Informazione
Facoltà di Ingegneria
University of Siena, Italy
branovic@dii.unisi.it, giorgi@unisi.it

[3] Dipartimento di Ingegneria dell'Informazione
Facoltà di Ingegneria
University of Pisa, Italy
prete@iet.unipi.it

Abstract

This paper describes possible advantages of adding an interactive tool with log capabilities, in an online learning environment. We describe the interactive, Java-based tool named JCachesim, which is used for experimenting cache behavior with simple assembly programs while varying cache features. The tool has embedded features that allow the teacher to monitor the progress of each individual student.

1. Introduction

Internet offers the technology that supplements traditional classroom training with Web-based components and learning environments, where the educational process is experienced online. The objective is not to duplicate the characteristics of an ordinary class, but to use the possibilities of the computer to actually do better than what normally occurs in the face to face class.

According to projections, by 2004, 75 percent of US college students will have taken at least one online course. The number of colleges and universities offering e-learning will more than double, from 1,500 in 1999 to more than 3,300 in 2004. Student enrollment in these courses will increase 33% annually during this time [1].

Educational advantages that arise when supplementing a course with Web-based tools include:

- Enabling student-centered teaching approaches
- Providing 24/7 accessibility to course materials
- Providing just in time methods to assess and evaluate student progress
- Reducing “administrivia” around course management

There are also other, less obvious, but equally compelling advantages in favor of online teaching. Students are judged solely by their submitted work and their participation in online discussion forums, not by how they look. This "anonymity filter" has proven to have a positive effect on shyer students, who are more likely to respond in class discussions and debates when they have the time to think beforehand, and to compose answers they feel good about.

Navigating through the screens of an interesting, colorful Web site maintains students' interest and can keep their brains active. Students can see other students' work and profit from their inspiration and understanding. Using conferencing, e-mail and other Internet features, students can also comment on each other's creations and discuss variations and other possibilities.

Structured note taking, using tools such as interactive study guides, and the use of visuals and graphics as part of the syllabus and presentation outlines contribute to student understanding of the course. Student discussion records, groups and project work and commentaries can be used to add to the content of the course.

2. Virtual classroom on computer architecture

There are a number of possible solutions for building a Web-based course on computer architecture. Detailed explanation about creating a virtual classroom, as well as examples of Web teaching environments can be found in [2]. For the purpose of creating our computer architecture classroom, we used a similar environment.

The consistent interface of distance learning environments speeds up the process of learning, and does not intimidate instructors and students with the ordeal of learning to use a new software application each time a new tool is incorporated into the course.

Although our students found reading text in lectures via computer screen sometimes tedious, they liked integrated simulators, prerecorded lectures, and quizzes.

One of the most interesting enhancements that we added recently is the possibility of using interactive tools based on Java applets. These tools allow the students both to exercize and to learn. We also embedded a facility to automatically log the student's use of the tool, and create a personalized record to make sure that he used the basic functionalities of the tool. In the following, we consider such interactive tool for a lesson regarding cache memories.

3. The JCachesim tool

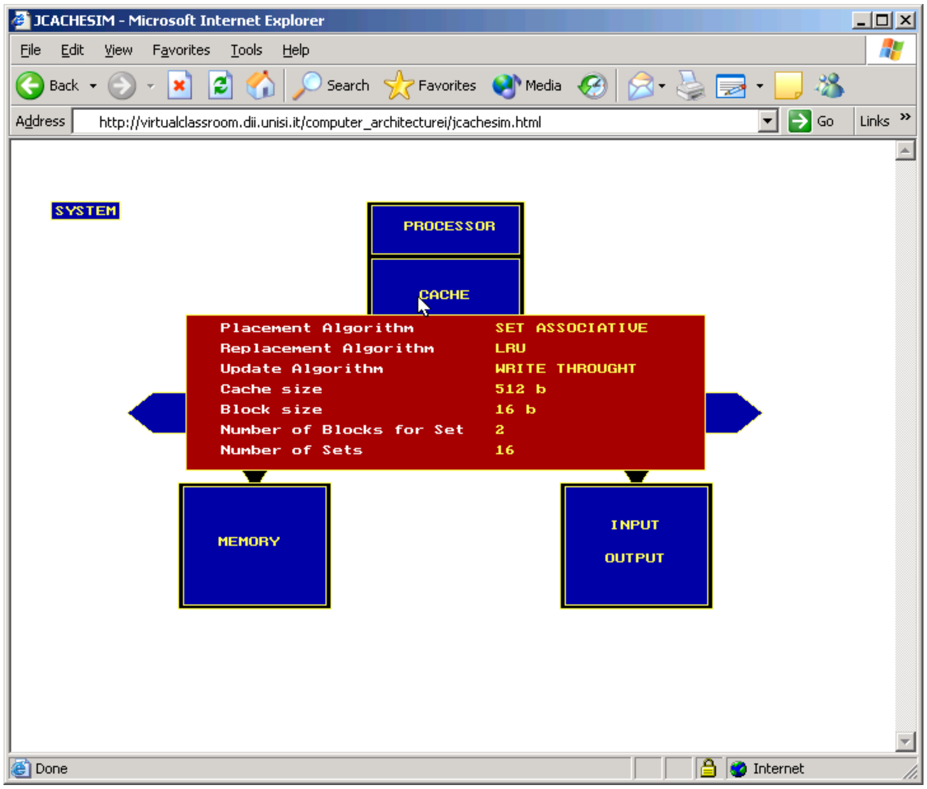
One of the lessons in our computer architecture classroom allows students to use an interesting tool for studying and analyzing a computer with cache memory, called JCachesim. It is based on previous experience on non Web-based tools [3].

JCachesim is a simulation environment of a computer with a cache memory. It allows the student to observe the CPU and the cache activities during the execution of a program, and in particular during read or write memory operation, to evaluate the system performance, to analyze the reference locality and the distribution of memory accesses due to the program execution.

An exercise is organized in three phases: configuration, simulation, and analysis.

In the first phase, students write a program in an assembly language, and then configure the system. For cache memory, the student chooses the cache capacity, the placement policy (direct, full, or set-associative mappings), the cache block size, the number of ways. Main memory size can be chosen, the main memory update policy and, finally, the block replacement policy (FIFO, random, or LRU).

For I/O devices, the student specifies the I/O type (monitor, keyboard or general purpose), the synchronization scheme (none or handshake), the interrupt scheme (none, vectored interrupt or non vectored interrupt), and the addresses of the relative device registers (Figure 1).


JCACHESIM - Microsoft Internet Explorer

Address http://virtualclassroom.dii.unisi.it/computer_architecturei/jcachesim.html

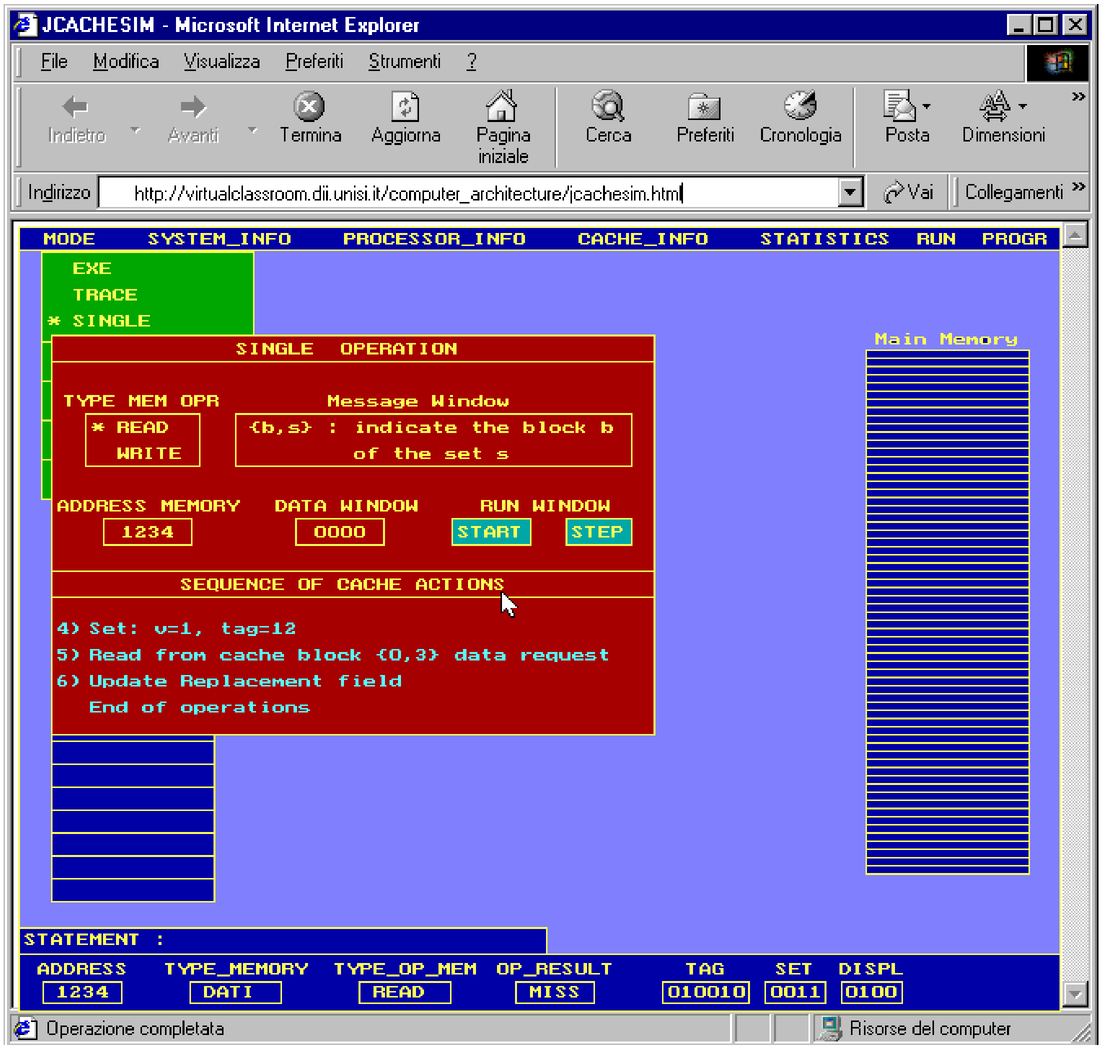
Placement Algorithm	SET ASSOCIATIVE
Replacement Algorithm	LRU
Update Algorithm	WRITE THROUGHT
Cache size	512 b
Block size	16 b
Number of Blocks for Set	2
Number of Sets	16

Figure 1: An example of configuring parameters, by clicking on the selected system component.

In the simulation phase, JCachesim can work in one of following three modes:

- Single – the student can ask for the execution of a memory operation by specifying the memory address and the operation type. In the single mode, JCachesim executes a single memory operation and shows, through an animation, the cache and main memory events, and the sequence of actions necessary to perform the required memory operation (Figure 2).
- Trace – the student can execute a program step-by-step, and examine cache or memory contents.
- Exe – the student can ask for the execution of the whole program (or a portion).

The student can watch the statistics regarding cache operations at any time (Figures 3, 4). JCachesim tool is written in the form of interactive applets that allow us to train students. However, one of the most useful features of JCachesim is its ability to create a log of student's activities. The log file contains the student name, the time he or she took the test, the chosen settings and what kind of experiments were performed. The log file is automatically stored and available to instructors (Figure 5).



http://virtualclassroom.dii.unisi.it/computer_architecture/jcachesim.html

SINGLE OPERATION

TYPE MEM OPR: * READ / WRITE

Message Window: {b,s} : indicate the block b of the set s

ADDRESS MEMORY: 1234

DATA WINDOW: 0000

RUN WINDOW

SEQUENCE OF CACHE ACTIONS

4) Set: v=1, tag=12
5) Read from cache block {0,3} data request
6) Update Replacement field
End of operations

STATEMENT :

ADDRESS	TYPE_MEMORY	TYPE_OP_MEM	OP_RESULT	TAG	SET	DISPL
1234	DATI	READ	MISS	010010	0011	0100

Figure 2: An example of JCachesim working in single mode.

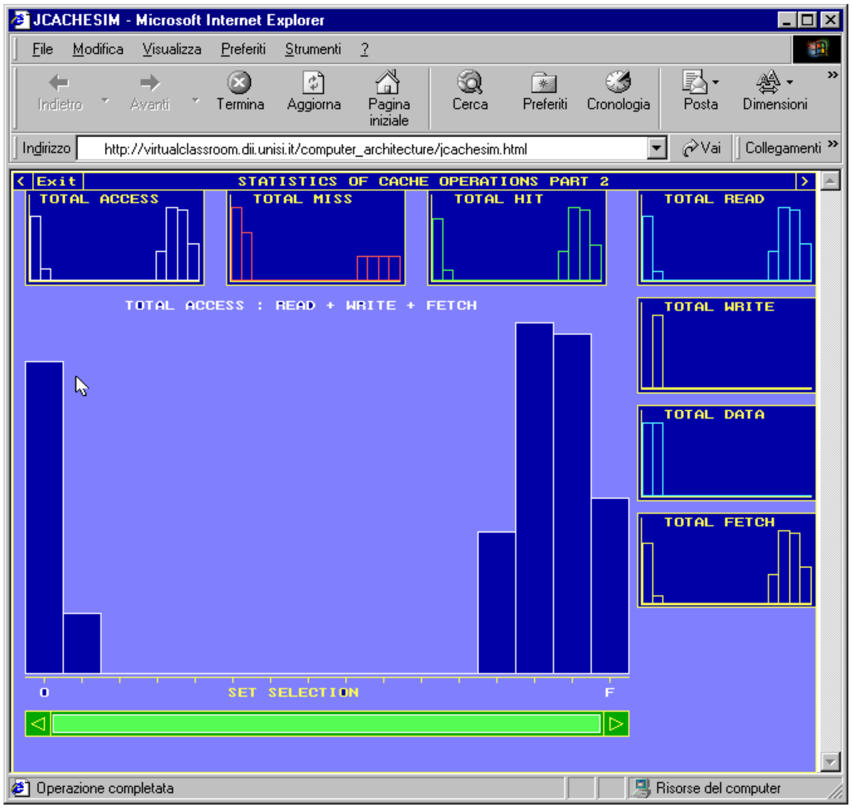
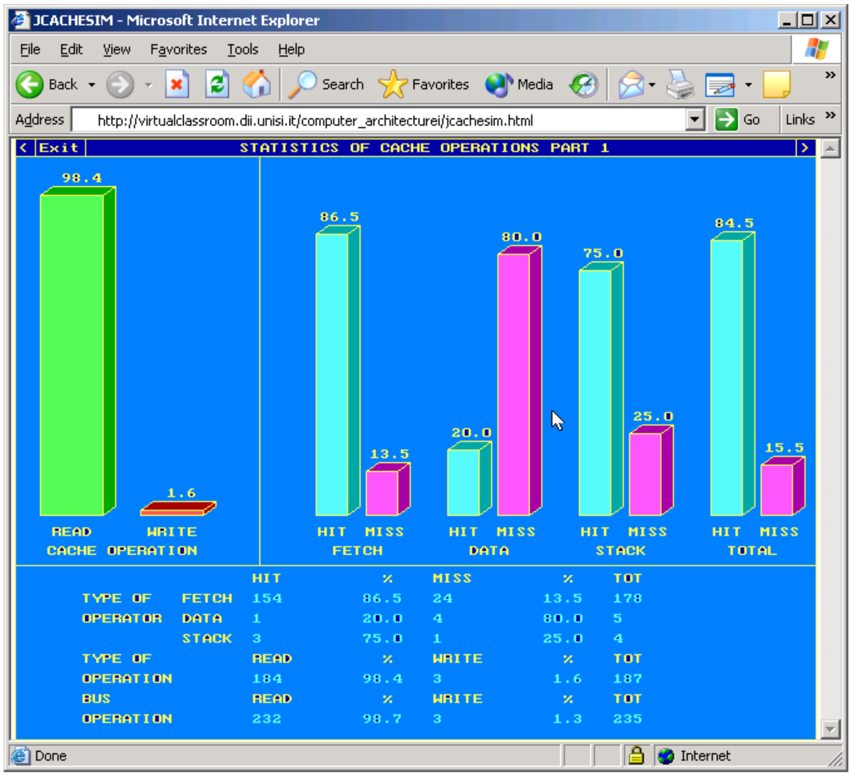

JCACHESIM - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro Avanti Termina Aggiorna Pagina iniziale Cerca Preferiti Cronologia Posta Dimensioni
Indirizzo http://virtualclassroom.dii.unisi.it/computer_architecture/jcachesim.html
Vai Collegamenti
< Exit STATISTICS OF CACHE OPERATIONS PART 2 >
TOTAL ACCESS
TOTAL MISS
TOTAL HIT
TOTAL READ
TOTAL ACCESS : READ + WRITE + FETCH
TOTAL WRITE
TOTAL DATA
TOTAL FETCH
0
SET SELECTION
F
Operazione completata
Risorse del computer

Figure 3: Pictures showing the locality of accesses in various memory areas.

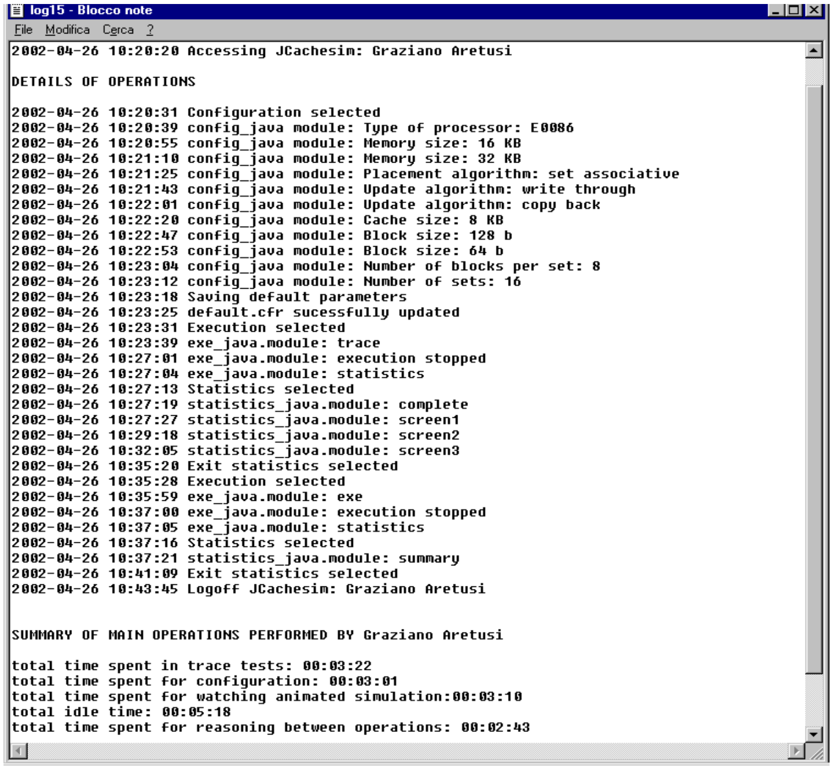

JCACHESIM - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://virtualclassroom.dii.unisi.it/computer_architecturei/jcachesim.html
Go Links
< Exit STATISTICS OF CACHE OPERATIONS PART 1 >
98.4
1.6
READ WRITE
CACHE OPERATION
86.5
13.5
HIT MISS
FETCH
20.0
80.0
HIT MISS
DATA
75.0
25.0
HIT MISS
STACK
84.5
15.5
HIT MISS
TOTAL
Done
Internet

```
                            HIT          %     MISS          %    TOT
      TYPE OF     FETCH     154       86.5     24         13.5    178
      OPERATOR    DATA      1         20.0     4          80.0    5
                  STACK     3         75.0     1          25.0    4
      TYPE OF               READ         %     WRITE         %    TOT
      OPERATION             184       98.4     3           1.6    187
      BUS                   READ         %     WRITE         %    TOT
      OPERATION             232       98.7     3           1.3    235
```

Figure 4: Global statistics of cache operations.

The generated log file also contains following information: total time spent for reasoning between operations, and total idle time. These data are used solely for tracking time the student spent using the tool, and not for measuring his performances.

The JCachesim tool is still in the prototype phase, but we hope to deliver a final version to the public as soon as possible.



```
2002-04-26 10:20:20 Accessing JCachesim: Graziano Aretusi

DETAILS OF OPERATIONS

2002-04-26 10:20:31 Configuration selected
2002-04-26 10:20:39 config_java module: Type of processor: E0086
2002-04-26 10:20:55 config_java module: Memory size: 16 KB
2002-04-26 10:21:10 config_java module: Memory size: 32 KB
2002-04-26 10:21:25 config_java module: Placement algorithm: set associative
2002-04-26 10:21:43 config_java module: Update algorithm: write through
2002-04-26 10:22:01 config_java module: Update algorithm: copy back
2002-04-26 10:22:20 config_java module: Cache size: 8 KB
2002-04-26 10:22:47 config_java module: Block size: 128 b
2002-04-26 10:22:53 config_java module: Block size: 64 b
2002-04-26 10:23:04 config_java module: Number of blocks per set: 8
2002-04-26 10:23:12 config_java module: Number of sets: 16
2002-04-26 10:23:18 Saving default parameters
2002-04-26 10:23:25 default.cfr sucessfully updated
2002-04-26 10:23:31 Execution selected
2002-04-26 10:23:39 exe_java.module: trace
2002-04-26 10:27:01 exe_java.module: execution stopped
2002-04-26 10:27:04 exe_java.module: statistics
2002-04-26 10:27:13 Statistics selected
2002-04-26 10:27:19 statistics_java.module: complete
2002-04-26 10:27:27 statistics_java.module: screen1
2002-04-26 10:29:18 statistics_java.module: screen2
2002-04-26 10:32:05 statistics_java.module: screen3
2002-04-26 10:35:20 Exit statistics selected
2002-04-26 10:35:28 Execution selected
2002-04-26 10:35:59 exe_java.module: exe
2002-04-26 10:37:00 exe_java.module: execution stopped
2002-04-26 10:37:05 exe_java.module: statistics
2002-04-26 10:37:16 Statistics selected
2002-04-26 10:37:21 statistics_java.module: summary
2002-04-26 10:41:09 Exit statistics selected
2002-04-26 10:43:45 Logoff JCachesim: Graziano Aretusi


SUMMARY OF MAIN OPERATIONS PERFORMED BY Graziano Aretusi

total time spent in trace tests: 00:03:22
total time spent for configuration: 00:03:01
total time spent for watching animated simulation:00:03:10
total idle time: 00:05:18
total time spent for reasoning between operations: 00:02:43
```

Figure 5: An example of student log file, to be considered by the teacher.

4. Conclusions

Computer architecture requires understanding a wide variety of issues and interactions among them. One important step is that the student makes use of simulation tools to understand concepts otherwise difficult to experience.

Our internal research indicates that teaching and studying at a distance is equally effective, even better than traditional instruction, provided there is timely teacher-to-student feedback.

We have described the possible advantages of integrating an interactive tool with log capabilities into a virtual classroom environment. Using an interactive tool like JCachesim allows students to indicate the settings of a cache memory, to observe the cache activity needed for a memory operation, to evaluate the system performance by varying the parameters, and to analyze the program behavior by the memory references. One of the most important features of this tool is the ability to generate log files, which can be used to monitor students' progresses and track their activities. Future plans for improving the JCachesim include providing more Java modules for enabling immediate interaction between students and instructors.

5. References

[1] International Data Corporation: *Distance Learning in Higher Education: Market Forecast and Analysis, 1999-2004*.

[2] Branovic, I., Milutinovic, V., Tutorial on Advances in Internet-based Education, (http://galeb.etf.bg.ac.yu/~vm/tutorials), School of Electrical Engineering, University of Belgrade, Serbia, Yugoslavia, 2001.

[3] Prete, A., "Cachesim: A graphical software environment to support the teaching of computer system with cache memories", Proceedings of 7-th SEI Conference on Software Engineering Education, Springer-Verlag, January 1994.