

A Computer Performance Course for an IS Program

John Maniotes
maniotes@calumet.purdue.edu
Information Systems & Computer Programming Department (ISCP)
Purdue University Calumet
Hammond, Indiana 46323, USA

Charles R. Winer
winer@calumet.purdue.edu
Information Systems & Computer Programming Department (ISCP)
Purdue University Calumet
Hammond, Indiana 46323, USA and

Sam A. Maniotes
samaniotes@aol.com
Information Systems & Computer Programming Department (ISCP)
Purdue University Calumet
Hammond, Indiana 46323, USA

Abstract

This paper presents ideas and actual experiences accumulated over the past 15 years on conducting a senior level course for Computer Information Systems (CIS) majors on benchmarking the performance of computer systems, applications software, and systems software. This paper also is intended to serve as a guide for those faculty who are interested in conducting such a course.

Keywords: Algorithms, benchmarking, computer performance, documentation

1. PURPOSE OF THE COURSE

This is a senior level course, which serves to integrate the knowledge and abilities joined through other computer related courses with a comprehensive performance (benchmark) project. Currently, this course is titled CIS 483 Computer Hardware/Software Selection (4 credit hours). It is a course that not only teaches the fundamentals of computer hardware and software selection, but it also has a computer performance project, called the Benchmark Project, that all students in the course do in teams (usually three students to a team).

Included are some of the topics related to the computer performance part of the course.

- (1) Surveys of the computer and IS market places
- (2) Organizational and administrative considerations in the selection process
- (3) Validation of the proposed system's characteristics and performance
- (4) Performance measurement approaches
- (5) Conversion and compatibility issues
- (6) Benchmark project

The authors will concentrate on topics 3 to 6 for this paper. Topics 1 and 2 have been described by (Maniotes 1999).

2. TOPICS

Anyone who engages in the art of selection must know the current market place, the technology trends occurring in the IS field, and organizational and administrative considerations in the selection process (Topics 1 and 2).

Early in the semester, definitions of the computer and IS markets (vendors) are discussed. We have found that the annual market surveys published in Wall Street Journal, Computerworld, Datamation, Upside, Red Herring, and Software magazines are very helpful and provide useful statistics on the composition of the market place as well as the market leaders.

Topics 3 and 4, validation of the proposed system and performance approaches range from techniques used to validate system characteristics and system times (performance). For the latter, we stress the importance of live, standardized, and artificial benchmarks, simulation packages, and hardware and software monitors that can be used to predict and validate throughputs and response times. The various types of benchmarks are important, since students will have to choose one or more benchmark algorithms and measure the performance of a computer system.

Topic 5, conversion and compatibility issues, we discuss the roles of emulators, simulators, sifters, and translators and identify vendors who specialize in conversion services. We also describe conversion techniques that have been used successfully such as parallel, piece meal, and radical (plunge) conversion methods. The reason we stress Topic 5 is that students must take their benchmark program and convert it to run on multiple systems with different operating systems and compilers.

3. THE BENCHMARK PROJECT

All students in the course participate on a computer performance project, commonly known as the Benchmark Project (Topic 6). Students are allowed to select their own team members, usually three per team. Each team then chooses their Project Leader.

Each team chooses a different benchmark and is expected to research, analyze, design, program, test, debug, implement, and document a benchmark program running on a variety of computer systems.

This course, CIS 483 Computer Hardware/Software Selection, simulates many of the conditions of a real-world job because the team is required to use project management tools, submit plans, perform formal presentations, and work together as a team with a defined project leader. In essence, CIS 483 also functions as a Senior Capstone course.

A benchmark is a set of computer programs which are designed to measure a multitude of computer characteristics. For example, a benchmark can be used to

evaluate the performance of a computer's CPU, I/O units, main and auxiliary storage units, applications software, and systems software such as the operating system, utilities, compilers, interpreters, etc. The data accumulated from the benchmarks can be used for comparing the performance of different systems with the same workload as well as tuning the performance of an existing system.

Over the past few years, the following general types of computer systems have been benchmarked:

- Personal computers
- Work Stations
- Midrange systems
- Minicomputers
- Mainframe systems
- Compute bound systems
- I/O bound systems
- RISC vs CISC based systems

Benchmark programs can be grouped into various classes such as:

- Compute-bound programs
- I/O-bound programs
- Source programs in different languages to be compiled
- Object programs to be optimized
- Business programs
- Scientific programs
- Engineering programs
- Graphics programs
- Mathematical programs

The last five items above are all applications dependent.

The Benchmark Project has been processed on a variety of specific systems at Purdue University Calumet, such as the: COMPAQ/DEC ALPHA 2100 and VAX 11/780 and 8600 minicomputers, IBM AS/400 midrange system, IBM 4341 mainframe, Sun and H/P workstations, and Intel x86 and Apple MAC PC's.

Programming languages used are: C, C++, BASIC, Visual BASIC, FORTRAN, PL/I, and COBOL.

Operating systems used are: VMS, UNIX, MVS, VM, OS/400, MS DOS, MS Windows, and MAC OS.

Also over the past few years, the following benchmark projects have been implemented using algorithms dealing with:

- Statistical computations (mean, variance, and standard deviation)
- Matrix operations (addition, subtraction, multiplication, and inversion)

- Solutions to simultaneous linear equations
- Square root approximations
- Evaluations of complex formulae and equations
- Polynomial evaluations
- Interpolation
- Random number generators
- Updating files (sequential and direct access files)
- Sequence checking
- Table look up
- Sorting files (ascending or descending order)
- Collating files
- Graphics (generation of fractals)

Sometimes students are allowed to choose any public domain compute-bound algorithm, subject to instructor approval, to measure the speed of various CPU's. Some of those chosen have been the Ziff-Davis (ZD), Dhystone, Linpack, Whetstone, SPEC, and Towers of Hanoi benchmarks (Grace 1996 and Herman 1995).

When students prepare benchmarks to run on a variety of computer systems, they are faced with problems they have not encountered in prior courses. For example:

- How do you select a benchmark that is representative of a client's workload?
- How do you evaluate the compiler or operating system if it is new (i.e., version 1) versus those systems that are mature and have been around for years.
- How do you process a benchmark under different operating environments such as batch, on-line, or on the internet?
- What do you do if a hardware feature (such as floating point hardware) is present on one system but missing on another system?

- What are the true costs and efforts to prepare and process benchmarks?

The outputs from the execution of a successful benchmark are the CPU time spent executing the assigned algorithm and the amount of main and disk storage utilized by the benchmark program. Also measured is the effect that the precision associated with the system, i.e. integer, single, double, and quad precision, has on the CPU time. Furthermore, whether the language optimizer is enabled or disabled, affects the CPU time. These results are summarized by students in the form of graphs and tables.

Some of the interesting student's benchmarks have dealt with sorting records in large files. In the past, students have benchmarked the following sort algorithms: Bubble sort, exchange sort, insertion sort, heap sort, merge sort, Shell sort, quick, quicker, and quickest sort.

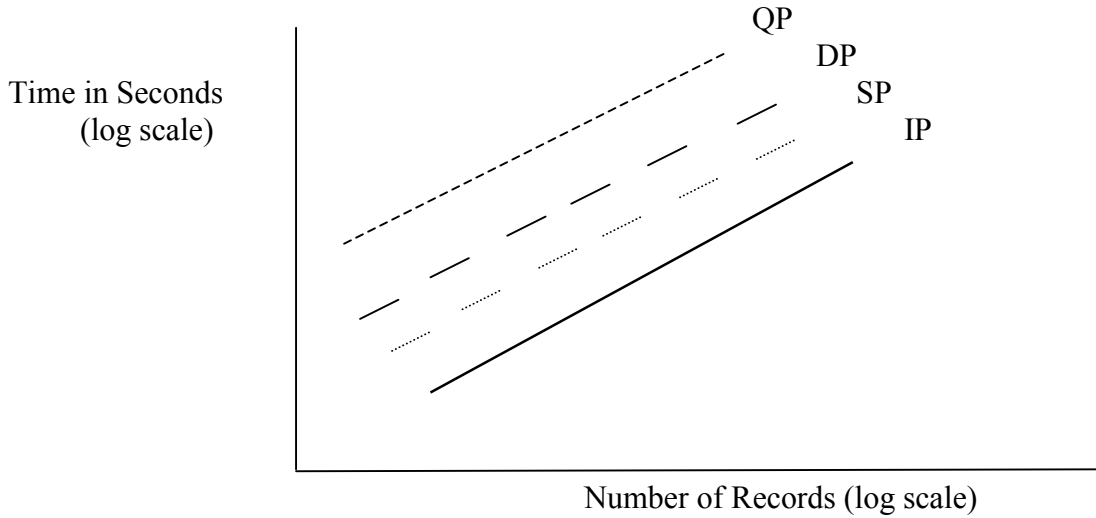
Students have also compared the performance of these sort algorithms against the sort utility that comes with the computer system. For example, the benchmarks results dealing with the algorithm for the Shell Sort are plotted on a 2-D log-log scale graph as shown in Figure 1.

Another useful table that students develop is the Average Relative Power (ARP), which is defined as:

$$\text{ARP} = \frac{\text{Processing time for system } x}{\text{Processing time for system } y}$$

A sketch of the table is shown in Table 1.

Figure 1 -- Log-Log Graph
 Algorithm: Shell Sort
 Computer System: COMPAQ/DEC ALPHA 2100
 Language: C
 Optimizer: Enabled



where IP = Integer Precision; SP = Single Precision; DP = Double Precision; and QP = Quad Precision

Table 1 – Average Relative Power

Is X (table value) Times Faster Than This System

| This System | System A* | System B* | System C* | Other* |
|-------------|-----------|-----------|-----------|--------|
| System A* | 1 | | | |
| System B* | | 1 | | |
| System C* | | | 1 | |
| Other* | | | | 1 |

*For a given programming language with the optimizer enabled. A similar table is prepared with the optimizer disabled.

4. PRESENTATIONS

During the last week of the semester, a one hour appointment is scheduled with each team so they can present their work accomplished during the semester. The presentations are open to faculty of our department,

school, and the university. Students from other teams are not allowed to participate in each others team's final presentations.

The instructor and faculty ask the questions. Students are expected to "defend" their benchmark designs and

demonstrate that they work according to the specifications.

The presentations take place in a large electronic classroom equipped with PC's. The benchmark images and results appear on a large 6' by 6' screen for viewing.

Prior to the final presentations, two other presentations are made during the semester by each team. For example, during the fifth week, a short presentation takes place regarding the initial benchmark requirements. A 30 minute presentation is also made during the tenth week. This in-depth presentation covers the specifics of the benchmark.

During the early presentations, serious design flaws and errors can easily be detected by the instructor and later corrected by the team. Also during the presentations, the students discuss their progress and problems.

During each presentation, each team is asked to record any problems, errors or flaws that are discovered during the presentation. These anomalies are later written up, and the next day a copy is given to the instructor. A copy of the anomalies is also placed in the team's project notebook.

5. THE PROJECT NOTEBOOK

All teams are required to keep all their pertinent benchmark activities in a project notebook. Furthermore, they are required to submit their project notebook each time they give a presentation to the instructor. After each presentation, the project notebook is graded and returned to the team with appropriate comments. The teams are also expected to prepare a professional formal report and place it in their project notebook

The documentation in the formal report contains:

- (1) A cover page consisting of suitable information identifying the team and its members, course, institution and project.
- (2) A table of contents
- (3) An abstract of the benchmark project
- (4) A Gantt chart of the project
- (5) A list of assumptions made pertaining to the benchmark
- (6) An introduction, including purpose, scope, etc
- (7) A description of the computer system, compiler, interpreter, operating system, and other software used
- (8) A copy of any algorithms used
- (9) A general hierarchy chart of the project
- (10) Instructions for operating the computer(s) and executing the benchmark program(s)
- (11) A list of any error/warning conditions (both system and program generated) concerning the project

- (12) A description of any problems encountered such as portability, inconsistencies, hardware/software/people type problems, etc.
- (13) A description of any innovative/unique solutions
- (14) Recommendations and conclusions regarding the benchmark project
- (15) Bibliography of references

Students also place in their project notebook all directory listings, program listings, printer outputs, menus and screens of the displays, and the diskette or CD containing all their benchmark programs.

6. FEEDBACK FROM CIS ALUMNI

The seniors who are enrolled in this course are systems/networking, application programming, or systems analysis and design majors. They invariably ask, during the course, if they will ever use any of the topics, especially topics 3 to 6. That is, performance measurements, conversions, and benchmarks. Our reply is always YES! However we caution the students that it will depend on where they work and in what capacity. Some of our older/more mature students, who have been working in the Information Systems field for many years have said, "I wish I had these topics earlier in my career."

Many of our CIS alumni who have entered the ranks of IS management, have told us how valuable the topics and experience gained in this course have been to their careers. Many of these alumni are engaged in the art of computer hardware, software, services selection and performance measurement. Others have become system administrators responsible for the performance of their computer systems and networks. Experience in benchmarking has been also a good starting point for those responsible for maintaining or upgrading their system's performance.

7. FUTURE DIRECTIONS

Currently, the ISCP department does not offer a follow-up course on computer performance and benchmarking. Many of our students and alumni have requested additional courses dealing with computer performance issues such as simulation and monitoring.

Our department had recently formed a committee to study the aforementioned requests. We are exploring the possibility of creating a new course incorporating computer simulation and monitoring tools, and to include advanced benchmarking techniques.

Furthermore, we are exploring the use of additional resources to support this new course. The issue of using dedicated high-powered servers, workstations, and PCs in a closed lab environment will be examined.

The issue of using monitoring tools such as CA-Unicenter TNG, H/P Openview, Sun Symon, etc will be researched. The use of network simulation tools from CACI Products, Image Net, Systems & Networks Inc., etc will be addressed. Finally, the issue of training the existing IS staff will be considered.

8. SOME CONCLUDING REMARKS

The following are recommendations to those faculty members thinking about teaching a course like CIS 483. Be prepared to spend extra time with your students. It's important that close supervision be maintained over the teams and the Benchmark Project.

Also, be prepared to update the papers and articles on reserve in the library, as well as your lecture notes. Develop a Web site to keep and post current information for your students. Many new developments are occurring daily in our Information Systems field. This is a dynamic field where new computer hardware, software, services and performance measurements appear daily

We endorse the concepts covered in this course for any IS curricula. This should be a senior level, capstone course taken by IS majors.

9. REFERENCES

- Grace, Richard, 1996, The Benchmark Book, Prentice Hall, New York.
- Herman, Gary, 1995, Benchmarking the Benchmarks, Datapro Information Services, Delran, NJ.
- Maniotes, John and Charles R. Winer, 1999, "Integration of Field Experiences in Computer Hardware/Software Selection: A Senior Capstone Course," Proceedings of the 5th World Multiconference on Systemics, Cybernetics, and Informatics (SCI), 1999, July 31-August 4, Vol. 8, pp 464-470.