Lab 3

Performance Measurement

Written By:

Tyler Holland

1/31/2010

**Purpose:**

In this lab, I compared the performance of a program run on three different machines. By comparing the time it takes to complete the program on each machine, I was able to see how each machine compared to the others based on CPU performance. I also used varying levels of compiler optimization to see how that effected the program's speed.

**Procedure:**

I ran the provided lab3 script, which ran bzip2 on a sample input file and then timed it using the Unix 'time' command. It also repeated that test with O0, O1, O2, O3 compiler optimizations, along with O2 and O3 with the gprof profiling tool.

**Results:**

**Figure 1** displays the run time (in seconds) of bzip2 with varying levels of optimization and profiling on each of the three different machines. Figure 2 shows the bar graph of this data.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tests** | **O0** | **O1** | **O2** | **O3** | **O2 with profiling** | **O3 with profiling** |
| **Tytanium** | 22.058 | 9.575 | 11.829 | 11.763 | 8.379 | 8.361 |
| **Falcon** | 29.992 | 10.722 | 14.477 | 14.530 | 7.602 | 7.824 |
| **Multicore** | 23.541 | 16.574 | 15.030 | 14.948 | 12.695 | 12.416 |

**Figure 1**: Run times (in seconds) for each machine and optimization/profile

**Figure 2**: Graph of run times (in seconds) for each optimization/profile

**Figure 3** shows a table of the gprof data that was generated for bzip2 with the O2 and O3 optimizations on each machine. The functions listed there are the most frequently executed functions in the program, while the percentages show the amount of CPU time spent per function.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tytanium** | **O2 with profiling** | |  | **O3 with profiling** | |
| BZ2\_blockSort | 90.73% | BZ2\_blockSort | 90.68% |
| BZ2\_compressBlock | 8.22% | BZ2\_compressBlock | 7.99% |
| handle\_compress | 0.75% | handle\_compress | 1.18% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Falcon** | **O2 with profiling** | |  | **O3 with profiling** | |
| BZ2\_blockSort | 86.39% | BZ2\_blockSort | 87.08% |
| BZ2\_compressBlock | 10.10 % | BZ2\_compressBlock | 9.20% |
| handle\_compress | 3.37% | handle\_compress | 3.59% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Multicore** | **O2 with profiling** | |  | **O3 with profiling** | |
| mainSort | 88.32 % | mainSort | 87.53% |
| BZ2\_compressBlock | 8.43% | BZ2\_compressBlock | 8.60% |
| handle\_compress | 2.19% | handle\_compress | 3.16% |

**Figure 3**: gprof data for O2 and O3 optimization profiles with each machine

**Figure 4** shows SPEC CINT2006 data for each of the machines that I ran bzip2 on. SPEC used the O3 compiler optimization with profiling to acquire their data.

**Machines:**

**Tytanium:** is my custom-built home computer. It has a single quad-core 64 bit Intel Core i7-920 processor (4 cores, 2 threads/core) with overclocked from 2.66ghz to 3.4ghz, 4 x 256 kb L2 Cache, 8 Mb L3 Cache, and 12 Gb of RAM.

**Falcon:** is a Sun x4200 M2 with 2 sockets of dual core 64 bit Opteron "Santa Rosa" 2216 CPUs (4 cores total).

**Multicore:** is a Dell Poweredge 6600 with 4 sockets of dual core 32 bit Xeon MP Gallatin CPUs (8 cores total).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Average Base Time | Average Peak Time | Base Time For bzip2 | Peak Time For bzip2 |
| **\***Asus P6T Deluxe (Intel Core i7-920)  ("**Tytanium**") | 25.7 | 28.6 | 15.4 | 15.6 |
| Sun Fire X4200  ("**Falcon**") | 11.6 | 13.3 | 9.65 | 10.3 |
| **\***Dell Poweredge 6850  ("**Multicore**") | 10.3 | 11.0 | 8.69 | 8.93 |

**Figure 4:** SPEC CINT2006 data for each machine (http://www.spec.org/cpu2006/results/cint2006.html)

**\*Note:** I was unable to find the exact Poweredge model that Multicore is, so I used the closest availiable (the Poweredge 6850 instead of the 6600). Also, because Tytanium is custom built, I used the closest computer I could find which is the Asus P6T Deluxe (Intel Core i7-920).

**Figure 5** shows a table of the computers, their MSRP, and performance/cost (measured with O3 optimization with profiling seconds / MSRP), and the lower number the better.

|  |  |  |
| --- | --- | --- |
| **Computer** | **MSRP** | **Performance / Cost** |
| **Tytanium** | $1,900 | .0044005 |
| **Falcon** | $4,255 | .0018387 |
| **Multicore** | $1,247 | .0099566 |

**Figure 5**: MSRP and Performance/Cost of the machines (lower number = better value)

**Lab Questions:**

**1) Why do we do three timing runs on each optimization?**

By doing 3 runs on each optimization, we can reduce the amount of error. If the CPU was under a large amount of load for one run, it would have inaccurate results, but the chance of that happening 3 times is a lot lower, so the results are more accurate. Just like anything with variables, the more times you recreate the experiment the more accurate the results will be.

**2) Briefly describe why you need to compile the profiled programs twice.**

During the first compilation, the profiling program sees what operations are taking the most amount of CPU time. During the second compilation, it gives the information to the compiler somehow and the compiler optimizes the program instructions based on the profile data, so it can reduce the time of the longest operation to provide the biggest time benefit.

**3) Why do the profiled programs require extra compiler flags?**

In order for the compiler to know to use the profiles, extra flags needed to be added on to the compilation command.

**4)** **What do these flags do?**

They tell the compiler what profile file to use, and what type of optimization it is.

**Conclusions:**

The effects of optimization are very apparent after completing this lab. Before optimization, Tytanium was far faster than Falcon in running bzip2. However, after O3 optimization and profiling, Falcon ended up being faster than Tytanium. It is interesting to see that Tytanium actually benefited more from O1 optimization than O2 and O3. I attribute this to how new the Core i7 processor is. I am guessing that these optimization techniques haven't been perfected for the new processors, which would create differences in timing like the ones shown through bzip2. Also, Multicore benefited the least from all of the optimizations, and went from being second fastest with no optimization to slowest with O3 optimization with profiling. Both Tytanium and Falcon benefited the same from each level of optimization, but when profiling was added Falcon had a little bit larger decrease in time than Tytanium. As I said before, this is most likely because of the new-ness of the i7 processor.

Some things that might have effected performance on Multicore and Falcon is the amount of users on the server at the same time. With different amounts of users running different amounts of programs, the CPU can have varying workloads, and the more work the CPU has to do at once, the slower the benchmarking program will run. For Tytanium, I am the sole user of the computer, so I was able to shut down all of my other processes and run just the benchmarking program. This allowed me to obtain the best results possible multiple times in a row.

This experiment is mainly about showing how it is possible to compare computer performance between multiple computers and be able to decide which computer to buy based on the performance. By running a single program, we know the instruction count should be the same or nearly the same on all of the computers, only varying by how the compiler handles things. This creates a very even playing field for the processors. By running different types of programs, we can see which processor performs best doing that type of computation. For example, if all I wanted to do is run bzip2 type programs, and I wanted to get the most performance out of my dollar, I would look at the results and conclude that I should buy a Falcon type computer. If all I want to do is get the most value for my dollar, Falcon would be the way to go, because it has a .0018 seconds per dollar ratio. Although it is more expensive, the performance benefit exceeds the extra cost. However, if I had a $2,000 spending limit, I could also see that a Tytanium type computer would be the way to go in that case. Although I could almost buy 2 Multicore type computers for that amount of money, one Tytanium more than doubles the performance/cost ratio (.00995 to .00440).

If I was looking at buying a computer for a specific application, I now know how to compare the different types of computers based solely on the tasks I know I will be running on that machine. If I don't have a price limit, I can see what machine performs best. If I do have a price limit, I can see which computer is best based on that price point. This lab definitely helped me learn how to make computer purchasing decisions based on performance.