Supercomputer Parallel Architectures: A Fifth-Year Study of Supercomputer Efficiency

Christopher Sewell

The purpose of this project, my fifth year of study of supercomputer efficiency, is to explore the relative merits of shared-memory and distributed-memory supercomputers; the effect of such factors as length of loops, total running length, vectorization capabilities, nature of commands within a parallelizable loop, and application purpose on the parallel processing abilities and overall speed of these two supercomputer types; and the maximum practically achievable speeds on both machines and its relationship to claimed top speeds. My null hypothesis was that neither the different natures of the tested programs nor the type of Cray supercomputer (shared-memory J-90 or distributed-memory T3E) used to run these programs would have any effect upon the parallel processing potentials of the programs nor on the total CPU or wall-clock timings on the two computers.

In order to test my null hypothesis, I used eight different types of original programs to run on the J-90 and the T3E. Program A was a simple two do-loop procedure designed to maximize computer performance. Program B was similar to Program A except that it included a print statement within the parallelizable do-loop, testing the computers' abilities to use parallel processing with input/output commands. Next, Program C was written with a long mathematical do-loop which solved linear programming problems to test the computers' recognition of long parallelizable loops. Program D was similar to Program C except that it solved triangle problems and had a much shorter running time as its inner loop repeated 100 times rather than 100,000 times, thus seeking to determine whether any differences discovered by previous test programs hold for shorter running times. Each of these parallel-processing capable programs were run ten times on both computers using one processor and again with four processors. Special instructions required to tell the T3E how to divide the programs were inserted for runs on the T3E with four processors. The last four programs, which were run ten times each on both computers using one processor, were not parallel-processing capable and were used only to continue making comparisons between the speeds of the two computers for a variety of application programs. Programs E, a Hewlett-Packard calculator simulator, and F, an inventory-tracking program, were unique in that they both outputted thousands of continuously updated results to a file. Programs G, a baseball game, and H, a basketball game, had relatively very short running times as each played one simulated game and outputted a box score. Program A was also run on the J-90 with sixteen processors and the T3E with thirty-two processors to determine the maximum practically attainable speeds.

The results from this experimentation led me to reject my null hypothesis and make several conclusions. Both computers were able to use parallel processing on a simple program with a short calculation loop. However, the T3E had two advantages over the J-90. It was capable of parallel processing even when input/output commands were involved because its relative independent processors each had the ability to execute such commands and all had access to the same memory information. The T3E was also able to use parallel processing on a simple but long calculation loop because it requires the user to instruct it how to divide the program. The J-90 was simpler to use because it automatically searches for loops on which to use parallel processing, but it cannot always recognize all loops on which parallel processing can be performed. CPU timings, on which user fees are based, were affected very little by parallel processing. The maximum speed achieved by this experimentation was 1610 megaflops on the T3E with the customized program running on thirty-two processors (6.7% of manufacturer's claimed top speed), followed by the J-90's sixteen-processor maximum, 740 megaflops (23% of claimed top speed). This speed was dramatically reduced by using fewer processors, as is usually done because not all processors are readily available to a single user at a given time, to a little over 200 megaflops. Without parallel processing, this figure dropped to about 80 megaflops. Using a program with a loop too long to be automatically parallelized or vectorized dropped the speed to about 120 megaflops on the T3E and under four megaflops on the J-90. Programs on which parallel processing and vectorization are impossible while only a few thousand calculations are performed or much time is spent outputting results decreased speeds to well under one megaflop on both computers, although the simpler overhead processes of the J-90 were somewhat better for the short run-time programs, while the more efficiently-outputting processors of the T3E performed better on those programs which displayed thousands of results. In comparing the two computers, it was found that it is usually less expensive to run vectorizable programs on the J-90 and scalar programs on the T3E. Results can be obtained fastest by running scalar or special-case parallel programs on the T3E and vectorizable non-parallel programs on the J-90. With four processors, wall-clock timings for programs on which parallel processing and vectorization are both easily performed are approximately the same on both computers. However, the T3E allows usage of up to thirty-six processors at a time while the J-90's limit is sixteen. Thus, significantly faster wall-clock speeds for the most efficient programs are attainable on the T3E. Other factors, such as the greater ease of use of the J-90 and the interference from other users on both computers were also considered in the comparison.

The J-90 and the T3E are both powerful but specialized tools. Often the fastest way to run programs is not the most economical or the easiest way. The relative demands of speed, ease of use, and money must be carefully weighed. Some programs benefit most from running on the J-90, some run best on the T3E, and some are not well-suited for supercomputers at all. Although this project provides some general outlines for which types of programs fall into each of these categories, each program (and each computer) is different, and the efficient programmer must be ever more savvy to make the best use of the the new tools of supercomputing. In a more general sense, this project can be beneficial to society at large because it serves as an example of the importance of balancing the demands of speed, efficiency, ease of use, outside interference, and cost in solving any problem.