CS 594 - Applications of Parallel Computing
Assignment 4
Due February 26th, 2003

Part1:
Implement, in Fortran or C, the six different ways to perform matrix multiplication by interchanging the loops.

C = C + A*B

(Use 64-bit arithmetic.) Make each impleme ntation a subroutine, like:

    subroutine ijk (c, m, n, ldc, a, k, lda, b, ldb)
    subroutine ikj (c, m, n, ldc, a, k, lda, b, ldb)
    …

Construct a driver program to generate random matrices and calls each matrix multiply routine with square matrices of o rders 50, 100, 150, 200, …, 500, timing the calls and computing the Mflop/s rate. Make sure you verify the correctness of your results. You can use the ATLAS routine's (see below) results to verify your routines' results. You should compute something like:

$$\frac{\| C_{ijk} - C_{atlas} \|}{\| C_{atlas} \| * macheps}$$

Run your program on a processor of one of the TORC and MSC machines.

torc0.cs.utk.edu – torc8.cs.utk.edu	Intel Pentium II I 550 MHz
msc01.cs.utk.edu – msc08.cs.utk.edu	Intel Pentium III 933 MHz

Use the highest level of optimization the compiler allows and experiment with this.
For measuring the time, number of operations, and rate of execution use PAPI.
See http://www.cs.utk.edu/~terpstra/using_papi/
Include in your timing routine a call to the BLAS matrix multiply routine DGEMM from ATLAS.

Download and build ATLAS for this part.

    call dgemm('No', 'No', n, n, n, 1.0d0, a, lda, b,
                    ldb,1.0d0, c, ldc)

For ATLAS see http://www.netlib.org/atlas/.
For PAPI see: http://icl.cs.utk.edu/papi/

Write-up a description of the timing and describe why the routines perform as they do.

Part 2:
The goal is to optimize matrix multiplication on these machines. Use whatever optimization techniques you can to improve the performance.