Abstract

Chain multiplication of matrices is widely used for scientific computing. It becomes more challenging when there is a large number of floating point dense matrices. Because, floating point operations take more time than integer operations. It would be interesting to lower the time of such chain operations. Now-a-days every multicore processor system has built in parallel computational power. This power can only be utilized when compatible parallel algorithms were used. So, in this work, a shared memory based parallel algorithms has been proposed to compute the multiplication of a long sequence of dense matrices. The algorithms have been tested with long sequence of matrices as input. The approach has been with $2\times10^8$ flops. The input matrix sequence length was typically varied from 2 to 30. Maximum number of processors used was eight (Eight core processor). Different parameters like speedup, efficiency etc. were also noted. It was concluded that the parallel algorithms could achieve approximately 90% efficiency at best case. The algorithms also showed improved scalability.
Chain Multiplication of Dense Matrices: Proposing a Shared Memory based Parallel Algorithm


- Czumaj, A; Parallel Algorithm for Matrix Chain Product and the Optimal Triangulation Problems (Extended abstract). STACS'93 version. Supported in part by the EC Cooperative Action IC 1000 Algorithms for Future Technologies; and by the grant KBN 2-1190-91-01, Pages 1-12.
- Czumaj, A; Parallel Algorithm for Matrix Chain Product and the Optimal Triangulation Problems (Extended abstract). STACS'93 version. Supported in part by the EC Cooperative Action IC 1000 Algorithms for Future Technologies; and by the grant KBN 2-1190-91-01, Pages 1-12.
- Patterson, J. L. Hennessy; Computer Organization and Design: The
Chain Multiplication of Dense Matrices: Proposing a Shared Memory based Parallel Algorithm


Index Terms

Computer Science

Algorithms

Keywords

Chain multiplication computing dense matrix multicore shared memory flops efficiency
speedup
scalability