COMPILER-DRIVEN RUNTIME I/O MINIMIZATION

Wendy Zhang

Southern University at New Orleans
Department of Computer Information Systems
New Orleans, USA, LA 70126
E-mail: wzhang@suno.edu

and

Ernst L. Leiss
University of Houston
Computer Science Department
Houston, USA, TX 77204
E-mail: coscel@cs.uh.edu

ABSTRACT

Most scientific programs have large input and output data sets that require out-of-core programming or use virtual memory management (VMM). VMM is not an effective approach because it results easily in substantial performance reduction. In contrast, compiler driven I/O management will allow a program's data sets to be retrieved in parts, called blocks or tiles. Out-of-core programming defines the partitions of the data sets and the control of I/O. Comanche (COmpiler Managed CaCHE) is a compiler combined with a user level runtime system that can replace and out-perform standard VMM for out-of-core programs. In addition, this system does not require any special services from the operating system and does not require modification of the operating system kernel. Our research aims at improving the applicability of Comanche to achieve high performance of I/O-intensive out-of-core computations. This research shows that block mapping I/O management may improve the performance of out-of-core problems by one order of magnitude or more. We have studied the relationship between data access patterns and data locality optimization and created a tool for compile time data transfer analysis. Our extension of Comanche involves the use an out-of-core programming schema for optimizing locality in loop nested to minimize I/O.

Key Words: I/O Management, VMM, Out-of-core, Compiler, Tile mapping