NASA

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle High Performance Input/Output Systems for High Performance Computing and Four-Dimensional Data Assimulation		5. Report Date 6. Performing Organization Code
7. Author(s) Geoffrey Fox		8. Performing Organization Report No. 10. Work Unit No.
9. Performing Organization Name and Address Syracuse University 113 Browne Hall Syracuse, New York 13244-5300		11. Contract or Grant No. NAS5-32337 USRA subcontract No. 5555-26
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001 NASA Goddard Space Flight Center Greenbelt, MD 20771		13. Type of Report and Period Covered Final August 1993 - July 1997 14. Sponsoring Agency Code

15. Supplementary Notes

This work was performed under a subcontract issued by
Universities Space Research Association
10227 Wincopin Circle, Suite 212
Columbia, MD 21044

Task 17

16. Abstract

The approach of this task was to apply leading parallel computing research to a number of existing techniques for assimilation, and extract parameters indicating where and how input/output limits computational performance. The following was used for detailed knowledge of the application problems:

1. Developing a parallel input/output system specifically for this application
2. Extracting the important input/output characteristics of data assimilation problems; and
3. Building these characteristics s parameters into our runtime library (Fortran D/High Performance Fortran) for parallel input/output support.

17. Key Words (Suggested by Author(s)) Parellel And Scalable Software for Input/Output	18. Distribution Statement Unclassified--Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	1	

NASA Form 1626 Oct 86

JUN 30 1997

High Performance Input/Output System for High Performance Computing and Four-Dimensional Data Assimilation

Final Report (7/1993 - 6/1997)
Geoffrey C. Fox, PI, Northeast Parallel Architectures Center
Chao-Wei Ou, Northeast Parallel Architectures Center

Project Overview

The Northeast Parallel Architectures Center of Syracuse University is applying basic computer science research in high performance input/output systems for parallel computers to the NASA grand challenge applications of four-dimensional data assimilation. Our approach is to apply leading parallel computing research to a number of existing techniques for assimilation, and extract parameters indicating where and how input/output limits computational performance. Using detailed knowledge of the application problems, we are:

- developing a parallel input/output system specifically for this application;
- extracting the important input/output characteristics of data assimilation problems; and
- building these characteristics as parameters into our runtime library (Fortran D/High Performance Fortran) for parallel input/output support.

Research Activities

1 **PASSION**: Parallel And Scalable Software for Input-Output

I/O for parallel systems has drawn increasing attention in the last few years as it has become apparent that I/O performance rather than CPU or communication performance may be the limiting factor in future computing systems. Large scale scientific computations, in addition to requiring a great deal of computational power, also deal with large quantities of data. At present, a typical Grand Challenge Application could require 1Gbyte to 4Tbytes of data per run. These figures are expected to increase by orders of magnitude as teraflop machines make their appearance. Although supercomputers have very large main memories, the memory is not large enough to hold this much amount of data. Hence, data needs to be stored on disk and the performance of the program depends on how fast the processors can access data from disks. Unfortunately, the performance of the I/O subsystems of MPPs has not kept pace with their processing and communications capabilities. A poor I/O capability can severely degrade the performance of the entire program. The need for high performance I/O is so significant that almost all the present generation parallel computers provide some kind of hardware and software support for parallel I/O.

In order to develop a successful assimilation system for Earth Science, there will be a continual need to process and reprocess data sets with ever-improving and more complete assimilation system. There will also be a requirement to diagnose the quality of the data sets. Data assimilation provides the most compute-intensive as well as I/O-intensive undertaking in NASA Earth Science research, and therefore, high-performance I/O capability will be essential to new generation data assimilation systems. The objective of designing the PASSION software is to develop software support for parallel I/O that permits scalable I/O operations to match the growing computational power of the new parallel supercomputer.

At Syracuse University, we consider the I/O problem from a language, compiler and runtime support point of view. We are developing a compiler and runtime support system called PASSION: Parallel And Scalable Software for Input-Output. PASSION software support is targeted for I/O intensive out-of -core loosely synchronous problems. The PASSION Runtime Library provides routines to efficiently perform the I/O required in out-of-core programs. The goal of the PASSION compiler is to translate out-of-core programs written in a data-parallel language like High Performance Fortran (HPF) to node programs with calls to the PASSION Runtime Library for I/O. Other components of the PASSION project include a Portable Parallel File System (VIP-FS), integrating task and data parallelizm using parallel I/O and file servers for multimedia applications.

1.1 PASSION Runtime Support for Parallel I/O

In out-of-core computations, data is stored in files on secondary storage such as disks. During program execution, data needs to be moved back and forth between disks and main memory. The PASSION Runtime Library provides routines to efficiently perform the I/O required in out-of-core programs. It provides support for loosely synchronous out-of-core computations which use a Single Program Multiple Data (SPMD) Model. PASSION uses a simple high-level interface, which is a level higher than any of the existing parallel file system interfaces. For example, the user only needs to specify what section of the array needs to be read in terms of its lower-bound, upper-bound and stride in each dimension, and the PASSION Runtime Library will fetch it in an efficient manner. PASSION thus provides a simple and portable level of abstraction above the native parallel file system provided on the machine. PASSION is designed to either be directly used by application programmers, or a compiler can translate out-of-core programs written in a high-level data parallel language like High Performance Fortran (HPF) to node programs with calls to the PASSION Runtime Library for I/O. A number of optimizations such as Two-Phase I/O, Data Sieving, Data Prefetching and Data Reuse, have been incorporated in the library for improved performance.

1.2 PASSION Compiler Support for Parallel I/O

The goal of the PASSION compiler is to compile out-of-core data parallel programs written in a language such as High Performance Fortran (HPF). The PASSION compiler has to perform the following two main tasks

- Read and write distributed arrays. The compiler obtains distribution information from the HPF directives.
- Perform automatic program transformations to improve I/O performance.

The PASSION Compiler takes an HPF program as an input and generates an node+MP+I/O program, with calls to the PASSION Runtime Library.

PASSION compiler uses two distinct models for compiling an out-of-core program. First model is called the Local Placement Model (LPM) and second model is called the Global Placement Model (GPM) .

2 **TCE**: Thread-based Communication Environment

TCE employs light-weight multi-threading. It assumes each I/O event is independent (a thread) and that scalable I/O can be accomplished by managing a parallel/distributed thread queue. The integration between PASSION and TCE can offer a more robust and unified view toward using meta-computing for scalable heterogeneous I/O for four-dimensional data assimilation.

TCE is designed:

- to provide an efficient, thread--based communication library capable of supporting distributed and parallel processing on variety on platforms.
- to ensure interoperability between different types of architectures with different CPUs and Operating Systems.
- to make the environment as simple as possible without compromising the performance or functionality.
- to assist the programmer in choosing computational nodes and style of interactions between his processes.

By abstract communication objects called ports and channels it possible to build the client-server connection as well as peer-to-peer parallel relations. By mapping application processes onto computational nodes both data parallelism and functional parallelism can be exploited. These two paradigms can be even mixed in one application. The multithreaded support is based on a user-level thread package, thus TCE can be easily ported to different processors. Different architecture are supported :

- clusters of heterogenous workstations -- through ports and channels
- shared memory parallel machines -- through multithreading
- distributed memory parallel machines -- through ports and channels

The differences in data formats (byte ordering) is taken care of internally by the library. Machine specific IPC operations are masked through higher level operations on channels and ports.

3 **SPRINT**: Scalable Partitioning, Refinement and INcremental partitioning Techniques

Load balancing of the distributed heterogeneous system is vital scalable I/O. Our use of meta-computing for large-scale four-dimensional data assimilation problems makes this more critical. The load balancing problem can be viewed as a graph-partitioning problem. Graph-partition problems belong to the class of NP-complete problems; hence exact solutions are computational intractable for large problems. However, good suboptimal solutions are sufficient for effective parallelization of most of applications. SPRINT - Scalable Partitioning, Refinement and INcremental partitioning Techniques - collects three important partitioning methods based on physical information (e.g., recursive inertial bisection, recursive orthogonal bisection and index-based partitioning). Index-based partitioning methods not only can be used to partitioning graphs but also can be used to improved the disk allocation.

Efficient methods for graph partitioning and incremental graph partitioning are important for parallelization of a large number of unstructured and/or adaptive applications. The key problem in efficiently executing irregular and unstructured data parallel applications is partitioning the data to minimize communication while balancing the load. Partitioning such applications can be posed as a graph-partitioning problem based on the computational graph. We have developed a library of partitioners (especially based on physical optimization) which aim to find good suboptimal solutions in parallel. This initial target use of these partitioning methods are for runtime support of data parallel compilers (HPC, HPC++, HPF, etc.).

SPRINT software focuses on a subclass of applications in which the computational graph is such that the vertices correspond to two- or three-dimensional coordinates, and the interaction between computations is limited to vertices that are physically proximate. Examples of such applications include finite element calculations, molecular dynamics, particle dynamics, particle-in-a-cell, region growing, and statistical physics. For these applications, partitioning can be achieved by exploiting the above property. Essentially proximate points are clustered together and form a partition such that the number of points attached to each partition are approximately equal. Most of the interactions are local and the amount of interprocessor communication is low if proximate points are clustered together.

The SPRINT library provides software for parallel graph-partitioning using coordinate information such as index-based partitioning, recursive coordinate bisection and recursive inertial bisection. SPRINT also provides incremental partitioning techniques based on the index-based method.

4 **A Real-time Terrain Rendering Application on a PC Cluster**

The goal of this terrain rendering project is to provide the user a real time interactive viewing environment for the available terrain data. We envision that the user starts out in the solar system, where Mars, Earth and Venus are visible. Then the user chooses to visit one of the three planets.

This journey can be broken into four main viewing stages: stage one, from solar orbit to high planetary orbit; stage two, from high planetary orbit to lower planetary orbit; stage three, from lower planetary orbit to high altitude flight path; and stage four, from high altitude flight path to low altitude fly-by. Each of these stages has distinct viewing characteristics that the terrain viewer must respect. As a result, multiple rendering techniques and data sets are needed to generate the images. In the following sections, the viewing characteristics, data requirements and rendering techniques of each stage are investigated.

We focused on a distributed PC cluster pioneered by Beowulf system developed by NASA. We have experimented this application on a 4-node 486/100MHz PC Cluster with LINUX. This effort was limited because of short funding.

Achievements Summary

- **PASSION** can either be directly used by application programmers or a compiler. *http://www.cat.syr.edu/passion.html*
- **TCE** can offer a more robust and unified view toward using meta-computing for scalable heterogeneous I/O for four-dimensional data assimilation. *http://www.npac.syr.edu/users/gcf/cps616threads/*
- **SPRINT** can be used to partitioning graphs but also can be used to improved the disk locality. *http://dante.npac.syr.edu:1996/SPRINT/index.html*
- **A Real-time Terrain Rendering Application** has been conducted on a PC Cluster. *http://www.npac.syr.edu/users/alvin/papers/terrain/terrain.html*

References

1. Rajesh Bordawekar, Alok Choudhary and J. Ramanujam. " Automatic Optimization of Communication in Out-of-core Stencil Codes". Scalable I/O Technical Report 114. November 1995. In Proc. of 10th ACM Intl. Conference on Supercomputing, May 1996.
   http://www.cat.syr.edu/~rajesh/ics96.ps
2. Rajesh Bordawekar, Alok Choudhary and J. Ramanujam. " Compilation and Communication Strategies for Out-of-core programs on Distributed Memory Machines". Scalable I/O Technical Report 113. November 1995.
   http://www.cat.syr.edu/~rajesh/jpdc.ps
3. Rajesh Bordawekar and Alok Choudhary. "Communication Strategies for Out-of-core Programs on Distributed Memory Machines". In Proc. of 9th ACM Intl. Conference on Supercomputing, July 1995.
   http://www.npac.syr.edu/techreports/html/0650/abs-0667.html

4. Rajeev Thakur. " Runtime Support for In-Core and Out-of-Core Data-Parallel Programs". Ph.D. Thesis, Dept. of Electrical and Computer Eng., Syracuse University, May 1995.
ftp://ftp.npac.syr.edu/pub/users/thakur/papers/phd_thesis.ps.Z
5. Rajeev Thakur and Alok Choudhary. " An Extended Two-Phase Method for Accessing Sections of Out-of-Core Arrays". Scalable I/O Initiative Technical Report CACR-103, Center for Advanced Computing Research, Caltech, June 1995.
ftp://ftp.npac.syr.edu/pub/users/thakur/papers/ext2ph.ps.Z
6. Rajesh Bordawekar, Alok Choudhary, Ken Kennedy, Charles Koelbel and Michael Paleczny. " A Model and Compilation Strategy for Out-of-Core Data Parallel Programs". In Proc. of the Fifth ACM SIGPLAN Symposium on Principles and Practices of Parallel Programming, July 1995.
http://www.npac.syr.edu/techreports/html/0650/abs-0696.html
7. Alok Choudhary, Rajesh Bordawekar, Sachin More, K. Sivaram and Rajeev Thakur. " PASSION: Runtime Library for the Intel Paragon". In Proc. of the Intel Supercomputer User's Group Conference, June 1995.
ftp://ftp.npac.syr.edu/pub/users/thakur/papers/isug95-passion.ps.Z
8. Rajesh Bordawekar and Alok Choudhary. " A Framework for Representing Data Parallel Programs and its Application in Program Reordering". NPAC Technical Report SCCS-698, March 1995.
http://www.npac.syr.edu/techreports/html/0650/abs-0698.html
9. Alok Choudhary, Rajesh Bordawekar, Michael Harry, Rakesh Krishnaiyer, Ravi Ponnusamy, Tarvinder Singh and Rajeev Thakur. " PASSION: Parallel and Scalable Software for Input-Output". NPAC Technical Report SCCS-636, Sept. 1994.
ftp://ftp.npac.syr.edu/pub/docs/sccs/papers/ps/0600/sccs-0637.ps.Z
10. Rajeev Thakur, Rajesh Bordawekar, Alok Choudhary, Ravi Ponnusamy and Tarvinder Singh. " PASSION Runtime Library for Parallel I/O". In Proc. of the Scalable Parallel Libraries Conference, October 1994.
ftp://ftp.npac.syr.edu/pub/users/thakur/papers/splc94_passion_runtime.ps.Z
11. Rajeev Thakur, Rajesh Bordawekar and Alok Choudhary. " Compiler and Runtime Support for Out-of-core HPF Programs". In Proc. of 8th ACM Int. Conf. on Supercomputing, July 1994, pp. 382-391.
ftp://ftp.npac.syr.edu/pub/projects/pcrc/f90d/docs/ics94-out-of-core-hpf.ps.Z
12. Michael Harry, Juan Miguel del Rosario and Alok Choudhary. "VIP-FS: A VIrtual, Parallel File System for High Performance Parallel and Distributed Computing". In Proc. of Ninth International Parallel Processing Symposium, April 1995.
ftp://ftp.npac.syr.edu/pub/docs/sccs/papers/ps/0650/sccs-0686.ps.Z
13. Rajesh Bordawekar, Juan Miguel del Rosario and Alok Choudhary. "Design and Implementation of Primitives for Parallel I/O". In Proc. of Supercomputing'93, Portland, OR, November 1993.
http://www.npac.syr.edu/techreports/html/0550/abs-0564.html
14. Rajesh Bordawekar, Juan Miguel del Rosario and Alok Choudhary. "An Experimental Evaluation of Touchstone Delta Concurrent File System". In Proc. of International Conference on Supercomputing 1993, July 1993.
http://www.npac.syr.edu/techreports/html/0400/abs-0420.html

15. Rajesh Bordawekar, Alok Choudhary and Rajeev Thakur. " Data Access Reorganizations in Compiling Out-of-core Data Parallel Programs on Distributed Memory Machines". NPAC Technical Report SCCS-622, Sept. 1994.
ftp://ftp.npac.syr.edu/pub/docs/sccs/papers/ps/0600/sccs-0622.ps.Z
16. Rajesh Bordawekar. " Issues in Software Support for Parallel I/O". Master's Thesis, ECE Dept, Syracuse University, May 1993.
http://www.npac.syr.edu/techreports/html/0450/abs-0487.html
17. Juan Miguel del Rosario, Rajesh Bordawekar and Alok Choudhary. " Improved Parallel I/O via a Two-Phase Run-time Access Strategy". IPPS'93 Workshop on Input/Output in Parallel Computer Systems, April 1993.
http://www.npac.syr.edu/techreports/html/0400/abs-0406.html
18. Divyesh Jadav, Chutimet Srinilta, Alok Choudhary and P. Bruce Berra. " Design and Evaluation of Data Access Strategies in a High Performance Multimedia-on-Demand Server". In Proc. of the Second Intl. Conf. on Multimedia Computing and Systems, May 1995.
ftp://ftp.npac.syr.edu/pub/projects/pcrc/f90d/docs/ICMCS95.ps.Z
19. Divyesh Jadav and Alok Choudhary. " Design Issues in High Performance Media-on-Demand Servers". IEEE Parallel and Distributed Technology Systems and Applications, Summer 1995.
ftp://ftp.npac.syr.edu/pub/projects/pcrc/f90d/docs/PDT.ps.Z
20. Divyesh Jadav, Chutimet Srinilta, Alok Choudhary and P. Bruce Berra. " Techniques for Scheduling I/O in a High Performance Multimedia-On-Demand Server". The Journal of Parallel and Distributed Computing, Fall 1995.
ftp://ftp.npac.syr.edu/pub/projects/pcrc/f90d/docs/JPDC.ps.Z
21. Divyesh Jadav, Chutimet Srinilta, Alok Choudhary and P. Bruce Berra. "An Evaluation of Design Tradeoffs in a High Performance Media-on-Demand Server". ACM Multimedia Systems Journal , January 1995.
22. Divyesh Jadav, Chutimet Srinilta and Alok Choudhary. "I/O Scheduling Tradeoffs in a High Performance Media-on-Demand Server". The 2nd Intl. Conference on High Performance Computing, December 1995, New Delhi, India.

23. Bhaven Avalani, Alok Choudhary, Ian Foster and Rakesh Krishnaiyer. " Integrating Task and Data Parallelism Using Parallel I/O Techniques". In Proc. of International Workshop on Parallel Processing, Dec. 1994.
ftp://ftp.npac.syr.edu/pub/projects/pcrc/f90d/docs/task_data.ps.Z
24. Chao-Wei Ou, Manoj Gunwani, and Sanjay Ranka. "An Architecture-Independent Locality-Improving Transformations of Computational Graphs Embedded in k-Dimensions". December 1994, ICS'95, July 1995, pp. 289-298.
ftp://ftp.npac.syr.edu/pub/docs/sccs/papers/ps/0700/sccs-0728.ps.Z
25. Chao-Wei Ou and Sanjay Ranka. " Parallel Incremental Graph Partitioning", IEEE Transactions on Parallel and Distributed Systems. to appear. (A preliminary version appeared as "Parallel Incremental Graph Partitioning Using Linear Programming" in Supercomputing '94. November 1994, pp. 458-467.)
ftp://ftp.npac.syr.edu/pub/docs/sccs/papers/ps/0650/sccs-0653.ps.Z

26. Chao-Wei Ou and Sanjay Ranka. "Parallel Remapping Algorithms for Adaptive Problems". Journal of Parallel and Distributed Computing, under revision. (A preliminary version of the paper appeared in Frontiers' 95, February 1995, pp. 367-374).
ftp://ftp.npac.syr.edu/pub/docs/sccs/papers/ps/0650/sccs-0652.ps.Z
27. Chao-Wei Ou, Sanjay Ranka and Geoffrey Fox. "Fast and Parallel Mapping Algorithms for Irregular Problems". Journal of Supercomputing, 1996, 10, pp. 119-140.
ftp://ftp.npac.syr.edu/pub/docs/sccs/papers/ps/0700/sccs-0729.ps.Z