SCALABLE DIGITAL LIBRARIES OF EVENT DATA AND THE NSCP META- CLUSTER

Stuart Bailey, Robert Grossman, and David Hanley
University of Illinois at Chicago, Chicago, IL, USA
Don Benton and Bob Hollebeek
University of Pennsylvania, Philadelphia, PA, USA

In this paper we present the design, implementation, and experimental results of a system to mine and visualize event data using cluster computing built upon an ATM network. Our approach is to build a system using light weight, modular software tools for data management, resource management, data analysis and visualization developed for local, campus and wide area clusters of workstations.

1 Introduction

In this paper we present the design, implementation, and experimental results of a system to mine and visualize high energy physics event data using cluster computing built upon an Asynchronous Transfer Mode (ATM) network. Our approach is to build a system using light weight, modular software tools for data management, resource management, data analysis and visualization developed for local, campus and wide area clusters of workstations.

By using an ATM network, we can scale by adding additional workstations to the various local clusters, and by adding additional local clusters, without suffering the performance bottlenecks associated with a traditional shared media network infrastructure. In this way, we can build a Meta-Cluster, or cluster of workstation clusters. The ATM network also allows us to exploit parallel input-output techniques so that we can scale both the data management capability of the Meta-Cluster by adding additional ATM circuits, as well as the processing capability of the Meta-Cluster by adding additional workstations.

This work was done as part of the National Scalable Cluster Project whose goal is to develop algorithms, software and model applications exploiting high performance broad band networking in support of local and wide area clusters of workstations and high performance computers. Currently, the collaboration involves the University of Illinois at Chicago (UIC), the University of Maryland (UMD), the University of Pennsylvania (UPenn), as well as IBM, Xerox and other corporate sponsors.

The project is in its initial phase and consists of laboratory and campus ATM clusters of workstations at UIC, and UPenn and UMD, as well as an IBM SP-2 at UPenn. During the fourth quarter of 1995, the three laboratory and campus ATM clusters will be connected to form a Meta-Cluster using an ATM cloud.

In Section 2, we describe the network and hardware infrastructure of the system. In Section 3, we describe the software infrastructure. The architectural design of the software system is based upon the following ideas:

- *Light Weight Object Management.* We employ an underlying object data model for the event data and manage the events using a light weight persistent

object manager called PTool in contrast to using a file system or database to manage the data. For this project, a version of PTool was developed and optimized for clusters of workstations. We also implement a distributed data scheme in order that data mining applications can benefit from a parallel 1/0 component.

- *Parallel Computation.* An important component of the NSCP is to explore the best means of interfacing high performance data management tools such as PTool with high performance computing tools exploiting emerging standards such as MPI.

- *Resource Management.* Jobs submitted to the meta-cluster are provided the necessary nodes and storage resources using resource management software. An important aspect of the NSCP is to develop better resource management tools for local and wide area cluster computing.

. *Formal Modular Components.* We are beginning to experiment with using formal methods to define the interfaces between different components in our system. For example, the APIs provided by light weight object managers may be all that is needed for some data analysis and visualization applications, while others may require the additional functionality provided by a CORBA interface.

In Section 4, we will describe the initial work in data modeling and analysis of HEP data that is being carried out on the Meta-Cluster.

2 Network and Hardware Infrastructure

One goal of the NSCP is to experiment with using an ATM network for both local and wide area clusters of workstations. The NSCP Meta-Cluster can scale both by adding additional workstations to existing clusters and by adding additional clusters. Since workstation to workstation communication is done using a "virtual circuit" rather than sharing a network medium, adding workstations does not necessarily adversely effect Meta-Cluster performance. Of course, as more workstations are added, the underlying ATM network infrastructure may have to be increased so that additional circuits are available when needed.

Many scalable, high performance cluster architectures have been developed such as the IBM SP-2. These architectures utilize specialized very high speed switching technology as the primary network infrastructure and standard workstations as the compute resources. With Asynchronous Transfer Mode (ATM) emerging as a standard and commodity high speed switching technology, high performance clusters can be constructed using commodity workstations. Furthermore, the adoption of ATM as a standard in the telecommunications industry, allows simple, yet high performance, wide area connectivity of geographically dispersed compute clusters into meta-clusters.

The high speed capacity of ATM technology also allows the NSCP to utilize parallel 1/0 techniques to manage potentially widely distributed data storage resources. The storage resources of the Meta-Cluster can be shared and scaled just

as the compute resources. Multi-Gigabyte drives and other storage devices placed on individual nodes of the cluster can all be shared and managed by the distributed storage resource software of the NSCP software infrastructure described below.

3 Software Infrastructure

This section is a summary of the material in "A Case for Using Light Weight, High Performance Persistent Object Managers in Scientific Computing[3]."

The software infrastructure of the NSCP employs standard and emerging distributed technologies to form three distinct, light weight layers. By light weight we mean that we emphasize combining "small" software tools rather than using "large" monolithic applications. We are especially interested in software tools with simple APIs and which employ low overhead protocols. In this way Meta-Cluster applications can access any desired software resources directly when needed. This approach encourages the layering of software and the incremental development of system and application software. The next three subsections describe the three software layers of the NSCP.

Low Overhead Object Management. A consensus is emerging within the HEP community that there are distinct advantages to computing using objects rather than using data with less structure and whose interfaces are less well defined. Objects which exist independently from the process which create them are called *persistent,* in contrast to transient objects whose existence terminates with the process which creates them.

Working with persistent objects represents a fundamental change: rather than work, as is traditional in high performance computing, with file based access to data, we are advocating viewing data as a distributed collection of persistent objects and working with the objects using light weight persistent object managers employing low overhead protocols. For conciseness, we speak of Low Overhead Object Management (LOOM) and Low Overhead Object Protocols (LOOP).

We have been using a low overhead object manager that we have developed called PTool[4, 5, 6]. PTool allows transparent access to Terabytes of data that may be striped across a number of nodes, and furthermore may be transparently compressed or committed to tertiary storage.

Parallel Computation. There are many software tools for message passing in cluster environments. Parallel Virtual Machine (PVM) and Message Passing Interface (MPI) are among the most widely used. The benefits of these tools on traditional clusters are well documented. The use of these tools over ATM is relatively new. PVM and, the more general, MPI have both recently been ported to utilize native ATM protocols, [2] and suggest favorable results. Therefore, the NSCP has adopted both of these libraries as the parallel computation software layer. An application can connect to the library of its choice though the API.

Resource Management. In a widely distributed Meta-Cluster, optimal resource usage depends on many factors include, load, availability, connection speeds, and connection times. In order that applications have as close to optimal resource usage as possible, a resource management software layer is needed. Such a layer

handles distribution and scheduling of individual job requests as well as processes within a parallel job. The resource management layer of the NSCP consists of storage resource management and compute resource management. Storage resource management should not be confused with the low overhead object management. LOOM is the logical management of data objects that are meaningful to the programmer. Storage resource management is the management of individual physical devices throughout the Meta-Cluster. Therefore, the LOOM layer might utilize the resource management layer to find the optimal device or devices on which to place objects. Likewise, the parallel computation layer might utilize the resource management layer to find the optimal processors on which to spawn the scatter phase of a "scatter and gather" operation.

For computation resource management, the NSCP is employing Platform Computing's Load Sharing Facility. In the future, we will be developing a storage resource manager that can be utilized by the LOOM layer of the software infrastructure and directly by applications if needed.

The three software layers combine emerging and standard technologies in cluster computing to provide a robust development and run-time environment for supercomputing applications such as the mining of HEP data.

4 Data Mining of HEP Data with the NSCP

We have made some initial trials using the NSCP tools in the processing of HEP data. The data is derived from the "Exotic" triggers on the CDF detector. This type of data was chosen both because of the interest of the authors in the possible physics which can be done with it, and because it is the type of data where the optimal processing strategy is not obvious and where the data might need to be mined many times. An initial sample of approximately 60 GBytes occupies a portion of the 300 GBytes currently attached to the UPenn IBM SP2. This data is in the traditional "raw" CDF form. This data can be reformatted into PTool object form or can be used in the raw form. It is important at this developmental stage to be able to do both.

For analysis of the data, traditional FORTRAN tools as well as PAW analysis tools can be applied to the data. Since this raw form is based on simple banks (using YBOS), we have found it possible to view most banks as objects. Indeed, we have implemented C++ code which is capable of reading both the raw and the PTool forms. This code converts the raw data form to object form. Traditional pointer based references in FORTRAN can be completely eliminated in the equivalent C++ code, and we find that relatively simple examples of physics analysis code containing hundreds of lines of complex code translate to tens of lines of quite understandable (to the non-expert eye) C++ code.

The translation of the data from bank to object and FORTRAN to C++ is accomplished by header files which describe the bank structure for raw banks. Additional objects constructed from these banks and available only in the C++ version can be used to further simplify complex physics analyses. Since many analysis tasks can already be done with the legacy FORTRAN code, it is to our advantage to structure the new code in a way which allows it to take advantage when possible of the

vast libraries of existing code. This has been accomplished by formulating the objects in a way which preserves their internal YBOS bank structure. As a result, by modifying the normal routine which locates the beginning of the bank in memory so that it points to the relevant object in C++, we have been able to write hybrid code which takes advantage of the legacy FORTRAN when necessary but can also take advantage of the special capabilities of the C++ language.

In addition to being able to write analysis code which is simpler to code and understand, the new techniques have the advantage that different parts of the data stream (banks) can be handled by different object oriented input routines. Analysis code which does not use a particular component can "skip" this data without reading it. Persistent object techniques can also be used to retrace data which has been previously analyzed. Finally, to further decrease the time necessary to process all of the data, we are taking advantage of the parallel capabilities of PENN's IBM/SP2. An initial parallel implementation (in IBM's Parallel Operating Environment) of this code is in the initial test stages.

Acknowledgments and for Additional Information

This research was supported in part by NASA grant NAG2-513, DOE grant DE-FG02-92ER25133, and NSF grants IRI 9224605 and CDA-9303433, and the National Scalable Cluster Project.

For additional information, contact R. Grossman, Laboratory for Advanced Computing, University of Illinois at Chicago, 851 S. Morgan Street, Chicago, IL 60607, USA, grossman@uic.edu.

1. S. Chang, D.H. Du, J. Hsieh, R.P. Tsang, M. Lin. "Enhanced PVM Communications over a High-Speed LAN," *IEEE Parallel and Distributed Technology,* Vol. 3, Num. 3, Fall 1995.
2. P.W. Dowd, S.M. Srinidhi, F.A. Pellegrino, T.M. Carrozzi, D.L. Guglielmi, R. Claus. "Impact of Transport Protocols and Message Passing Libraries on Cluster-based Computing Performance," *Proc. IEEE MILCOM95*, Oct. 1995.
3. S. Bailey, R. L. Grossman, and D. Hanley, "A Case for Using Light Weight, High Performance Persistent Object Managers in Scientific Computing," submitted for publication.
4. R. L. Grossman and X. Qin, "Ptool: a scalable persistent object manager," *Proceedings of SIGMOD* 94, ACM, 1994, page 510.
5. R. L. Grossman, N. Araujo, X. Qin, and W. Xu, "Managing physical folios of objects between nodes," *Persistent Object Systems (Proceedings of the Sixth International Workshop on Persistent Object Systems),* M. P. Atkinson, V. Benzaken and D. Maier, editors, Springer-Verlag and British Computer Society, 1995.
6. R. L. Grossman, X. Qin, D. Valsamis, W. Xu, C. T. Day, S. Loken, J. F. MacFarlane, D. Quarrie, E. May, D. Lifka, D. Malon, L. Price, "Analyzing High Energy Physics Data Using Databases: A Case Study," *Proceedings of the Seventh International Working Conference on Scientific" and Statistical Database Management,* IEEE Press, 1994.