

An InterBase System at BNR

Omran Bukhres*, Jiansan Chen*, and Rob Pezzoli
BNR Inc.
35 Davis Drive
RTP, NC 27709-3478
ROBP@BNR.CA

1 Introduction

Like many companies with distributed applications, the computing environment at Bell Northern Research (BNR) Inc. is a collection of heterogeneous software packages and hardware platforms. The BNR network includes IBM mainframes and workstations scattered across the U. S., Canada, Great Britain, and Japan. As computing resources have been distributed to the various BNR sites, a heterogeneous network of self-controlled or autonomous computing services has arisen. Consequently, although many computing services exist, there is usually a lack of integration, accessibility, and global information sharing. Knowing what applications are available, what computer systems house these applications, and how to access these applications is a formidable task.

In response to problems typically associated with such environments, in Purdue University's InterBase Lab, research has led to the design of an environment for global applications. This environment, called InterBase [BCD⁺92], provides a tool-based interface that facilitates application development in a distributed environment of heterogeneous software resources such as databases, tool libraries, and application programs. InterBase has been designed to take care of the details of locating and starting remote services, transferring and transforming data among different services, managing failures, and controlling parallelism between multiple global applications running concurrently.

*Computer Sciences Dept, Purdue University, W. Lafayette, IN 47907, {bukhres,jchen}@cs.purdue.edu

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

SIGMOD /5/93/Washington, DC, USA

© 1993 ACM 0-89791-592-5/93/0005/0426...\$1.50

BNR anticipates realizing two major benefits from the utilization of the InterBase System. The first benefit is cost reduction. Since InterBase can reduce the need for global databases by obtaining data from its local sources, fewer resources are dedicated to disk space and database administration. The second benefit is convenience; users can now access and process data from otherwise inconvenient or disjointed data sources, such as Ingres databases on HP workstations and DBS databases on IBM mainframes.

This paper describes, in general terms, the current application of the InterBase System at BNR and the future projects which could benefit from the use of InterBase.

2 System Architecture

The logical architecture of InterBase at BNR, which is conforms to the model implemented at Purdue University, is illustrated in Figure 1. Physically, the InterBase System at BNR consists of a Distributed Flex Transaction Manager (DFTM) that is distributed among several SUN and HP workstations and eight Remote System Interfaces (RSIs) that are installed on different computer platforms, which consist of UNIX workstations and mainframes running IBM CMS Operating System.

There are two major components in InterBase: the DFTM, and a set of RSIs, as depicted in Figure 1. The DFTM is the center of InterBase which interprets and coordinates the reliable execution of global applications over the entire system. The DFTM provides a distributed task specification language, the InterBase Parallel Language (IPL) [CBE93], by which users can specify a global application. A graphical interface, InterBaseView, is also provided over InterBase which aids users in writing and executing global applications in IPL. RSIs are specially designed InterBase

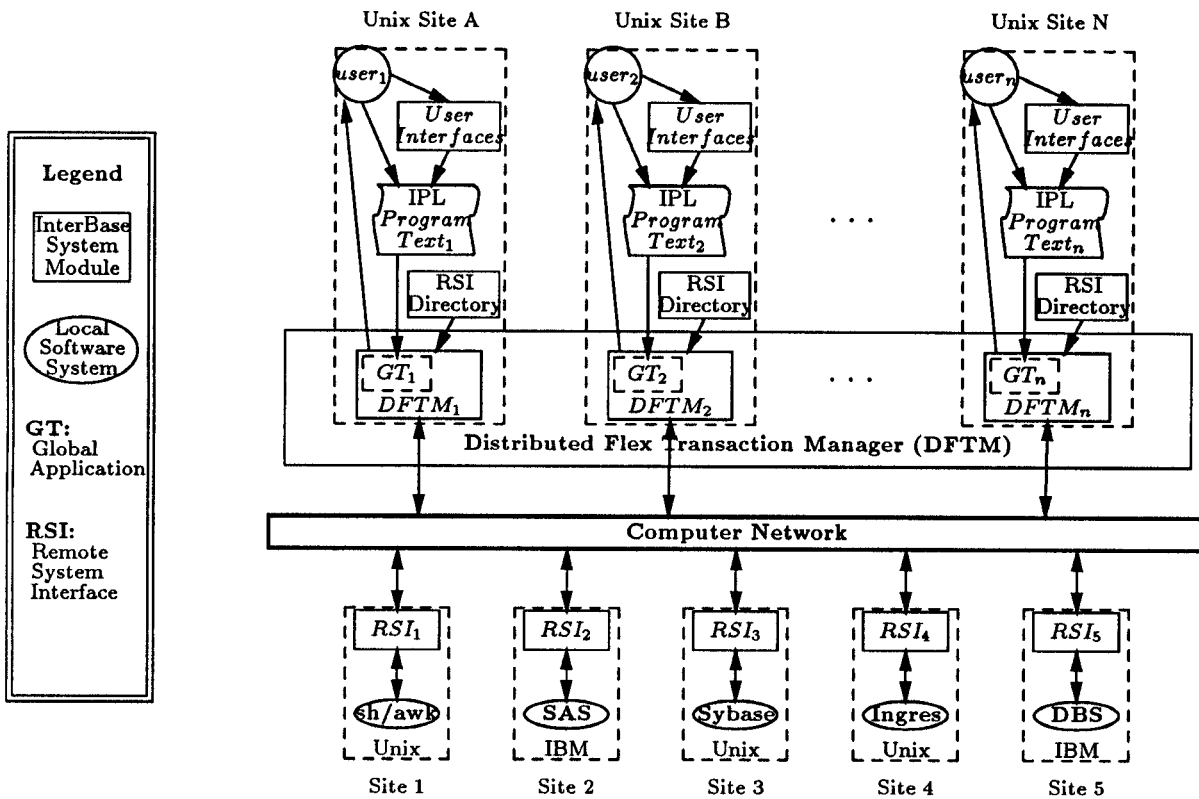


Figure 1: The logical architecture of InterBase at BNR

agents which are superimposed on the individual software systems. RSIs provide a uniform system-level interface between the DFTM and the software systems, deal with the heterogeneity of the software systems, including transforming command and data format among RSIs, thus relieve the DFTM from dealing with each software system directly. The RSI Directory stores information such as location, communication and connection protocols, as well as allowable data transfer methods for different RSIs, thus supporting location and distribution transparency for the system.

The interactions among InterBase modules are presented in Figure 1, where arrows indicate the data and control flow. Currently, InterBase runs on an interconnected network with a variety of hosts, such as UNIX workstations and IBM mainframes, and supports global applications accessing many software systems including SAS, Sybase, Ingres, DBS, and UNIX utilities. The InterBase System represented in Figure 1 is a simplified logical architecture; physically, it can be tailored to meet different physical environments. For example, if there are several Ingres database systems incorporated into InterBase, a distinct RSI must

be provided for each, although the Ingres RSIs are identical in their codes. Different RSIs can be run simultaneously on the same machine. The InterBase System therefore provides flexibility for heterogeneous system integration.

One of the strengths of this architecture is its decentralized nature, which can be seen in Figure 1. The DFTM is distributed over all the machines from which global applications are executed; that is, an DFTM image is responsible for the consistent and reliable execution of its associated global application. For example, IPL program $Text_1$ is executed as the global application GT_1 which is in turn carried out by the DFTM image $DFTM_1$, as illustrated in Figure 1. Exchange of information within Interbase is performed via computer network, so that each module of InterBase has location transparency.

3 Challenges

Several challenges were confronted while installing InterBase at BNR. For example, InterBase had not previously been installed on a mainframe under CMS.

As mainframes employ an atypical C compiler (White-Smiths LTD's C Language for System/370), correct compilation of the codes proved to be a complex process. A formidable challenge was presented by the fact that only one application can run within a virtual machine (VM) at a given time. While InterBase components must run continuously in order to detect an incoming request for service, if the VM is constantly engaged in looking for requests it cannot at the same time execute the service it is being asked to perform. This problem was approached by dividing the corresponding RSIs into two parts: one is on UNIX a site, responsible for processing incoming request for service; the other part is on a mainframe site, responsible for executing the service. The communication between the two part is via computer network. Finally, it was found that BNR's mainframes use a communication protocol, Knet, which is different from that used in the InterBase Lab at Purdue University. It was necessary to analyze the Knet protocol and then write the appropriate assembly language programs. After the establishment of communications through these programs, the communication calls were incorporated in the C programs.

4 Accomplishments

The InterBase System at BNR allows users to develop global applications, including both database applications and data analysis applications, on workstations that access mainframes and other workstations. Users can now write RSIs for a mainframe that, strictly speaking, allows only one process to be resident in memory. These mainframe RSIs utilize Knet, which had not previously been used as a communication protocol. The restriction by the memory of the mainframe to a single resident application necessitated the delegation of a separate user id or virtual machine for each RSI. It was also necessary, through a few modifications, to adapt the system to another type of UNIX workstation from that on which it had originally been developed. RSIs were written for the workstation platforms so that workstations would be capable of intercommunication, as well as having access to the mainframe itself.

5 Functionality

Despite the small number of InterBase components installed on InterBase at BNR, the primary functions

of database access and data analysis are effectively provided. InterBase at BNR lets users develop global applications, including both database applications and data analysis applications, on workstations that access mainframes and other workstations.

The system allows users to operate through a workstation, to perform mainframe database queries and capture the resulting output. For example, a user on node A has a file of values, such as a list of last names, on node B, which map to a database key field on node C. The user may enter a query, such as "select projects where manager =", on node A, to which the results should be sent. The system thus learns the location of the database and the file of values, merges the query with the file of names to generate the complete query, and sends it to the database on node C. The results of the query are then sent back to node A.

Data analysis is the other important capability of the InterBase System at BNR. Data are accumulated in databases on many nodes and in data sets at different locations. Data capture is an integral part of data analysis. With the InterBase System, data capture and analysis can be accomplished in one global application, as illustrated in the following IPL program:

```

program
  subtrans S1 : charString use shell at nodeB
    beginexec
      cat /users/$HOME/search.values
    endexec
  endsubtrans
  subtrans S2 (S1) : charString use DBS at node A
    beginexec
      FIND ACTIVITIES WHERE ACTIVITY = !
      SET FIELD ORDER CURRENT
      ACTID SYSTEM PROJECT STATUS
      LIST ALL
      BYE
    endexec
  endsubtrans
  subtrans S3 (S2) : charString use SAS at nodeC
    beginexec
      options nocenter;
      input actid $ 1-6 system $ 8-14
        project $ 16-25 status $ 27-30;
      proc sort; by project status;
      proc means; freq status;
      proc print noobs;
      run;
    endexec
  endsubtrans
  dependency
    S1 : S2 ;
    S2 : S3 ;
  enddep

```

endprogram

The program executes several tasks. First, a file of values to search against is obtained from node B. Second, a query is generated on node A, comprised of DBS query language statements incorporated with the file of values produced at node B. This query is sent to the DBS database on the same node. Third, an analysis program in SAS is activated on node C, using the results returned from the second step; thus in turn generates a report of the results. All these steps are performed automatically within one IPL program.

Before using InterBase, the statisticians at BNR performed data analysis as follows: 1) login to corresponding databases manually and make queries to these databases to acquire required data; 2) transfer these data to the data analysis node; 3) transform these data into a format understandable by the statistics package; 4) run the statistics package with transformed data and obtain the statistics results. This procedure is both tedious and time-consuming, and the same analysis must be performed periodically because the data are kept to be updated. By using InterBase, all the above four steps can be described in one application program. During the execution of the application program, InterBase logs in to corresponding databases automatically and makes queries to these databases to acquire required data; it then transforms the data into the InterBase data format and transfers them to the data analysis node. These data are transformed automatically into a format understandable by the statistics package before InterBase sends them to the statistics package for execution. The results are automatically collected by InterBase and displayed at the node where the application program is initiated for execution. Before acquiring InterBase, each statistician could perform at most two data analyses per day; with InterBase, this number has been increased to at least ten data analyses per day. As a result, each data analysis takes at most one-fifth the time it did in the past.

6 Conclusions

The success of InterBase has encouraged BNR to pursue a broader development, encompassing more varieties and greater sophistication of applications, an increased number of platforms to which the system has been ported, and availability to a greater array of departments. For example, the Quality Department has initiated development of a database which collects in-

formation from several databases at different sites. InterBase will assist in initially populating the database and then periodically updating it, checking the data for accuracy and consistency. When completed, this project will simplify the retrieval of data from diverse databases on separate nodes and will provide a clean and consistent interface to users.

RSIs for other databases within BNR are being considered. These RSIs will allow users to simultaneously query several databases located on separate nodes. They will also have the capability to simultaneously update multiple autonomous databases, the functions which are more complex than read only functions.

We will continue to work in the following ways. First, we will host workshops with the purpose of introducing the InterBase System to many at BNR yet unfamiliar with it, demonstrating how the system is used and teaching the stuff how to write application programs using the IPL language. Since it is not difficult to write applications in IPL, it is hoped that new interfaces will be written by BNR staff to make other databases available to a broad range of departments within the organization. Second, we would like to write other applications for the Quality Department as a way of expanding the use of the current installation. Third, we will write a user interface to the system based on the X window interfaces written by Massachusetts Institute of technology (MIT) to make writing applications easier and consistent with other existing tools at BNR. Fourth, we are looking for other applications at BNR which will benefit from the capability of updating several autonomous database simultaneously. There are several projects which would be good candidates and which have shown interest in the InterBase System. The research results will be presented in future reports.

References

- [BCD+92] O. Bukhres, J. Chen, W. Du, A. Elmagarmid, and R. Pezzoli. An Overview of the InterBase System. Technical Report SERC-TR-127-P, Department of Computer Sciences, Purdue University, Nov 1992.
- [CBE93] Jiansan Chen, Omran A. Bukhres, and Ahmed K. Elmagarmid. IPL: A Multidatabase Transaction Specification Language. In *Proc. of the 13th International Conference on Distributed Computing Systems*, 1993. (to appear).