

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Massively Parallel Processing Distributed Database for Business Intelligence

Seifedine Kadry and Khaled Smaili
Faculty of Sciences, Lebanese University, Lebanon

Abstract: Business Intelligence (BI) is a business management term which refers to applications and technologies which are used together, provide access to and analyze data and information about company operations. Business intelligence systems can help companies have a more comprehensive knowledge of the factors affecting their business, such as metrics on sales, production, internal operations and they can help companies to make better business decisions. Business intelligence applications and technologies can help companies to store and analyze data, such as data mining, data farming and data warehouses. Data warehouses are used to store large amounts of data. This data is often used for On-Line Analytical Processing (OLAP) where short response times are essential for on-line decision support. One of the most important requirements of a data warehouse server is the query performance. The main focus of our research is finding adequate solutions to improve query response time of typical OLAP queries and improve scalability using parallel programming technique in a distributed environment. So the target of this study is to propose a design and implementation of a data warehouse in a distributed environment, using parallel programming technique. Using distributed environment, together with parallel programming is a good choice to increase the performance and to reduce cost. To show the advantage of the proposed strategy an application on banking system is given.

Key words: Distributed databases, data warehouse, parallel programming, OLAP, load balancing

INTRODUCTION

During recent years, there has been tremendous growth in the data warehousing market. Despite the sophistication and maturity of conventional database technologies, the ever-increasing size of corporate databases, coupled with the emergence of the new global Internet database, suggests that new computing models may soon be required to fully support many crucial data management tasks. In particular, the exploitation of parallel algorithms and architectures holds considerable promise, given their inherent capacity for both concurrent computation and data access. Data warehouses can be described as decision support systems in that they allow users to assess the evolution of an organization in terms of a number of key data attributes or dimensions. Typically, these attributes are extracted from various operational sources (relational or otherwise), then cleaned and normalized before being loaded into a relational store. By exploiting multi-dimensional views of the underlying data warehouse, users can drill down or roll up on hierarchies, slice and dice particular attributes, or perform various statistical operations such as ranking and forecasting. This approach is referred to as On-Line Analytical Processing (OLAP).

A large body of work exists in applying parallel processing techniques to relational database systems. The objective is to apply either inter-query or intra-query parallelism to improve performance. Query processing in parallel and distributed databases has been the focus of much research on the field (Apers *et al.*, 1983; Chen and Chengwen, 2000; Zhou and Williams, 1997). Although standardized relational algebra operators used in databases lend themselves well to parallelization, issues are raised concerning mainly join processing and communication overheads associated with both joins and other operation needs to exchange data (Pramanik and Vineyard, 1988; Shasha and Tsong-Li, 1991). Parallel Hash Join algorithms (Liu and Hao, 1996a) and Placement Dependency algorithms (Liu and Hao, 1996b) are typical proposals to reduce some of the overhead associated with join operations. Data placement has been another important issue in parallel and distributed database architectures (Hua and Lee, 1990) also reviewed in (Zhou and Williams, 1997). Partitioning the data for independent processing of fragments can typically reduce response time drastically but simultaneously increase total work, due to the need to exchange and merge data and partial results (Copeland *et al.*, 1988). For complex heavy join queries this can result in important speed

up deterioration. Data warehouses are precisely that kind of environment in which node partitioning can improve response time but also cause important extra overheads.

Many commercial and scientific applications manipulate voluminous data sets (Oldfield and Kotz, 1998; Kotz and Ellis, 1991; Poole, 1994) and in many case the input and output systems meet problems the same occurs in the processors. This question is best characterized when complex operations on a data base are executed.

Solutions that involve great amounts of data in centered systems make possible the implementation of centered Data Warehouse. In these cases they necessarily need processors and systems of storage of data with high performance. However, even with such systems the execution of complex searches can be very slow. Distributed Data Warehouse architecture reduces the systems needs especially high performance. This approach is still more advantageous if part of the data is remotely used and information is segregated according to some criteria, especially the regional criteria (Costa and Craveiro, 2001).

The proposed system, in this study, uses distributed processing data base concepts allowing the information processing and access in remote places through nets of high or low speed. It includes and can be applied together necessities. Either systems OLTP as OLAP can be adapted to use this solution.

Parallel programming using MPI, Message Passing Interface, are used to standardize the codification and to minimize the time of execution of the searches. The execution of these activities in distributed environment will prevent difficulties of performance of the complex searches. Such searches will become more efficient reducing databases or increasing numbers of processors.

OPERATIONAL AND DATA WAREHOUSE DATA

The operational data systems are characterized by processing information daily. They are used as storage for the commercial transactions such as inventory control, payment control and bank automation.

Data warehouse is used in systems planning, management systems and systems forecast. This knowledge is obtained from the operational data, transformed into information and later into knowledge.

Table 1 makes a comparison between data warehouse and operational data.

OLTP and OLAP: OLTP (On Line Transaction Process) Systems are used to support business operations of processes of an organization day by day. OLAP

Table 1: Data warehouse and operational data

Data warehouse	Operational data
Subject oriented	Application oriented
Integrated data	Limited integrated data
Non volatile	Permanent update
Permanent data	Current data
Ad hoc searches	Predictable searches

systems are used on decisions support. They provide administrators of an organization a multidimensional vision used to analyze the existing business-oriented profiles or to create conditions of analysis of existing standards of behavior. The following characteristics differentiate systems OLTP of systems OLAP:

- In OLTP systems the data are updated and shown in details, in OLAP systems the data are historical, summarized and consolidated from some operational bases, enclosing long periods.
- The size of the data stored in OLTP systems are ordered by sets of ten of Gigabytes, in systems OLAP hundreds of Terabytes (Poess and Raghunath, 2005).
- In systems OLTP the searches are simple like inserting, retrieving, updating. In OLAP the searches are complex, but using a standard model.
- The data model of data in OLTP is normalized, unlike OLAP.
- OLAP represents a set of projected technologies to support analysis and search ad hoc (Kimball, 1996).

Data mining and KDD process: The data mining is the most important phase in the process of transformation of the operational data in knowledge. This transformation is known as discovered of knowledge in data bases (KDD-Knowledge Discovery in Databases). The objective of the KDD process is to facilitate comprehension of standards to the people through interpretation of the existing data. What searching knowledge needs in the data bases is a consequence of the growth of the storage of the historical data. To use the advantage contained in this data set, they are organized in a easy form to identify standards that can help in the prediction of future actions. With the aid of statistical techniques a mechanism for such accomplishments is created (Michael and Gruenwald, 1999). It is possible to use some tools and techniques for the mining, being the most used in those databases based in consultation such as language SQL.

A data mining is the most important part of a process involving the discovery of the knowledge. The process of KDD involves other stages:

- Data warehousing
- Pre-processing
- Cleanness, selection and codification
- Enrichment
- Data mining
- Post-processing

DATA WAREHOUSE ARCHITECTURE

The data warehouses are especially dedicated and voluminous data bases containing integrated data from various independent sources, supporting customers whom desire is to analyze the data and to verify trends and anomalies. The analysis process is usually executed by operations like adding, filtering and grouping the data in a variety of ways (Neil and Quass, 1997). The data, stored in the Data Warehouses, are proceeding from systems that produce operational data. The operational data have a lot of information and are used in the daily operations of the informational systems. Using the processes of extraction, transformation and loading (ETL-extract, transformation and load) from the operational data, it gets the data for fulfilling the Warehouse. The Data Marts are subset of the Data Warehouses that group part of the information, generally applied to one determined purpose. Generally, they are data referring to a subject in particular, for example, selling, engineering, controlling or different levels of summarizing, such as, annual selling, monthly selling, 5 years selling and focusing on one or more specific areas. Data marts extract portions of Data Warehouses to the specific requirements of sectors or departments of enterprises. There are some approaches for the architecture of one Data Warehouse, among them are the following:

- Centralized data warehouse
- Virtual data warehouse virtual
- Distributed data warehouse

Centralized data warehouse: Centralized Data Warehouse is usually used when organizations or companies have a clear definition of the users needs access. The centralized data propitiate greater quality and integrity.

Virtual data warehouse: Virtual Data Warehouse provides access to the operational data for the final user to do searches of behavior knowledge or profile direct into the operational base. This solution restricts the types of search implementation but the bases are also used by the transactional systems. To facilitate this access layers

of software are created with the prescribed objective of the access to the bases shared with two purposes operational and informational. One advantage of this solution is the low cost; therefore it does not have duplicity in the storage of data. A disadvantage is that the complex searches are carried through in the operational base, being able to diminish the availability of this base, many of the data cannot be in the necessary form to be used by a final user, i.e., the development were for systems OLTP and not OLAP (Noaman, 2000).

Data marts: The Data Marts supplies the data of interest of a user, a sector or a department. This allows having more control of the part of the user in terms of requirements of data and manipulations. The data in the data marts are finer or of bigger interest for the user while data in the data warehouses are detailed. This characteristic makes possible for the Data Marts to be minor, increasing the performance of the search (Noaman, 2000). The disadvantage of this technique is that a global vision of the data does not exist.

Data warehouse and data marts: This data architecture is a combination of the centralized data warehouse and the Data Marts. It gets advantage of the two solutions: complete integration of the data warehouse and the optimized access of the data marts. They supply the data of interest on a user, sector or department allowing bigger control of the user in terms of requirements of data and manipulation.

Distributed data warehouse: The proposal of Inmon, Fig. 1, for implementation of the distributed data warehouses (Inmon, 2002) mentions the existence of central and local data warehouse, where the data are mutually exclusive. The extraction and loading in this architecture are also being distributed.

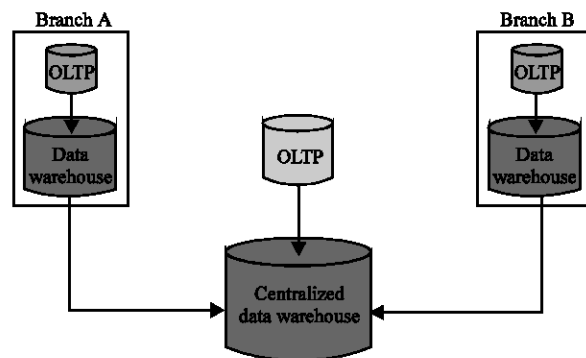


Fig. 1: Distributed data warehouse (Inmon, 2002)

A PROPOSED DISTRIBUTED DATA WAREHOUSE ARCHITECTURE

The architecture proposed claims to distribute the Data Warehouses in distant points receiving the load from the data of local business systems. From a central point programs are distributed to be executed in the distant points. The result of this processing is returned to the central point and is consolidated, being stored in a posterior database for consultation. The considered system, for example an application to a banking environment, allows the creation of behavior reports of the customers. The information explored in the Distributed Data Warehouses is sent to the central system for consolidation and to provide a global vision of results. The use of Data Warehouse allows extracting information of the logs registers of the transactional systems, OLTP, for posterior verification. These analysis are important for the knowledge of the behavior of the customers in several aspects.

It is possible to forecast the type of product to be offered to one determined customer, being based on information as sex, age, income, geographic profile and economic profile. Moreover, there is information of the relationship of the customer with the institution whose canals of contact are used, what frequency, which products the customer already possesses, which type of claims the customer has already effected, etc.

Based on these information it is possible to forecast the propensity use, of purchase and the behavior proper to the customers. It is also possible to identify customers with similar profiles, where the results of the actions will get success with minor effort. Proposed technique architecture presents similarities with the proposal of Inmon (Inmon, 2002). The main difference is that the proposal is not centralized. The operational data are converted and stored into local data warehouse through techniques of extraction, transformation and loading. In the central point the requests to the data warehouses are really distributed by the agencies (Fig. 2). These requests are dealt with through execution of the searches in the processing of the branches, being the results returned to the central point, to be consolidated and stored. With the objective to diminish the time of replying to similar requests that are carried out by other users, a data base in the central point for storage of the result of the searches is created, or the same of the gotten reports. These results will be consolidated for the use in the central point, being able to be consulted by the remote points, through a portal.

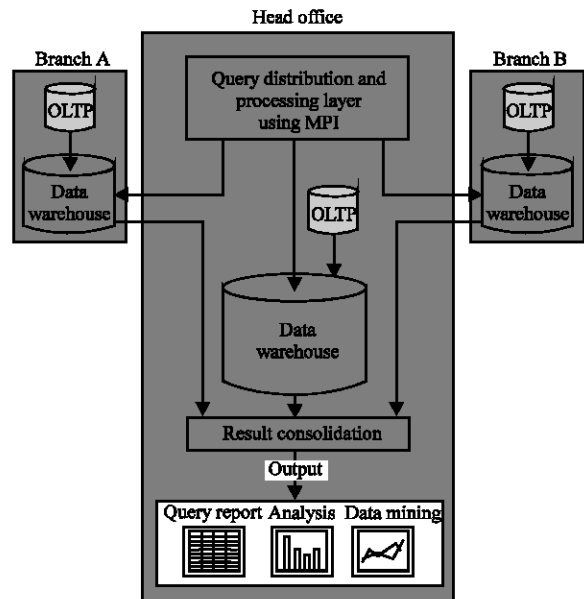


Fig. 2: Proposed architecture

The central point data base is used to store the reports of the searches temporarily but not being organized as one data warehouse. The period of storage can be very small, for example, one month, therefore an old report will be requested and a new search in the warehouses distributed data is affected in the way that then the stored volume is low. At the central point, optionally, it could have one data warehouse, not for centralization of the data purpose, but only for local use of the branches.

MESSAGE PASSING INTERFACE SOLUTION

One method of parallel programming computation is the use of a messages ticket library. This library transfers data between instances of programs being processed using multiple processors. With this method it is possible to have available great space of memory and a bigger number of processing central units. Therefore it is possible to solve problems of high complexity that are not solved normally with traditional methods (Gropp *et al.*, 1996).

In the proposed solution the implementation of parallelism of tasks is done through the paradigm master slave (Fig. 3).

The programs, using the message passing interface library, are composed of programs with multiple instances that can be communicated through the calls of the library. These calls are divided in four classes:

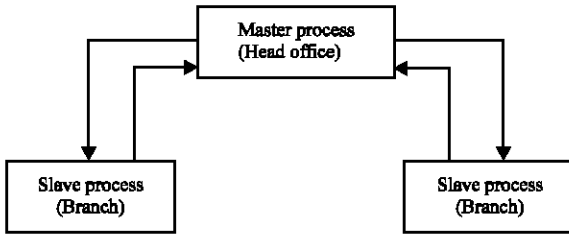


Fig. 3: MPI process

- Initiation, Management and End of communication.
- Communication between pair of processes
- Communication operations among several processes.
- Data types creation.

The communication is made by using message passing. Pair of processes communicates by using commands send and receive. Generally, the message is composed of an envelope indicating the source and the destination is a body with data to be transmitted. The master process is executed in the central point. It distributes tasks for the enslaved processes sending programs to be processed. Each slave executes its task returning the result. The master, then, makes the consolidation mounting the reports. The programs are written in language C using library MPI (Message Passing Interface) (Quinn, 2003) for the ticket of the messages and distribution of the code between the slaves. It also used on a library SQL for the execution of the necessary commands for access to the data banks. Either OLTP or OLAP systems can be adapted to use the solution that combines the use of distributed data bases with the ticket of messages programming.

A CASE STUDY ON BANKING SYSTEM

As a case study it was chosen a presentation of a bank application, having involved concepts of CRM (Customer Relationship Management). This application involves the Data Warehouses located in the branches. At headquarters there is the point of distribution of the programs to be executed in all the branches. The information is extracted of the transactional systems and is loaded in the Data Warehouse, one in each branch, obeying a model of data that allows through exploration of data the generation of knowledge for boarding to the customers. With this commercial law action they could be carried through in the scope of a branch, of a group, or the same, involving all the branches of the bank.

The model of data, in the applications of Data Warehouse, is one of the most important points for the success of the implementation of the project. In systems

Table 2: Data dictionary

Table	Description	No. of rows
SCPF	This file holds all accounts	2,045,026
S1PF	This file holds details of all credit rate changes that have been entered against an account.	567213
S2PF	This file holds details of all debit rate changes that have been entered against an account.	406764
C8PF	This file contains currency codes, descriptions and mnemonics.	171
C5PF	This file contains account type information.	285
C4PF	This file stores customer type codes and text.	69
CAPF	This file holds the name, address and processing rules for each branch.	5

Table 3: Results CPU time (sec) using aggregate and joins functions

CPU time (sec)	Sum	Avg.	Count	Max.	Min.	6-Join	G/O by
1-server	29	31	27	28	28	58	63
5-servers	6	7	6	6	6	9	11

with the technique vision only in its implementation the technical project can be a success under the performance, availability and security aspects, but not in the business-oriented aspects where it finishes being a failure and with little usability for the users. Therefore it is important to think about two aspects: technical and business one.

Each branch has its data warehouse that is loaded with the data proceeding from the transactional systems obeying a data model. The model is identical for all the branches. The historical information is stored in the Data Warehouse, once stored it is possible use tools to explore information in local searches.

The global searches using all branches Data Warehouses are made from the master unit through the use of message interface to send commands SQL for all the agencies. After the accomplishment of the requested operations for the master the results are returned for consolidation having generated reports by itself that are stored in the data base specified in the proposed architecture.

This frequent research is requested by the users. In this case, the results are available in the master, being able to be consulted through a vestibule, where it is not necessary to use any tool of exploration.

In this application the model of data and the data dictionary are showed in Fig. 4 and Table 2, respectively.

We have made some experiments comparing the performance using one and five computers with the same characteristics. The results with one computer (1PC) simulate the centralized data warehouse and 5PCs is data warehouse striping applied to five computers. The results of Table 3 shown the case of using any subset of the aggregate (sum, Average, count, max and min) and Join (Join, Group and Order by) functions.

From the Fig. 5 we measure an average gain of 4.71 times between using one computer (1 Branch) and using five computers (5 Branches). This near optimal speedup is explained by the fact that the partial queries only access a small part of the fact table.

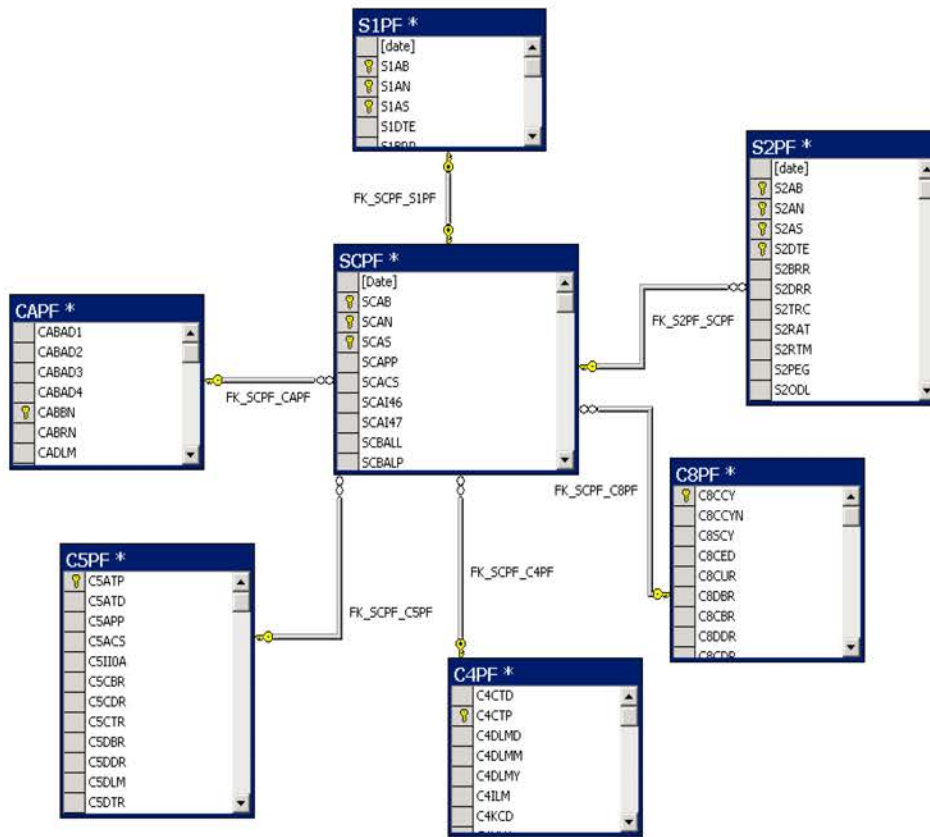


Fig. 4: Integration of the architecture proposal to the conceptual model of data, showing the main elements of the solution

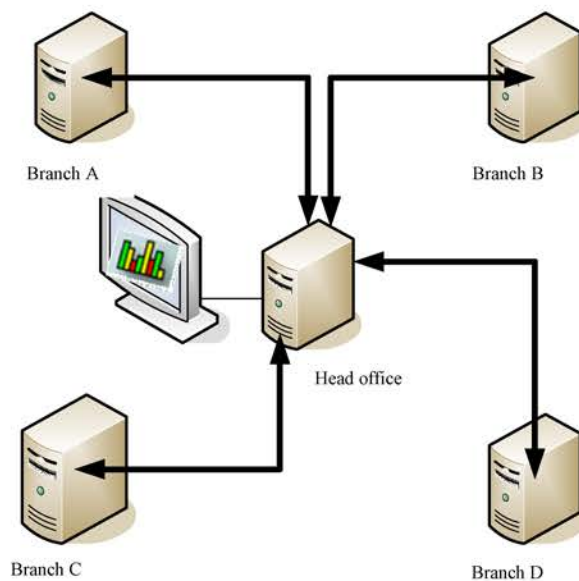


Fig. 5: Application on banking architecture

CONCLUSIONS

The proposed architecture uses distributed data bases driven from a central point and it is generic to be able to apply it in many environments such as information and Data Warehouse. The parallel programming standardizes the codification and minimizes the time of execution for determined searches. The system makes possible consolidated vision of the information possessing distributed databases that is done through the publication of the main searches in portal. As future work, it can be studied the aspects of performance under the point of view of the net of communication between the master and the branches, as well as, implementation of a tolerant system to the imperfection.

REFERENCES

- Apers, P.M.G., A.R. Hevner and S.B. Yao, 1983. Optimization algorithms for distributed queries. *IEEE Trans. Software Eng.*, 9: 57-68.
- Chen, H. and L. Chengwen, 2000. An efficient algorithm for processing distributed queries using partition dependency. *International Conference on Parallel and Distributed Systems, ICPADS*, pp: 339-346.
- Copeland, G.P., A. William, E. Boughter and T. Keller, 1988. Data Placement in Bubba. *SIGMOD Conference*, pp: 99-108.
- Costa, N. and J. Craveiro, 2001. Considerations about the integration of a database and an its data warehouse on a system of parallel files: Ph.D Thesis, Polytechnic School, Sao Paulo, Brazil.
- Gropp, W., E. Lusk, N. Doss and A. Skjellum, 1996. A high-performance, portable implementation of the MPI message-passing interface standard. *Parallel Comput.*, 22: 789-828.
- Hua, K.A. and C. Lee, 1990. An adaptive data placement scheme for parallel database computer systems. *Proc. VLDB Conf.*, Brisbane, Australia.
- Inmon, W.H., 2002. *Building the Data Warehouse*. John Wiley and Sons.
- Kimball, R., 1996. *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*. John Wiley and Sons.
- Kotz, D. and C.S. Ellis, 1991. Practical prefetching techniques for parallel file systems. In: *Proceedings of the 1st International Conference on Parallel and Distributed Information Systems*, pp: 182-189.
- Liu, C. and C. Hao, 1996a. A hash partition strategy for distributed query processing. *International Conference on Extending Database Technology (EDBT)*.
- Liu, C. and C. Hao, 1996b. A heuristic algorithm for partition strategy in distributed query processing. *ACM Selected Areas in Cryptography SAC*.
- Michael, G. and L. Gruenwald, 1999. A survey of data mining and knowledge discovery software tools. *ACM SIGRID.*, 1: 20-33.
- Neil, P.O. and D. Quass, 1997. Improved query performance with variant indexes. In: *Proceeding of the ACM SIGMOD Conference*, Tuscon, Arizona.
- Noaman, A.Y., 2000. Distributed data warehouse architecture and design. Ph.D Thesis, University of Manitoba, Canada.
- Oldfield, R. and D. Kotz, 1998. Applications of parallel I/O. Technical Report PCS-TR98-337, PCS-TR96-297, Computer Science, Dartmouth College.
- Poess, M. and K. Raghunath, 2005. Large Scale Data Warehouses on Grid: Oracle 10 g and HP Proliant Servers. *Proceedings of the 31st VLDB Conference*, Trondheim, Norway.
- Poole, J.T., 1994. Preliminary survey of I/O intensive applications. Technical Report CCSF-38, Scalable I/O Initiative, Caltech Concurrent Supercomputing Facilities, Caltech.
- Pramanik, S. and D. Vineyard, 1988. Optimizing join queries in distributed databases. *IEEE Trans. Software Eng.*, 14: 1319-1326.
- Quinn, M.J., 2003. *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill Science.
- Shasha, D. and W. Tsong-Li, 1991. Optimizing equijoin queries. In: *Distributed databases where relations are hash partitioned*. *ACM Trans. Database Syst.*, 16: 279-308.
- Zhou, S. and M.H. Williams, 1997. *Data Placement in Parallel Database Systems*. Parallel Database Techniques, IEEE Computer Society Press.