DISTRIBUTED DATA SHARING SYSTEMS

Edited by
R. P. Van de Riet
and W. Litwin

DISTRIBUTED DATA SHARING SYSTEMS

Proceedings of the Second International Seminar on Distributed Data Sharing Systems held in Amsterdam, The Netherlands, 3-5 June, 1981

edited by

R. P. VAN DE RIET
Vrije Universiteit
Amsterdam, The Netherlands

and

W. LITWIN
INRIA
Paris, France

1982

NORTH-HOLLAND PUBLISHING COMPANY
AMSTERDAM • NEW YORK • OXFORD

ISBN: 0 444 86374 5

Published by:
NORTH-HOLLAND PUBLISHING COMPANY – AMSTERDAM • NEW YORK • OXFORD

Sole distributors for the U.S.A. and Canada:
ELSEVIER SCIENCE PUBLISHING COMPANY, INC.
52 Vanderbilt Avenue
New York, N.Y. 10017

PRINTED IN THE NETHERLANDS

Second International Seminar on
Distributed Data Sharing Systems
Amsterdam, The Netherlands, 3-5 June, 1981

Organized by

Department of Mathematics and Informatics,
Vrije Universiteit, Amsterdam
and
Institut National de Récherche du Informatique
et Automatique (INRIA), Paris

under the sponsorship of

IGDD

Chairman
Reind van de Riet

Co-Chairman
Witold Litwin

Secretary
Peter Apers

NORTH-HOLLAND PUBLISHING COMPANY
AMSTERDAM • NEW YORK • OXFORD

PREFACE

The Second Seminar on Distributed Data Sharing Systems was held at the Vrije Universiteit, Amsterdam, from June 3 to June 5, 1981. From the papers presented, a selection was chosen by the Program Committee and published in this book.

The Workshop dealt with problems which have to be solved when data base systems are automatically communicating with each other by means of (public) computer networks. These problems are of a fundamental character when we ask how a distributed data base system (DDBMS) can be defined, they are of a practical character when we ask how to recover from a crash or how data integrity can be maintained or where to process data base queries such that response time or transportation cost is optimal.

The Workshop was split into two parts. During the first part, six small-sized working groups, led by chairmen, met around the following themes:

- Concurrency control and interprocess synchronization (G. Schlageter);
- Crash recovery (E. Holler);
- Query processing, resource allocation and distributed computing (J.S. Riordon);
- Models and architectures of DDBMS's (J.C. Chupin, R. desJardins);
- Security and privacy and semantic integrity with regard to DDBMS's (S. Miranda).

In these working groups 47 position papers were presented and discussed. The sessions were oriented more towards discussion than to presentation. The proceedings of all the position papers have been published by the Vrije Universiteit.

The second part of the seminar consisted of plenary sessions. During these sessions chairmen gave overviews of the discussion in their groups. Chairmen reports gave rise to the plenary discussion about the state-of-the-art of the domain and about future trends.

The formula of the conference, small working groups and plenary sessions, turned out to be quite successful. This was primarily due to high quality and fast work of the chairmen. It was felt, however, that the time for the plenary sessions (3.5 hours) was not enough.

In addition to the two-page position papers, the participants were asked to submit full papers. This book contains a selection of these full papers and the chairmen reports. Most of the papers are research papers, proposing new solutions to the main problems which one encounters when designing DDBMS's. One paper (S. Spaccapietra) is a tutorial paper devoted to the comparison of some actual research projects implementing DDBMS's. A few papers describe DDBMS's which are implemented for a given practical application. Finally, one paper (T. Kitagawa) points out the fascinating perspective of the automatic production of knowledge from the enormous mass of information becoming available through DDBMS's to come soon. We feel that this choice of papers should render the book interesting not only to researchers, but also to professionals and users concerned about distributed data base systems.

We wish to express our thanks to Peter Apers, who organized the workshop, to the chairmen for their efforts during the workshop and for their reports, and to the participants for their position papers, their full papers and their cooperation in the workshop.

The list of members of the Program Committee, which consisted of the chairmen and three well-known authorities, together with the list of referees who assisted the Program Committee, are presented on the next page.

Reind van de Riet
Vrije Universiteit
Amsterdam

Witold Litwin
INRIA
Paris

PROGRAM COMMITTEE

J.C. Chupin (*France*)
S.M. Deen (*U.K.*)
R. desJardins (*U.S.A.*)
E. Holler (*F.R.G.*)
W. Litwin (*France*)
S. Miranda (*France*)
R.P. van de Riet – Chairman (*The Netherlands*)
J.S. Riordon (*Canada*)
F.A. Schreiber (*Italy*)
G. Schlageter (*F.R.G.*)
H. Weber (*F.R.G.*)

List of Referees

H. Breitweiser (*F.R.G.*)
K. Dittrich (*F.R.G.*)
G. Eizenberg (*France*)
D.D. Falconer (*Canada*)
M. Leszak (*France*)
S. Mahmoud (*Canada*)
K.C. Toth (*Canada*)
P. Wilms (*U.S.A.*)

TABLE OF CONTENTS

1.
TUTORIAL

DISTRIBUTED DATA SHARING SYSTEMS
R.P. van de Riet, W. Litwin (Editors)
North-Holland Publishing Company

DISTRIBUTED DBMS ARCHITECTURES

Stefano SPACCAPIETRA

INSTITUT DE PROGRAMMATION
Université Pierre et Marie Curie (Paris 6)
4, place Jussieu
F- 75230 Paris Cedex 05

This paper is intended to present the main ongoing research projects developing a distributed data base management system. The presentation is based on a feature analysis schema developed by the AFCET working group on distributed data bases. The following topics are dealt with : data base design, data distribution, system architecture, query processing and consistency. Note : This text is an updated translation of the document which has been distributed as written support for the tutorial on distributed data base management systems architectures given at the International Symposium on Distributed Data Bases held in Paris, March 11-14, 1980.

1. INTRODUCTION

Throughout the seventies distributed processing has shown itself to be one of the most promising lines in the development of computer systems, perhaps even leading to the development of a new approach to computer usage. This is due as much to the availability of reliable communication networks between geographically-distributed computers as it is to the tremendous boom in mini and micro-computers; these have jointly introduced computing into all sectors of activity, including the leisure industry and even into our own homes.

In the world of research this line of thought has attracted numerous research teams. Amongst these, a certain number have devoted themselves to the development of one particular type of distributed system : distributed data base management systems (DDBMS). As we can see from the name, a DDBMS is a software system which maintains a data base where the data is distributed amongst subsets (called local bases) stored at various sites, geographically-dispersed but interconnected by a communication network. The users themselves (amongst them the application programmers), are distributed amongst the various sites of the network. To effectively fulfill its purpose the DDBMS has accordingly to be realised in the form of a set of components distributed and/or duplicated on the different sites contributing to the running of the system.

This type of system is intended to respond to the needs of organisations made up of many cells, these being relatively independent as regards completing their tasks and acquiring their data, namely : banks, chains of stores, railways, geographically-dispersed businesses etc. Equally, this type of system particularly concerns aggregates of organisations, nonetheless independent, set up with a common objective in mind. They are : multinational companies, international bodies, many-sectored businesses, cooperating bodies (for example, libraries which join together to give their users a better service), etc...

These potential needs are sufficiently important for two countries to have already begun research on the national level. First of all in France where the pilot project SIRIUS, directed by INRIA, a detailed description of which is to be

found in [12], was set going in 1976 ; several prototypes within this project have already been implemented (notably ETOILE, FRERES and POLYPHEME) and a complete DDBMS is currently in the development stage (SIRIUS-DELTA). Then in Italy, where in 1979 the Consiglio Nazionale delle Ricerche launched the DATANET project which united around ten research teams.

To this coordinated research may be added other independent, yet nonetheless important, research being undertaken principally in the Federal Republic of Germany (DISCO, ESA, POREL and VDN projects), the United States (INGRES, R* and SDD-1 projects), Canada (project ADD) and in the United Kingdom (PRECI project).

2. OBJECTIVES OF THE AFCET WORKING GROUP

Given that suggestions and ideas abound it is difficult today, for the non-specialist, to easily extract a synthetic knowledge of this field of study. Each project has its own initial hypotheses, objectives and solutions, as well of course as its terminology. Finding a common lead in this stockpile of facts and then actually synthesising all research currently being undertaken : such was the objective adopted in 1978 by the "Distributed Data Bases" working group, set up in November 1975 within AFCET. The means to this end developed by the group is a questionnaire in which we express a problem by posing various sorts of solutions that we have identified. This is how the different aspects of a DDBMS have been gone through : data base design, distribution, system architecture, processing of requests and consistency. To complete the study of the DDBMS itself we have added two chapters which endeavour to examine the data-processing and organisational context the DDBMS supposedly lies whithin. These two chapters, however, are at the moment of limited application, bearing in mind that DDBMSs are currently being developed outside the context of usage.

Once the questionnaire, a sort of feature analysis schema, is set up the various proposals for a DDBMS may then be studied. The responses thus collected enable us to describe the characteristics of a DDBMS. It is then easy to compare two DDBMSs in as much as they can be compared. The questionnaire and the results obtained from its use should enable the following objectives to be realised :

1) To offer to those wishing to understand DDBMSs an analysis of the systems and of the state of the art in the subject.
   Moreover, the questionnaire at times goes beyond the current state of the art and examines certain aspects which have not as yet been tested by means of a prototype.

2) To give to those wishing to design a DDBMS a catalogue of the problems they will have to face together with some proposed solutions ; in a sense it acts as a design aid.

3) To give to those who want to use a DDBMS, and who are trying to make up their mind about a particular system, the means to quickly select one which corresponds to their needs which are also identified with the help of the questionnaire.

4) Lastly, to show, with a study of how the responses are distributed, current tendencies in DDBMS implementation.

We should point out that as it is still very much a tool, and not the final version, the questionnaire may be relatively sterile. Nevertheless it suffices in so far its scope as the responses are gathered directly by the group, through discussion with the team developing the DDBMS, or, if this is not possible, by studying relevant publications. The present version has so far been used when studying the projects SDD-1, DISCO, POREL, VDN, POLYPHEME, SIRIUS-DELTA and partly R*.

Before presenting the significant results which the work has already produced, and to make them more understandable, we shall firstly illustrate the general way in which a DDBMS operates : this ought to enable the reader with limited specialised knowledge in this field to familiarise himself with the objet of the study.

We shall then briefly introduce the different projects before relating them aspect by aspect to the questionnaire.

3. SCHEMA ARCHITECTURE IN A DDBMS

Before it may be used, a distributed data base (DDB) must be set up. That is, the schemas which describe the different views of the DDB must be defined and the data must be loaded. The alternatives which even now are suitable for this level of design will be discussed later (§ 6.2). Here we would simply like to show the hierarchy of the schemas and each of their roles in the most general case. In order to do this we use the well-known concepts defined in [3] . We refer to this hierarchy in the following paragraph. In going from users to data, a request, as it progresses, will be faced with (*cf. figure* 1) :

1) an external schema (DES). This describes the data of the DDB concerning the application in which the requesting user is operating. This schema helps maintaining program/data independence ;

2) the conceptual schema (DCS), describing intrinsically the data of the DDB. It acts as a central reference schema ;

3) the partitioning schema (DPS) of the DDB. This describes a logical partitioning where each part will be entirely stored in one (or several) local bases(s). This schema enables the distribution to be better tailored according to users's needs ;

4) the distribution schema (DDS) of the DDB. This describes the distribution in the local bases of the parts defined in the DPS. It enables the location of each data item to be known. At this point the request can be decomposed into sub-requests, each one referring only to data belonging to the same local base.

Then for each local base implied by the request :

5) the local conceptual sub-schema (CSS_i) containing for each local base the DCS descriptions for all the data stored in this local base. This schema allows a sub-request transmitted from one place to another inside the DDBMS to be interpreted when it arrives. This schema can obviously be replaced by a complete copy of the DCS ;

6) the local external schema (LES_i). This describes the local base in the local formalism. This schema enables the DDBMS to behave as if it were a normal user of the local DBMS which manages the access to the local data ;

7) the local conceptual schema (LCS_i), specific to the local DBMS ;

8) the local internal schema (LIS_i), specific to the local DBMS.

In practice this architecture is more often realised in a simplified form (cf. figure 2) : the DCS, DPS and DDS are combined into just one schema called the global schema (GS) ; the homogeneity of the local DBMSs enables not to use the CSS_i ; the local DBMSs themselves only possess a single schema (LS, Local Schema) which acts as LES, LCS and LIS ; lastly, the DESs are sometimes not realised, the users referring directly to the data of the DCS-GS.

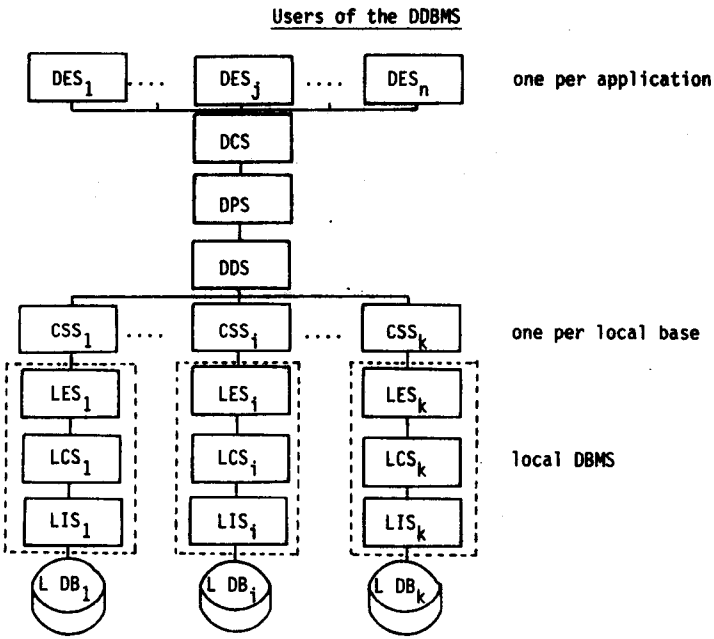

Fig. 1 : Schema architecture of a DDB

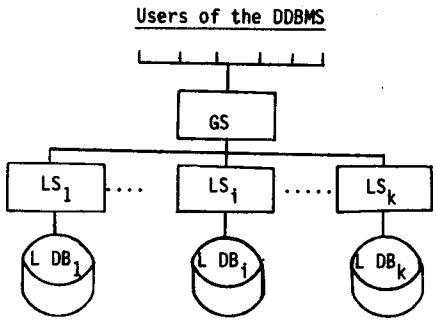

Fig. 2 : One possible implementation of the schema architecture.

4. GENERAL VIEW OF DDBMS OPERATION

Having defined the schemas we can now outline the general principles behind the operation of a DDBMS. In order to do this we will follow a user request as it progresses through the system, from its formulation to its result. At the starting point there is the user, who, unaware of the distribution, enters into dialogue with the system. The dialogue, whether it is programmed or at the terminal, consists of a certain number of requests for manipulation of data. In order to safeguard the consitency of the DDB whilst at the same time allowing its manipulation, the requests are grouped into semantic units called transactions : within a transaction the DDB's consistency may be temporarily destroyed but its state must on all accounts be checked at the end of every transaction and consistency verified, otherwise the transaction and all its effects are deleted. Since the definition of transactions is done on a semantic basis, it is up to the user to define them. A user dialogue therefore takes the form : start transaction$_1$, request$_{11}$,... request$_{1n}$, end transaction$_1$, start transaction$_2$, request$_{21}$, etc...

At the beginning of a transaction, the DDBMS sets about identifying the transaction (either by numbering or by another technique). According to the way it has been realised the DDBMS also sets about allocating the (implicitly or explicitly) requested data in order to carry out the transaction. When several copies of the requested data exist the allocation process may include a choice of the copies to be attributed to the transaction.

If there have been update requests during the transaction, at the end of the transaction the DDBMS first proceeds to consolidate updates, i.e. to effectively integrate into the DDB the updates which had until then been wisely put aside. Then the de-allocation of the data allocated to it takes place before the transaction disappears leaving nothing but its memory in the system's archives.

Whilst in the process of transaction, a request is firstly checked in relation to the external schema (DES) to which it refers : syntactically and semantically (integrity constraints,...). It is then translated into the internal language (called the pivot language). These tasks are entirely analogous to what takes place in a standard DBMS. The same is true for the checks which take place in the next phase, in relation to the conceptual schema (DCS). The access rights (with respect to privacy) and the preliminary allocation of the data are also controlled at this level (the latter not so in DDBMSs with dynamic allocation, in which case the data allocation takes place straightaway).

If the request proves correct it may be executed. It is then processed by determining both the partitions which are involved (information from the DPS) and their locations (information from the DDS).

Everything is now ready for the request to be decomposed into a group of sub-requests, each one only containing references to data of a single local data base, thus enabling it to be fully executed on just one site. The execution of a sub-request will generate a partial result which works towards establishing the final result. The sub-requests have to be chosen so that the processing develops globally at the least expense. The expense in question covers the data transfers through the network which are much more costly than traditional transfers between secondary memories and the central memory. It is for this reason that several researchers have devoted themselves to the search for an optimal algorithm for decomposing requests.

Note that additional sub-requests intended for checking the consistency of the date base, and, if the case arises, for maintaining the consistency of the multiple copies of one data item, have to be included in the decomposition.

The result of the decomposition will be equivalent to a program which contains :