FACTA UNIVERSITATIS (NIŠ)
Series: Electronics and Energetics vol.8, No.2(1995), 183–190

COMPUTER WAN WITH LOCALLY TRANSPARENT USE OF RESOURCES

Wilfried Haas and Wolfgang Weber

Abstract. Local transparency means that the location of distributed components in a computer network are hidden for the user of the net. All available components are resources or elements of the network-connected computers. The access to such distributed elements is realized in network operating systems and distributed operating systems via the client/server organization model. This model subdivides the network programs into instances which initiate the executions (client program part of the network program) and into service offering resp. executing partner instances (server program part of the network program). The access to distant elements (elements of another computer) is not realized directly and requires an ordering relation between local client process and distant server process.

1. Definition of Objects

An important aspect for the access to the elements distributed in the network is their clear notation. Their identifiers are within context of the client/server ordering relation unequivocal. Regarding this fact it is not the purpose of the development to offer a global name space for the notation of all possible elements. But of importance is the unequivocal identification of the offered services and the localization of the corresponding server processes allover the network. Such a system represents a service administration instance, for the client/server model. Its administrated objects are the functions declare as services which can be applied to the elements of a computer. Such objects are best described and can be identified by their special functional attributes. Such a form of identification is not a suitable method for a system of a locally transparent use of distributed components

Manuscript received June 15, 1995.

Dipl. Ing. W. Haas is with Ruhr - University Bochum, 44721 Bochum, Germany. Prof. Dr Dr hc. W. Weber is director of Institute for Computer Science, Ruhr - University Bochum, 44721 Bochum, Germany.

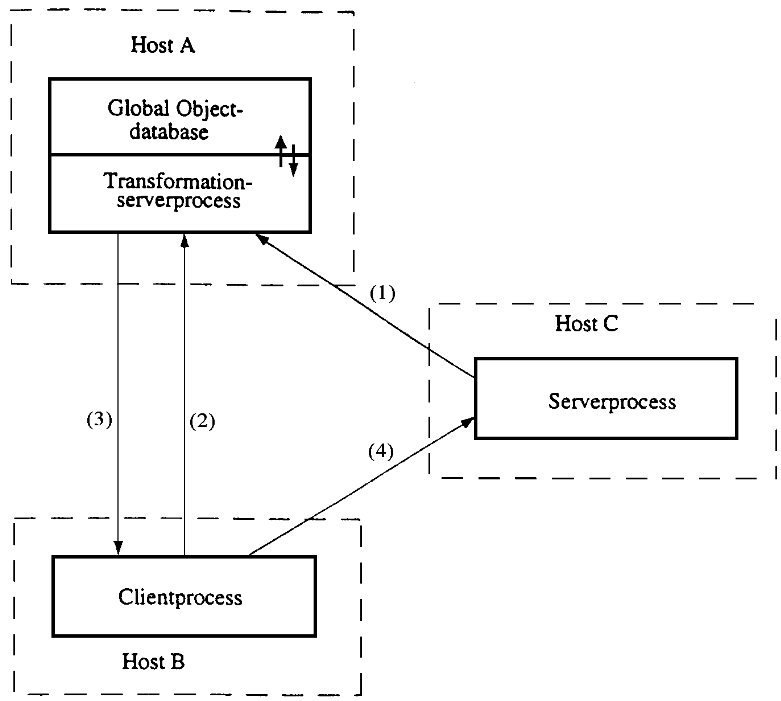

Fig. 1. Centralized lookup service

of computer network. Instead of it usually globally unequivocal identifiers - representing the functional attributes - are used for the identification.

2. System Functions

The main system function is the transformation of the identifiers. The transformation produces the addresses of available server processes. To provide this system function a relation between the identifier marking the service and the location of the respective server process must be produced and stored for future demands. This process is named registration and initiated by the start of the server program.

3. The Localization of Objects

From the above named steps two elementary questions for the system design arise:

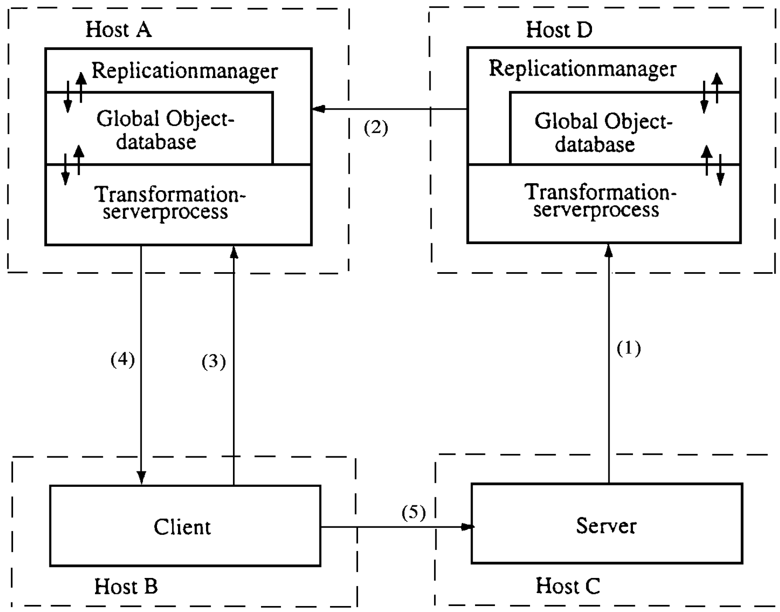

Fig. 2. Centralized lookup service with replicated database

- Shall the relations needed for the transformation be stored in a central or in a distributed relationship table?
- Shall from a logical point of view a centralized service execute the transformation or all connected computers in the distributed system which are anyhow responsible for the administration of their functional objects?

The usual classical methods for the localization of objects/base on the fact that there are one or several computers in the network (socalled transformation servers) which are offering this service. Prior to the addressing of the server process of an identified service (realizing the access to a distant resource) the transformation servers are examined for suitable addresses. In the case of a single transformation server we have the model of a central object administration with one central relationship table (Fig. 1). To increase the availability and reliability of the transformation service mostly several transformation servers - each with a complete copy of the relationship table - are connected for the network (Fig. 2).

A third variant of proved methods is the use of contextual 1 partitioning hierarchical structured identification mechanisms which means that parts of the relationship table are distributed to several transformation servers of the network. Each transformation server informs only about the relations which are stored in his part of the data bases (Fig. 3).

All these methods are centralized solutions independent from the question if the relations are distributedly or centrally stored. Examples are the global *Location broker* of the NCA (totally replicated association tables) and the *clearinghouse* (contextual partitioning hierarchically structured identification mechanism). The disadvantages of such solutions are either low availability and reliability or high costs for the consistency maintenance of replicated relations.

4. Characteristics of System Design

The here proposed system now does not need explicitly named instances for the administration of the associations between unequivocal service identifiers and the addresses of the executions locations. Each computer sharing the system is responsible for the administration of the services. It offers and has to prepare the functions of registration and transformation. This functions are realized by a background process - the so called object manager - available on each sharing computer system. So the disadvantages of centralized procedures are avoided. The results are the following positive properties of the distributed administration:

- High efficiency if prior to the service access no communication with an additional transformation server is necessary.
- No uncertainty regarding the availability of the service if the access can be realized directly to the object administration of the addressed computer.
- No problems in the maintenance of consistency because no distanced transformation server must be informed about the object registration or object deregistration.

For the distributed administration the computer network must offer a communication form which allows the distribution of an information to all receivers. This communication procedure is named *broadcasting call* and is not assisted by all data transfer devices. In this case broadcasting calls must be simulated by repeated sending of the same information to the receivers. As a result there arise high communication costs. To solve these problems new solutions must be found. The following methods try to minimize the communication costs.

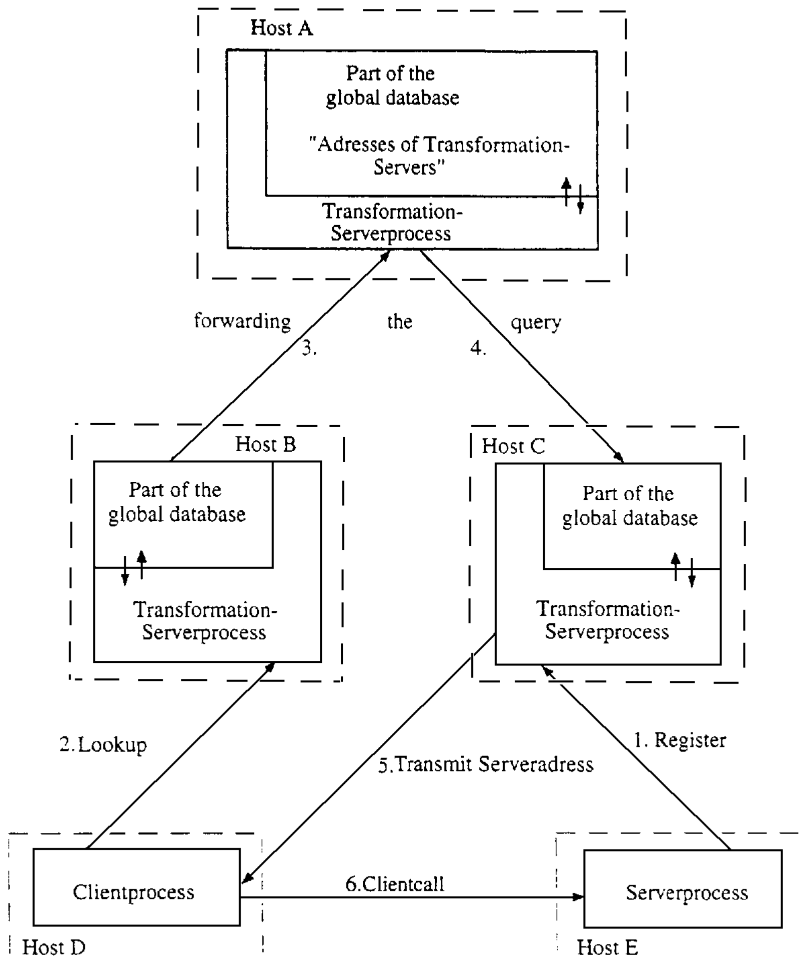

Fig. 3. Logically centralized lookup service with several distinct transformation-servers

In short the following basic ideas can help to limit the data stream:

- Only those selective broadcasting calls are admitted which base on receiver lists defined by the sender
- Only event driven information transport is admitted.

A selective broadcasting call is an information flow to a certain receiver group. In the presented solution the decision which information value a message represents for a receiver is left to the sender. This means that a message will not be sent to all possible receivers and leaves it to them to perform the selection regarding content or group specific aspects. (With such a procedure the amount of data flow would not be reduced). Instead of that the sender of a message prepares an associated receiver list. For each restored relation a separate receiver list is associated. The receiver list contains either the addresses of service users or all service interested receivers. If the addresses stored in the receiver lists contain the locations of clients which already have used successfully a service, so this knowledge can be applied in case of a deregistration of the server process. If a demanded service could not be executed, this also can be stored as service demand or service interest and furtheron be used for a future registration of a service executing server process informing potential other interested stations about the availability of the service. Additionally stored service interests can implicitly inform about stations where further informations about the locations of service ready server processes are available. So certain messages are only sent to addresses for which the message contains processable informations.

By a second design criteria for the minimization of communication amount must be defined which are the events relevant for messages. For the developed system/client - initiated service interests and service preparations associated to such already formulated service interests are the two only valuable events. Cyclic messages propagating the availability of registrated service offers are not admitted.

In the named context both design criteria serve as integration supplements to limit the data flow in the network. So the method of distributed objects administration disposes of basic properties admitting its application in big WAN with many connected computers. Such networks normally result by the connection of LAN where the networking media are not prepared for broadcasting calls.

5. Local Architecture

Figure 4 explains the architecture of the object manager realizing the above demanded properties on the system design. The main part of the object manager is a data base serving as a storage for relevant events in the form of relations. Besides the relations describing registrated server processes (identifiers point to port numbers) and besides relations for the identification of distanced located execution addresses (identifiers point to service offering addresses), also port number bound identifiers can point to service demanding addresses and so-called not bound identifiers can point to addresses of interested instances. In this case the addresses of interested users describe the location of the clients which have already announced the demand for a service indicated by the identifier.

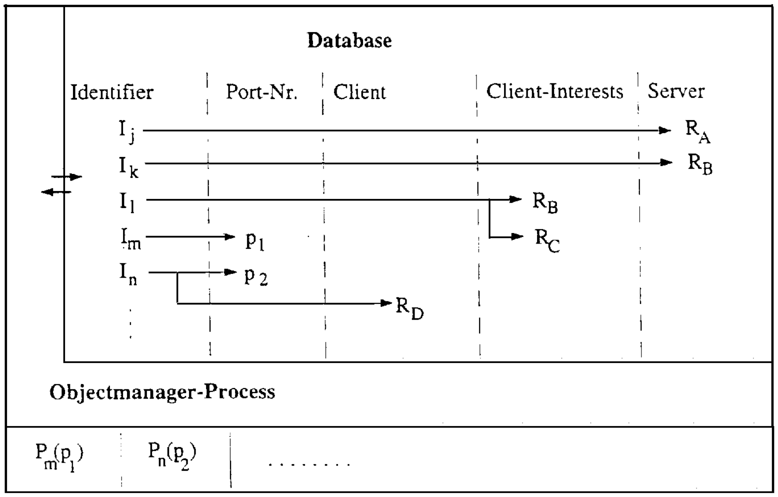

Fig. 4. Structure of the objectmanagement-daemon.
Remarks: P_m and P_n are server processes bound with the local communication adresses p_1, and p_2.
$R_A \ldots R_D$ are network adresses of hosts $A \ldots D$.

REFERENCES

1. M. ACCETA: *Resource location protocol.* Network Working Group, RFC887, December 1983, pp. 16
2. K. BIDLINGMAIER, P. PAWLITA: *DCE – distributed computing environment: meilenstein in der verteilten verarbeitung.* Zeitschriftenaufsatz: Datacom, Band 8 (1991), pp. 74–76
3. D.R. CHERITON, T.P. MANN: *Decentralizing a global naming service for improved performance and fault tolerance.* ACM Trans. on Computer Systems, Vol. 7, No.2, May 1989
4. IXOS SOFTWARE GMBH: *DCE remote procedure call.* GUUG, Jahrestagung, Wiesbaden, 23. September 1991
5. E.J. KAUFFELS: *OSF DCE: Der Weg zur verteilten Datenverarbeitung.* Zeitschriftenaufsatz: Datacom, Band 8, (1991), pp. 87–90, 92–94
6. M. KONG, T.H. DIENEN, P.J. LEACH, E.A. MARTIN, J.N. PATO, G.L. WAYANT: *Network computing system reference manual.* Englewood Cliffs, NJ: Prentice Hall, 1990

7. A. Kroneberg: *The OSF distributed computing environment.* GUUG, Jahrestagung, Wiesbaden, 23. September 1991

8. B.W. Lampson, M. Paul, H.J. Siegert: *Distributed systems–architecture and implementation.* Lecture Notes in Computer Science, Springer Verlag Berlin, Heidelberg, New York, 1981

9. B.W. Lampson: *Designing a global name service.* 5th ACM Symp. on Principles of Distributed Computing, Calgary, Canada, 1986

10. P. Mockapetris: *Domain names – concepts and facilities.* Network Working Group, RFC1034, November 1987, pp. 55

11. P. Mockapetris: *Domain names – implementation and specification.* Network Working Group, RFC1035, November 1987, pp. 55

12. D.C. Oppen, Y.K. Dalal: *The clearinghouse: a decentralized agent for locating named objects in a distributed environment.* ACM Trans. on Office Information Systems, Vol. 1, No.3, July 1983, pp. 230–253

13. A. Schill: *Namensverwaltung in verteilten systemen: ein überblick.* Zeitschriftenaufsatz: PIK Praxis der informationsverarbeitung und kommunikation, Band 15 (1992), pp. 11–21

14. D. Wybranietz: *Multicast–Kommunikation in verteilten systemen.* Informatik Fachberichte 242, Herausgeber: W. Brauer im Auftrag der Gesellschaft für Informatik (GI), Springer–Verlag Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong

15. L. Zahn, T.H. Dineen, P.J. Leach, E.A. Martin, N.W. Mishkin, J.N. Pato, G.L. Wayant: *Network computing architecture.* Englewood Cliffs, NJ: Prentice Hall, 1990