

Secondary Publication



Haase, Oliver; Henrich, Andreas

Query processing techniques for partly inaccessible distributed databases

Date of secondary publication: 19.03.2025

Accepted Manuscript (Postprint), Conferenceobject

Persistent identifier: urn:nbn:de:bvb:473-irb-1070643

Primary publication

Haase, Oliver; Henrich, Andreas (1997): Query processing techniques for partly inaccessible distributed databases, in: Carol Small, Paul Douglas, Roger Johnson, u. a. (Ed.), Advances in databases : 15th British National Conference on Databases, BNCOD 15, London, United Kingdom, July 7 - 9, 1997 ; proceedings, Berlin u.a.: Springer, pp. 123–125, doi: 10.1007/3-540-63263-8_11.

Legal Notice

This work is protected by copyright and/or the indication of a licence. You are free to use this work in any way permitted by the copyright and/or the licence that applies to your usage. For other uses, you must obtain permission from the rights-holders.

This document is made available with all rights reserved.

Query Processing Techniques for Partly Inaccessible Distributed Databases

Oliver Haase and Andreas Henrich

Praktische Informatik, Fachbereich Elektrotechnik und Informatik,
Universität Siegen, D-57068 Siegen, Germany,
e-mail: {haase|henrich}@informatik.uni-siegen.de
WWW: <http://www.informatik.uni-siegen.de/pi/>

Abstract. One important characteristic of distributed database management systems (DBMS) is that due to network or machine failure the environment may become partitioned into sub-environments that cannot communicate with each other. However, there are various application areas where the individual sub-environments should remain operable even in such situations [2, 5]. In particular queries to the database should be processed in an appropriate way [4].

To this end, the final and all intermediate results of queries in distributed DBMS must be regarded as potentially vague or incomplete. As a consequence, we have to deal with vague values and vague collections during query processing. Whereas vague values have been addressed in various papers dealing with null values in the relational context [1, 3], vague collections stemming from inaccessibility represent a novel research topic. In this work we present suitable representations for vague sets, vague multisets and vague lists as well as appropriate adaptations of the usual query language operators for these representations.

Our representation for *vague sets* consists of an enumerating and a descriptive part: The enumerating part in turn consists (1) of the explicit part of a lower bound of the desired set and (2) the explicit part of an upper bound of the desired set. Elements in (1) are known to be surely contained in the desired query result while elements in (2) may be in the query result. The descriptive part is a three-valued logical predicate specifying the missing elements of the vague set. These elements complete the enumerating parts to actual bounds.

The descriptive part can be employed to check for potential candidates if they belonged to the desired query result. This is applicable, because in the case of inaccessibility not only inaccessible elements can be missing in the result of a query, but also accessible elements that could not be reached by the normal query evaluation. An example for such a situation may arise when we want to traverse a path built by a multi-step relationship which is disconnected due to inaccessibility.

The adaptations of the query operators to vague sets define the explicit parts of the vague result set as well as the descriptive part. This way we implicitly define how the descriptive part of the result of a query can be calculated. Furthermore the adaptations of the operators employ the descriptive parts of the operands to revise their enumerating parts during query processing. By that way our approach can significantly improve the correctness of the vague result set.

When we are concerned with *vague multisets*, there are not only missing elements, but also the number of occurrences of enumerated elements can be vague. In our approach the first kind of vagueness is dealt with quite analogously to vague sets, while the second kind of vagueness is treated as follows: the number of occurrences of each individual element is represented by a set of possible numbers of occurrences instead of a single number. The adapted query language operators have to combine these vague occurrences when computing a vague result multiset.

Vague lists are more complicated, because when sorting a vague set or a vague multiset, even the arising order can be vague. Our approach comprises an adequate representation for vague lists, too. Our representation of vague lists minimizes the vagueness in the order of the list by enforcing the typical requirements for an order such as irreflexivity, transitivity and trichotomy. The vagueness concerning the uncertain membership of elements of the vague lists is dealt with analogously to vague sets and vague multisets.

Additionally we consider *vague aggregate functions*. In the case of inaccessibility the evaluation of an aggregate function must also be done three-valued resulting in a set of possible aggregate values. In general this evaluation can cause considerable efficiency problems. To overcome these problems, we identify two special classes of aggregate functions, that can be computed efficiently even in the case of vagueness; these classes are called set-monotone aggregations and element-monotone aggregations.

Finally we describe how our hybrid representations for vague collections can serve as the basis for the implementation of a query language for a distributed database.

Compared to other approaches — e.g. in the field of null values — the main contribution of our approach is the use of a descriptive part which describes and restricts the missing elements in a vague collection. This descriptive part can be employed in various situations to improve the enumerating parts of the query result.

References

1. E. F. Codd. Extending the database relational model to capture more meaning. *ACM Transactions on Database Systems*, 4(4):397–434, December 1979.
2. European Computer Manufacturers Association, Geneva. *Portable Common Tool Environment — Abstract Specification (Standard ECMA-149)*, June 1993.
3. G.H. Gessert. Four valued logic for relational database systems. *ACM SIGMOD Record*, 19(1):29–35, March 1990.

- O. Haase and A. Henrich. Error propagation in distributed databases. In *Proc. 4th Int. Conf. on Information and Knowledge Management (CIKM'95)*, pages 387–394. ACM press, November 1995.
- A. Heuer and A. Lubinski. Database access in mobile environments. In *Proc. 7th Int. Conf. on Database and Expert Systems Applications*, volume 1134 of *LNI CS*, pages 544–553, Zürich, 1996. Springer.