System P: Completeness-driven Query Answering in Peer Data Management Systems

Armin Roth
aroth@informatik.hu-berlin.de
Humboldt-Universität zu Berlin, Germany

Felix Naumann
naumann@hpi.uni-potsdam.de
HPI, Universität Potsdam, Germany

Abstract: Peer data management systems (PDMS) are a highly dynamic, decentralized infrastructure for large-scale data integration. They consist of a dynamic set of autonomous peers inter-connected with a network of schema mappings. Queries submitted at a peer are answered with local data and by data that is reached along paths of mappings. Due to redundancies in the mapping network, query answering in PDMS can be very inefficient if the complete query result is to be computed.

System P, a fully functional PDMS, compromises the completeness of the query result and reduces cost by pruning the query plan at mappings that are estimated to yield only few result tuples. The demo illustrates the following main components of System P: (1) adaptive estimation of result cardinalities of intermediate queries using histograms, (2) completeness-driven query planning under limited resources using specialized heuristics, and (3) the automatic generation of heterogeneous PDMS test instances, controlled by a rich set of parameters.

1 Optimizing completeness in large-scale data integration

Sharing and integrating relevant information is a pressing problem. Peer data management systems (PDMS) have been proposed to solve this task, because they can be extended to large scales solely by de-centralized coordination [HHNR05]. Query answering in PDMS can be very inefficient due to high redundancy in the network of mappings (Fig. 1). Additionally, the schema mappings between the peers suffer from information loss due to selections, which capture implicit knowledge about neighboring peers, and projections, which reflect the fact that different peers may offer different attributes about a certain real-world entity type.

In contrast to recent work [TH04], System P [1] consequently meets the important practical requirement of peers to retain as much autonomy as possible. In our PDMS, peers only exchange queries with an associated budget and resulting tuple sets. The main idea of completeness-driven query planning in System P is to prefer mappings to peers with potentially large cardinalities and to prefer mappings to peers that preserve much of the data collected by the peers "behind" a certain mapping. To prepare such decisions, each peer ranks all of its outgoing peer mappings according to the potential amount of data

[1] See animated demo at www.informatik.hu-berlin.de/mac/SystemP/

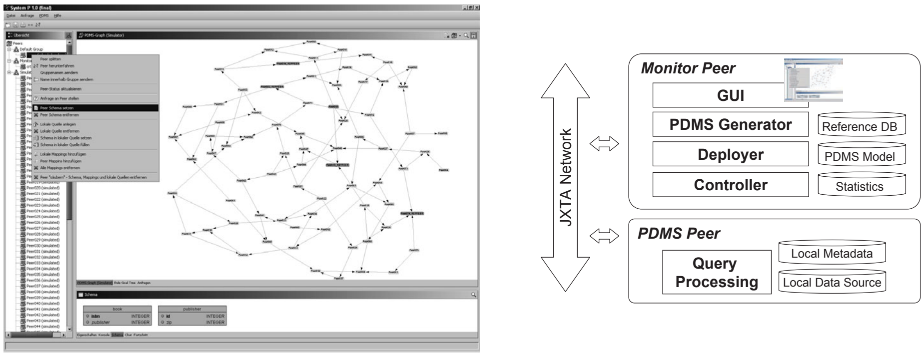

Figure 1: System P with automatically generated PDMS (left) and architecture of System P with monitor peer and PDMS peer (right).

returned. On one hand, this is achieved by estimating result cardinalities using multi-dimensional histograms. Second, the influence of data returned by an individual mapping on the completeness of the local query plan of a peer is assessed using a completeness model (Sec. 2.1). System P implements different strategies to control query planning referring to these completeness estimates (2.2).

Examples for application areas of large-scale PDMS are cooperations of scientific institutions or disaster management, which will be covered in our demo. In this field, the computing and networking resources may be restricted due to damaged infrastructure. PDMS can also serve as a de-centralized infrastructure for mediation between ontologies in the Semantic Web [HHNR05].

In general, PDMS tend to extreme inefficiency or even intractability when scaled up to tens or hundreds of peers. Therefore, techniques optimizing query answering even at cost of completeness are crucial for large-scale PDMS.

2 Components of System P

A query reformulated at a certain peer heavily branches into many more downstream queries. If there is no pruning, the results of all these subsequent queries are transported back. However, projections in the mappings may lead to a significant decrease of the amount of data collected by all peers involved in downstream query answering. So assessing that a certain mapping probably returns only few tuples can significantly save effort.

2.1 Estimating cardinality contributions

In [RN05] we discuss how a completeness model drawn from previous work can be applied to PDMS. It provides formulas to locally calculate the completeness of query plans based on projections and estimated result sizes of the mappings. Many peers are only reachable indirectly from a particular peer. System P employs multi-dimensional histograms [AC99] to estimate the cardinality of query results accessible by a certain mapping. This approach uses query feedback to maintain the histograms and is especially suited for PDMS, because their dynamic nature forces to continuously adapt histograms to changing data distributions. Because pruning of the search space and exploiting query feedback are conflicting goals, System P trades off between them to sufficiently keep track of data "behind" a mapping.

2.2 Completeness-driven query answering

Query reformulation in System P adapts to the data distribution in the PDMS and the information loss in mappings. Basically, a peer receiving a query must rank different local query plans according to their potential result sizes and prune some.

Pruning subplans. The approach shown in the demo valuates each mapping in isolation by comparing the result sizes assessed for local query plans (1) with and (2) without the mapping. Intuitively, this provides us with a measure for the data contribution of a certain mapping in the context of a certain local query plan. Our simple threshold-based pruning approach only uses mappings, whose potential amount of returned data is above a certain threshold [RN05].

Limited resources. To additionally bound resource consumption, we implemented query planning strategies, which assume that peers accept a budget to be spent for answering a query [RNHS06] . We distinguish a depth-first strategy and an approach that distributes budget in proportion to the potential data contribution of a mapping. Both can be combined with the refunding of budget not usable at a peer. We experimentally compared all of these strategies and found that the efficiency gains of the budget-driven approach are comparable to the threshold-based strategies. Our algorithms lead to interesting effects in exploring the search space, which will be shown in the demo.

3 System P Architecture and Demonstration

The main components of System P are depicted in Fig. 1. PDMS Peers form the actual, physically distributed PDMS. They run as stand-alone Java programs on any machine in the network. The monitor peer serves as a central unit for controlling the creation of the PDMS, providing a GUI, and collecting statistics about experiments. Additionally, System P includes a powerful generator to automatically create large PDMS test instances. The monitor peer serves as GUI to pose queries to any PDMS peer and to visualize how queries propagate in the system and how query results are passed along.

Demonstration. The demonstration consists of three phases: (1) Generation of PDMS instances, (2) initial adaption of the histograms, and (3) user queries and dynamic changes of the PDMS instance.

PDMS Generation. In the first phase, the viewer can create a PDMS test instance by using a wizard provided by the monitor peer. Many parameters influencing properties of the resulting PDMS can be varied. We use a complex reference schema from disaster management dealing with concepts like hospitals, medications, victims, and relief personnel. Next, the resulting PDMS instance is distributed to PDMS peers running in a physically distributed network: System P will run on several connected PCs to demonstrate that the demo is not a simulation but a full-fledged PDMS.

Histogram adaption. Before user queries can be answered efficiently, the histograms related to the mappings are adapted to the skewed multi-dimensional data distribution in the PDMS. During this phase, the monitor peer reports the estimation accuracy of the histograms.

Query answering in a dynamic PDMS. Now the system will answer user queries, optionally controlled by a user-defined budget. The viewer can choose to pose conjunctive queries with semi-interval comparison predicates to any peer or observe a randomly generated workload. System P graphically visualizes the propagation of intermediate peer queries as well as the transfer of query results back to the peer the user query originated. This intuitively displays how the search space of the PDMS is explored over time and thus shows the differences between our strategies for budget spending. Additionally, we will present PDMS instances illustrating the advantages and limitations of our strategies. The efficiency of query answering is determined by comparing the cost in terms of mappings used and the result size. Moreover, the monitor peer displays the resulting overall query plan for each user query. During this main phase, the viewer can change the PDMS topology and data distribution by selectively turning peers offline and observe how the histograms and the query answers of System P adapt to these changes.

Acknowledgments. This research was supported in part by the German Research Society (DFG grant no. NA 432).

References

[AC99] A. Aboulnaga and S. Chaudhuri. Self-tuning Histograms: Building Histograms Without Looking at Data. In *Proc. of the ACM Int. Conf. on Management of Data (SIGMOD)*, 1999.

[HHNR05] R. Heese, S. Herschel, F. Naumann, and A. Roth. Self-Extending Peer Data Management. In *Proc. of the Conf. Datenbanksysteme in Business, Technologie und Web (BTW)*, Karlsruhe, Germany, 2005.

[RN05] A. Roth and F. Naumann. Benefit and Cost of Query Answering in PDMS. In *Proc. of the Int. Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P)*, 2005.

[RNHS06] A. Roth, F. Naumann, T. Hübner, and M. Schweigert. System P: Query Answering in PDMS under Limited Resources. In *Proc. of the Workshop on Information Integration on the Web (IIWeb)*, 2006.

[TH04] I. Tatarinov and A. Y. Halevy. Efficient query reformulation in peer data management systems. In *Proc. of the ACM Int. Conf. on Management of Data (SIGMOD)*, 2004.