What Can Databases Do for Peer-to-Peer?

Paper's authors: S. Gribble, A. Halevy, Z. Ives, M. Rodrig, D. Suciu

Introduction (1/3)

- Peer-to-Peer (P2P) is totally decentralized distributed system where peers act both servers and clients
  - Dynamic: nodes may arrive or leave the system when ever they want
  - No single point of failure
  - System's robustness, availability and performance might grow with the number of the peers
  - Data placement must be done in totally distributed manner

Introduction (2/3)

- Semantics provided by P2P systems is typically weak
  - Only popular content is readily accessible
- P2P systems don't support updates to content and support only retrieval of objects by name

Introduction (3/3)

- P2P systems are lacking in the areas of
  - Semantics,
  - Data transformation and
  - Data relationships
- Those are the core strenghts of the data management community

Data Placement for Peer-to-Peer (1/2)

- Peer may have any or all of the following roles:
  - *Data origin* provides original content to the system
  - *Storage provider* stores materialized views
  - *Query evaluator* evaluates the set of queries forming its workload
  - *Query initiator* poses new queries

Data Placement for Peer-to-Peer (2/2)

- Data placement problem
  - Distribute data and work
  - Full query workload is answered
  - With lowest cost under the existing resource and bandwidth constraints

P2P Design Choices Affecting Data Placement

- Dimensions that affect the data placement problem in P2P
  - Scope of desicion-making
  - Extent of knowledge sharing
  - Heterogeneity of information sources
  - Dynamicity of participants
  - Data granularity
  - Degrees of replication
  - Freshness and update consistency

Scope of decision making

- Scale at which query processing and view materialization decisions are made
  - All queries are optimized together or
  - Every decision is made on a single-node
- Because the decisions are expensive to make on global scale, smaller scope must be used

Extent of knowledge sharing

- How much knowledge is available to a system during its query optimization process
- Centralized catalog of all views and their locations
  - Single point of failure, potential scalability bottleneck
- Replicate the complete catalog at all peers
  - Too much update traffic
- Construct a hierarchical organization as in DNS or LDAP

Heterogeneity of information sources

- Only few authoritative sources or every participant might be allowed to contribute data
- Level of heterogeneity of the data influences the degree to which a system can ensure uniform, global semantics for the data
- Single schema might be too restrictive
- Limited number of data sources and schemas is allowed

Dynamicity of participants

- Some systems assume fixed set of nodes but usually peers may join and leave at will
- If original data is distributed uniformely across network, it may become impossible to reliably access certain items
- If all data is placed only on the set of static ”servers”, the system’s flexibility and performance suffers
- Intermediate approach places all original content on the consistenly available nodes and replicates data at dynamic peers

Data granularity

- Atomic granularity level: data consists of a collection of invisibke objects
  - Place an entire object at peer or not at all
- Hierarchical granularity level: objects can be grouped into larger objects

Degrees of replication

- Data items can be replicated at will or not at all
- Large degree of replication improves query time and efficiency but makes updates harder and increases retrieval complexity
- Typical solution is to have each object be owned by a singel master

Freshness and update consistency

- Many possible ways of propagating updates from the data origins to intermediate nodes
  - Invalidation messages pushed by the server or client-initiated validation messages incur overhead and limits scalability
  - Timeout/expiration-based protocol

Complexity of the Data Placement Problem (1/3)

- Simplified form of the problem:
  - N peers, each node n_j has storage B_j and query workload $Q_j=\{q_{j1}, \ldots, q_{jm}\}$, where each query q_{ji} has an associated non-negative weight q_{ji}. Weights sum up to 1.
  - Nodes n_s and n_t is connected by the edge $e_{s,t}$ with cost $c_{s,t}$ per unit of data transferred
  - Object queries: given object identifier oid_a, return object o_a. Object o_a consumes s_a units of space.

Complexity of the Data Placement Problem (2/3)

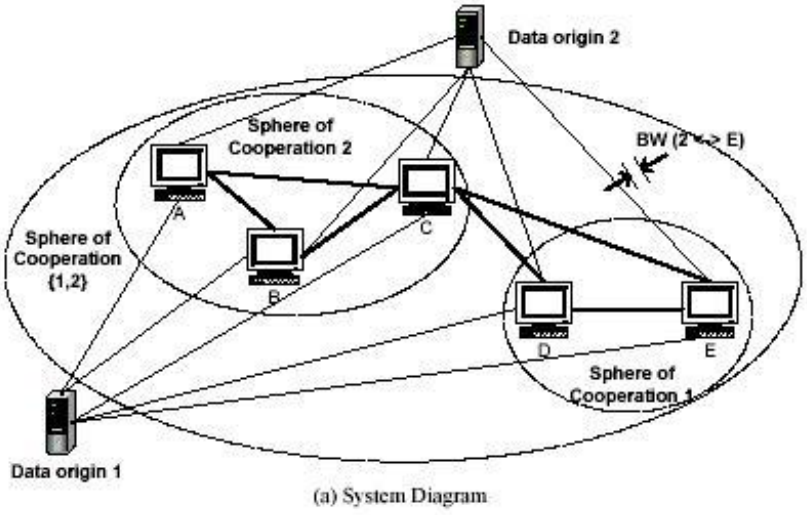
- Query cost: $s_a \times c_{s,j}$, where s_a is object's size and $c_{s,j}$ the cost of edge between the querying node and the closest node providing the queried object
- Cost of the workload at node is the weighted sum of the costs of its constituent queries
- Cost of the data placement is the sum of the costs of the workloads of the peers in the network

Complexity of the Data Placement Problem (3/3)

- Given a graph G describing a network of peers, the static data placement problem is to perform data placement with optimal cost where queries are zero-cost object lookups
- The static data placement problem is NP-complete even if all queries in the workloads in G are object queries
- Challenge is to find more specific settings in which to study problem
  - Dynamic data placement problem includes dynamic data, dynamic query workloads and dynamic peer membership

Exploiring Peer-to-Peer with the Piazza System (1/4)

- Focuses on the dynamic data placement problem
- Peers forms spheres of cooperation



(a) System Diagram

Exploiring Peer-to-Peer with the Piazza System (2/4)

- Focuses on data freshness guaranteeing and the query optimization
- For guaranteering data freshness, the materialized views must be refreshed when original data is updated
  - Piazza uses expiration times on the data items

Exploiring Peer-to-Peer with the Piazza System (3/4)

- Query optimization exploits commonalities and available data
  - Takes current query workload, finds commonalities among the queries, exploits materialized views whenever cost-effective, distibutes work under resource and bandwidth constraints, and determines whether certain results should be materialized for future use
  - Decisions are made at a level of the sphere of cooperation

Exploiring Peer-to-Peer with the Piazza System (4/4)

- Propagating information about materialized views
  - Node advertises its materialized views to its neighbors
  - Each node consolidates the advertisements and propagates them to its neighbor
- Consolidating query evaluation and data placement
  - All un-evaluable queries are broadcast within the cluster, which identifies commonalities and then assigns roles to specific nodes

Conclusions

- In my opinion the paper was quite a shallow (maybe too shallow) review to different data management issues in the P2P systems
- But some little issues came to my mind while reading the paper
  - How those spheres are actually formed (and how ”big” those are) in the Piazza system?
  - How much the advertising of materialized views in the Piazza system really add the traffic? There are also other search algorithms than the broadcasting one.

References

- Gribble S., Halevy A., Ives Z., Rodrig M., Suciu D., ”What Can Databases Do for Peer-to-Peer?”, In Proc. Fourth International Workshop on the Web and Databases (WebDB) 2001.
- Course Material of Microcomputer Applications 2002, <URL: http://appro.mit.jyu.fi/2002/kevat/ohjelmistot/luennot/luento10/>.