SkyFrame: a framework for skyline queries over a P2P network

Fernando Stefanini, Mattias Palo, and Victor Berger

School of Computer Science and Communication (CSC),
Royal Institute of Technology KTH, Stockholm, Sweden

Abstract. The paper focuses on the implementation of the Skyframel framework, for dealing with skyline queries in an efficient way in order to apply the technique in real world data. The selected framework deal with peer-to-peer (P2P) network information and suggests two implementations of querying methods: one optimized for network communication and the other one focusing on query response time, and it is tested over a Content-Addressable network, on synthetic data. The outcome is the performance of Skyframe when dealing with big sets of data, which are common when dealing with modern database systems.

Keywords: skyline query, P2P network

1 Introduction: The skyline query problem

The idea of skyline queries is fetching tuples from a table which are extremal in several criteria.

For example, on a 2D dataset, a skyline query fetching minimal values on columns a and b of a table can be expressed this way using tuple calculus:

$$\{ \ t \mid TABLE(t) \ AND \ NOT \ (\exists t')(TABLE(t') \ AND \ t'.a < t.a \ AND \ t'.b < t.b) \ \}$$

The idea is to fetch all tuples which are not dominated by any other. Defining a tuple to be dominated by an other if this other tuple has better (according to the critera of the query) values on all queried dimensions.

2 CAN, a P2P network

As Skyfame is made to be used over a P2P network, a implementation of a Content-Addressable Netowrk (CAN) protocol [2] was made, taking in consideration its core operations.

The mentioned protocol was developed on top of Peersim, a Java open source framework for simulating P2P networks.[1]

2.1 Understanding CAN

CAN assumes that the whole of the network is contained inside a domain, and every node and data inside the network can be described by a coordinate inside this domain. As nodes are added to the network, they start being owners of the area around them and of every data that is inside that area. Also, In CAN, two nodes are considered neighbors if the areas that they located at are adjacent.

When a new node wants to join the network, a 'join' message is sent to a random node in the network. This node then will be responsible to forward the message to the neighbor closest to the area where the new node should be located. When the owner of the area where the new node should be receives the message, it'll then split its on area in half and assign one portion to the joining node, as well as the data inside that area.

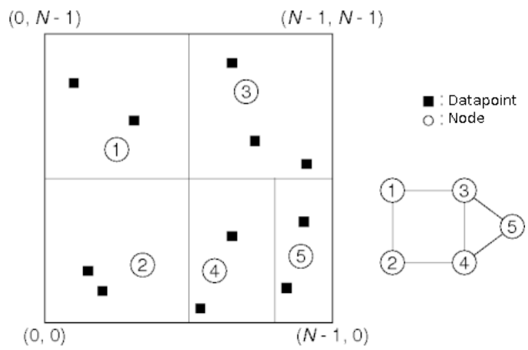

Fig. 1. A two dimensional visualization of a CAN Network

A 2D representation of CAN, with 5 nodes, can be seen in Fig. 1. On the other hand, CAN can easily be extended to support a higher number of dimensions.[2]

2.2 Simulation using Peersim

Because of its high flexibility for implementing and testing P2P networks, Peersim was chosen to be the platform on which the protocol were to be developed.

Among the features that were crucial to this decision, can be highlighted: the user-friendly documentation of the whole project and its event driven type of simulation (where network messages could be easily sent as events). Also, Peersim is capable of running the whole simulation in a single machine, what suited the limitations of this project.

2.3 About the implementation of CAN

The implementation done for this project has a few specific features that made running the simulations more convenient.

First, it allows any arbitrary number of nodes in the network, size of the data set and number of dimensions to be worked with, as long as all of these three values are natural numbers greater than 0.

Another detail is that the developed network only focus in constructing a Content-Addressable Network with the given number of nodes, dimensions and size of the data set. I.e., after the network is constructed, it only allows for queries to be done on top of it, and it doesn't support operations that change the network topology (like 'join' requests, removal of nodes, etc...).

Last, to try ensuring a better split domain, whenever a node has to split its region to give half away to a new node, it'll make the split along it's less split axis.

3 SkyFrame

SkyFrame is a framework described in [3] to optimize skyline queries in peer-to-peer networks and can be applied to CAN networks. It includes algorithms for making the queries and methods of load balancing the nodes but we only implement the search algorithms.

The framework describes two algorithms, one optimized for network communication and the other one focusing on query response time.

In both algorithms the node that sent the initial query is the query initiator and will receive and combine the results from the searched nodes. The idea is that as few nodes as possible should receive the query and according to the paper $O(2\ d\ N^{1/d})$ messages will be sent over the network to complete a query using these algorithms.

3.1 Greedy skyline search

This algorithm is optimized to minimize the amount of network communication and consist of two phases. In the first phase it is finding a global skyline point to create the search region with. For this the node responsible for the optimal point for the given query is located, for a (MIN,MIN)-query it would be the node responsible for point (0,0). This node is called the skyline query starter node and its local skyline points are also guaranteed to be global skyline points. The point with the most domination region, p_{md}, is choosen and the search region is defined as the region not dominated by p_{md}. With an initial search region the algorithm moves to phase 2.

In phase 2 the local skyline points are first sent back to the query initiator. The search region is then distributed to its neighbors, making sure that none of the subregions overlap and that the current node isn't covered. The query is sent to each neighbor with the search region distributed to it so that the neighbor nodes can continue the search in phase 2.

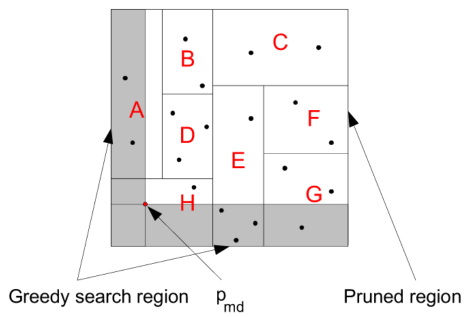

Fig. 2. Search regions of the query processed by SkyFrame

3.2 Relaxed skyline search

The nodes at the border are always searched in the greedy search since the search region defined in the end of phase 1 always cover the border nodes. In order to improve the responsiveness these nodes can be searched as soon as they are found without it being any unnecessary work. The relaxed skyline search is a modification to the greedy skyline search that start by sending queries directed to the border of each dimension.

When a border node is encountered its local skyline points are sent back to the query initiator and a query is sent to each unvisited neighbor along the border.

Since the SQ-starter node is bordering every dimension the query initiator will at some point receive a result from it and can define the search region and continue as in phase 2 of the greedy search except that nodes at the border are skipped since those are already being searched.

4 Results

We ran simulations of the algorithms according to our computation capabilities which were very limited compared to the ones of [3].

We thus simulated skyline queries on random datasets of 100.000 uniformly distributed points, in both 2D and 3D, on networks ranging from 128 to 4096 nodes.

We ran each simulation 16 times and computed averages value displayed on figure 3.

On these criteria, our results were very similar to the ones presented in [3].

Total number of nodes	Nodes involved				Exchanged messages			
	2D gss	2D rss	3D gss	3D rss	2D gss	2D rss	3D gss	3D rss
128	23.5	25.3	60.7	61.6	28.6	52.2	186.5	187.6
256	37.7	35.7	95.2	98.4	73.5	70.7	291.7	307.6
512	47.8	52.1	154.7	159.4	93.1	106.1	482.6	508.5
1024	68.5	67.8	257.7	244.4	141.6	136.8	811.9	770.8
2048	105.0	104.2	432.3	439.3	213.2	207.4	1374.2	1415.6
4096	148.4	165.8	709.3	673.2	306.5	369.4	2276.4	2203.5

Fig. 3. Results of the simulations: average number of involved nodes and exchanged message in the processing of a skyline query for different network sizes. Both for 2D and 3D datasets, using the greedy and relaxed algorithms.

References

1. A. Montresor and M. Jelasity: PeerSim: A scalable P2P simulator. In *Proc. of the 9th Int. Conference on Peer-to-Peer (P2P'09)*, pages 99–100. Seattle, WA, September 2009
2. A. Popescu, D. Ilie, D. Kouvatsos: On the implementation of a content-addressable network. 1st International Workshop on System and Theory of Cognitive Radio Networks (STOCK), Karlskrona, Sweden.4
3. S. Wang, Q.H. Vu, B.C. Ooi, A.K.H. Tung and L. Xu: Skyframe: A framework for skyline query processing in peer-to-peer systems. VLDB Journal, 2009, vol. 18, pp. 345-362.