1. Introduction to Skyslines

The skyline is the set of most competitive points in a dataset.

The skyline simplifies data for users by removing points that are “worse” (larger) on every dimension than other points... like those with incoming black arrows to the left.

Given d-dimensional points p in set S, Skyline(S) = \{ p \in S | \exists p' \in S, (p = p') \land (\forall d \in [0, d] : p_d < p'_d) \}

Skyline construction times increase with: more points | more dimensions | more anticorrelation in the data

2. Motivation

- Online skyline querying can take 5 minutes
- Multicore paradigm [1] = Divide & conquer
- Sequential algorithms rely on sorting [1], trees [3]
- Best sequential algorithms faster than multicore by efficiently using best global skyline points

Partition these points optimally for 2 threads

3. The Hybrid Skyline Algorithm

Two-phased loop optimizes for order and parallelism

Simplified pseudocode:
1. Sort S monotonically.
2. for i = 0, i < |S|, i = i + \alpha do*
3. for j \in [i, i + \alpha) parallel do**
4. Compare point j to known skyline
5. Compare point j to [i, j - 1]
6. Append skyline points from [i, i + \alpha)

*< \alpha sub-optimal comparisons/point
**Work on j terminates once dominated

4. Global Data Structures

Comparisons to known skyline points accelerated with array-based tree structure:

- Induce quad-tree partitioning [3]
- Four coalesced, append-only arrays
- Only updated when synchronizing (after every \alpha points)
- Sort by partition popcount + value + point \lambda, norm = locality + monotonicity
- One level of indirect mapping partitions to start of points

5. Comparison to State-of-the-Art

Scalability (data size)

Scalability (threads)

6. Granular Performance Analysis

Run times on default workload (n = 10^6, d = 12) are decomposed into 6 categories:

Download our SkyBranch experiment suite at:
http://cs.au.dk/research/research-areas/data-intensive-systems/repository/

References


31st IEEE International Conference on Data Engineering (ICDE 2015)