**Finding K-dominant skyline point in distributed data sites**

Project group 5

Ritika (15111036) & Swapnil Mhamane (15111044)

**Abstract:**

In majority of applications, large set of attributes are associated with objects and set of such large data object is stored over different physical sites forming cluster(Each site is connected to each other). In such case finding k-dominant skyline objects[1] with naïve approach of collecting all data object at one site and then finding k-skyline objects of user interest amongst them seems to have higher cost in terms of network traffic and time consumption as well. Achieving such objective is with lesser network traffic and lower time consumption is challenging task. As per survey the currently there has not been any work done in this area.

There has been some optimization work done in computation of distributed skyline queries in efficient manner so as to reduce network traffic with the design of execution plan for different sites in. Here some of sites are also pruned from existing algorithm at query site[2]. Based on partial approach from this framework we will process distributed k-dominant skyline query.

**Algorithm:**

1. Let Sorg be the original site at which query is fired. Collect the MBR from all sites at Sorg.
2. Prune sites whose MBR completely dominated by MBR of some other site. (Here we are using the fact that any point from dominating MBR will definitely K-dominate all points from pruned site.)
3. Execute queries at each site in order formed m-way tree structure (m-way structure to define degree of parallelism)
4. At each site find probable k-dominated points and full skyline points using two scan algorithm [1].
5. From leaf node to root node, at each level in execution tree pass on local k-dominants and full skyline points to parent.
6. At each intermediate site in execution tree prune local as well as k-dominants and full skyline points from child site (This will lower down the network traffic further).

**References:**

1. **“**Finding k-Dominant Skylines in High Dimensional Space” Chee-Yong Chan, H.V. Jagadish, Kian-Lee Tan, Anthony K.H. Tung, Zhenjie Zhang

2. “Efficient Execution Plans for Distributed Skyline Query” Processing Joao B. Rocha-Junior, Akrivi Vlachou, Christos Doulkeridis, and Kjetil Norvag