

Parallel Density-Based Clustering of Complex Objects

Stefan Brecheisen, Hans-Peter Kriegel
and Martin Pfeifle

University of Munich, Germany
Institute for Informatics
Database and Information Systems
www.dbs.ifi.lmu.de



Outline

- Introduction
- Server-Side Data Partitioning
- Client-Side Clustering
- Server-Side Merging
- Experimental Results
- Conclusions

Outline

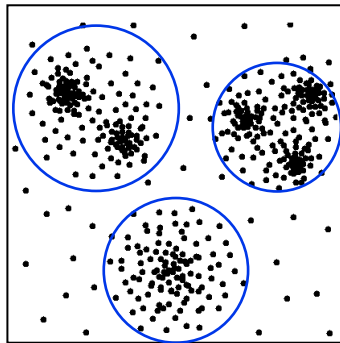
- Introduction
- Server-Side Data Partitioning
- Client-Side Clustering
- Server-Side Merging
- Experimental Results
- Conclusions

Clustering

- Clustering
 - Efficiently grouping the database into sub-groups (clusters) such that
 - similarity within clusters maximized
 - similarity between clusters minimized

Flat Clustering

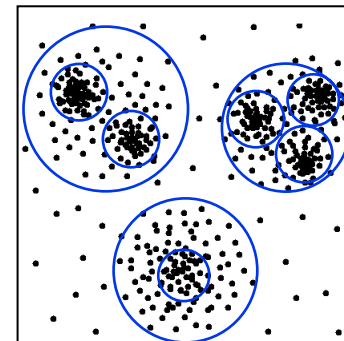
one level of clusters



e.g. density-based clustering algorithm
DBSCAN [KDD 96]

Hierarchical Clustering

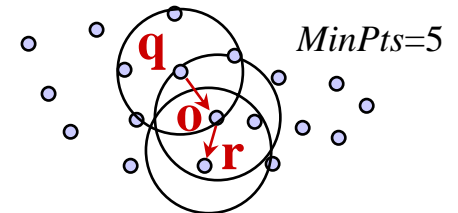
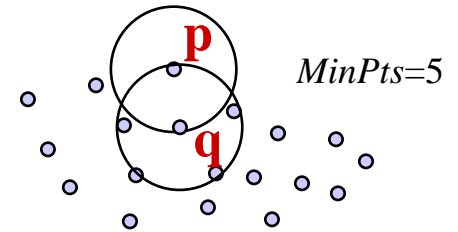
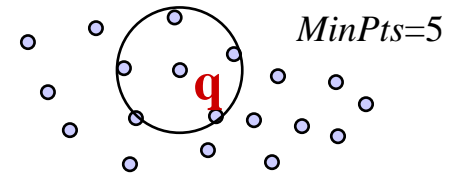
nested clusters



e.g. density-based clustering algorithm
OPTICS [SIGMOD 99]

Density-Based Clustering (1)

- Parameters
 - range ε and minimal weight $MinPts$
- Definition: core object
 - q is **core object** if $|rangeQuery(q, \varepsilon)| \geq MinPts$
- Definition: directly density-reachable
 - p **directly density-reachable** from q if q is a core object and $p \in rangeQuery(q, \varepsilon)$
- Definition: density-reachable
 - density-reachable**: transitive closure of “directly density-reachable”

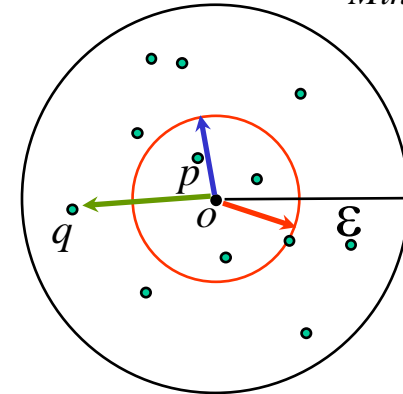


Density-Based Clustering (2)

- Core Idea of OPTICS:

Order the objects linearly such that objects of a cluster are adjacent in the ordering.

$MinPts = 5$



- Definition: core-distance

$$core-dist_{\epsilon, MinPts}(o) = \begin{cases} \infty & \text{if } |rangeQuery(o, \epsilon)| < MinPts \\ MinPts - dist(o) & \text{otherwise} \end{cases}$$

- Definition: reachability-distance

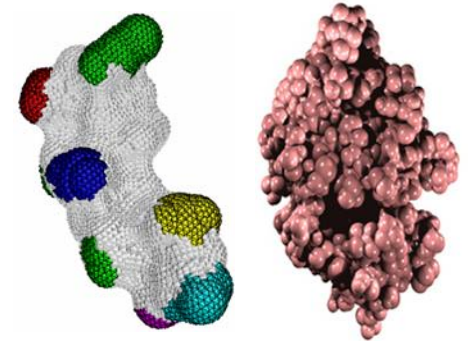
$$reach-dist_{\epsilon, MinPts}(p, o) = \max(core-dist_{\epsilon, MinPts}(o), dist(p, o))$$

Outline

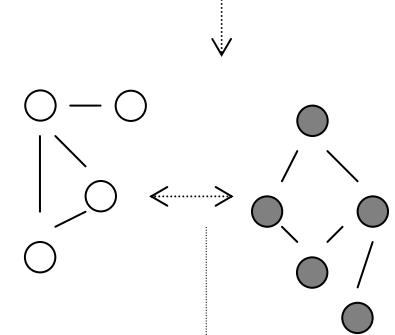
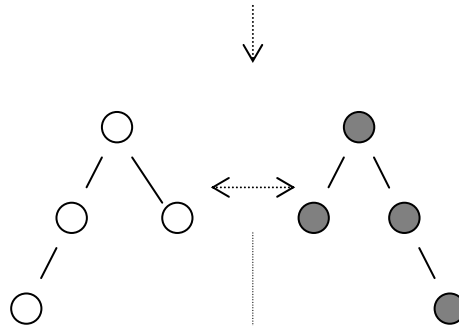
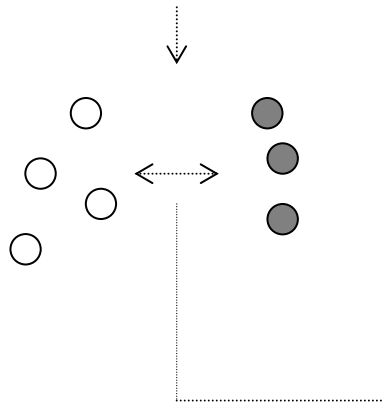
- Introduction
- Server-Side Data Partitioning
- Client-Side Clustering
- Server-Side Merging
- Experimental Results
- Conclusions

Complex Objects

complex
objects

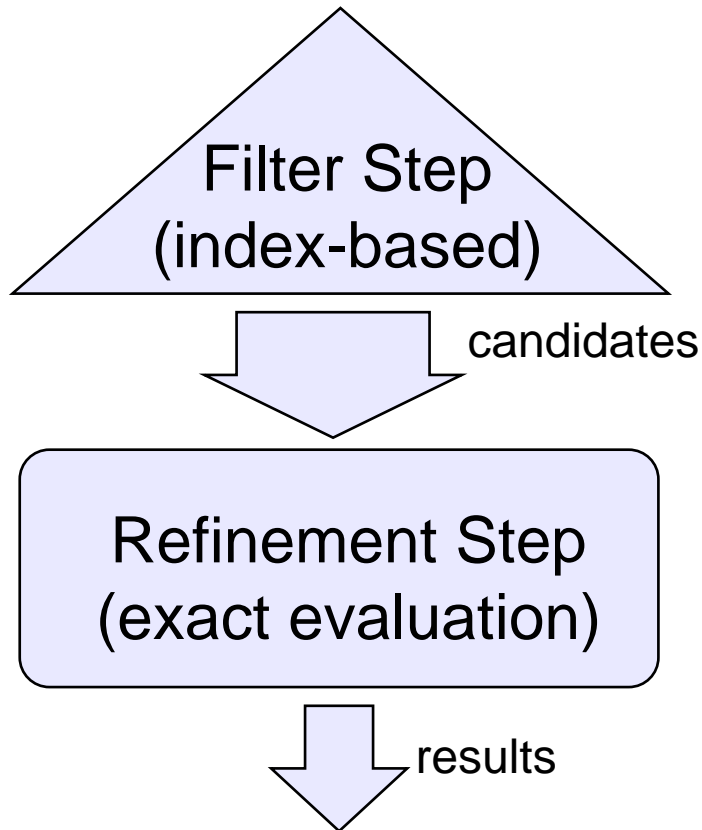


complex
models



complex distance measure

Multi-Step Query Processing



- Multi-Step Similarity Search
 - Range Queries (*Faloutsos et al. 94*)
 - k -Nearest Neighbor Queries (*Korn et al. 96*)
 - Optimal k - Nearest Neighbor Queries (*Seidl, Kriegel 98*)
- No False Drops?

Lower-Bounding Property

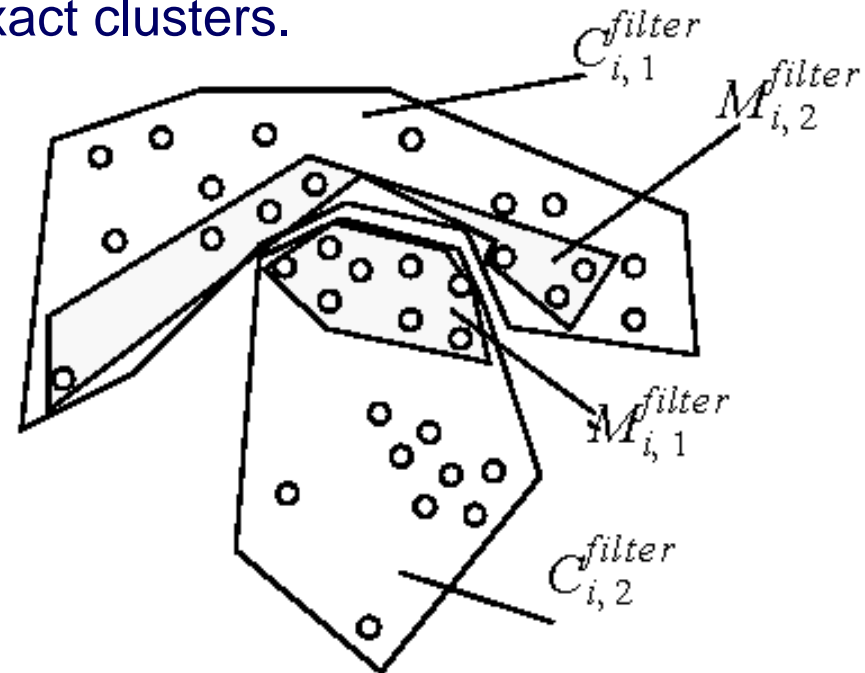
$$d_f(p, q) \leq d_o(p, q)$$

filter distance object distance

Server-Side Partitioning

Perform OPTICS based on the filter distances to obtain:

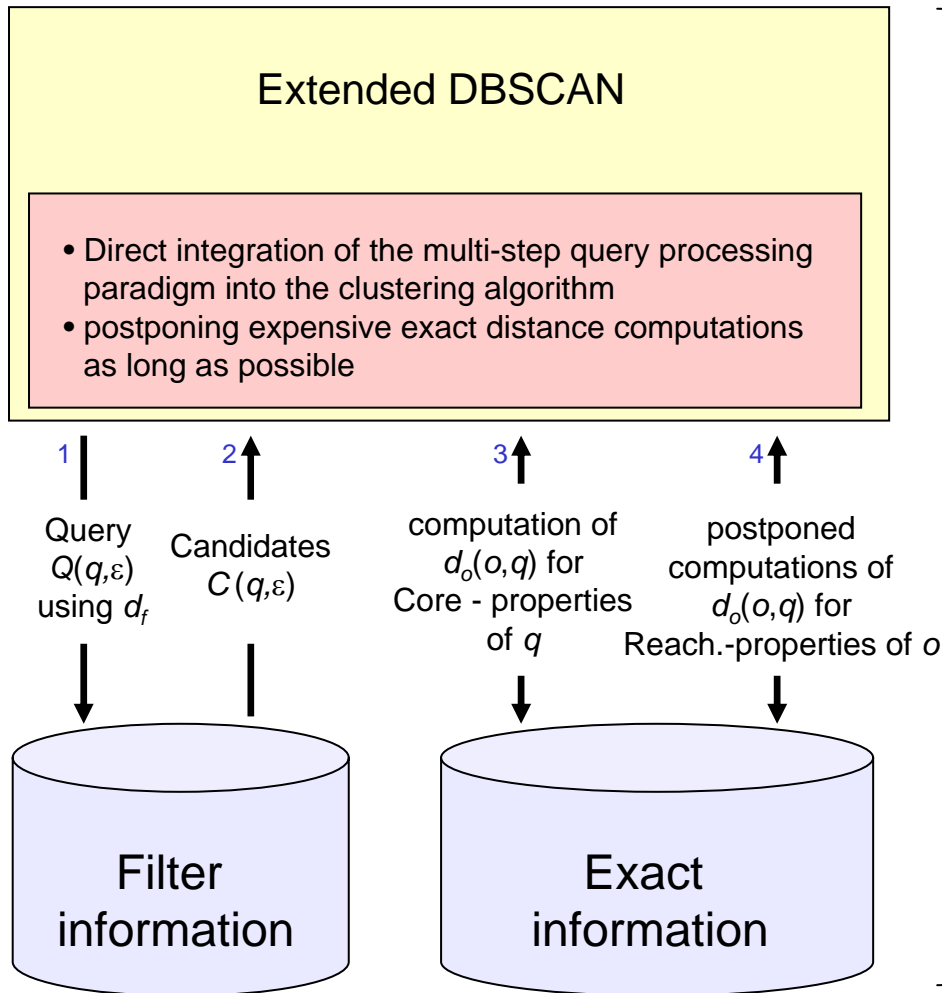
- Conservatively approximated clusters, i.e. filter clusters are supersets of exact clusters.
- Progressively approximated noise, i.e. filter noise is a subset of exact noise.
- Filter merge points.
Split large filter clusters to evenly distribute load on the clients.
Needed to determine core properties during client-side clustering



Outline

- Introduction
- Server-Side Data Partitioning
- Client-Side Clustering
- Server-Side Merging
- Experimental Results
- Conclusions

Integrated Multi-Step Clustering



- For each database object q , we perform one range query on the filter information (1,2).
- Only those exact distances $d_o(o, q)$ are computed which are necessary to determine the core-properties of q (3).
- A beneficial heuristic for determining the reachability-properties is applied which saves on exact distance computations (4).
- Adapted from [ICDM 04] to also handle filter merge points.

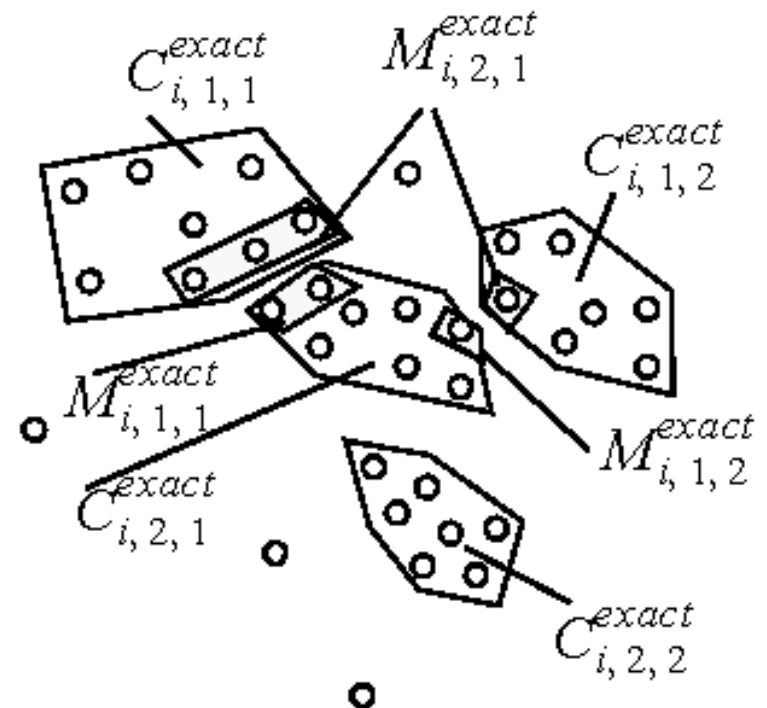
Outline

- Introduction
- Server-Side Data Partitioning
- Client-Side Clustering
- Server-Side Merging
- Experimental Results
- Conclusions

Server-Side Merging

Merge the local clusterings using:

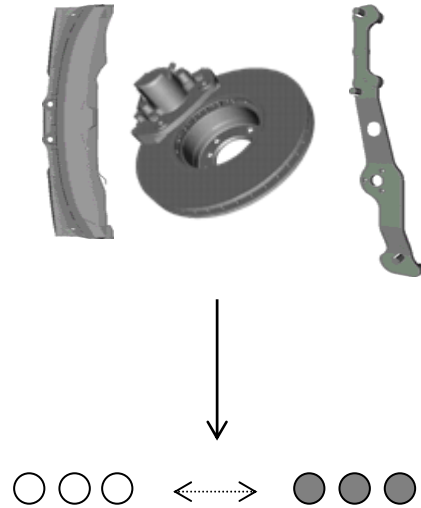
- Exact merge points, i.e. those filter merge points which are density-reachable from an exact local cluster.
- Cluster connectivity graph, i.e. vertex for each exact local cluster, edge between those clusters which are connected by an exact merge point.
- Database connectivity graph, i.e. the union graph of all cluster connectivity graphs.
Set of maximal connected subgraphs corresponds to global exact DBSCAN clustering.



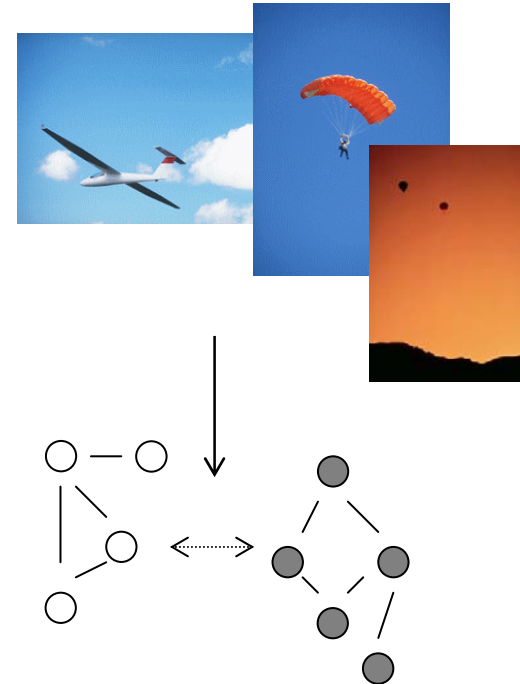
Outline

- Introduction
- Server-Side Data Partitioning
- Client-Side Clustering
- Server-Side Merging
- Experimental Results
- Conclusions

Test Data Sets



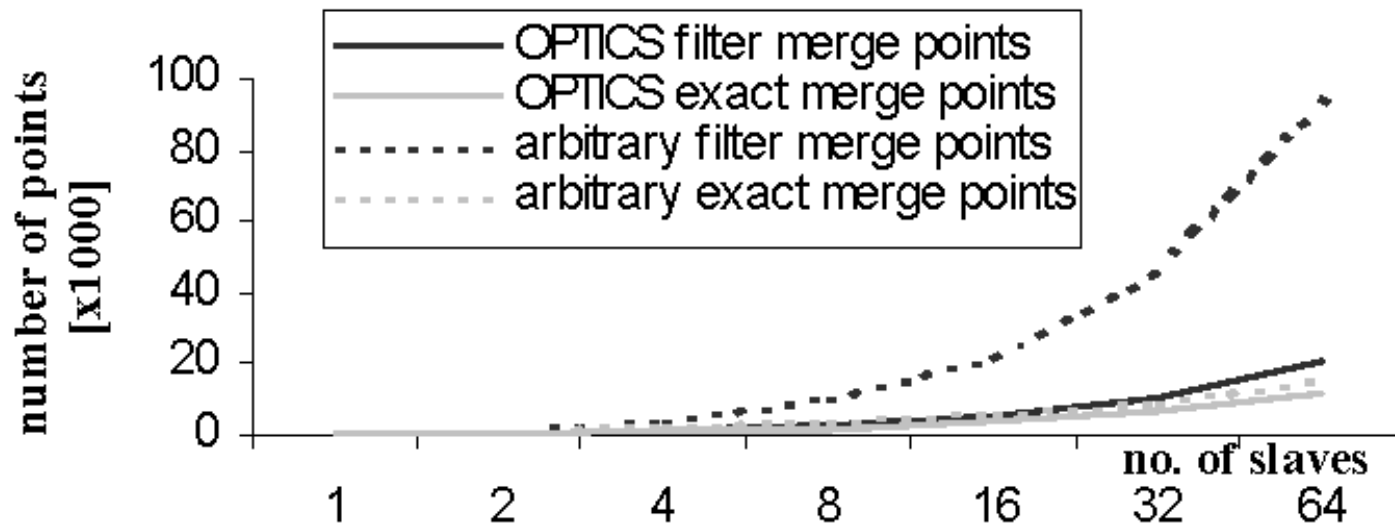
- High dimensional feature vectors and sets of feature vectors representing CAD objects [DASFAA 03, SIGMOD 03]
- not very selective filters used (Euclidean norm, Centroid distance)



- Graphs representing images [DAWAK 03]
- Expensive exact distance function
- Selective filter used

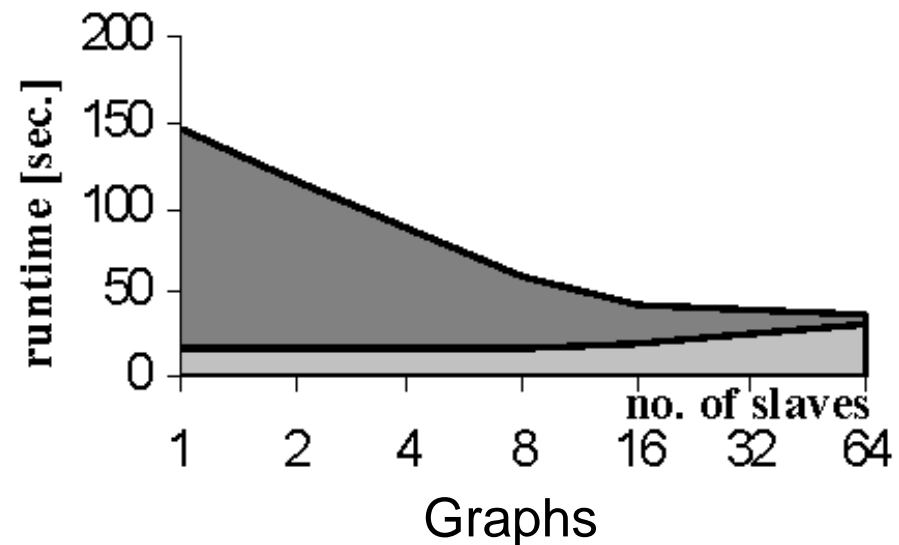
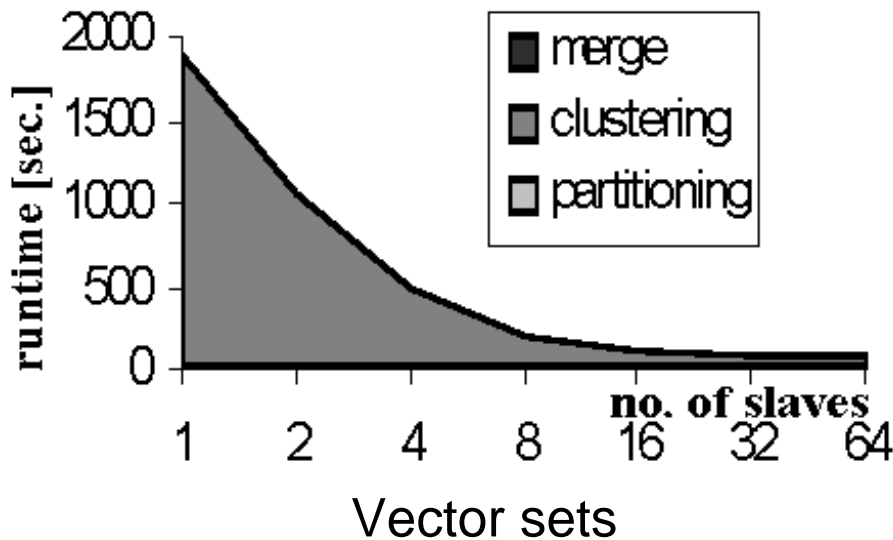
Merge Points

Number of merge points w.r.t. a varying number of slaves for the graph dataset



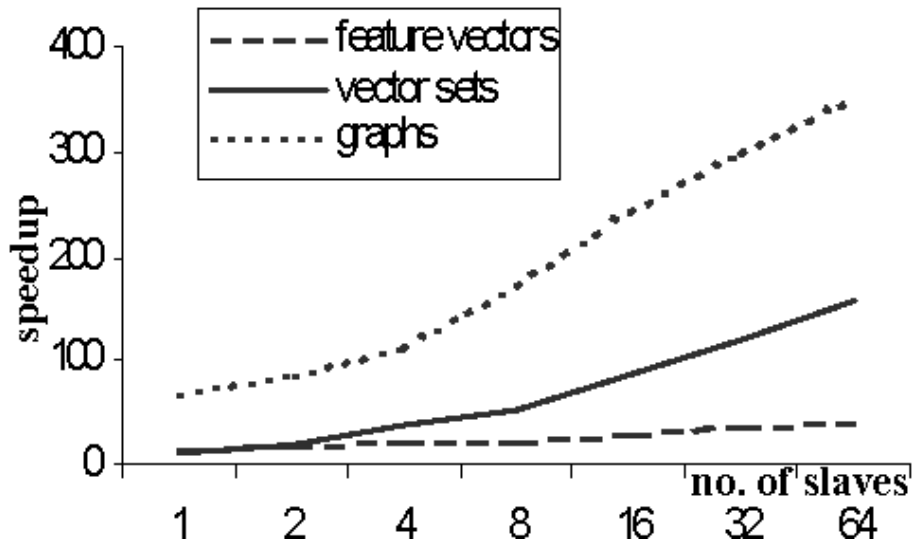
Runtimes

Accumulated runtimes w.r.t. a varying number of slaves

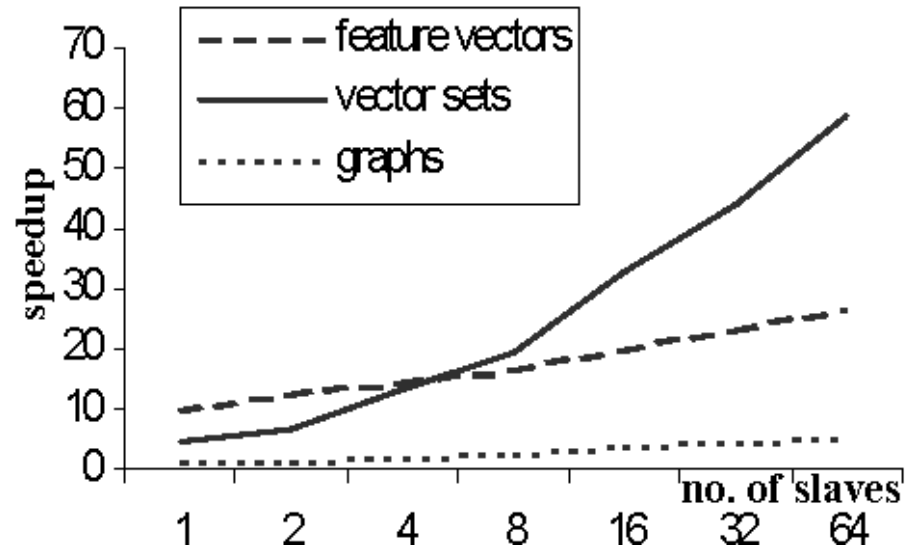


- partitioning step includes communication costs
- no communication costs during the clustering step

Overall speedup w.r.t. a varying number of slaves



In relation to DBSCAN
based on a full table scan



In relation to DBSCAN based
on traditional multi-step query
processing

Outline

- Introduction
- Server-Side Data Partitioning
- Client-Side Clustering
- Server-Side Merging
- Experimental Results
- Conclusions

Conclusions

Summary

- Fair and suitable partitioning strategy using OPTICS as a “space filling curve” for general metric objects.
- Efficient local clustering by integrating the multi-step query processing paradigm.
- Global cluster connectivity graph based on merge points.

Future Work

- Parallelizing other data mining algorithms based on lower-bounding distance functions.

Thank you!
Any Questions?