Large-Scale Data Management and Analysis for Astronomical Research

Presenter: Cheng-Hsien Tang

Authors:
Cheng-Hsien Tang, Min-Feng Wang, Wei-Jen Wang, Meng-Feng Tsai*, Yuji Urata, Chow-Choong Ngeow, Induk Lee, and Kuiyun Huang

Date: 2011/03/25

Outline

- Introduction
- Architecture
- Parallel Hierarchical Agglomerative Clustering System
- Similarity Classification System
- Astronomical Information Management System
- Conclusions
- Q & A

Introduction

Motivation

- Major source of abundant data
  - Business: e-commerce, transaction, stock, …
  - Science: bioinformatics, simulation
  - Daily life: news, digital camera, etc
- Pressing need for data mining
  - Statistics, Classification , …
- Scale of data
  - Terabytes or Petabytes of data

We need better analytical tools!

Distributed Computing

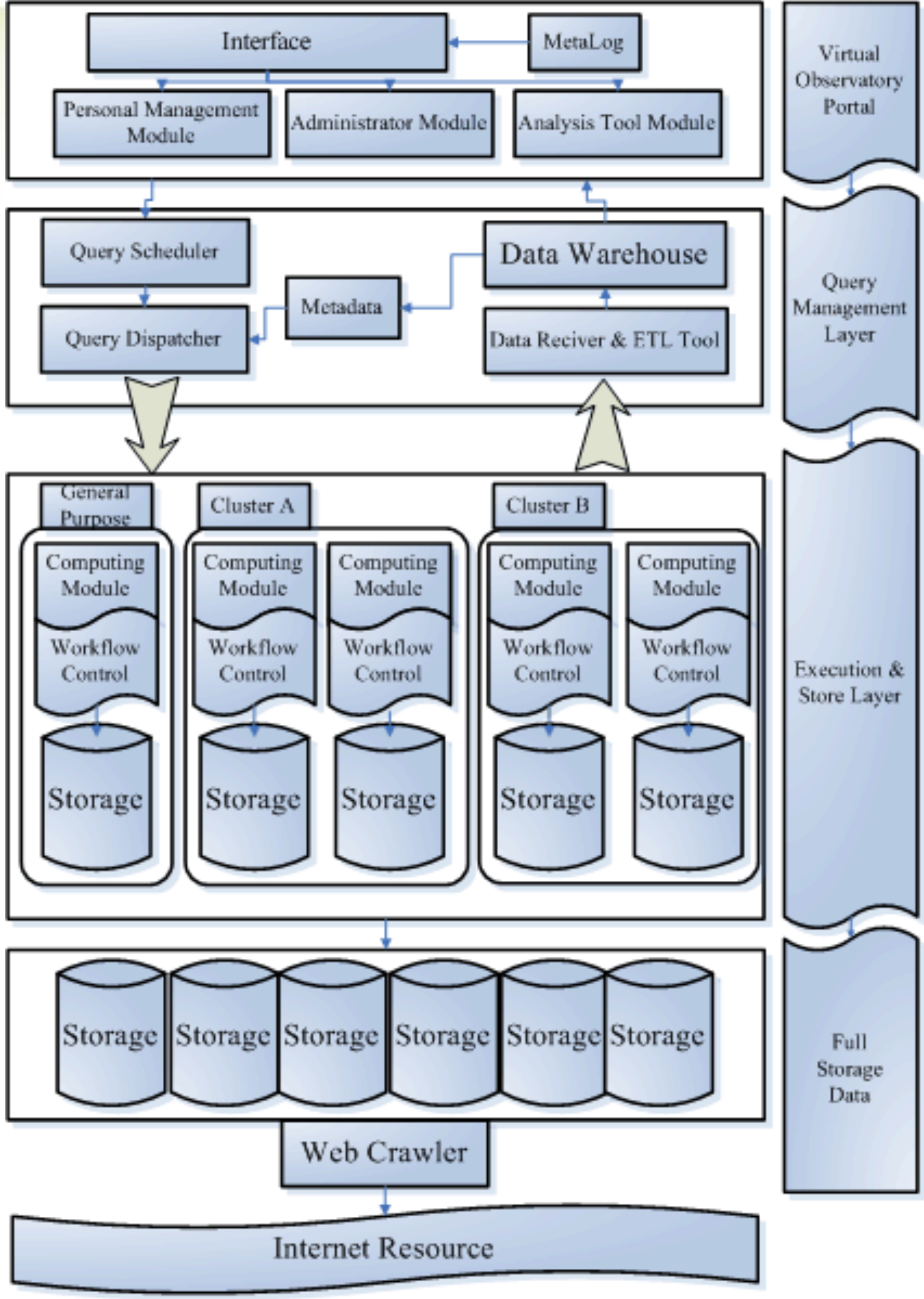
- The “New” Moore’s Law
  - Computers no longer faster, just wider
- Limits of single CPU computing
  - Small memory size
  - Long execution time

We can use parallel computing to accelerate big data analysis!

Objectives

- Applying parallel computing to astronomical research
- Refining existing algorithms for a better performance
- Providing an application template
- Developing management system to maintain large-scale data

Architecture


Interface
MetaLog
Personal Management Module
Administrator Module
Analysis Tool Module
Virtual Observatory Portal
Query Scheduler
Data Warehouse
Metadata
Query Dispatcher
Data Reciver & ETL Tool
Query Management Layer
General Purpose
Cluster A
Cluster B
Computing Module
Workflow Control
Storage
Execution & Store Layer
Storage
Full Storage Data
Web Crawler
Internet Resource

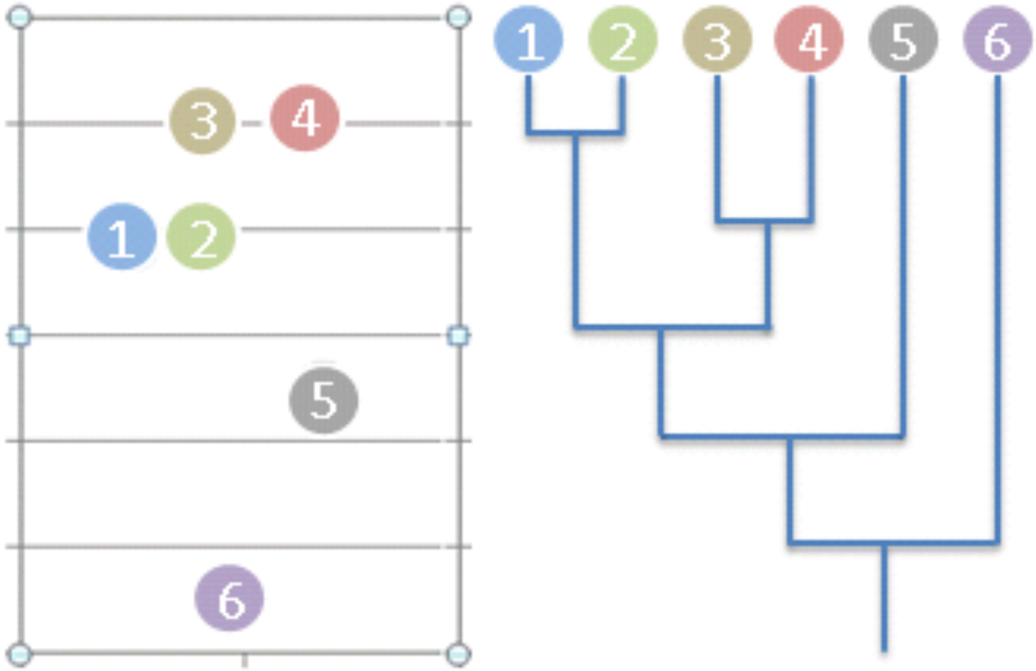
Systems

- PARallel Hierarchical Agglomerative Clustering System (PARHACS)
  – A system with distributed message-passing algorithm to calculate a hierarchical cluster
- SIMilarity Classification System (SIMCS)
  – A decentralized Multiple Classifier System (MCS) framework to support a complex classification procedure using multiple classifiers.
- ASTROnomical Information Management System (ASTROIMS)
  – An integrated interface with multidimensional data-warehouse design for fast data retrieval and management.

Parallel Hierarchical Agglomerative Clustering System

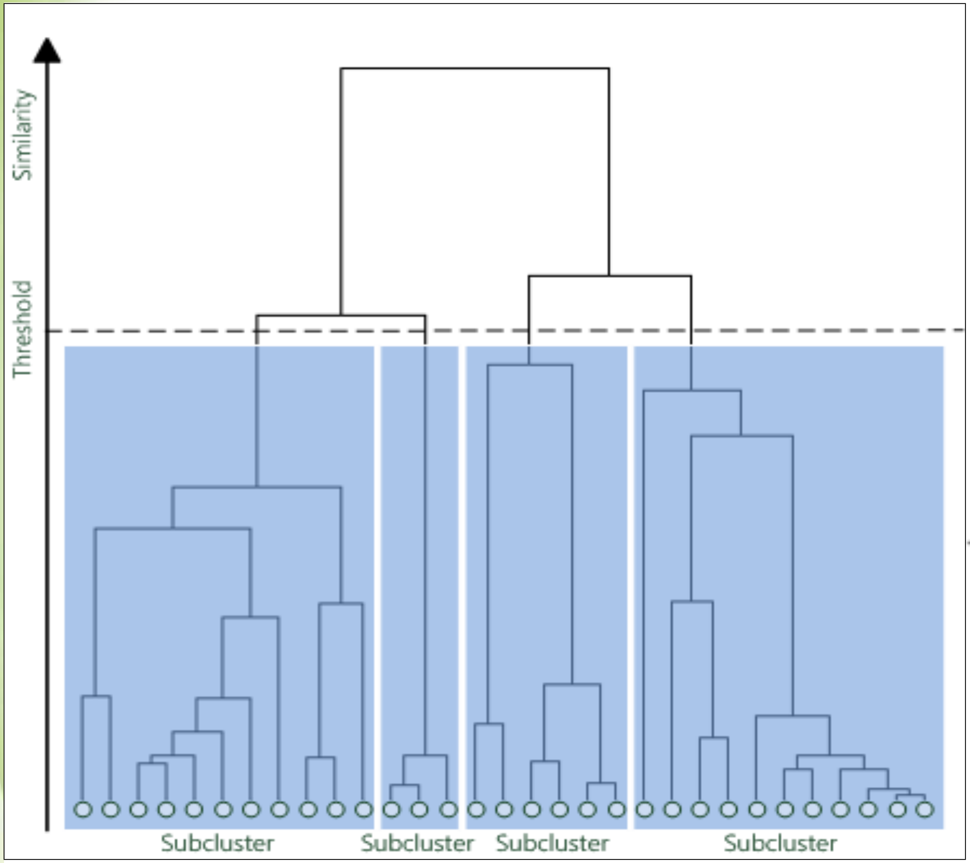
Clustering Algorithms

- Hierarchical clustering
  - Divisive way
  - Agglomerative way



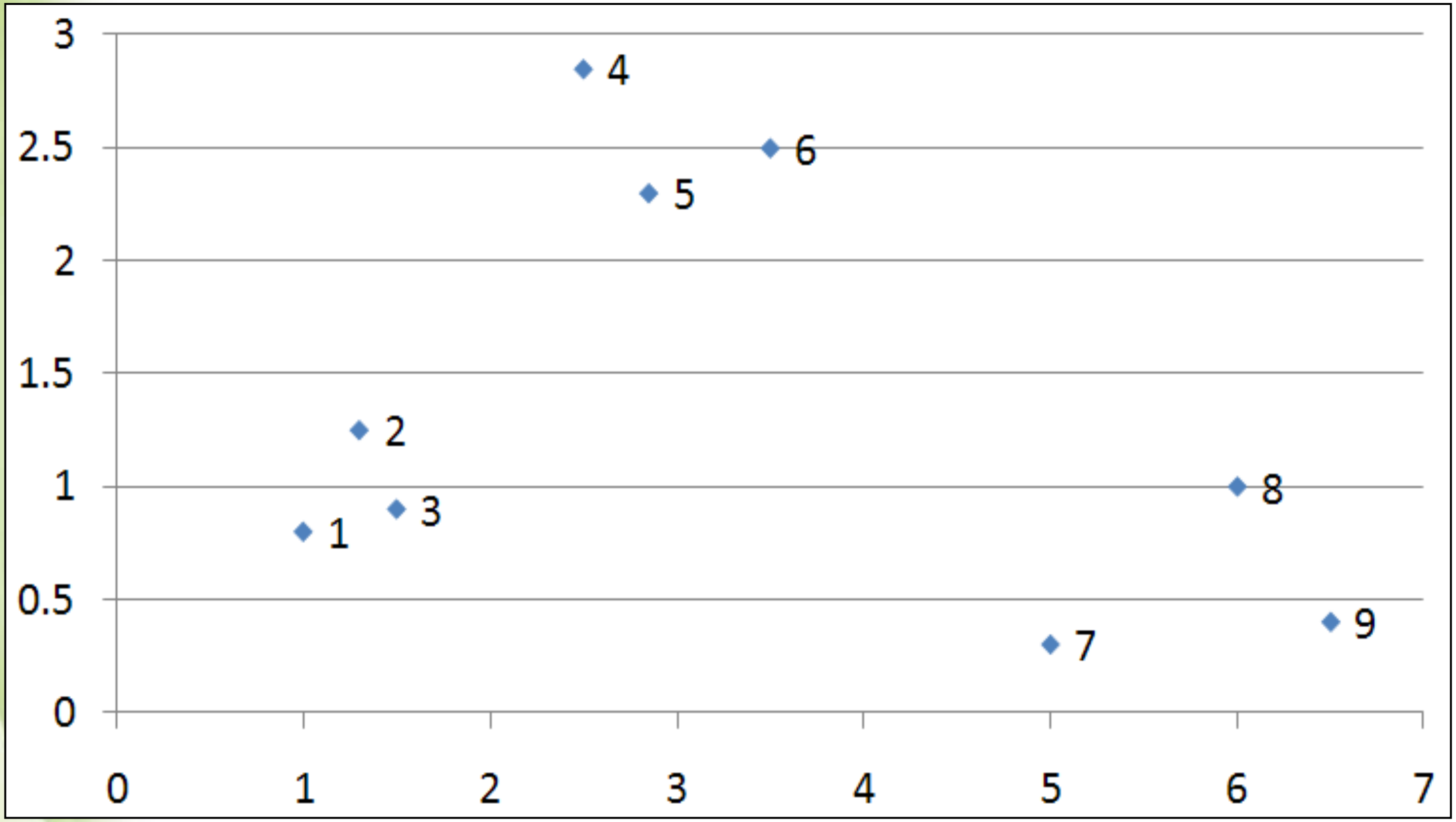
Applying Divide-and-Conquer

- Use a similarity threshold to parallelize the clustering phase and then merge to a single hierarchical tree



Example

	1	2	3	4	5	6	7	8	9
X	1	1.3	1.5	2.5	2.85	3.5	5	6	6.5
Y	0.8	1.25	0.9	2.85	2.3	2.5	0.3	1	0.4



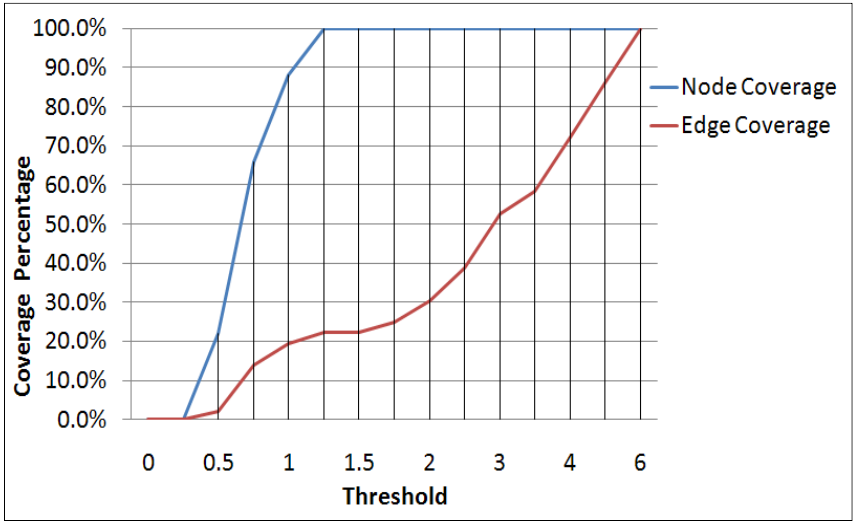
Stage 1

- Parallelism strategy of Computing similarity matrix in parallel
  – Row-based

	1	2	3	4	5	6	7	8	9
1									
2	0.5408								
3	0.5099	0.4031							
4	2.5401	2.0000	2.1914						
5	2.3817	1.8721	1.9448	0.6519					
6	3.0232	2.5303	2.5612	1.0594	0.6800				
7	4.0311	3.8200	3.5510	3.5710	2.9364	2.6627			
8	5.0039	4.7066	4.5011	3.9588	3.4077	2.9154	1.2206		
9	5.5145	5.2690	5.0249	4.6906	4.1149	3.6619	1.5033	0.7810	

Stage 1 (cont)

- Data coverage
  - Node coverage
    - the ratio of data items the threshold can cover.
  - Edge coverage (Set coverage)
    - the ratio of cells in the similarity matrix the threshold can cover.



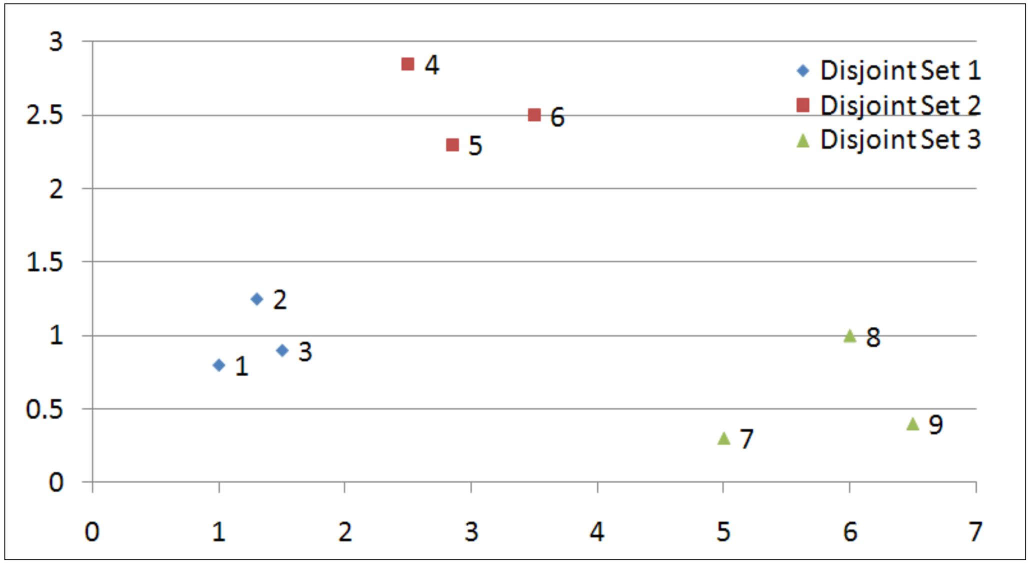
Stage 1 (cont)

- Reduce space cost
  - Assume the threshold is 1.25

	1	2	3	4	5	6	7	8	9
1									
2	0.5408								
3	0.5099	0.4031							
4	2.5401	2.0000	2.1914						
5	2.3817	1.8721	1.9448	0.6519					
6	3.0232	2.5303	2.5612	1.0594	0.6800				
7	4.0311	3.8200	3.5510	3.5710	2.9364	2.6627			
8	5.0039	4.7066	4.5011	3.9588	3.4077	2.9154	1.2206		
9	5.5145	5.2690	5.0249	4.6906	4.1149	3.6619	1.5033	0.7810	

Stage 2

- Using disjoint set algorithm



Stage3

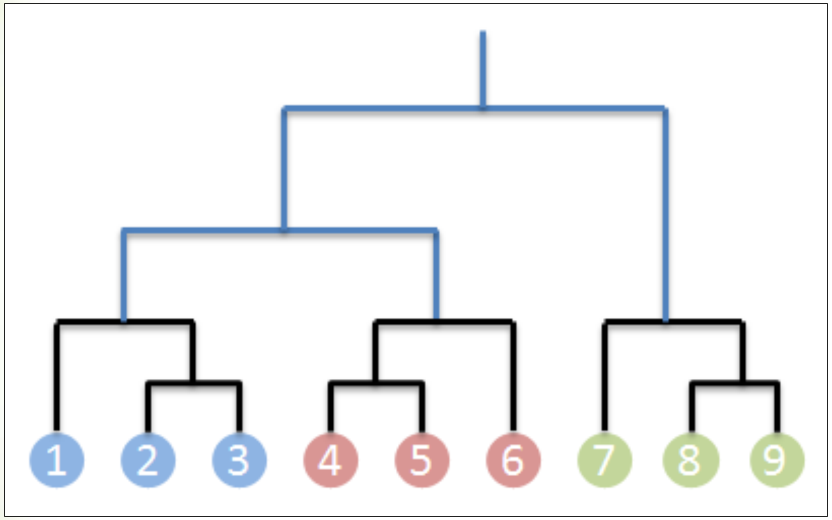
- Similarity of disjoint sets

	1	2	3	4	5	6	7	8	9
1									
2									
3									
4	2.5401	2.0000	2.1914						
5	2.3817	1.8721	1.9448						
6	3.0232	2.5303	2.5612						
7	4.0311	3.8200	3.5510	3.5710	2.9364	2.6627			
8	5.0039	4.7066	4.5011	3.9588	3.4077	2.9154			
9	5.5145	5.2690	5.0249	4.6906	4.1149	3.6619			

- Parallelism strategy
  – Set-based

Stage4

- Clustering of disjoint sets
  - Using the result of stage1 and 2 to clustering lower structure
  - Using the result of stage3 to clustering upper structure



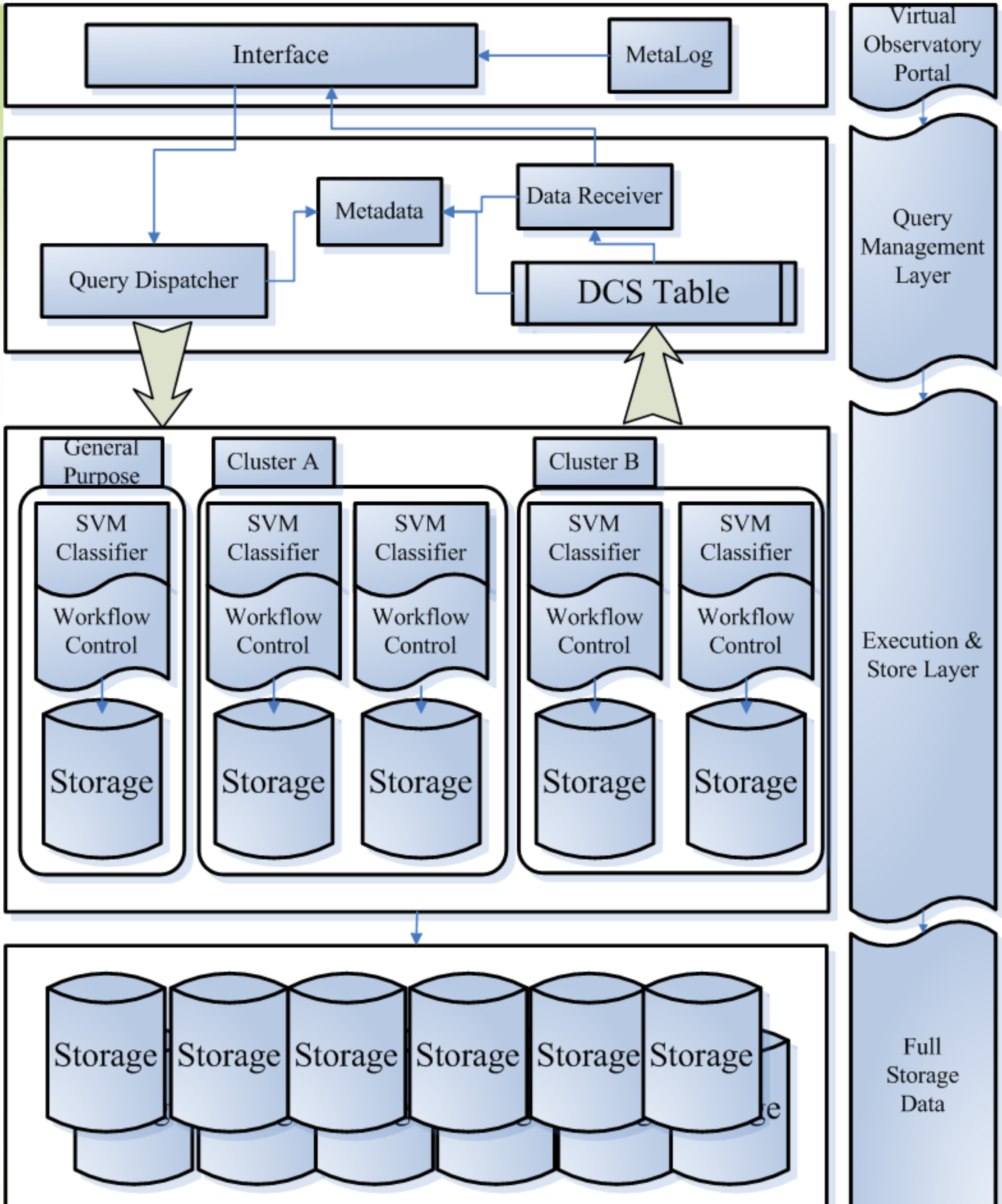
Similarity Classification System

Similarity Classification System

- A decentralized multiple classifier system (MCS) base on SVM and machine learning
- Why SVM
  - Competitive with existing classification methods and relatively easy to use
  - “Predict” which group the new coming data belong to base on the old classified data

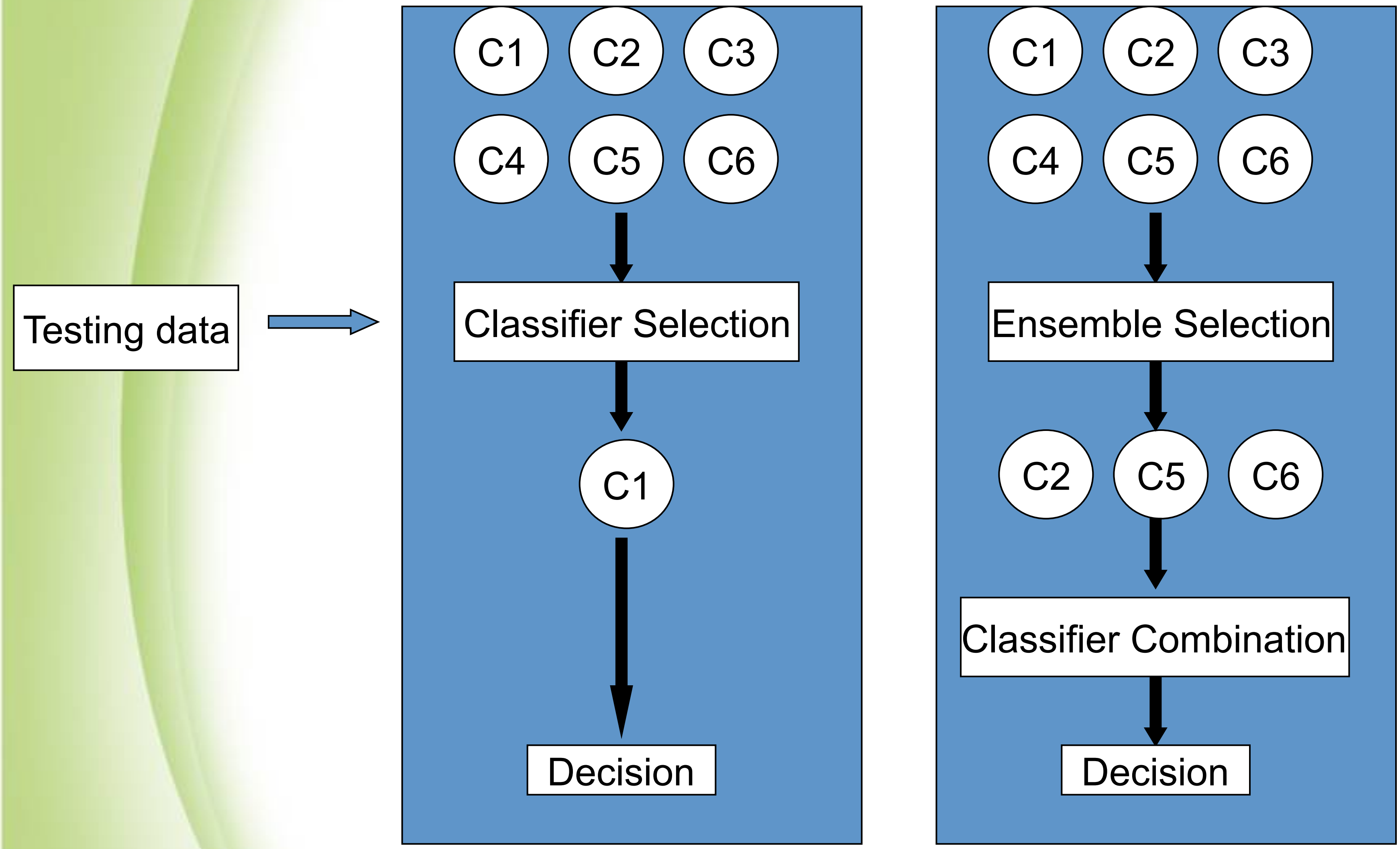
8

  - You don’t need to know the conditions when you are doing classification



Interface
MetaLog
Virtual Observatory Portal
Metadata
Data Receiver
Query Dispatcher
DCS Table
Query Management Layer
General Purpose
Cluster A
Cluster B
SVM Classifier
Workflow Control
Storage
Execution & Store Layer
Full Storage Data

Classifier Selection/Combination



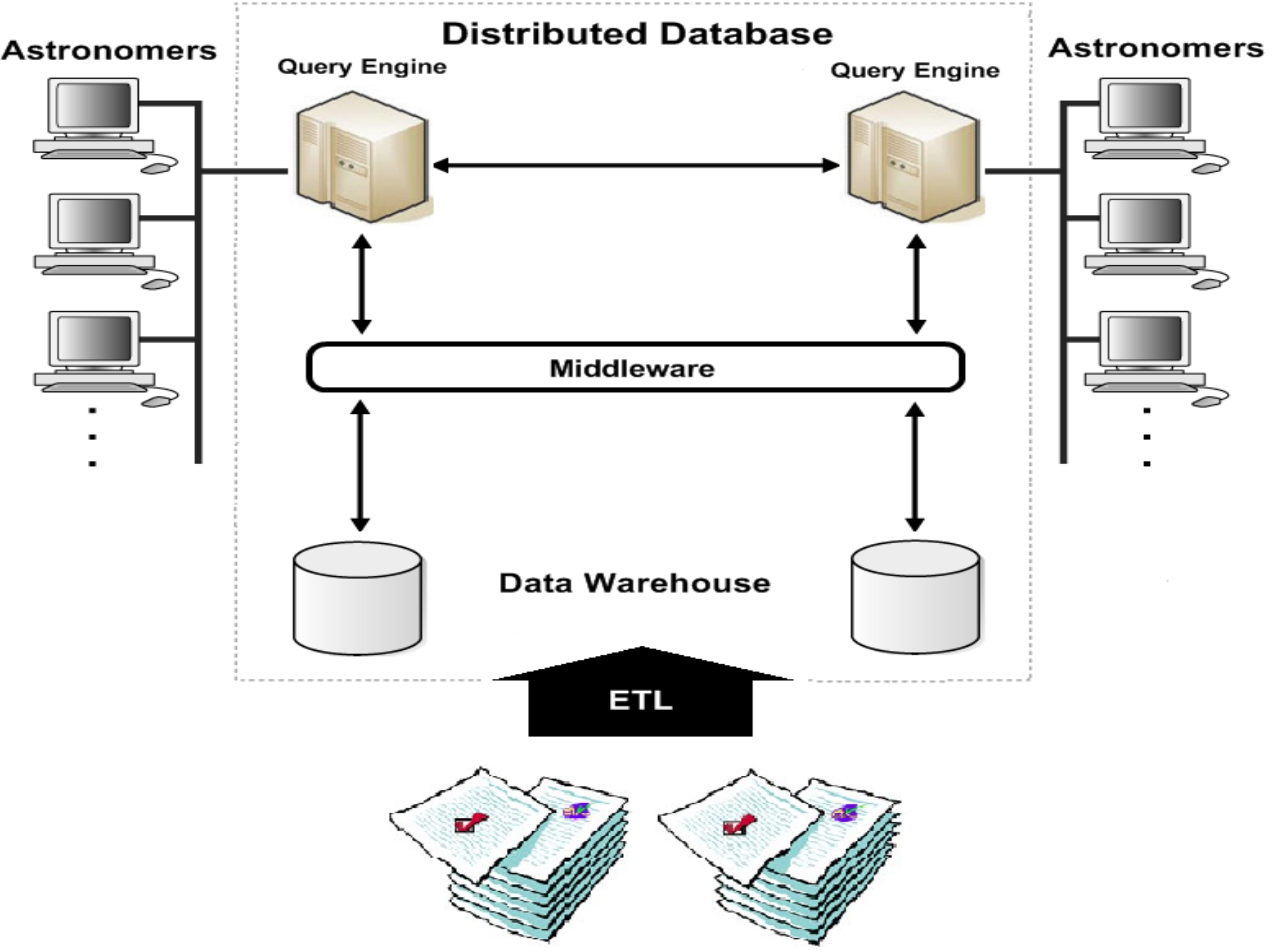
Why Multiple Classifier System

- Multiple Classifier System
  - Divide data into small chunks, and classify the chunks in parallel with multiple similar tools
  - Can deal with large-scale data
  - Can enhance the correctness
  - Can process in parallel

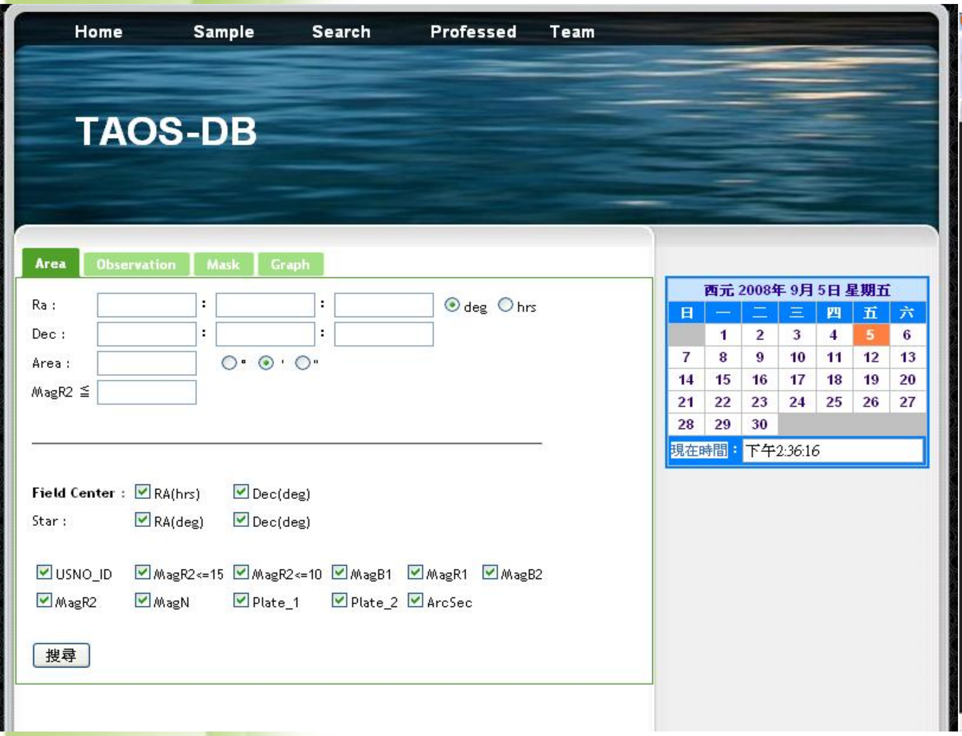
Astronomical Information Management System

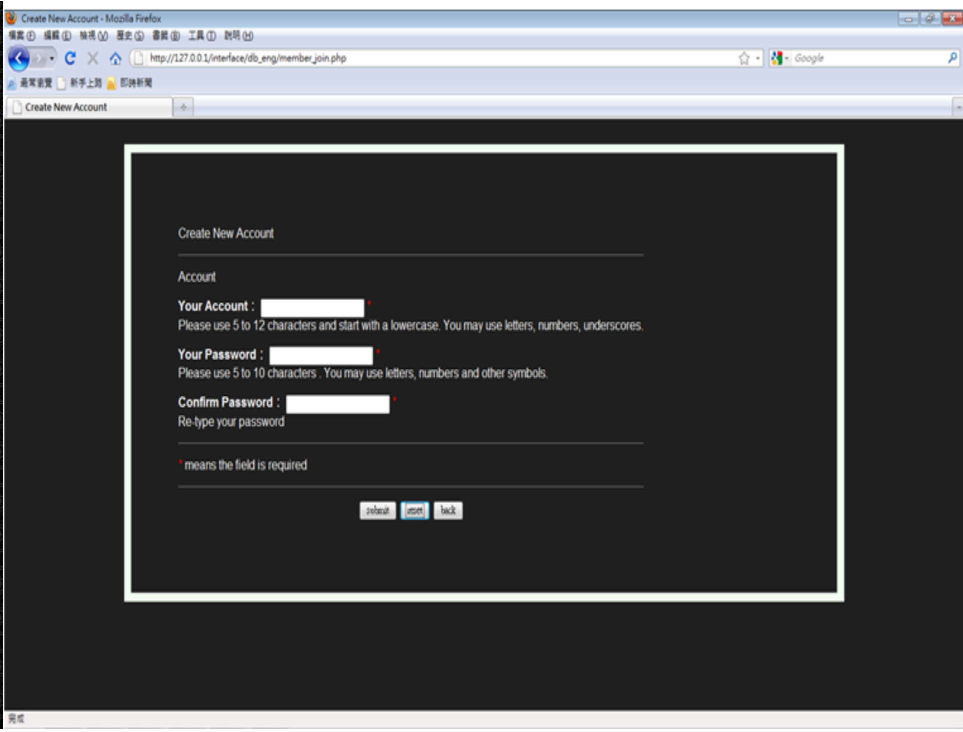
Astronomical Information Management System

- Improving data analysis
  - Data Warehouse design
  - New schema for analysis of large amount of astronomical data
- Managing data in grid environments
  - Distributive and algebraic functions
  - Distributed data storage base on data warehouse

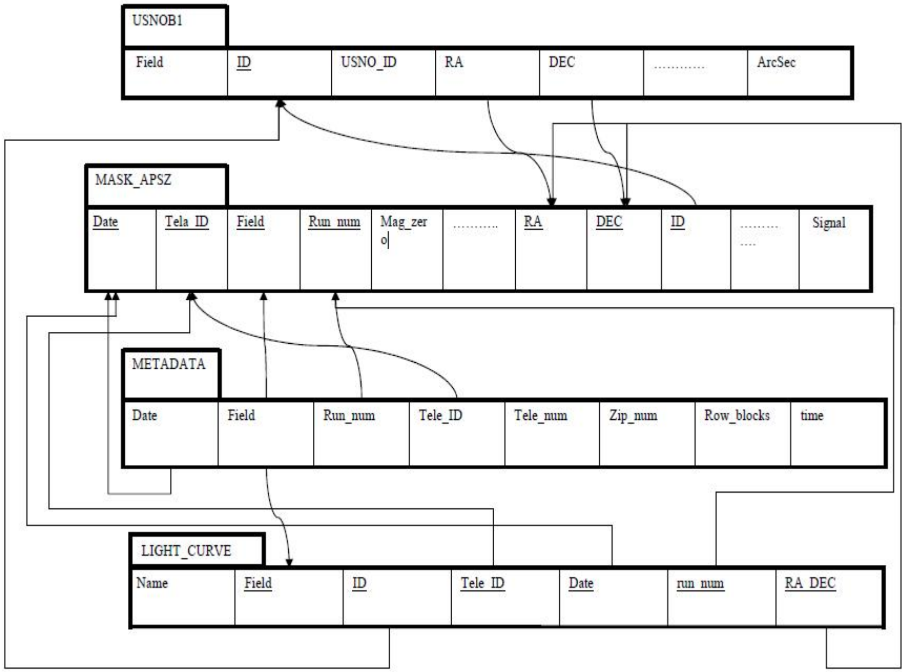

Astronomers
Distributed Database
Query Engine
Query Engine
Astronomers
Middleware
Data Warehouse
ETL

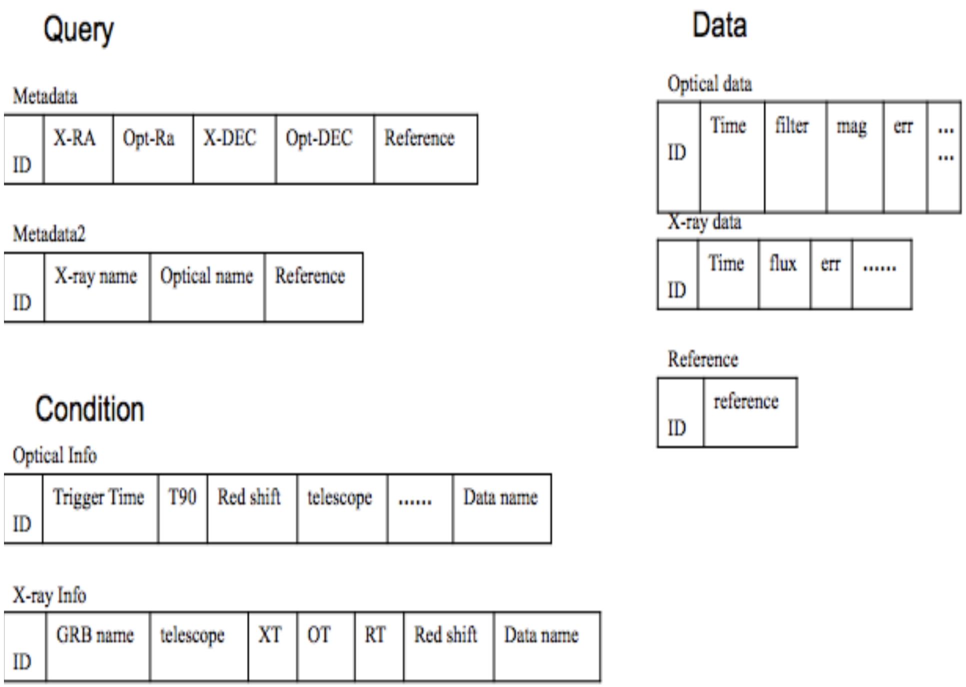
Interface Example



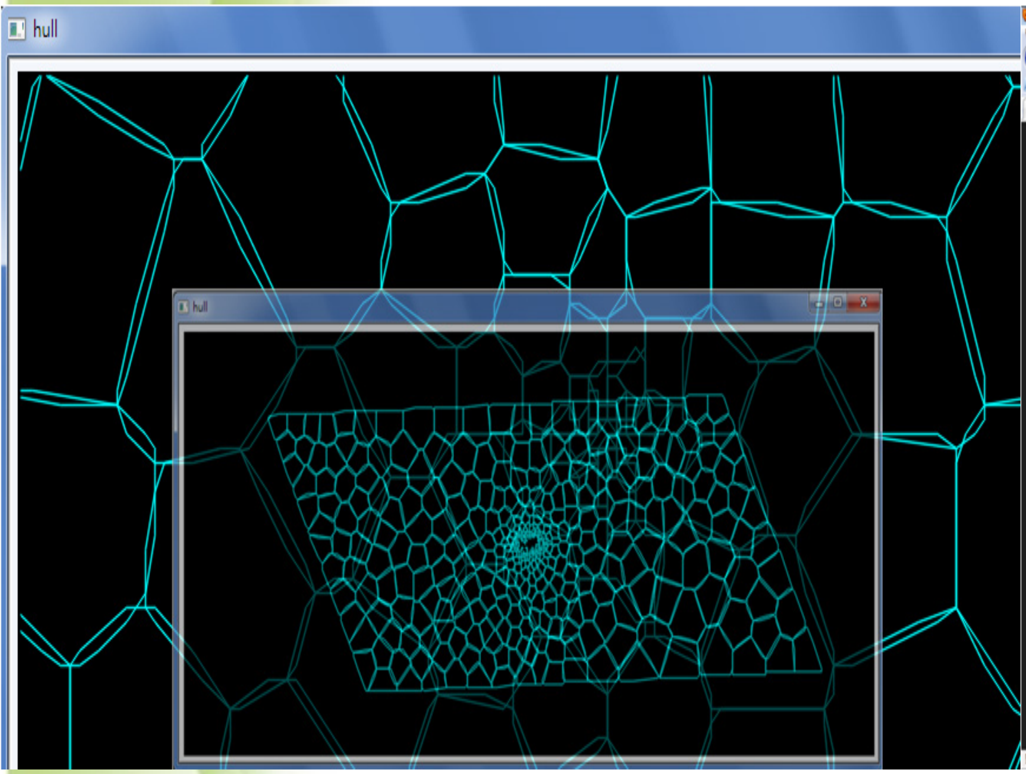


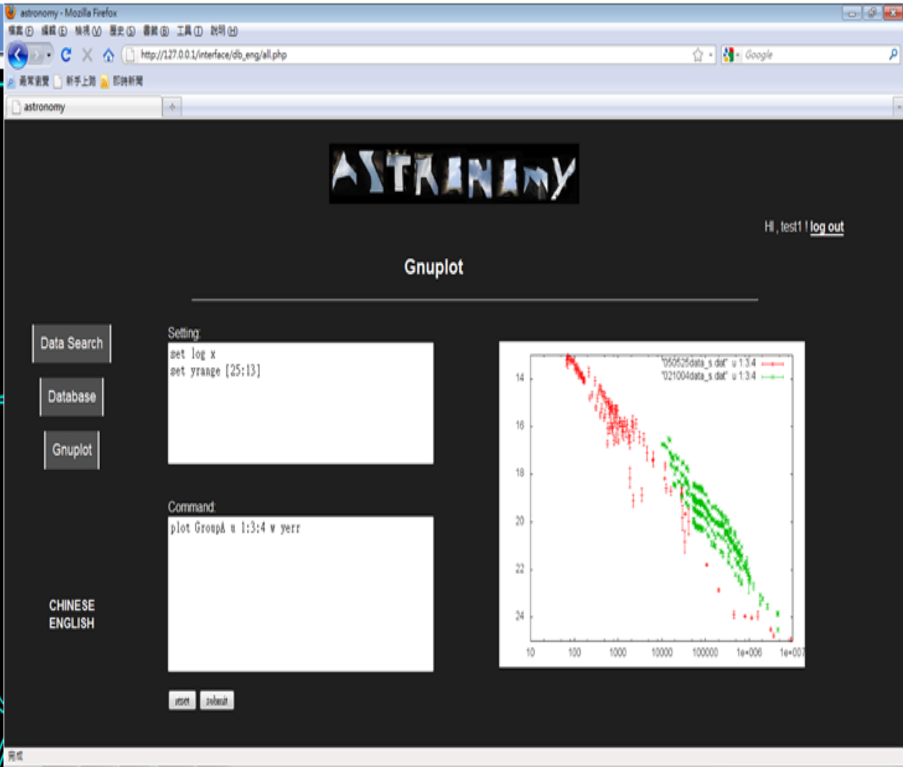
Subject Oriented Schema Example





Analysis Tool Module Example





Conclusions

Conclusions

- Apply parallel computing to astronomical research
  - Develop a apply program to parallel computing
- Refine the process of existing algorithms
  - Speed-up execution
  - Save lots of storage space
- Provide a program template
  - Users can rewrite their similarity functions to fit their needs
- Develop information management system
  - We have a concise, integrated, and scalable platform for fast data retrieval and management

Q & A

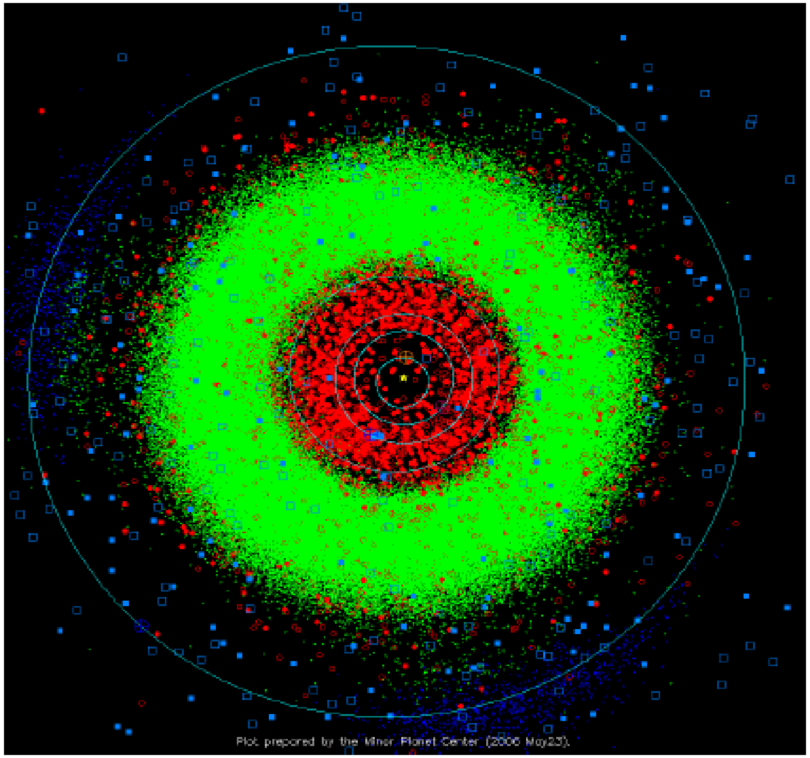
Experimental Results

Experimental Data Set

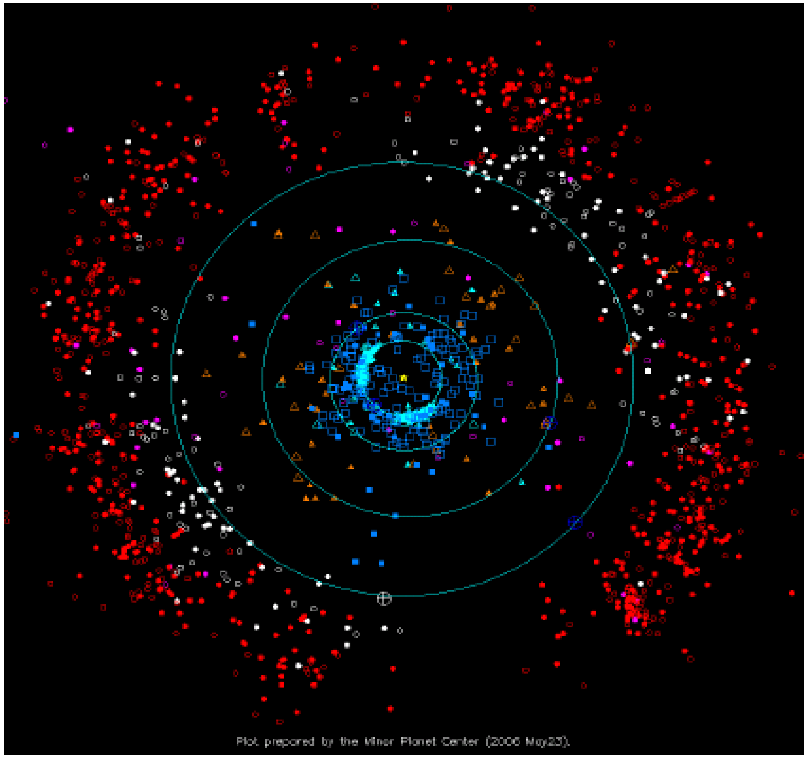
- Asteroid hierarchical clustering
- The MPC Orbit (MPCORB) database
  - Contains 6 orbital elements of minor planets
  - Release date : 2008/12
  - About 370k orbital records
    - Similarity Matrix: 1583.35G
- Similarity function d:

$$\left(\frac{d}{na}\right)^2 = k_a\left(\frac{\delta a}{a}\right)^2 + k_e\,(\delta e)^2 + k_i\,(\delta \sin i)^2 + k_\Omega\,(\delta\Omega)^2 + k_{\bar{\omega}}\,(\delta\bar{\omega})^2$$

Asteroids in the Solar System



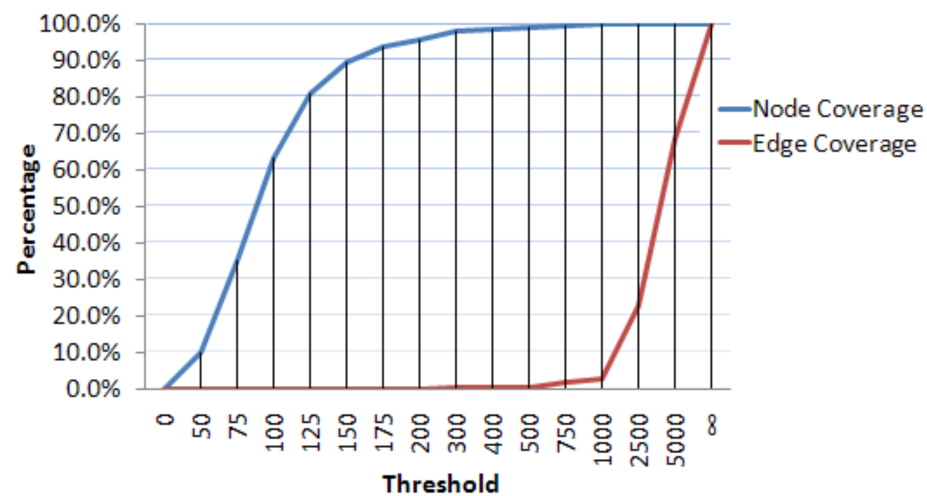
Inner Solar System



Outer Solar System

Experimental Design

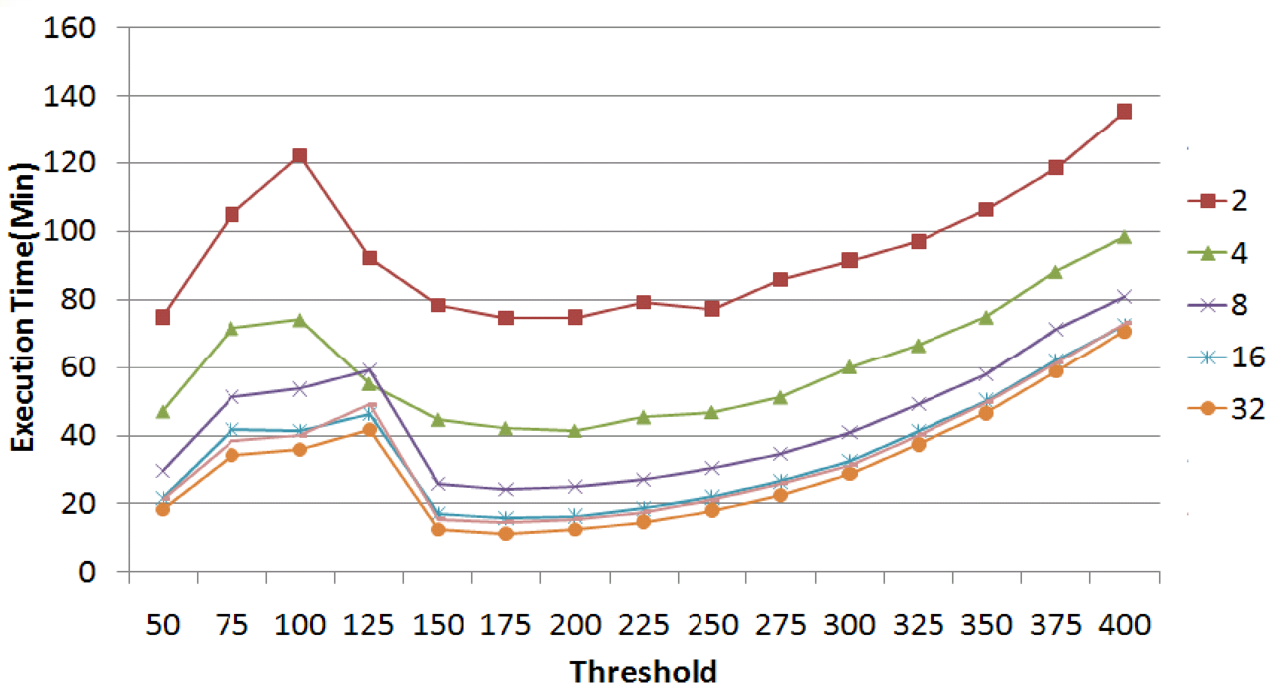
- Observation of the relationship between
  - Threshold
  - Process number
  - Execution time
  - Number of disjoint set

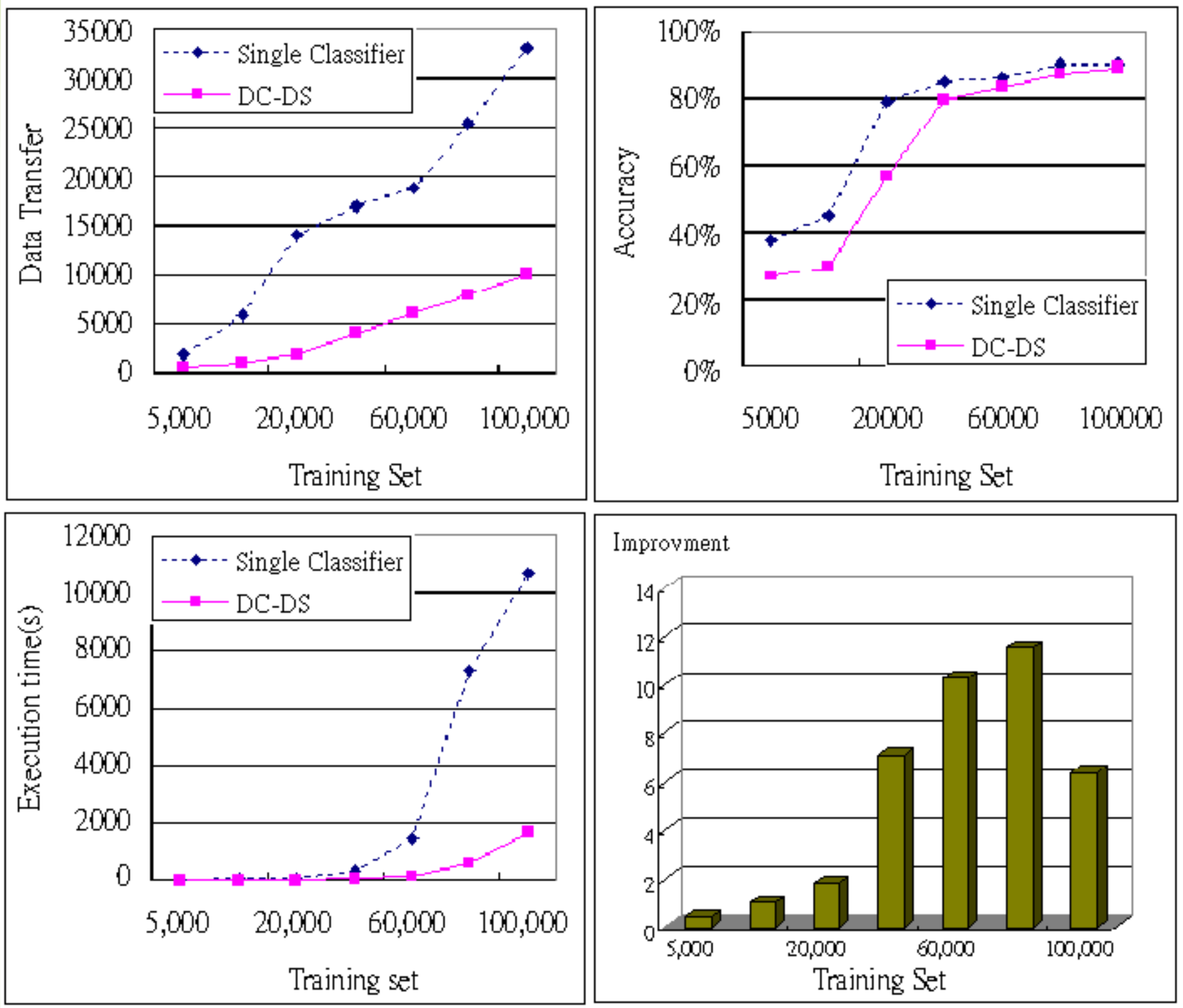


- We use
  - 50,75,100,125,…400 as our observation target

Overall experimental results (cont.)

- Overall execution time vs. threshold using different numbers of processes

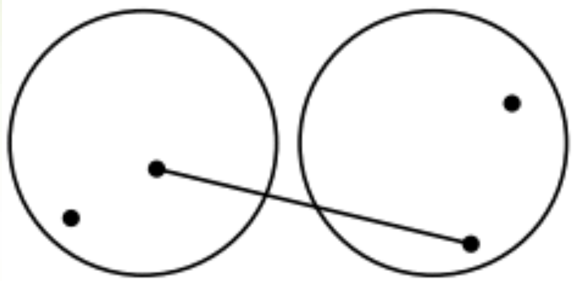



Data Transfer
35000
30000
25000
20000
15000
10000
5000
0
Single Classifier
DC-DS
5,000
20,000
60,000
100,000
Training Set
Accuracy
100%
80%
60%
40%
20%
0%
Single Classifier
DC-DS
5000
20000
60000
100000
Training Set
Execution time(s)
12000
10000
8000
6000
4000
2000
0
Single Classifier
DC-DS
5,000
20,000
60,000
100,000
Training set
Improvment
14
12
10
8
6
4
2
0
5,000
20,000
60,000
100,000
Training Set

Computing similarity of clusters

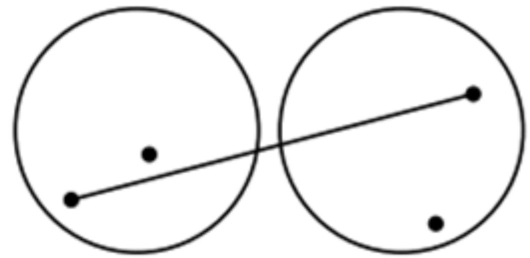
- Single-link

  $S(C_i , C_j) = \min_{a, b} S(a, b)$



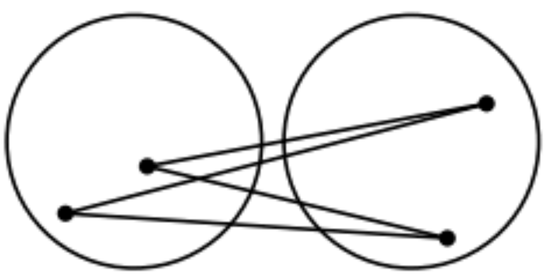
Complete-link

  $S(Ci , Cj) = maxa, b\ S(a, b)$

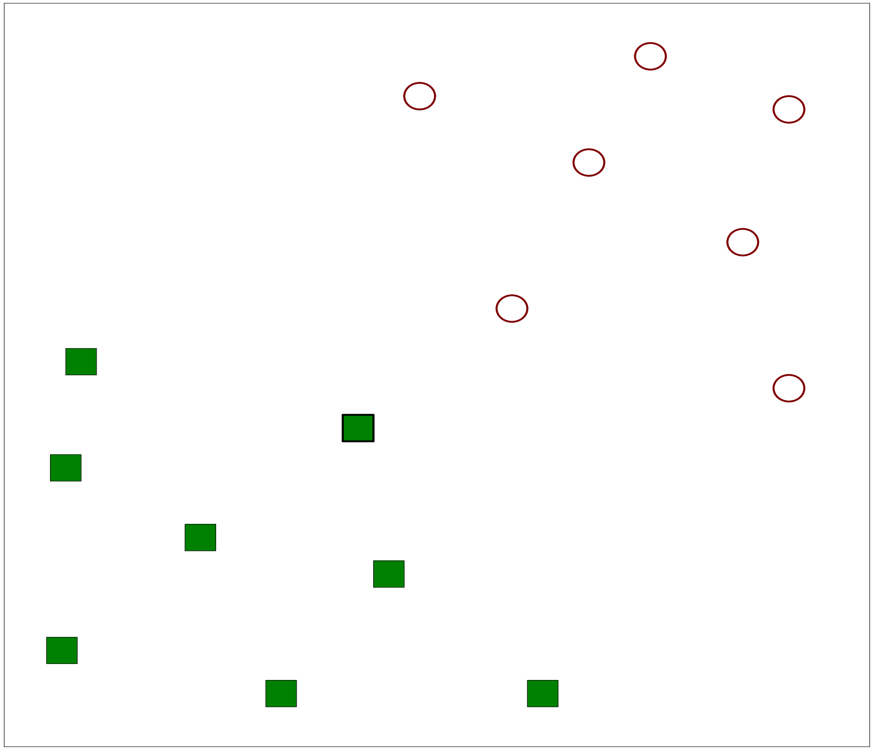


- Average-link

  $S(Ci, Cj) = \Sigma_{a,b} S(a, b) / (|C_i||C_j|)$

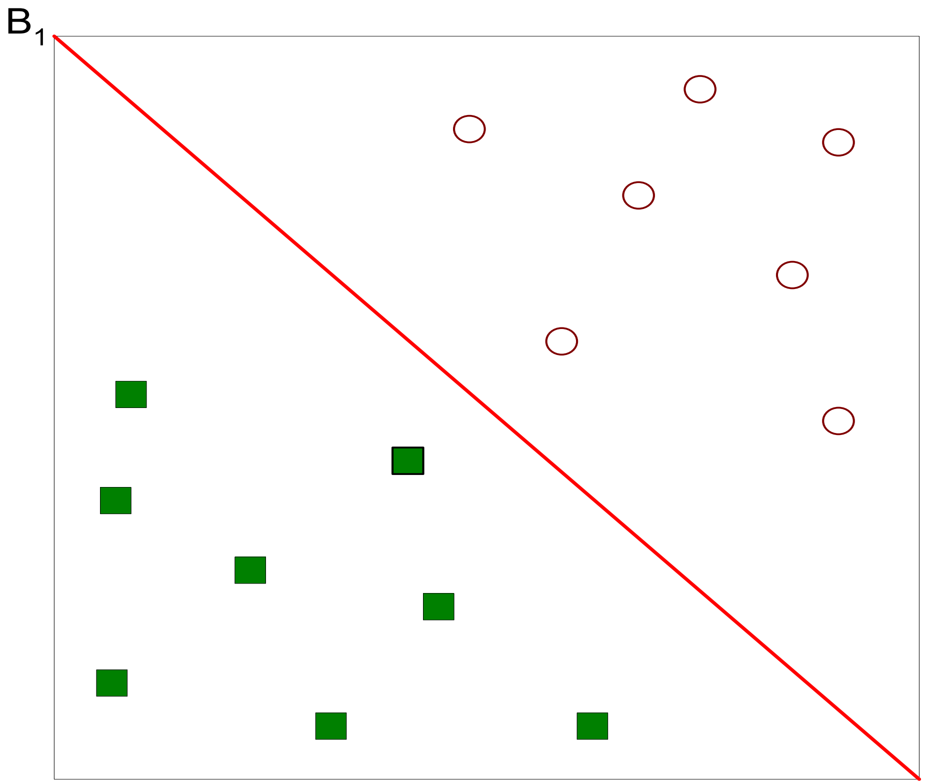


Support Vector Machines



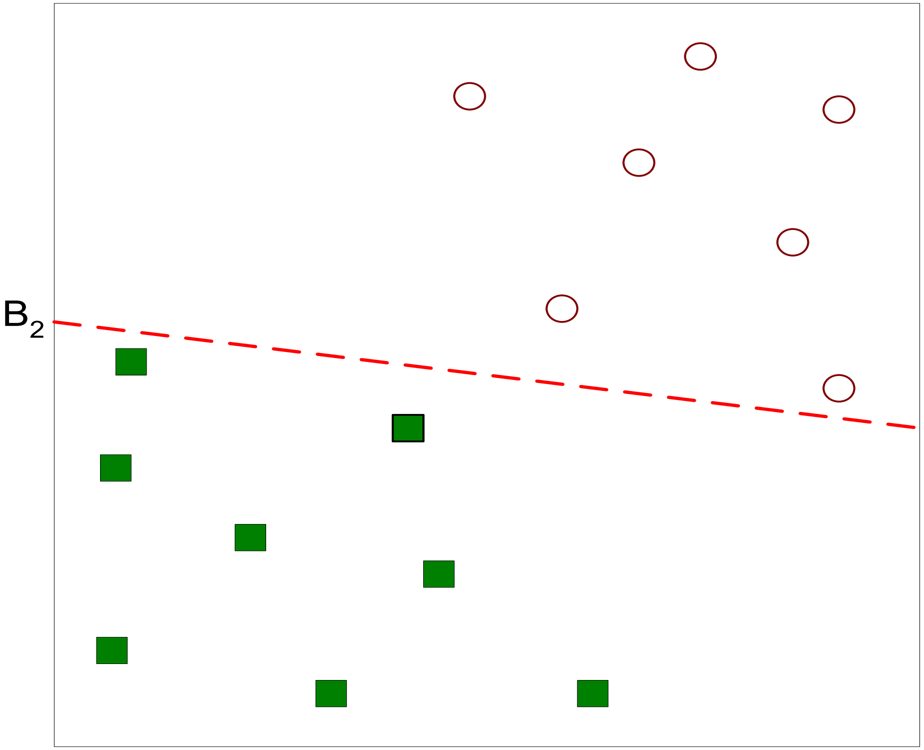
- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



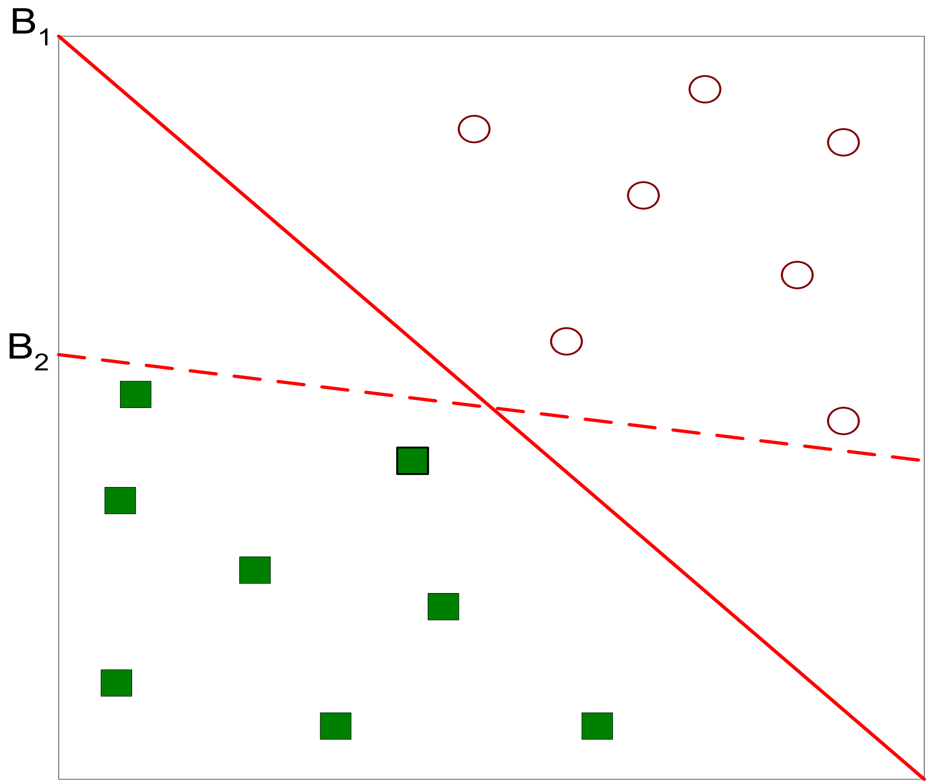
- One Possible Solution

Support Vector Machines



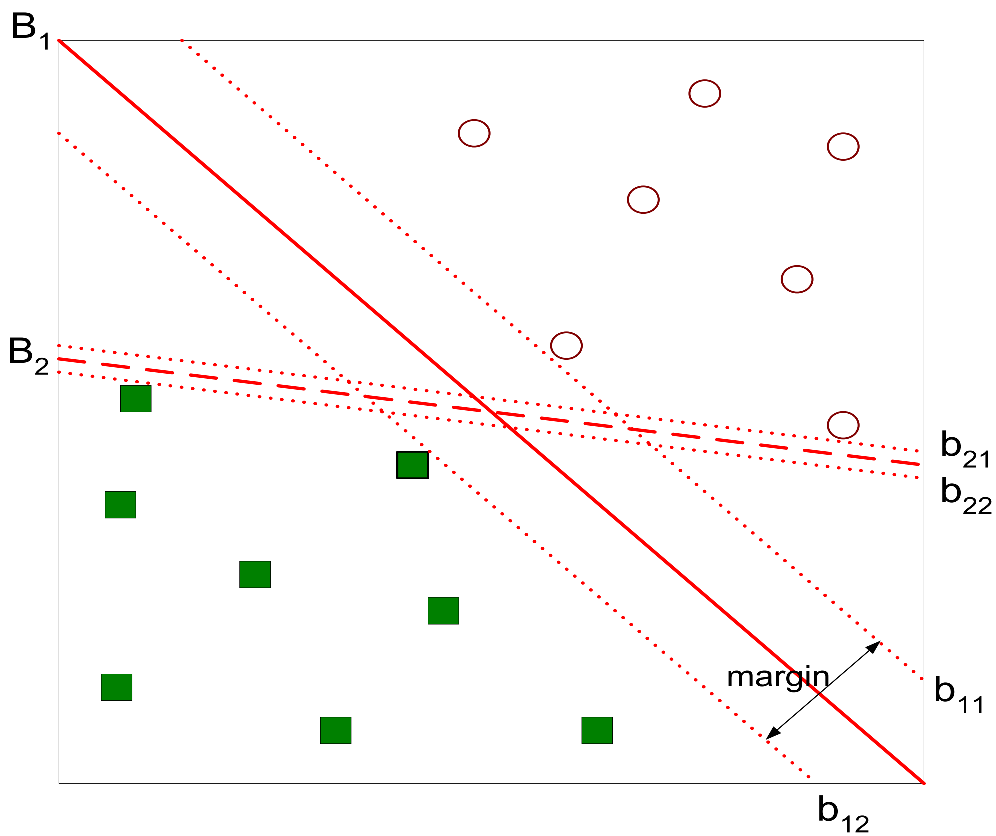
- Another possible solution

Support Vector Machines



- Which one is better? B1 or B2?
- How do you define better?

Support Vector Machines



- Find hyperplane maximizes the margin => B1 is better than B2

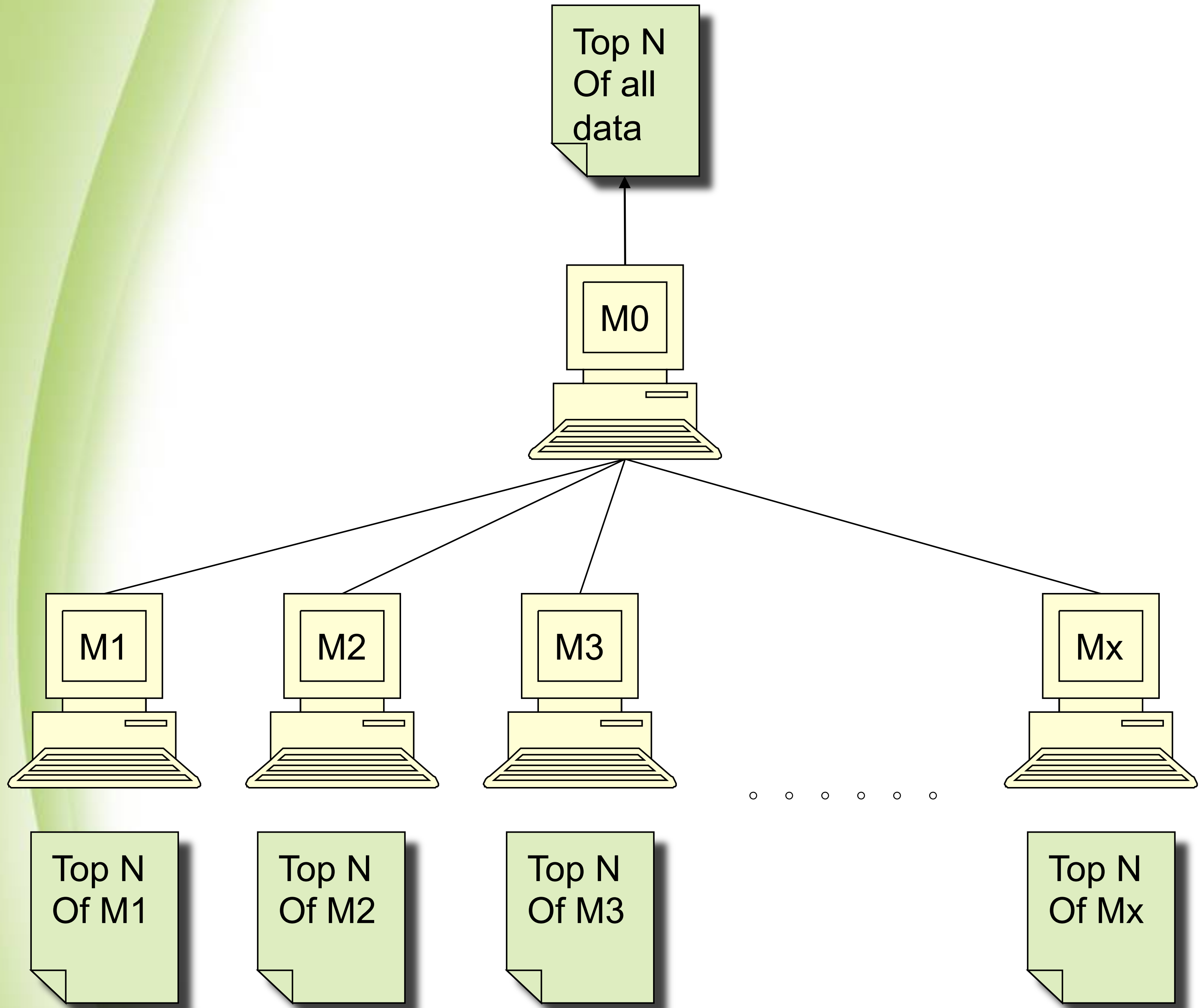
Method for Top-N Query

- Compute the pair distance and store the data base on:
  - threshold
  - Top “N”
- Merge the result

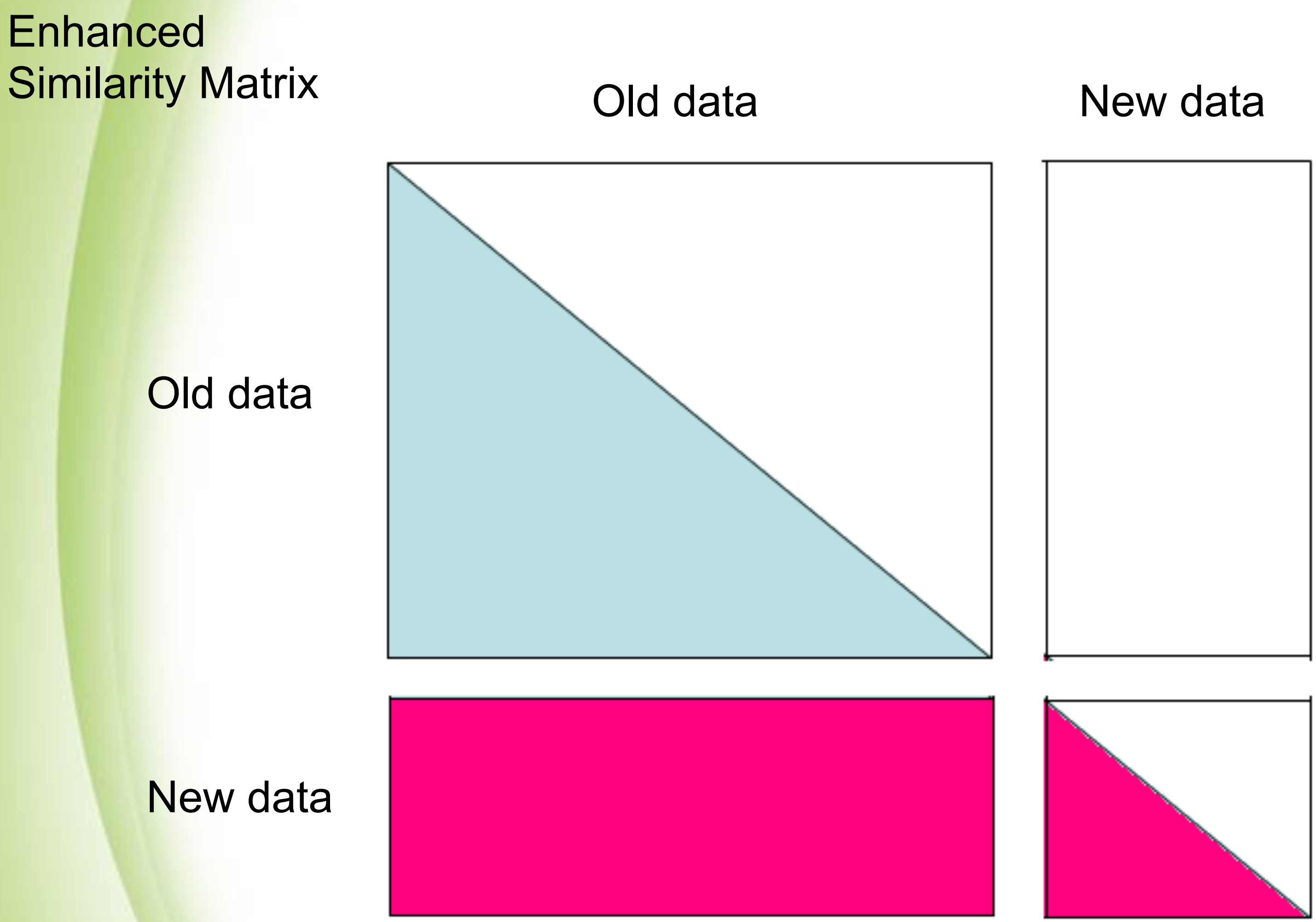
Computing of Similarity Matrix

- Parallelism strategy of Computing similarity matrix in parallel
  - Row-based

	1	2	3	4	5	6	7	8	9
1									
2	0.5408								
3	0.5099	0.4031							
4	2.5401	2.0000	2.1914						
5	2.3817	1.8721	1.9448	0.6519					
6	3.0232	2.5303	2.5612	1.0594	0.6800				
7	4.0311	3.8200	3.5510	3.5710	2.9364	2.6627			
8	5.0039	4.7066	4.5011	3.9588	3.4077	2.9154	1.2206		
9	5.5145	5.2690	5.0249	4.6906	4.1149	3.6619	1.5033	0.7810	

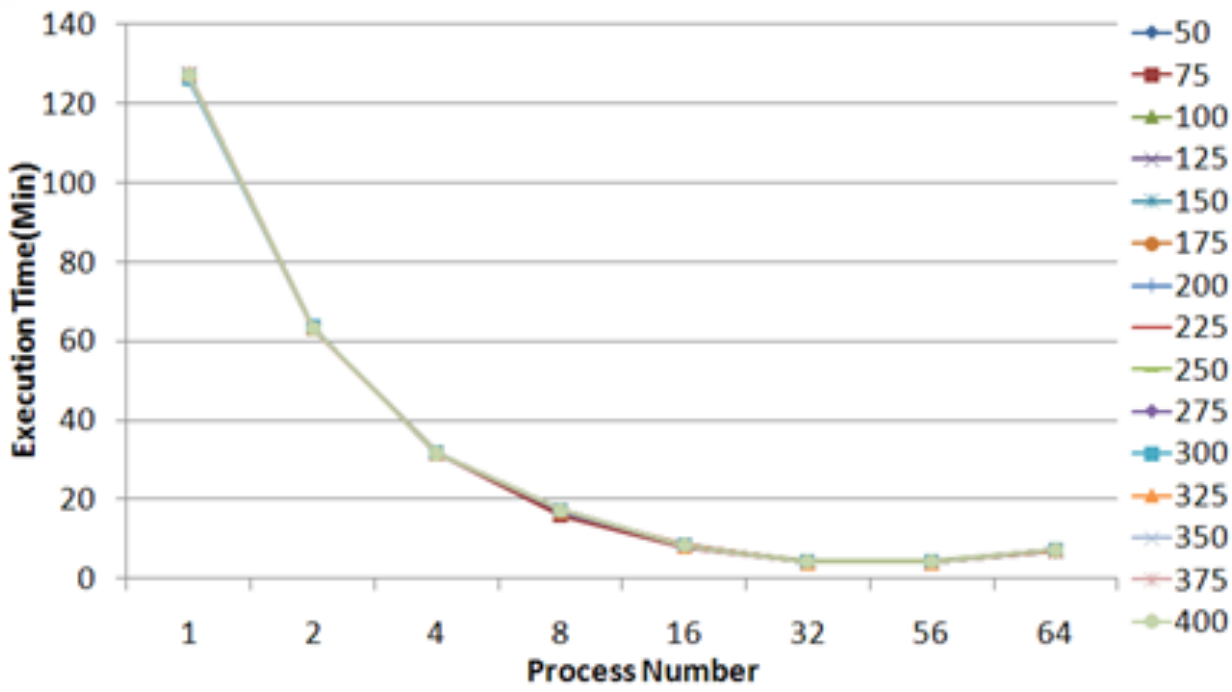

Top N
Of all
data
M0
M1
M2
M3
Mx
Top N
Of M1
Top N
Of M2
Top N
Of M3
Top N
Of Mx

Compute the Distance of New Data



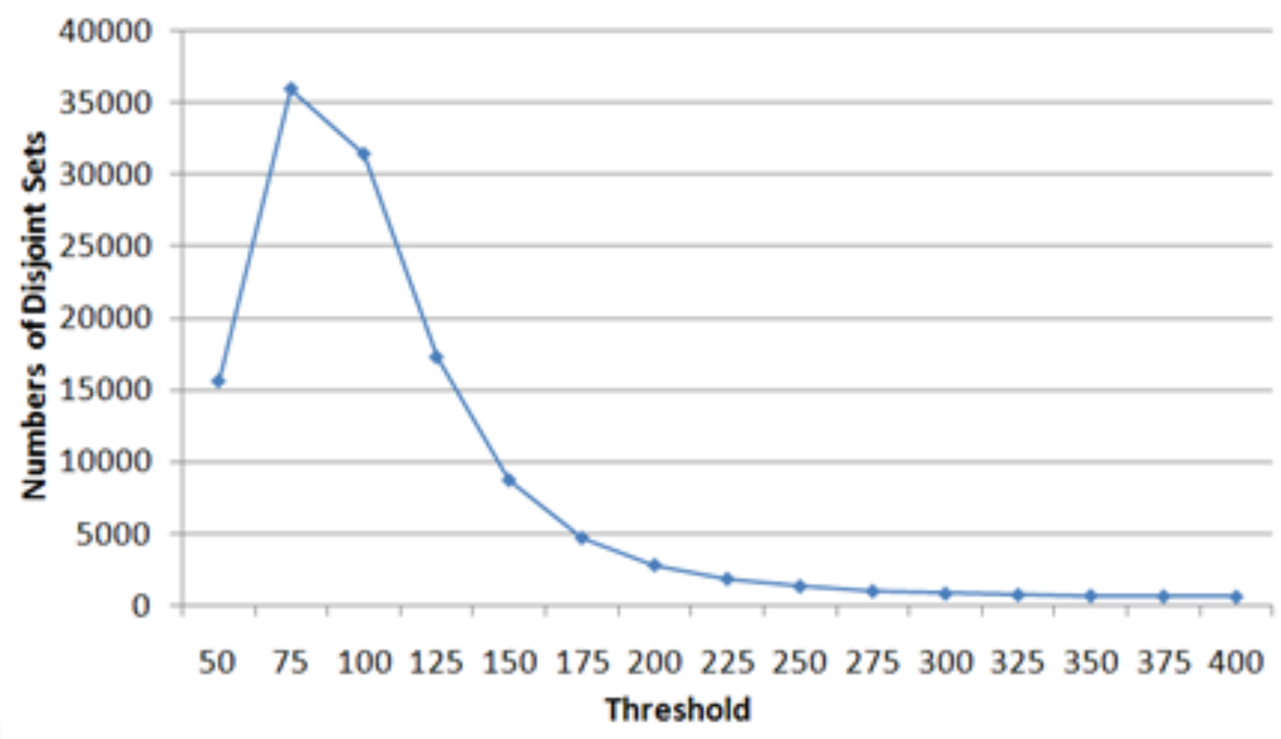
Experiments for stage 1

- Execution time of computing the similarity matrix vs. number of processes



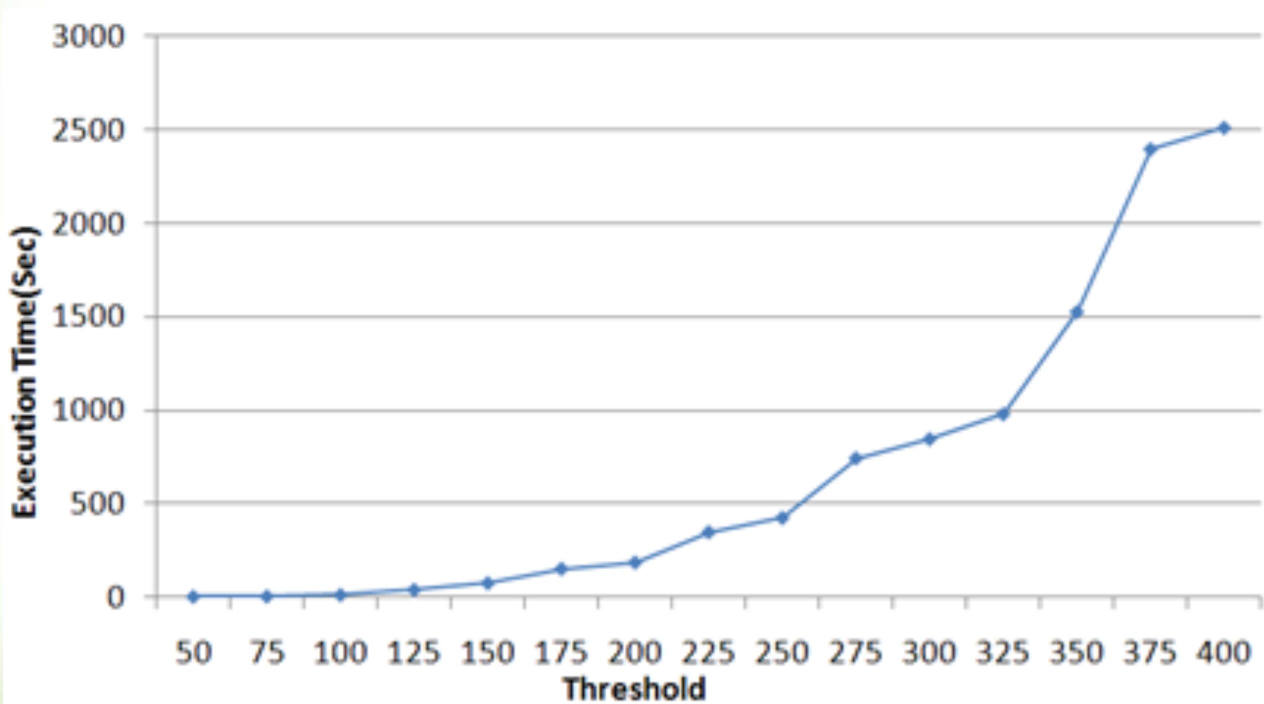
Experiments for stage 2

- Threshold vs. number of disjoint sets using a single processor



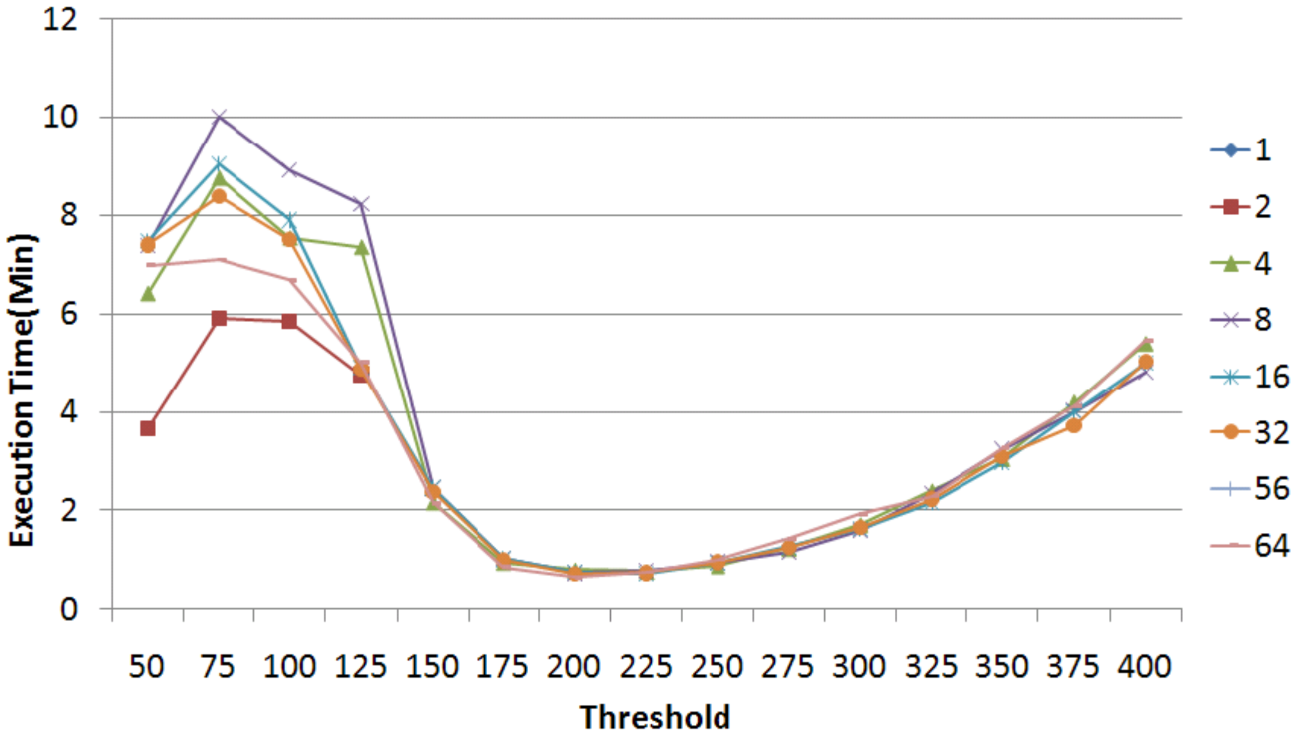
Experiments for stage 2 (cont)

- Threshold vs. execution time while identifying the disjoint sets using a single processor



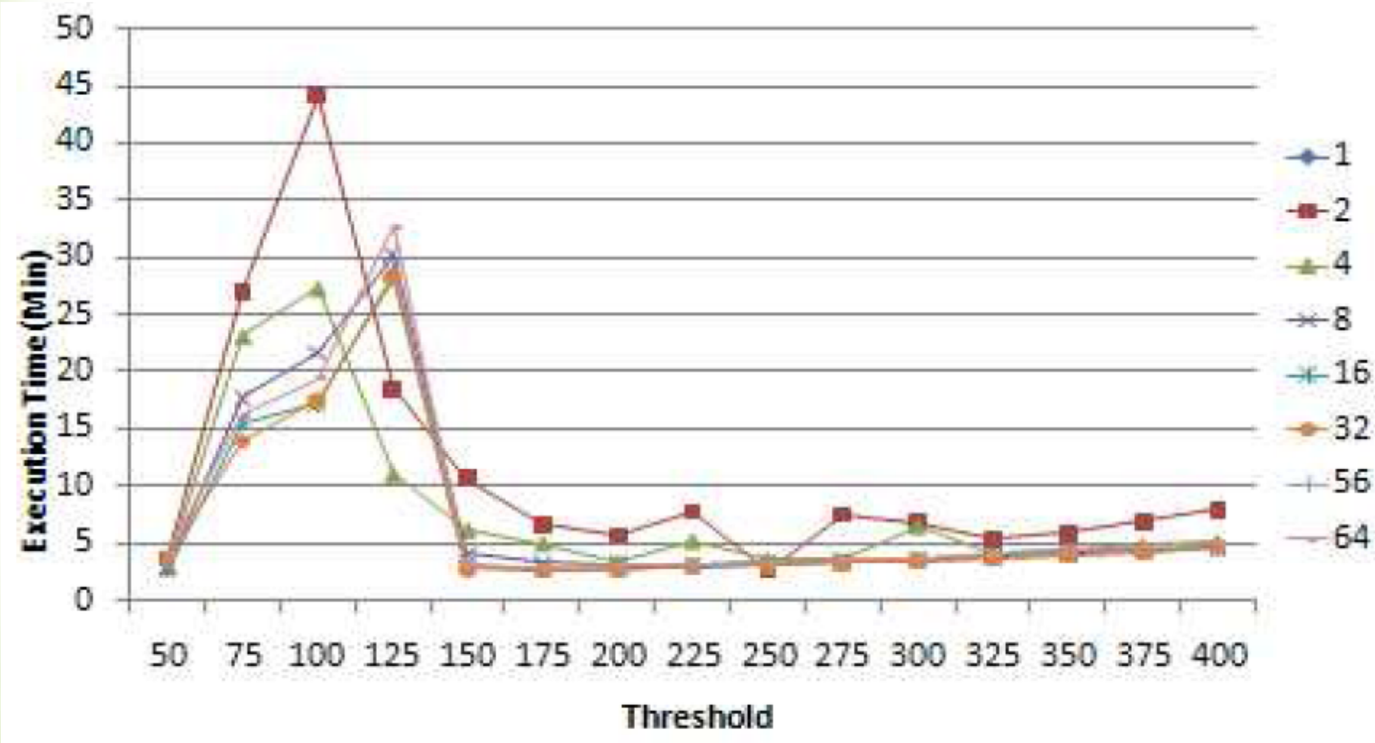
Experiments for stage 4 (cont)

- Threshold vs. execution time using different numbers of processes while clustering the disjoint sets



Experiments for stage 3

- Execution time vs. number of processes while computing the similarity matrix of the disjoint sets



Clustering Algorithms (cont)

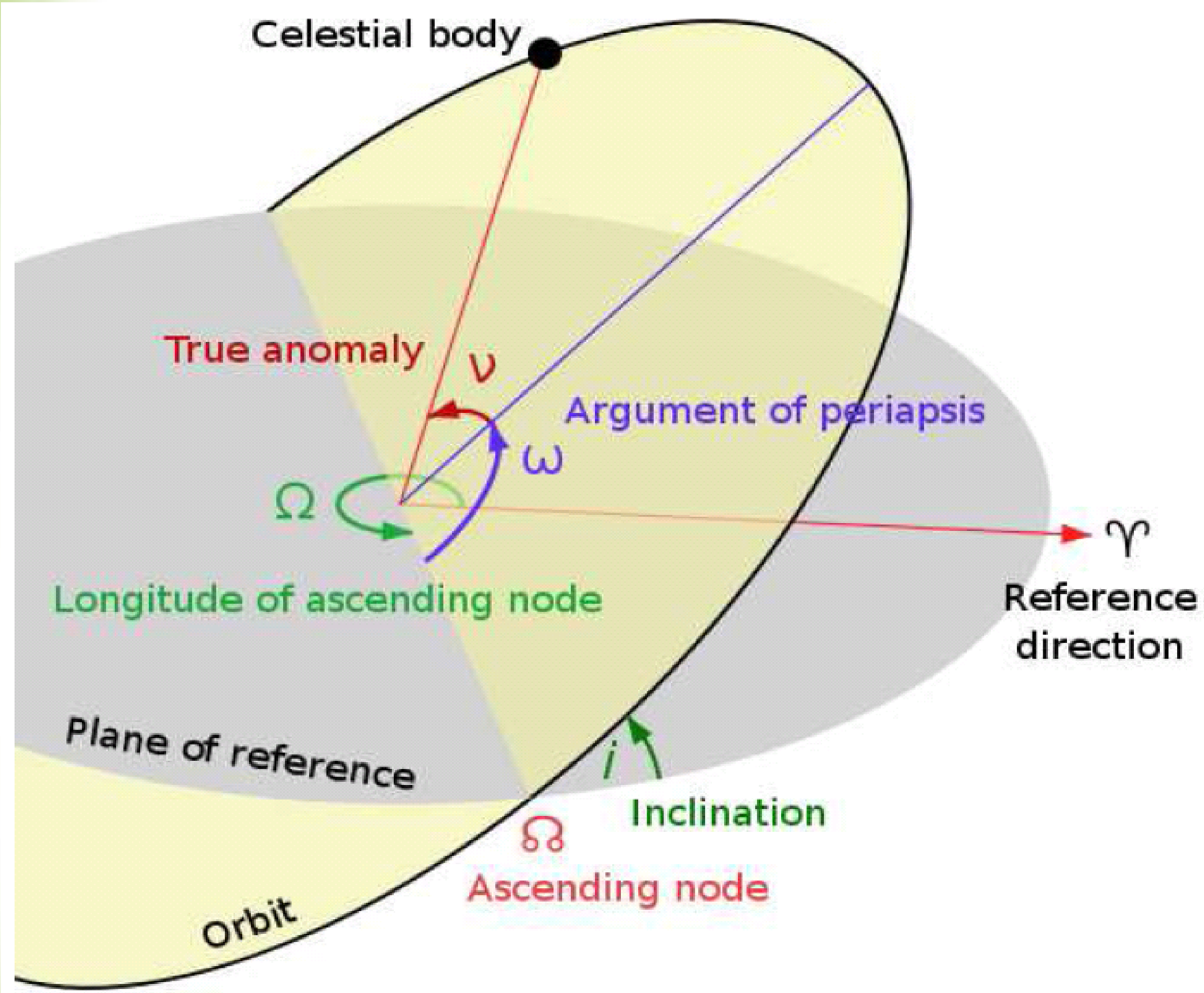
- Partition based clustering
  - K-Means

	Partition-Based Clustering	Hierarchical Clustering
Need to assign number of cluster	Yes	No
Easy to parallelize	Yes	No
unique	No	Yes
Structure Output	No	Yes

MOPS Project

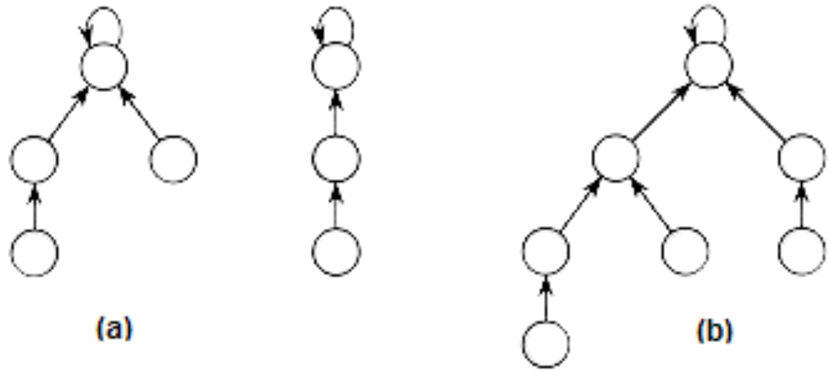
- MOPS:
  - Moving Objects Processing System
  - A sub-system in Pan-STARRS project
  - Identify/determine orbits of Near-Earth Objects
  - Over 800k objects have been marked now

Orbital Elements

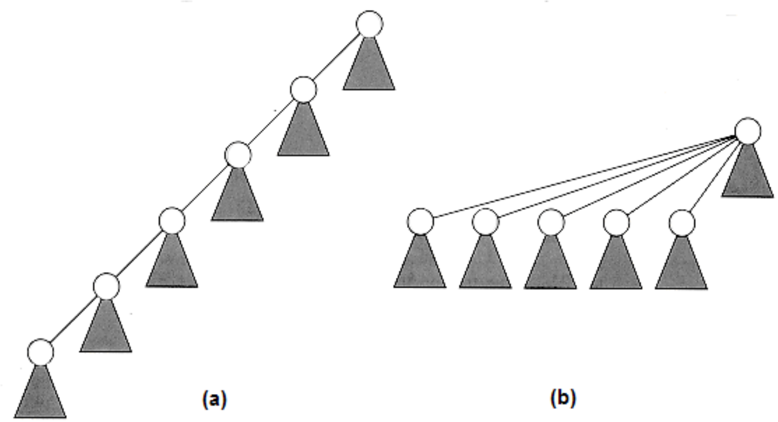


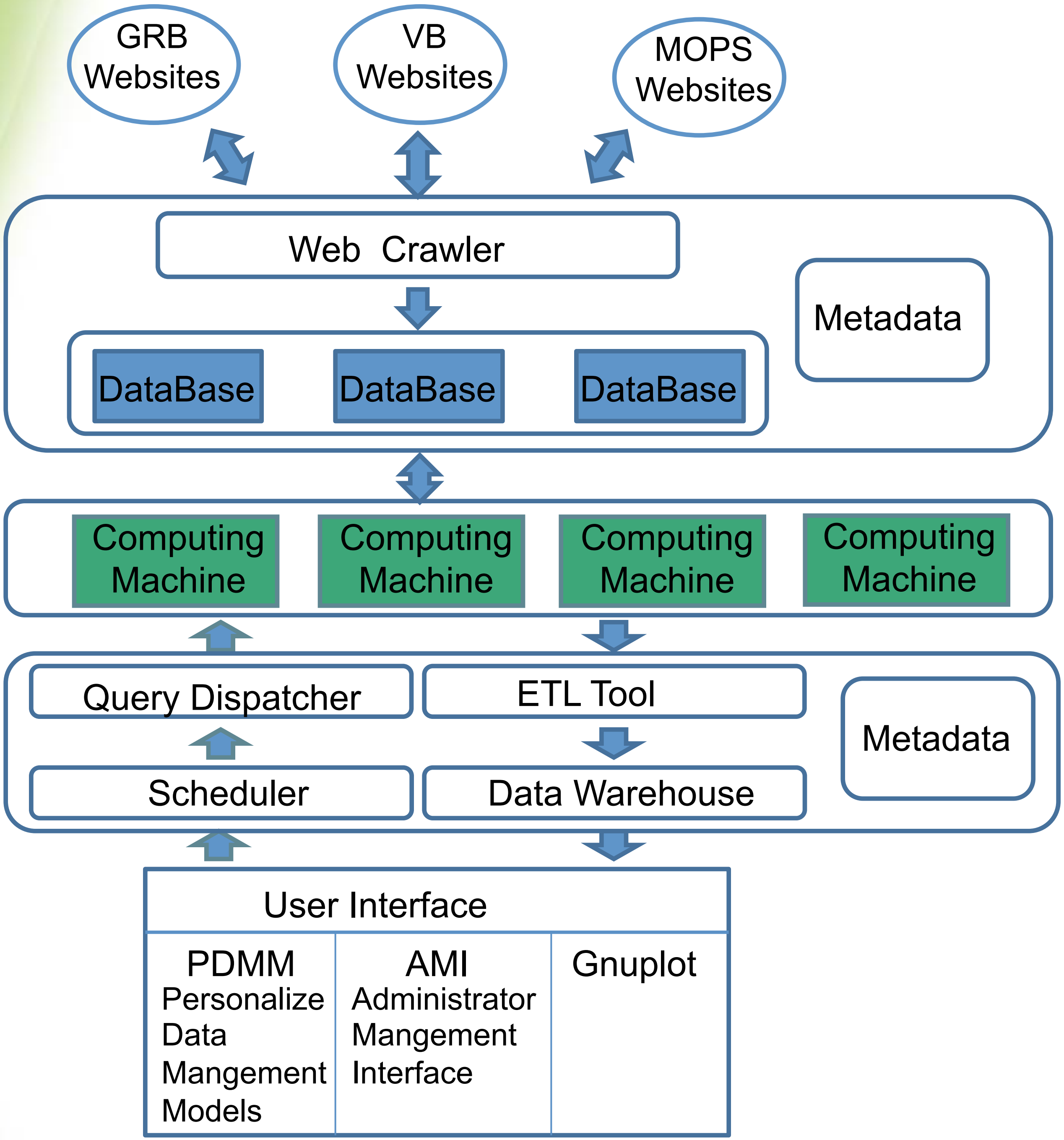
Disjoint Set Operations

- Example



- Optimization
  - Union by rank
  - Path compress
  - Time Complexity
    - θ (m α (n)), α (n)=4
    - m is the operation counts of union, find root and link




GRB Websites
VB Websites
MOPS Websites
Web Crawler
Metadata
DataBase
DataBase
DataBase
Computing Machine
Computing Machine
Computing Machine
Computing Machine
Query Dispatcher
ETL Tool
Metadata
Scheduler
Data Warehouse
User Interface
PDMM Personalize Data Mangement Models
AMI Administrator Mangement Interface
Gnuplot