Mux-Kmeans: Multiplex Kmeans for Clustering Large-scale Data Set

Chen Li, **Yanfeng Zhang**, Minghai Jiao, Ge Yu

Northeastern University, China

Northeastern University

BIG DATA
IN A SINGLE DAY ONLINE

ENOUGH INFORMATION IS CONSUMED TO FILL
168 MILLION DVDS

294bn E-MAILS ARE SENT

MINUTES SPENT ON FACEBOOK **4.7M**

2 MILLION BLOG POSTS ARE WRITTEN

VIDEO UPLOADED TO YOUTUBE **864,000 HRS**

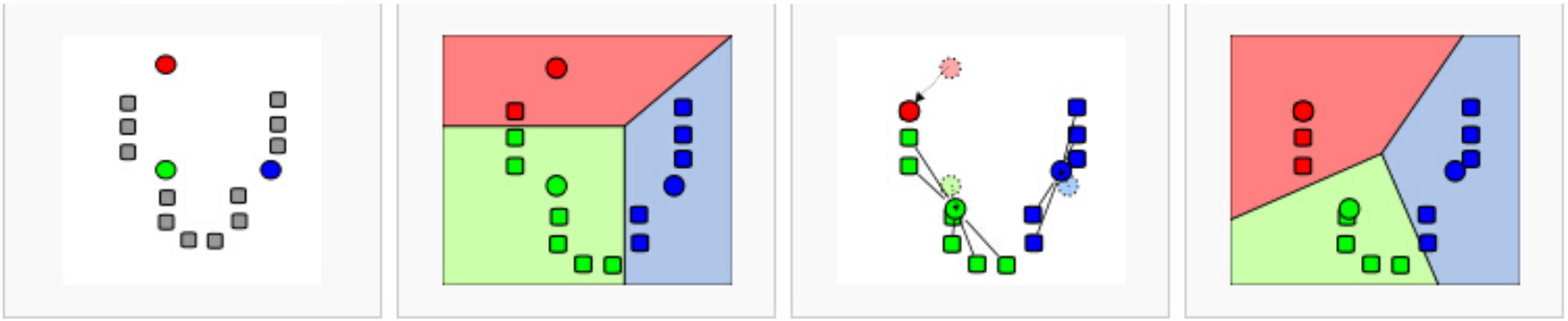
MORE IPHONES ARE SOLD **THAN BABIES BORN**

Clustering

Kmeans

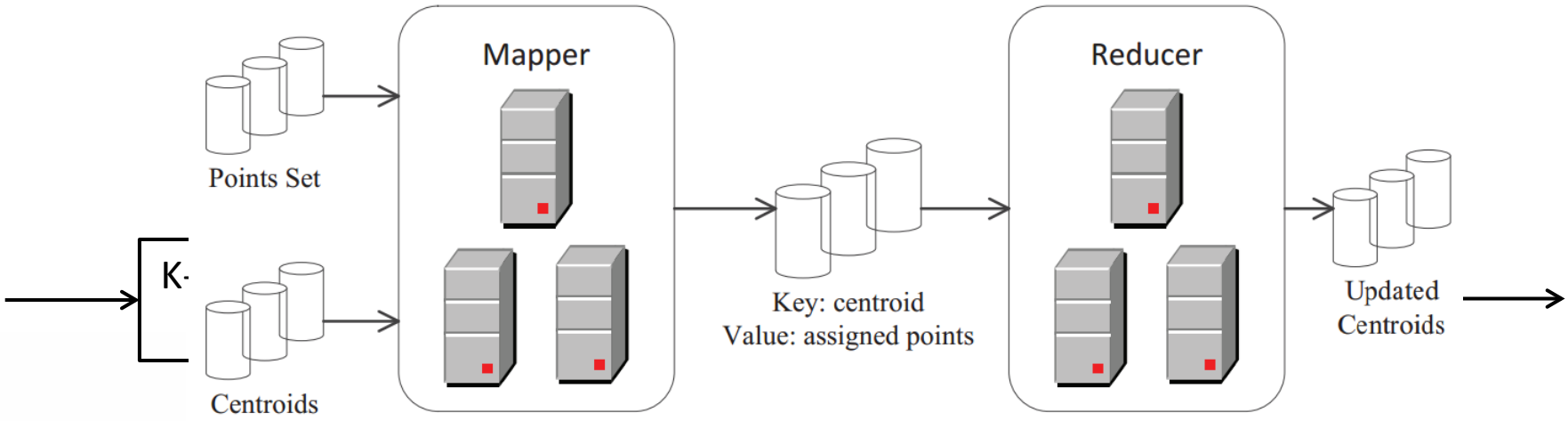
Kmeans

- Kmeans: accept *K* center patterns and a data set, divide the set into *K* clusters
- Goal:
  - ✓ 1. similar – data patterns in same cluster;
  - ✓ 2. dissimilar – data patterns in different clusters.



http://en.wikipedia.org/wiki/K-means_clustering

Kmeans on MapReduce

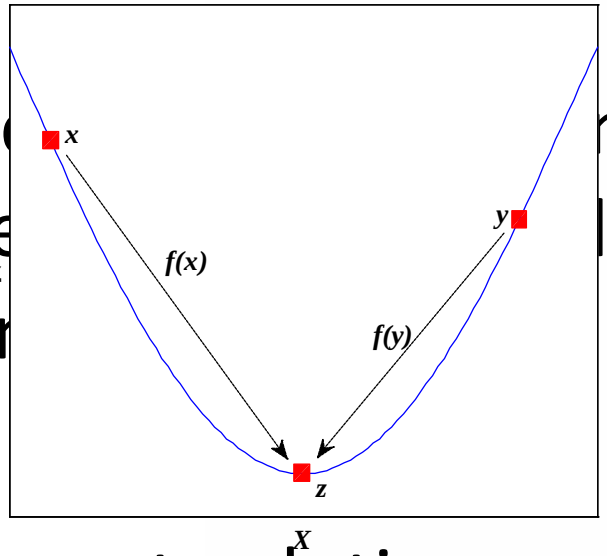


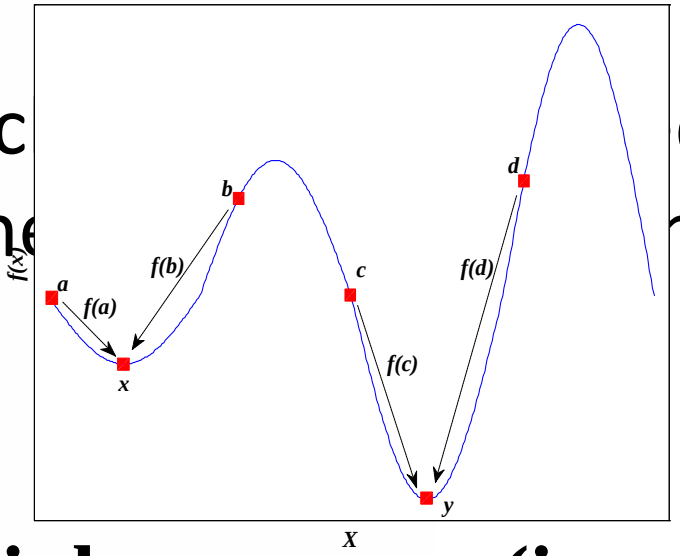
procedure MAPPHASE($< i,\ x_i >$)
  for $c_j, j \in \{1, \ldots, k\}$ **do**
    $j \leftarrow$ *find the closest centroid* c_j *to* x_i, $j \in \{1, \ldots, k\}$
  end for
  output $< j,\ x_i >$
end procedure

procedure REDUCEPHASE($< j,\ [x_i] >$)
  $c_j \leftarrow average\ [x_i]$
  output $< j,\ c_j >$
end procedure

Shortcoming of Kmeans

- Th ns c d by th l th itial cer





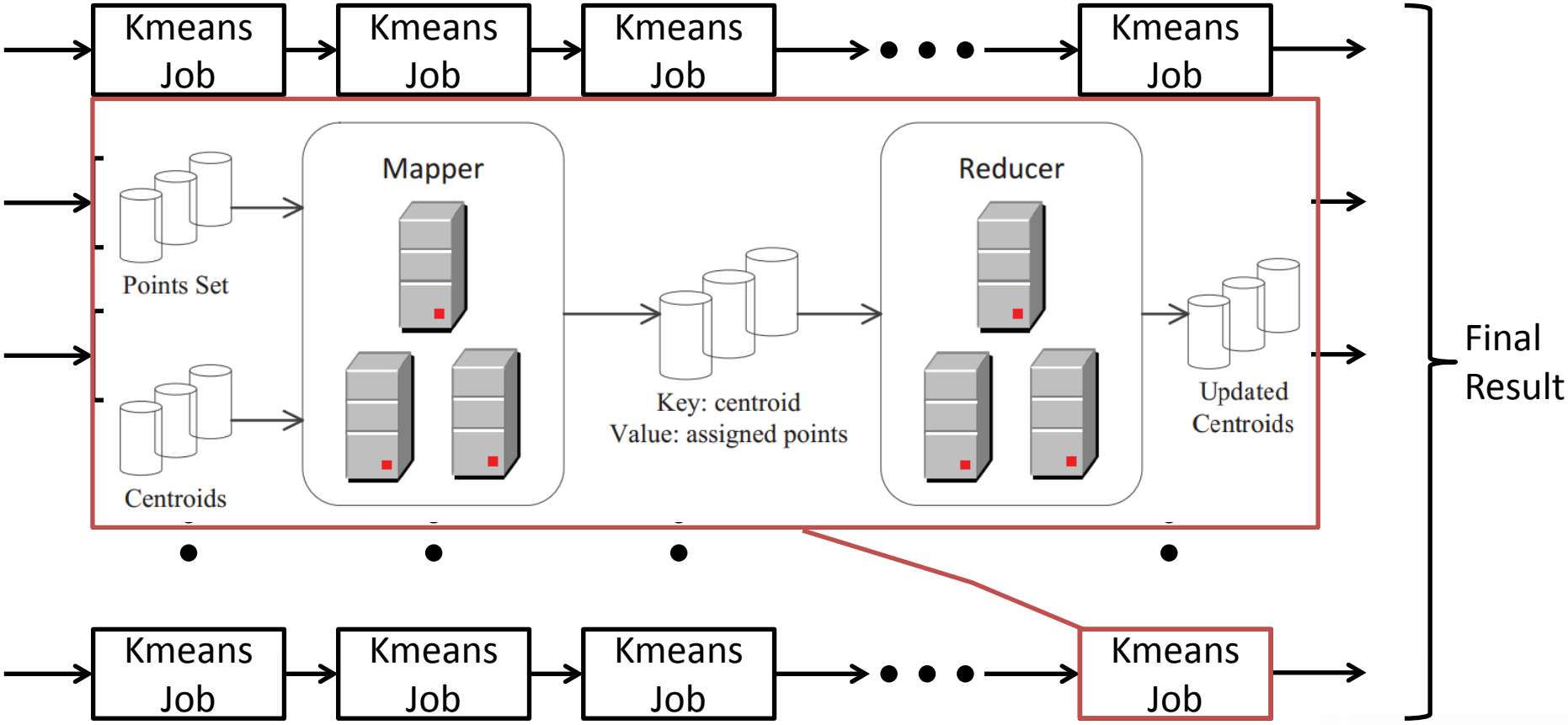
- Current solution: **multiple attempts (in series)**
  - Start from multiple groups of initial centroids
  - Execute multipl kmeans processes, obtain multiple loc
  - Pick the one y

Efficiency Problem

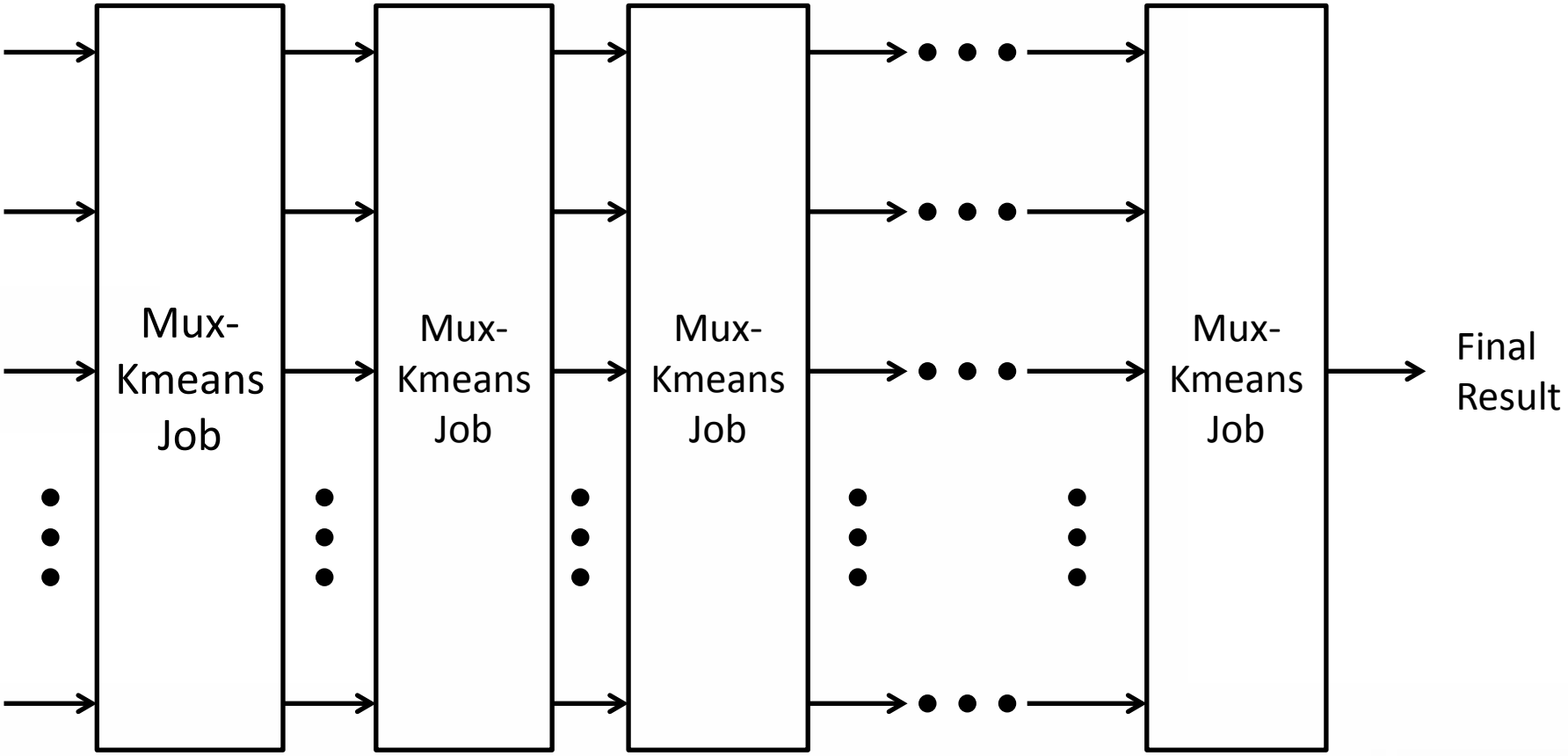
Mux-Kmeans

- Idea:
  - Execute multiple Kmeans attempts in parallel
  - Share states across different Kmeans processes
  - Terminate “hopeless” attempts in early stage
  - Expand searching scale and try more attempts
- Goal:
  - Guarantee the clustering quality
  - Decrease the runtime

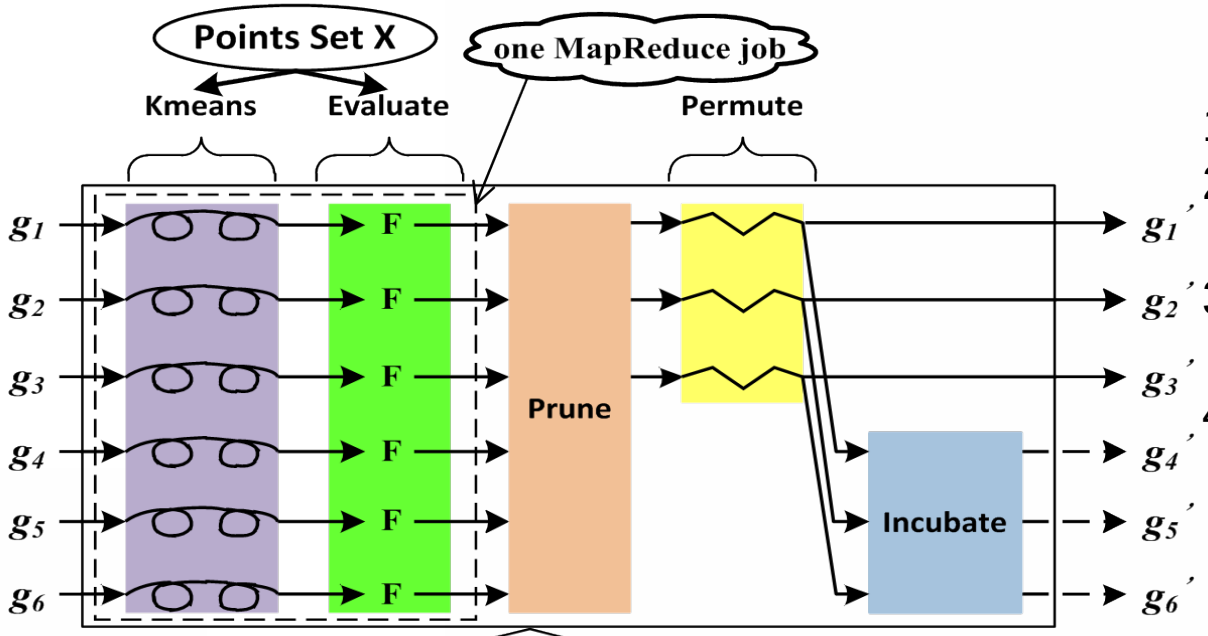
Naive Kmeans *VS.* Mux-Kmeans



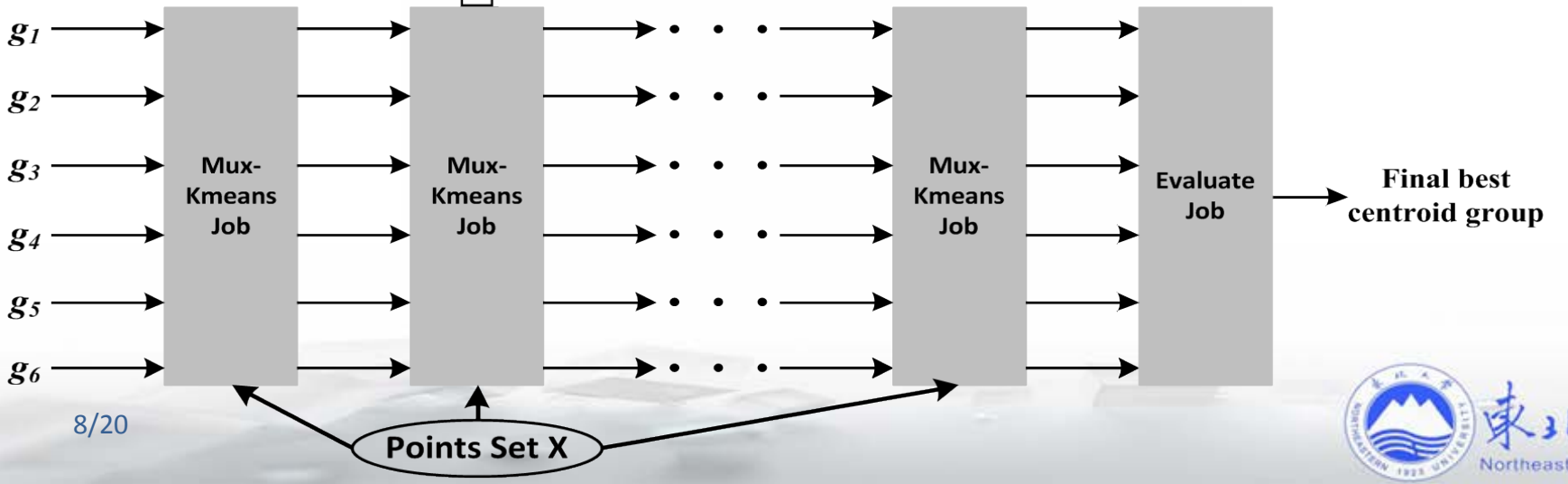
Naive Kmeans *VS.* Mux-Kmeans



Inside Mux-Kmeans Job

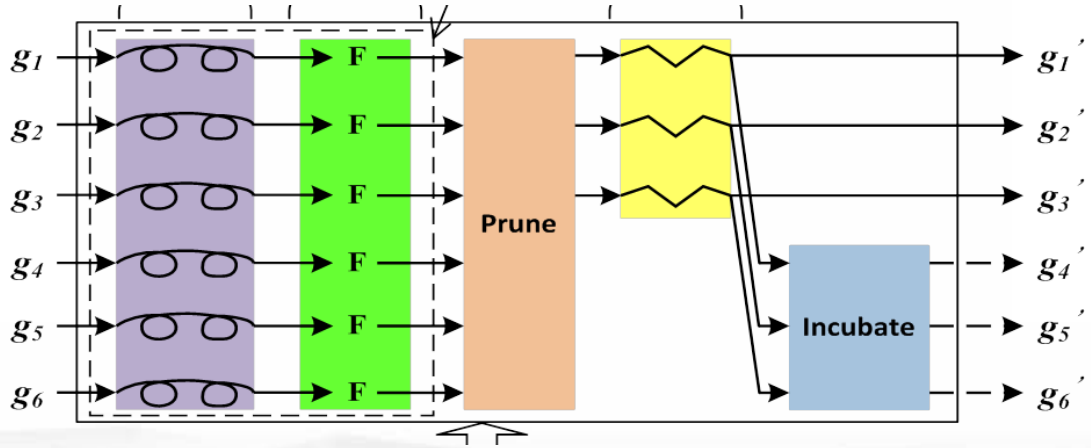


1. Kmeans clustering
2. Evaluate intermediate result's quality
3. Prune some Kmeans attempts with low quality
4. Incubate new groups of centroids for next iteration



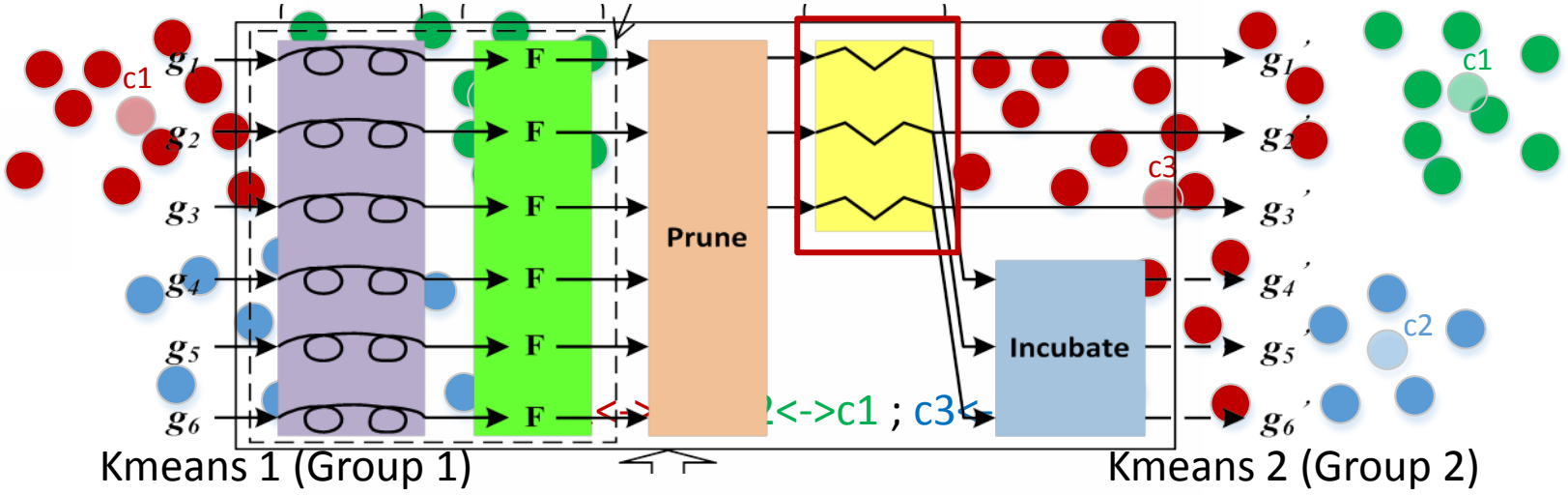
Kmeans, Evaluate & Prune

- Use Kmeans algorithm to do clustering
  - Get multiple updated centroid groups
- Use Total Within-Cluster Variation (TWCV) to evaluate different centroid groups' quality.
- Prune x% centroid groups with relatively low quality



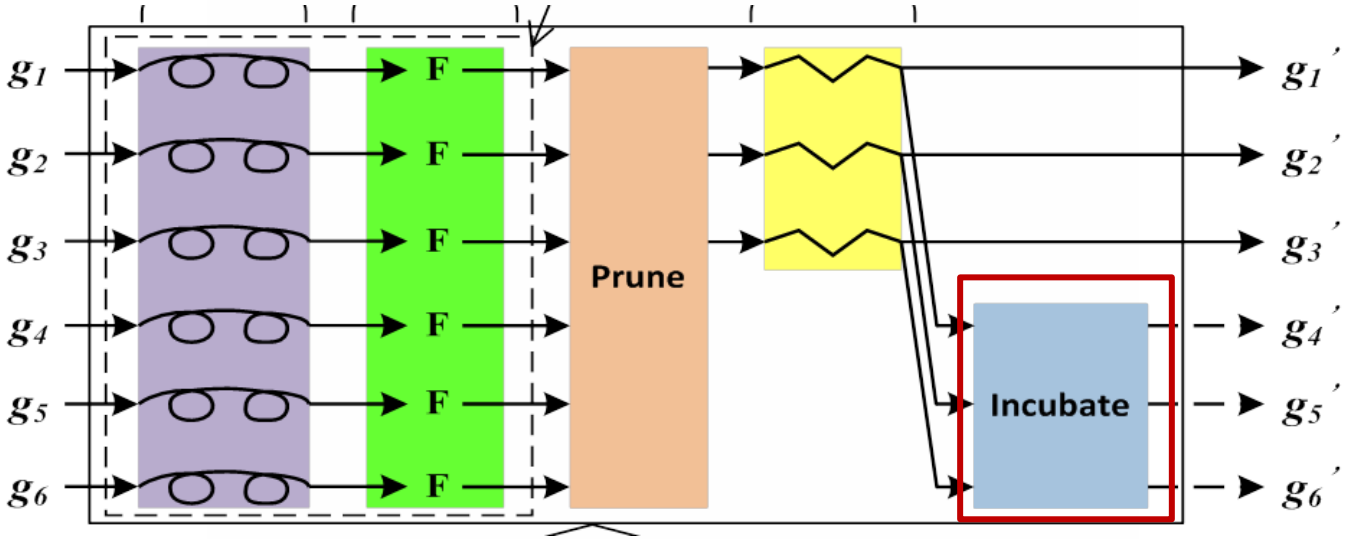
Permute

- A preprocessing step before the incubate step
- Aim to find the related centroids between different groups

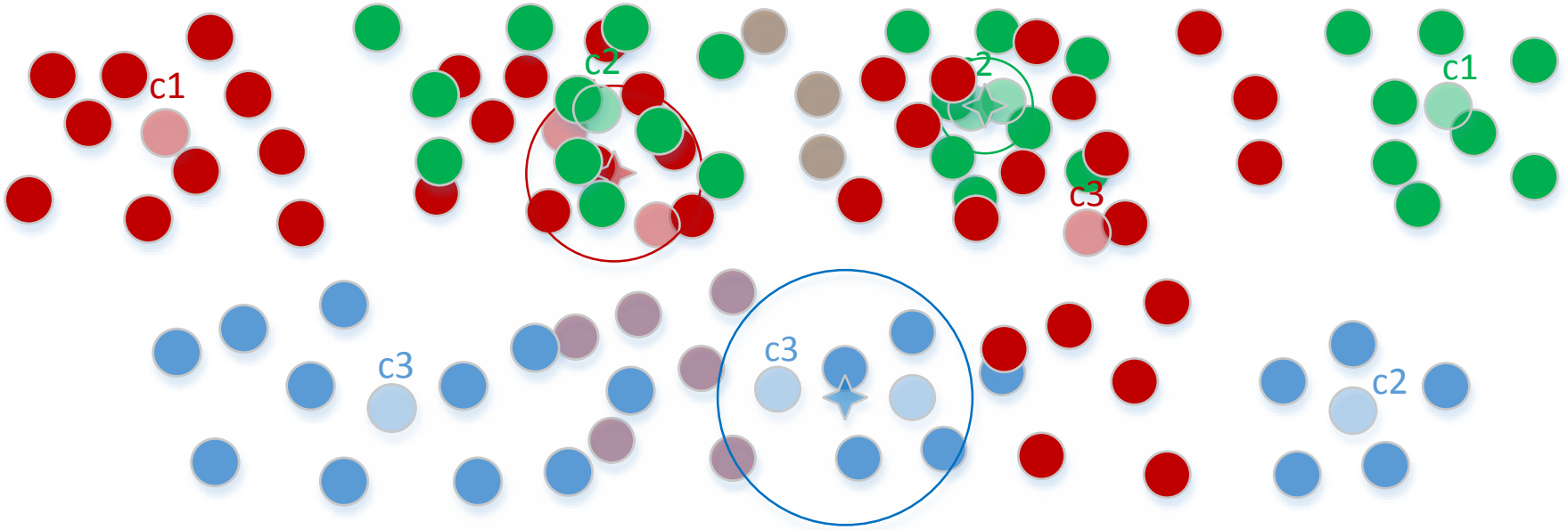


- $\mathrm{sim}(c_i, c_j) = \frac{1}{1+\mathrm{distance}(c_i, c_j)}, \mathrm{sim}(ci, cj) \in [0, 1]$

Incubate



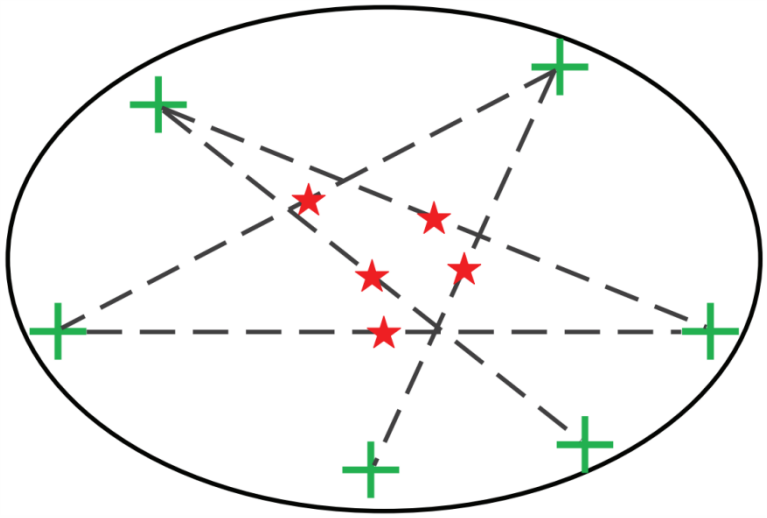
Incubate 1: *RSDS*



Random Search within a Definite Scope

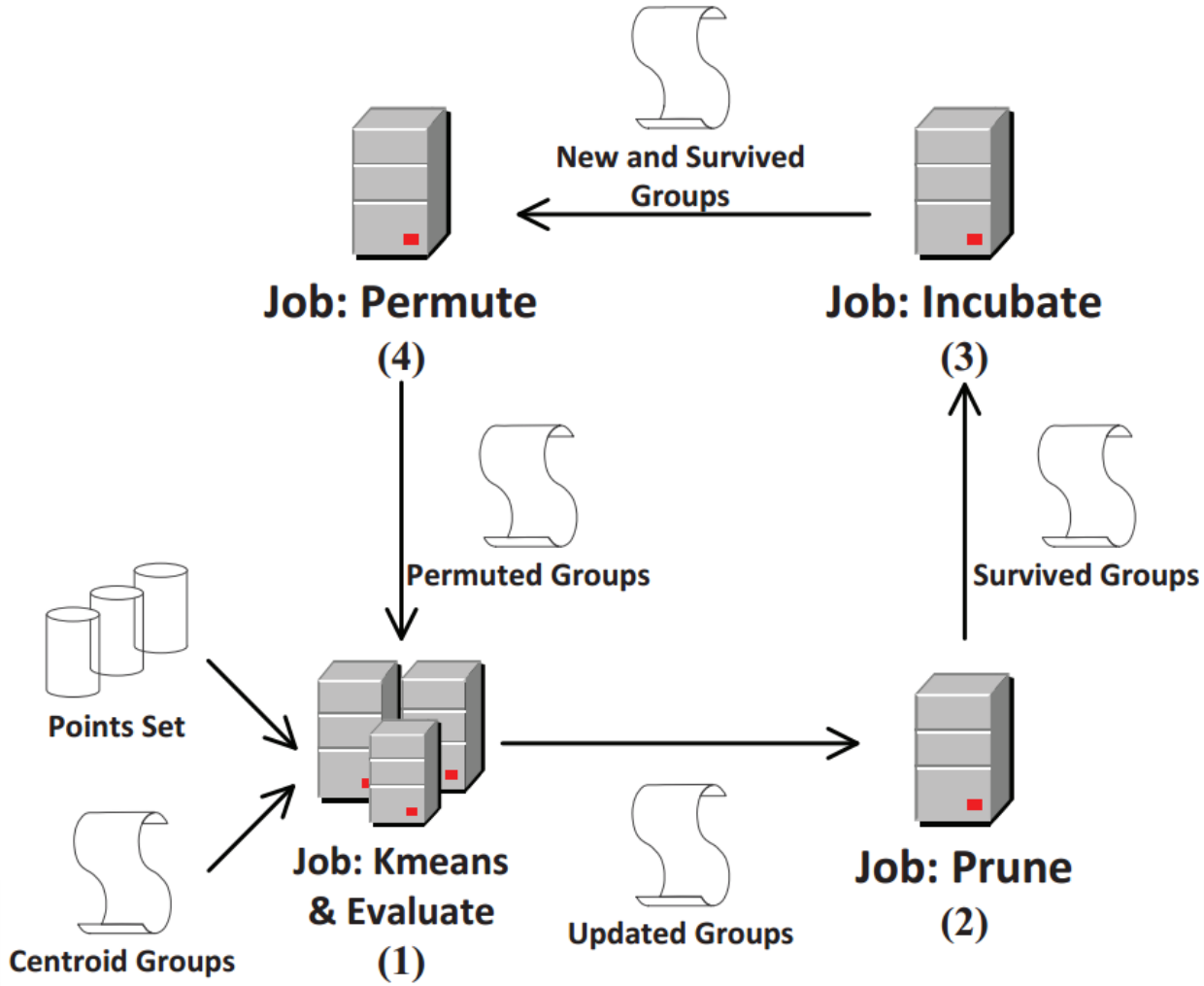
1. Star = compute the middle of two related centroids
2. Random search around the star, radius = 2*distance(centroid, start)

Incubate 2: *ADGP*



Average of Dissimilar Group Pairs

Implementation



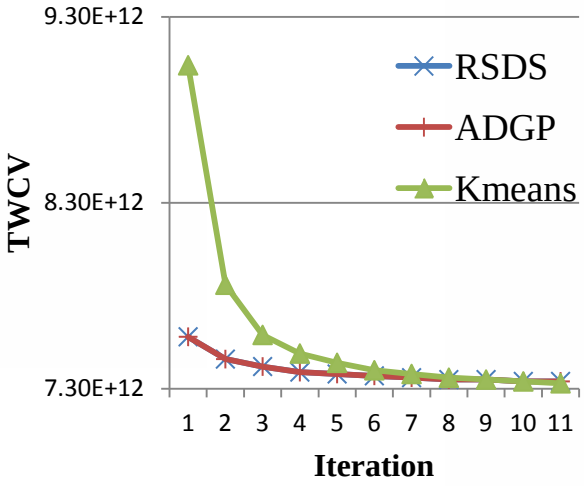
Experiment Setup

- Experiment Environment
  - Amazon EC2: 16 nodes, Ubuntu 12.04, Hadoop 1.4.1
  - Each node: 2 ECUs and 1 CPU; 3.7 GB memory; 410 GB storage; moderate network performance
- Experiment Dataset

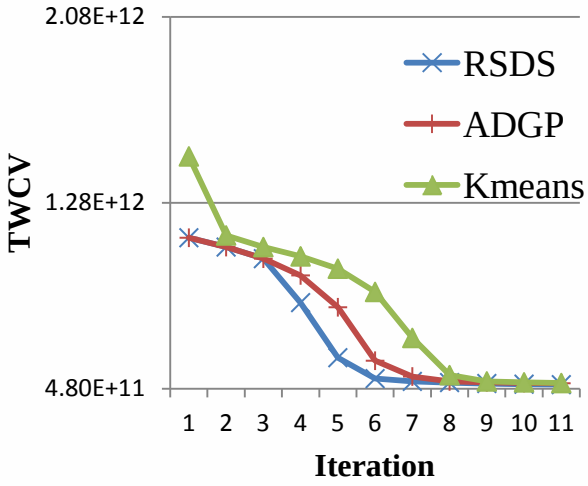
	Points	features
Bio_train	145751	74
Netflix	17770	1000
Lastfm	359330	40

Clustering Quality

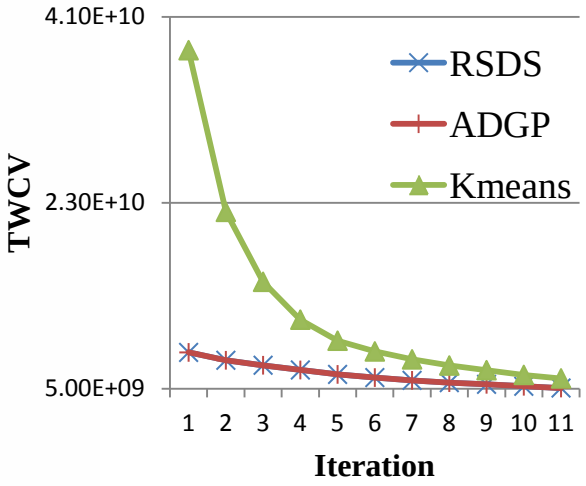
Different data set



Lastfm



Bio_train



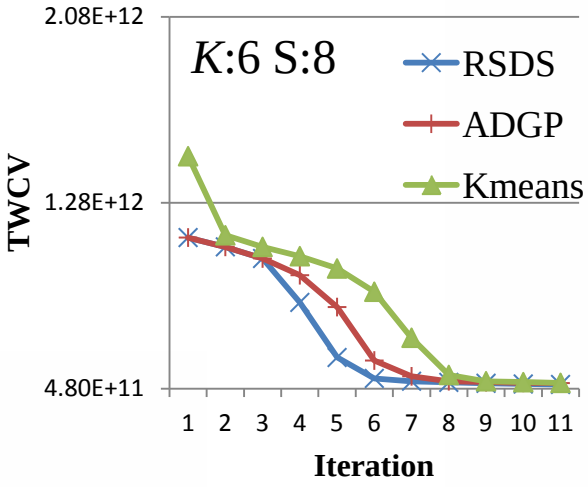
Netflix

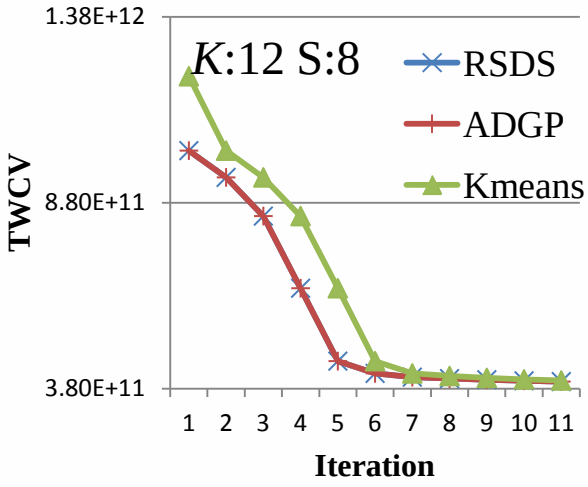
K value:6 centroid group amount:8

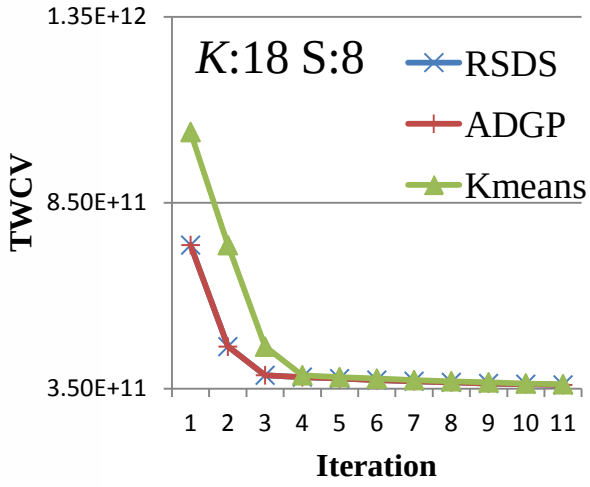
Clustering Quality

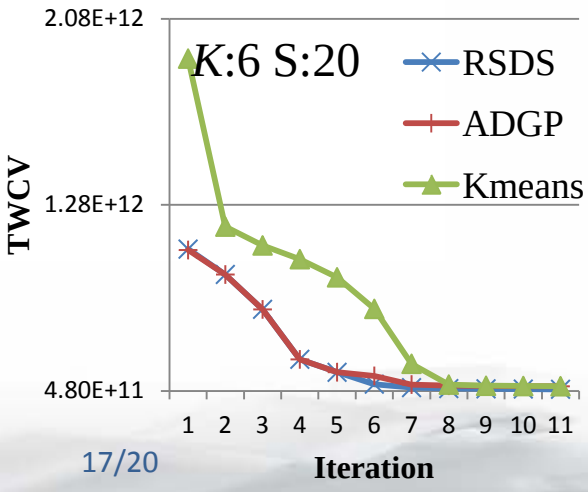
Same data set: Bio_train

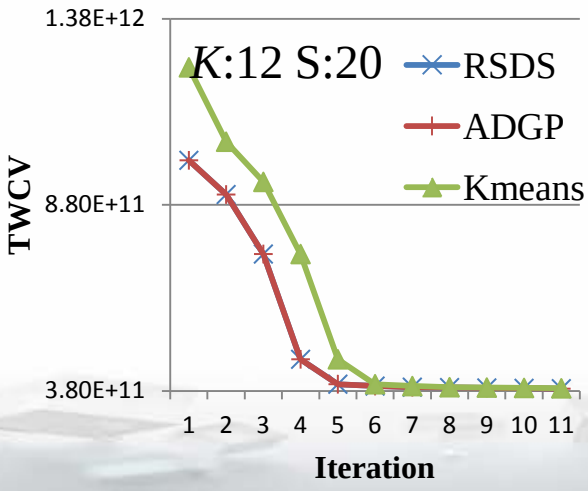
※S: centroid groups amount

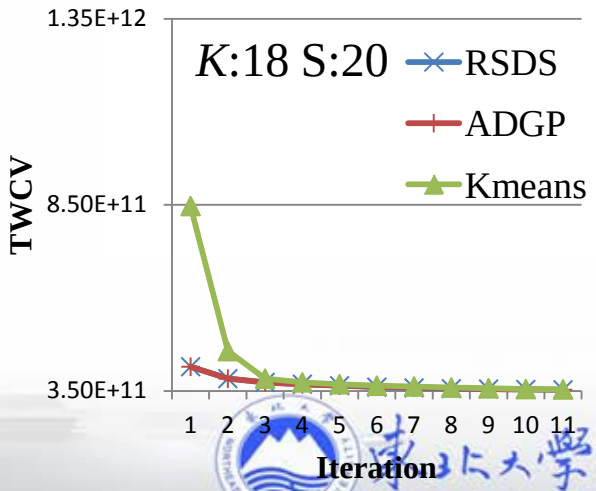




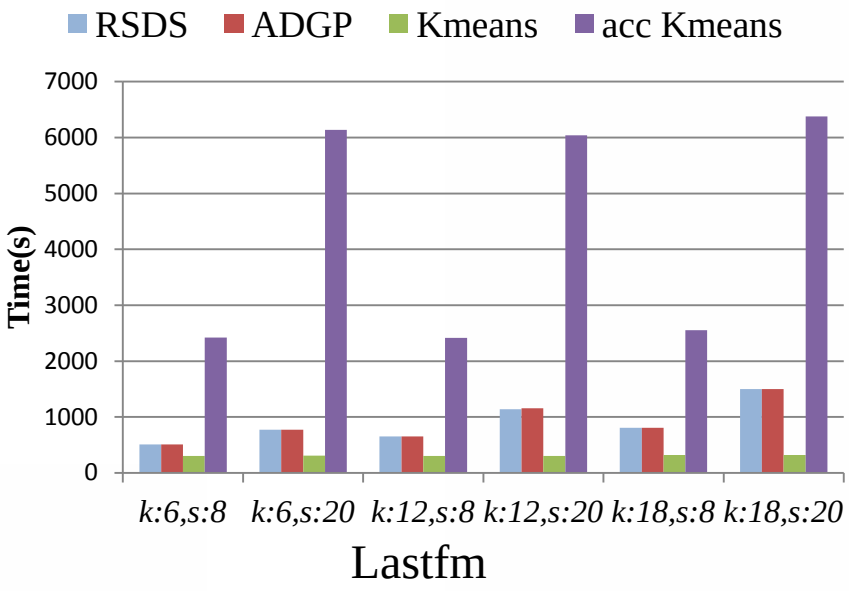




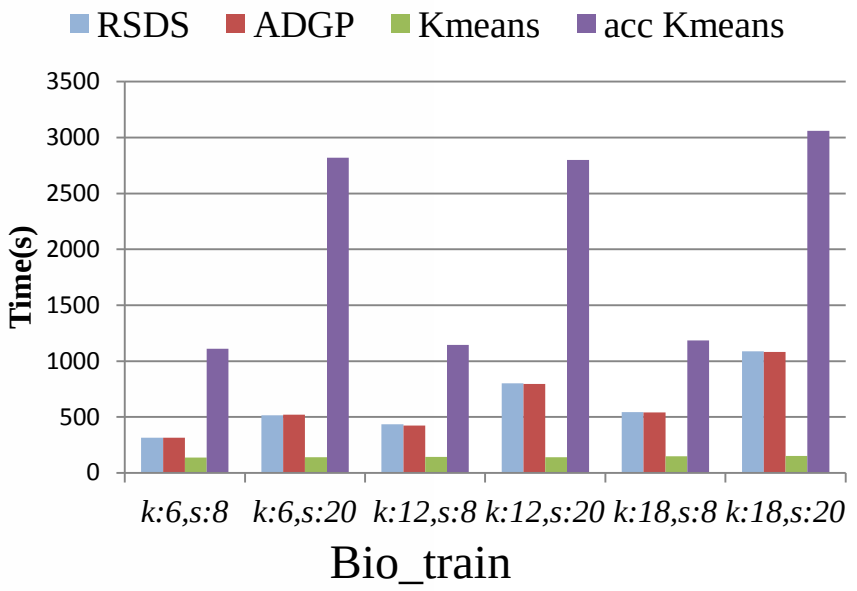




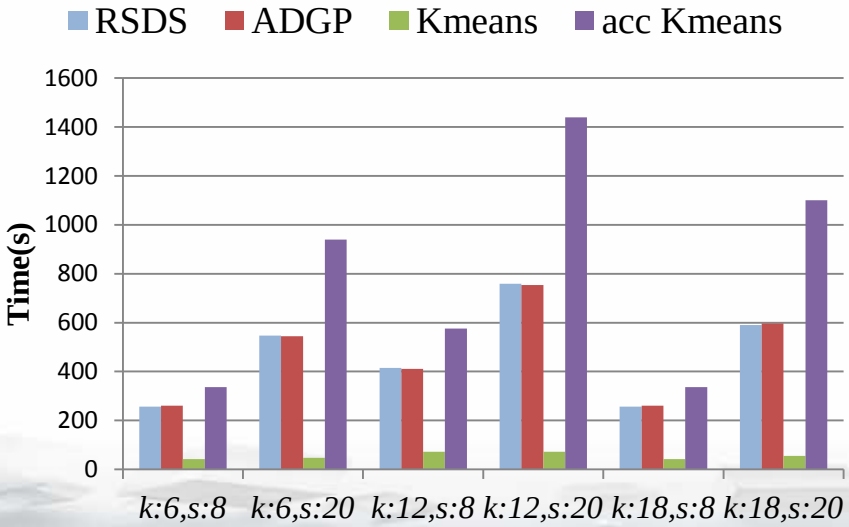
Elapsed Time



Lastfm



Bio_train



Netflix

Summary & Future Work

- The Mux-Kmeans algorithm.
  - Idea: execute multiple attempts in parallel, share states across different kmeans processes, terminate “hopeless” attempts in early stage, expand search scale and try more attempts
  - Implementation: deployed on MapReduce
  - Result: better clustering quality and shorter runtime when processing multiple centroid groups
- Future work
  - Different k in different centroid groups
  - Many possible Mux-XXX algorithms (Mux-EM, Mux-FCM, etc.)

Thank you!

QUESTIONS?