

# Parallel Clustering of Large Document Collections

Xiaohu Li, Deyun Gao, Zheyuan Yu

31 July 2003

**Document clustering** is the process of organizing documents into clusters so that

- Documents within a cluster have high similarity in comparison to one another.
- But are very dissimilar to documents in other clusters.

# An application of document clustering

**Vivísimo**  Search the Web

[Advanced Search](#) [Help!](#) [Tell Us What You Think!](#)

**Clustered Results**

- parallel computing (234)
  - High Performance Computing (37)
  - Programming (27)
    - Projects, University (4)
      - UCLA Parallel Computing Laboratory
      - Temple University Synergy Project
      - University of Alberta - Software Systems Research Group
      - Parallel Computing Works
    - MPI, Cluster (3)
      - LAM/MPI Parallel Computing
      - Open Directory - Computers: Parallel Computing
      - Parallel Programming with MPI
    - Programming Environment (5)
    - Supercomputing and Parallel Computing (2)
    - Caltech Concurrent Computation Program (2)
    - Parallel Scientific Computing (2)
    - Bulk Synchronous Parallel (2)
    - Research And Programming (2)

Category **parallel computing** > **Programming** > **Projects, University** contains 4 documents.

- UCLA Parallel Computing Laboratory** [New Window] [Full Window] [Preview]  
University research lab presents its **programming projects**, its contracts and its faculty profiles.  
URL: [pcl.cs.ucla.edu/](http://pcl.cs.ucla.edu/)  
Source: [Looksmart 9th](#), [MSN 12th](#), [Netscape 17th](#)
- Temple University Synergy Project** [New Window] [Full Window] [Preview]  
Synergy is a **programming** environment for a cluster of Unix systems. Evaluation packages are available for up to 3 processors.  
URL: [joda.cis.temple.edu/~shi/synergy.html](http://joda.cis.temple.edu/~shi/synergy.html)  
Source: [Looksmart 123th](#)
- University of Alberta - Software Systems Research Group** [New Window] [Full Window] [Preview]  
Department of **Computing** Science lab housed at the **University** of Alberta specializes in **parallel programming**. Find research **projects**, technical reports, and theses.  
URL: [www.cs.ualberta.ca/~systems/](http://www.cs.ualberta.ca/~systems/)  
Source: [Looksmart 56th](#)
- Parallel Computing Works** [New Window] [Full Window] [Preview]  
**Parallel Computing Works** This book describes work done at the Caltech Concurrent Computation Program , Pasadena, California. This **project** ended in 1990 but the work has been

## Previous Works

- Hierarchical Methods:
  - Agglomerative and Divisive.
  - Reasonably accurate but not scalable.
- Partitioning Methods:
  - Efficient, scalable, easy to implement.
  - Clustering quality degrades if an inappropriate number of clusters is provided.

## Vector Space Model

- Each document is represented by n-vector  $d_i$  of term weight.
- term weight: term frequency (tf), inverse document frequency (idf).  $w_{i,j} = 0$  if a term is absent
- Each direction of the vector space corresponds to a unique term in the document collection
- Vectors assembled into Term Frequency Matrix  $M = (d_1, d_2, \dots, d_m)$

## A Term by Document Matrix

	Doc 1	Doc 2	Doc 3	...	Doc n
business	5	5	2	...	1
capital	2	4	3	...	5
fund	0	0	0	...	1
⋮	⋮	⋮	⋮	⋮	⋮
invest	6	0	0	...	3

## Challenges in document clustering

- High dimensionality.  
K. Beyer et. al.[1] have shown that in high dimensional space, the distance to the nearest data point approaches the distance to the farthest data point.  
The similarity measure of the clustering algorithms do not work effectively, hence the meaningfulness of clustering may be doubtful
- High volume of data.
- Consistently high clustering quality.

## **Our goal**

To fight the challenges of document clustering, we want to obtain an scalable and effective parallel document clustering algorithm with reasonable speed up.



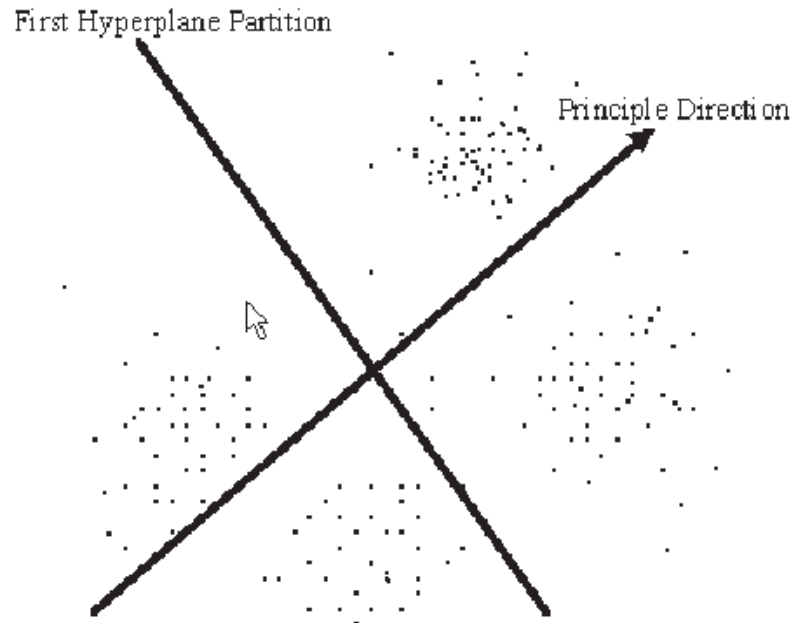
## Principal Direction Divisive partitioning

- based on the principal component analysis instead of traditional distance or similarity measure, reported to be scalable and effective.
- Related Methods - Principal Component Analysis
  - PCA: To discover or to reduce the dimensionality of the data set.
  - LSI
  - PDDP computes just first eigenvector.

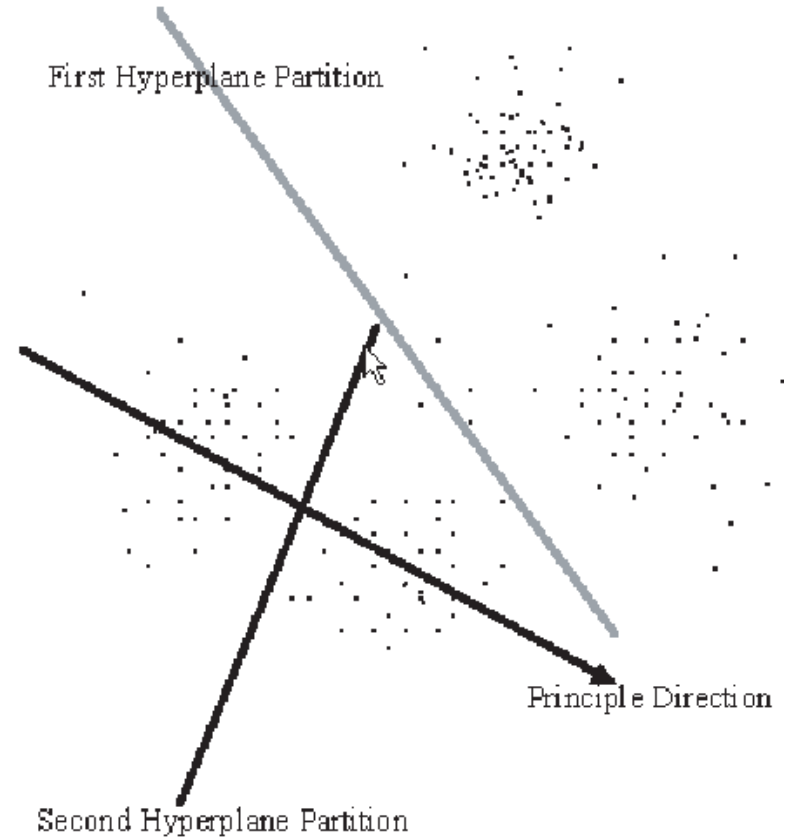
## Principal Direction Divisive partitioning (Cont)

- Get leading principle direction  $u$  of  $M - we^T$  with  $SVD$ , where  $w = \frac{1}{m} \sum_{i=1}^m d_i = \frac{1}{m} Me$ ,  $e = (1, 1, \dots, 1)^T$
- Split documents by value of projection  $u^T(d_j - w)$ ,  $j = 1, 2, \dots$
- Repeat the process on each cluster recursively

# Principal Direction Divisive partitioning - Splitting Steps



*Two Total Clusters*



*Three Total Clusters*

## Approach - Algorithm Issues: Fast Lanczos Solver

- Total cost dominated by cost of finding principal direction.
- Use efficient sparse matrix eigensolver "Lanczos".
- Matrix used only to form matrix-vector products.
- Use Bisection and Sturm sequence to find the largest eigenvalue.

## Our improvement for implementation of Lanczos

- Covariance matrix multiply vector:  $Cv$ 
  - Lanczos algorithm computes  $Cv$  for each iteration
  - If  $C = (M - we^T)(M - we^T)^T$  is calculated directly, the sparsity of the matrix is destroyed.
  - To keep the sparsity and avoid matrices multiplication for memory and computational efficiency: we implement

$$Cv = (M - we^T)(M - we^T)^T v$$

as

$$M(M^T)v - Mew^T v - we^T M^T v + we^T ew^T v$$

## Our improvement for implementation of Lanczos

- Bisection Sturm Sequence Algorithm
  - In Lanczos algorithm, the most time consuming step is to get the largest eigenvalue of tridiagonal  $T$ .
  - In PDDP algorithm the general approach to compute the largest eigenvalue by getting all the eigenvalues and picking the largest one.
  - Bisection sturm sequence algorithm can directly compute the largest eigenvalue of tridiagonal matrix  $T$

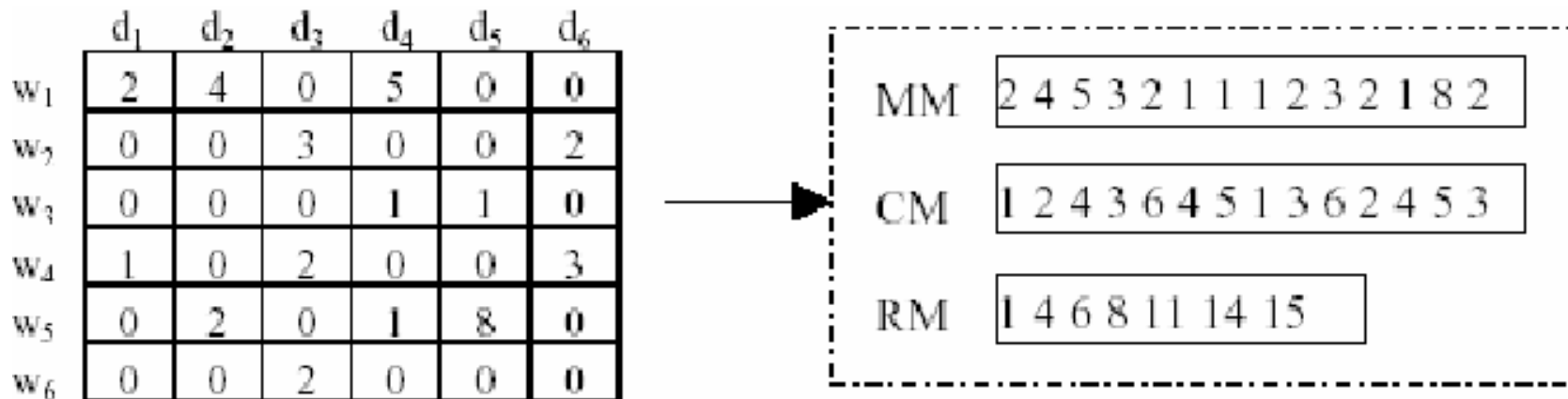
## Principal Direction Divisive partitioning (Cont)

Data Sets:

- D1: 2340 docs, 21,839 words
- D3, D9, D10: reduced dictionaries
  - D3: 8104 words
  - D9: 7358 words
  - D10: 1458 words

## Data Storage and Distribution

- Represent set of document by term-by-document matrix
- The matrix is vary sparse
- Choose Compressed Sparse Row (CSR) storage format





## Data Storage and Distribution - Continue

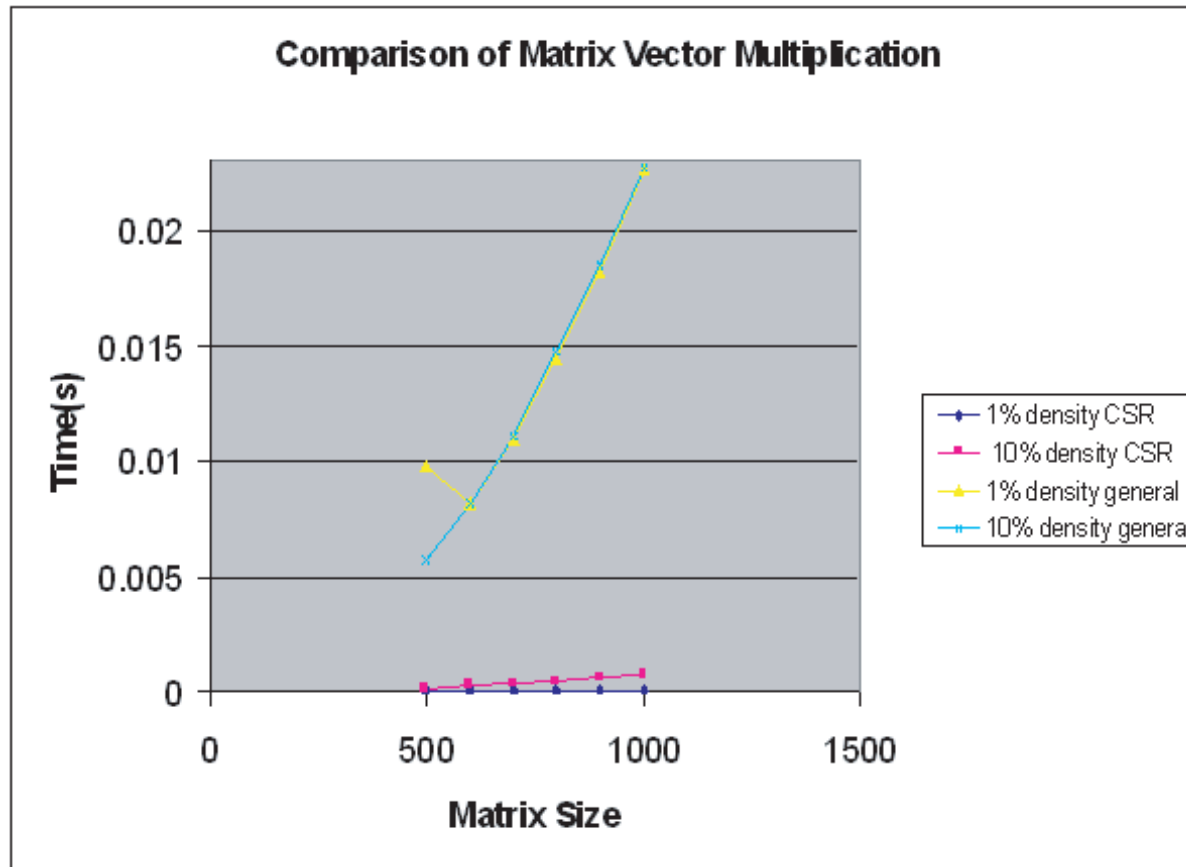
Comparison of matrix storage Save storage cost: from  $M \times N$  to  $(2 \times Nz + N + 1)$

- Save storage cost: from  $M \times N$  to  $(2 \times Nz + N + 1)$

Document Set	# of Doc	# of Term	None Zero	Density	General Storage (bytes)	CSR Storage (bytes)
f1	98	5624	27186	4.9%	4,409,216	348,732
j1	185	10536	586130	3%	15,593,280	7,075,708
k1	234	21839	34980	0.6%	40,882,608	507,120

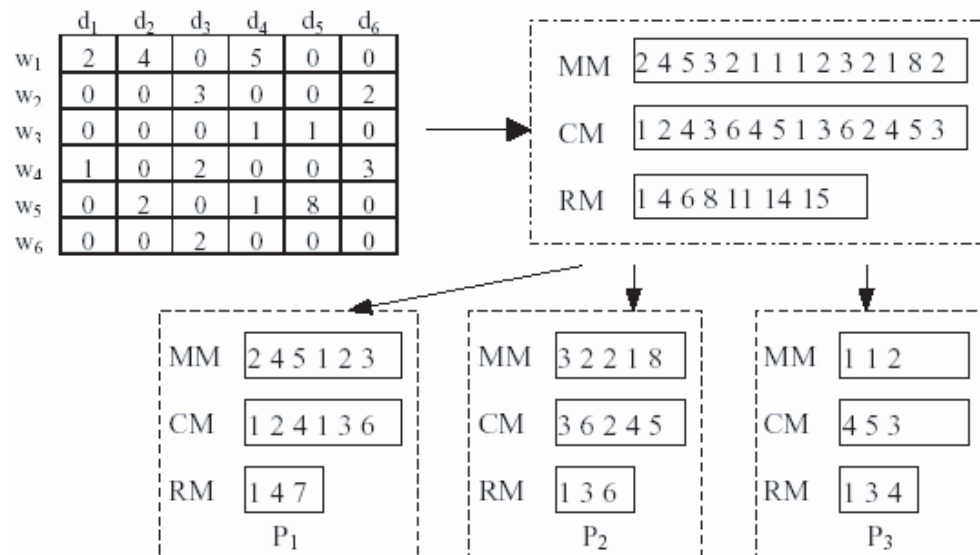
double=8bytes, int =4bytes

# Reduce time complexity for Matrix Vector multiplication



## Data Distribution for Parallel

- Matrix vector multiplication is one of the most time consuming operations.
- Data allocation is performed by rows.

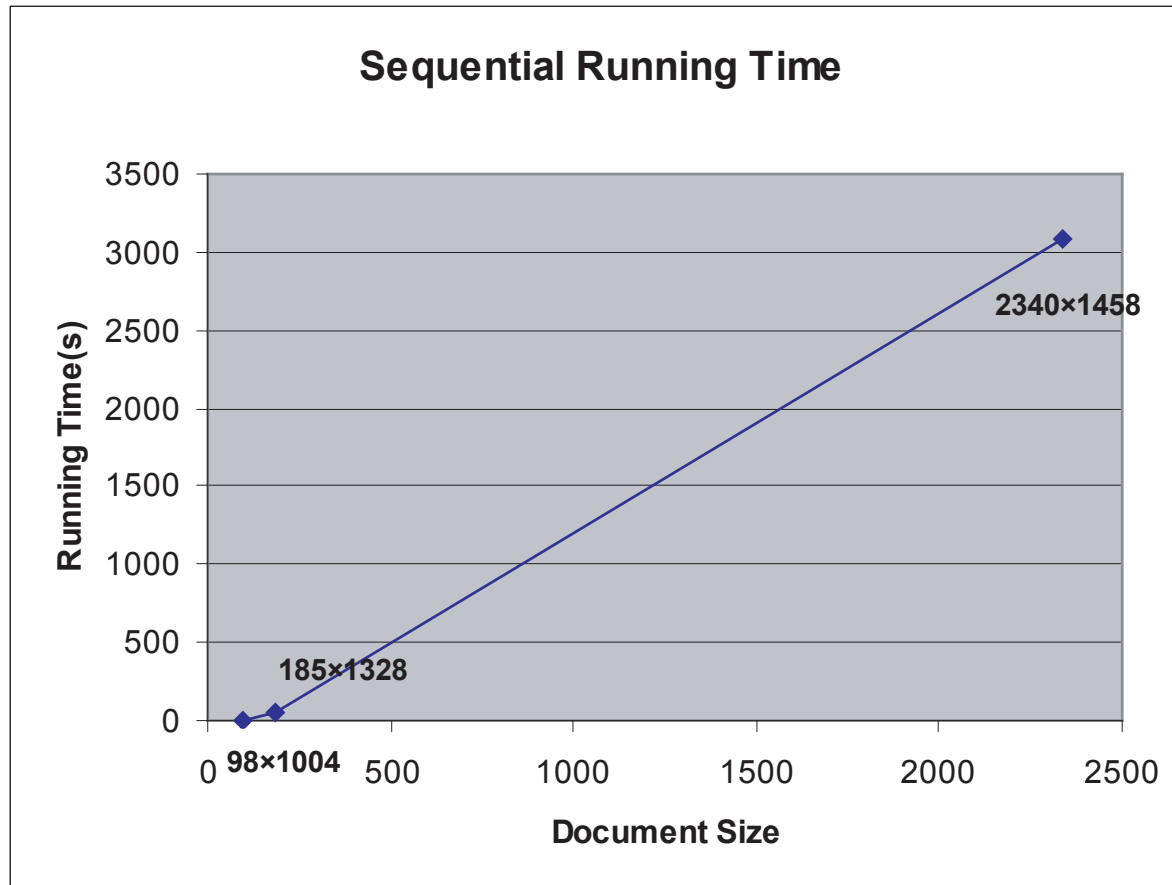


## Data Distribution for Parallel - Continue

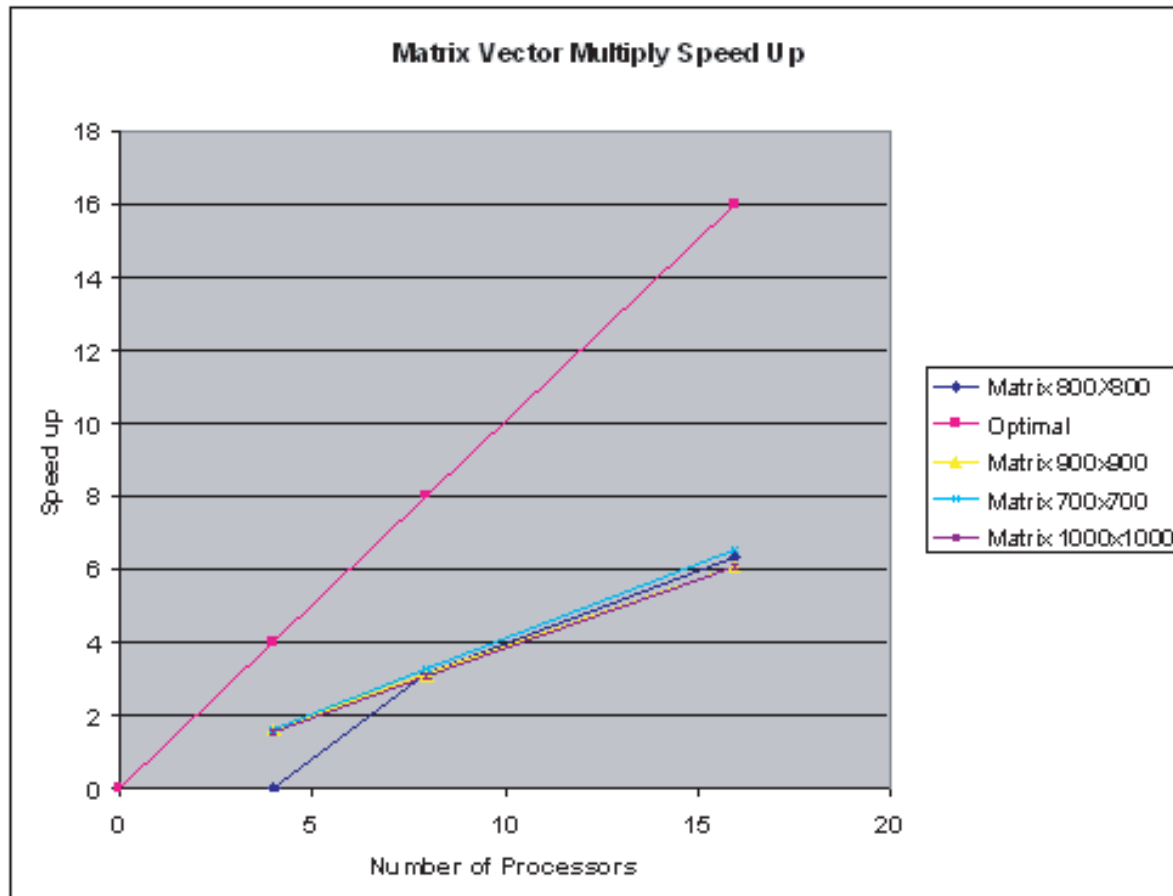
- During the processing, document set is divided into clusters
- Corresponding matrix is also divided vertically into sub-matrices.
- Only need local column re-allocation

$$A = \begin{bmatrix} A_{loc,0} & A_{ext,0}^1 & \dots & \dots & A_{ext,0}^{p-1} \\ A_{ext,1}^0 & A_{loc,1} & \dots & \dots & A_{ext,1}^{p-1} \\ \vdots & & & \vdots & \vdots \\ \vdots & & & \vdots & \vdots \\ A_{ext,p-1}^0 & A_{ext,p-1}^1 & \dots & \dots & A_{loc,p-1} \end{bmatrix} \quad v = \begin{bmatrix} v_{loc}^0 \\ v_{loc}^1 \\ \vdots \\ \vdots \\ v_{loc}^{p-1} \end{bmatrix} \begin{array}{l} \rightarrow \text{Node } 0 \\ \rightarrow \text{Node } 1 \\ \rightarrow \vdots \\ \rightarrow \vdots \\ \rightarrow \text{Node } (p-1) \end{array}$$

# Evaluation



## Evaluation - Continue



## Evaluation - Continue

- Evaluate speedup of the whole application
- Evaluate with larger document set
- Cluster quality evaluation: Entropy Purity
- Compare with other document clustering algorithm, such as K-means.

## REFERENCES

1. K. Beyer et. al., When is nearest neighbor meaningful?, In proceeding of the 7th ICDT, Jerusalem, Israel, 1999
2. D.L. Boley, Principal Direction Divisive Partitioning, Technical Report TR-97-056, University of Minnesota, Minneapolis, 1997
3. ShuTing Xu and Jun Zhang, A Hybrid Parallel Web Document Clustering Algorithm and Its Performance Study, Technical Report No. 366-03, Department of Computer Science, University of Kentucky, 2003”