Parallel Clustering of Large Document Collections

Xiaohu Li, Deyun Gao, Zheyuan Yu

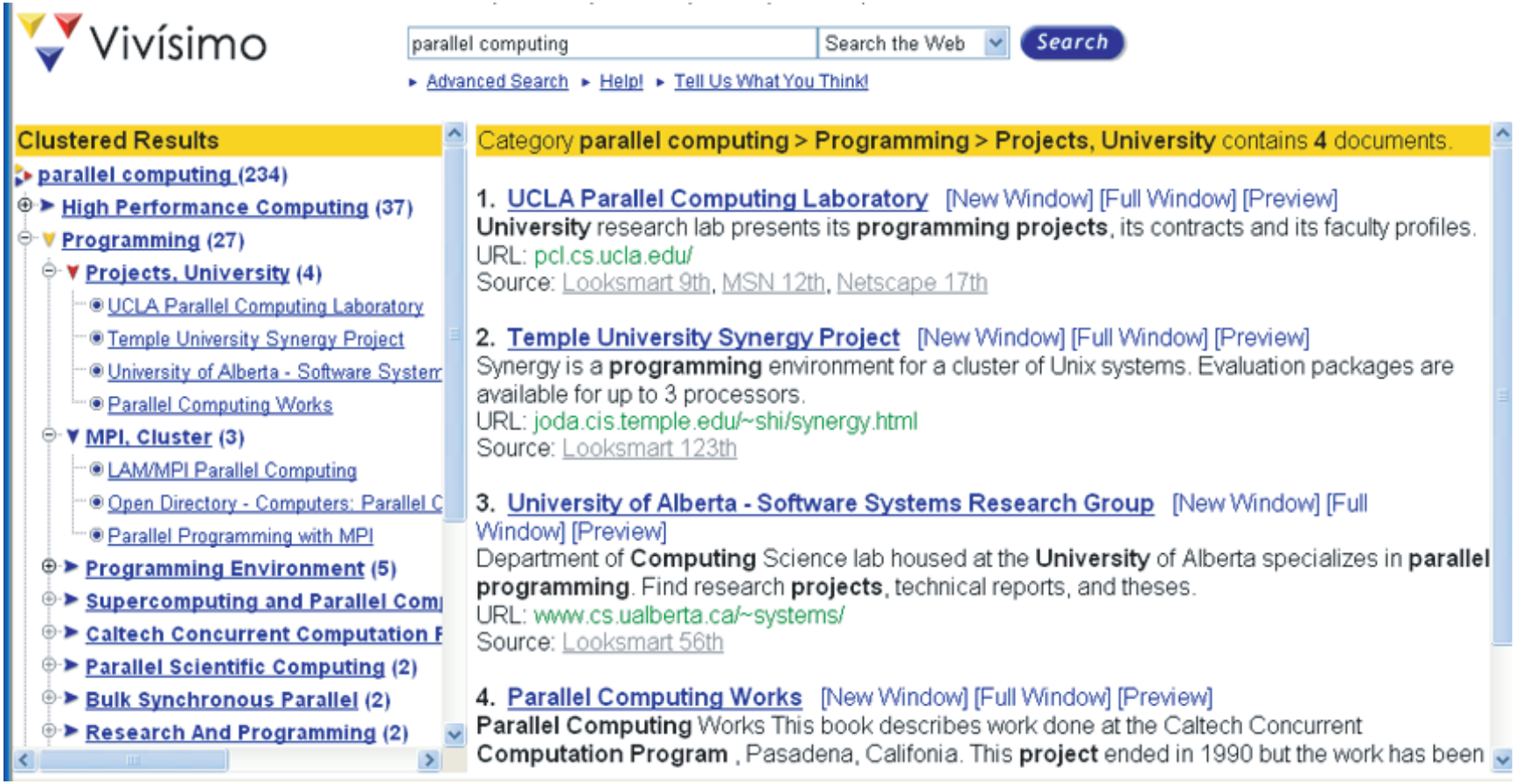
31 July 2003

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



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 $M = (d_1, d_2, ..., d_m)$

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

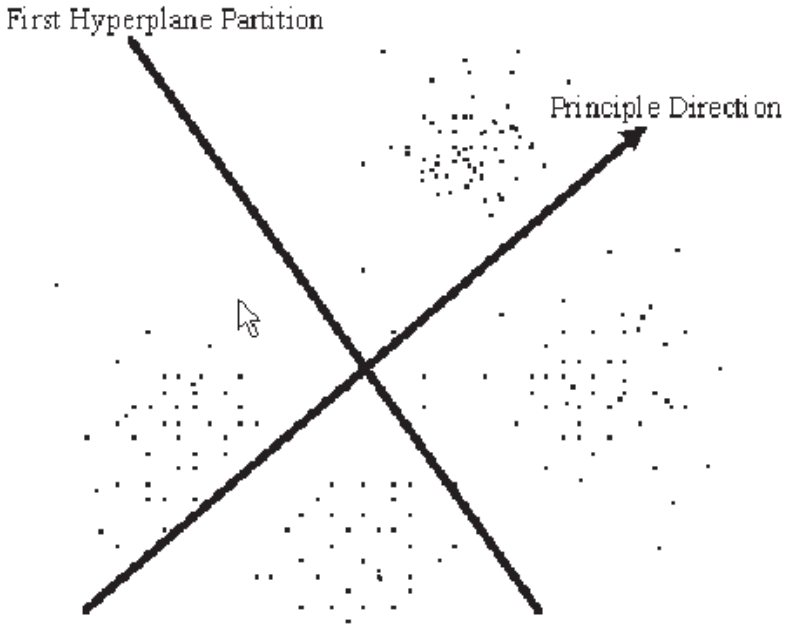
- based on the principal component analysis instead of traditional distance or similarity measure, reported to be scalable and effective.

- Related Methods - Principal Component Analysis
  - PCA: To discover or to reduce the dimensionality of the data set.
  - LSI
  - PDDP computes just first eigenvector.

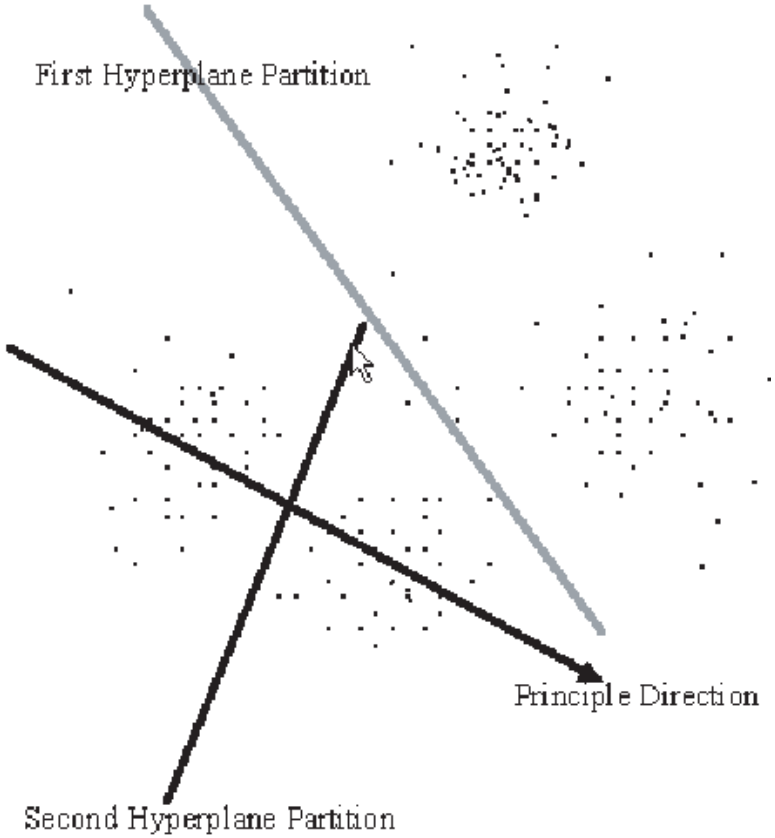
Principal Direction Divisive partitioning (Cont)

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

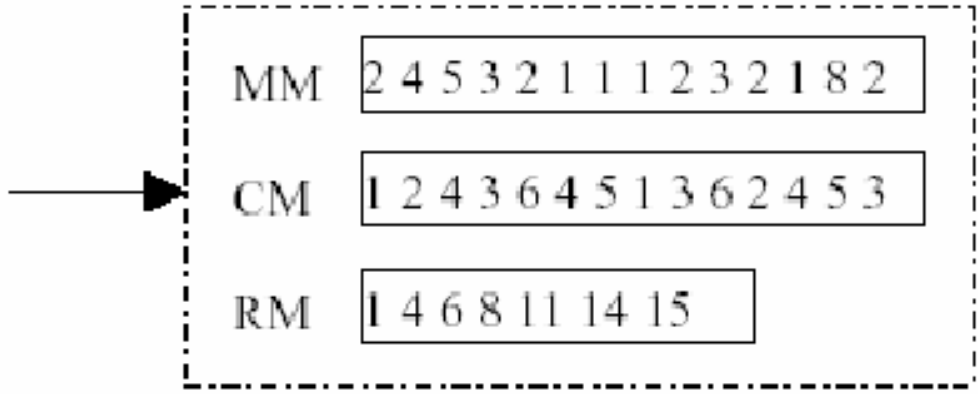
Data Sets:

- 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

	d_1	d_2	d_3	d_4	d_5	d_6
w_1	2	4	0	5	0	0
w_2	0	0	3	0	0	2
w_3	0	0	0	1	1	0
w_4	1	0	2	0	0	3
w_5	0	2	0	1	8	0
w_6	0	0	2	0	0	0



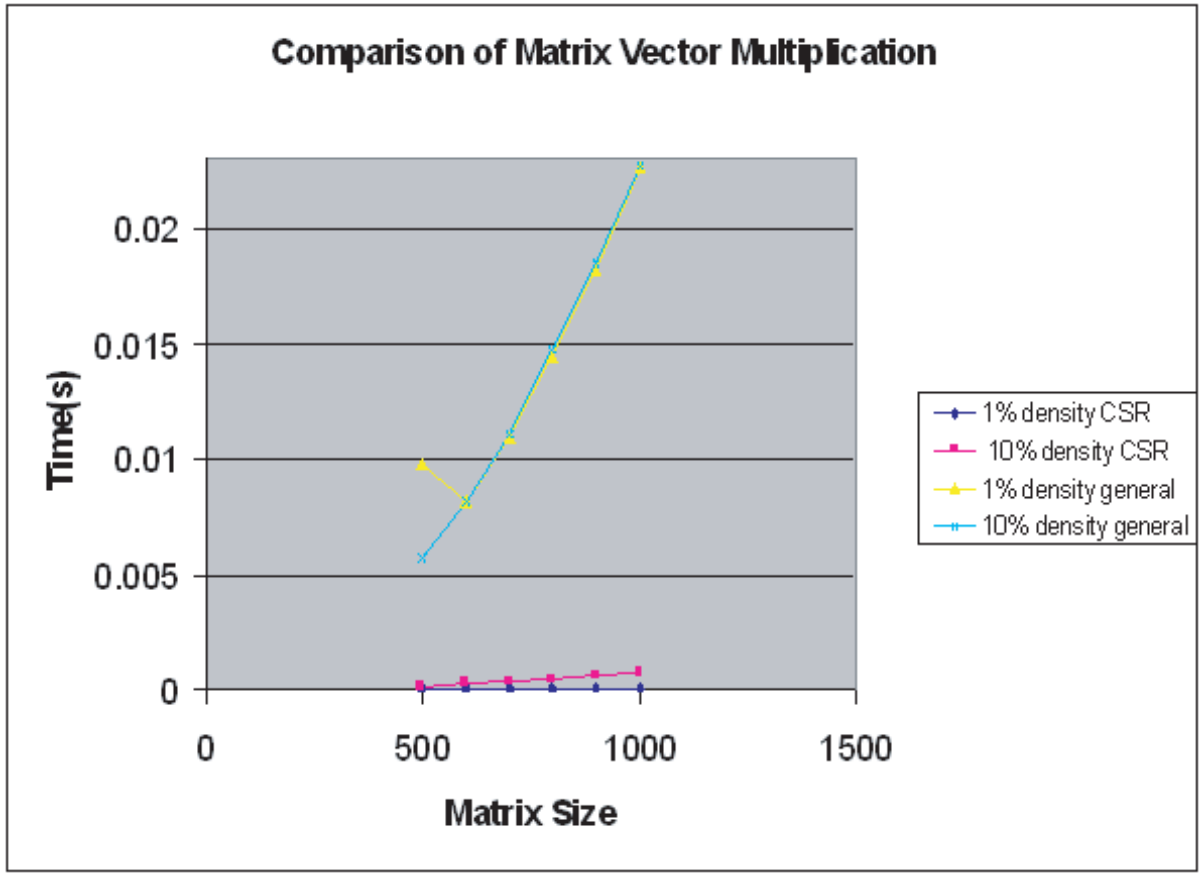
Data Storage and Distribution - Continue Comparison of matrix storage Save storage cost: from MxN to (2xNz+N+1)

- Save storage cost: from MxN to (2xNz+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

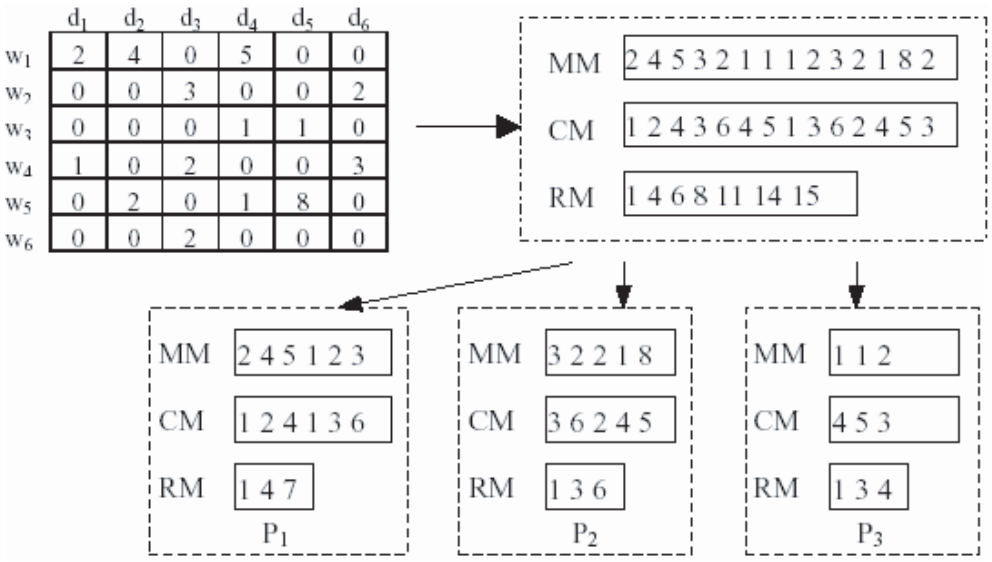
Reduce time complexity for Matrix Vector multiplication



Data Distribution for Parallel

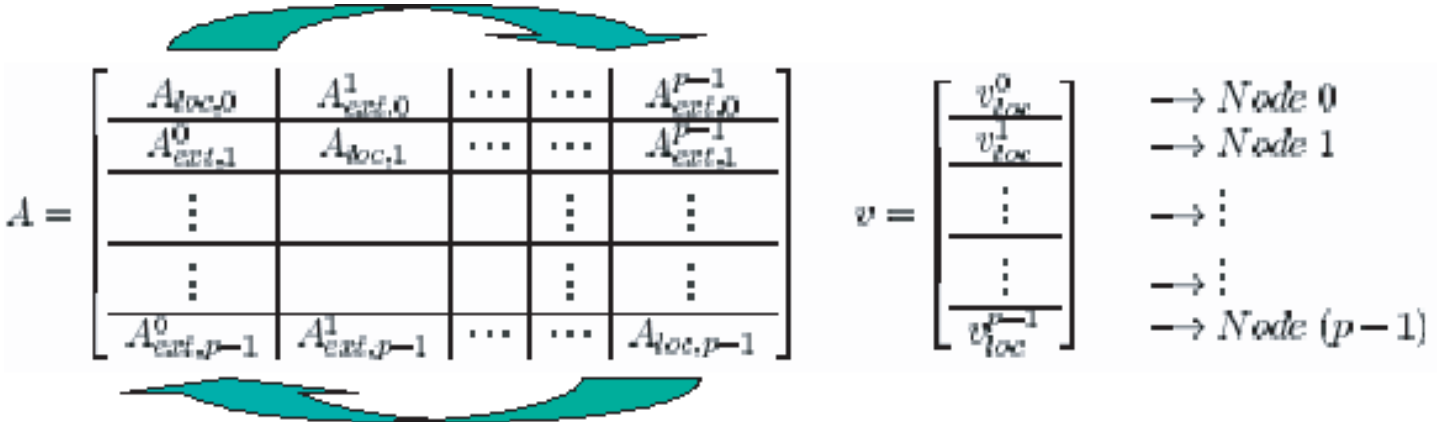
- Matrix vector multiplication is one of the most time consuming operations.

- Data allocation is performed by rows.

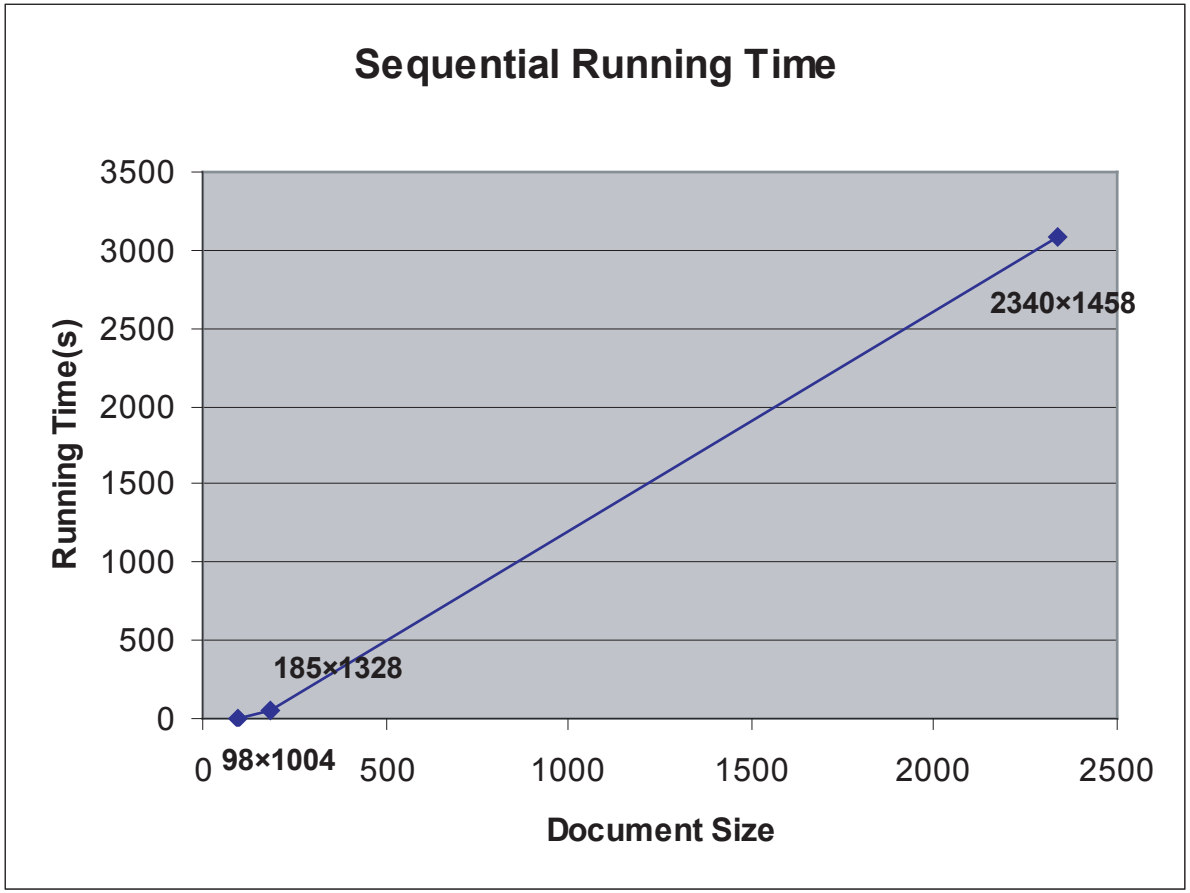


Data Distribution for Parallel - Continue

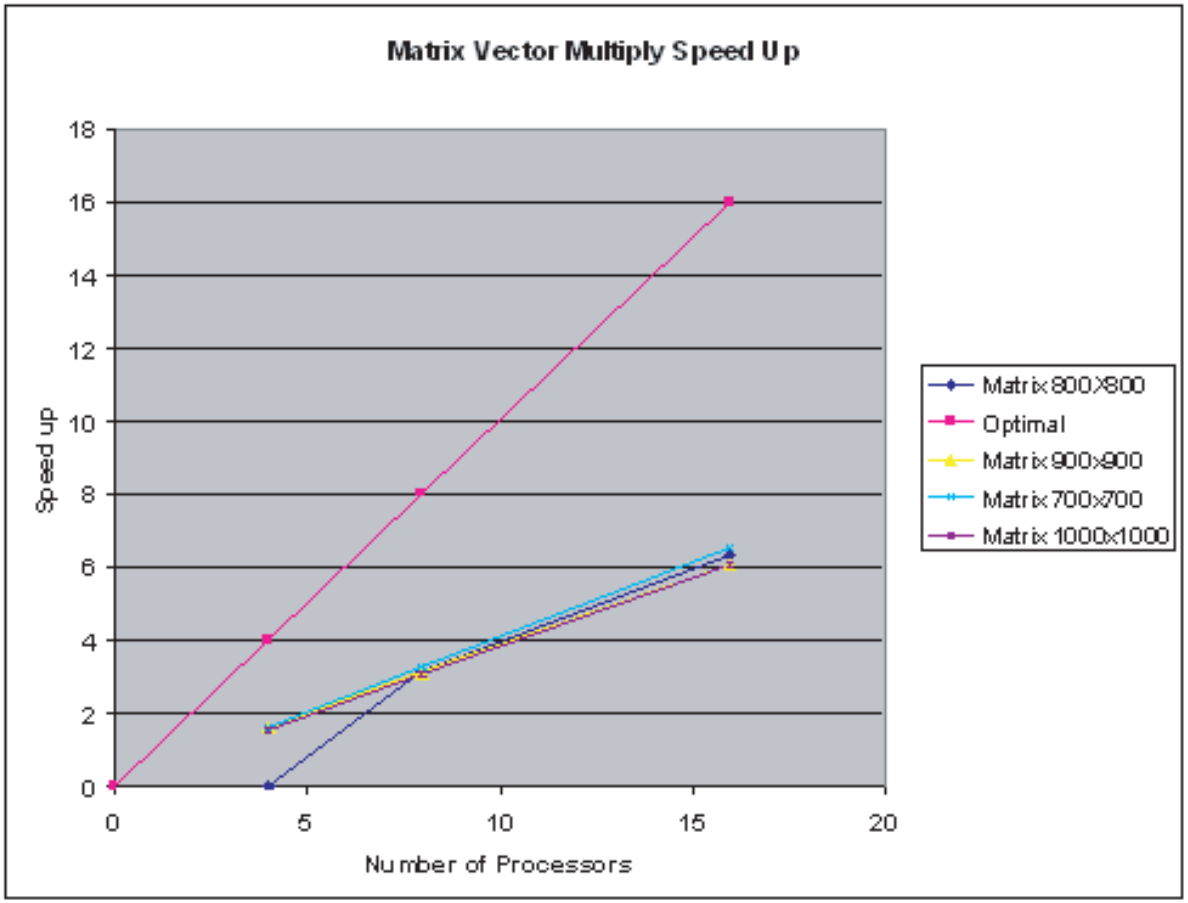
- During the processing, document set is divided into clusters
- Corresponding matrix is also divided vertically into sub-matrices.
- Only need local column re-allocation



Evaluation


Sequential Running Time
Running Time(s)
3500
3000
2500
2000
1500
1000
500
0
2340×1458
185×1328
98×1004
0
500
1000
1500
2000
2500
Document Size

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"