Survey topic: Parallel Clustering Algorithms
Detailed Annotated Bibliography and Classification of the Results

Wooyoung Kim
CSC 8530 Parallel Algorithms
Spring 2009

Classification of the papers

The major challenges of sequential clustering algorithms are scaling up to very large database sizes, which leads to high computational and spatial costs. Therefore, development of parallel clustering algorithms are natural process. But, we also notice that clustering task itself is used for parallelization of other algorithms as it groups the data set into smaller subsets.

In other words, clustering algorithm is essential preprocessing step in some applications for parallelization. Therefore, the papers in this survey can be classified into 5 broad categories as shown in the table [1].

<table>
<thead>
<tr>
<th></th>
<th>General Information of Clustering Analysis</th>
<th>70, 33, 36, 9, 32, 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Various Approaches to Data Clustering and Applications</td>
<td>61, 48, 75, 48, 55, 70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32, 33, 29, 28, 23, 51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>66, 69, 3, 39, 13, 36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34, 1, 45, 10, 19, 41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>44, 62, 71, 74, 30, 34, 63</td>
</tr>
<tr>
<td>3</td>
<td>Parallel Clustering Algorithms</td>
<td>12, 2, 64, 68, 27, 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>47, 11, 12, 50, 5, 57, 59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8, 49, 24, 52, 53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>72, 76, 34, 20, 35, 25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>65, 38, 11, 15, 22</td>
</tr>
<tr>
<td>4</td>
<td>Applications of Parallel Clustering Algorithms</td>
<td>17, 40, 74, 40, 16, 52, 37, 42</td>
</tr>
<tr>
<td>5</td>
<td>Applying Clustering Algorithms for Parallelization</td>
<td>26, 14, 60</td>
</tr>
</tbody>
</table>

Table 1: Score of the survey

First, the general clustering analysis are described along with motivations and the history. Clustering algorithms have been developed in broad areas of studies with different approaches. For the
second class of papers, therefore, introduce various data clustering algorithms and show the applications of the algorithms in various fields. The papers belong to the third and fourth category are the main focus of this survey. The 3rd category papers are presenting various parallel version of clustering algorithms. In the fourth category, we discuss the papers that use parallel clustering algorithms in real applications. The papers that use clustering algorithms to parallelization of some algorithms or tasks belong to the last category.

1 1st Class: General Information of Clustering Analysis

The papers belong to this category will give general information about clustering analysis and the applications. Mostly we will refer books to give a big picture of the clustering algorithms and the trends.

- **Algorithms for clustering data** [33], by Jain et al.(1988)
  This book gives general information about clustering algorithms starting from the description of data representation. The clustering methods and algorithms are classified into hierarchical, partitional clusterings. The validation of hierarchical and partitional clusters are discusses as well. Various applications of clustering algorithms are presented such as image segmentation and registration. The advantage of tree structure lies on the pruning technique, and this approach is applied repeatedly until the size of observed data is one for each node.

- **Data clustering: a review** [32], by Jain et al.(1999)
  This paper belongs two categories in this survey. It starts with the motivation of clustering analysis for data analysis, and describes components of a clustering task. The representation problems in data analysis and various measurements are discussed. It categorizes different clustering algorithms which belongs to the next class. In this paper, cluster analysis is defined as the "organization of a collection of patterns into clusters based on similarity". With the definition, clustering is distinguished with other analysis as follows;
  
  – **Clustering vs. Discriminant Analysis (supervised classification)**
    Discriminant analysis label a new unlabeled data based on a collection of labeled patterns, whereas clustering analysis group a given group of unlabeled patterns into ‘similar’ subgroups.
  
  – **Clustering analysis and prior assumptions**
    The author explains that clustering methodology is good for exploring interrelationships among data to interpret the structure. Therefore, as little as possible prior information is preferable.
  
  – **Components of a Clustering Task** The components of a clustering task is proposed as follows in this paper. The components in fact leads the clustering procedures introduce in [70].
    1. Pattern representation
    2. Measurements appropriate to the data domain
3. Clustering or Grouping
4. Data Abstraction
5. Assessment of output

The paper presents a taxonomy of clustering approaches as a tree structure. Hierarchical clustering and partitional clustering are distinguished at the top level. But there are several cross-cutting issues for the overall structure of taxonomy.

- Agglomerative vs. divisive
  This aspect refers to algorithmic structure and operation, since an agglomerative approach is a bottom-up approach while divisive is top-down approach.

- Monothetic vs. polythetic
  In polythetic algorithm, all features enter into the system while features sequentially enter into the computation in monothetic algorithm.

- Hard vs. fuzzy
  A hard clustering assigns one point to only one cluster, while fuzzy clustering algorithm allocates one data point into several clusters.

- Deterministic vs. stochastic
  For the partitional clustering approaches, this aspect refers to the optimization scheme.

- Incremental vs. non-incremental
  The algorithm where the size of data can be increased is considered as incremental.

- Survey of clustering algorithms [70], by Rui Xu and Wunsch, D., II (2005)

  This paper also belongs two categories in this survey. It gives general information of clustering analysis by comparing it with classification, and prediction tasks. The categorization of clustering algorithms part belongs to the second class.

  - Supervised classification vs. Unsupervised classification
    In supervised classification, the goal is to assign an input into one of class based on some mathematical function which is learned with labeled training data.
    In unsupervised classification, called clustering or exploratory data analysis, no labeled data are available. The goal is to separate or partition a set of unlabeled data set into several subsets based on the conceptual or hidden properties of the input data.

  - Unsupervised predictive learning vs. Nonpredictive clustering
    Clustering is different from unsupervised predictive learning such as vector quantization, probability density function estimation and entropy maximization. Unsupervised predictive learning provides an accurate characterization of unobserved samples generated from the same probability distribution, while nonpredictive clustering divide the data into several subsets based on their subjective measurements.
– Partitional clustering attempts vs. Hierarchical clustering attempts
Based on the motivation, clustering tasks can be divided into two category: partitional and hierarchical attempts. Partitional attempts can be divided with hard partition and soft partition as well. The soft partition is known as fuzzy clustering, which will be reviewed later.

– The procedure of cluster analysis
In the introduction of this paper, there are four processes of clustering task.
1. Feature selection or extraction
2. Clustering algorithm design or selection
3. Cluster validation
4. Results interpretation

– Challenges in cluster tasks
* Repetitions or trials
* No universal guide of feature selection or feature extraction
* No universal validation criteria for the quality of the results
* Various applications in various fields, no standardized solution.

– Various fields using clustering algorithms
  Engineering Machine learning, Artificial Intelligence, Pattern recognition, Mechanical engineering, Electrical engineering, etc.
  Computer Science Web mining, Spatial database analysis, Textual document collection, Image segmentation, etc.
  Life and Medical Science Genetics, Biology, Microbiology, Paleontology, Psychiatry, Clinic, Pathology, etc.
  Earth Science Geography, Geology, Remote sensing, etc.
  Social Science Sociology, Psychology, Archeology, Education, etc.
  Economics Marketing, Business.

Pattern Recognition and Machine Learning [9], by Bishop (2006)
In the introduction chapter (chapter 1) in this book, the clustering is one of unsupervised learning problems. Basically, learning systems are either supervised or unsupervised, depending on whether they assign new inputs to one of a finite number of discrete supervised classes or unsupervised categories, respectively. In supervised learning, the target vectors are known while unsupervised learning is given a set of input vectors without target values. If the goal of unsupervised learning is to divide the whole data into several subsets of similar data, then it is called clustering. If the goal is to decide the distribution of input data, then it is called density estimation. The goal of visualization is to project the high-dimensional data into lower dimension space.
2 2nd Class: Various Approaches to Data Clustering and Applications

With a couple of survey papers regarding to the clustering algorithms, we will observe several categories of clustering algorithms and the applications. Based on the categorization, we will describe several key concepts with supporting papers.

- **Bayesian classification (AutoClass): theory and results [13], by Cheeseman et al.(1996)**
  
  This paper is about AutoClass system which is based on Bayesian method for determining optimal classes in large data sets. In the category of [70], it belongs to the mixture densities-based clustering. In the probabilistic view, data points are assumed to be generated according to probability distributions. Combining it with clustering point of view, each cluster is represented with different probability distributions, (different type or different parameters). The algorithms belonging to this category mostly use expectation-maximization(EM) approach. It first initialize the parameters of each cluster. It computes the complete data log-likelihood in e-step and select new parameters maximizing the likelihood function. AutoClass considers a number of families of probability distributions including Gaussian, Poisson and Bernoulli, for different data types. A Bayesian approach is used in AutoClass to find out the optimal partition of the given data based on the prior probabilities.

- **A density-based algorithm for discovering clusters in large spatial databases with noise [23], by Ester et al.(1996)**
  
  This paper explains the original DBSCAN algorithm with a single processor. DBSCAN was developed to cluster large-scale data sets in the context of data mining. It requires that the density in a neighborhood for a data should be high enough if it belongs to a cluster. A new cluster from one data point is created by including all points in its neighborhood. The threshold of neighborhood of a data point is user-specific. DBSCAN uses $R^*_\ast$-tree structure for more efficient queries. The authors showed the effectiveness and efficiency of DBSCAN using synthetic data and SEQUOIA 2000 benchmark date as well.

- **BIRCH: A new data clustering algorithm and its applications [75], by Zhang et al.(1997)**
  
  This paper proposes a Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) as a new data clustering method using clustering feature (CF)-tree data structure, and suggests that this method can be easily parallelized. Because of growing number of data and concerns for the robustness to outliers, BIRCH is developed. The CF-tree is a height-balanced tree designed to store the summaries of the input. The tree structure stores the clustering information while requires less storage. After the CF-tree is constructed, an agglomerative hierarchical clustering is applied to perform global clustering, and the time complexity is linear.

- **An Efficient k-Means Clustering Algorithm [3], by Alsabti et al.(1998)**
K-means clustering algorithm is one of the partitioning approach clustering algorithm using square-error-sum measurements. Using a k-d tree structure, the paper proposes an efficient k-means algorithm.

  Most of clustering algorithms including K-means depend on the initial points in a great deal. This paper presents a process of refining initial points so that it leads to a better local minimum points.

- **Data clustering: a review**, by Jain et al. (1999)
  It also belongs to the first classification and in this class, clustering algorithms are classified based on their techniques, motivations and applications. This review provides a taxonomy of clustering approaches as a tree structure. The root is clustering and the two children nodes are hierarchical and partitional approaches. Single link and complete link are the children of hierarchical approaches. Partitional clustering have four children: square error, graph theoretic, mixture resolving and mode seeking approaches. K-means clustering is a child of square error, and expectation maximization (EM) approach is a child of mixture resolving. Applications of clustering algorithms are also described.

- **A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining**, by Huang (1995)
  This paper provides k-modes algorithm which extends the k-means methods to categorical domains, as k-means only deal with numerical data sets. Many of data mining applications consists of categorical data, not numerical data. Therefore, converting these data into numerical to apply conventional K-means algorithm often reveal meaningless results. Therefore the author of this paper suggests K-modes algorithm using new similarity measurements for the categorical objects.

  Previously, k-prototypes are proposed by Huang in 1997, to cluster large data sets with mixture of numeric and categorical measurements. One of the problems using k-prototypes algorithm is to choose a proper weight for the categorical measurements.

  The k-modes algorithm, therefore, is improved version of k-prototypes, as it only takes categorical attributes for the measurements. If there is any numeric attribute, then it can be converted into categorical attribute. The centers are called modes in k-modes algorithm, and the modes are updated with frequencies.

  The advantages of k-modes algorithm over k-means are efficiency and scalability. It is faster than k-means as it does not need the distance of each data, but need the frequencies. Also it is scalable as it only need to update the frequencies of a new data.

- **CURE: an efficient clustering algorithm for large databases**, by Guha et al., (2001)
  The authors provide a new algorithm called CURE which abbreviate Clustering Using Representatives. This is one of agglomerative hierarchical clustering algorithm and it deals
with large scale data sets like other hierarchical clustering algorithms including BIRCH [75] developed in recent years. While centroid-based hierarchical clustering algorithm such as BIRCH have the restriction in the shape of cluster, CURE is not limited to the shapes nor sizes of the clusters. CURE uses a set of well-scattered points to form each cluster, then the clusters are further shrunk toward the cluster center with an adjustable parameter to limit the effects of outliers. To reduce computational complexity, CURE use random sampling and partition techniques.

  
  This paper presents an efficient implementation of Lloyd’s k-means clustering algorithm, and call it the filtering algorithm using kd-tree data structure.

  The algorithm starts by storing all data points in a kd-tree, and maintains a subset of candidate centers. The candidate centers are filtered as they are passed to its children. As the kd-tree is constructed based on the data points, it does not need to be updated at each iteration, which saves the time overall.

- **An Introduction to Bioinformatics Algorithms [34], by Jones and Pevzner(2004)**

  This book focuses on the bioinformatics algorithms and the clustering and trees chapter (chapter 7) describes many clustering algorithms that have been developed for the bioinformatics fields. Hierarchical clustering, K-means clustering, CAST and many other clustering techniques are illustrated based on the bioinformatics data. This book, however, only deals with a single process clustering algorithms, not parallel clustering algorithms. However, most of parallelized clustering algorithms also can be applied in this bioinformatics data.

- **Survey of clustering algorithms [70], by Rui Xu and Wunsch, D., II (2005)**

  In this class, this paper surveys data clustering algorithms in various fields including statistics, computer science and machine learning and describes the applications. It includes some parallel algorithms as well. The author organized the various clustering algorithms as follows:

  - Hierarchical clustering (Agglomerative/Divisive)
  - Squared Error-Based
  - pdf Estimation via Mixture Densities
  - Graph Theory-based
  - Combinatorial Search Techniques-based
  - Fuzzy
  - Neural Networks-based
  - Kernel-Based

  Furthermore, it also describes sequential data clustering, and large-scale data clustering, in addition to visualizing of data and dealing with high-dimensional data.
• **Analysis of Biological Networks** [36], by Junker and Schreiber (2008)

Clustering algorithms are also very useful for biological networks such as Protein-Protein Interaction (PPI), Transcriptional Regulatory Network and Metabolic Networks. As they can be represented as a graph, the chapter of Network Clustering in this book uses graph structure for the network. Then it reviews Clique-based and Center-based clustering techniques for small data sets.

For the large data sets, it refers some techniques including distance k-neighborhood, k-cores and quasi-cliques as well.

• **A roadmap of clustering algorithms: finding a match for a biomedical application** [4], by Andreopoulos et al. (2009)

This paper separate the clustering algorithms into five categories: partitioning, hierarchical, grid-based, density-based, model-based and graph-based algorithms.

Partitioning clustering methods are useful for bioinformatics applications including gene expression data where a fixed number of clusters are required. The partitioning methods are further divided into numerical methods and discrete methods. K-means algorithm and Farthest First Traversal k-center (FFT) algorithm, K-medoids or PAM (Partitioning Around Medoids), CLARA (Clustering Large Applications), CLARANS (Clustering Large Applications Based Upon Randomized Search) and Fuzzy K-means belong to numerical methods. Discrete methods include K-modes, Fuzzy K-modes, squeezer and COOLCAT. K-prototypes is a mixed of discrete and numerical clustering methods. The algorithms except K-means are in fact modification of K-means algorithms with various purposes.

Hierarchical clustering algorithms divide the data into a tree of nodes, where each node represents a cluster. Hierarchical clustering algorithms are often divided into two categories based on their methods or the purposes: Agglomerative vs. Divisive; Single vs. Complete vs. Average linkage. In bioinformatics applications, hierarchical clustering methods are more popular as natures can have various levels of subsets, such as, representing protein family relationships. But hierarchical methods are slow, errors are not tolerable and information losses are common when moving the levels. Like partitioning methods, hierarchical methods consists of numerical methods and discrete methods. BIRCH, CURE and Spectral clustering are numerical methods while ROCK, Chameleon and LIMBO are discrete methods.

Grid-based clustering forms a grid structure of cells from the input data. Then each data is distributed in a cell of the grid. STING combines a numerical grid-based clustering method and hierarchical method.

Density-based clustering algorithms use a local density standard. Clusters are dense subspaces separated by low density spaces. Examples of bioinformatics application using density-based methods include finding the densest subspaces in interactome (protein-protein interaction) networks. DBSCAN, OPTICS, DENCLUE, WaveCluster, CLIQUE use numerical values for clustering. SEQUOPTICS is used for sequence clustering. HIERDENC (Hierarchical Density-based Clustering), MULIC (Multiple Layer Incremental Clustering), Projected (subspace) clustering, CACTUS, STIRR, CLICK, CLOPE use discrete values for clustering.
Model-based clustering uses a model which is often derived by a statistical distribution. The methods combine background information into gene expression, interactomes and sequences for bioinformatics applications. Self-Organizing Maps is an example of numerical model-based methods and COBWEB is a method of discrete model-based clustering algorithm. On the other hand, BILCOM (Bi-level clustering of Mixed Discrete and Numerical Biomedical Data) mixes numerical and discrete model-based clustering methods, using empirical Bayesian approach. Gene expression clustering and protein sequence clustering are the applications with this method. Other examples include AutoClass, SVM Clustering methods.

Graph-based clustering algorithms were applied to interactomes for complex prediction and to sequence networks. MCODE (Molecular Complex Detection) is for detecting subnetworks in an interactome. SPC (Super Paramagnetic Clustering) is similar to COOLCAT and RNSC (Restricted Neighborhood Search Clustering) is similar to ROCK and Chameleon. MCL (Markov Clustering), similar to projected clustering, is used for interactomes, by simulating a flow. Other methods include TribeMCL, SPC, CD-HIT, ProClust and BAG algorithms.

Those methods are used for bioinformatics applications. For gene expression clustering, k-means, hierarchical and SOMs have been used. Based on the problems encountered, alternative algorithms can be applied. For interactomes, AutoClass, SVM clustering, COBSEB or MULIC were used. For sequence clustering, hierarchical clustering algorithms are most appropriate.

3rd Class: Parallel Clustering Algorithms

The papers belong to this class propose parallelized version of various clustering algorithms. Most of the parallel algorithms for clustering tasks are done in the way of message-passing methods.

- Parallel algorithms for hierarchical clustering and cluster validity [47], by Li(1990)

The paper presents parallel hierarchical clustering algorithms on SIMD machine. It also proposes cluster validity computation. Traditional hierarchical clustering requires \( O(N^3) \) for \( N \) number of data points (or called patterns) in sequential. In the parallel version of single-link and complete-link hierarchical clustering algorithm use an alignment network, especially shuffle-exchange network. The data matrix (pattern matrix or proximity matrix) is first stored in the central memory then it is distributed to \( Q \) number of processor modules. With two storage rules proposed, the data matrix is stored in rows, columns, folds partial-row-pairs and partial-column pairs which is accessed in parallel. As a result, if \( N < Q \), then the time becomes \( O(N \log Q) \) and \( O(N^2 \log Q) \) for the simple-link and complete-link respectively. If \( N \geq Q \), then they become \( O(N^r N/Q^{\lceil \log Q \rceil}) \) and \( O(N^{2r} N/Q^{\lceil \log Q \rceil}) \).

- Clustering on a Hypercube Multicomputer [57], by Ranka and Sahni(1991)

This paper describes squared error clustering algorithms for SIMD hypercubes. With \( MN \) number of processors in hypercube network, where \( M \) is the number of features and \( N \) is
the number of data, the algorithm computes $K$ number of clusters in $O(K + \log NM)$. The square error for each cluster is defined the sum of square distance between a member in the class and the center. This can be understood as an intra-distance in each cluster. Therefore the objective function in the square error clustering algorithm is to minimize the sum of all intra-distances. The sequential algorithm starts with initial set of $K$ clusters, and move each pattern (or data) to a cluster if it minimizes the square error. With single processor, one iteration takes $O(NMK)$. To reduce the time complexity for a single iteration computing, several algorithms have been introduced. Li and Fang \[46\] introduced the algorithm which takes $O(K\log NM)$ for this one pass, using SIMD hypercube network. In the other hand, this paper uses subhypercube concept for broadcasting data or computing data. Therefore, the time is reduced into $O(K + \log NM)$ with $NM$ processors, and in $O(\log NMK)$ with $NMK$ processors.

- **Clustering with evolution strategies** \[5\], by Babu and Murty(1994)

  Evolutionary strategy (ES) is one of the best-known evolutionary techniques developed in 1981 by Schwefel \[61\]. It is population-based stochastic optimization technique tool, and is modeled using biological evolution concepts with selection, recombination and mutation. According to \[29\], if we set $\mu$ be the number of parents, $\lambda$ be the number of offspring, the ES model is represented as $(\mu, \lambda)$-ES. The early $(\mu, \lambda)$-ES model has only one parent and one offspring, that is, $(1+1)$-ES. Later multi-membered ES with $(\mu + 1)$-ES where $\mu > 1$ is proposed by introducing recombination operator. The $(\mu + \lambda)$-ES and $(\mu, \lambda)$-ES where $\mu > 1$ and $\lambda > 1$ were introduced for parallel computation of ES. When it select parents, $(\mu + \lambda)$-ES uses all individuals for the next population while $(\mu, \lambda)$-ES uses only the given offsprings.

  After brief description of how ES can be used to optimize a function, it discusses clustering with ES. In clustering, two types of objective functions are described: centroid type and non-centroid type. The centroid type of objective function, where objective function is based on the centroid of clusters, is used to find optimal cluster centers. ES is used to optimize this object function, in case of hard and fuzzy clustering as well.

  The non-centroid type of objective function is applied to discrete optimization problems. The entire mutation is controlled by global mutation and a solution string is an ordered sequence of $T$ discrete parameters, each of which is the cluster label of a data. Using ES, the objective function takes a solution string as input and outputs the clusters based form on the assignment.

  Parallel ES algorithm is described in terms of the master/slave model in section 6. Assume that the number of processors are $O(\mu)$. Then there is a master processor that maintains a number of parent solutions and processes selection operation and passes parent solutions to child(slave) processors. The child processors perform mutation and recombination processes and evaluate with the objective function. New solutions are sent back to the master process and it repeats the procedures until a near-optimal or optimal solution is found.

- **Graph Clustering:Complexity, Sequential and Parallel Algorithms** \[1\], by Abbas (1995)
It is one of graph theory-based clustering algorithm used for non-hierarchical clusters, according to the survey paper of [70]. It studies graph clustering on bipartite and chordal graphs. This paper describes two types of clustering problems: one is a partition of the vertices of a graph with minimum diameters of subgraphs; the other one is a subset of fixed number of vertices which induces maximum possible number of edges. Putting main interests onto the first problem, the author proves that the partitioning problem is NP-complete on bipartite and chordal graphs. But, this paper also proposes linear time sequential algorithms on bipartite permutation and interval graphs. By proving and giving efficient parallel algorithm for constructing a certain ordering of the vertices on the biconvex graphs, called a biconvex straight ordering, the author presents efficient parallel algorithms for the general problem on biconvex and interval graphs. At last, the paper also provides an efficient parallel algorithm for solving the second problem on a class of unit interval graphs.

- **Parallel Algorithms for Hierarchical Clustering** [53], by Olson (1995)

This paper discusses the parallel algorithms for agglomerative hierarchical clustering tasks which will return a dendrogram as a result. The section that reviews sequential algorithms, two distance metrics considered: Graph metrics which determine inter-cluster distances according to the cost functions of the edges of members in the two clusters, including single, average and complete link; Geometric metrics which determines intra-cluster distances between the cluster center and the members including centroid, median and minimum variance, etc. Sequentially, clustering with the single link metric is mostly done by finding the Euclidean minimal spanning tree and the time is $O(n^2)$ in [73]. For the median and centroid metrics, Day and Edelsbrunner [18] showed that it takes overall $O(n^2)$ time as well. In general case, if the modified inter-cluster distance can be updated in a constant time, the clustering takes $O(n^2 \log n)$ time as shown in [18].

Then the author reviews previous parallel algorithms as well. The previous parallel clustering algorithm with single linkage metric on SIMD array processor was discussed by Rasmussen and Willett [58]. They parallelized SLINK algorithm [63], Prim’s minimal spanning tree algorithm [56] and Ward’s minimum variance method [67]. Li and Fang [46] parallelized hierarchical clustering using the single link metric on hypercube network and butterfly, by parallelizing the Kruskal’s minimal spanning tree algorithm [43]. The $n$-nodes hypercube takes $O(n \log n)$ time while the $n$-nodes butterfly takes $O(n \log^2 n)$ time. However, these times did not consider the step of updating distances of clusters, which will increase the time to $O(n^2)$. The relaxed heap introduced by Driscoll et al. [22] is used for parallel implementation of Dijkstra’s minimal spanning tree [21] and reduces the time into $O(n \log n)$ with $m/(n \log n)$ processors on a PRAM. Bruynooghe [11] describes a parallel nearest neighbors clustering algorithm for a parallel supercomputer, using message-passing master/slave model.

For each of the metrics on PRAMs and butterflies, the parallel implementations are described in the next section. With single-linkage, centroid and median metrics, finding MST techniques are used. According to the algorithm introduced in this paper, the inter-cluster distance and nearest neighbor arrays are obtained in $O(n/p)$ at each step, where $n$ is the number of data and $p$ is the number of processors. As the number of steps is $O(n)$, total
time would be $O(n^2/p)$. With minimum variance, average and complete link metrics, each processor store the location of each cluster center and find the nearest neighbor in $O(\log n)$ with $O(n/\log n)$ processors, leading total $O(n\log n)$ time. For the general metrics, create a priority queue in $O(n)$ time on a processor and in $O(\log n)$ time on a $n$ processor PRAM. After each iteration the priority queue is updated, total $n$ steps lead $O(n\log n)$ time for updating queue.

- **Large-Scale Parallel Data Clustering**, by Judd et al.(1998)

  In this paper, algorithmic enhancements are described which reduces large computational efforts in mean square-error data clustering. These improvements are incorporated into a parallel data-clustering tool. The basic Parallel-CLUSTER (P-CLUSTER) algorithm was first reviewed in this paper, then provides the results of tasks on tow NOWs (network of workstations) platforms. The basic CLUSTER program consists of two components of a $K$-means pass and a forcing pass. In a $K$-means pass, after the first clustering is created, the subsequent clusterings are created based on the previous clustering. Then in a forcing pass, another set of clusters by merging existing clusters is created, two at a time, to determine whether a better clustering is being achieved. The basic P-CLUSTER algorithm uses a client-server(or master-slave) process configuration. The data set is partitioned into blocks by the server process, sending the initial centroid list and job assignments to each of the clients. Each client process each block of data set to assign it to the appropriate cluster. The server then collect the information of each clients and calculates new centers to send them back to clients. By observing that after a small number of passes, relatively few changes occur, the authors of this paper proposes three techniques to reduce the number of distance calculations to reduce the total computational time. One is to pre-compute a set of radii for each centroid, called sphere of guaranteed assignment. The second technique is using maximum movement effect. That is, if there is no changed membership for a pattern then all computation for the pattern is avoided. Lastly, partial sum technique is used to reduce additional computations for floating point addition: they maintain a set of partial sums associated with each block of data. Patterns that change assignment are subtracted from the old set of sums and added to the new set. So a computation is carried out only when a memberships change takes place.

- **An Distributed clustering for ad hoc networks**, by Basagni.(1999)

  Traditionally, clustering in the network is performed in two steps: clustering setup by picking clusterheads and clustering maintenance by collecting the neighbors of clusterheads. However existing clustering algorithms are based on the assumption that the nodes are not moving in the process. Mobility-adaptive algorithm is introduced for mobile networks, whose decision is based on the local topology of head’s neighbors with the largest degree.

  In this paper, a distributed clustering algorithm (DAC) and a distributed mobility-adaptive (DMAC) algorithm are proposed. The authors partition the nodes in a fully mobile network or ad hoc network, and organize a hierarchical structure.

  The DAC generalizes the common clustering in network, and it is executed at each node, each of which communicate with others with message-based approach. It selects the clusterheads
based on the new weight-based criterion, and each node decides its role with its current one hop neighbors as opposed to one and two hop neighbors by previous algorithms.

As DAC algorithm is suitable for the quasi-static networks, the author propose DMAC for highly mobile networks. Without the assumption that during the clustering process the nodes are not moving, DMAC lets each node to properly react to the failure of a link with another node on top of the messages from others.

• **Parallel K-Means Clustering Algorithm on NOWs** [38], by Kantabutra and Couch.(1999)

Instead of a serial K-means clustering algorithms, this paper reduces the time complexities by applying parallel algorithms.

Using the intra-cluster, (inner summation) and inter-cluster (outer summation) criterions, the paper extend the conventional serial K-means algorithms to parallel algorithms. The parallel K-means algorithm is developed on the message-passing model of a network of workstations (NOWs).

The master process first randomly form $K$ equal subsets and send each to each $K$ slaves. Each slave process compute the mean of each subset then broadcast it to other slaves. Each slave compute the distance of each data from each mean, then choose the closest vector members for each $K$ subset, which is also broadcast. Then for the new subset by collecting those closest members, and send it back to the master process.

The total time estimated with this algorithm is $O(2R_p K |P|)$ where $R_p$ is the number of iterations and $|P|$ is the size of subset and the total space complexity is $O(N)$, where $N$ is the number of data.

• **A Fast Parallel Clustering Algorithm for Large Spatial Databases** [72], by Xu et al. (1999)

This paper first illustrates DBSCAN algorithm which is a density-based algorithm for discovering clusters in large spatial databases. DBSCAN is one of density-based clustering algorithms, which can have arbitrary shape of clusters and efficient enough to be used for large spatial data sets as it uses spatial access methods. The authors then introduce PDBSCAN, a parallel DBSCAN algorithm using shared-nothing architecture with multiple computers interconnected through a network. As a data structure, the dR*-tree is introduced for a distributed spatial index. The parallel model is master/slaves model. A master program starts a clustering slave on each available machine and distributes the data set to the slaves. Each slave clusters only its data locally. The data structure for index gives an efficient access of data and the communication between processors is message-passing model. PDBSCAN algorithm has three steps: dividing the input data into several partitions, cluster partitions concurrently using DBSCAN, and merging the local clusters into global clusters for the whole data. The results suggests that PDBSCAN offers nearly linear speedup.

• **Parallel Algorithms for Hierarchical Clustering and Applications to Split Decomposition and Parity Graph Recognition** [17], by Dahlhaus.(2000)
The authors provide parallel algorithms for hierarchical clustering methods and apply it to split decomposition in graphs. The paper observes that both of a single linkage method and the algorithm determining overlap components can be the sub-procedures in an efficient parallel algorithm for split decomposition. Then it provides a parallel algorithm for the single linkage method, computing a minimum spanning tree (MST) with multiple processors. The overlap components can be determined in linear time sequentially and in $O(\log n)$ with $O(n)$ processors using parallel prefix computation. Finally, an efficient parallel split decomposition algorithm is presented using the above two algorithms.

- **A Data-Clustering Algorithm on Distributed Memory Multiprocessors** [20], by Dhillon and Modha. (2000)

  The authors of this paper propose a parallel implementation of the k-means clustering algorithm using the message passing model.

  The motivation comes from the challenges of large scale data mining. The parallel k-means algorithm in this paper designed based on the single program multiple data model using message-passing model. Message-passing model assumes that $P$ processors are connected with its own local memory. After initial $k$ centers are selected, $n$ data points are divided into $P$ blocks. Observing that each processor can carry out the distance calculation in parallel if the centers are available to them, each processor deals with only $n/P$ number of data points to assign to clusters. Each iteration consists of an asynchronous computation phase and a synchronous communication phase. In conclusion, the paper shows a nearly linear speedups with multiple processors, and it is also scalable with increasing data points.

- **Distributed data clustering can be efficient and exact** [24], by Forman and Zhang. (2000)

  This paper presents a technique of clustering large and distributed data sets using efficient distributed algorithms. It illustrates the parallelized center-based data clustering algorithms.

  A center-based clustering algorithms iteratively locate abstract centers into partition or into fuzzy clusters by the local density functions.

- **Scalable Parallel Clustering for Data Mining on Multicomputers** [25], by Foti et al. (2000)

  This paper illustrates the design and implementation of parallel version of AutoClass system [13], which is called P-AutoClass. The authors selected the SPMD (Single Program Multiple Data) approach for the P-AutoClass. The SPMD parallel can be used in AutoClass by dividing the input into the processors and updating parameters of classifications. It avoids replicating the whole input on each processor and also avoids balancing problems as each processor perform the same function on equal size of data. In addition, not much exchanges are necessary as they are locally executed.

- **A scalable parallel subspace clustering algorithm for massive data sets** [50], by Nagesh et al. (2000)
• **Parallel implementation of CLARANS using PVM** [76], by Zhang et al. (2004)

This paper introduces a parallel CLARANS (Clustering Large Applications Based Upon Randomized Search) algorithms using PVM (Parallel Virtual Machine). PVM supports a complete message-passing model and make a collection of distributed computers function as a single machine. In the PVM, all the computers are working in master/slave mode. Master program can assign tasks to other slaves and the communication between computers is based on the message-passing. CLARANS algorithm randomly draw a sample at each iteration, and the $K$-Medoids are chosen from this sample. In PVM, the master program assigns each tasks to $K$ slaves, and each slave randomly select a neighbor of each center and compute cost differential of the two nodes. With the smaller cost, the center can be replaced. This parallel version of CLARANS is called ICLARANS (Improved CLARANS) and it outperforms CLARANS in terms of time complexity, and scalability.

• **PBIRCH: A Scalable Parallel Clustering algorithm for Incremental Data** [27], by Garg et al. (2006)

This paper describes a parallel BIRCH [75] clustering algorithms. PBIRCH algorithm is applied to the Single Program Multiple Data (SPMD) model with message-passing. First, the data is divided equally into the processors. Each processor constructs each CF-tree. Initially one processor chooses $k$ initial centers for k-means algorithm, broadcast it to all processors. Then each processor use the initial centers to cluster its own CF-tree locally. The centers of the local clusters are exchanged, then the global centers are computed. Now having $k$ global centers, each processor recomputes the clusters and the processes are repeated until it converges.

If new data are entered they are cyclically distributed as well.

4th Class: Applications of Parallel Clustering Algorithms

Several applications of clustering algorithms are classified in this category.

• **Image segmentation by a parallel, non-parametric histogram based clustering algorithm** [40], by Khotanzad and Bouarfa(1990)

This paper presents a non-parametric parallel clustering algorithm for the task of unsupervised image segmentation. It builds a multi-dimensional histogram of the feature vectors to see the distribution of feature vectors and approximate the density of the data. Each peak is considered as a cluster. The process of identifying clusters are automatic as it proposes systematic methods for choosing an appropriate histogram cell size. After the clustering, the segmentation task is followed and they are implemented in a SEQUENT parallel computer. The parallel computation is done using message-passing protocol as well.

• **A parallel bottom-up clustering algorithm with applications to circuit partitioning in VLSI design** [16], by Cong and smith.(1993)
The paper proposes a parallel bottom-up clustering algorithms and apply it to partition circuits in VLSI. Unlike many top-down partitioning algorithms for layouts, the bottom-up layout clustering algorithms are useful as circuit sizes increase. The parallel clustering algorithm is to divide the graph into multiple processors. Each processor forms clusters of the graph, occasionally swaps parts so that cliques to be found. The bottom-up clustering tasks also are used as preprocessing for partitioning as each cluster is then considered each node. This saves time and space complexity a lot.

- **An energy efficient hierarchical clustering algorithm for wireless sensor networks [7], by Bandyopadhyay and Coyle.(2003)**

The paper presents a distributed and randomized clustering algorithm to layout the sensors in a wireless sensor network. On top of the clustering network sensors, the authors organize the clusterheads hierarchically, so that it saves the energy more. It assumes the sensors in a network are simple and all the sensors communicate at a fixed power level. Using the experimental fact that a network with one level of clustering has an optimal number of clusterheads minimizing the energy, this paper proposes an analysis to obtain the optimal number of clusterheads at each level of clustering.

### 5 5th Class: Applying Clustering Algorithms for Parallelization

The input data is first clustered into smaller subsets for the parallelization of other algorithms.

- **Cluster-partitioning approaches to mapping parallel programs onto a hypercube [60], by Sadayappan and Ercal(1988)**

This paper addresses the task-to-processor mapping problem, which is essentially NP complete. In other words, clustering task is used for distributing tasks for parallel computations. Assuming that the multiprocessors are connected with a hypercube interconnection network, two cluster-based mapping strategies are introduced: a nearest neighbor approach and a recursive clustering.

There are two distinct settings for the task-mapping problem: Task Precedence Graph (TPG) model and Task Interaction Graph (TIG) model. Each model can be represented as a graph.

The vertices in a TPG graph represent tasks, and directed edges represent execution dependencies. Vertex weight is the execution time and edge weight is the amount of data to be communicated between vertices.

The parallel program modelled with TIG graph, however, does not capture the execution dependencies between tasks. Therefore, the TIG model is more suitable for more complex parallel programs.

The paper focuses on static mapping with TIG model. The parallel computer is also represented as a graph, where the vertices represent the processors and the edges represent the communication link. Between a processor and its neighbor has a communication cost $T_s$ and a data transmission cost which is the product of the data size and the transmission rate $T_t$. 


The nearest-neighbor mapping strategy maps any pair of nodes sharing an edge in TIG to nearest-neighbor processors. It starts from an initial mapping of clustering nodes of TIG onto the clusters of processors having nearest-neighbor properties. Then it modifies the mapping by reassigning with a boundary-refinement process to improve the mapping cost defined in some way.

The recursive-clustering strategy has two phases: cluster formation and processor allocation. In the cluster formation phase, the TIG is first partitioned into as many clusters as the number of processors. It is done with top-down fashion, recursively. Then the clusters are allocated to some processors in order to minimize the total inter-processor communication cost.

The paper concludes by comparing the two schemes. The nearest-neighbor mapping tries to improve load balance and implicitly reduces the communication cost, while the recursive-clustering strategy tries to minimize communication cost and implicitly improves the load balance. Therefore, the former one is more appropriate for the system with high message start-up time and for the finite element graphs, while the latter is more proper for the system with lower message start-up times and for the random task graphs.

• Hierarchical parallel coordinates for exploration of large datasets [26], by Fua et al. (1999)

This paper focuses on the interactive visualization of large multivariate data sets based on a number of extensions to the parallel coordinates display technique. The parallel coordinates display technique represent each data dimension as a vertical axis, and the multiple axes are organized as uniformly spaced lines. A data is mapped to a poly-line that traverse across all of the axes at a position proportional to its value for that dimension. Over the conventional orthogonal technology, the parallel coordinates have an advantage of visualizing the data only by the horizontal resolution of the screen, making it scalable. However, as the data size increases, the level of clutter present in the visualization restricts the amount of information that one can perceive. Therefore, the hierarchical organization on the data set which is proposed in this paper, reduces the amount of clutter. The data sets are hierarchically clustered according to each dimension, and the structure is formed as a tree. By adjusting the horizontal line, different levels of parallel coordinates displays are possible, and they can be used for filtering tools. That is, the multiresolutional view of the data through hierarchical clustering is developed and a variation on parallel coordinates to carry aggregation information for the clustering results is used. By navigating the structure, users can find the right level of details. In addition to the visualization, with this technique, users can interactively uncover data patterns as well.

• Efficient clustering of large EST data sets on parallel computers [37], by Anantharaman et al. (2003)

This paper develops a PaCE which is parallel and fast clustering method of large-scale EST (Expressed Sequence tags) in gene identification. ESTs are sequences at most a few hundred base pairs determined by single-pass sequencing of the 5’ or 3’ ends of cDNA. EST representations provide a dynamic view of genome content and expression. With the huge EST data library, more than 5 million human ESTs, the primary goal in ESTs is to derive sets of
unique genes from the data which represent the transcriptome of each species. Therefore, the first step would be clustering ESTs to remove redundancy in EST sets. The authors of this paper developed a parallel software program, named PaCE, which clusters ESTs into the same gene or paralogous genes. Then each cluster is further processed with an assembly tool for generating one consensus sequence per putative transcript. The memory with PaCE in \(O(n)\) and the input size of the assembly tools is now reduced to the size of each cluster.

The clustering with PaCE is a bottom-up approach, that is, each pair of clusters are grouped recursively.

The PaCE software has two phases: GST(generalized suffix tree) construction as a preprocessing phase, and pair generation, pairwise alignment and EST cluster management as the clustering phase. That is, a distributed representation of the GST is first built in parallel. One processor, master processor, maintain and updates the EST clusters and the rest of processors generate batches of promising pairs and perform pairwise alignment on selected promising pairs. Master process is also responsible for selecting promising pairs at each step.

- **FGK model: A Efficient Granular Computing Model for Protein Sequence Motifs Information Discovery** [14], by Chen et al.(2006)

In this paper, two clustering algorithms have been used. Fuzzy-C means algorithm is exploit to divide the large data set into smaller subsets so that parallel computation is possible. Each processor then run k-means clustering algorithms to find out protein sequence motifs. Protein motif sequence is a repeated pattern in a set of protein sequences, which are potentially important biologically and chemically. However, with large data set, discovering motifs becomes hard as the task of grouping the proteins with similar patterns should proceed. Hence, in this paper, a number of protein sequence which represent its protein family information is segmented by shifting 9 window size, and they are clustered first to find consensus at each cluster. The consensus sequence at each cluster is now considered a potential sequence motif.

References


[31] Zhexue Huang. Clustering large data sets with mixed numeric and categorical values. In In The First Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 21–34, 1997.


