Comparing clustering and partitioning strategies

Carlos Afonso, Fábio Ferreira, José Exposto, and Ana I. Pereira

Citation: AIP Conference Proceedings **1479**, 782 (2012); doi: 10.1063/1.4756254

View online: http://dx.doi.org/10.1063/1.4756254

View Table of Contents: http://scitation.aip.org/content/aip/proceeding/aipcp/1479?ver=pdfcov

Published by the AIP Publishing

Comparing Clustering and Partitioning Strategies

Carlos Afonso[+], Fábio Ferreira[+], José Exposto[+], Ana I. Pereira[*+]

[+] *Polytechnic Institute of Bragança, Portugal*
[*] *ALGORITMI, University of Minho, Portugal*

Abstract. In this work we compare balance and edge-cut evaluation metrics to measure the performance of two well-known graph data-grouping algorithms applied to four web and social network graphs. One of the algorithms employs a partitioning technique using Kmetis tool, and the other employs a clustering technique using Scluster tool. Because clustering algorithms use a similarity measure between each graph node and partitioning algorithms use a dissimilarity measure (weight), it was necessary to apply a normalized function to convert weighted graphs to similarity matrices.

The numerical results show that partitioning algorithms behave clearly better than to the clustering counterparts when applied to these types of graphs.

Keywords: Clustering. Partitioning. Web graph.
PACS: 02.60.Pn

INTRODUCTION

Clustering and partitioning techniques are used in a very range of applications such as the partitioning of the Web space for cooperative crawling [1], VSLI circuits [2], image compression based on fuzzy clustering [3] and data mining [4].

In this work we propose to compare the Kmetis [5] and Scluster [6] tools. In order to convert weight assigned to an edge to similarity values used by the clustering tools, we implemented a normalization function. The manipulation of graphs, loading and saving from/to file is based on the Jung framework [7]. We also developed a tool to interconnect all the formats using Java language.

The evaluation metrics of the resulting partitioning and clustering files, produced through the application of each algorithm, were the balance and the edge-cut. The balance measures, how well distributed are the vertices of the graph by the obtained clusters/partitions after the algorithm is applied. If the balance value equals one it means all partitions have the same number of vertices. The edge-cut is the sum of the weights of the edges cut by the obtained partitions.

The datasets used for the experiments are high dimensions undirected graphs which represent Web graphs and social networks [8].

CLUSTERING AND PARTITIONING TECHNIQUES

Partitioning is a technique for dividing a data group. For a good partitioning, the number of cut edges between the partitions created, should be as small as possible. One of the concerns of partitioning is a division of data into balanced groups [9].

Clustering is a technique used to identify sample groups that show the same behavior or similar characteristics. The objects within a group are similar between them, and different from the objects in other groups [10].

According to [2] the main difference between clustering and partitioning is that clustering typically implies a bottom-up cell grouping mechanism that generates a large number of small groups (clusters), while partitioning implies a top-down cell grouping mechanism that results in a small number of large groups (parts).

The handling of the graphs of the used datasets was made by the WebGraph's API. The loading of graphs is implemented with the developed tool "GraphReader" in Java, as well as the conversion to Kmetis and Scluster graph syntax, to apply the algorithms.

In addition to the algorithms discussed in this work, Jung's VoltageClusterer was also tested. This algorithm revealed several inconsistencies, which led to its withdrawal: (i) the number of requested partitions/clusters did not match the number of the obtained partitions/clusters; (ii) compared to the remaining algorithms, the execution time of VoltageClusterer is extremely slow.

Numerical Analysis and Applied Mathematics ICNAAM 2012
AIP Conf. Proc. 1479, 782-785 (2012); doi: 10.1063/1.4756254
© 2012 American Institute of Physics 978-0-7354-1091-6/$30.00

Evaluation metrics

The graphs used for the experimental results are characterized by having unit weights associated to the vertices, varying the weights associated to edges. The calculation of the balance is processed as follows:

$$\text{Balance} = \frac{\text{higher value} \sum \text{weights per (partition, cluster)}}{\sum \text{weights vertexes}} * \text{num. (partitions, clusters)}$$

and for the edge-cut is used the following expression

$$\text{Edge-cut} = \sum \text{weights cut edges.}$$

Normalized function

The Metis syntax refers that a graph has a set of vertices (nodes) and edges (arcs). Associated to the vertices and the edges we can have weights, and these must be integers greater or equal to zero.

The Cluto syntax uses an adjacency matrix to store a graph. The biggest difference to Metis syntax is each node has a similarity value, which represents the affinity of one node to another.

In this sense it was decided to make a comparison between both techniques by applying a hypothetical formula to normalize the weights, Equation (1). The calculation allows the function returns the similarity value associated with the connection. Thus, it is possible to convert the graphs of the Metis for Cluto and allow a fair comparison between them.

$$\mathrm{X}\mathit{similarit}\mathrm{y} = \left| \frac{Xi - Xmax}{Xmax - Xmin} \right| \tag{1}$$

After loading the graph, in Metis's format, the maximum weight (*Xmax*) and the minimum weight (*Xmin*) of the graph is calculated. The variable Xi is the weight associated to the link between nodes. Applying the formula, the data is normalized and it is assigned a value (X*similarity*) between 0 and 1 to the link.

NUMERIC RESULTS

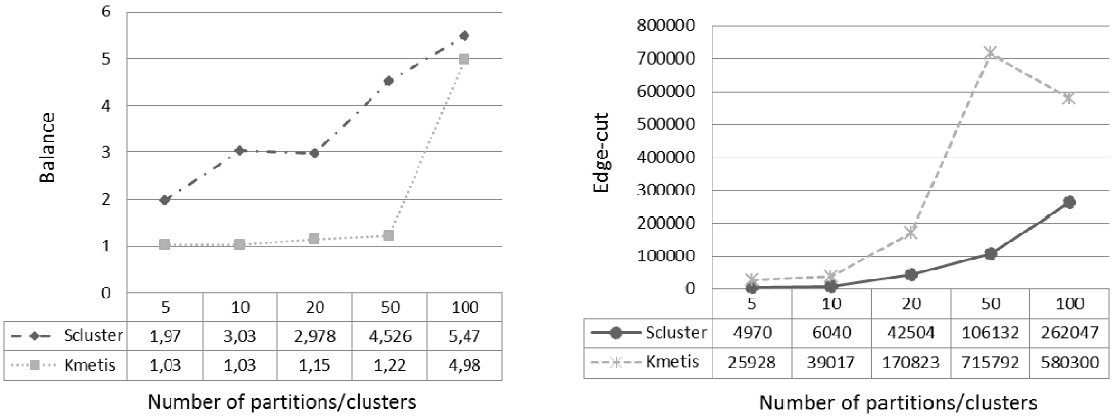
To make the comparison tests we used a virtual machine with a 2.93GHz quad-core (Intel$^{(R)}$ Core$^{(TM)}$ i7-870) and 8GB of RAM.

TABLE 1. Datasets Characteristics

Name of graph	*# Nodes*	*# Arcs*	*Description*
cnr-2000	325557	3216152	A very small crawl of the Italian CNR domain.
dblp-2010	326186	1615400	DBLP is a bibliography computer service
amazon-2008	735323	5158388	A graph describing similarity among books as reported by the Amazon store.
dblp-2011	986324	6707236	DBLP is a bibliography computer service

To conduct the experiments we applied Kmetis and Scluster algorithms to the described dataset using five different number of partitions/clusters (5, 10, 20, 50, 100), measuring both balance and the edge-cut.

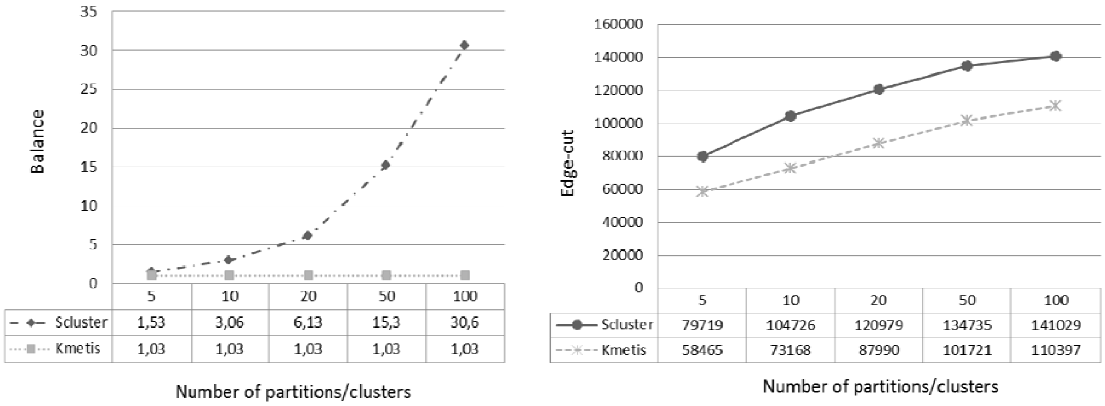
Graph: cnr-2000



	5	10	20	50	100
Scluster	1,97	3,03	2,978	4,526	5,47
Kmetis	1,03	1,03	1,15	1,22	4,98

	5	10	20	50	100
Scluster	4970	6040	42504	106132	262047
Kmetis	25928	39017	170823	715792	580300

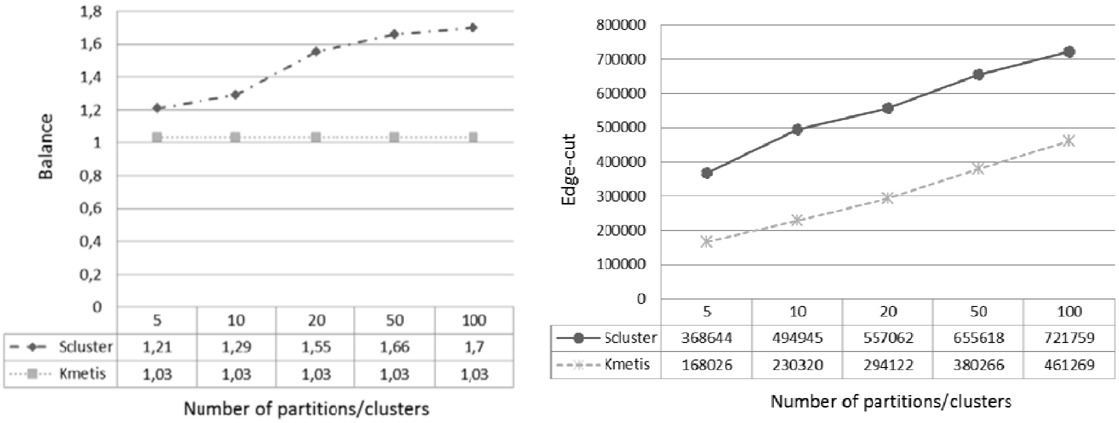
Graph: dblp-2010



	5	10	20	50	100
Scluster	1,53	3,06	6,13	15,3	30,6
Kmetis	1,03	1,03	1,03	1,03	1,03

	5	10	20	50	100
Scluster	79719	104726	120979	134735	141029
Kmetis	58465	73168	87990	101721	110397

Graph: amazon-2008



	5	10	20	50	100
Scluster	1,21	1,29	1,55	1,66	1,7
Kmetis	1,03	1,03	1,03	1,03	1,03

	5	10	20	50	100
Scluster	368644	494945	557062	655618	721759
Kmetis	168026	230320	294122	380266	461269

Graph: dblp-2011

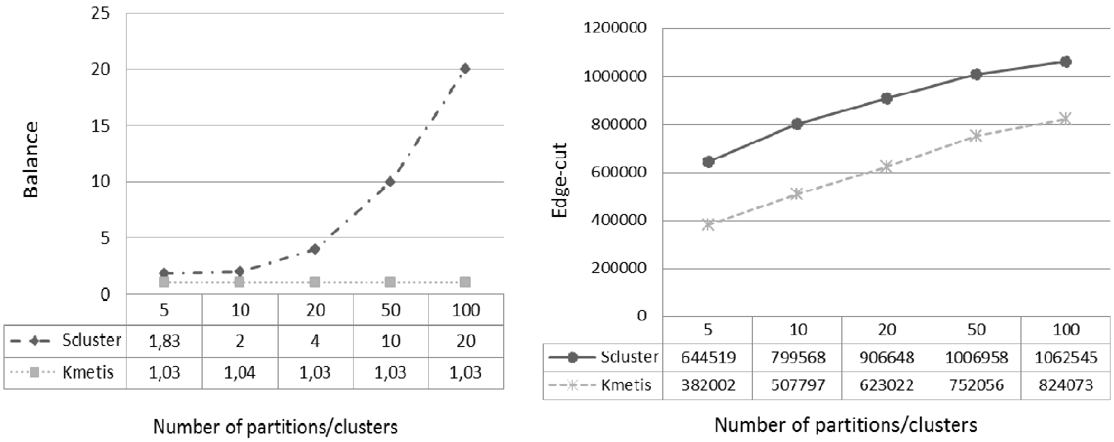


FIGURE 1. The charts on the left show the balance, and the graphics on the right, the edge-cut, by two different algorithms.

Analyzing Fig. 1, we can see a clear advantage for the partitioning algorithm. Kmetis always obtained a better balance, having almost always a consistent value near 1. The edge-cut was better on graph cnr-2000 for Scluster, but for the other three graphs, Kmetis achieved consistent better results.

Furthermore, Kmetis always performed better in terms of execution time for all tests, results which we omit because of space considerations.

CONCLUSIONS AND FUTURE WORK

The partitioning algorithm generally obtained much better results than the clustering algorithm for the used datasets. Balance is always better for the partitioning algorithm, making it most adequate when the equilibrium among the partitions is a priority. The partitioning algorithm also outperformed the clustering algorithm in the execution time. Although, the partitioning algorithm did not always outperformed the clustering algorithm for the edge-cut, it obtained consistent better results.

In the future we plan to test even more web and social graphs to clarify any inconsistencies that may arise, in particular, a deeper study that may better distinguish the suitability of each algorithm to each graph category: web or social network graph.

REFERENCES

1. J. Exposto, J. Macedo, A. Pina, A. Alves, J. Rufino, "Geographical Partition for Distributed Web Crawling" in GIR'05, November 4, Bremen, Germany (2005).
2. H. Vaishnav, M. Pedram, "Delay-Optimal Clustering Targeting Low-Power VLSI Circuits", IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 18, No. 6, June (1999).
3. M. Kaya, "An Algorithm for Image Clustering and Compression", Turk J Elec Engin, Vol.13, No.1, TÜBİTAK (2005).
4. P. Berkhin, "Survey of Clustering Data Mining Techniques", Accrue Software, Inc. (2002).
5. G. Karypis, V. Kumar, *http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/manual.pdf*, Metis version 4.0, University of Minnesota, Minneapolis, MN 55455 (1998).
6. G. Karypis, *http://glaros.dtc.umn.edu/gkhome/fetch/sw/cluto/manual.pdf*, Cluto release 2.1.1, University of Minnesota, Minneapolis, MN 55455 (2003).
7. S. White, J. O'Madadhain, D. Fisher, *http://jung.sourceforge.net/*, University of California, Irvine (2003).
8. P. Boldi, S. Vigna. *http://webgraph.dsi.unimi.it/*, The WebGraph framework I: Compression Techniques (2004).
9. K. Andreev, H. Räcke, "Balanced Graph Partitioning", 16th Annual ACM Symposium on Parallelism in Algorithms and Architectures (2004).
10. H. Lin, *http://www.cs.cmu.edu/afs/andrew/course/15/381-f08/www/schedule.html*, Clustering, Artificial Intelligence, Carnegie Mellon's School of Computer Science (2010).