BAB 6

KESIMPULAN DAN SARAN

Skripsi ini membangun sebuah perangkat lunak yang dapat melakukan *cluster analysis* dalam sebuah set data berukuran besar menggunakan metode agglomerative clustering .Hal tersebut di implementasikan di dalam perangkat lunak yang dibangun dengan memanfaatkan infrastruktur MapReduce pada sistem terdistribusi Hadoop.

Dalam skripsi ini, penulis telah melakukan studi terhadap teknik-teknik agglomerative clustering untuk konstruksi dendrogram, melakukan eksperimen-eksperimen awal untuk memahami kinerja dari MapReduce, merancang dan mengimplementasikan teknik-teknik konstruksi dendrogram dengan MapReduce pada sistem terdistribusi Hadoop, menguji kualitas dari perangkat lunak yang telah dibuat, membandingkan performa perangkat lunak yang terdistribusi dengan versi *standalone*, dan menguji perangkat lunak untuk menganalisis big data dalam sebuah studi kasus.

Perangkat lunak yang memanfaatkan infrastruktur MapReduce pada sistem terdistribusi Hadoop berhasil dibangun dan mampu melakukan konstruksi dendrogram dan pengelompokan pada big data dengan algoritma agglomerative. Dengan itu, tujuan-tujuan penelitian pada skripsi ini telah tercapai beserta dengan hasil-hasilnya.

6.0.1 Kesimpulan

Dari hasil eksperimen algoritma hierarchical clustering dengan agglomerative clustering untuk konstruksi dendrogram pada sistem terdistribusi Hadoop, terdapat beberapa kesimpulan yang dapat diambil, yaitu:

1. Aproksimasi *cluster* dengan dendrogram *align* dan *label* tidak 100% akurat. Hal ini mungkin disebabkan partisi data yang tidak seimbang (satu partisi mendapatkan mayoritas data) karena partisi dilakukan secara *random*. Ukuran partisi yang tidak seimbang ini menyebabkan dendrogram lokal tidak seimbang dan hasil dendrogram *align* menjadi tidak akurat.

   Untuk meminimalisasi hal itu maka jumlah data sebaiknya cukup besar agar partisi *random* akan cenderung membagi data secara merata. Pemilihan jumlah partisi juga harus dilakukan dengan hati -hati karena bisa sangat berpengaruh pada hasil akhir.

2. Waktu komputasi perangkat lunak akan lebih kecil dari agglomerative clustering *standalone* jika jumlah data besar. Hal ini dikarenakan adanya *overhead* dari sistem terdistribusi Hadoop seperti seperti komunikasi antar daemon ,membaca file dari komputer lain dan sebagainya.

3. Semakin banyak partisi m yang digunakan , maka semakin cepat proses kinerja perangkat lunak. Hal ini disebabkan karena kompleksitas MapReduce adalah $O((\frac{N}{m})^2)$. dan kompleksitaks dendgroram align adalah $O(N)$.

4. Jika ukuran file input terlalu besar bisa terjadi *error* pada tahap *align* dan *label* karena kehabisan *memory*. Penyebabnya adalah ukuran file yang dihasilkan pada tahap MapReduce terlalu besar (diatas 5 GB).

Hasil dari keseluruhan penelitian adalah sebagai berikut:

1. Algoritma agglomerative clustering untuk konstruksi dendrogram yang dibutuhkan dalam membuat perangkat lunak telah dipahami dan dipelajari melalui studi literatur.
2. Arsitektur Hadoop, terutama pada MapReduce, telah dipahami dan dipelajari melalui studi literatur dan eksperimen.
3. Perangkat lunak sudah diimplementasikan dan mampu melakukan *aglomerative clustering* pada sistem terdistribusi Hadoop
4. Eksperimen untuk menguji performa perangkat lunak yang terdistribusi diangdingkan dengan versi *standalone* sudah dilakukan. Untuk perangkat lunak yang terdistribusi waktu komputasi lebih kecil jika ukuran data besar.

6.0.2 Saran

Agar penelitian ini dapat dikembangkan lebih lanjut, saran yang diusulkan oleh penulis adalah:

1. penelitian ini dapat dikembangkan untuk menerima input dengan atribut selain bilangan *real*.
2. Modifikasi algoritma MapReduce agar file output yang dihasilkan ukurannya tidak terlalu besar.
3. Modifikasi algoritma dendrogramAlign agar dapat dilakukan secara terdistribusi.

DAFTAR REFERENSI

[1] Wu, X., Zhu, X., Wu, G.-Q., dan Ding, W. (2014) Data mining with big data. *IEEE Trans. on Knowl. and Data Eng.*, **26**, 97–107.

[2] Feldman, R. dan Sanger, J. (2006) *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, New York, NY, USA.

[3] Spiliopoulou, M., Wang, H., Cook, D. J., Pei, J., Wang, W., Zaïane, O. R., dan Wu, X. (ed.) (2011) *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, BC, Canada, December 11, 2011*. IEEE Computer Society.

[4] Han, J. dan Kamber, M. (2006) *Data Mining: Concepts and Techniques*, second edition. Morgan Kaufmann Publishers, 500 Sansome Street, Suite 400, San Francisco, CA 94111.

[5] Lin, Y., Chen, Q., dan Zhang, X. (2014) Numerical analysis based fast intra prediction algorithm in hevc. SPAC, 376-380(2014).

[6] Lam, C. (2010) *Hadoop in Action*, 1st edition. Manning Publications Co., Greenwich, CT, USA.

[7] Murthy, A. C., Vavilapalli, V. K., Eadline, D., Niemiec, J., dan Markham, J. (2014) *Apache Hadoop YARN: Moving Beyond MapReduce and Batch Processing with Apache Hadoop 2*, 1st edition. Addison-Wesley Professional.

[8] Dean, J. dan Ghemawat, S. (2008) Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, **51**, 107–113.

[9] Sammer, E. (2012) *Hadoop Operations*, 1st edition. O'Reilly Media, Inc.

[10] Rasmussen, E. M. dan Willett, P. (1989) Efficiency of hierarchic agglomerative clustering using the icl distributed array processor. *J. Doc.*, **45**, 1–24.

[11] Samatova, N. F., Ostrouchov, G., Geist, A., dan Melechko, A. V. (2002) Rachet: An efficient cover-based merging of clustering hierarchies from distributed datasets. *Distributed and Parallel Databases*, **11**, 157–180.

[12] Johnson, E. L. dan Kargupta, H. (2000) Collective, Hierarchical Clustering from Distributed, Heterogeneous Data. Bagian dari Zaki, M. J. dan Ho, C.-T. (ed.), *Large-Scale Parallel Data Mining*. Springer Berlin Heidelberg, Berlin, Heidelberg.