The combined graph for efficiency at every set of work stations is drawn below:

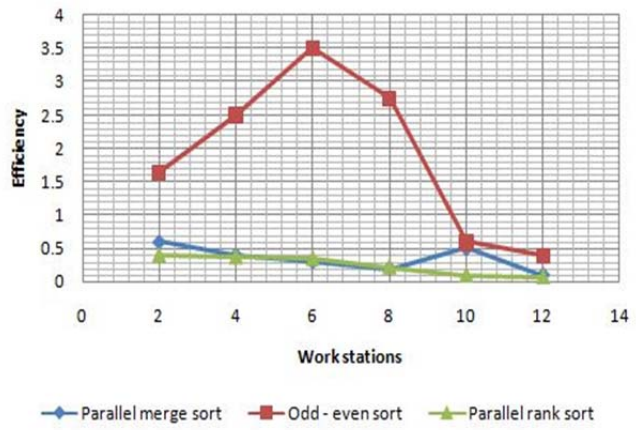

Fig 6: Combined graph for efficiency

V. CONCLUSIONS

1. If we talk about the total execution time, the least amount of time was taken by parallel merge sort whereas parallel rank sort took maximum time for the complete execution. This implies that parallel merge sort algorithm is the quickest of all for the given data set whereas the parallel rank sort algorithm is the slowest.
2. From (speed up) point of view, odd – even sort algorithm stands as the best algorithm followed by parallel merge and then parallel rank sort.
3. Talking about efficiency, again odd – even sort algorithm leads of all, worst stands the parallel rank sort algorithm.
4. Finally it can be concluded that, from the chosen three algorithms, parallel rank sort performs worst in each of the cases.
5. It is upon the user to go for parallel merge sort or odd – even sort depending upon his needs. If one wishes to have shortest execution time i.e. fastest processing algorithm, one should choose parallel merge sort whereas if someone believes in efficiency, one may choose odd – even sort algorithm.

REFERENCES:

[1] Kalim Qureshi and Haroon Rashid,"A Practical Performance Comparison of Parallel Matrix Multiplication Algorithms on Network of Workstations.", IEE Transaction Japan, Vol. 125, No. 3, 2005.

[2] Kalim Qureshi and Haroon Rashid," A Practical Performance Comparison of Two Parallel Fast Fourier Transform Algorithms on Cluster of PCS", IEE Transaction Japan, Vol. 124, No. 11, 2004.

[3] Kalim Qureshi and Masahiko Hatanaka, "A Practical Approach of Task Partitioning and Scheduling on Heterogeneous Parallel Distributed Image Computing System," Transaction of IEE Japan, Vol. 120-C, No. 1, Jan., 2000, pp. 151-157.

[4] K. Sado, Y. Igarashi, Some Parallel Sorts on a Mesh-Connected Processor Array and Their Time Efficiency, Journal of Parallel and Distributed Computing, 3, pp. 398-410, 1999.

[5] D. Bitton, D. DeWitt, D.K. Hsiao, J. Menon, A Taxonomy of Parallel Sorting, ACM Computing Surveys, 16,3,pp. 287-318, September 1984.

[6] Song, Y.D., Shirasi, B. A Parallel Exchange Sort Algorithm. *South Methodist University, IEEE 1989*.

[7] B.R. lyer, D.M. Dias, System Issues in Parallel Sorting for Database Systems, Proc. Int. Conference on Data Engineering, pp. 246-255, 2003.

[8] F. Meyer auf der Heide, A Wigderson, The Complexity of Parallel Sorting, SIAM Journal of Computing, 16, 1, pp. 100-107, February 1999.