Abstract

The complexity of handling the scalability problem of huge data can be reduced with parallel processing. The efficiency of parallel processing changes as the number of processors or number of threads change. Parallel processing is more appropriate for the field like data mining as it is the technique of analyzing large quantities of data to extract useful knowledge. Data mining is very much essential to the modern society as more and more data is being collected from various fields. Experiments are conducted to test the run time efficiency of the apriori algorithm on dual core processor by changing the number of threads for different databases at different support counts. This paper also present the comparison of real time, user time and system time with multiple threads on dual core compared to sequential implementation.
Performance Evaluation of Apriori on Dual Core with Multiple Threads

- Fayyad, Usama, Gregory Piatetsky-Shapiro, and Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine Volume 17 Number 3 (1996).
- Herb Sutter, "The Free Lunch Is Over A Fundamental Turn Toward Concurrency in Software," This article appeared in Dr. Dobb's Journal, 30(3), March 2005.
- Jiawei Han, Hong Cheng, Dong Xin, Xifeng Yan, "Frequent pattern mining: current status and future directions," In the Journal of Data Min Knowl Disc (2007) 15:55–86, Springer Science+ Business Media, LLC 2007.
- J Han, J Pei, Y Yin, "Mining frequent patterns without candidate generation," In: Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD), Dallas, TX, pp 1–12.
- Li Liu, Eric Li, Yimin Zhang, Zhizhong Tang, "Optimization of Frequent Itemset Mining on Multiple-Core Processor," VLDB &apos;07, September 23–28, 2007, Vienna, Austria.
Performance Evaluation of Apriori on Dual Core with Multiple Threads

- Kent Milfeld, "Introduction to Programming with OpenMP", September 12th 2011, TACC.

Index Terms

Computer Science
Artificial Intelligence

Keywords
apriori data mining efficiency multi core openMP parallel processing