TDT24: Mining Association Rules from Multi-stream Time Series Data on Multiprocessor Systems

Magne Tøndel, magne@toendel.com
Trondheim, 27.10.2009

Abstract

- Mining association rules from multi-stream data has received a lot of attention to the data mining community.
- It is quite effective and useful to discover such rules.

Abstract

- The challenge: Discovering these rules is a very time consuming and expensive when we are mining them from enormous time ordered real valued continuous data sets.

Abstract

- This strongly motivates the need of efficient parallel processing techniques and algorithms.
- In the presented paper, they use parallel processing to discover dependencies from a large amount of time series multi-stream data.
- They also present two parallel programming techniques (OpenMP and MPI) to implement this.

Introduction

- Discovery of dependencies in multi-stream time series data is an important problem with great significance.
- Example: The stock price.
  Rise and fall of the price on some stocks obviously cause the price of other stocks to rise and fall.
- These dependencies can help us to decide when to by stocks.
- And these dependencies can also be expressed as association rules.

Introduction

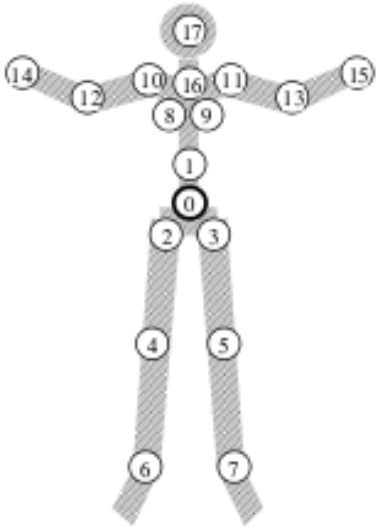
- The task of finding all association rules can require a lot of computational and memory resources, especially when the data sets are enormous and high dimensional.
- It is therefore crucial to leverage the aggregate computational power of multiple processors to find the association rules from the huge data sets.

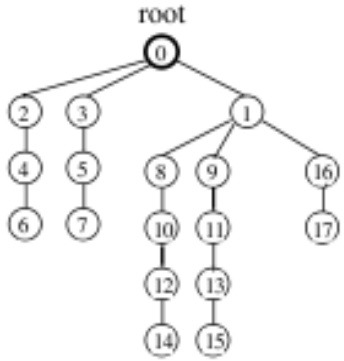
Introduction

- In the presented paper, they focus on human motion data.
- For example, find association rules discovered from motion data about «walking».
- Human motion data is three dimensional in nature, but they are converting the large amount of three dimensional motion data to different symbols in one dimension.

Discovery of Association Rules

- The human motion data captured by a motion capturing system consists of various types of information of the body parts.
- The body parts can also be represented as a tree structure.





0: lower torso (root)
1: upper torso
2: right hip
3: left hip
4: right knee
5: left knee
6: right ankle
7: left ankle
8: right collar bone
9: left collar bone
10: right shoulder
11: left shoulder
12: right elbow
13: left elbow
14: right wrist
15: left wrist
16: neck
17: head

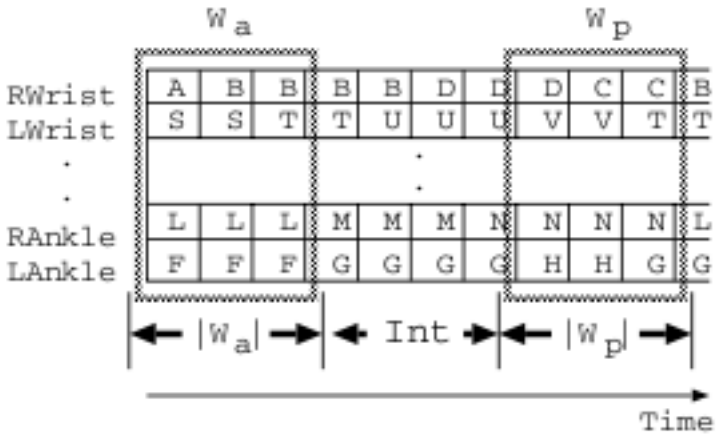
NTNU
Norwegian University of Science and Technology

Discovery of Association Rules

- The dependency on data of various operations in motion data depends on operations performed in the past (they call them «active operations»).
- «Active operations» has a relation to affect on generating operations («passive operations») in the future.
- Such a dependency is called an association rule.

Discovery of Association Rules

- In order to find «active» and «passive» operations, they use two windows Wa («active») and Wp («passive») with the fixed interval Int.
- The Int is the interval between the windows.



Discovery of Association Rules

- The strength of association rules in motion data is defined by using the probability of occurrence for two operations.
- The probability is calculated by the following function: $\delta = t(Pa \wedge Pp) / t(Pa)$
- Pa and Pp are the occurrences of the active and passive operations respectively.
- And $t(Pa \wedge Pp)$ represents the number of simultaneous occurrences of Pa and Pp.

Parallel Data Mining

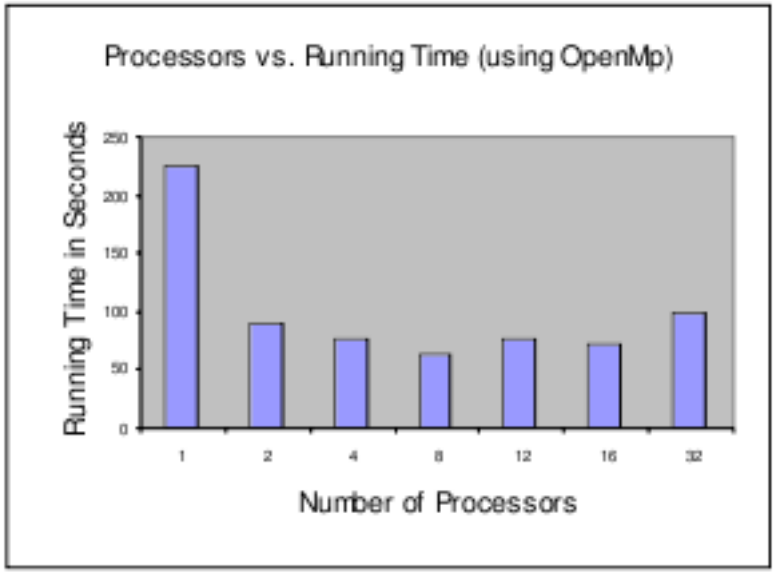
- The paper presents two pseudo-code examples of the algorithm (OpenMP and MPI).
- Both of them is based on the rule mining algorithm called *apriori*.

Experiments

- 64-node SGI Origin 3000 DSM-system
- Each node consists of 2 MIPS R10000 processors running at 500 MHz with a total of 8 processors in a node board.
- The database is stored on an attached 6 GB local disk.
- For the test data sets, they use 50 different kinds of performed motions such as walking, running, dancing etc.

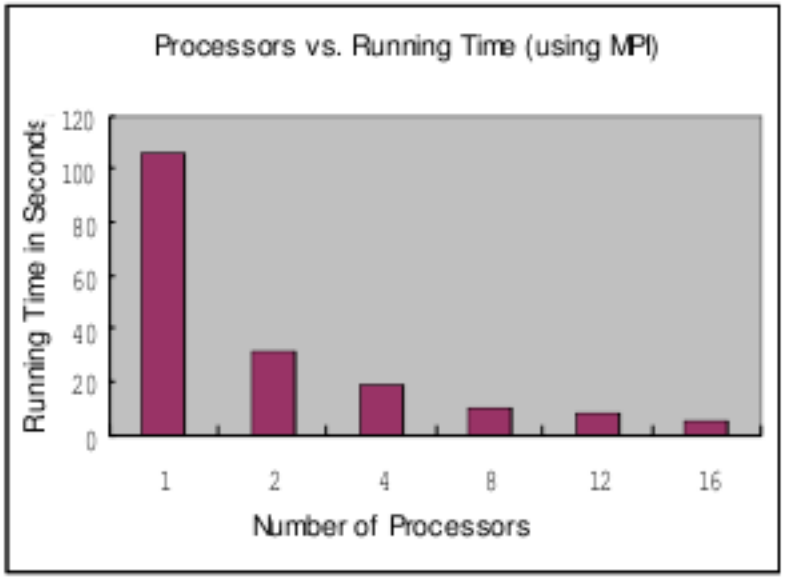
Experimental Results (OpenMP)

- The results from the OpenMP implementation are shown in the table.



Experimental Results (MPI)

- Because of the bad scalability, they made an implementation using MPI (Message Passing Interface).



NTNU
Norwegian University of Science and Technology

Conclusion

- The MPI implementation scaled better than the OpenMP implementation for this particular problem.