

# ***“Data Mining on an OLTP System (Nearly) for Free”***

## **SUMMARY**

### **INTRODUCTION**

Any commercial data mining operation requires great amounts of free memory, processor cycles, interconnect bandwidth, and disk bandwidth. These resources are seldom available on a typical online transaction processing system. Therefore, most organizations wishing to find interesting patterns in their data bases are forced to buy a dedicated data mining system. This means enormous initial investment in order to start a mining project. Such computers usually sport numerous CPU's, gigabytes of memory, huge amounts of disk storage, and often cost millions of dollars. The costs are further increased by the fact that the entire data base must be copied over to the new system. The organization needs to maintain two sets of data and keep the data mining set updated.

The modern online transaction processing systems use Active Disks as the storage devices. These are intelligent hard drives which have built in RISC core processors and random access memory. The devices have the ability to run application level programs directly on their own hardware. Studies have shown that most modern systems contain more processing power distributed among the active hard drives than in the central server itself. The paper by Riedel, Christos, Ganger, and Nagle proposes a scheme to utilize the active disks to perform the majority of the data mining workload on the online transaction processing system. The authors have discovered that on third of the entire disk bandwidth can be dedicated to data mining with a minimum affect on the foreground process.

## **PREVIOUS WORK**

Previous papers propose various ways of running data mining algorithms directly on the active hard drives. The researchers have designed methods for doing nearest neighbor search, association rules, and clustering away from the data mining server. Studies have shown that the disks provide sufficient processing power and memory to perform these tasks. It has also been shown that distributing the data mining workload over a number of active disks reduces the use of interconnect bandwidth.

Most researchers propose the use of two types of workloads on a single system: the foreground online transaction processing workload, and the background data mining workload. Memory allocation schemes, that use this foreground-background model, have been designed to manage memory accordingly to process priorities. Many of the previous studies have also concluded that disk bandwidth remained the critical resource and a major factor in distributed computation.

## **METHODOLOGY**

The goal of the paper is to propose a way of utilizing the existing disk bandwidth on a typical online transaction processing system to perform a background data mining process. The authors demonstrate that enough disk bandwidth exists and conclude that one third of this resource can be dedicated to the background process without affecting the foreground task. Using the proposed algorithms allows the online transaction processing and the data mining to be performed on the same computer system.

The necessary bandwidth can be obtained by utilizing the time taken by the hard drive to seek a new block and rotate its disk. . When the disk services OLTP read

requests, it looks in the block queue, reads the first block, and then searches for the next block to be read in. Due to the rotational latency, the disk is idle during the trip from the first block to the second one. The authors propose a way to utilize this idle time for servicing the data mining application.

According to the proposed system, the disk maintains two block request queues: one for the foreground OLTP process and one for the data mining application that runs in the background. While moving between blocks, the drive head passes over other blocks on the disk. Those blocks can be silently read in without affecting the efficiency and throughput of the foreground requests. The drive checks the background request queue every time it passes over blocks during disk rotation. If the blocks have been requested by the data mining process, they are read in.

The authors' biggest assumption is that the results of the data mining process stay independent of the order of data that the disks read in. The proposed scheduling system is opportunistic in respect to servicing the background process. It cannot guarantee that the block requests will be filled in order received. The entire idea lies in the fact that the data does not have to be sent to the server. Because of active disk architecture, the information can be processed by software running on the disk. Full table scans, aggregations, and association rule discovery can be performed without the need of sending the data to the server. Only when the data is processed, it is sent to the server where the multiple parts are aggregated into a whole.

## **RESULTS**

The experiments to test the proposed scheduling algorithms were performed on a disk simulation software. The transaction workload was taken from a typical OLTP server. In all experiments, the multiprogramming level, specified in disk requests, was varied in order to simulate increasing foreground load on the system. All simulations ran for one hour and three approaches were tested. The first method relies on servicing background block requests only when there are no foreground requests. The second method fills the data mining queue only when seeking the blocks for the foreground queue. Finally, the third approach combines the first two together.

### **ALGORITHM I**

The experiments with the first algorithm show that the addition of the data mining workload has very small impact on the OLTP. The background block requests are assigned a lower priority than the OLTP requests. They are only performed when the foreground request queue is empty. As the foreground workload increases (more disk requests by the OLTP) the data mining service is cut back.

The results show that the OLTP throughput initially increases as there are more OLTP requests scheduled. This increase gradually levels off when the disk queues become saturated with the foreground requests. On the other hand, the data mining throughput is at its peak when there are no OLTP requests. Its decrease is proportional to the increase in the number of OLTP requests.

The online transaction processing response time is slightly lower than normal at low levels of OLTP multiprogramming. That is, when the OLTP process requests few blocks, the data mining requests are being fulfilled. When a new foreground request comes in, there is a high probability that a background request is being serviced at the time. The foreground request must thus wait a while to be processed. This is the cause of the slight increase in OLTP response time at low levels of multiprogramming. Furthermore, as the OLTP increases its number of requests the response time becomes proportional to the number of requests. The background requests are then no longer interfering with the foreground ones.

#### ALGORITHM II

The results of simulating the second algorithm show that the data mining throughput increases as the number of OLTP requests increase. When OLTP request are low then the mining throughput is also low. This is due to the fact that the mining requests are serviced only when the drive seeks for OLTP blocks. As the foreground process requests more blocks the disks get more opportunities to service the background blocks. When the OLTP does not provide any requests the drives sit idle and not data mining request are serviced.

The experiment shows no increase in the OLTP response time as was the case in the first experiment. Because no background blocks are read in during the idle time, the OLTP requests can be serviced as soon as they come in.

Furthermore, its been shown that the rate of reading background blocks drops as the overall pool of unread data mining blocks decreases. The rate also decreases when

the remaining blocks reside on areas of the disk not accessed by the OLTP. The authors suggest that appropriate positioning of data on the disk can increase the rate. Also, the system can be optimized by reading the remaining data via high priority requests. One would need to find an appropriate threshold which provided the most optimal tradeoff between foreground and background performance.

### ALGORITHM III

The third algorithm integrates the previous two scheduling methods. The mining requests are read when the disks are idle and also during the disk seek time. At low OLTP request levels, the first algorithm is active. When the foreground queue is empty, the disk performs background process services. When the OLTP requests increase, the first algorithm fades out and the second one is automatically triggered. As the drive begins to process new OLTP block requests, it opportunistically reads in the blocks requested by the data mining task. The resulting graphs include the 'best parts' of curves from the previous two experiments. The maximum mining throughput reached was 2.0 MB/s which is about one third of the raw disk bandwidth.

Experiments simulating a single and multiple disk systems were performed on the third algorithm. Most commercial OLTP systems actually have more than one disk and the databases are distributed. The authors have shown that the mining throughput increases with the distribution of workload across multiple active disks. A system with N number of disks offers N times the performance of the same server with only one disk.

## REAL WORKLOAD COMPARISON

The results obtained from the disk simulator were later compared to the results of experiments ran on a real computer system. A 300Mhz Pentium II computer, with 128 MB RAM, and two active disk drives was used. The system ran Windows NT and Microsoft SQL server on a one gigabyte test data base. The two drives were Quantum Viking disks, each 2.2 Gigabytes, with 7,200 RPM speed, and the average seek time of 8ms. The results obtained on the real system were similar to those obtained from the simulated one. The mining throughput was slightly lower on the real system because the data base was not evenly spread across the disk as was assumed in the simulation.

## **CONCLUSION**

Replacing 'dumb' disks with Active Disk devices provides a necessary disk bandwidth to perform OLTP and data mining on a single system. Utilizing the proposed scheduling algorithms harnesses the available disk bandwidth so it becomes no longer necessary to buy a dedicated data mining system. The paper has shown that, on the average, a 2 Gigabyte disk can be read in its entirety in under 28 minutes. That means over 50 whole disk scans during one day.