Faculty of COMPUTER SCIENCE

UNB

Notice of MCS Oral Defence

Thursday, November 14th, 2013 - 1:30pm
Oland Hall, Room OH120 (Video Conference Room)

Performance Evaluation of Fast Integer Compression Techniques Over Tables

By

Ikhtear Md. Sharif Bhuyan

Examining Committee

Dr. Hazel Webb, Dr. Daniel Lemire and Dr. Owen Kaser (CS), Supervisors
Dr. Christopher Baker (CS), Chair
Dr. Weichang Du (CS), Internal Reader
Dr. Tim Alderson (MATHSCI), External Reader

ABSTRACT

Compression is used in database management systems to improve the performance by preserving memory and storage. Compression and decompression requires substantial processing by the Central Processing Unit (CPU). Queries in large databases such as Online Analytical Processing (OLAP) databases involve processing huge amounts of data. Database compression not only reduces disk space requirements, but also increases the effective Input/Output (I/O) bandwidth since more data is transferred in compressed form to cache for query processing. As a result, database compression transforms I/O-intensive database operations into more CPU-intensive operations. We are most likely to benefit from compression if encoding and decoding speeds significantly exceed I/O bandwidth. Hence, we seek compression schemes that can be used in databases with good compression ratios and high speed. We examined the performance of several compression schemes such as Variable-Byte, Binary Packing/Frame Of Reference (FOR), Simple9 and Simple16 which have reasonable compression ratio with fair decompression speed over sequences of integers. As variations on Binary Packing, we also studied patched schemes such as NewPFD, OptPFD and FastPFOR: they have good compression ratios and decompression speed though they need more computational time during compression than Binary Packing. In our study, we aim to quantify the trade-offs of fast integer compression schemes with respect to compression ratio and speed of compression and decompression. We are able to decompress data at a rate of around 1.5 billion of integers per second (in Java) while sometimes beating Shannon's entropy. In our tests, Binary Packing is significantly faster than all other alternatives. Among the patched schemes we tested, the recently introduced FastPFOR is most competitive. Hence, we found that it is worth using a patching scheme because we get both a good compression ratio and a high decompression speed. However, the higher compression and decompression speed of Binary Packing makes it a better choice when speed is more important than compression ratio. We also assessed the effects that row ordering and sorting have on compression performance. We discovered that sorting can significantly improve the performance of compression. We obtained around 15% gain in decompression speed and 12% gain in compression ratio by sorting compared to random shuffling. Additionally, we found that sorting on the highest cardinality column was more effective than sorting on lower cardinality columns.

GRADUATE STUDENTS ARE ENCOURAGED TO ATTEND