A GPU-aware Parallel Index for Processing High-dimensional Big Data

Abstract—The problem of the curse of dimensionality for processing large high-dimensional datasets has been an open challenge. Numerous research efforts have been proposed for improving query performance in high-dimensional space through hierarchical indexing using the R-tree or its variants and exploring parallel processing of the R-tree on GPUs. Despite these existing efforts, the curse of dimensionality remains to be a grand challenge since the existing methods deteriorate drastically as the dimensionality of datasets increases. To cope with this problem, we present a novel GPU-aware parallel indexing method called G-tree, which offers consistent and stable performance in high-dimensional space. The rationale of the G-tree is to combine the efficiency of the R-tree in low-dimensional space with the massive parallel processing potential of GPUs by introducing a new data structure and three new optimization techniques to better utilize the GPU memory structure for accelerating both index search and index node access on GPUs. The first two optimizations promote effective parallelism utilization in GPU memory access. We dedicate the third optimization to further speed up the G-tree index by conducting progressive filtering using our dimension filters. We evaluate the validity of the G-tree approach by extensive experiments on high-dimensional datasets, showing that the G-tree outperforms the existing state-of-the-art techniques.

CONCLUSIONS
We have presented a GPU-aware parallel index scheme, called G-tree, for large-scale high-dimensional query processing. The rationale of the G-tree design is to combine the efficiency of the R-tree in lower-dimensional space with the parallel computing capability of GPUs in higher dimensionality. We employ three design strategies. First, we introduce a new data structure (a structure of arrays) to better utilize the GPU memory structure, accelerating both index search and index node access on GPUs byromoting effective parallelism utilization in GPU memory
access. Second, unlike previous approaches, the G-tree by design introduces a BFS base lookup without a queue or a stack, enabling the G-tree to be more efficient at handling datasets of high dimensionality.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb

**SOFTWARE REQUIREMENTS:**

- Operating system : Windows 7/UBUNTU.
- Coding Language : Java 1.7 , Hadoop 0.8.1
- IDE : Eclipse
- Database : MYSQL

**REFERENCES**
