Virtual Shuffling for Efficient Data Movement in MapReduce

ABSTRACT:

MapReduce is a popular parallel processing framework for large-scale data analytics. To keep up with the increasing volume of datasets, it requires efficient I/O capability from the underlying computer systems to process and analyze data in two phases (mapping and reducing). Between these phases, MapReduce requires a shuffling phase to globally exchange the intermediate data generated by the mapping phase. We reveal that data shuffling, by physically moving segments of intermediate data across disks, causes significant I/O contention and compounds the I/O problem. In this paper, we propose a novel virtual shuffling strategy to enable efficient data movement and reduce I/O for MapReduce shuffling, thereby reducing power consumption and conserving energy. Virtual shuffling is realized through a combination of three techniques including a three-level segment table, near-demand merging, and dynamic and balanced merging subtrees. Our experimental results show that virtual shuffling significantly speeds up data movement in MapReduce and achieves faster job execution. Particularly, its reduction in disk I/O accesses results in as much as 12% savings in power consumption for MapReduce programs.

INTRODUCTION

MAPREDUCE has emerged as a popular and easy-to-use programming model for large-scale data analytics in data centers. It is an important application for numerous organizations to process explosive amounts of data, perform massive computation, and extract critical knowledge out of big data for business intelligence. The efficiency of MapReduce performance and scalability can directly affect our society’s ability to mine knowledge out of raw data. In addition, energy consumption accounts for a large portion of the operating cost of data centers in analyzing such big data. While business and scientific applications are increasingly relying on the MapReduce model, the energy efficiency of MapReduce is also critical for data centers’ energy conservation.
EXISTING SYSTEM

In Existing System mapping and reducing phases, a ReduceTask needs to fetch a segment of the intermediate output from all finished MapTasks. Globally, this leads to a shuffling of intermediate data (in segments) from MapTasks to Reduce Tasks. For data-intensive MapReduce programs such as TeraSort, data shuffling can add a significant number of disk accesses, contending for the limited I/O bandwidth.

DisADVANTAGE OF Existing SYSTEM

✓ Serious bottleneck along with the severe disk I/O contention in data-intensive MapReduce programs, which entails further research on efficient data shuffling techniques.

PROPOSED SYSTEM

In Proposed System virtual shuffling as a new strategy to enable efficient data movement for MapReduce applications. Accordingly, we have designed and implemented virtual shuffling as a combination of three techniques including a three-level segment table, near-demand merging, and dynamic and balanced merging subtrees.

ADVANTAGE OF PROPOSED SYSTEM

Virtual shuffling significantly relieves the disk I/O contention problem and speeds up data movement in MapReduce programs. In addition, it reduces power consumption of MapReduce programs by as much as 12%.
**HARDWARE REQUIREMENTS:**

- System: Pentium IV 2.4 GHz.
- Hard Disk: 40 GB.
- Floppy Drive: 44 Mb.
- Monitor: 15 VGA Colour.

**SOFTWARE REQUIREMENTS:**

- Coding Language: Java 1.7, Hadoop 0.8.1
- Database: MySql 5
- IDE: Eclipse