PRISM: Fine-Grained Resource-Aware Scheduling for MapReduce

ABSTRACT:

MapReduce has become a popular model for data-intensive computation in recent years. By breaking down each job into small map and reduce tasks and executing them in parallel across a large number of machines, MapReduce can significantly reduce the running time of data-intensive jobs. However, despite recent efforts toward designing resource-efficient MapReduce schedulers, existing solutions that focus on scheduling at the task-level still offer sub-optimal job performance. This is because tasks can have highly varying resource requirements during their lifetime, which makes it difficult for task-level schedulers to effectively utilize available resources to reduce job execution time. To address this limitation, we introduce PRISM, a fine-grained resource-aware MapReduce scheduler that divides tasks into phases, where each phase has a constant resource usage profile, and performs scheduling at the phase level. We first demonstrate the importance of phase-level scheduling by showing the resource usage variability within the lifetime of a task using a wide-range of MapReduce jobs. We then present a phase-level scheduling algorithm that improves execution parallelism and resource utilization without introducing stragglers. In a 10-node Hadoop cluster running standard benchmarks, PRISM offers high resource utilization and provides 1:3 improvement in job running time compared to the current Hadoop schedulers.}

INTRODUCTION

BUSINESSES today are increasingly reliant on large-scale data analytics to make critical day-to-day business decisions. This shift towards data-driven decision making has fueled the development of MapReduce, a parallel programming model that has become synonymous with large-scale, data-intensive computation. In MapReduce, a job is a collection of Map and Reduce tasks that can be scheduled concurrently on multiple machines, resulting in significant reduction in job running time. Many large companies, such as Google, Facebook, and Yahoo!, routinely use MapReduce to process large volumes of data on a daily basis. Consequently, the performance and efficiency of MapReduce frameworks have become critical to the success of today’s Internet companies.
EXISTING SYSTEM

Existing System scheduling problem becomes significantly easier to solve if we can assume that all map tasks (and similarly, all reduce tasks) have homogenous resource requirements in terms of CPU, memory, disk and network bandwidth. Indeed, current MapReduce systems, such as Hadoop Map-Reduce Version 1:x, make this assumption to simplify the scheduling problem. These systems use a simple slot-based resource allocation scheme, where physical resources on each machine are captured by the number of identical slots that can be assigned to tasks.

DisADVANTAGE OF Existing  SYSTEM

Run-time resource consumption varies from task to task and from job to job. Several recent studies have reported that production workloads often have diverse utilization profiles and performance requirements. Failing to consider these job usage characteristics can potentially lead to inefficient job schedules with low resource utilization and long job execution time.

PROPOSED SYSTEM

In Proposed System to address this limitation, we introduce PRISM, a fine-grained resource-aware scheduler that coordinates task execution at the level of phases. We first demonstrate how task run-time usage can vary significantly over time for a variety of MapReduce jobs. We then present a phase-level job scheduling algorithm that improves job execution without introducing stragglers.
ADVANTAGE OF PROPOSED SYSTEM

PRISM offers high resource utilization and provides 1:3 improvement in job running time compared to the current Hadoop schedulers.

ARCHITECTURE:
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