MISSISSIPPI STATE UNIVERSITY

HIGH PERFORMANCE DATA ANALYTICS

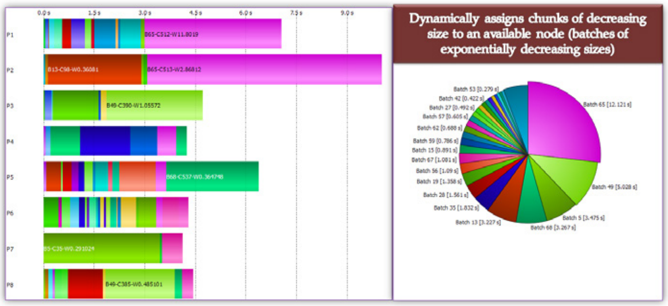
Graph Walk Optimization

CHALLENGE

Data collection and analysis is rapidly changing the way scientific, national security, and business communities operate. Data analytics applications, specifically those involving graph analytics, have received increasing attention over the last several years. The performance of these applications is essential, even sometimes critical, to achieve the objectives proposed by the domain areas making use of them.

The complexity of working with applications involving big data requires computing in parallel and distributed environments. Scheduling big data computations on parallel non-dedicated heterogeneous systems, where the computing resources may differ in availability and reliability, is a challenging task requiring resilient scheduling methods for an efficient execution.

> To employ scheduling algorithms along with fault tolerance mechanisms for achieving high performance on parallel graph walks, while addressing both data scaling and resilience during the analytic search queries over large graphs.



Dynamic scheduling enables load balancing of parallel tasks.

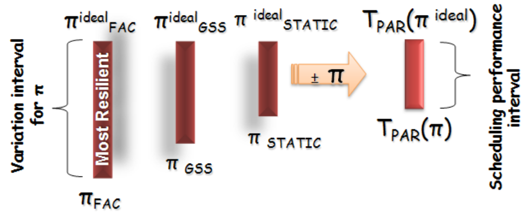
CURRENT PRACTICE

Many research efforts have been attempted to optimize performance. These optimizations include improving performance (per core), increasing scalability of their execution in parallel and distributed environments, and dealing with dynamically changing large data sets. Moreover, the sharp increase in number of computing units is likely to continue, which translates in expected growth in the failure rate and corresponding decrease in the mean time to interruption (MTTI) of the computing system. Therefore, frequently occurring resource failures will drastically affect the execution of big data applications running on high performance computing systems. In order to hide the occurrence of faults, or the sudden unavailability of resources, fault-tolerance mechanisms (e.g., replication or check-pointing-and restart) are usually employed. These optimizations are especially important with applications involving big data.

TECHNICAL APPROACH

In this work, we propose to enhance the flexibility and resilience of GEMS's current implementation for performing parallel graph walks (conceptualized as nested loops), which only support basic static heuristics to schedule an exponential number of parallel tasks (result of executing loop structures) in heterogeneous computing environments. First, this approach will be achieved by incorporating a node level scheduling functionality inside the GMT layer of GEMS library. This will assist with fine grain parallelism of computational tasks by dynamically scheduling of parallel tasks with diverse computational needs spawned by the graph walks across the cores (workers) inside the nodes, in order to avoid performance degradation due to load imbalance (where query data is highly skewed) at cluster level. Secondly, we plan to add provision of fault tolerance to the GMT layer so that the data pertaining to a failed node is not lost, and the parallel tasks scheduled on a failed node can be rescheduled, thus adding resilience to GEMS stack in the presence of resource failures.

The ongoing research work will address significant challenges faced by algorithms used to parallelize graph walks in order to achieve robust performance. The scheduling algorithms will take into account various properties of graph problems, as well as those of the system on which the query is being executed, such as: the unstructured and highly irregular nature of data, node failures, and failure detection and data recovery.



Resilient scheduling (that can handle the largest variations of π (perturbations)) results in having lowest impact on the performance of the parallel graph walk.

IMPACT

By analyzing and developing functionality for reliability of graph walk schedules in GEMS library, an increasing number and more complex big data analytic queries can be answered in the required set delivery times. Furthermore, the results of the analysis can be stored and learned, to enable dynamic selection of the most resilient scheduling algorithm from a portfolio of scheduling algorithms for a given execution scenario.

Contacts

Ioana Banicescu
Professor, Dept. of Computer Science & Engineering
662.325.7508
ioana@cse.msstate.edu

Nitin Sukhija
Assistant Research Professor
662.325.5952
nitin@dasi.msstate.edu

Funding provided by a grant from Pacific Northwest National Laboratory.