Methods and Algorithms for Adapting Machine Learning and Numerical Computations to High-Performance Computing Systems

Summary

This dissertation concerns the search for methods to improve the computational and energy efficiency of machine learning and numerical computations, achieved in the environment of high-performance computing (HPC) systems. The following original contributions have been achieved:

1. a method was developed to reduce energy consumption in applications parallelized under the data parallelism model. The primary goal of this method is to mitigate the effects of load imbalance among threads or processor cores, which typically arises from applying widely used data-parallel approaches. The method is based on managing the frequency of carefully selected groups of cores, enabling a better alignment of processor operating parameters with actual workloads. This approach effectively addresses load imbalance, achieving significant energy savings without performance loss. The proposed solution outperforms commonly used techniques such as Dynamic Voltage and Frequency Scaling (DVFS), offering better adaptation to application-specific characteristics.
2. Algorithms for selecting heterogeneous frequency configurations were proposed, providing a practical implementation of the developed energy reduction method for data-parallel applications. These algorithms significantly reduce the number of required test configurations, accelerating the search for the optimal frequency setup. The first algorithm makes decisions based on energy consumption measurements, while the second eliminates the need for such measurements, relying solely on execution time analysis. This enables practical deployment even in environments where energy measurement is difficult or even impossible.
3. Two methods were designed and implemented to improve the performance of counting queries execution strategies based on representing query variable

states with bitmaps and using a Depth-First Search (DFS) traversal algorithm. The first method focuses on reducing the number of bitmap intersections (AND operations), while the second minimizes the number of visited tree nodes. Both methods were implemented in the SABNAtk library, a tool for efficient execution of counting queries.

4. a method for automatic selection of counting queries execution strategies was developed, with the goal of reducing query stream processing time. The approach combines the strengths of three complementary strategies — BV, RAD, and CT — and uses online regression as a decision-making mechanism to select, for each query in the stream, the strategy with the lowest estimated execution time. The original contributions include:
    a) Formulation of theoretical cost models for counting query execution under the three strategies, derived from in-depth analysis of their operational principles. These models estimate query execution cost based on query properties and dataset size.
    b) Development of an efficient algorithm for incrementally updating the cost models, enabling effective model refinement using new observations without storing past data and increasing computational overhead.
    c) Design of a query stream processing algorithm that defines the complete workflow of the method — from model initialization and training, through adaptive strategy selection, to handling memory-bound queries — ensuring consistency and practicality of the approach.
    d) Implementation of the method in the official SABNAtk library, enabling its use in real-world scenarios and providing access for a broader community of users.
5. a fully thread-safe implementation of the automatic counting queries execution strategy selection method was designed and implemented. The solution minimizes the number of write operations and employs mutex mechanism to eliminate race conditions during parallel processing. This implementation enables task parallelism, where counting queries are treated as independent tasks executed concurrently across multiple processor cores using any framework (e.g., OpenMP, Intel TBB).

6. a method for parallel execution of counting queries tailored to new ccNUMA (cache-coherent Non-Uniform Memory Access) architectures was proposed. Such systems consist of multiple NUMA domains and exhibit heterogeneous memory access latencies depending on the physical location of data relative to the executing core. The proposed method is independent of the specific counting queries execution strategy and can be applied universally, specifically in combination with the automatic strategy selection method.