DI

TOPOLOGY-AWARE DISTRIBUTED GRAPH PROCESSING FOR TIGHTLY COUPLED CLUSTERS

Allocation: Innovation and Exploration/10 Knh
PI: Indranil Gupta[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

Cloud applications have burgeoned over the last few years, but they are typically written for loosely coupled clusters such as datacenters. In this research, we investigated how one can run cloud applications in tightly coupled clusters and network topologies on supercomputers. Specifically, we looked at a class of distributed machine learning systems called distributed graph processing systems, and ran them on NCSA's Blue Waters. The key to achieving performance in distributed graph processing systems is to partition the graph intelligently across the nodes of the cluster. Through this work, we have presented new partitioning techniques that are topology-aware. Our strategies partition the graph in a manner that reduces both runtime and network traffic in the supercomputer topology. Compared to previously existing work, our new Restricted Oblivious and Grid Centroid partitioning approaches produce 25%–33% improvement in runtime, along with a sizable reduction in network traffic. To help operators select the best graph partitioning technique, we organized our experimental results into a decision tree.

RESEARCH CHALLENGE

Recent years have seen a massive proliferation in the amount of graph data available, from social networks such as Twitter and Facebook, peer-to-peer networks such as Gnutella, Web graphs, and autonomous systems. These data troves provide an opportunity to glean useful insights about the nature of such interconnected systems. However, these graphs are enormous—the Facebook graph alone has over a billion vertices and a trillion edges, and the human brain network has many billions of vertices. The sheer size of the graphs, therefore, requires the use of distributed graph processing frameworks such as Pregel and PowerGraph to analyze them. An important step in the analysis is partitioning the graph across the various workers in the cluster. Our work studies how we can partition these graphs in a network topology-aware way so as to avoid sending data over long routes.

METHODS & CODES

Our first partitioning strategy (Grid Centroid) determines the nodes that should perform state computations for each vertex to minimize network communication cost with other mirrors. Our second partitioning strategy (Restricted Oblivious) uses a greedy heuristic while creating vertex mirrors to ensure they are all placed close to each other and to the node performing the computation.

RESULTS & IMPACT

We developed two partitioning strategies that consider the topology of the underlying network while partitioning an input graph across worker nodes. We also analyzed the performance of our partitioning strategies and developed a decision tree to help operators determine the appropriate strategy to use based on parameters such as the cluster size and the nature of the input graph. In certain cases, our strategies were able to reduce the graph processing time by up to 33% by reducing average vertex master-mirror distance and the mean number of hops travelled by network data. Another observation from our experiments was that certain algorithms such as PageRank send comparatively less data over the interconnect and instead benefit from frequently flushing the network buffers.

These results demonstrate how tightly coupled clusters can be more efficiently used for distributed graph processing.

WHY BLUE WATERS

Blue Waters' 3D Torus interconnect provides a topology with heterogeneous communication costs. It was thus an important part of our experiments.

DI

GA

ML

DEEP LEARNING APPLICATIONS IN ENGINEERING AND QUANTITATIVE FINANCE

Allocation: Illinois/115.25 Knh
PI: Justin Sirignano[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

Our research team used the Blue Waters supercomputer to support research into applications of deep learning in engineering and quantitative finance. The first application involved solving high-dimensional partial differential equations using a deep learning algorithm. The second application modeled high-frequency financial data using deep learning.

RESEARCH CHALLENGE

Deep learning has revolutionized fields such as image, text, and speech recognition. Due to this success, there is growing interest in applying deep learning to other fields in science, engineering, medicine, and finance. We used Blue Waters to develop deep learning methods and models for important applications in engineering and quantitative finance. We developed a deep learning method for solving high-dimensional partial differential equations, which have been a longstanding computational challenge. In another project, we developed a deep learning model for high-frequency financial data.

METHODS & CODES

Solving partial differential equations with deep learning. High-dimensional partial differential equations (PDEs) are important in engineering, physics, and quantitative finance. However, they are computationally challenging to solve. Finite difference methods are infeasible in higher dimensions. Therefore, we developed a deep learning algorithm to solve PDEs and implemented it on the Blue Waters supercomputer. The algorithm is mesh-free, which is crucial for solving high-dimensional PDEs. The deep learning algorithm was able to accurately solve several high-dimensional PDEs.

We tested this deep learning algorithm on several challenging PDEs. First, we solved a high-dimensional free- boundary PDE for American options (a financial derivative on a portfolio of stocks) in up to 200 dimensions. We also tested the algorithm for solving a high-dimensional Hamilton–Jacobi–Bellman PDE. Finally, we solved Burgers' equation using the deep learning algorithm, which approximated the general solution to Burgers' equation across a range of different boundary conditions and physical conditions.

Modeling high-frequency data with deep learning. We analyzed a large high-frequency dataset of electronic market quotes and transactions for U.S. equities. Our approach used a large-scale deep learning approach and revealed several interesting insights into the relationship between price formation and order flow, which is the submission and cancellation of buy and sell orders. We found that the deep learning model is relatively stable across time and is able to provide a "universal model" across a range of different stocks. This implies that the relationship between price formation and order flow is far more stationary and universal across different stocks than previously believed.

RESULTS & IMPACT

Due to the success of deep learning in traditional computer science fields (e.g., image recognition), there is now significant interest in applying deep learning techniques in engineering, science, medicine, and finance. Our projects, supported by Blue Waters, are some of the first attempts at developing deep learning methods for engineering and finance applications.

WHY BLUE WATERS

Deep learning uses multilayer neural networks (i.e., deep neural networks) to build statistical models of data. This training of the deep learning model can be computationally intensive due to both the large number of parameters and the large amounts of data. Graphics processing units (GPUs) can be used to accelerate training of deep learning models. We leveraged Blue Waters' large amount of GPU resources to develop deep learning models for applications in engineering and finance. Blue Waters' technical staff provided invaluable help throughout the project, including solving a number of technical issues related to deep learning computational frameworks such as PyTorch and TensorFlow.

PUBLICATIONS & DATA SETS

Sirignano, J., and K. Spiliopoulos, DGM: A deep learning algorithm for solving high-dimensional partial differential equations. arXiv: 1708.07469 (2017).

Sirignano, J., and R. Cont, Universal features of price formation in financial markets: perspectives from Deep Learning. arXiv: 1803.06917 (2018).