ИСПОЛЬЗОВАНИЕ APACHESPARK ИOPENCL ДЛЯ РАСПРЕДЕЛЁННЫХ КЛАСТЕРНЫХ ВЫЧИСЛЕНИЙ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Жабинский С. В.

Татур М. М. - д-р техн. наук, профессор

В условиях увеличивающегося объёма данных, накапливающегося в современных хранилищах, увеличивается и потребность в их эффективной обработке. В настоящее время для обработки больших объёмов данных как правило используют технологии из инфраструктуры Hadoop: HDFS, Hive, Cassandra, MapReduce, Spark и другие. В настоящей работе приводятся результаты исследования совместного использования Apache Spark и платформы для параллельных вычислений OpenCL. Данная связка позволяет более эффективно использовать ресурсы каждого вычислительного узла.

Как правило, вычислительный кластер состоит из управляющего узла, одного или нескольких вычислительных узлов, и распределённого хранилища данных. При этом узлы, на которых расположены реплики данных, чаще всего являются и вычислительными узлами. «Бутылочным горлышком» в большинстве систем обработки данных являются задержки, связанные с чтением и передачей данных. Технологии из семейства Hadoop позволяют снизить объём передаваемых между узлами данных за счёт передачи обработчика к данным, а не наоборот. Таким образом достигается не только высокая производительность, но и масштабируемость, так как при необходимости в обработке большего объёма данных достаточно добавить дополнительные вычислительные узлы.

Несмотря на эффективность таких систем, их производительность может быть увеличена за счёт ускорения вычислений на каждом из узлов. В последние годы большое развитие получили технологии параллельной обработки информации на графических процессарах (GPU) [1, 2]. Данные процессоры, изначально ориентированные на обработку компьютерной графики, имеют SIMD архитектуру, большую часть операций в которой представляет собой выполнение одинаковой операции одновременно над разными данными. Системы с вычислениями на GPU имеют гетерогенную структуру. Центральный процессор отвечает за последовательность операций, загрузку и выгрузку данных с графического ускорителя. Основые вычисления происходят на GPU.

На рисунке 1 представлена схема гетерогенного кластера, на котором каждый вычислительный узел содержит CPU и GPU, использующийся в качестве ускорителя.

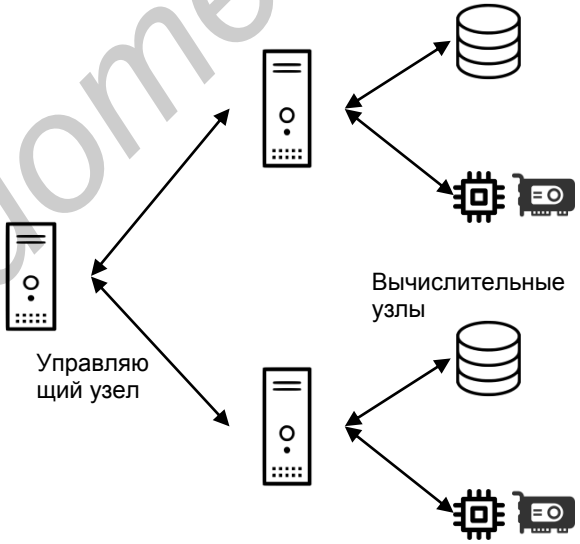

Рис. 1 — Схема гетерогенного кластера, состоящего из узлов с GPU в качестве ускорителя.

Программа запускается на управляющем узле. Она состоит из подпрограмм, которые передаются на вычислительные узлы. Каждый узел содержит свою порцию данных, которая, если того требует подпрограмма, может быть загружена в память GPU и обработана графическим процессором.

Описанная система была реализована на основе библиотеки SparkCL [3]. SparkCL позволяет использовать OpenCL в подпрограммах, выполняющихся на узлах кластера. Для этого разработчику требуется написать ядро (kernel), совместимое со стандартом OpenCL. Данные ядра могут выполняться на ускорителе, либо непосредственно на CPU. Выбор среды выполнения может быть сделан во время выполнения программы. Например, в зависимости от объёма обрабатываемых данных, типа операций или конфигурации узла, на котором выполняется подпрограмма, может быть принято решение о неэффективности использования графического ускорителя.

Одним из главных недостатков вычислений на GPU является то, что время на загрузку и выгрузку данных в/из памяти графического процессора может превысить время, сэкономленное за счёт параллельной обработки. Поэтому необходимо определить класс задач, для которых использование GPU имеет смысл.

Для тестирования данного подхода были проведены сравнительные эксперементы на разных типах задач. Сравнивалось время выполнения вычислений с использованием классического Spark и SparkCL. Результаты эксперементов показали ожидаемый прирост производительности в задачах, которые требуют большого объёма вычислений для каждого элемента данных, и в которых каждый элемент данных имеет большую размерность. К таким задачам можно отнести, например, обработку изображений.

Для расширения класса задач, для которых оправдано использование связки Spark и OpenCL необходим механизм, позволяющий сохранять данные в памяти GPU по завершении работы каждого ядра. Это позволит уменьшить издержки, связанные с обменом данных между центральным и графическим процессорами.

Исследованиеподдержанопроектом CERES. Centers of Excellence for young RESearchers (Reg.no. 544137-TEMPUS-1-2013-SK-JPHES),

Списокиспользованныхисточников:

1. Jian, L. Parallel data mining techniques on Graphics Processing Unit with Compute Unified Device Architecture (CUDA) / L. Jian, C. Wang, Y. Liu, S. Liang, W. Yi, Y. Shi // The Journal of Supercomputing. - 2013. - Vol. 64, iss.3 — p. 942-967.
2. Keckler, S. GPUs and the Future of Parallel Computing. / S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, D. Glasco // IEEE Micro. - 2011. - Vol. 31, No 5 - p. 7-17.
3. Segal, O. SparkCL: A Unified Programming Framework for Accelerators on Heterogeneous Clusters [Электронныйресурс] / O. Segal, P. Colangelo, N. Nasiri, Z. Qian, M. Margala. - Режимдоступа: https://arxiv.org/ftp/arxiv/papers/1505/1505.01120.pdf - Датадоступа: 28.03.2017.

ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ПРИ ДИАГНОСТИКЕ ЗАБОЛЕВАНИЙ НА ОСНОВЕ ДАННЫХ ЭКГ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Голиков А.В. – аспирант кафедры ЭВМ

Фролов И.И. – к.т.н.доцент

Всем известно, что диагнозы сердечно-сосудистых заболеваний можно ставить по данным электрокардиограмм. Однако, профессором, доктором медицинских наук В.М.Успенским предложен новый метод диагностики, который позволяет диагностировать широкий спектр заболеваний внутренних органов по ЭКГ. Выявлено, что различные расстройства вносят уникальные характеристики в ЭКГ-сигнал, а значит знаки приращений интервалов и амплитуд последовательных кардиоциклов можно использовать для диагностики информации о состоянии здоровья человека и наличии расстройств. Многие заболевания сказываются на работе сердца задолго до проявления клинических симптомов, что позволяет использовать ЭКГ для ранней диагностики.

Суть метода заключается в использовании свойства вариабельности сердечного ритма (интервалов кардиоциклов), которое позволяет определить текущее состояние здоровья человека. Для диагностики важны знаки приращений интервалов и амплитуд последовательных кардиоциклов. Для обеспечения высокого качества диагностики требуется испозовать электрокардиографию с разрешением более 500 Гц.

Технологию информационного анализа данных по ЭКГ можно разбить на 2 этапа: этап предварительной обработки ЭКГ-сигнала, и этап машинного обучения.

На этапе предварительной обработки осуществляется:

1) демодуляция сигнала — вычисление амплитуд, интервалов и углов по кардиограмме длиной 600 кардиоциклов;
2) дискретизация — преобразование данных ЭКГ-сигнала в 599-символьную строку в 6-буквенном алфавите (кодограмму);
3) векторизация — преобразование в вектор 63=216 частот триграмм.

На этапе машинного обучения происходит:

1) формирование последовательностей триграмм, которые характеризуют определенное состояние здоровья или недугов; проведение статистического анализ информативности признаков (триграмм);
2) обучение модели классификации – на базе существующих данных ЭКГ здоровых и больных пациентов;