СЕРВИС ДЛЯ РЕШЕНИЯ КОМБИНАТОРНЫХ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ И ТЕХНОЛОГИИ CUDA

Колосов А.Д.
(г. Иркутск, Институт динамики систем и теории управления имени В.М.Матросова СО РАН)

SERVICE FOR SOLVING OF COMBINATORIAL PROBLEMS USING NEURAL NETWORKS AND CUDA TECHNOLOGY

Kolosov A.D.
(Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences)

We consider service to automate the computational experiments for solving combinatorial optimization problems by using the Wang recurrent neural network in distributed heterogeneous computing environment . The implementation of the algorithm for solving the traveling salesman problem on the GPU using technology CUDA is described. Experimental results with the algorithm Ant System (AS) and its implementation on the GPU are compared.

В докладе рассматривается разработка и реализация программных средств для решения ряда задач комбинаторной оптимизации, возникающих при организации распределенных вычислений в гетерогенной интегрированной кластерной системе, с применением технологии искусственных нейронных сетей (ИНС). Процесс проектирования и обучения ИНС приводит к существенным издержкам времени для сетей высокой размерности. Интенсивное развитие высокопроизводительной вычислительной техники предоставляет возможность снизить эти издержки путем применения параллельных технологий, позволяющих, в частности, производить вычисления общего назначения на графических адаптерах. Одной из наиболее перспективных является технология CUDA, разрабатываемая компанией Nvidia. Нейросетевые алгоритмы хорошо подходят для распараллеливания с помощью CUDA [1-3].

В настоящее время представляется актуальным создание инструментальных средств, обеспечивающих разработку программных сред для доступа предметных специалистов к таким высокопроизводительным вычислительным ресурсам и использования этих ресурсов без необходимости углубленного знания вычислительных архитектур и низкоуровневых средств разработки приложений. Одновременно прослеживаются тенденции, с одной стороны, использования сервис-ориентированного подхода, подразумевающего организацию сервисов доступа к вычислительным ресурсам посредством сети интернет, с другой стороны, оформления функций приложения пользователя в виде сервиса.

Целью проводимых исследований являлась разработка сервис-ориентированных средств для автоматизации проведения вычислительных экспериментов при решении упомянутых выше задач с помощью рекуррентных нейронных сетей, активно использующихся для их решения в настоящее время. Однако подбор таких параметров функции энергии сети (в случае использования рекуррентных нейронных сетей), при

которых достигается хорошая устойчивость сети и близость к оптимальному значению результата, приводит к необходимости проведения многовариантных расчетов. Такие вычисления являются не только ресурсоемкими, но и требуют больших трудозатрат пользователя. Эти обстоятельства актуализируют создание инструментария, сокращающего время выполнения расчетов и повышающего эффективность работы пользователя.

В докладе рассматривается разработка сервиса, автоматизирующего проведение вычислительных экспериментов при решении комбинаторных задач с помощью рекуррентных нейронных сетей и использованием технологии CUDA, описываются аспекты программной реализации сервиса путем применения инструментальных средств, описанных в работе [4,5]. Приводятся результаты тестирования этого сервиса для решения ряда примеров задачи о коммивояжере с помощью ИНС Вана [6] в гетерогенной вычислительной среде. Для создания сервиса использовались инструментальные средства HpcSoMaS (High-performance Service-oriented Multi-agent system) Framework [4].

Рекуррентная нейронная сеть. Отдельную группу ИНС составляют сети с обратными связями между различными слоями нейронов. Это так называемые рекуррентные сети. Их общая черта — передача сигналов с выходного или скрытого слоя во входной слой. Главная особенность таких сетей — динамические зависимости на каждом этапе функционирования. У сетей без обратных связей нет памяти, их выход полностью определяется текущими входами и значениями весов. В отличие от сетей прямого распространения, которые выдают результат за определенное количество шагов, результатом рекуррентной сети является устойчивое состояние. Примером рекуррентной сети может служить ИНС Вана.

Структура сети Вана подробно описывается в работе [7]. Сходимость и устойчивость сети рассмотрена в работах [8,9,10]. Сеть Вана широко применяется для решения различных комбинаторных задач, задач оптимизации. С помощью данной нейронной сети решаются так же некоторые задачи, возникающие при организации распределенных вычислений. Например, с использованием нейронной сети Вана успешно решаются задачи о назначении [7,10]. Задача маршрутизации может быть сведена к задаче размещения [11].

Нейронная сеть Вана описывается дифференциальным уравнением [7]:

$$\frac{du_{ij}^t}{dt} = -\eta \sum_{l=1}^{n} x_{il}^t - \eta \sum_{k=1}^{n} x_{kj}^t + 2\eta - \lambda c_{ij} e^{-\frac{t}{\tau}} \qquad (1)$$

где

$$x_{ij} = \frac{1}{1+e^{-\beta u_{ij}}}. \qquad (2)$$

Данное уравнение решается численным методом. Одним из самых простых в реализации является метод Эйлера. Для решения уравнения методом Эйлера строится следующий итерационный процесс:

$$u_{ij}^{t+1} = u_{ij}^t - \Delta t \left[\eta \left(\sum_{l=1}^{n} x_{il}^t + \sum_{k=1}^{n} x_{kj}^t - 2 \right) + \lambda c_{ij} e^{-\frac{t}{\tau}} \right], \quad (3)$$

где Δt – шаг по времени; t – момент времени; c_{ij}–стоимость назначения элемента i в позицию j; Δt, η, λ, τ, β – коэффициенты, подбираемые экспериментально. Подбор коэффициентов существенно влияет на скорость сходимости сети и качество решения.

Решение уравнения сети Вана можно ускорить, применив принцип «Победитель получает всё» (“Winner takes all”). Согласно этому принципу, активным остается нейрон с самым большим выходом, остальные нейроны в строке и столбце становятся неактивными [9,11]. В данной работе нейронная сеть Вана применяется для решения задачи

коммивояжера. Задача коммивояжера – классическая задача комбинаторной оптимизации. Целью является нахождение гамильтонова цикла с минимальной суммарной стоимостью во взвешенном полносвязном графе. Эта задача выбрана для тестирования сервиса, автоматизирующего проведение вычислительных экспериментов с ИНС на GPU, т.к. для нее существует большой набор тестов с известным оптимальным решением, и можно сравнивать полученные результаты с имеющимися.

Алгоритм для сети Вана. Для ускорения ИНС Вана была распараллелена на GPU с применением технологии CUDA. Нейросетевая динамика рекуррентной нейронной сети Вана состоит в многократном циклическом пересчете матрицы весовых коэффициентов и матрицы состояния сети. К каждому элементу матрицы применяется одинаковый набор инструкций, что соответствует архитектуре SIMD (Single Instruction, Multiple Data). GPU хорошо подходит для реализации параллелизма по данным. Каждому потоку достаётся по одному элементу матрицы для обработки. Это позволяет получить заметное ускорение по сравнению с последовательной версией. Перенос данных CPU-GPU минимален, так как матрицы переносятся на GPU однократно при запуске программы, больше в алгоритме нет затратных операций обмена. Алгоритм WTA в программе выполняется последовательно в силу своей «последовательной природы». Кроме того, выполнение WTA занимает менее одного процента от времени работы программы. Программная реализация алгоритма на языке C++ с использованием технологии CUDA названа в докладе gtsp.

Разработка сервиса. Сервис обеспечивает следующие возможности, поддерживаемые инструментарием HpcSoMas Framework:

- Web-интерфейс для запроса пользователя на решение задачи. Задание параметров для проведения вычислительного эксперимента: входные и выходные данные программы, реализующей нейросетевой алгоритм, наблюдаемые выходные параметры, на основании значения которых делается вывод об эффективности решения для заданных входных параметров.
- Прозрачное для пользователя параллельное решение для различных вариантов входных параметров в распределенной вычислительной среде.
- Информирование пользователя об окончании решения. Обработка и визуализация результатов решения.

Сервис для решения задачи коммивояжера (TSP) содержит следующие группы входных параметров (рис.2):

- Параметры Δt, η, λ, τ, β , из формул (1)-(3), которые могут варьироваться в процессе вычислительного эксперимента. Эти параметры могут быть заданы одним значением (заполняется только крайнее левое поле ввода на форме, представленной на рис. 1), в этом случае сервис подбирает подходящий ресурс для запуска задания пользователя и последующих этапов процесса выполнения. Эти параметры могут так же быть заданы интервалом изменения значений от нижней границы до верхней с определенным шагом. При этом сервисом организуются многовариантные вычисления с целью подбора такой комбинации параметров, при которой будет достигнуто оптимальное решение. В этом случае сервис формирует набор заданий, подбирает подходящие вычислительные ресурсы, запускает задания (в зависимости от загруженности ресурса) или ставит в очередь к соответствующей системе управления заданиями.

- Параметры gridXY и blockXY, определяющие структуру сетки (grid) для GPU.
- Параметры, задающие входные данные – размерность задачи (количество городов), расстояния между городами, количество итераций, предотвращающее зацикливание алгоритма при попытке улучшить решение.
- Параметры, задающие постобработку выходных данных, называемых наблюдаемыми параметрами. Эти параметры используются для получения статистических данных: средней длины маршрута, среднего и общего времени выполнения всего задания, среднего и общего времени выполнения отдельных этапов и т.д. Вид обработки включает нахождение среднего, минимального, максимального или суммарного значения для наблюдаемого параметра.
- Параметр сервиса – количество повторных запусков для получения статистики по наблюдаемым параметрам.

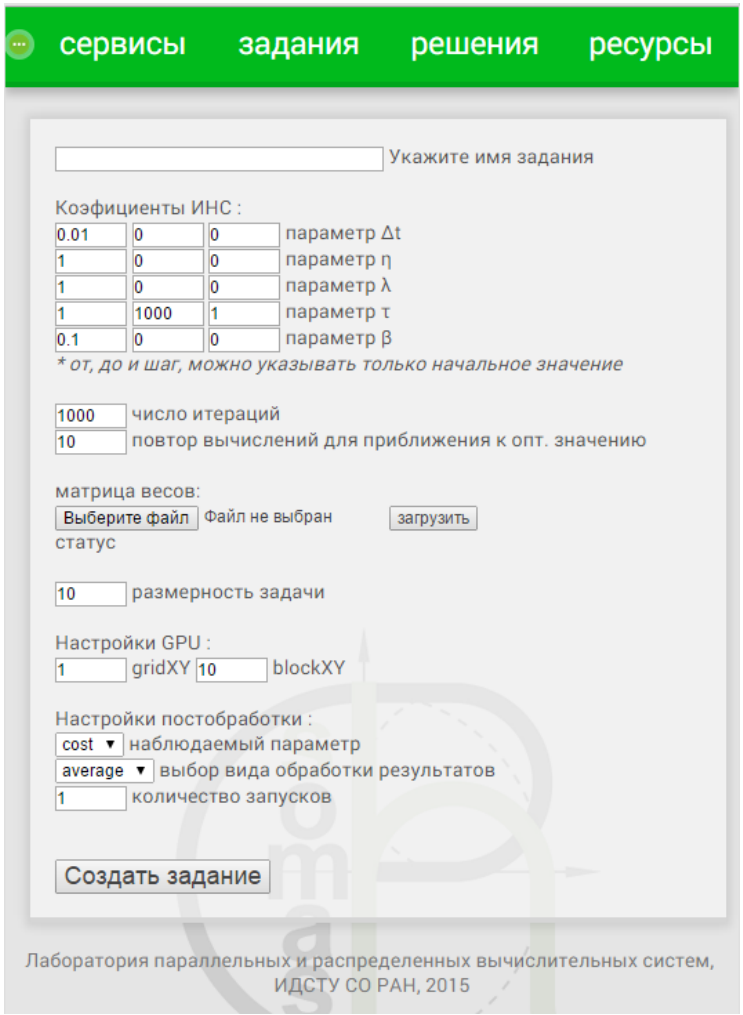

Рис. 1. Web-интерфейс сервиса для решения задачи TSP с помощью ИНС и использованием GPU.

На этапе обработки данных для решения задачи коммивояжера можно визуализировать путь с помощью сервиса обработки. Ниже приведен пример визуализации для задачи небольшой размерности, 10 городов. Пример взят из работы [12]. С помощью варьирования параметра Δt постепенно получает оптимальный маршрут. На рис. 2 показаны субоптимальный и оптимальный маршруты.

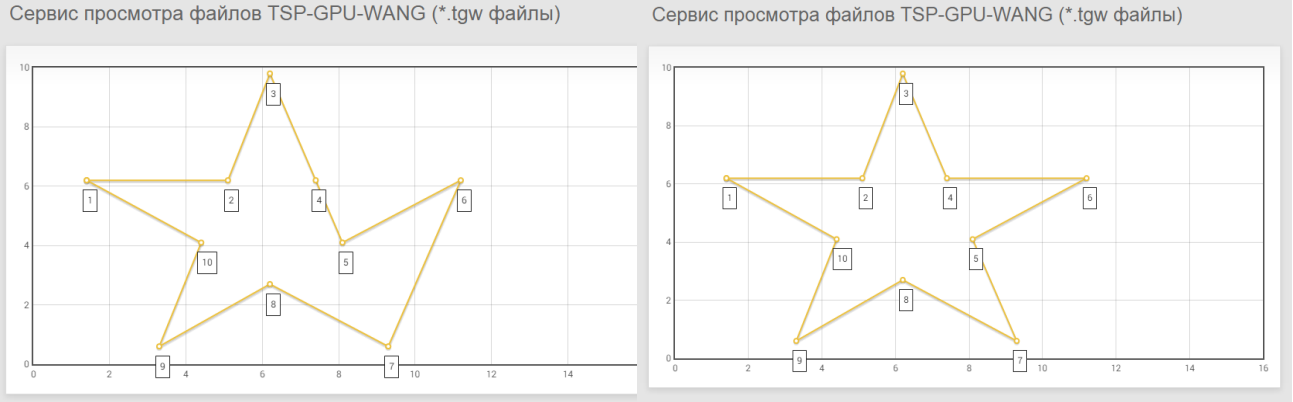

Рис. 2. Визуализация маршрутов с помощью сервиса.

Инструментарий HpcSoMas Framework. Инструментальные средства (ИС) HpcSoMaS Framework предназначены для разработки мультиагентных систем, автоматизирующих процесс управления распределенными вычислениями. Особенностью таких систем является реализация агентов в виде сервисов. ИС включают средства планирования ресурсов, обеспечивающие выбор наименее загруженного вычислительного кластера (ВК), отвечающего требованиям пользователя, и средства организации взаимодействия с системой управления заданиями (СУПЗ) ВК. В состав ИС HpcSoMaS Framework входят: средства создания агентов на базе нейронных сетей; библиотека разработки сервисов на основе стандарта REST (Representational State Transfer) и протокола SOAP (Simple Object Access Protocol), а так же готовые сервисы, реализующие базовые функции агентов, созданные на основе библиотечных классов и требующие для своего использования только конфигурационную настройку; графические средства проектирования сервисов; документация по используемым форматам файлов конфигурации сервисов.

Основными компонентами в HpcSoMaS Framework являются агенты-менеджеры, которые и производят основную работу по распределению заданий, подбор наилучшей вычислительной среды под конкретную задачу, анализ состояний СУПЗ, перераспределение заданий, их декомпозицию. Данные агенты представлены в виде rest-сервисов. Эти сервисы реализованы как сервлеты с помощью средств Java EE. Для выполнения различных функций на вычислительных кластерах, зависящих от конфигурации ВК, они запускают soap-сервисы.

Этапы проведения вычислительного эксперимента с помощью рассматриваемого сервиса приведены на рис. 3. Агент-менеджер через свой web-интерфейс получает и обрабатывает запросы, запуская для обработки вычислительные сервисы и сервисы визуализации результатов.

Вычислительные эксперименты. При создании сервиса в качестве распределенной вычислительной среды была использована разнородная кластерная Grid-система ИДСТУ СО РАН, включающая: однородный ВК «Blackford» с SMP-узлами, неоднородный ВК «Академик В.М. Матросов» с SMP-узлами и узлом с графическими процессорами NVidia C2070 («Fermi»); неоднородный ВК «Tesla», в состав которого входит 4 четырехъядерных процессора и 8 GPU NVidia «Tesla» C1060. Для тестирования сервиса были выбраны задачи из библиотеки TSPLIB [13], для которых в работах [12,14,15] были приведены результаты, полученные с помощью других алгоритмов.

Эксперимент проводился следующим образом. С помощью сервиса варьировались коэффициенты, используемые в формулах (1)-(3), затем с этими коэффициентами производилась серия запусков решения каждой задачи с нахождением средней длины пути для оценки качества полученного решения. Сервис показал работоспособность и экономию трудозатрат при проведении экспериментов. Результаты экспериментов запусков gtsp сравнивались с результатами, приведенными в [12,15] для известного «муравьиного» алгоритма AS (Ant System), обозначен как AS-CPU, и его параллельной реализации на GPU (AS-GPU). На рис. 4 приведено качество решения для задач с минимальной размерностью 51 и максимальной 13509. Качество (аналогично [15]) определяется как отношение оптимального значения к полученному в результате работы тестируемого алгоритма.

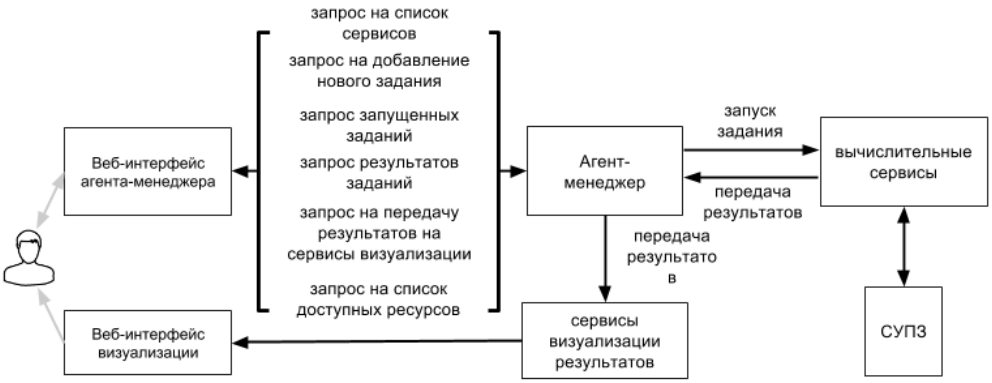

Рис. 3. Схема решения задачи с использованием сервиса.

Для городов небольших размерностей качественное отклонение от оптимального решения достигает приблизительно 0,8, как у последовательного алгоритма AS, работающего на CPU, так и у рассматриваемого в докладе gtsp, использующего GPU. Качество решения лучше при этом у AS-CPU, однако с увеличением размерности, начиная с примера pcb442 (для 442 городов), качество решения становится одинаковым, с увеличением размерности качество решения AS-CPU начинает падать, а для размерностей от 2392 данные для этого алгоритма отсутствуют. Разработанный алгоритм gtsp работает довольно стабильно. На рис. 5 приведены данные, характеризующие качество решения, для трех алгоритмов, gtsp, AS-CPU и AS-GPU для примеров, приведенных в [12,15]. Видно, что у AS-GPU качество решения ухудшается с увеличением размерности. Данные по городам большей размерности для него автору тоже неизвестны.

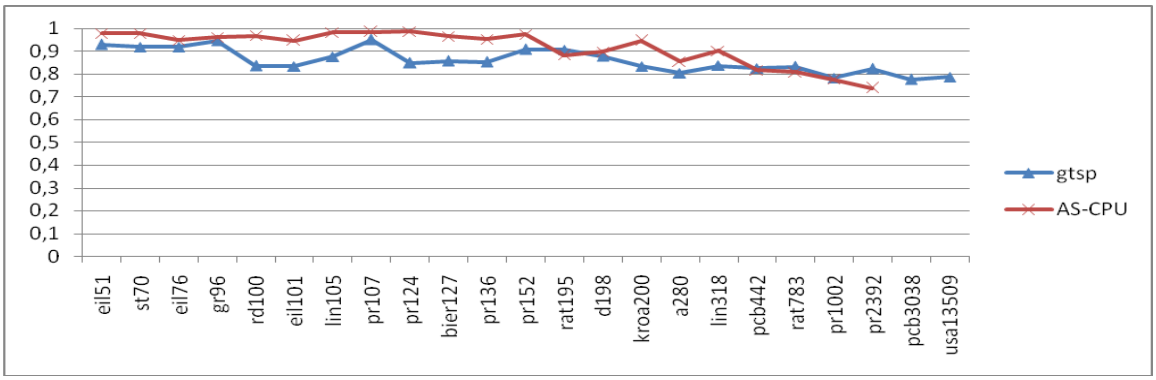

Рис. 4. Сравнение качества решения gtsp и AS-CPU.

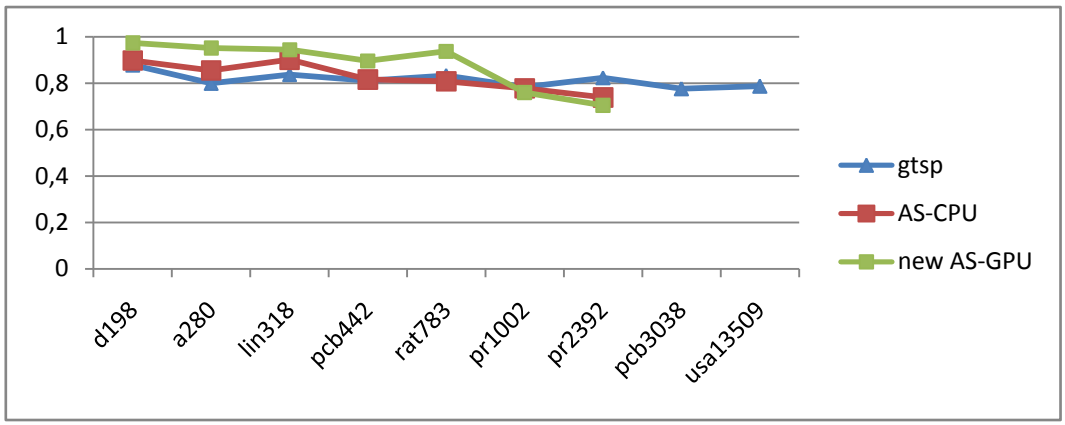

Рис. 5. Сравнение результатов качества работы трех алгоритмов.

На рис. 6 приведено ускорение реализаций на GPU в сравнении с CPU, данные времени выполнения AS-CPU и AS-GPU в табл. 1 взяты из работы [14]. Эти данные приведены для 100 итераций работы AS-CPU и ASD-GPU. Поэтому для сравнения gtsp тоже запускался для 100 итераций. При большем количестве итераций качество решения, представленное на рис. 4 и рис. 5, получаемое с помощью gtsp, может быть улучшено, иногда существенно.

Таблица 1. Общее время выполнения для 100 итераций.

		Общее время решения, в мсек.		
Примеры	Размерность	AS-CPU	AS-GPU	gtsp
d198	198	2080,72	263,91	255,96
a280	280	4844,59	505,51	529,95
lin318	318	9797,03	897,29	979,79
pcb442	442	18534,37	1153,95	1467,74
rat783	783	123220,58	5673,15	5315,48
pr1002	1002	381949,72	8706,18	11633,98
pr2392	2393	5867605,87	208478,2	163687,9

Ускорение gtsp, как видно из рис. 6, иногда меньше, чем у AS-GPU, но довольно стабильное. Ускорение же AS-GPU резко падает с увеличением размерности.

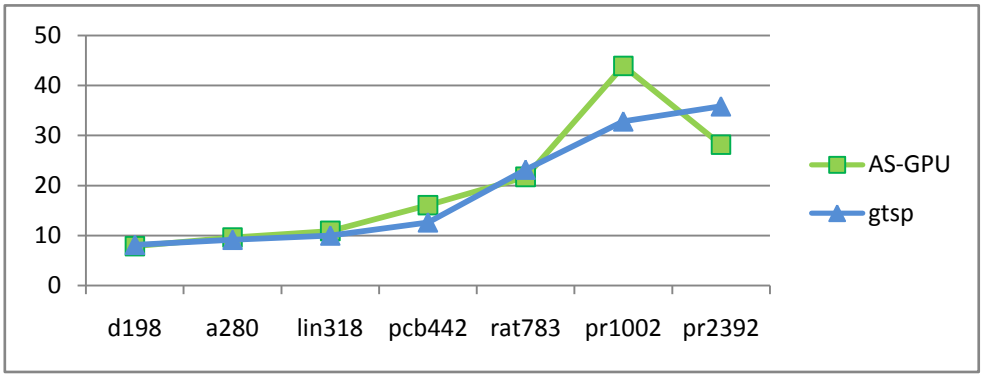

Рис. 6. Ускорение gtsp и AS-GPU по сравнению с AS-CPU.

Заключение. В целом можно сделать вывод, что рассматриваемые в докладе программные средства показали свою работоспособность и эффективность для задач достаточно большой размерности. Дальнейшее направление исследований связано с улучшением качества решения, оптимизацией использования GPU и применением представленного подхода для решения новых задач.

ЛИТЕРАТУРА

1. Sierra-Canto X., Madera-Ramirez F., Cetina V.U. Parallel Training of a Back-Propagation Neural Network Using CUDA// ICMLA, IEEE Computer Society. 2010. Pp. 307-312.
2. Bastos Filho C., Oliveira M., Silva D., Santana R.A. Optimizing a routing algorithm based on Hopfield Neural Networks for Graphic Processing Units // FOCI, IEEE. 2011. Pp. 88-93.
3. Ciresan D.C., Meier U., Masci J., Gambardella L.M., Schmidhuber J. High-Performance Neural Networks for Visual Object Classification // CoRR. – 2011. – Pp. 1250–1254.
4. Богданова В.Г., Пашинин А.А. Мультиагентная система управления распределенными вычислениями HpcSoMas. – Свид. о гос. рег. программы для ЭВМ №2014661359. – М.: Федеральная служба по интеллектуальной собственности, патентам и товарным знакам, 2014.
5. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Пашинин А.А. Мультиагентные методы и инструментальные средства управления в сервис-ориентированной распределенной вычислительной среде // Труды Института системного программирования РАН. 2014. Т. 26. Вып. 5. С. 65-82.
6. Hung D.L., Wang J. Digital hardware realization of a recurrent neural network for solving the assignment problem // Neurocomputing. 2003. Vol. 51. pp. 447-461.
7. J. Wang, An analog neural network for solving the assignment problem, Electron. Lett. 28 (11) (1992) 1047–1050.
8. J. Wang, Analysis and design of a recurrent neural network for linear programming, IEEE Trans.2 Circuits Syst. I 40 (9) (1993) 613–618.
9. М.С.Тарков, Параллельный алгоритм построения гамильтоновых циклов в графе распределенной вычислительной системы рекуррентной нейронной сетью. УДК 004.032.26(06).
10. Hung D.L., Wang J. Digital hardware realization of a recurrent neural network for solving the assignment problem// Neurocomputing. 2003. vol. 51. pp. 447-461.
11. Siqueira P.H., Steiner M.T.A., Scheer S. A new approach to solve the travelling salesman problem// Neurocomputing. 2007. vol. 70. pp. 1013-1021.
12. Paulo Henrique Siqueira, Maria Teresinha Arns Steiner, Serrgio Scheer. A new approach to solve the traveling salesman problem// Neurocomputing 70 (2007). pp. 1013–1021.
13. http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/index.html.
14. Akihiro Uchida, Yasuaki Ito, Koji Nakano. An Efficient GPU Implementation of Ant Colony Optimization for the Traveling Salesman Problem // International Conference on Computing, Networking and Communications. − 2012. − pp. 94-102, doi:10.1109/ICNC.2012.22.
15. Laurence Dawson, and Iain A. Stewart. Improving Ant Colony Optimization performance on the GPU using CUDA // IEEE Congress on Evolutionary Computation, pp. 1901-1908. IEEE, (2013)