

УДК 004.75

ПРИМЕНЕНИЕ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ ДЛЯ СОЗДАНИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

¹Ляшов М.В., ²Берёза А.Н., ²Бабаев А.М., ²Алексеев Ю.В., ²Авдеева Т.Г.

¹Донской государственный технический университет, Ростов-на-Дону, e-mail: maxl85@mail.ru;

²Институт сферы обслуживания и предпринимательства (филиал), Донской государственный технический университет, Шахты, e-mail: anbirch@mail.ru, babaev-andrei2011@yandex.ru, ajulyav@mail.ru, tan.awdeewa2011@yandex.ru

В работе предложено решение проблемы объединения разнородных вычислительных ресурсов в единую распределенную систему с применением сервис-ориентированной архитектуры и концепции Grid-вычислений. Применение сервис-ориентированной архитектуры для создания Grid-систем позволит осуществить интеграцию разнородных вычислительных ресурсов не только локальных, но и географически распределенных. Предлагаемая платформа не предназначена для замены или вытеснения других Grid-систем. Вместо этого она дает возможность использования в едином вычислительном процессе на базе сервис-ориентированной архитектуры различные платформы: центральные процессоры (CPU), графические ускорители (GPU), программируемые логические интегральные схемы (FPGA). Для проведения экспериментальных исследований были выбраны алгоритмы Deep Learning, которые требуют высокопроизводительных вычислений. Тестирование работы алгоритмов Deep Learning было произведено на наборах данных MNIST, CIFAR-10, Molecular.

Ключевые слова: высокопроизводительные вычисления, сервис-ориентированная архитектура, GRID-вычисления, гетерогенные системы, .NET, Windows Communication Foundation

APPLICATION OF SERVICE ORIENTED ARCHITECTURE FOR DISTRIBUTED COMPUTING SYSTEMS

¹Lyashov M.V., ²Bereza A.N., ²Babaev A.M., ²Alekseenko Yu.V., ²Avdeeva T.G.

¹Don State Technical University, Rostov-on-Don, e-mail: maxl85@mail.ru;

²Institute of Entrepreneurship and Service sector (branch) Don State Technical University, Shakhty, e-mail: anbirch@mail.ru, babaev-andrei2011@yandex.ru, ajulyav@mail.ru, tan.awdeewa2011@yandex.ru

The paper provides a solution to the problem of uniting heterogeneous computing resources into a unified heterogeneous system using service-oriented architecture (SOA) and the conception of Grid-computing, where computing resources of clients are presented as a service. The use of SOA for creating Grid-systems will allow the clients to implement integration of not only local separated heterogeneous computing resources, but also geographically separated ones. The suggested platform is not intended to replace or displace other Grid-systems. Instead of this, it gives the possibility of using different platforms based on SOA in a unified computing process. For experimental studies Deep Learning algorithms and datasets MNIST, CIFAR-10 have been chosen, which require high-performance computing.

Keywords: high performance computing, service-oriented architecture, Grid-computing, heterogeneous system, .NET, Windows Communication Foundation

В настоящее время возросла потребность в масштабных расчетах уже не только в научных целях, но и на производстве, например в материаловедении, химии, разработке лекарственных препаратов, конструировании высокотехнологичных машин и др. Суперкомпьютерные системы могут решать большой спектр задач, но производство и эксплуатация таких компьютеров требует больших расходов, поэтому сегодня в мире их насчитывается всего несколько сотен.

В Grid-системах можно объединять разнородные вычислительные ресурсы, и они широко используются для быстрой и относительно дешевой обработки большого объема вычисленных задач. Обычно это достигается путем деления сложных задач на более простые подзадачи (задания), ко-

торые могут быть обработаны параллельно с использованием доступных вычислительных ресурсов [7].

При организации Grid-вычислений необходимо решить проблему объединения разнородных вычислительных ресурсов в единую распределенную систему. В настоящее время в ИТ индустрии при создании информационных систем получила большое распространение сервис-ориентированная архитектура, благодаря своему главному преимуществу: возможности эффективной интеграции информационных ресурсов [1, 6, 7].

Сервис-ориентированная архитектура (англ. Service-Oriented Architecture, SOA) – это модульный подход к разработке программного обеспечения. Данный подход

использует распределенные, заменяемые и слабо связанные компоненты, которые взаимодействуют между собой по стандартизированным протоколам (SOAP и др.). При этом компоненты реализуются как набор веб-служб [1, 2].

В данной работе показана возможность применения концепции SOA для распределенных вычислений. Применение сервис-ориентированной архитектуры для создания Grid-систем позволит осуществить интеграцию разнородных вычислительных ресурсов не только локальных, но и географически распределенных.

Технологии SOA, применяемые для построения Grid-систем

Рассмотрим основные технологии, которые могут быть использованы для построения Grid-систем на основе SOA с использованием веб-служб. Веб-служба (или веб-сервис, англ. web service) – программная система со стандартизированным интерфейсом, которая доступна по определенному веб-адресу. Взаимодействие веб-служб с другими службами либо со сторонними приложениями происходит на основе протоколов SOAP, XML-RPC, REST и т.д. [1, 2].

Для обеспечения взаимодействия веб-служб с другими службами либо со сторонними приложениями используются следующие стандарты: XML: Расширяемый язык разметки, предназначенный для хранения и передачи структурированных данных; SOAP: Протокол обмена сообщениями на базе XML; WSDL: Язык описания внешних интерфейсов веб-службы на базе XML.

Веб-служба состоит из нескольких частей, имеющих собственное назначение и поведение в системе. На рис. 1 представлена структурная схема веб-службы.

Для Grid-систем в качестве службы может выступать любая программная платформа, которая управляет вычислительным ресурсом. Клиенты службы используют канал, который совместим с каналом службы, чтобы вызывать методы службы, передавать необходимые данные для вычислений

и получать назад результаты расчетов. Сообщения могут передаваться через этот канал в любом направлении.

Главная особенность использования веб-служб в SOA – для того чтобы выполнить определенную задачу службы, необходимо вызвать метод данной службы с четко определённым интерфейсом. При этом веб-служба заранее ничего не знает о приложении, которое их вызовет, а приложение не знает, каким образом веб-службы выполняют свою задачу. Следовательно, SOA может рассматриваться как способ построения Grid-систем, который позволяет объединять разнородные ресурсы, путём комбинации слабосвязанных и взаимодействующих сервисов. Эти сервисы будут взаимодействовать на основе строго определённого кроссплатформенного интерфейса. Определение интерфейса скрывает языково-зависимую реализацию веб-службы.

Таким образом, можно выделить следующие достоинства применения SOA для построения распределенных вычислительных систем:

- Использование SOA позволяет создать не жестко связанную систему, в которой клиент службы и реализация службы имеют в общем случае список доступных для всех служебных действий и определенную структуру данных.

- Клиент знает только интерфейсы службы, имена и типы входных и возвращаемых значений функцией службы. Никакой другой зависимости нет.

- Платформы и языки программирования могут быть различны как у клиента, так и у службы.

- Разделение задач. Реализация службы и клиента может выполняться разными работчиками.

Windows Communication Foundation

Проанализировав требования к системе распределенных вычислений и наиболее распространенные на данный момент аналоги, авторы при выборе программной платформы остановились на платформе

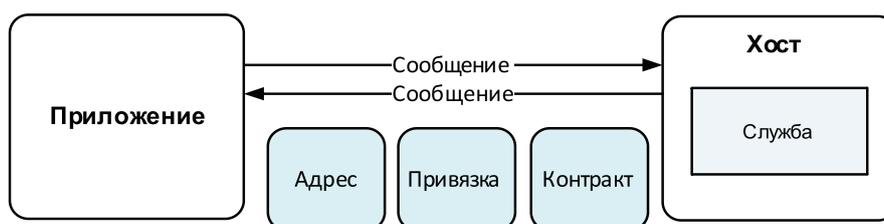


Рис. 1. Работа службы

.NET и программном фреймворке Windows Communication Foundation (WCF) от компании Microsoft [2, 6]. При реализации модулей системы предполагается активно использовать стандарт OpenCL. Это позволит добиться максимальной независимости от архитектуры клиентских ЭВМ и одновременно с этим позволит прозрачно использовать GPU и FPGA, для значительного ускорения проводимых расчетов.

WCF – это самый простой способ предоставлять и использовать службы на платформе Microsoft. Пользуясь WCF, разработчики могут сосредоточиться на приложениях, а не на коммуникационных протоколах. Это классический пример инкапсуляции технологии в инструментальных средствах [2].

Предлагаемая архитектура

Система для высокопроизводительных вычислений, разработанная в соответствии с сервис-ориентированной архитектурой, была реализована как набор веб-служб, взаимодействующих по протоколу SOAP. Данный протокол используется для реализации удалённого вызова процедур и для обмена произвольными сообщениями между узлами системы в формате WSDL. Разнородные вычислительные ресурсы (CPU, GPU и FPGA) клиентов предоставляются как сервис и реализованы как набор веб-служб, т.е. они представляют собой идентифицируемую веб-адресом программную систему со стандартизированными интерфейсами. Вычислительные ресурсы распределенной вычислительной системы, реализованные как веб-службы, могут взаимодействовать друг с другом, с сервером и со сторонними приложениями посредством сообщений, основанных на протоколе SOAP.

Наш проект не предполагает разработку и внедрение нового фреймворка на основе SOA. Вместо этого предложен способ предоставления вычислительных ресурсов с множеством настраиваемых сервисов

WCF, который может преодолеть ограничения традиционных веб-служб XML.

В общем виде архитектура разработанной гибридной реконфигурируемой системы для высокопроизводительных вычислений приведена на рис. 2. Любая вычислительная задача, решаемая на представленной системе, состоит из клиентской и серверной частей. Основная вычислительная нагрузка ложится на клиентов, которые предоставляют свои вычислительные ресурсы в качестве WCF-сервиса [2].

Серверная часть представляет собой WCF-клиент, который по локальной сети или через Интернет обращается к вычислительным ресурсам клиентов. Основными функциями серверной части разработанной гибридной системы являются: оценка производительности клиентов и скорости соединения; мониторинг клиентов и процесса вычисления; окончательная обработка результатов вычислений, полученных от клиентов.

Экспериментальные исследования

Для экспериментальных исследований на предлагаемой платформе были выбраны алгоритмы Deep Learning, поскольку это все более популярная тема в области искусственного интеллекта и одна из самых востребованных областей в компьютерной науке [5]. Применение алгоритмов Deep Learning позволило создавать системы, которые способны распознавать зашумленные и искаженные предметы на фотографиях, делать сложные предсказания о возможном будущем поведении человека в видеопотоке, анализировать текстовую информацию и т.п. При этом реализация данных алгоритмов требует высокопроизводительных вычислений.

На данный момент многими разработчиками и исследователями в области Deep Learning используется библиотека TensorFlow [3]. TensorFlow – это новая библиотека для создания систем машинного обучения с открытым исходным кодом от Google.

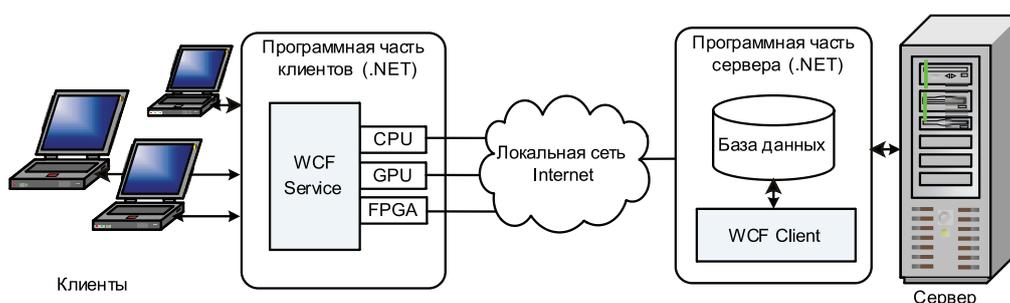


Рис. 2. Архитектура предлагаемой системы для высокопроизводительных вычислений

Тестовые наборы данных

Набор данных (Datasets)	Размер	Кол-во параметров	Скрытых слоев	Скрытых нейронов
MNIST	60 К×784 = 39,2 Мб	1,8 М	2	512
CIFAR-10	60 К×1024 = 170 Мб	4,1 М	2	1024
Molecular	150 К×2871 = 430 Мб	14,2 М	3	2048

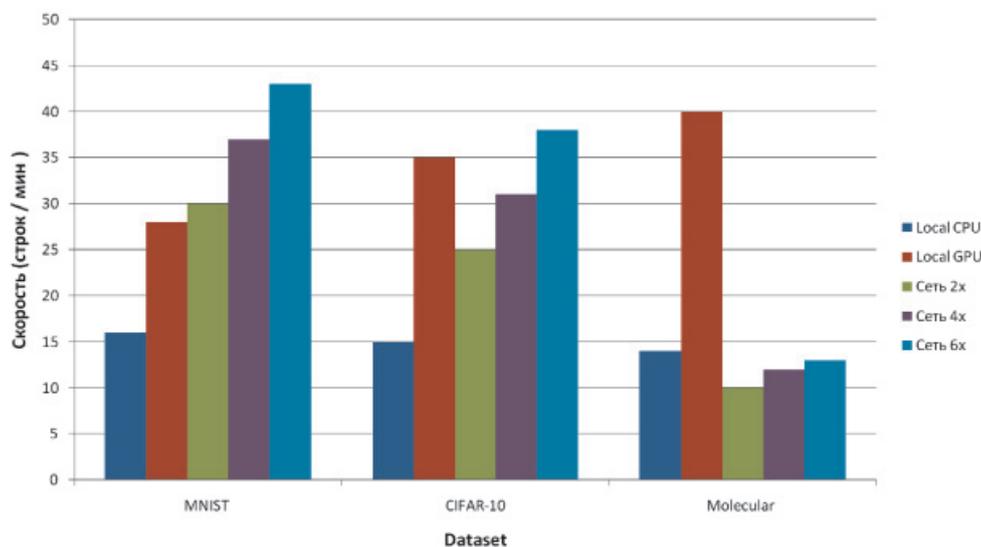


Рис. 3. Результаты экспериментальных исследований

Для экспериментальных исследований были использованы тестовые наборы данных (таблица), которые чаще всего используются при сравнении алгоритмов Deep Learning: MNIST – это набор изображений рукописных цифр; CIFAR-10 – это набор цветных изображений 10 различных классов; Molecular – набор молекулярной активности веществ для создания лекарств [3– 5].

В последних трех столбцах таблицы указаны параметры реализованных глубоких нейронных сетей.

На рис. 3 показаны результаты обучения глубоких нейронных сетей на разработанной платформе.

Для исследований были построены сети с 2, 4 и 6 клиентами. Из этих результатов видно, что у предлагаемой системы такой же недостаток, как и у подобных ей систем: чем больше передаваемый объем данных по сети, тем ниже эффект от ее использования.

Для преодоления этого недостатка необходимо учитывать объем передаваемых данных по сети и подбирать соответствующие алгоритмы. Наиболее часто используемый метод оптимизации для обучения глубоких нейронных сетей – это стохастический градиентный спуск (SGD). Классический алгоритм SGD по своей сути является последова-

тельным, что делает нецелесообразным его применение к очень большим наборам данных, т.к. требуется много времени для передачи данных последовательным способом.

Для распределенного обучения нейронных сетей мы применили асинхронный стохастический градиентный спуск [4]. Основная особенность этого алгоритма состоит в том, что на каждом клиенте находится свой собственный экземпляр данных и обучаемая модель. Это значительно сокращает нагрузку на сеть. Результаты экспериментов показаны на рис. 4.

Из этого графика видно, что предпринятые меры позволили добиться эффективности в расчетах на разработанной платформе. То есть с добавлением клиентов в сети растет и скорость вычислений.

Выводы

Разработанная архитектура предлагает больше гибкости, простоты и совместимости, которые необходимы, чтобы обеспечить интеграцию между сетевыми ресурсами. Предлагаемая платформа не предназначена для замены или вытеснения других Grid-систем. Вместо этого она дает возможность использования в едином вычислительном процессе различных платформ (CPU, GPU,

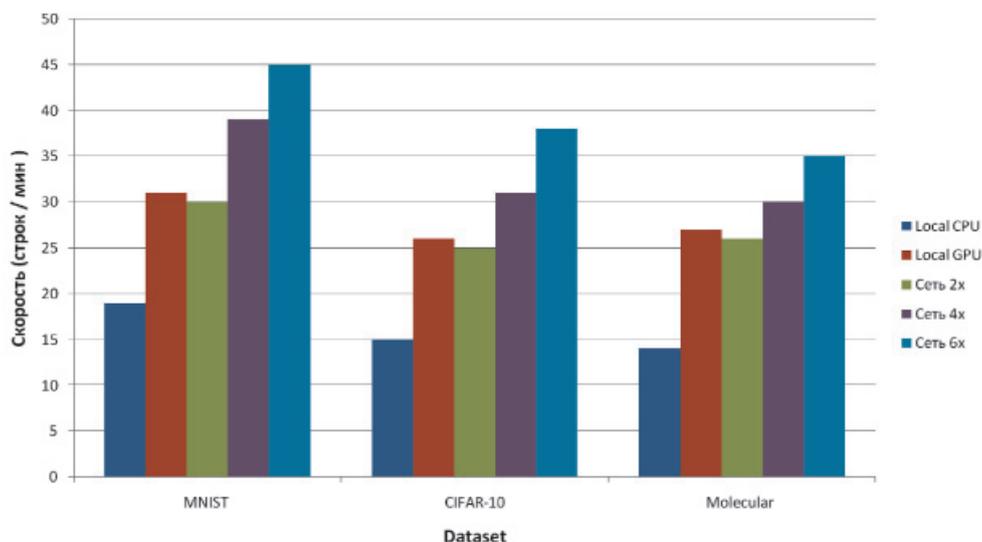


Рис. 4. Результаты экспериментальных исследований алгоритма асинхронного стохастического градиентного спуска

FPGA) на базе сервис-ориентированной архитектуры, а использование OpenCL позволяет создать в рамках этой сети гетерогенную вычислительную среду. Применяя технологию распределенных вычислений и задействовав неиспользуемые ресурсы персональных компьютеров, система позволяет снизить стоимость вычислений на единицу производительности.

Работа выполнена при частичной финансовой поддержке грантов РФФИ № 16-37-60016, № 14-07-00816.

Список литературы

1. Данилов И.Г. Сервис-ориентированная архитектура как основа для построения современных распределенных систем // Известия ЮФУ. Технические науки. – 2010. – № 7. – С. 168–173.
2. Cibraro P. Professional WCF 4: Windows Communication Foundation with .NET 4. – Microsoft Press, 2010.
3. Dean J., Rajat M., et al. TensorFlow: Large-scale machine learning on heterogeneous systems. TensorFlow.org. Google Research. Retrieved 10 November 2015.
4. Dean J., Corrado G., et al. Large Scale Distributed Deep Networks. NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada, United States, December, 2012. – P. 1232–1240.
5. Li Deng and Dong Yu Deep Learning: Methods and Applications // Foundations and Trends in Signal Processing. – 2014. – Vol. 7, № 3–4. – P. 197–387.
6. Lyashov M.V., Bereza A.N., Alekseenko J.V., Blanco L.M.L. The hybrid reconfigurable system for high-

performance computing // Application of Information and Communication Technologies (AICT), 2015 9th International Conference on. – Rostov on Don, 2015. – P. 258–262.

7. Shrihari, S.V., Pawan, K., Vishnu, M., Mahajan, S. Technological aspects of grid computing (2015) // 2nd International Conference on Electronics and Communication Systems, ICECS. – 2015. – art. № 7124829. – P. 1463–1468.

References

1. Danilov I.G. Servis-orientirovannaya arkhitektura kak osnova dlya postroyeniya sovremennykh raspredelennykh sistem. Izvestiya YUFU. Tekhnicheskiye nauki. 2010. no. 7. pp. 168–173.
2. Cibraro P. Professional WCF 4: Windows Communication Foundation with .NET 4. Microsoft Press, 2010.
3. Dean J., Rajat M., et al. TensorFlow: Large-scale machine learning on heterogeneous systems. TensorFlow.org. Google Research. Retrieved 10 November 2015.
4. Dean J., Corrado G., et al. Large Scale Distributed Deep Networks. NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada, United States, December, 2012, pp. 1232–1240.
5. Li Deng and Dong Yu (2014), Deep Learning: Methods and Applications, Foundations and Trends in Signal Processing: Vol. 7: no. 3–4, pp. 197–387.
6. Lyashov M.V., Bereza A.N., Alekseenko J.V., Blanco L.M.L. The hybrid reconfigurable system for high-performance computing. Application of Information and Communication Technologies (AICT), 2015 9th International Conference on, Rostov on Don, 2015, pp. 258–262.
7. Shrihari, S.V., Pawan, K., Vishnu, M., Mahajan, S. Technological aspects of grid computing (2015) 2nd International Conference on Electronics and Communication Systems, ICECS 2015, art. no. 7124829, pp. 1463–1468.