

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 519.24

ПРИМЕНЕНИЕ ВЕБ-ТЕХНОЛОГИЙ ДЛЯ ПОСТРОЕНИЯ РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ*

М.М. КОНОНОВ

630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, магистр. E-mail: miklekononov@gmail.ru

В мире существует необходимость решать ресурсоемкие задачи. Чаще всего такие задачи невозможно решить в разумные сроки последовательными алгоритмами. Многие из них можно выполнить только на базе высокопроизводительной техники с помощью методов параллельных вычислений [1–2].

Благодаря постоянному росту пропускной способности сетей передачи данных и удешевлению сетевого оборудования стало возможным создавать дешевые вычислительные кластеры на базе сетей предприятия (также учебных классов).

Идея создания Распределенной вычислительной системы (РВС) на базе серийных ПК существует давно [3]. Коммуникации территориально разрозненных ПК в РВС осуществляются через глобальную сеть интернет. Таким образом, теоретическое увеличение производительности всей системы не имеет верхней границы за счет включения в нее новых ПК.

Вычисления производятся с помощью специального ПО, которое необходимо «инсталлировать» (installation) на ПК. В случае «добровольных вычислений» [4] по ряду причин не все пользователи удаленных ПК могут инсталлировать себе такое ПО.

В статье предлагается альтернативный подход к созданию средств построения РВС из серийных ПК. В качестве промежуточного ПО используется браузер, в который загружается web-приложение с необходимым для вычислений функционалом. Использование такого подхода стало возможным благодаря «эволюции» браузерных технологий. В совокупности это позволит создать высокопроизводительные вычислительные средства за счет более простого способа присоединения новых ПК к «добровольным вычислениям».

Ключевые слова: браузер, сложные задачи, параллельный алгоритм, высокопроизводительные вычислительные средства, javascript, распределенные вычислительные системы, добровольные вычисления, WebRTC

DOI: 10.17212/2307-6879-2016-1-100-109

* Статья получена 21 сентября 2015 г.

ВВЕДЕНИЕ

В наши дни существует необходимость решать сложные задачи посредством машинного вычисления.

Под термином «сложные задачи» обычно понимают проблемы, решение которых требует не только построения сложных математических моделей, но и проведения огромного, на многие порядки превышающего характерные для серийных ЭВМ, количества вычислений (с соответствующими ресурсами ЭВМ: размерами оперативной и внешней памяти, быстродействием линий передачи информации и др.). Такими являются задачи из области прочностного моделирования, математического моделирования физических процессов, задачи криптоанализа, задачи динамики жидкости и газа, электромагнитные и ядерные взаимодействия, а также задачи планирования, расчет сопротивления материалов и др.

Параллельным вычислениям в последнее время уделяется большое внимание. Это связано главным образом с двумя факторами. Первый фактор обусловлен научно-техническим прогрессом, в результате которого появились новые области знаний, требующие применения методов математического моделирования. Сами модели также существенно усложнились. В итоге происходит неуклонное возрастание потребности в ресурсоемких расчетах, которые в ряде случаев можно выполнить только на базе высокопроизводительной техники с помощью методов параллельных вычислений.

На протяжении многих лет производители процессоров постоянно увеличивали тактовую частоту и параллелизм на уровне инструкций до тех пор, пока не столкнулись с физическими ограничениями: размер микросхем и тепловыделение. Дальнейшего увеличения производительности удастся достичь, главным образом, за счет совершенствования техпроцесса (уменьшения норм производства), а также внедрением нескольких вычислительных устройств (ядер) в один микропроцессор. Однако на практике производители серийных процессоров ограничиваются 16 ядрами [5] и менее. Тенденция дальнейшего увеличения количества ядер не делает работу пользовательской ОС быстрее [6].

1. АКТУАЛЬНОСТЬ

Упомянутые трудоемкие задачи решаются методами параллельных вычислений. Задача разбивается на N слабо связанных частей (крупноблочное распараллеливание), каждая из которых вычисляется на отдельном вычислительном устройстве [7]. Идея создания Распределенной вычислительной системы (РВС) на основе «добровольных вычислений» (ДВ) имеет несколько

успешных реализаций [8]. Пользователи добровольно предоставляют вычислительные мощности своих ПК во время простоя (ночью или на перерыве).

Например на платформе BOING [9] много лет функционирует известный проект `siti@home` [10] (поиск сигналов внеземных цивилизаций) и многие другие: поиск лекарств от рака, расчет движения объектов солнечной системы и т. д.

Чтобы предоставить ресурсы своего ПК для ДВ, пользователю необходимо установить и настроить клиентское ПО. По ряду технических причин это возможно не для всех желающих. Дополнительные сложности могут возникнуть, если ПО не совместимо с ОС на ПК пользователя.

Благодаря росту популярности ПК и носимых компьютеров (смартфоны, планшетные компьютеры, умные устройства и т. п.), значительно увеличилось количество серийных моделей и их производителей. Этим обусловлено многообразие аппаратной базы и операционных систем (ОС), что, в свою очередь, создает дополнительные трудозатраты при проектировании и реализации совместимого ПО. Программа должна создаваться для каждой ОС отдельно (возможно, отдельным, профильным разработчиком).

Следует отметить, что существуют программные библиотеки и «фреймворки» (framework) для создания ПО PBC. Хорошим примером является MPI. Это технология обмена сообщениями в PBC. Она имеет разные версии реализации ПО под ОС Windows и Linux, не совместимые между собой.

В качестве примера кросс-платформенной библиотеки подойдет JADE [11], реализованная на языке JAVA. Посредством библиотеки осуществляется обмен сообщениями в распределенных мультиагентных системах. Однако, для запуска агента требуется установить JAVA-машину на ПК пользователя, что также создает дополнительные трудности при добавлении агентов в PBC.

Интернет представляет собой огромную распределенную гетерогенную систему с разрозненными вычислительными ресурсами. Нет технологии, которая позволяет объединять любые ПК в одну PBC, без инсталляции дополнительно ПО.

Предлагается создать web-приложение, которое будет использовать браузер в качестве промежуточного ПО, для выполнения P-алгоритмов.

2. РЕАЛИЗАЦИЯ

Веб-приложение загружается в браузер один раз и работает без перезагрузки веб-страницы.

Загрузка в браузер с сигнального сервера (СС) осуществляется по стандартному для браузера протоколу прикладного уровня HTTP [12]. При этом в

базу данных (БД) СС добавляется IP-адрес ПК, на котором запустили веб-приложение, и его статус (свободен или занят вычислениями).

Обмен служебной информацией с СС (актуальный список свободных агентов и состояние занятых) осуществляется с помощью технологии AJAX [13], которая позволяет взаимодействовать с сервером без перезагрузки web-страницы.

При завершении работы браузера все промежуточные результаты сохраняются в его памяти. Это возможно с помощью новой функции локального хранения данных (local storage) стандарта HTML5 [14]. При восстановлении вкладки с веб-приложением работа агента продолжается.

Коммуникации между агентами осуществляются посредством TCP/IP. При этом обмен данными между агентами происходит по протоколу SCTP [15].

Расширенная версия веб-приложения доступна только администраторам. Ее функционал дополнен возможностью администратора загружать Р-программу и инициировать ее выполнение. За распределение параллельных ветвей программы, удаленное выполнение и сбор полученных данных отвечает системный диспетчер (СД).

Все взаимодействия между агентами осуществляются с помощью обмена сообщениями, но эти механизмы являются прозрачными для разработчика параллельной программы. Программе-инициатору предоставляется API (Application Program Interface) для создания и управления параллельным процессом, а факт его удаленного выполнения скрыт. Взаимодействие с конкретным процессом возможно только по его идентификатору. Любой процесс способен порождать новые процессы, если СД предоставляет для этого ресурсы (свободных агентов).

Все агенты обладают одинаковым функционалом, следовательно, создаваемая РВС относится к классу однородных вычислительных систем (ОРВС) [16].

Благодаря функциональной однородности агенты взаимозаменяемы. При избытке вычислительных ресурсов возможно дублирование процессов на разных агентах для повышения отказоустойчивости. После успешного завершения процесса уничтожается его копия. Если пользователь закроет вкладку браузера с веб-приложением, работа агента будет прервана. В этом случае процесс-копия корректно завершает вычисления, вернет результат родительскому процессу и ОРВС продолжит вычисления.

ОРВС должна обладать высоким уровнем отказоустойчивости. Каждый пользователь должен иметь возможность остановить процесс выполнения агента и закрыть вкладку браузера в любой момент времени. За счет однородности РВС и дублирования процессов можно добиться требуемого уровня отказоустойчивости.

3. ЯЗЫК Р-ПРОГРАММЫ

Агент может инициировать выполнение программы в двух случаях:

- если он работает в расширенном режиме и агент-инициатор загрузил в него параллельную программу;
- если агент получил данные и программу от СД.

Очевидно, чтобы запускать Р-процессы на различных агентах независимо от аппаратных компонентов их ПК и ОС, требуется интерпретируемый язык программирования (ЯП).

Любой современный браузер имеет встроенный интерпретатор ЯП JavaScript. Кроме этого, любую JavaScript-программу можно представить в виде строки символов формата JSON. В этом формате можно представить также любую структуру данных. Таким образом, в передаваемую строку JSON можно поместить данные и инструкции к ним в виде ветви Р-программы.

JSON полностью соответствует синтаксису JavaScript. Принятая удаленным агентом строка в этом формате будет выполнена как понятный браузеру JavaScript-код.

При длительном выполнении JavaScript браузер может зависнуть или прекратить выполнение программы. Устранить эти проблемы можно благодаря технологии фоновых процессов (worker) [17], входящей в HTML5. Worker позволяет запускать параллельные нити JavaScript в фоновом режиме. Для пользователя визуально это не будет заметно.

Установление соединения типа точка–точка (peer-to-peer или p2p) между агентами и обмен данными возможны благодаря технологии WebRTC. Технология является новой и пока поддерживается не всеми браузерами, но стандарт HTML5 уже утвержден консорциумом w3c (World Wide Web Consortium) [18], поэтому со временем с этой технологией будут совместимы все браузеры.

Несмотря на то что JavaScript является интерпретируемым языком, его производительность в движке v8 улучшилась и незначительно уступает некоторым компилируемым языкам [19]. А благодаря поддержке аппаратного ускорения браузер может задействовать вычислительные ресурсы графического процессора ПК.

ЗАКЛЮЧЕНИЕ

Количество используемых ЭВМ в мире растет и уже сейчас превышает 3 миллиарда. Большинство из них имеет высокоскоростное подключение к глобальной сети интернет.

Исходя из анализа имеющихся в современных браузерах технологий можно создать ПО, представляющее собой web-приложение и СС. Выполнение математических операций на JavaScript в браузере GoogleChrome показывает хорошие результаты по сравнению с другими ЯП.

Благодаря использованию данного подхода можно объединять любые ПК в единую ОРВС независимо от их программных и аппаратных составляющих. При применении JavaScript в качестве языка для Р-программы отпадает необходимость создавать набор инструкций и команд управления Р-процессами. Следует добавить, что вычислительная мощность современных носимых ПК не уступает настольным ПК [7]. Такие устройства тоже имеют встроенные браузеры и могут быть привлечены к ДВ.

СПИСОК ЛИТЕРАТУРЫ

1. *Мищенко П.В.* Вопросы построения распределенных вычислительных систем с применением технических и программных средств сетей ЭВМ // Современные проблемы технических наук: тезисы докладов Новосибирской межвузовской научной студенческой конференции «Интеллектуальный потенциал Сибири». – Новосибирск, 2011. – Ч. 1.
2. *Sansom C.* The DNA deluge [Electronic resource] // Scientific Computing World: website. – URL: http://www.scientific-computing.com/features/feature.php?feature_id=168 (accessed: 28.04.2016).
3. *Евреинов Э.В.* Однородные вычислительные системы, структуры и среды. – М.: Радио и связь, 1981. – 208 с.
4. Добровольные вычисления [Электронный ресурс] // Википедия. Свободная энциклопедия. – URL: https://ru.wikipedia.org/wiki/Добровольные_вычисления (дата обращения: 28.04.2016).
5. Intel® Xeon® Processor E7-8890 v3 [Electronic resource] // Intel®: website. – URL: <http://ark.intel.com/products/84685/Intel-Xeon-Processor-E7-889> (accessed: 28.04.2016).
6. Прошлое роста производительности: конец гонки частот, многоядерность и почему прогресс увяз на одном месте [Электронный ресурс] // Хабрахабр: web-сайт. – URL: <https://habrahabr.ru/company/intel/blog/174719/> (дата обращения: 28.04.2016).
7. *Мищенко П.В.* Алгоритмы и структуры управления параллельными процессами в распределенных вычислительных системах: магистер. дис. – Новосибирск, 2012. – 83 с.
8. *Андреев А., Манзюк М., Ватутин Э.* Весь мир как суперкомпьютер / беседа вел С. Попов // Троицкий вариант – наука. – 2012. – № 16 (110). – С. 7.

9. Open-source software for volunteer computing [Электронный ресурс] // BOING: web-сайт. – URL: <http://boinc.berkeley.edu/> (дата обращения: 28.04.2016).
10. Seti@home: web-сайт [Электронный ресурс]. – URL: <http://setiathome.berkeley.edu/> (дата обращения: 28.04.2016).
11. *Bellifemine F., Caire G., Greenwood D.* Developing multi-agent systems with JADE. – Hoboken, NJ: John Wiley & Sons, 2007. – 300 p.
12. Hypertext Transfer Protocol -- HTTP/1.1 [Electronic resource] / R. Fielding, J. Gettys, J.C. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. – URL: <https://tools.ietf.org/html/rfc2616> (accessed: 28.04.2016).
13. *Ostani A.* Learning JavaScript design patterns. – Sebastopol, CA: O'Reilly Media, 2012. – 254 p.
14. *Freeman E., Robson E.* HTML5 programming: building web apps with JavaScript. – Sebastopol, CA: O'Reilly Media, 2011. – 608 p.
15. *Таненбаум Э.* Компьютерные сети. – 4-е изд. – СПб.: Питер, 2003. – 992 с.
16. *Хорошевский В.Г.* Архитектура вычислительных систем. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 512 с.
17. *Green I.* Web workers multithreaded programs in JavaScript. – Beijing: O'Reilly, 2013. – 62 p.
18. HTML5. A vocabulary and associated APIs for HTML and XHTML: W3C Recommendation 28 October 2014 // W3C (The World Wide Web Consortium): website. – URL: <https://www.w3.org/TR/html5/> (accessed: 12.04.2016).
19. The computer language benchmarks game [Electronic resource] // Debian: web-сайт. – URL: <http://benchmarksgame.alioth.debian.org/u64/javascript.php> (дата обращения: 29.04.2016).

Кононов Михаил Михайлович, магистр техники и технологий по специальности «Информатика и вычислительная техника» Новосибирского государственного технического университета. Основное направление исследований – однородные распределительные вычислительные системы. Имеет 2 публикации. E-mail: miklekononov@gmail.com

Using the web technologies to build distributed computing system*

M.M. Kononov

Novosibirsk State Technical University, 20 Karl Marks Avenue, Novosibirsk, 630073, Russian Federation, master. E-mail: miklekononov@gmail.com

In the world there is a need to solve intensive computing. Such problems can not be solved quickly sequential algorithms. A lot of this can be performed using parallel computing techniques only on the basis of high technology [1-2]. With the development of network technology, you can use the local network of businesses (as classrooms) for parallel computing has made possible the creation of cheap computing clusters.

The idea of creating a distributed computing system on the basis of serial PC there is for a long time [3]. Communications geographically disparate PC carried out via the Internet. Thus, the theoretical increase in performance of the entire system does not have the upper limit due to the inclusion of new PCs.

Calculations are made using special software that you need to install on the PC. In the case of "volunteer computing" [4], for a number of reasons, not all users of remote PCs can install such software yourself. The article proposes an alternative approach to the creation of tools for building distributed computing system that consists of the many PC. Browser serves as middleware. Web-application is loaded into the browser. Using such an approach has been made possible thanks to the advent new functions of browser. Together, this will create high-performance computing facilities, due to a simple method of joining many PC to volunteer computing.

Keywords: browser, intensive computing, parallel algorithm, high-performance computing, javaScript, distributed computing, volunteer computing, WebRTC

DOI: 10.17212/2307-6879-2016-1-100-109

REFERENCES

1. Mishchenko P.V. [Questions for building distributed computing systems with the use of hardware and software for computer networks]. *Sovremennye problemy tekhnicheskikh nauk: tezisy докладов Novosibirskoi mezhvuzovskoi nauchnoi studencheskoi konferentsii "Intellektual'nyi potentsial Sibiri"*. Novosibirsk, 2011, pt. 1.
2. Sansom C. The DNA deluge. *Scientific Computing World*: website. Available at: http://www.scientific-computing.com/features/feature.php?feature_id=168 (accessed 28.04.2016)

* Received 21 September 2015.

3. Evreinov E.V. *Odnorodnye vychislitel'nye sistemy, struktury i sredy* [Homogeneous computing systems, environments, and structure]. Moscow, Radio i svyaz' Publ., 1981. 208 p.
4. Dobrovol'nye vychisleniya [Volunteer computing]. *Vikipediya. Svobodnaya entsiklopediya* [Wikipedia. The free encyclopedia]. Available at: https://en.wikipedia.org/wiki/Volunteer_computing (accessed 28.04.2016)
5. Intel® Xeon® Processor E7-8890 v3. *Intel®: website*. Available at: <http://ark.intel.com/products/84685/Intel-Xeon-Processor-E7-889> (accessed 28.04.2016)
6. *Proshloe rosta proizvoditel'nosti: konets gonki chastot, mnogoyadernost' i pochemu progress uvyaz na odnom meste* [Past productivity growth: the end of the race frequency, multi-core and why progress is stuck in one place]. *Khabrakhbr* [Habrahabr]: website. Available at: <https://habrahabr.ru/company/intel/blog/174719/> (accessed 28.04.2016)
7. Mishchenko P.V. *Algoritmy i struktury upravleniya parallel'nymi protsesami v raspredelennykh vychislitel'nykh sistemakh*: magister. diss. [Algorithms and parallel to the process management structure in distributed computing systems: master's diss.]. Novosibirsk, 2012. 83 p.
8. Andreev A., Manzyuk M., Vatutin E., Popov C. (interviewer). *Ves' mir kak superkomp'yuter* [The whole world as a supercomputer]. *Troitskii variant – Nauka*, 2012, no. 16 (110), p. 7.
9. Open-source software for volunteer computing. *BOINC: website*. (In Russian) Available at: <http://boinc.berkeley.edu/> (accessed 28.04.2016)
10. *Seti@home: website*. (In Russian) Available at: <http://setiathome.berkeley.edu/> (accessed 28.04.2016)
11. Bellifemine F., Caire G., Greenwood D. *Developing multi-agent systems with JADE*. Hoboken, NJ, John Wiley & Sons, 2007. 300 p.
12. Fielding R., Gettys J., Mogul J.C., Frystyk H., Masinter L., Leach P., Berners-Lee T. *Hypertext Transfer Protocol -- HTTP/1.1*. Available at: <https://tools.ietf.org/html/rfc2616> (accessed 28.04.2016)
13. Osmani A. *Learning JavaScript design patterns*. Sebastopol, CA, O'Reilly Media, 2012. 254 p.
14. Freeman E., Robson E. *HTML5 programming: building web apps with JavaScript*. Sebastopol, CA, O'Reilly Media, 2011. 608 p.
15. Tanenbaum A.S. *Computer Networks*. 4th ed. Upper Saddle River, NJ, Prentice Hall PTR, 2003. 891 p. (Russ. ed.: Tanenbaum E. *Komp'yuternye seti*. 4th ed. St. Petersburg, Piter Publ., 2003. 992 p.).

16. Khoroshevskii V.G. *Arkhitektura vychislitel'nykh sistem* [Architecture of computer systems]. Moscow, Bauman MSTU Publ., 2005. 512 p.
17. Green I. *Web workers multithreaded programs in JavaScript*. Beijing, O'Reilly, 2013. 62 p.
18. HTML5. A vocabulary and associated APIs for HTML and XHTML: W3C Recommendation 28 October 2014. *W3C (The World Wide Web Consortium)*: website. Available at: URL: <https://www.w3.org/TR/html5/> (accessed 12.04.2016)
19. The computer language benchmarks game. *Debian*: website. Available at: <http://benchmarksgame.alioth.debian.org/u64/javascript.php> (accessed 29.04.2016)