

4. Choi H. Botnet detection by monitoring group activities in DNS traffic / H. Choi, H. Lee, H. Lee, and H. Kim // In proceedings of the 7th IEEE International Conference on Computer and Information Technology. IEEE Computer Society. 2007. – P. 715–720.
5. Savenko O. Review of botnet detection tech-niques. / Savenko O., Lysenko S, Kryshchuk A // Proceedings of the international conference "10 IEEE East-west design and test simposium, 2012. - P.479-482.
6. Savenko O. Botnet detection based on multi-agent approach / Savenko O., Lysenko S, Kryshchuk A // Radioelectronic and computer systems. – 2012. – №5. – P.97-112 (in Ukrainian).
7. Savenko, O., Lysenko, S., Kryschuk, A. Multi-Agent Based Approach of Botnet Detection in Computer Systems / Savenko, O., Lysenko, S., Kryschuk, A. // 19th Conference on Computer Networks, CN 2012, Szczyrk, Poland. CCIS. Springer, Heidelberg 2012. V. 291. P. 171-180.

References

1. Cooke, E. The zombie roundup: Understanding, detecting, and disrupting botnets / E. Cooke, F. Jahanian, and D. McPherson // Proceedings of the USENIX SRUTI Workshop. 2005. – P. 39–44.
2. C. Mazzariello. IRC traffic analysis for botnet detection / C. Mazzariello // Fourth International Conference Information Assurance and Security ISIAS'08. – 2008. – P. 318–323.
3. Akiyama M. A proposal of metrics for botnet de-tection based on its cooperative behavior / M. Akiyama, T. Kawamoto, M. Shimamura, T. Yokoyama, Y. Kadobayashi, and S. Yamaguchi // Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium. 2007. – P. 82–82.
4. Choi H. Botnet detection by monitoring group activities in DNS traffic / H. Choi, H. Lee, H. Lee, and H. Kim // In proceedings of the 7th IEEE International Conference on Computer and Information Technology. IEEE Computer Society. 2007. – P. 715–720.
5. Savenko O. Review of botnet detection tech-niques. / Lysenko S, Kryshchuk A // Proceedings of the international conference "10 IEEE East-west design and test simposium, 2012. - P.479-482.
6. Savenko O. Botnet detection based on multi-agent approach / Lysenko S, Kryshchuk A // Radioelectronic and computer systems. – 2012. – №5. – P.97-112 (in Ukrainian).
7. Savenko, O., Lysenko, S., Kryschuk, A. Multi-Agent Based Approach of Botnet Detection in Computer Systems / Savenko, O., Lysenko, S., Kryschuk, A. // 19th Conference on Computer Networks, CN 2012, Szczyrk, Poland. CCIS. Springer, Heidelberg 2012. V. 291. P. 171-180.

Рецензія/Peer review : 27.9.2013 р. Надрукована/Printed :24.11.2013 р.
Рецензент: Шалапко Ю.І., д.т.н., проф.

УДК 004.75

Е.В. РЫНДИЧ

Черниговский национальный технологический университет

ИСПОЛЬЗОВАНИЕ СОЦИАЛЬНЫХ СЕТЕЙ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЯХ

В статье представлен анализ современных глобальных сетей с целью их использования в распределенных вычислениях. В качестве способа организации взаимодействия вычислительных узлов предложено использовать ресурсы социальных сетей. Широкое распространение социальных сетей и большое количество пользователей позволяет построить распределенную архитектуру с большим количеством установленных доверительных связей.

Ключевые слова: связь, глобальные сети, распределенные вычисления.

Y.V. RYNDYCH

Chernihiv National University of Technology

USING SOCIAL NETWORKING IN DISTRIBUTED COMPUTING

Abstract – The aim of the research – to explore probability of using social networks in distributed computing. Relevance of research caused fact that modern science and technology more and more attention paid to high-performance computing. Such calculations often performed using clusters or supercomputers. These systems can significantly reduce the time to perform calculations by the decomposition of the problem and the parallel execution of several sub-tasks.

The wide spread of social networks and a large number of users allows to build distributed architecture with lots of established trust relationships. Proposed to build a logical network topology using users contacts in a social network. Such approach to computer network has a number of advantages, among which is to provide high data security and user resources, as well as the high stability of the logical topology. Security is provided by private resources, due to the peculiarities of the runtime computing tasks - browsers. The stability provided by the topology of the users who use social networking sites with a certain regularity.

Keywords: communication, wide area networks, distributed computing.

Введение

Актуальность тематики исследования обусловлена тем, что в современной науке и технике все большее внимание уделяется высокопроизводительным вычислениям. Такие вычисления чаще всего выполняют с использованием кластеров или суперкомпьютеров. Эти системы позволяют значительно уменьшить время выполнения расчетов за счет декомпозиции задачи и параллельного выполнения нескольких подзадач. Использование компьютерных сетей связано со стремительным развитием как технических возможностей сети Интернет, развитием веб-технологий, а также вычислительных мощностей персональных компьютеров.

Согласно закону Мура, количество транзисторов, размещаемых на кристалле интегральной схемы, удваивается каждые 24 месяца. А использование новых технологий производства вычислительных элементов позволяет предположить, что при этом их производительность возрастает значительно быстрее. Практика

показывает, что большинство пользователей не используют и половины имеющихся у них ресурсов. Эти факты легли в основу идеи сетей GRID вычислений, таких как, возобновившая свою работу 5-го декабря 2011 года сеть, SETI@home. Эта сеть, посредством предоставляемых пользователями ресурсов, производит анализ радиосигналов с радиотелескопа ATA (Allen Telescope Array) для поиска инопланетных цивилизаций [1].

Параллельно с ростом вычислительных мощностей происходит бурное развитие web-технологий. В наши дни любая сложная система, в той или иной мере, взаимодействует с всемирной сетью, а web-сайты перестали быть простым средством предоставления информации и способны решать множество сложных задач. Часто, коммуникации и обмен данными являются критическим модулем системы, который необычайно важен для становления и развития проекта. Разработчики все время связаны с web-средой, в частности, при обмене данными отдельных компонентов приложений или активном взаимодействии с пользователем грамотное использование web-технологий определяет эффективность, а иногда, и работоспособность всей системы. Поэтому можно утверждать, что достаточно четкие представления об этой области нужны при решении любой современной задачи, как обязательные элементы искусства программирования.

Очевидно, что значительные шаги в развитии web-технологий дают возможность взглянуть по новому на проблемы распределенных вычислений. Использование современных веб-технологий, открывает перед разработчиками множество новых возможностей.

Основной раздел

Основным протоколом Всемирной паутины, распределенной системы, предоставляющей доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к Интернету, является протокол прикладного уровня передачи данных http (рис. 1а). Данный протокол не сохраняет своего статуса между запросами, что подразумевает типичный сеанс связи в режиме «запрос – ответ». Такой классический подход применялся до тех пор, пока не была создана технология AJAX (рис. 1б). Суть данной технологии заключается в «фоновом» обмене данными между браузером и веб-сервером, то есть технология AJAX позволяющая делать запросы к серверу уже после загрузки страницы. Эта технология сейчас очень популярна, а сайты и веб-приложения использующие её, объединяют термином web 2.0[2].

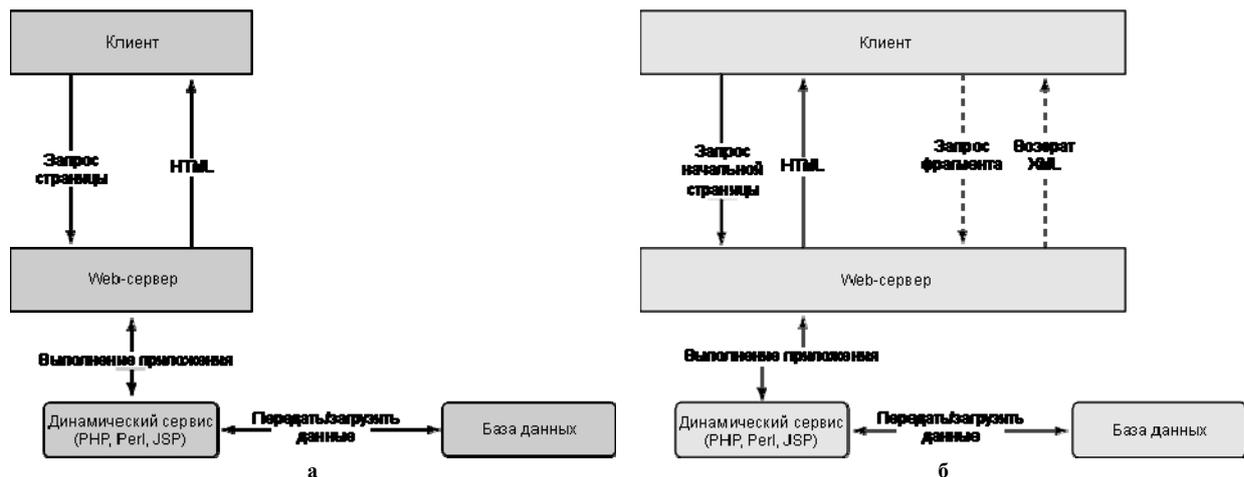


Рис. 1. Схема взаимодействия браузера и сервера

Такой тип взаимодействия называется polling, браузер просто “вытягивает” данные по мере необходимости, без полной перезагрузки страницы. Преимущества перед полным обновлением содержимого очевидны. Эта технология стала первым шагом на пути становления новой эпохи web, она выступила основой для ряда других технологий, таких как:

- Long polling – технология позволяющая делать запросы, которые оставляют соединение открытым, и возвращают данные, как только они были изменены.
- HTTP-streaming – клиент создает всего одно соединение, через которое, при необходимости, получает данные от сервера в реальном времени
- Forever frame – создает iframe, в который сервер постоянно дописывает события по мере их поступлений.
- script tags – динамически создаваемые JS-блоки, что позволяет кросс-доменные коммуникации.
- XMLHttpRequest object – реализует двухстороннее соединение, посредством двух одновременных запросов (один на передачу, другой на прием).

Любой из описанных подходов, в той или иной мере, — обман протокола, когда есть необходимость использовать специально разработанное решение. Рабочая группа HTML5, очевидно, хорошо понимают эту ситуацию и потому в стандарт внесен такой метод как WebSockets. WebSockets — это самое кардинальное расширение протокола HTTP с его появления. Это не дополнение, это сдвиг парадигмы HTTP. Изначально синхронный протокол, построенный по модели «запрос — ответ», становится полностью асинхронным и симметричным. Теперь уже нет клиента и сервера с фиксированными ролями, а есть два равноправных участника обмена данными. Вместе с эффективным средством коммуникаций HTML5 привносит в инструменты разработчика ряд других средств[3]:

- WebWorkers – определяет API для запуска скриптов, в основном JavaScript, в фоновом режиме,

независимо от любых сценариев пользовательского интерфейса. Это позволяет обеспечить работоспособность скриптов производящих длительные вычисления, которые не прерываются скриптами взаимодействующими с пользователем и также позволяет решать ресурсоемкие задачи не нарушая отзывчивость страницы.

- Web Storage и DOM Storage (Document Object Model) – методы веб-приложений, используемые для хранения данных в веб-браузере. Интернет-хранилище поддерживает режим постоянного хранения данных, похожий на cookies, но со значительно расширенной емкостью, и не записывающийся в заголовки HTTP-запросов.

- Web SQL Database – API веб-страницы для хранения данных в базах данных, которые могут быть получены с помощью варианта SQL.

- The Indexed Database API, или Indexed DB – является предложением веб-браузера, стандартный интерфейс для локальной базы данных, для преобразования простых значений в иерархические объекты.

Перечисленные технологии дают возможность по-новому взглянуть на распределенные вычисления. WebSockets дает возможность создавать гибкие и быстрые системы коммуникаций, в обход ограничений протокола. С WebWorkers появляется возможность произвести сложнее вычисления без риска захватить 100% ресурсов процессора. Суммарный объем хранилищ данных в несколько раз превосходит объем требуемый для решения задач высокой сложности, а наличие нескольких хранилищ с различными методами доступа позволяют максимально эффективно использовать ресурсы и строить необычайно гибкие приложения. Объем хранимых данных: 50Мб IndexedDB + 10Мб localStorage + cookies.

Отдельного внимания требует еще одна инновационная технология, разрабатываемая компанией Google и активно развивающаяся в последние годы – это GWT. Google Web Toolkit (GWT) — свободный Java фреймворк, который позволяет веб-разработчикам создавать Ajax приложения на основе Java. Технология GWT делает акцент на повторное использование и кросс-браузерную совместимость. GWT предоставляет разработчику специальные Java API и позволяет разрабатывать компонентно-ориентированные графические пользовательские интерфейсы (GUI), избегая работы с JavaScript и абстрагируясь от протокола HTTP и DOM моделей, лежащих в основе браузеров. Все это достигается с использованием GWT компилятора. GWT компилятор берет клиентский Java код и генерирует JavaScript. Скомпилированная клиентская часть GWT приложения состоит из фрагментов HTML, XML и JavaScript. Google Web Toolkit позволяет создавать целостные, высокоэффективные, безопасные и в тоже время необычайно гибкие системы, используя все достоинства ООП и языка программирования Java. Строгая иерархия и следования принципам ООП делает эту технологию идеальной для построения систем выполняющих сложные вычисления. А поскольку, как код клиентской части, так и код, выполняемый на сервере, написан на языке java, он может в полной мере включать все преимущества java и использовать все представленные в языке средства. К тому же весь код может быть с легкостью покрыт любыми доступными в java тестами. Возможность использования систем тестирования и гибких методик разработки может стать критическим пунктом при разработке сложной системы. Перечисленные технологии дают возможность создать полноценную сеть распределенных вычислений.

Используя описанные технологии возможно осуществить распределенные вычисления в веб. Примером вычислительной задачи является распределенная система подбора пароля при знании его хэша, полученного с помощью алгоритма md5 [3]. Оптимальной архитектурой для решения данного класса задач является архитектура «клиент-сервер», поскольку задачу однозначно можно распределить на независимые равнозначные подзадачи одинакового характера.

Однако для решения более сложных задач, которые требуют взаимодействия клиентов между собой (примером такой топологии есть связанный граф), необходимо устанавливать связи между приложениями-клиентами, то есть между пользователями у которых нет доверительных взаимоотношений. Для этого необходимо иметь структуру сети взаимодействия, в которой клиенты не смогут привести к нарушению политики безопасности и принципиально будут иметь только санкционированный доступ к ресурсам конечного пользователя.

Такую структуру сети могут нам предоставить социальные сети, где уже есть установленные доверительные связи между пользователями системы. Каждая современная социальная сеть поддерживает механизм создания и использования специальных программных многопользовательских приложений, которым предоставляется доступ к средствам социальной сети. При этом безопасность пользователя и персонального компьютера контролируется социальной сетью.

Социальная сеть имеет как уже сформированные связи, которые упрощают процесс создания сети, а также позволяют четко определенным безопасным способом создавать временные связи (на рис. 3 показано пунктиром) между несвязанными пользователями. При этом структура связей не нарушается.

Важным аспектом использования социальных сетей в качестве сети взаимодействия является популярность таких сервисов среди пользователей. Наиболее известной социальной сетью является Facebook, количество активных пользователей на сентябрь 2012 года которой составляет более 1 млрд пользователей [4]. Ежемесячно среднестатистический пользователь проводит около 7 часов социальных сетях, но более 60% пользователей пользуются социальными сетями каждый день и ежедневно проводят в них более 3 часов. Каждый пользователь имеет хотя бы 1 ядро процессора производительностью не менее 8 Гфлопс, то производительность такой сети эквивалентна $1 \cdot 10^{18}$ флопс, что значительно больше производительности существующих суперкомпьютеров.

В качестве практического эксперимента была реализована задача, описанная в [3]. Задача, получаемая клиентом, представлена в виде класса PasswordBruteTask с заданными границами возможных паролей. Как видно на диаграмма PasswordBruteTask наследует класс Task и переопределяет абстрактный метод runTask. Такой подход упрощает процесс создания новых заданий, для решения широкого круга задач.

```

@Override
public void runTask() {
    if (i++ < intervalEnd) {
        try {
            MessageDigest messageDigest =
MessageDigest.getInstance("MD5");
            messageDigest.update(String.valueOf(i).getBytes());
            byte[] message = messageDigest.digest();
            StringBuffer hash = new StringBuffer();
            for (int j = 0; j < message.length; j++) {
                String hex = Integer.toHexString(0xFF & message[j]);
                if(hex.length() == 1) {
                    hash.append('0');
                }
                hash.append(hex);
            }
            if (hash.toString().equals(taskHash)) {
                setResult(String.valueOf(i));
                setFinished(new Date());
            }
        } catch (NoSuchAlgorithmException e) {
            showException(e);
        }
    } else {
        setFinished(new Date());
    }
}

```

Листинг 1. Упрощенный метод runTask переопределенный в классе PasswordBruteTask.

Метод, представленный в листинге, производит подбор пароля посредством проверки совпадения с известным MD5 хэшем. Если пароль найден или достигнут конец интервала выполнение задание прекращается. Представленный подход дает возможность создания модуля добавления заданий написанных на языках отличных от java, но легко интегрируемых в java среду, таких как JavaScript и Groovy. На рис.2 представлен интерфейс выбора задания.

Web Methods Project

New Task		Refresh					
4	698d51a19d8a121ce581499d7b701668	owner	test task	2011-11-04 16:33:07.412000000	Not completed	No result	
5	698d51a19d8a121ce581499d7b701668	owner	test task	2011-11-09 11:28:04.635000000	Not completed	No result	
6	698d51a19d8a121ce581499d7b701668	owner	test task	2011-11-09 11:28:37.861000000	Not completed	No result	
7	698d51a19d8a121ce581499d7b701668	Owner Name	Description	2011-11-19 21:39:26.056000000	Not completed	No result	

Рис. 2. Интерфейс выбора задания

Для доказательства работоспособности теории сети распределенных вычислений, в основе которой лежат web приложения, было принято решение построить сеть производящую распределенный подбор пароля на основании MD5 хэша (для упрощения системы, описан пример выполняющий подбор только цифровых паролей). В качестве основного хранилища данных была выбрана база данных PostgreSQL. В результате проектирования была получена архитектура системы представленная диаграммой зависимостей классов на рис. 3.

Коммуникации между серверной и клиентской частями приложения реализует GWT, посредством асинхронных вызовов.

Сейчас число пользователей ПК и Интернет растет, растет и число ядер. Безусловно, число 16 эксафлопс – идеальное, никто не будет производить вычисления круглые сутки, но если бы каждый персональный компьютер работал над вычислениями хотя бы 1 минуты в день, то такая сеть сравнится с IBM Sequoia [3].

Проведенные опрос пользователей социальной сети «ВКонтакте» показал, что больше половины пользователей готовы безвозмездно участвовать в распределенных вычислениях с использованием социальной сети. Следовательно по результатам исследования, применяя данный подход, социальная сеть «ВКонтакте» по производительности распределенных вычислений превзошла Sequoia. А стремительное развитие веб-технологий уже сейчас позволяет веб-приложениям по своей функциональности приблизиться к настольному программному обеспечению.

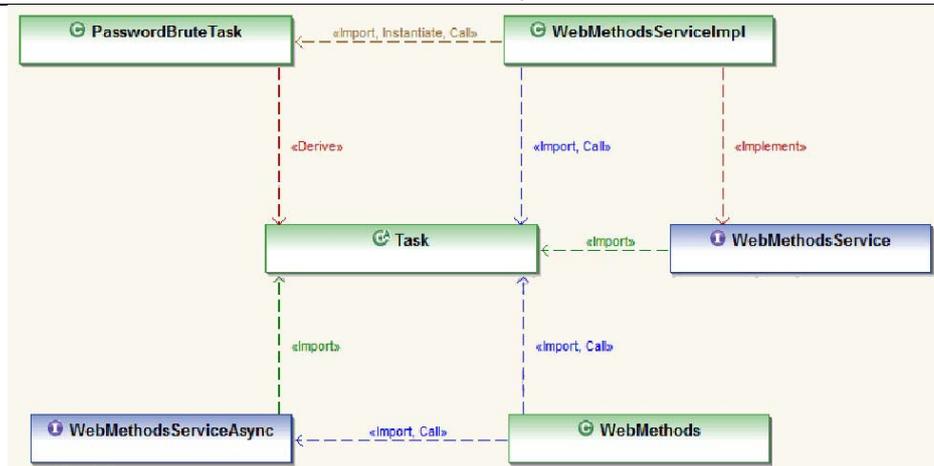


Рис. 3. Диаграмма зависимостей классов

Выводы

Анализ показал, что производительность современных персональных компьютеров и развитие глобальных сетей позволяет использовать их для решения современных вычислительных задач, характерных для суперкомпьютеров. В качестве физических коммутационных сетей предложено использовать глобальную сеть Интернет, а в качестве вычислительных узлов – персональные компьютеры. При этом предложено строить логическую топологию сети используя контакты пользователей в социальной сети. Такой подход к организации вычислительной сети имеет ряд преимуществ, среди которых стоит выделить высокую безопасность данных и ресурсов пользователей, а также высокую стабильность логической топологии. Безопасность частных ресурсов обеспечивается, благодаря особенностям среды выполнения вычислительных задач – браузерам. Стабильность топологии обеспечивается самими пользователями, которые пользуются социальными сетями с определенным постоянством [5].

Литература

1. Что такое SETI@home? [Электронный ресурс]. – Режим доступа: <http://setiathome.berkeley.edu/>
2. Ajax: A New Approach to Web Applications [Электронный ресурс]. – Режим доступа: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>
3. Распределенные вычисления на JavaScript: Сегодня [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/119692/>
4. Список социальных сетей с более 100 миллионов пользователей [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Список_социальных_сетей_с_более_100_миллионов_пользователей
4. Сколько времени люди проводят в социальных сетях? [Электронный ресурс]. – Режим доступа: <http://36on.ru/news/people/2108>

References

1. "Shto takoe SETI@home?" (Last day visited May 07, 2013) <http://setiathome.berkeley.edu/>
2. "Ajax: A New Approach to Web Applications" (Last day visited April 17, 2013) <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>
3. "Raspredelennye vychysleniya JavaScript: Segodnya" (Last day visited April 17, 2013) <http://habrahabr.ru/post/119692/>
4. "Spisok socialnih setei s bolee 100 millionov polzovatelei" (Last day visited May 07, 2013) http://ru.wikipedia.org/wiki/Список_социальных_сетей_с_более_100_миллионов_пользователей
5. "Skolko vremeni ludi provodyat v socialnih setyah?" (Last day visited April 17, 2013) <http://36on.ru/news/people/2108>

Рецензія/Peer review : 5.8.2013 р. Надрукована/Printed :24.11.2013 р.
 Рецензент: Кувшинов А.В., Заместитель начальника ВИТИ НТУУ «КПИ», д.т.н., профессор, Военный институт телекоммуникаций и информатизации Государственного университета телекоммуникаций,