

МЕТОД МАРШРУТИЗАЦИИ В ОБЛАЧНОЙ РАСПРЕДЕЛЕННОЙ СРЕДЕ НА ОСНОВЕ МОБИЛЬНЫХ ОДНОРАНГОВЫХ СЕТЕЙ

К.С. Ткаченко, Н.Л. Корепанова

Севастопольский национальный
технический университет
г. Севастополь, ул. Университетская, 33
E-mail: kvt@sevgtu.sebastopol.ua

В статье предлагается метод маршрутизации в облачной распределенной среде на основе мобильных одноранговых сетей.

Введение. Постановка проблемы в общем виде и ее связь с важными научными и практическими задачами. В настоящее время существуют различные виды атак, как на локальные сети, так и на реализованные на их основе средства облачных вычислений [1 – 3]. Мобильные одноранговые сети предоставляют мобильные узлы для формирования самоорганизующейся сети без необходимости наличия постоянной инфраструктуры. Такие сети могут быть использованы при защите инфраструктуры приложений при стихийных бедствиях или в удаленной местности, где не может быть проложена фиксированная проводная сеть. При этом они ограничены не только своей собственной пропускной способностью, но и сроком службы батарей. По этой причине, одним из основных условий существования подобной связи является наличие эффективных маршрутов между ее узлами, причем маршруты должны динамически адаптироваться к изменению топологии. Это делает сеть уязвимой к таким атакам, как подслушивание, отказ в обслуживании. С другой стороны в архитектуре облачных вычислений существуют разнообразные службы, такие как брокеры, которые ответственны по предоставлению услуг конечным пользователям. Эффективные системы управления связью, обеспечивающие безопасность связи в облаке, в них отсутствуют.

Анализ последних исследований и публикаций. Зарубежными авторами [1 – 3] предложены различные решения данной проблемы. Существуют методы

генерирования стеганографических ссылок главным узлом кластера, позволяющие выполнять обмен таблицами маршрутизации между узлами кластера, что позволяет динамически назначать новые главные узлы. Предложены алгоритмы маршрутизации, позволяющие выполнять двухуровневую аутентификацию с использованием ключевых пар (открытые и закрытые ключи) для сетевых протоколов.

Проблема. Нерешенными прежде частями общей проблемы, которым посвящена статья, являются отсутствие в алгоритмах и методах из работ [1 – 3] учета необходимости наличия высокой производительности в структурах облачных распределенных сред. Целью данной статьи является формулировка метода маршрутизации в облачной распределенной среде, построенной на основе мобильных одноранговых сетей.

Гетерогенная мобильная одноранговая сеть. К гетерогенной мобильной одноранговой сети, а также работающему поверх нее программному обеспечению, необходимо предъявить требования, аналогичные требованиям к облачной распределенной среде [4].

При решении данной задачи требуется регулирование совместного использования среды передачи данных, которая определяется на уровне доступа к среде передачи данных. Этот уровень называют MAC-уровнем (Media Access Control), на котором устанавливаются правила совместного использования среды передачи данных одновременно несколькими узлами гетерогенной сети. При этом определены типы архитектуры – Ad Hoc и Infrastructure Mode. В режиме Ad Hoc (точка-точка) станции непосредственно взаимодействуют друг с другом, что обеспечивает минимальные издержки оборудования. Каждая рабочая станция оснащается беспроводным адаптером. При этом не требуется дополнительных расходов на создание сетевой инфраструктуры. Но диапазон действия сети становится ограничен.

В режиме Infrastructure Mode станции взаимодействуют друг с другом не напрямую, а через точку доступа – концентратор. Имеется два режима взаимодействия с точками доступа – BSS

(Basic Service Set) и ESS (Extended Service Set). Известно, что при функционировании концентратора в режиме BSS связь между всеми станциями осуществляется только через точку доступа – мост к внешней сети. Для рассматриваемых в статье задач требуется использовать возможности расширенного режима ESS, когда существует инфраструктура нескольких сетей BSS, а сами точки доступа взаимодействуют друг с другом, что позволяет передавать трафик между ними с помощью как сегментов кабельной сети, так и радиомостов.

Имитационная модель и балансировка нагрузки. Можно выделить следующие преимущества переноса высоконагруженных сайтов в Windows Azure [4]. Ферма веб-серверов на базе веб-роли Windows Azure может динамически реагировать на изменяющуюся нагрузку с тем, чтобы предотвратить отказы в обслуживании и повысить качество работы сервиса. При этом не требуется капитальных инвестиций в резервирование оборудования. Высокомасштабируемое хранилище позволит существенно снизить стоимость хранения данных, обеспечить потенциал будущего роста без деградации качества предоставляемого сервиса, повысить надежность и способность к восстановлению после сбоев. Облачные системы баз данных спроектированы таким образом, чтобы линейно масштабироваться вместе с приложением. Прикладная роль Windows Azure помогает реализовать сколь угодно производительный набор сервисов для фоновой асинхронной обработки данных. Веб-фермы, которым требуется быстрое масштабирование, могут быть перенесены в облако, однако это может потребовать соответствующего изменения архитектуры, как представлено на рисунке 2. К изменениям можно отнести: ферма веб-серверов и обработчики очереди реализованы как набор экземпляров веб-роли и прикладной роли соответственно; компоненты обработки данных используют программный интерфейс, предоставляемый Windows Azure для доступа к облачным системам баз данных (Azure Storage Queue, Table, BLOB); статичное содержимое сайта переносится в базу данных.

Анализ результатов имитационного моделирования говорит о том, что вероятность безотказной работы системы находится на интервале [0.73; 0.74].

В настоящий момент имеются методы балансировки нагрузки в фермах веб-серверов, основанные на близости IP-адресов (IP Affinity). Microsoft Forefront ISA Server (Forefront) поддерживает два варианта для балансировки нагрузки веб-публикации фермы [4]. Первый метод основан на сессиях, второй метод основан на близости адресов клиентов. Недостатком второго метода является его полная неэффективность, когда многие удаленные клиенты находятся за устройством NAT (маршрутизатор либо межсетевой экран, предоставляющий возможность преобразования IP-адресов транзитных пакетов путем замены адреса источника при прохождении пакета в одну сторону и обратной замене адреса назначения в ответном пакете), поскольку все они будут направляться на один и тот же сервер. Для упрощения реализации IP Affinity возможно использовать следующий алгоритм. Целью функционирования алгоритма IP Affinity является то, что весь трафик от конкретного клиента всегда будет направлен к одному конкретному серверу в ферме, до тех пор, пока этот сервер находится в состоянии подключения к Интернет. Когда этот сервер перейдет в состояние неподключения к Интернет, то весь трафик от его потенциальных клиентов будет распределен в равной степени между подключенными серверами в ферме. Внутреннее состояние фермы можно описать с помощью двух векторов. Первый вектор содержит ссылки на все опубликованные серверы в ферме. Второй вектор содержит ссылки только на серверах, которые в настоящее время подключены к Интернет.

В этом алгоритме используется альтернативный индекс [4], который рассчитывается по формуле:

$$AI = \text{floor}\left(\frac{IP}{OpC}\right)\%OnC, \quad (1)$$

где AI – альтернативный индекс сервера; floor – наименьшее целое снизу; IP – 32-х битное слово-представление IPv4-адреса маски веб-фермы; OpC – количе-

ство опубликованных (подключенных к веб-ферме) серверов; ОнС – количество «онлайн» (подключенных к Интернет) серверов; «%» – вычисление остатка от деления.

Распределение трафика происходит в две фазы.

Фаза № 1. Клиент подает запрос в ISA Server. Алгоритм балансировки нагрузки рассчитывает индекс сервера веб-фермы, который будет выступать в качестве сервера для заданного клиента, как IP-модулю числа серверов в ферме, где И – индекс сервера, IP – 32-х битное слово-представление IPv4-адреса маски веб-фермы, ОпС – количество опубликованных (подключенных к веб-ферме) серверов. Если сервер с расчетным индексом находится в режиме «онлайн», то ссылка на этот сервер возвращается. Это обеспечивает справедливое распределение последовательных IP-адресов из одного диапазона для опубликованных серверов, и сопоставление серверов к клиентам (если серверы подключены к сети).

Фаза № 2. Если сервер с расчетным индексом находится в режиме «офлайн», то алгоритм переходит в фазу № 2. Для выбора сервера из вектора «онлайн»-серверов вычисляется альтернативный индекс по формуле (1), который указывает на сервер из вектора «онлайн» серверов.

Когда сервер переходит в автономный режим, то весь назначенный ему трафик распределяется поровну между другими серверами в ферме в соответствии с формулой (1).

Параллельная обработка больших массивов данных. Рассматривается типичное использование облачной платформы – параллельная обработка больших массивов данных [4]. В данном случае «толстый» клиент (обеспечивающий в противовес «тонкому» клиенту расширенную функциональность независимо от центрального сервера) обращается к веб-сервису. Прикладной код получает входные данные и помещает запрос на обработку в очередь сообщений. Далее выполнение приложения происходит в асинхронном режиме. Параллельно работающие прикладные роли (наборы функций для выполнения определённого круга задач, назначаемых серверу), оп-

рашивают очередь сообщений и получают запрос на обработку данных. Обработав данные и поместив результат в Azure Table, прикладные роли продолжают опрос очереди сообщений на предмет появления новых заданий.

Можно выделить следующие подсистемы [4]:

1. Толстый клиент – подсистема, предназначенная для получения данных от пользователя, их обработки как самостоятельно, так и с использованием «облака», выдачи данных пользователю.

2. Веб-сервер – подсистема, предназначенная для выполнения буферизации между толстым клиентом и остальной частью среды.

3. Windows Communication Foundation (WCF) – подсистема, предназначенная для обеспечения реализации веб-роли.

4. Прикладной код – подсистема, предназначенная для выполнения хранимых процедур бизнес-логики.

5. Azure Storage – подсистема, предназначенная для обработки нереляционных данных.

6. Фоновый обработчик – подсистема, предназначенная для обработки в фоне части данных, не обработанных прикладным кодом.

Среда рассматривается с позиций теории массового обслуживания (ТМО) как система массового обслуживания (СМО) и общих принципов моделирования систем. Критериями оценки эффективности среды являются максимум пиковой производительности и минимум совокупной стоимости владения.

В результате машинных экспериментов получено, что в Парето-эффективное множество входят решения с предельными значениями производительностей и предельными значениями емкости буферов. Поэтому в зависимости от задач можно предложить ЛПР выбор: Если облако используется для управления критическим объектом, то для обеспечения эффективной загрузки процессоров и стоимости рекомендуется использовать вариант, где имеет место высокая производительность узлов, иначе, для учебных приложений, где имеет место высокая загрузка буферов и ресурсов.

Предлагаемый алгоритм. Описание шагов алгоритма приводится ниже.

Вначале происходит ожидание получения от главного сервера кластера (одновременно являющегося брокером) запроса на получение таблиц маршрутизации. Затем происходит обновление таблиц маршрутизации. После чего начинается прослушивание в сети, как локальной проводной, так и беспроводной.

Затем, в бесконечном цикле, повторяются следующие этапы алгоритма:

1. Если время_прохождения_запроса + значение_ГСЧ > максимальное_значение_времени_прохождения_запроса, то переслать пакет с использованием резервных нешифрованных каналов.

2. Если время_обновления_таблиц_маршрутизации + значение_ГСЧ > максимальное_значение_времени_обновления_таблиц_маршрутизации, то послать запрос на обновление таблиц маршрутизации с использованием резервных нешифрованных каналов.

3. Если время_подтверждения_передачи + значение_ГСЧ > максимальное_значение_времени_подтверждения_передачи, то начать прослушивание от соседей кластера.

4. Если получены входные данные, то: а) если найдено стеганографическое сообщение в пакете, то: найденный адрес, резервные адреса присвоить расшифровать (полученные входные данные). Если были добавлены новые каналы, то обновить таблицы маршрутизации, послать запрос на обновление таблиц головного кластера; б) Перейти к обработке следующего запроса

5. Если имеется запрос на обновление таблиц маршрутизации и найдены изменения в таблицах маршрутизации, то: таблица маршрутизации присвоить обновить таблицы; послать кластеру запрос на обновление

6. Если запрос приветствие, то: таблица маршрутизации присвоить добавить(обновить) таблицы, получить таблицы у клиента)

7. Для каждого соседа: получить_таблицы_маршрутизации; послать_запрос_на_обновление

Видно, что принципиальным отличием данного алгоритма от представленных в [1 – 3] является ориентация ядра системы на брокер, который находится вне распределенной среды.

Заключение. В данной работе сформулирован метод маршрутизации в облачной распределенной среде, которая построена на основе мобильных односторонних сетей. Перспективами дальнейших изысканий по данной проблеме станет дальнейшее совершенствование метода.

СПИСОК ЛИТЕРАТУРЫ

1. *Vinayakray-Jani P.* Security Architecture for Cluster based Ad Hoc Networks / Preetida Vinayakray-Jani, Sugata Sanyal // DA-IICT, Gandhinanagar, India. – 5 p. – <http://arxiv.org/pdf/1207.1701>
2. *Raj G.* Secure Cloud Communication for Effective Cost Management System through MSBE / Gaurav Raj, Kamaljit Kaur // Lovely Professional University, Phagwara, India. – 12 p. – <http://arxiv.org/pdf/1207.2706>
3. *Ткаченко К.С.* Ферма веб-серверов на базе облачных вычислений в распределенных средах / К.С. Ткаченко, Н.Л. Корепанова // Збірник наукових праць Академії військово-морських сил імені П.С.Нахімова. Випуск 2(10). – Севастополь: АВМС України імені П.С. Нахімова, 2012. – Вип. 2(10). – С. 142 – 147.
4. *Ткаченко К.С.* Повышение эффективности использования распределенной среды на базе облачных вычислений / К.С. Ткаченко, Н.Л. Корепанова // Збірник наукових праць Академії військово-морських сил імені П.С.Нахімова. Випуск 2(10). – Севастополь: АВМС України імені П.С. Нахімова, 2011. – Вип. 3(7). – С. 123 – 126.