УДК 004.75

Ю. А. Кулаков, Е. Ю. Лопушен

СПОСОБ ДИНАМИЧЕСКОЙ БАЛАНСИРОВКИ НАГРУЗКИ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

Аннотация: Программно-конфигурируемые сети позволяют значительно расширить функциональность сетевого оборудования. Целью работы является разработка способа динамической балансировки нагрузки в распределённом приложении, которое выполняется на кластере. Балансировка нагрузки выполняется на контроллере SDN. Предложенный способ позволяет рационально использовать ресурсы кластера, а также обеспечивает устойчивость приложения к высоким нагрузкам.

Ключевые слова: программно-конфигурируемые сети, SDN-контроллер, балансировка нагрузки, гибридное облако, mininet.

Введение

Современным web-приложениям необходимо выдерживать колебания нагрузки, которые могут быть вызваны изменяющимся количеством пользователей. Для увеличения количества пользователей, которых система может обслуживать в определённый момент времени, необходимо произвести масштабирование. Вертикальное: путём наращивания мощности сервера, или горизонтальное: путём поддержки нескольких экземпляров приложения и балансировкой нагрузки между ними. Средства программно-конфигурируемых сетей позволяют реализовать способ динамической балансировки нагрузки на контроллере SDN.

Обзор существующих решений

В [1] и [2] рассмотрены возможности программно-конфигурируемых сетей в контексте распределения нагрузки на узлы. Однако, не предусмотрена оптимизация использования вычислительных ресурсов в условиях изменяющейся нагрузки. Также, в случае нехватки ресурсов возможна ситуация, в которой приложению не хватит ресурсов для обработки всех запросов, поступающих от пользователей. Это может привести к перебоям в работе или полному выходу сервиса из строя. Предлагаемый способ решает эту проблему путём подключения лишь необходимого количества узлов, и, при необходимости, подключения облачных ресурсов.

Постановка задачи

Для организации эффективной работы распределённого приложения в условиях изменяющейся нагрузки необходимо рационально использовать ресурсы кла-

стера и иметь возможность подключать дополнительные вычислительные мощности, в случае, если ресурсов локального кластера недостаточно. В [3] и [4] рассмотрено применение гибридного облака как резервного ресурса, подключаемого в случае недостатка локальных ресурсов. Возможности программно-конфигурируемых сетей позволяют реализовать способ балансировки нагрузки в распределённом приложении, который обладает следующими преимуществами: подключение дополнительных ресурсов при необходимости и оптимальное использование ресурсов кластера, как следствие - уменьшение затрат на вычислительные ресурсы.

Решение поставленной задачи

Для динамической балансировки нагрузки контроллер программно-конфигурируемой сети должен располагать данными о загрузке системы. Для получения этих данных используется агент мониторинга ресурсов - программное обеспечение, позволяющее контроллеру отслеживать загрузку каждого узла. В агенте настраивается расчёт коэффициента нагрузки в зависимости от задачи: для разных задач более приоритетными являются разные ресурсы. Поскольку в программно-конфигурируемых сетях контроллер оснащён REST-интерфейсом и возможностью подключения плагинов, в роли агента мониторинга ресурсов может выступать простейшее программное обеспечение, разработанное на произвольном языке программирования и поддерживающее тот же интерфейс, что и контроллер.

В данной работе рассматривается кластер, узлы которого находятся в одной физической сети, а узлы задействованные в работе приложения находятся в программно-конфигурируемой сети (SDN). Масштабирование приложение происходит путём расширения SDN. Приложение автоматически определяет новый сервер в SDN и задействует его в обслуживании запросов. Данный способ масштабирования приложения рассматривается в [5].

В случае, если все доступные узлы кластера заняты, в качестве резервного ресурса подключается узел в облаке. Поставщики облачных вычислительных ресурсов располагают интерфейсом для программного конфигурирования серверов, что даёт возможность контроллеру подключать и перенаправлять туда запросы. После снижения нагрузки контроллер отключает облако, запросы обслуживаются только кластером.

Балансировка нагрузки между узлами в локальном кластере происходит за счёт изменения таблицы маршрутизации в контролере. Создание узлов в облаке осуществляется автоматически через API поставщика облачных ресурсов (Amazon, Azure). Перенаправление запросов в облако с контролера осуществля-

ется через REST-интерфейс, также можно использовать любую систему обмена сообщениями (RabbitMQ, JMS, Amazon SQS), для которой реализована поддержка языка программирования, на котором реализована логика контроллера.

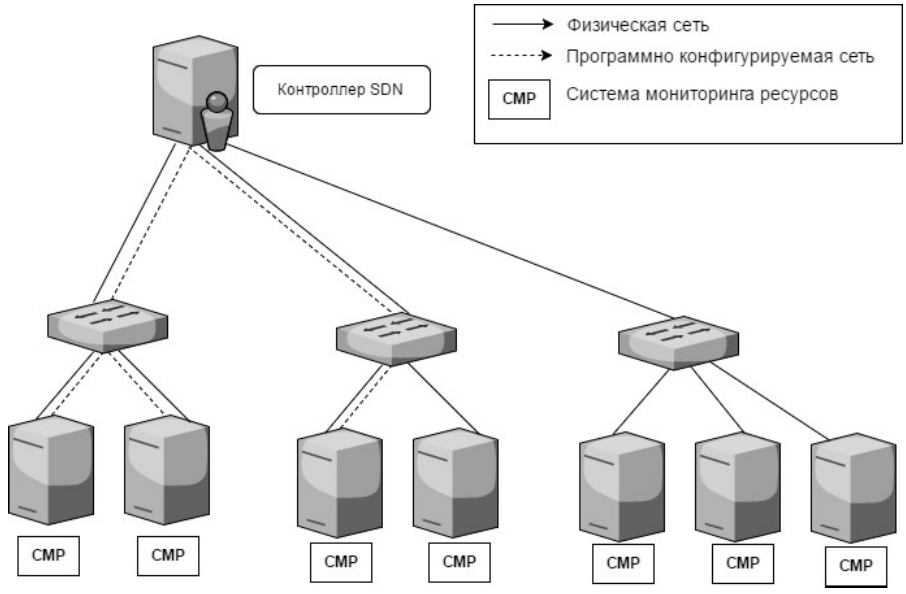

Рис. 1. - Конфигурация кластера для распределённого приложения

Для принятия решения о масштабировании кластера, система должна располагать информацией об использовании ресурсов уже работающих узлов. Критерий оценки зависит от приложения: приложение может использовать много оперативной памяти, но не сохранять файлы на дисковое пространство, либо служить хранилищем данных, где ключевой характеристикой является объём доступного дискового пространства. Разработка сложных критериев загруженности сети может являться темой для отдельного исследования. Для примера примем за критерий количество доступной оперативной памяти.

Построим топологию сети, указанную на рисунке 1 в эмуляторе mininet. Зададим наличие выделенной ОЗУ для приложения на каждом из серверов.

Таблица 1

Объём выделяемой ОЗУ для каждого из узлов кластера

№ узла	***1***	***2***	***3***	***4***	***5***	***6***	***7***
Объём выделенной ОЗУ, Mb	700	***800***	***1400***	***500***	***1500***	***2300***	***1200***

Для эксперимента возьмём отрезок времени длительностью 1000 секунд. Каждый узел способен обрабатывать данные объёмом до 500 Mb/s, поэтому, контроллер должен предотвратить переполнение ОЗУ каждого из узлов. За отведенный промежуток времени нагрузка на приложение изменяется как показано на рис. 2.

Рис. 2. - График зависимости поступления трафика на приложения от времени

Для каждого узла справедливо условие $LC_i = k \times C_i$, где C_i – ОЗУ на каждом i-ом узле. LC (Limit Capacity) – граничный объём задействованной ОЗУ, в случае, если меньший объём занят обработкой запросов, подача новых запросов на узел контроллером не производится. Таким образом, после обработки данных, которые уже находятся на узле, сервер может быть отключён от программно-конфигурируемой сети за ненадобностью. Такой подход позволяет рационально использовать ресурсы кластера, не включая лишние узлы, максимально используя уже подключенные.

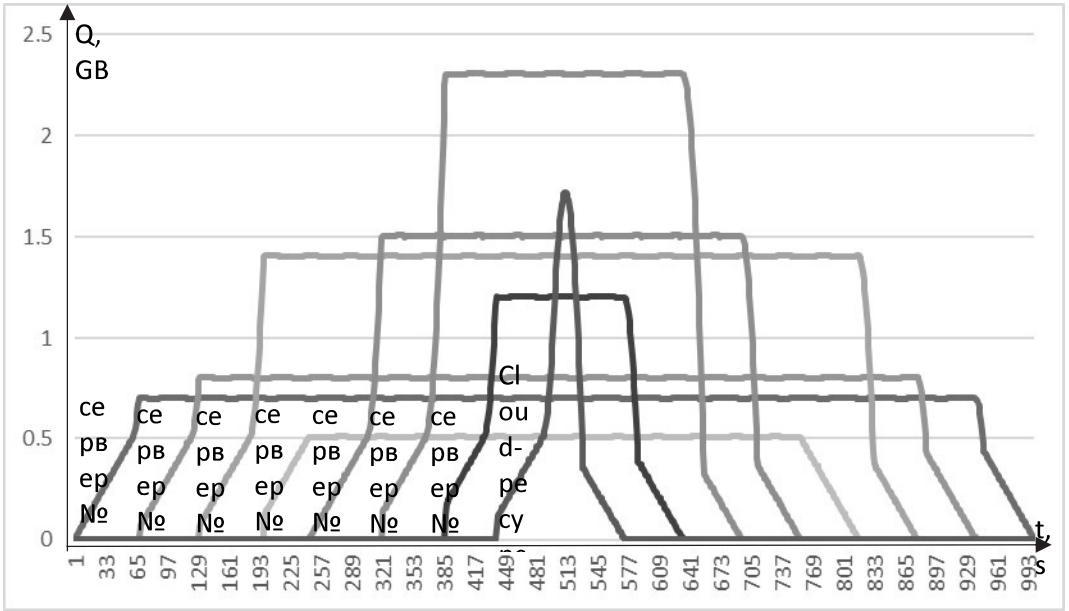

Рис. 3. - График зависимости поступления трафика на каждый из узлов от времени

 ISSN 1560-8956

Контроллер SDN осуществляет запрос на системы мониторинга ресурсов на каждом из задействованных узлов каждую секунду, на основании полученных данных принимает решение о реконфигурации сети.

Как можно увидеть на графике зависимости поступления трафика на каждый из узлов от времени, новые узлы подключаются в тот момент, когда на уже подключенные узлы поступает пиковая нагрузка, которая продолжает расти (см. рис 2). Когда все сервера максимально загружены, подключается ресурс в облаке. В данной эмуляции cloud-ресурс не ограничен, так как для большинства задач возможно подключить столько ресурсов в облаке, сколько необходимо.

Выводы и дальнейшая работа

В данной статье предложен способ балансировки нагрузки в программно-конфигурируемых сетях. Предложенный способ позволяет, не используя стороннее программное обеспечение реализовать балансировку нагрузки средствами SDN. Способ позволяет настраивать характеристики узлов, отдавая приоритет наиболее важным для конкретной задачи. Проведено тестирование способа на эмуляторе mininet. Эмуляция показала, что способ работоспособен и обладает достаточной гибкостью для работы в реальных условиях изменяющейся нагрузки. В качестве дальнейшей работы, возможно усовершенствование способа путём добавления механизма прогнозирования нагрузки.

Список использованных источников

1. Towards an Elastic Distributed SDN Controller / [A. Dixit, H. Fang, M. Sarit та ін.] // ACM SIGCOMM Computer Communication Review. — 2013. — № 43 (4). — С.7-12.

2. Belyaev M. Towards load balancing in SDN-networks during DDoS-attacks / M. Belyaev, S. Gaivoronski // Science and Technology Conference (Modern Networking Technologies) (MoNeTeC). — 2014. — № 1. — С.1-6.

3. Intelligent workload factoring for a hybrid cloud computing model / [H. Zhang, G. Jiang, K. Yoshihira та ін.] // SERVICES ‘09 Proceedings of the 2009 Congress on Services. — 2009. — № 1. — С.701-708.

4. Van den Bossche R. Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads / R. Van den Bossche, K. Vanmechelen, J. Broeckhove // 2010 IEEE 3rd International Conference on Cloud Computing. — 2010. — № 3. — С.228-235.

5. Кулаков Ю. А. Способ масштабирования распределённых приложений в программно-конфигурируемых сетях с использованием гибридного облака / Ю.А. Кулаков, Е.Ю. Лопушен // Міжнародний науковий журнал" Науковий огляд". — 2017. — № 8 (40). — С.46-56.