**BC/NW 2017 № 1 (30):3.2**

**АЛГОРИТМ СБОРА СЕТЕВОГО ТРАФИКА ВЫЧИСЛИТЕЛЬНОЙ СЕТИ ПРИ ПЕРЕНОСЕ ИНФРАСТРУКТУРЫ В ОБЛАКО**

Кармилин Н.А.

В современном мире одной из самых динамично развивающихся технологий являются облачные технологии. Облачные вычисления, облачные хранилища, сервера, расположенные в облаке, уже сейчас довольно широко распространены. С каждым годом объем услуг, предоставляемых поставщиками, только увеличивается, как в России, так и по всему миру.

Облачные вычисления (англ. cloud computing) — информационно-технологическая концепция, подразумевающая обеспечение повсеместного и удобного сетевого доступа по требованию к общему пулу конфигурируемых вычислительных ресурсов (например, сетям передачи данных, серверам, устройствам хранения данных, приложениям и сервисам — как вместе, так и по отдельности), которые могут быть оперативно предоставлены и освобождены с минимальными эксплуатационными затратами или обращениями к провайдеру. Компьютер пользователя выступает при этом рядовым терминалом, подключенным к Сети. Компьютеры, осуществляющие облачные вычисления, называются «вычислительным облаком».

Таким образом, облачные вычисления дают возможность получения необходимых вычислительных мощностей по запросу из сети, причем пользователю не важны детали реализации этого механизма и он получает из этого «облака» все необходимое, многие пользователи, зачастую, даже не подозревают, что они взаимодействуют с облаками или что их данные хранятся в облаке.

Причин для переноса инфраструктуры в облако довольно много, во-первых, это снижение затрат, так как ресурсы оплачиваются в зависимости от частоты их использования, плата взимается за время работы того или иного сервиса или технологии. Во-вторых, при высокой загрузке выделенных ресурсов существует возможность автоматического масштабирования, что позволяет избежать больших задержек при высокой нагрузке на сервисы. В-третьих, облачные системы обладают более высоким уровнем безопасности, чем их традиционные аналоги. В-четвертых, в облачных решениях присутствуют встроенные средства анализа работы, отчеты об ошибках.

Переход к использованию облачных технологий позволяет повысить эффективность работы предприятия и снизить затраты на инфраструктуру.

Однако при переносе инфраструктуры в облако, компании сталкиваются с некоторыми проблемами, такими как:

1) Отсутствие собранной статистики о действующей инфраструктуре компании не позволяет выбрать конфигурацию облака, которая могла бы работать с той же эффективностью, что и текущая система

2) При переносе в облако, зачастую, полагаются на «прикидки», оценивая будущую структуру облака, отсутствует автоматизация в проектировании облачной среды.

3) Работа в облаке с выбранной конфигурацией зачастую оказывается менее эффективной, чем работа в существующей среде на предприятии, например, увеличение времени отклика, снижение объемов хранимых и передаваемых данных.

4) Выбранная облачная среда оказывается дороже в содержании, чем прежняя инфраструктура. Есть вероятность, что перенос в облако был не обязателен, а был навязан поставщиком облачных услуг.

5) При использовании облачных технологий нарушается работа некоторых сервисов, которые не были рассчитаны на перенос в облако.

Под переносом инфраструктуры в облако обычно понимают перенос инфраструктурных сервисов предприятия.

Инфраструктурные сервисы – это комплекс базовых приложений и информационных систем, обеспечивающих самые необходимые потребности организаций.[1]

Обычно в облако переносят те сервисы, которые должны быть доступны постоянно как внутри сети предприятия, так и за ее пределами, например web-сервисы и SQL-сервисы. Перенос в облако помимо технических результатов, приносит и преимущества менеджмента и мониторинга ресурсов. Многие облачные провайдеры предлагают поминутную оплату, для экономии финансов, постоянный мониторинг состояния облака, легкое управление и настройка через тонкий клиент (например, web-браузер) [2].

В ходе работы над проблемой создания подходящей конфигурации облачной среды, планируется разработать алгоритм на основе анализа собранных данных о вычислительной сети предприятия и построения архитектуры облака на основе этих данных. Эту задачу можно поделить на следующие подзадачи:

* Сбор данных об инфраструктуре;
* Анализ собранных данных;
* Создание проекта облачной среды;
* Тестирование и оценка полученного решения.

Алгоритм собирает информацию о серверах, использующихся на предприятии, о сервисах, которые работают на этих серверах, о состоянии сети, каналах связи, топологии сети.

На данном этапе разработан алгоритм анализа собранных данных о сети на основе перехваченного трафика с помощью сетевого сниффера (сниффер (от англ. to sniff — нюхать) — программа или устройство для перехвата и анализа сетевого трафика своего и/или чужого)[3]. Чтобы получить наиболее репрезентативные данные, необходимо запустить сниффер на каждом из серверов, записанные пакеты планируется анализировать непосредственно на том сервере, на котором они были собраны.

В ходе исследования были использованы данные вычислительной сети предприятия, полученные с помощью Snort.

Snort – программное обеспечение с открытым кодом, предназначенное для предотвращения сетевых вторжений, способное выполнять анализ трафика в реальном времени и ведения логов пакетов в сетях IP. Он может выполнять анализ протоколов, поиск по содержимому, и может быть использован для обнаружения различных атак, таких как переполнение буфера, стелс сканирование портов, CGI нападения, SMB зонды и многое другое.

Snort может работать в режиме сниффера, собирая пакеты для последующего анализа. Существует также режим IDS, в котором происходит анализ трафика по определенным заранее правилам на наличие атак.

Snort был выбран в качестве сниффера, так как он зарекомендовал себя в качестве эффективного анализатора трафика.

Рассмотрим корпоративную сеть, в которой есть четкое разделение компьютеров на клиентские машины и серверы. Сделаем допущение, что работа на предприятии ведется внутри сети, без выхода данных за ее пределы, без обращений к внешним серверам.

В случае если сервис работает на кластере, следует рассматривать кластер как один единый сервер.

На вход алгоритму необходимо подать список тех сервисов, которые нужно перенести в облако, либо сформировать список сервисов, подлежащих переносу, уже после сбора данных.

Выходными данными алгоритма являются захваченные сетевые пакеты, сгруппированные сначала по временным интервалам, а после – по портам, используемым сервисами.

Пусть m – количество клиентских машин, n – количество серверов в сети. Весь набор клиентских машин можно представить как $C=(c\_{1}, c\_{2}, c\_{3}, …, c\_{m})$, а набор серверов как $S=\left(s\_{1}, s\_{2}, s\_{3}, …, s\_{n}\right).$ Клиентские машины производят обращения к серверам, передают и принимают данные, общаясь, таким образом, с сервисами, которые находятся на серверах. Обозначим данные, отправляемые от *i*-го клиента на *j*-й сервер, как $d\_{ij}$, а данные, принимаемые *i*-м клиентом от *j*-го сервера, как $u\_{ij}$. Заметим, что значения $d\_{ij}$ и$ u\_{ij}$ неотрицательны.

Тогда можно составить матрицы обращений клиентов к серверам в виде $D= \left(\begin{matrix}d\_{11}&\cdots &d\_{1m}\\\vdots &\ddots &\vdots \\d\_{n1}&\cdots &d\_{nm}\end{matrix}\right)$ и $U= \left(\begin{matrix}u\_{11}&\cdots &u\_{1m}\\\vdots &\ddots &\vdots \\u\_{n1}&\cdots &u\_{nm}\end{matrix}\right)$. Каждый элемент этих матриц обладает общими свойствами: содержит идентификаторы отправителя и получателя (IP-адрес, MAC-адрес, NetBIOS-имя), номера портов отправителя и получателя, по которым производилась передача данных.

Допустим, что сервисы, к которым обращаются клиенты на сервер, имеют два порта – входящий и исходящий порты, $ p\_{i}, p\_{j}$ соответственно, причем у одинаковых сервисов на разных машинах порты также должны быть одинаковыми. Обозначим общее количество сервисов в системе как k, на каждом из серверов может быть запущено от 1 до k сервисов, обозначим использующиеся сервисы за $W=(w\_{1}, w\_{2}, w\_{3}, …, w\_{k})$, тогда можно представить $d\_{ij}$ или$ u\_{ij}$ как набор из кортежей вида $\{c\_{i}, s\_{j}, p\_{l1}, p\_{l2}, w\_{l}, V\}$, где $c\_{i}$ – *i*-тая клиентская машина, $s\_{j}$ – сервер, $p\_{l1},p\_{l2}$ – входящие и исходящие порты сервиса $w\_{l}$, к которому произошло обращение, $V$ – объем переданных данных.

Так как мы сделали допущение, что одинаковые сервисы имеют одинаковые порты, мы можем идентифицировать сервис по его портам.

Необходимо вычислить нагрузку, которая приходится на сервисы, работающие на серверах, для этого нужно выделить из общего числа обращений к сервисам те обращения, которые направлены на данный сервис.

Обозначим долю трафика, приходящуюся на на l-тый сервис, как $q\_{l}=\frac{V\_{l}}{V}$, где $V\_{l}$ – объем трафика, который предназначен для сервиса, а $V$ – совокупный объем трафика, переданный за время сбора данных. $q\_{l}$ – показатель нагрузки на выбранный сервис среди всех остальных, он показывает, какой объем передаваемого трафика необходим для работы данного сервиса.

Следует понимать, что для достоверной оценки нагрузки необходим сбор данных в сети, как во время рабочего дня, так и в выходные дни. Для более детальной оценки можно выделить три показателя: $q\_{l min}, q\_{l avg}, q\_{l max}$ – минимальная, средняя и максимальная нагрузки.

Из всех сервисов в системе выделим те, которые необходимо перенести в облако. Тогда, используя значения объемов передаваемых данных каждым сервером, можно найти рекомендуемую пропускную способность сети, выраженную в МБ/с.

Используя соотношение портов и сервисов, можно классифицировать выбранные серверы по ролям: SQL-сервер, WEB-сервер, Active Directory, почтовый сервер и т.п.

# Сбор информации о загрузке серверов в сети

Помимо сбора сетевого трафика, на сервере необходимо получить уровень загрузки процессора и оперативной памяти, эти данные потребуются для более точного подбора конфигурации виртуальных машин.

Показатели должны собираться через равные промежутки времени, в течение рабочего дня, а так же во время простоя. Наиболее важными из них являются показатели, собранные в течение рабочего дня, в особенности – максимальные значения.

Пусть уровень загрузки CPU j-того сервера в определенный момент времени *t* равен $cpu^{t}=\left\{cpu\_{ min}, cpu\_{avg}, cpu\_{max}\right\}$, а оперативной памяти – $ram^{t}=\{ram\_{min}, ram\_{avg}, ram\_{max}\}$, тогда соотнеся значения снятых показателей и объем передаваемых данных, можно описать *j*-тый сервер в момент времени *t* как $s\_{j}^{t}=\{W\_{j}^{t}, cpu^{t},ram^{t}\}$, где $W\_{j}^{t}$ – набор запущенных на данном сервере сервисов.

Как было сказано ранее, для более точной оценки необходимой производительности будущего облака важны пиковые значения загрузки CPU и оперативной памяти. Средние значения собранных показателей будут отображать примерную загрузку новой системы, новая конфигурация облака должна выполнять те же задачи, работать с сервисами, используя столько же ресурсов.

# Структура данных для анализа инфраструктуры предприятия

Для того чтобы эффективнее проанализировать собранные данные необходимо представить их в удобном для работы формате.

В качестве итоговой структуры данных выступает набор так называемых «снимков состояния системы», каждый элемент которого представляет собой набор показателей загруженности системы, а также набор собранных сетевых пакетов.

Снимки собираются с определенной периодичностью, в зависимости от заданной точности оценки. Независимо от периода, сетевые пакеты собираются постоянно, но, тем не менее, распределяются по снимкам в соответствии со временем захвата пакета и моментом наступления очередного сбора информации о системе.

Конечную структуру данных для выбранного сервера можно представить как $I^{t}=\left\{cpu^{t}, ram^{t}, N\_{t+T}^{t}\right\}$, где *N* – это набор собранных сетевых пакетов с момента времени *t* по *t+T*, где *T* – период, за который собираются данные.

# Заключение

Собрав и обработав данные с серверов, получаем таблицу ролей компьютеров в сети, а также информацию о необходимой для комфортной работы пропускной способности сети. Используя полученные сведения после проведения анализа данных в дальнейшем, можно сконфигурировать облако, выделив под каждую роль одну или несколько машин в зависимости от нагрузки.

В дальнейшем необходимо провести анализ полученного решения и оценить его эффективность, провести тестирование производительности облачной среды.

В ходе исследований были проведены следующие действия:

* Была уточнена постановка задачи сбора данных об инфраструктуре сети предприятия;
* Выбрана структура собираемых данных;
* Разработана методика сбора данных в корпоративной сети предприятия и выбраны инструментальные средства;

**Литература**

1. Мастерская информационных систем – WISYS.RU. [Электронный ресурс]: Электрон. текстовые данные. — Режим доступа: <http://wisys.ru/page/34>, свободный. — Загл. с экрана.
2. Инфраструктурные сервисы Windows Azure. [Электронный ресурс]: Электрон. текстовые данные. — Режим доступа: <https://msdn.microsoft.com/ru-ru/jj822933.aspx>, свободный. — Загл. с экрана.
3. Анализатор трафика. [Электронный ресурс]: Электрон. текстовые данные. — Режим доступа: [https://ru.wikipedia.org/wiki/Анализатор трафика](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80%20%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0), свободный. — Загл. с экрана.
4. Клементьев И. П. Введение в облачные вычисления [Электронный ресурс] / И. П. Клементьев, В. А. Устинов. — URL: <http://www.intuit.ru>.
5. SaaS. [Электронный ресурс]: Электрон. текстовые данные. — Режим доступа: <https://ru.wikipedia.org/wiki/SaaS>, свободный. — Загл. с экрана.
6. OpenNMS. [Электронный ресурс]: Электрон. текстовые данные. — Режим доступа: <https://ru.wikipedia.org/wiki/OpenNMS>, свободный. — Загл. с экрана.
7. От хаоса к порядку, или Выводим ИТ-инфраструктуру из кризиса. [Электронный ресурс]: Электрон. текстовые данные. — Режим доступа: [http://www.osp.ru/cio/2013/03/13034661](%20http%3A/www.osp.ru/cio/2013/03/13034661), свободный. — Загл. с экрана.
8. Леохин Ю.Л. Анализ информационной структуры корпоративной сети. [Электронный ресурс]: Электрон. текстовые данные. — Режим доступа: <http://cyberleninka.ru/article/n/analiz-informatsionnoy-struktury-korporativnoy-seti.pdf>, свободный. — Загл. с экрана.