УДК 004.051 ***10.5281/zenodo.3594781***

В. Гайдамако,
Институт машиноведения и автоматики НАН КР, Бишкек

ОБЗОР МЕТОДОВ МОНИТОРИНГА И ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ КОМПОНЕНТ ОБЛАЧНОЙ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ

В статье рассмотрены направления и способы осуществления мониторинга и оценки производительности компонент облачной среды. В частности, рассматриваются методы мониторинга производительности приложений, облачной инфраструктуры и сети, отдельных компонент. Приводятся распространенные показатели производительности, описываются способы оценки производительности, основанные на измерениях, аналитические способы оценки на базе математического или имитационного моделирования.

Ключевые слова: облачные информационно-измерительные системы, мониторинг, мониторинг приложений, мониторинг сети, мониторинг инфраструктуры, оценка производительности, аналитическая оценка, имитационное моделирование

Введение

В парадигме облачных вычислений ресурсы из пула виртуализированных ресурсов предоставляются пользователю по требованию посредством сети. Облачные сервисы – сервисы, которые Центры Обработки Данных (ЦОД) предоставляют пользователям, могут динамически масштабироваться в соответствии с потребностями пользователей, таким образом избавляя их от необходимости содержания собственной инфраструктуры (оборудования и обслуживающего персонала) [1]. Типичными моделями обслуживания являются Инфраструктура-как-сервис(Infrastructure-as-a-Service – IaaS), Платформа-как-сервис (Platform-as-a-Service - PaaS), и Программы-как-сервис (Software-as-a-Service - SaaS). Как сервис могут предоставляться и другие услуги, например физические датчики через виртуализацию предоставляются пользователю в Облачной Информационно-Измерительной Системе (ОИИС) [2], сетевые технологии также виртуализируются и предоставляются как услуга – Network-as-a-Service. Для поддержки работоспособности облака и гарантии качества предоставляемых услуг проводится мониторинг облачной среды – постоянная процедура проверки состояния компонент и управления рабочим процессом облачной инфраструктуры и связанными процессами [3]. Мониторинг облачной среды необходим и провайдеру облачных услуг, и пользователю – клиенту облака. Переводя всю или часть своей инфраструктуры в облако, клиент облака, а в случае ИИС клиентом является также и поставщик датчиков, ожидает, что он сохранит контроль над своими ресурсами в привычном объеме, причем значительную часть работы по поддержанию рабочего состояния ресурсов возьмет на себя провайдер облака. Провайдеру облака мониторинг нужен для обеспечения предоставления услуг гарантированного качества, балансировки нагрузки, планирования использовании ресурсов и инфраструктуры, обеспечения безопасности, надежности как инфраструктуры, так и данных, учета использования ресурсов пользователями, своевременного обнаружения и устранения отказов в обслуживании. Сбор и анализ данных для мониторинга производятся в режиме реального времени для обеспечения своевременного реагирования и предотвращения возможных нарушений безопасности, отказов в обслуживании и предоставления услуг ненадлежащего качества. ОИИС,

предоставляющая доступ к виртуализированным физическим датчикам, выдвигает свои требования к мониторингу, добавляя такую задачу, как мониторинг физических и виртуальных датчиков.

В настоящее время на ИТ-рынке представлено множество программных продуктов для облачного мониторинга, предлагаются разные подходы к оценке производительности облачных приложений и сервисов.

Виды мониторинга облачной среды

Мониторинг облачной среды включает мониторинг производительности приложений, мониторинг инфраструктуры (серверов, сетевых устройств, систем хранения данных, виртуальных ресурсов), мониторинг сети. Приложения в облаке могут разворачиваться как на физических среверах, так и в динамических средах – Виртуальных Машинах (ВМ), контейнерах, микросервисах. В таких условиях мониторинг состяния сети становится сложнее – если для мониторинга сетевого трафика обычного приложения может использоваться физический или виртуальный брокер сетевых пакетов, то для мониторинга контейнерных приложений этот метод неприменим, так как контейнеры, на которых работают приложения, могут быть добавляться, удаляться, завершаться, изменяя топологию сети. Контейнерные приложения могту размещаться на одном или нескольких серверах, могут мигрировать, могут использовать шифрованные оверлейные сети, делая свой трафик недоступным для анализа.

Провайдеру облака для оценки производительности отдельного приложения и производительности облака в целом нужна информация о коммуникациях между компонентами инфраструктуры, в том числе виртуальными, о производительности приложений, работающих на ВМ и контейнерах. При добавлении нового сервера, сетевого устройства, ВМ или контейнера требуется немедленный старт мониторинга этого компонента, нужно определить какие метрики отслеживать, как собирать, куда отправлять данные мониторинга. При завершении работы или удалении компонента мониторинг должен быть остановлен. [4]

Мониторинг приложений. Приложения в облаке – как правило, много-уровневые, многоконтейнерные, распределеные по множеству серверов. Цель системы мониторинга приложений – выявление отклонений от базовой производительности приложения, обнаружение узких мест, даже ошибок в коде, причин падения производителности. Программные продукты для мониторинга приложений собирают и анализируют данные, позволяющие оценить состояние физического оборудовани, состояние ВМ и Java машины, состояние контейнера, поведение приложения, состояние баз данных, кэшей, достижимость сторонних веб-ресурсов. Для мониторинга производительности приложений – сбора событий, метрик, трассировок – используются агенты, размещаемые на сервере, ВМ или контейнере. Собираемые агентами данные пересылаются в различных направлениях, на сервера мониторинга для сохранения истории, а затем на веб-сервера для визуализации и представления администраторам. ПО для мониторинга приложений – ManageEngine [5], Zabbix [6], Sysdig Monitor [**Ошибка! Источник ссылки не найден.**], RPTG [8], Prometheus [9].

Для **мониторинга инфраструктуры** также используются агенты, собирающие данные об использовании процессоров, оперативной памяти, дискового пространства, причем они могут собирать такие данные и о контейнерах. В последнее время для управления контейнерами (оркестровки) фактическим стандартом становится открытая система Kubernetes [10], позволяющая производить автоматическое развертывание, масштабирование и управление контейнерными приложениями. Мониторинг самого Kubernetes становится частью мониторинга инфраструктуры, в частности, проект Prometheus [9], предназначенный для сбора метрик из контейнеров Kubernetes [4.] Для

мониторинга инфраструктуры могут использоваться программные продукты Zenoss [11], RPTG[8], Sysdig Monitor [7]. Продукты, используемые для мониторинга сетей, как правило, включают в себя также элементы мониторинга инфраструктуры.

Мониторинг сетей основывается на полном доступе к сети и предполагает перехват пакетов. Проводится сбор данных о состоянии всех сетевых устройств, проверяется качество соединений (скорость, потеря пакетов). Также важно отслеживать поведение пользователей для поддержки производительности и обеспечения безопасности. Следует обратить внимание, что, вслед за вычислительными мощностями и средствами хранения, виртуализируются также и сети – появились VLAN, VXLAN и программно-определяемые сети (Software-defined Networks – SDN). Для сбора данных о состоянии виртуальных и контейнерных сетей традиционные методы непригодны. Здесь могут быть использованы два подхода. Первый – на ВМ и контейнерах устанавливаются агенты, которые собирают и отправляют пакеты на сервера мониторинга, которые обрабатывают пакетные данные и результаты отправляют Веб-серверам для предоставления администраторам. Этот подход используют поставщики ПО мониторинга сетей Extrahop[12], Gigamon[13] и Nubeva[14] Обработка данных производится на серверах, что снимает часть нагрузки с приложений. Но сами сервера мониторинга могут становиться узким местом. Для децентрализации обработки используется второй подход – обработка пакетов производится агентами, а серверу (или сервису) мониторинга отправляются результаты. Здесь возникает другая проблема – при изменениях в топологии сети требуется распространение изменений. Такого подхода придерживаются в ПО мониторинга CloudLens [15] и инструментарии компании Nubeva [4].

Мониторинг сетей в контейнерных средах – используется только для контейнеров и предполагает три подхода: использование сетевого плагина, агентов рабочей нагрузки и вспомогательных sidecar-контейнеров. При использовании первого подхода сетевой интерфейс контейнера является фреймворком (рабочей средой) для сетевого плагина. Трафик, проходящий через сетевой интерфейс, перехватывается виртуальной сетью, обрабатывается и передается по назначению. При использовании агента (агент встраивается в образ контейнера) агент запускается с рабочей нагрузкой, собирает пакеты и отправляет их для дальнейшей обработки или сам обрабатывает их и результаты отправляет для дальнейшего анализа. Агент может быть запущен как отдельный контейнер параллельно с рабочей нагрузкой контейнера («sidecar»), тем самым освобождая основной контейнер от работы по сбору и анализу пакетов. Кроме того, некоторые контейнеры сами могут перехватывать пакеты (Envoy proxy[16]) и могут заменять другие средства перехвата сетевых пакетов.

Можно выделить специфичные направления мониторинга, характерные для облачных сред и, в частности ОИИС.

Мониторинг веб-сайтов

В ОИИС веб-сайт, или портал используется для доступа клиентов облака к своим ресурсам – физическим и виртуальным датчикам и архивным данным, для отображения состояния системы для администраторов облака и сети, для отображения данных, которые могут быть интересны общественности. Это может быть один портал, несколько реплик или несколько разных веб-серверов, размещаемых на физических серверах провайдера облака или в арендуемых облачных структурах. Для мониторинга веб-сайта необходимо собирать и отображать данные о сетевом трафике, доступности, использовании ресурсов, производительности веб-приложений, доступности поиска по имени, времени загрузки страниц. Важным аспектом мониторинга веб-серверов является анализ готовности и производительности веб-сервиса за счет имитации

загрузки из сети. Простой мониторинг доступности веб-сайта может производиться с интервалом 1 -5-10 минут в зависимости от характера контента и статистики обращений. Некоторые параметры сайта должны отслеживаться из разных точек сети с интервалом от 10 секунд до минуты для своевременного решения проблем, например, с DNS-серверами (адрес сайта не распознается, хотя сам сайт физически доступен – ping работает), большим временем отклика, выполнением запланированных задач, долгой загрузкой файлов, подключением к базам данных и долгим ожиданием результатов. Мониторинг работоспособности включает проверки функционала сайта, особенно в случае изменений, с помощью тестовых приложений, генерирующих различные условия.

Мониторинг баз данных

Производится мониторинг доступности, запросов, процессов, обращений, целостность данных, все, что отражает использование данных в реальном времени. Анализ этих данных позволит вовремя производить изменения в инфраструктуре (обновления, апгрейд серверов) для предотвращения отказов или задержек в выдаче данных.

Мониторинг облачного хранения данных

Ресурсы хранения разных серверов объединяются в виртуальное пространство хранения (базы данных, распределенные файловые системы). Отслеживается состояние устройств и баз данных, файловых серверов, файлов, связь между серверами хранения, действия пользователей, объем свободных ресурсов хранения.

Мониторинг виртуальных ресурсов

Ресурсы предоставляются в пользование в виртуальном виде. В ОИИС это ресурсы виртуальных датчиков, ресурсы для хранения, вычислительные ресурсы для обработки данных. Это могут быть ВМ и контейнеры. Отслеживаются доступность, потребление ресурсов для каждой отдельной ВМ или контейнера - загрузка процессора, загрузка оперативной памяти, оставшееся место на жестком диске, и т.д. Эта информация необходима также для балансировки нагрузки и обеспечения эластичности – например, при превышении заранее установленных значений загрузки процессора, использования оперативной памяти, свободного дискового пространства могут подключаться дополнительные ресурсы – ВМ или контейнеры, и при достижении нижних пределов – количество задействованных контейнеров и ВМ может быть уменьшено.

Мониторинг состояния датчиков

Мониторинг виртуальных датчиков осуществляется также, как и мониторинг других виртуальных ресурсов – с помощью агентов, устанавливаемых на ВМ или контейнер. Для мониторинга физических датчиков отслеживается, прежде всего, доступность, соблюдение расписания связи, качество данных. Эти параметры могут собираться и анализироваться как координатором сенсорной сети, так и сервером облака (получающим сенсорную информацию) и сервером мониторинга.

Безопасность приложений в облаке

Хранение и обработка данных в облаке менее безопасна по сравнению с хранением и обработкой на локальных серверах. В центре обработки данных возможно обеспечить надлежащее обслуживание и безопасность инфраструктуры, но также должна быть обеспечена безопасность при передаче данных и в конечной точке использования. Первой линией защиты может служить, например, двухфакторная аутентификация пользователей и другие протоколы безопасности. Мониторинг использования обеспечивает возможность отслеживания опасных шаблонов действий

пользователя и своевременное принятие мер безопасности в случае определения возможного вторжения.

Оценка доступности

Доступность сервиса (компонента) может быть рассчитана как вероятность отклонения запроса на обслуживание. Вероятность отклонения запроса может зависеть от множества причин – общий объем ресурсов в системе – количество серверов, размера буфера (очереди), пропускной способности сети, методов управления облаком – политики диспетчеризации, применяемой в ЦОД, трафиком в сетях доступа к ЦОД и т.д.

На доступность сервиса, коме ограниченных ресурсов системы, влияют также отказы и ошибки в работе системы. При проектировании инфраструктуры облака могут быть использованы различные методы повышения устойчивости к отказам, что может привести, с одной стороны, к повышению доступности, но, с другой стороны могут привести к дополнительным задержкам в обслуживании и снижению пропускной способности системы.

Ну и конечно, на доступность сервиса могут повлиять проблемы с сетевым подключением. Это могут быть проблемы с физической невозможностью доступа или, например, с невозможностью доступа к сетевому узлу из-за отказов в работе DNS серверов.

Как определять доступность на практике? Когда сервис будет считаться доступным? Это не такой простой вопрос, и лучше, если он будет оговорен в Соглашении о качестве услуг SLA (Service License Agreement).

В [17] предлагается оценивать доступность сервиса по трем главным группам факторов: :

1. Доступность сети (с учетом провайдера, сетевых карт серверов, маршрутизаторов, коммутаторов, SAN-переключателей и т.д.),
2. Доступность гипервизора, определяющая доступность процессоров и памяти ВМ
3. Доступность Системы Хранения Данных (СХД).

Произведение значений этих параметров и бужет составлять интегральный показатель доступности. Для реализации такого подхода использовались агенты, устанавливаемые на ресурсах пользователей, что не всегда приемлемо – пользователь не всегда соглашается на установку дополнительного ПО. Далее использовалась фиксация сбоев сети, но и этот метод показал себя как неудовлетворительный. Тогда было решено создать пул виртуальных машин, максимально сходных с ВМ пользователей, установить на них агенты и проверять доступность этих машин (подход «sidecar»). Количество таких «тестовых» ВМ изменяется в зависимости от количества ВМ пользователей. [17]

При сбое в облаке система мониторинга фиксирует недоступность какой-то из эталонных ВМ, и это соответствует недоступности части ВМ заказчика, входящих в пул с эталонной. Исходя из этой статистики и рассчитывается «интегральный показатель доступности», который затем попадает в отчеты заказчику.

Метрики (показатели) оценки производительности

Для успешного осуществления мониторинга необходимо определить измеряемые метрики и их предельные значения. Для каждого вида ресурсов определяются данные, необходимые для управления и обеспечения качества предоставления услуг и принятия информированных решений. Должны быть определены события, требующие мониторинга, регистрации в системных журналах, отчетности, и способы оповещения о них – алерты и уведомления в случаях, требующих немедленного вмешательства

(попытки несанкционированного доступа, превышение порогов использования ресурсов, отказы, отсутствие сетевого соединения с устройствами и т.д.)

В [18] приводится каталог метрик для измерения, разделенный на группы:

Группа 1 – метрики для оценки коммуникаций предназначены для оценки качества связи – передачи данных или сообщений между облачными службами или между клиентом и облаком, между датчиками и облаком. Для оцеки передачи по TCP/IP MPI (Message Passing Interface) применяются разные показатели.

К метрикам оценки качества коммуникаций относятся:

- частота потери пакетов и коэффициент потерь Определяется как соотношение между временем потери пакетов и общим временем передачи, а коэффициент потерь определяется как соотношение между потерянными пробными пакетами к общему количеству проб. Доступность измеряется временем потерь, надежность – количеством сбоев;
- соотношение между общим временем работы и временем на сетевые коммуникации. Может быть получена путем сбора данных на множестве приложений в облаке. Показатель может использоваться для качественной оценки влияния коммуникаций на приложения в облаке;
- задержки (время) передачи MPI и TCP/UDP/IP (с, мс, мкс) и
- скорость передачи MPI и TCP/UDP/IP (Б/с, MB/s, ГВ/s).

Группа 2. Метрики для оценки вычислительной мощности. Вычислительная мощность характеризует высокопроизводительную (с большим количеством вычислений) обработку данных в облаке. Для оценки общей производительности облака используются крупномодульные (coarse-grain) облачные приложения,а для оценки производительности вычислений конкретного устройства используются приложения, интенсивно использующие процессор. К метрикам оценки производительности вычислений относятся:

- производительность на тестовом наборе и производительность экземпляра сервиса – измеряют производительность вычислений конкретного экземпляра сервиса в виде процентов от установленного порога. Для определения производительности на тестовом наборе пороговым значением является теоретический максимум для данного тестового набора, а для производительности экземпляра сервиса – теоретический максимум процессора;
- скорость в эластичных вычислительных блоках ECU (Elastic Compute Unit) – используется вместо традиционного FLOPS (количество операций с плавающей точкой в секунду). ECU – это мощность процессора 1.0-1.2 GHz 2007 Opteron или Xeon. Изначально эта метрика использовалась для оценки производительности экземпляра Amazon EC2 [19]
- загрузка процессора – используется для поиска узких мест, препятствую-щих росту производительности. Например, низкая загрузка процессора при максимальном количестве коммуникации показывает, что передача данных на этот объект является узким местом при данном наборе приложений;

Группа 3. Оценка эффективности памяти (кэша). Оперативная память (ОП) и кэш предназначены для обеспечения быстрого доступа к временно сохраняемым данным, получаемым с жесткого диска. Влияние ОП и кэша на общую производи-тельность оценить довольно сложно, поэтому практических методов и метрик оценки производительности ОП/кэша не так много. Кроме обычной оценки размера ОП и кэша, используются следующие метрики:

- скорость обновления оперативной памяти

- среднее время чтения данных из кэша (Mean Hit Time), то есть время доступа к данным, если они находятся в кэше (с);

- время получения/загрузки/отклика кэша (мс);
- скорость передачи одного бита/байта из/в ОП, Мб/с, Гб/с).

Группа 4. Метрики оценки систем хранения. Системы хранения используют-ся для хранения данных, получаемых от датчиков, служебных данных о пользова-телях, датчиках, состоянии системы, собственных данных пользователей и т.д. Данные хранятся долгосрочно, до удаления или остановки сервисов. По сравнению с доступом к памяти и кэшу время доступа к постоянно хранимым данным больше. Метрики оценки систем хранения:

- скорость доступа к одному байту данных (байт/с) – производительность доступа к данным очень маленького размера может быть доминирующей при обмене между устройствами памяти;
- скорость выполнения операций на тестовом наборе (оп/с, ops)
- производительность передачи между Blob и таблицей
- гистограмма скорости получения данных — представляется в виде графика, а не численного значения. Гистограмма наглядно показывает изменения скорости получения данных за период времени, отражает доступность облачного сервиса.
- задержка ввода/вывода на тестовом наборе (мин, с, мс)
- время ввода-вывода данных из Blob/Таблицы/Очереди (с, мс) – Blob/Table/Queue I/O Operation Time. Устройства хранения могут быть трех типов по способу предоставления данных – Blob, таблица или очередь. Ти-пичные операции ввода-вывода для Blob – это выгрузка (Download) и за-грузка (Upload); для таблиц - Get, Put и Query; для очереди - Insert, Retrieve, и Remove;
- скорость ввода-вывода бита/байта на тестовом наборе (бит/с, КБ/с, МБ/с);
- скорость ввода-вывода бита/байта из Blob (бит/с, КБ/с, МБ/с)

Группа 5. Метрики оценки общей производительности. В дополнение к оценке отдельных физических параметров, существует множество метрик для оценки общей производительности сервисов. Используются для оценки общей производительности коммерческих облачных систем. Некоторые из метрик:

- относительная производительность выше пороговой (отношение). – используется для стандартизации набора результатов оценки производительности для последующего сравнения.
- устойчивая производительность системы (Sustained System Performance (SSP). Для получения общей производительности облачных сервисов используется набор приложений. Включает другие метрики: геометрическое среднее производительности различных приложений на ядре процессора умножается на количество ядер процессора:
- средневзвешенное время отклика (Average Weighted Response Time (AWRT) – в качестве веса используется потребление ресурсов на запрос, метрика показывает, как долго средний пользователь будет дожидаться завершения за-прошенного сервиса. Потребление ресурсов на запрос оценивается как произве-дение времени выполнения запроса на количество экземпляров сервиса.

Группа 6. Оценка масштабируемости. Масштабируемость по-разному определяется для разных контекстов или разных точек зрения. Но, вне зависимости от определения, оценка масштабируемости облачных сервисов неизбежно производится при изменяющейся рабочей нагрузке и/или потребляемых облачных ресурсах. Так как эта варьирующая нагрузка/ресурсы обычно представляются в виде диаграмм и таблиц, метрики также представляются в виде диаграмм и таблиц. Фактически, в отличие от оценки других характеристик производительности, оценка масштабируемости (а также изменчивости) обычно подразумевает сравнение по наборам данных, которое удобно представлять в виде диаграмм и таблиц. Метрики:

- общая производительность (Aggregate Performance) и Падение производи-тельности ниже пороговой (Performance Degradation/Slowdown over a Baseline) – эти две метрики часто используются для оценки масштабируемости облачной системы при увеличивающейся нагрузке. Масштабируемость оценивается с точки зрения рабочей нагрузки;
- увеличение производительности выше пороговой (Performance Speedup over a Baseline) – используется для оценки масштабируемости при различном количестве облачных ресурсов, масштабируемость оценивается с точки зрения ресурсов;
- падение производительности ниже пороговой (Performance Degradation/Slowdown over a Baseline) – эта метрика может быть интуитивно воспринята как противоположная к «Увеличение производительно-сти выше пороговой», но она более значима для отображения масштабируемости, когда запрашиваемый сервис работает в различном количестве экземпляров или с разной рабочей нагрузкой. Масштабируемость оценивается с точки зрения рабочей нагрузки.

Группа 7. Метрики оценки вариабельности (изменчивости). В контексте оценки облачных сервисов Изменчивость показывает пределы изменчивости значений конкретного показателя производительности облачного сервиса. Изменчивость результатов оценки может быть вызвана различиями в производительности в зависимости от времени или расположения. Даже в одно время в одном месте в кластере экземпляров сервиса может наблюдаться изменчивость. Метриками также являются таблицы и графики. К метрикам изменчивости относятся:

- среднее, минимальное и максимальное значения какой-либо метрики, представляемые совместно;
- коэффициент вариации (ковариации);
- разница между минимумом и максимумом (%);
- среднее и стандартное отклонение;
- плотность вероятности функции;
- график значений с медианным/средним значением;
- представление нескольких графиков совместно для сравнения;
- представление отдельных графиков;
- таблицы.

Группа 8. Метрики оценки эластичности. Оценивают способность быстрого предоставления и освобождения ресурсов облака. Эластичность облачного сервиса означает способность реагировать и на растущую, и на падающую нагрузку. Метрики эластичности оценивают время предоставления или освобождения ресурсов.

- время захвата ресурса (Resource Acquisition Time). Захват ресурса – получение дополнительных ресурсов облака при возрастании нагрузки. Общее время захвата ресурса может быть разделено на время предоставления и время загрузки ресурса. Время предоставления – это интервал между моментом запроса на ресурс и его включением (подачей питания), время загрузки – интервал между предоставлением ресурса и его готовностью к использованию;
- время освобождения ресурса. Освобождение ресурса – это возвращение ставших ненужными ресурсов при падении нагрузки для экономии затрат. Общее время освобождения ресурса делится на время останова и время удаления. Время останова – время остановки работающего облачного ресурса, а время удаления – интервал между моментом полного останова и удалением ресурса из использования.

Это только часть метрик, которые могут использоваться при оценке производительности облачных сервисов, здесь представлены базовые, основные показатели. При планировании мониторинга, а также постановке различных эсперимента следует

выбрать нужные метрики и способы сбора информации о них. В таблице 1 приведены типичные метрики, используемые для оценки производительности облачных сервисов.

Таблица 1 – Типичные метрики, используемые при оценке производительности облачных сервисов

Метрика	Описание
Время отклика сервиса (задержка)	Задержка (время) между запросом на сервис и завершением сервиса
Пропускная способность сервиса	Количество заданий в единицу времени, выполняемое провайдером сервиса
Доступность сервиса	Вероятность принятия запроса провайдером сервиса
Использование системы (коэффициент загрузки системы)	Процент системных ресурсов требуемых для предоставления сервиса
Устойчивость системы	Стабильность производительности системы во времени, особенно при импульсных нагрузках
Масштабируемость системы	Способность системы сохранять производительность при росте нагрузки из-за увеличения размера или объема системы.
Эластичность системы	Способность системы адаптироваться к изменениям нагрузки

Методы оценки производительности

В целом подходы к оценке производительности облачных услуг по методологии исследования можно разделить на две или даже три большшие группы – оценка, основанная на измерениях, оценка, основанная на аналитическом моделировании, оценка, основанная на имитационном моделировании [20, 21]

Оценка производительности облачных услуг, основанная на измерениях. Исследования в области производительности облаков начались сразу же после их появления, в то время оценка часто оосновывалась на измерениях, проводимых на тестовой облачной инфраструктуре (испытательном стенде) с тестовой нагрузкой.

Процедура проведения измерений на облачном испытательном стенде состоит из следующих этапов: определение целей и границ оценки, определение оцениваемых свойств облачных сервисов, определение метрик производительности и выбор тестовых приложений. Далее проводится настройка среды и проводятся эксперименты.

Для методов оценки, основанных на измерениях, очень важна генерация соответствующей тестовой нагрузки. В случае ОИИС это нагрузка, источником которой являются пользователи, а также нагрузка, генерируемая физическими датчиками – данные должны быть своевременно собраны и перенаправлены на сответствующие виртуальные датчики.

Аналитические методы оценки производительности облачных сервисов менее затратны по сравнению с методами, основанными на измерениях, так как не требуют испытательных стендов, они основаны на моделировании, как математическом, так и имитационном. Эти методы позволяют оценить влияние большого количества параметров на производительность системы еще на стадии планирования и разработки системы и сервисов.

Классическим подходом к моделированию и анализу компьютерных систем является **Теория Массового Обслуживания (ТМО**). В [20] приведен обзор моделей, созданных с использованием ТМО. При проведении исследования очень важно создать модель нагрузок с заданными характеристиками, что довольно сложно из-за большого

разнообразия облачных сервисов и приложений. В большинстве исследований предполагалось, что процесс поступления запроса на обслуживание является пуассоновским процессом с экспоненциально распределенным временем между поступлениями заявок (inter-arrival time). Однако, большое разнообразие приложений, использующих разные облачные сервисы, может генерировать рабочие нагрузки с разными шаблонами. Поэтому разработка соответствующих и гибких моделей для рабочих нагрузок облачных сервисов, которые точно представляют трафик, генерируемый широким спектром приложений, является важной открытой проблемой. В ОИИС рабочая нагрузка генерируется физическими датчиками, с одной стороны, и пользователями – с другой. Распределение времени обслуживания во многих работах принималось экспоненциальным или произвольным. Разработка моделей для рабочих нагрузок от датчиков и пользователей является одной из задач моделирования ОИИС.

Методы ТМО обычно применяются для анализа конкретной архитектуры с конкретными допущениями по реализации сервисов. Но введение виртуализации привело к тому, что один и тот же сервис от одного провайдера может быть реализован различным образом для различных пользователей, например, сервисы для разных пользователей могут в разных центрах обработки данных, на серверах разных типов. Реализация сервиса может изменяться со временем, например, из-за обновления системы или перемещения виртуальной миграции на другой сервер. Кроме того, принцип SOA (Service Oriented Architecture, сервис-ориентированная архитектура) в предоставлении облачных услуг позволяет конечным пользователям использовать облачные сервисы без каких-либо знаний об их реализации. Абстракция ресурсов и инкапсуляция сервисов, обеспечиваемая виртуализацией и SOA в облаке, делают любые предположения об определенной архитектуре и технологии реализации облачной системы недопустимыми для оценки производительности с точки зрения пользователя. [20]

Сетевое исчисление (Network calculus) [22, 23] сделало возможным подход, основанный на профилировании. Основная идея этого подхода заключается в том, чтобы основывать свое моделирование и анализ на информации, относящейся к качеству обслуживания (Quality of Service – QoS), вместо того, чтобы моделировать какие-либо конкретные реализации сервисов и рабочую нагрузку. Эта информация – значения выбранных показателей (метрик) производительности, описывается в Соглашении об уровне обслуживания (Service License Agreement – SLA). Как правило, сюда включается пропускная способность сервисов, которая должна гарантироваться поставщиком, и максимальная рабочая нагрузка, которую пользователь может предоставить. Если разработать профили для минимальной пропускной способности, гарантируемой поставщиком услуг, и максимальной рабочей нагрузки, генерируемой пользователем, то на их основе можно будет получить некоторые границы для показателей производительности, например, наихудшую задержку обслуживания и максимальную длину очереди запросов на обслуживание.[20] Два ключевых понятия в сетевом исчислении – это кривая поступления запросов и кривая обслуживания. По сути, кривая обслуживания – это функция времени, которая дает нижнюю границу пропускной способности, которую сервер предоставляет клиенту. Точно так же кривая поступления запросов в сетевом исчислении является функцией времени, которая определяет максимальный объем рабочей нагрузки, которую пользователь может сгенерировать в течение произвольного интервала времени. Используя кривые поступления и обслуживания, сетевое исчисление позволяет определить верхнюю границу задержки любого запроса на обслуживание и максимальной длины очереди запросов на сервере.

Стохастические сети с вознаграждениями (Stochastic Reward Net – SRN) часто применяются совместно с методами ТМО для оценки производительности облачных сервисов. SRN по существу являются дополненными стохастическими сетями Петри (Stochastic Petri Nets – SPN) с возможностью определения выходных показателей в качестве наградных функций для оценки производительности сложных систем. В [24] авторы применили стратегию подмодели для упрощения моделирования и анализа крупномасштабных систем облачных вычислений IaaS. Для трех основных этапов предоставления услуг – предоставление ресурсов, подготовка ВМ и выполнение – разрабатываются подмодели на основе марковских цепей с непрерывным временем (Continuous-Time Markov Chain CMTC): Затем разрабатывается монолитная модель для представления взаимодействия между тремя этапами с использованием стохастических сетей с вознаграждением (SRN), все подмодели объединяются для получения общих результатов производительности облачной службы.

Имитационное моделирование используется, если объект моделирования изучен недостаточно, создание математической модели слишком сложно, нужно осуществить наблюдение за поведением компонент (элементов) процесса или системы в течение определенного периода, объекта моделирования не существует или он недоступен для изучения и наблюдения, наблюдаемые процессы или поведение системы контролируются путем замедления или ускорения явлений в ходе имитации, при изучении поведения системы в случае изменения или внесения новых компонент, в новых ситуациях, а также в проектируемых системах [25**Ошибка! Источник ссылки не найден.**]. Для моделирования облачных систем применяются могут применяться средства моделирования GridSim[26], CloudSim и ее расширения [], GreenCloud [28], ICanCloud [29], Simic [30], SimGrid [31]. В результате моделирования данные о работе компонент собираются в файлы трассировки, которые в дальнейшем используются для анализа поведения моделируемой системы.

Заключение

Мониторинг облачной среды и оценка производительности облачных сервисов необходима и провайдеру, и клиентам облака. В настоящей статье приводится обзор методов мониторинга производительности приложений, инфраструктуры и сети в облаке, приводятся основные метрики производительности, которые могут быть использованы в дальнейшем при оценке производительности имитационных моделей ОИИС, отдельных сервисов. Рассматриваются подходы к оценке производительности облачных сервисов – оценка на основе измерений и аналитические методы. Аналитические методы оценки используют ТМО, сетевое исчисление, стохастические сети с вознаграждениями, разрабатываются новые методы. Каждый из подходов имеет свои преимущества и недостатки, в таблице 2 приведен их краткий обзор.

Таблица 2 – Преимущества и недостатки основных подходов к оценке производительности облачных сервисов

Метод	**Преимущества**	**Недостатки**
Методы, основанные на измерениях	Предоставляют информацию о производительности доступных облачных сервисов; отражают эффективность обслуживания в реалистичных сценариях работы; показывают производительность	Ограничены доступными облачными испытательными платформами и их настройками; проведение тестов затратно, невозможно предсказать производительность при

	сервисов при практическом применении	введении новых сервисов и/или изменении настроек
Методы, основанные на ТМО (могут применяться совместно с SRN)	Отстутствуют затраты на эксперимент, возможно предсказать производительность новых сервисов и настроек, можно оценить влияние большого количества параметров	Трудно моделировать гетерогенную облачную инфраструктуру, трафик различных приложений, может не отражать производительность реальных сценариев обслуживания
Методы профилирования	Применимы к гетерогенным реализациям облачных сервисов, различным сетевым нагрузкам приложений, отражает особенности виртуализации и абстракции	Эффективность зависит от точности профилей обслуживания и спроса, текущий анализ для худшего случая производительности

Рассмотренные методы и виды мониторинга облачной среды могут быть применены при определении методов мониторинга ОИИС и воспроизведены при имитационном моделировании. Оценка производительности на основе измерений может применяться для сравнительной оценки производительности моделей ОИИС различного состава и топологии. Методы аналитической оценки могут быть реализованы в математических моделях ОИИС, а также применены для имитации тестовой рабочей нагрузки от физических датчиков и пользователей веб-портала – клиентов ОИИС, клиентских приложений, поставщиков физических датчиков и администраторов.

Литература

1. Cloud monitoring guide. / URL:https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/manage/monitor/(дата обращения 25.10.2019).
2. Гайдамако В.В. Инфраструктура Sensor-Cloud – облачные информационно-измерительные системы. // Проблемы автоматики и управления – 2018. – №2 (35). С. 109–118.
3. URL:https://www.motadata.com/ru/cloud-monitoring-all-you-need-to-know/ (дата обращения 25.10.2019)
4. Кочуков А. Какие существуют подходы для мониторинга в «облаке»? URL: https://networkguru.ru/cloud-monitoring-part2/ (дата обращения 25.10.2019).
5. ManageEngine Applications Manager, URL: https://www.manageengine.com/ru/applications_manager/ (дата обращения 25.10.2019)
6. Zabbix URL: https://www.zabbix.com/ru/(дата обращения 25.10.2019).
7. Sysdig monitor. Container + Kubernetes monitoring, alerting, and troubleshooting. URL: https://sysdig.com/products/monitor/ (дата обращения 25.10.2019)
8. RPTG Network Monitor URL: https://www.ru.paessler.com/packet_loss_test (дата обращения 25.10.2019)
9. Prometheus URL: https://prometheus.io/ (дата обращения 25.10.2019).
10. Production-Grade Container Orchestration URL: https://kubernetes.io/ (дата обращения 25.10.2019).
11. Zenoss URL: https://www.compuway.ru/monitoring/ (дата обращения 25.10.2019)

12. Network Detection & Response URL: https://www.extrahop.com/ (дата обращения 25.10.2019).
13. Gigamon URL: https://www.gigamon.com/(дата обращения 25.10.2019).
14. Nubeva URL: https://www.nubeva.com/ (дата обращения 25.10.2019).
15. CloudLens URL: https://www.ixiacom.com/products/cloudlens (дата обращения 25.10.2019).
16. EnvoyProxy URL: https://www.envoyproxy.io/ (дата обращения 25.10.2019).
17. https://www.onlanta.ru/press/media/it-weekly/16445/ (дата обращения 25.10.2019)
18. Z. Li, L. O'Brien, H. Zhang, R. Cai, On a catalogue of metrics for evaluating commercial cloud services, in: Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing, 2012, pp. 164–173.
19. Вопросы и ответы по Amazon EC2. URL: https://aws.amazon.com/ru/ec2/faqs/ (дата обращения 25.10.2019)
20. QiangDuan, Cloud service performance evaluation: status, challenges, and opportunities – a survey from the system modeling perspective. // Digital Communications and Networks, Volume 3, Issue 2, May 2017, Pages 101-111. URL: https://www.sciencedirect.com/science/article/pii/S2352864816301456, (дата обращения 25.10.2019
21. Ворожцов А.С., Тутова Н.В., Тутов А.В. Оценка производительности облачных центров обработки данных. // T-Comm – Телекоммуникации и Транспорт, 2014. URL: https://cyberleninka.ru/article/v/otsenka-proizvoditelnosti-oblachnyh-tsentrov-obrabotki-dannyh (дата обращения 25.11.2019).
22. URL: https://ru.wikipedia.org/wiki/Сетевое_исчисление (дата обращения 25.10.2019)
23. J.L. Boudec, P. Thiran, Network Calculus: A Theory of Deterministic Systems for the Internet, Springer Verlag, Berlin, 2001.
24. R. Ghosh, F. Longo, V.K. Naik, K.S. Trivedi, Modeling and performance analysis of large scale IaaS Clouds, Future Gener. Comput. Syst. 29 (5) (2013) 1216–1234.
25. Замятина О. М. Вычислительные системы, сети и телекоммуникации. Моделирование сетей, учебное пособие для магистратуры. – Москва : Издательство Юрайт, 2019. – 159 с. – (Университеты России). – ISBN 978-5-534-00335-2. – Текст : электронный // ЭБС Юрайт [сайт]. – URL: https://urait.ru/bcode/433938 (дата обращения: 28.11.2019).
26. The GridSim Simulator. URL: https://gridsim.hevs.ch/ (дата обращения 25.05.2019)
27. CloudSim: A Framework For Modeling And Simulation Of Cloud Computing Infrastructures And Services URL:https://github.com/Cloudslab/cloudsim (дата обращения: 28.11.2019).
28. GreenCloud – the green cloud simulator. URL:https://greencloud.gforge.uni.lu/ (дата обращения 25.05.2019).
29. iCanCloud. URL:http://icancloud.org/ (дата обращения 25.05.2019).
30. Simix. URL:https://www.windriver.com/products/simics/ (дата обращения 25.05.2019)
31. SimGrid: Versatile Simulation of Distributed Systems/ URL:https://simgrid.org/ (дата обращения 25.05.2019).