Тестирование и методология сравнения облаков

Антоненко Виталий

План

- Характеристики рабочей нагрузки облачных приложений
- Показатели производительности для облачных приложений
- Тестирование облачных приложений
- Инструменты тестирования производительности
- Нагрузочное тестирование и обнаружение «узких мест»

Бенчмаркинг

- Бенчмаркинг важен по следующим причинам :
  - Распределение и планирование ресурсов
    - Процесс подготовки и планирования ресурсов для облачных приложений включает в себя определение объема вычислительных ресурсов, памяти и сетевых ресурсов для обеспечения функционирования приложения.
    - Помощь в сравнении альтернативных архитектур развертывания и выборе лучшей архитектуры развертывания, которая соответствует требованиям к производительности приложения.
  - Эффективное использование (утилизация) ресурсов
    - Помощь в определении количество вычислительных ресурсов, памяти и сетевых ресурсов для приложений и отметить ресурсы, которые либо недостаточно используются, либо чрезмерно предоставлены приложению.
  - Готовность приложений (Production level)
    - Производительность приложения зависит от характеристик рабочих нагрузок. Различные типы рабочих нагрузок могут привести к снижению производительности для одного и того же приложения.
    - Чтобы обеспечить готовность приложения к работе, важно смоделировать все типы рабочих нагрузок, и сравнивать с показаниями приложения на схожих рабочих нагрузках.

Бенчмаркинг облачного приложения - ЖЦ

- Сбор/Генерация рабочей нагрузки
  - Сбор трафика реальных приложений.
  - Для генерации трафика рабочей нагрузки используется специальный инструментарий для анализа информации, такой как запросы, отправленные пользователями, отметки времени запросов и т. д.
- Моделирование рабочей нагрузки
  - Включает создание математических моделей для генерации синтетического трафика.
- Спецификация рабочей нагрузки
  - Так как модель рабочей нагрузки отличается в зависимости от приложения. Необходимо иметь инструментарий для описания рабочей нагрузки. Например, IXIA profile.
- Генерация синтетической рабочей нагрузки
  - Важным требованием для синтетического генератора рабочей нагрузки является то, что генерируемые рабочие нагрузки должны соответствовать реальным нагрузкам.

Подходы для генерации синтетической рабочей нагрузки

- Эмпирический подход
  - В этом подходе трафик приложений отбирается и воспроизводится для генерации синтетических рабочих нагрузок.
  - Эмпирический подход не обладает гибкостью, поскольку полученные реальные данные, используются для создания рабочей нагрузки, что может не отражать рабочие нагрузки на других системах с различными конфигурациями и условиями функционирования.

- Аналитических подход
  - Использует математические модели для определения характеристик рабочей нагрузки, которые используются синтетическим генератором рабочей нагрузки.
  - Аналитический подход является гибким и позволяет создавать рабочие нагрузки с различными характеристиками путем изменения атрибутов модели генерации.
  - С помощью аналитического подхода можно изменять параметры модели рабочей нагрузки по одному и исследовать влияние на производительность приложения для измерения чувствительности приложения к различным параметрам.

Эмуляция пользователя vs Агрегированные потоки

Наиболее распространенными методами для генерации рабочей нагрузки являются:

- Эмуляция пользователя:
  - Каждый пользователь эмулируется отдельной сущностью (нитью, приложением, процессом), который имитирует действия пользователя, чередуя запросы и простаивая..
  - Атрибуты для формирования рабочей нагрузки в методе эмуляции пользователей включают, например, Think Time, типы запросов, зависимости между запросами.
  - Эмуляция пользователя позволяет осуществлять контроль (и исследование) аспектов поведения пользователей, взаимодействующих с тестируемой системой.
- Генерация агрегированной рабочей нагрузки:
  - Позволяет указать точные моменты времени, в которые запросы должны поступать в тестируемую систему.
  - НЕ оперирует понятием индивидуального пользователя при генерации рабочей нагрузки, поэтому этот подход нельзя использовать, когда необходимо учесть зависимости между запросами.
  - Зависимости могут быть двух типов: между запросами и зависимостями по данным.
  - Зависимость между запросами: текущий запрос зависит от предыдущего запроса, в то время как зависимость по данным: текущим запросам требуются входные данные, которые получают из ответа предыдущего запроса.

Характеристики рабочей нагрузки

- Сетевые сессии
  - Набор последовательных запросов, представленных пользователем, представляющий собой сеанс работы с приложением.
- Межсессионный интервал
  - Время между сессиями пользователя.
- Время «на подумать» (Think Time)
  - В сеансе пользователь последовательно отправляет ряд запросов. Временной интервал между двумя последовательными запросами называется Think Time.
- Длительность сессии
  - Количество запросов, отправленных пользователем в сеансе, называется длительностью сеанса.
- Workload Mix
  - Комбинация рабочей нагрузки определяет переходы между различной функциональностью приложения и пропорцию, в которой используется та или иная функциональность.

Показатели производительности для облачных приложений

Наиболее часто используемые показатели производительности для облачных приложений:

- Время отклика
  - Временной интервал между моментом, когда пользователь отправляет запрос в приложение и момент, когда пользователь получает ответ.

- Пропускная способность
  - Количество обработанных запросов в секунду.

Требования к методологии бенчмаркинга

- Точность
  - Точность методологии бенчмаркинга определяется тем, насколько сгенерированные синтетические рабочие нагрузки имитируют реалистичную рабочую нагрузку.
- Простота использования
  - Должна быть удобной для пользователя и должна включать минимальное ручное кодирование для написания скриптов для генерации рабочей нагрузки, которые учитывают зависимости между запросами.
- Гибкость
  - Должна позволять осуществлять гранулярный контроль над такими атрибутами рабочей нагрузки, как Think Time, межсессионный интервал, продолжительность сеанса, комбинирование рабочих нагрузок.
  - Типовой анализ осуществляется путем одновременного изменения одной характеристики рабочей нагрузки, при этом остальные остаются неизменными.
- Широкая область применения
  - работает для широкого круга приложений и не привязана к архитектуре приложения или типам нагрузки.

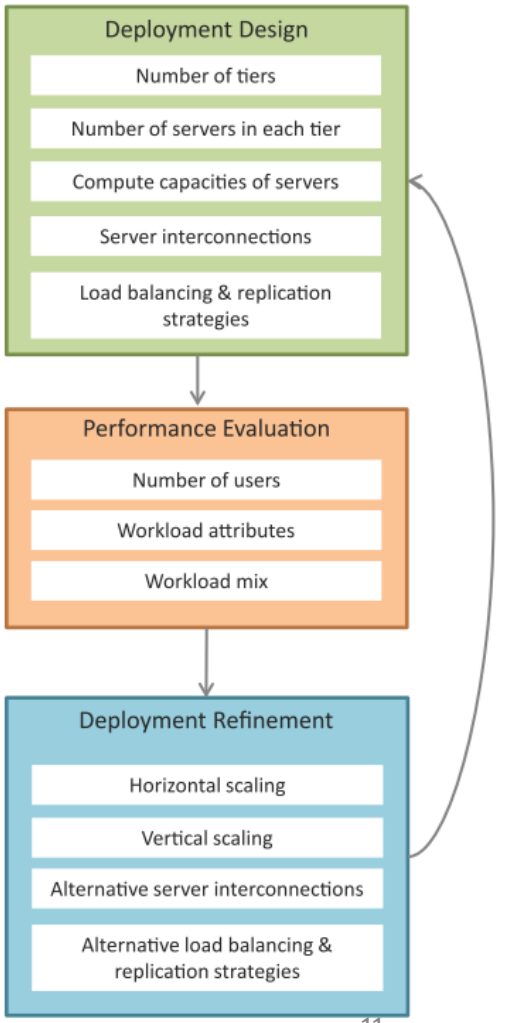
Типы тестирований

- Базовое тестирование
  - Для сбора данных показателей производительности всего приложения или компонента приложения.
  - Используются для сравнения различных изменений настройки производительности, которые впоследствии производятся в реальном приложении.
- Нагрузочное тестирование
  - Нагрузочные тесты оценивают производительность системы с несколькими пользователями и разными уровнями рабочей нагрузки, которые встречаются в реальной жизни.
  - Количество пользователей и комбинация рабочей нагрузки обычно указываются в конфигурации нагрузочного тестирования.
- Стресс тестирование
  - Стресс-тесты загружают приложение в том месте, в которой оно должно «сломается».
  - Эти тесты выполняются для фиксирования сбоя приложения, а также условий сбоя приложения и отслеживаемых показателей, которые могут предупреждать о последующих сбоях при повышенных уровнях рабочей нагрузки.
- Тесты на выдержку (Soak Tests)
  - Предполагают длительное применение приложения к фиксированному уровню рабочей нагрузки.
  - Тесты на выдержку помогают определить стабильность приложения при длительном использовании и то, как производительность изменяется со временем.

Прототипирование

- Прототипирование может помочь в выборе вариантов архитектуры развертывания.
- Сравнивая производительность альтернативных архитектур развертывания, прототипирование может помочь в выборе наилучшей и наиболее экономичной архитектуры развертывания, которая может соответствовать требованиям к производительности приложений.

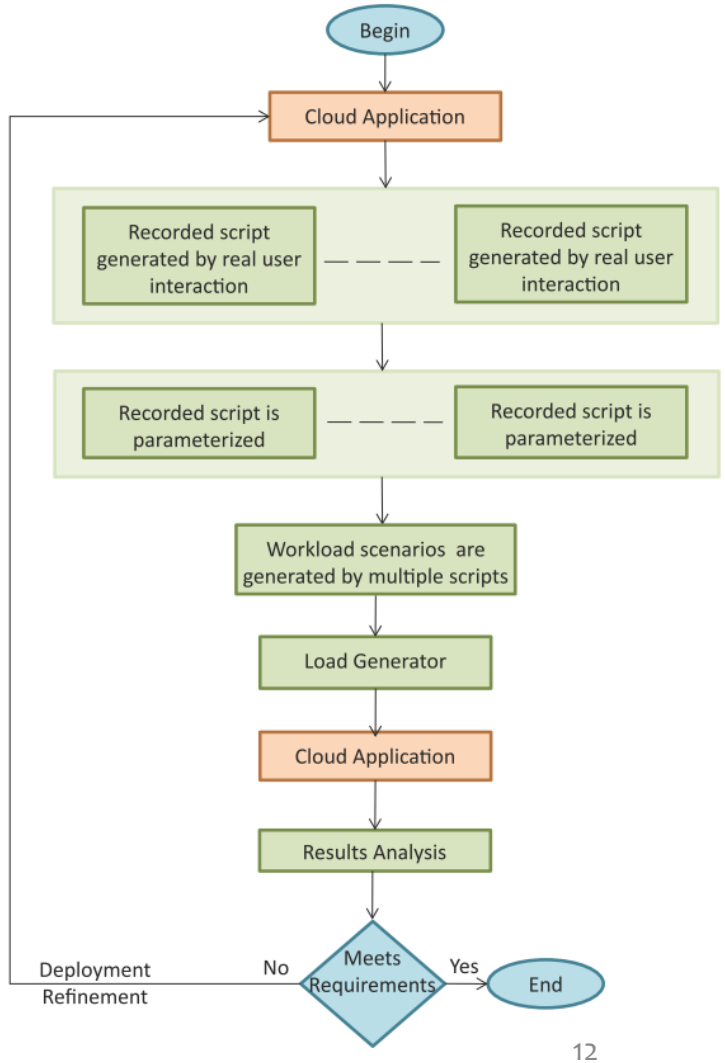
- Развертывания приложения - это итеративный процесс, который включает в себя следующие шаги:
  - Выделение ресурсов и размещения приложения.
  - Оценка производительности: проверка удовлетворяет ли приложение требованиям к производительности при развертывании
  - Уточнение развертывания: Развертывания уточняются на основании оценок производительности. На этом этапе могут существовать различные альтернативы, такие как вертикальное масштабирование, горизонтальное масштабирование.



Анализ производительности

Традиционных подход

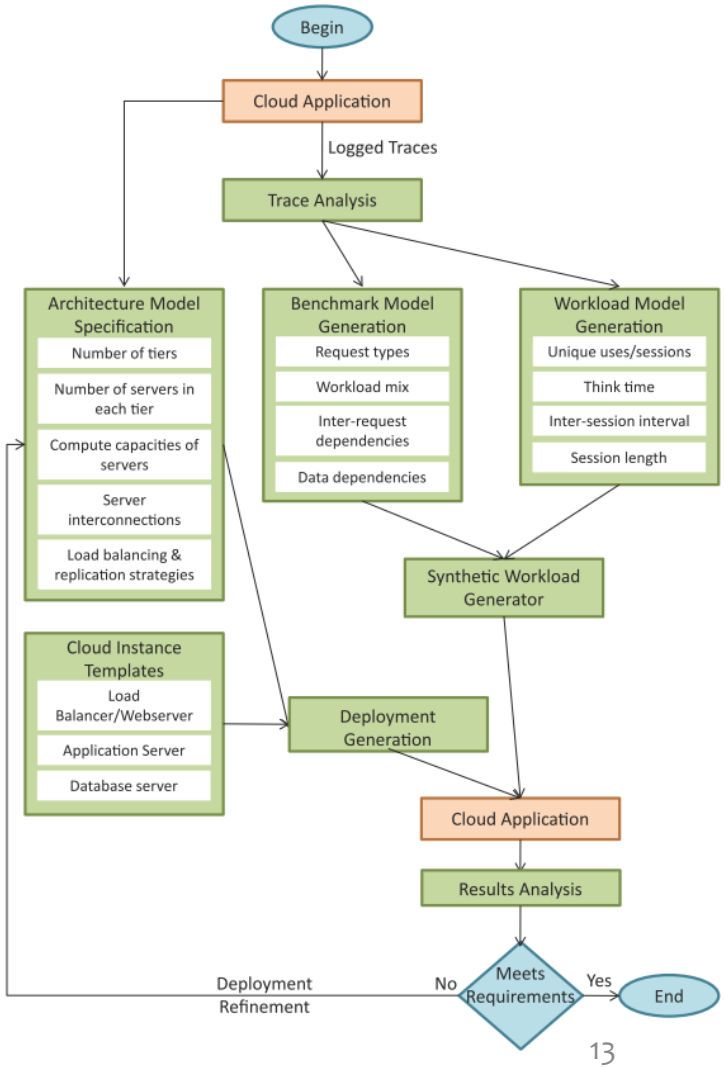
- Сначала подготавливает модель, эмулирующая поведение пользователя.
- Записанные сценарии вариантов использования приложения затем параметризуются для учета случайности в параметрах приложения и рабочей нагрузки.
- Для создания различных сценариев рабочей нагрузки необходимо записать несколько сценариев.
- Чтобы добавить новые спецификации для набора рабочей нагрузки и новых запросов, новые сценарии должны быть записаны и параметризованы вручную.
- Не способны генерировать синтетические рабочие нагрузки, которые имеют те же характеристики, что и реальные рабочие нагрузки.
- Не позволяют быстро сравнивать различные архитектуры развертывания.



Анализ производительности

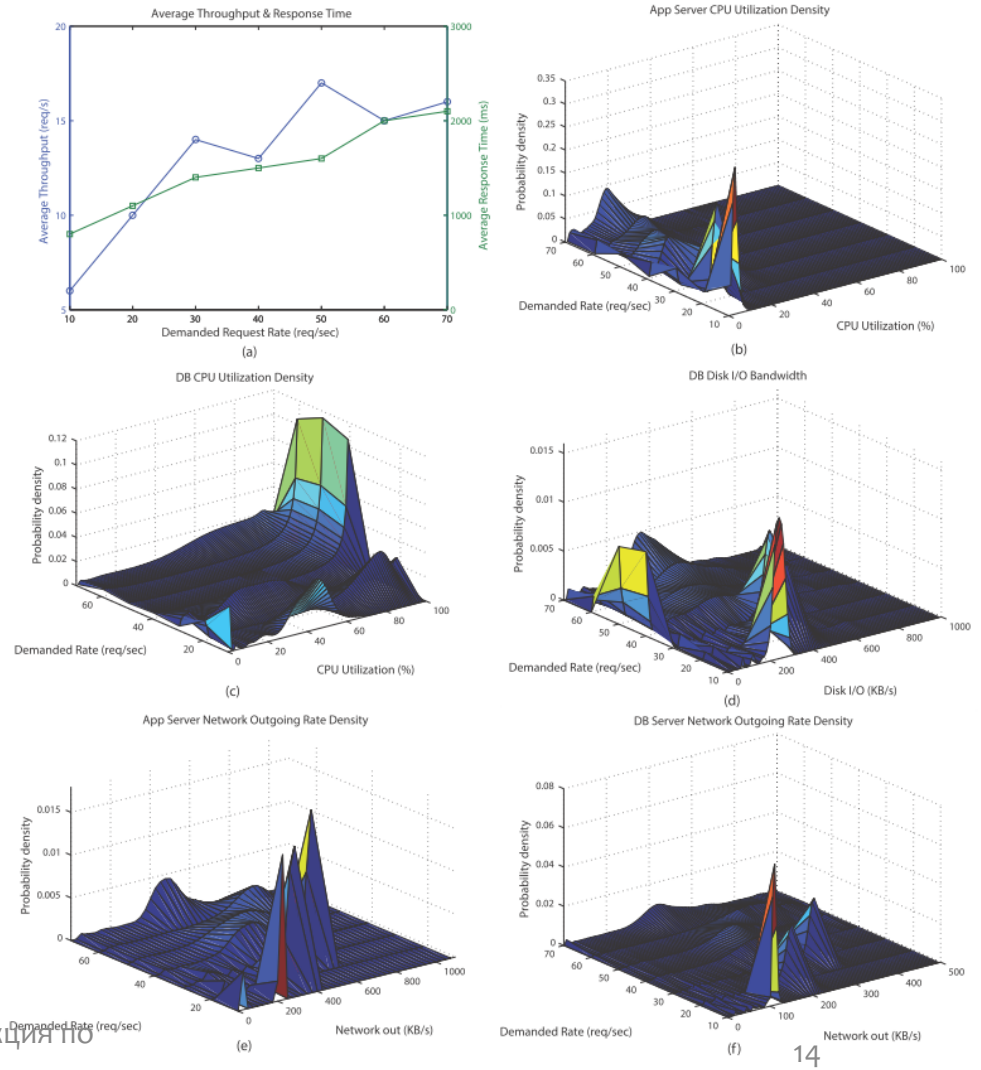
Полностью автоматизированный подход

- Реальный трафик анализируются с целью создания моделей генерации рабочей нагрузки, отражающих характеристики облачных приложений.
- Статистический анализ пользовательских запросов в реальном трафике выполняется для определения правильных распределений, которые могут использоваться для моделирования атрибутов модели рабочей нагрузки.
- Реальные трафик анализируется для создания моделей рабочих нагрузок.
- Различные сценарии рабочей нагрузки могут быть созданы путем изменения спецификаций модели рабочей нагрузки.
- Генерируемые синтетические рабочие нагрузки имеют те же характеристики, что и реальные рабочие нагрузки.



Анализ результатов тестирования

- Рис. (A) показывает среднюю пропускную способность и время отклика. Наблюдаемая пропускная способность возрастает по мере увеличения частоты запросов. Чем больше запросов подается в секунду в приложение, тем больше время ответа. Наблюдаемая пропускная способность насыщается при 50 запросов в секунду.
- На рис. (B) показана загрузка центрального процессора одного из серверов приложений. Этот график показывает, что процессор сервера приложений является дефицитным ресурсом.
- Рис. (C) показывает загрузку ЦП сервера базы данных. Из этого графика плотности мы видим, что ЦП базы данных тратит большой процент времени частоте запросов более 40 запросов в секунду.
- На рис. (D) показан график загрузки полосы пропускания дисковых операция ввода-вывода базы данных.
- На рисунке (E) показана скорость выходного сетевого потока сети для одного из серверов приложений
- На рис. (F) показан график скорость выходного сетевого потока для сервера базы данных. На этом графике мы наблюдаем непрерывное насыщение скорости сети до отметок 200 Кбит / с.



Вывод из тестирования

- Пропускная способность непрерывно увеличивается, когда требуемая частота запросов находиться в диапазоне с 10 до 40 запросов в секунду. При увеличении частоты до 40 запросов в секунду, мы наблюдаем, что пропускная способность насыщена, что связано с высокой загрузкой центрального процессора ЦП сервера базы данных. Из анализа графиков загрузки различных системных ресурсов мы видим, что ЦП базы данных является «узким местом» системы.

Спасибо за внимание!

Вопросы?

anvial@lvk.cs.msu.su

Антоненко Виталий