

ПОДХОД К ОЦЕНКЕ ПРОИЗВОДИТЕЛЬНОСТИ WEB- БАЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Рассматриваются методы оценки производительности информационных систем, а также особенности web-базированных информационных систем (WБИС) и проблемы оценки их производительности. Приводится описание метода Software Performance Engineering, а также предлагается его модификация для оценки производительности WБИС: описываются этапы SPE, которые должны быть изменены для оценки производительности WБИС на всех стадиях жизненного цикла разработки.

1. Введение

Современные информационные системы (ИС) становятся все более сложными и зависимыми от сетевых технологий, доступных благодаря Internet. Размещение приложений, зависящих от Web-сайтов, интрасетей, а также клиент-серверные технологии порождают проблему обеспечения их должного функционирования с приемлемой производительностью. Создание масштабируемых Web-служб приводит к появлению определенных трудностей из-за большого разрыва между средней и пиковой нагрузками [1]. Служба, хорошо приспособленная для работы при пиковых нагрузках, является неэкономной и использует слишком много ресурсов. С другой стороны, если система адекватно работает только при средних значениях нагрузки, то она наверняка будет плохо работать при пиковых нагрузках.

Даже очень хороший Web-сайт с высокой функциональностью бесполезен, если для соединения с ним или выгрузки из него полезной информации нужно тратить слишком много времени. Пользователи стараются избегать сайтов подобного рода. На практике проблемы производительности могут иметь разнообразные нежелательные последствия, в том числе финансовые потери и спад продаж, снижение продуктивности работы компании и ухудшение ее имиджа. Это подчеркивает важность заблаговременного планирования производительности систем, работающих в сетях. Планирование возможностей и производительности включает умение предсказать момент, в который существующие приложения не смогут обеспечить требуемый уровень обслуживания.

Планирование производительности также важно для моделирования работы нового приложения на этапе проектирования, или приложения, переносимого с мейнфреймов в клиент-серверные среды [3]. Разработчики приложений сталкиваются с необходимостью выбора из большого количества различных архитектур, сильно отличающихся по производительности и стоимости. Примерами таких вариантов являются: клиент-серверная архитектура, трехзвенная клиент-серверная архитектура, распределение функций и таблиц баз данных в пределах нескольких серверов, выбор типов клиентов и серверов, а также сетевая связность. Отказ от предварительного планирования возможностей и производительности вплоть до момента, когда приложения будут физически реализованы, может привести к катастрофическим последствиям, поскольку в подобной ситуации потребуются дорогостоящее повторное проектирование и реализация кода. Модели производительности необходимо применять еще и для того, чтобы новое клиент-серверное приложение удовлетворяло требованиям по уровню обслуживания.

Существует множество показателей производительности, однако они являются специфичными для различных типов информационных систем.

2. Формулирование проблемы

На сегодняшний день имеется множество методов оценки производительности информационных систем, таких как метод Software Performance Engineering (SPE); построение моделей производительности на базе систем массового обслуживания; моделирование производительности программного обеспечения (ПО) с использованием UML и сетей Петри и др. Однако все они ориентированы на информационные системы в целом и не учитывают специфику работы веб-базированных систем.

Таким образом, актуальными задачами являются выделение показателей производительности Web-базированных информационных систем (WБИС), построение модели производительности таких систем на базе существующих подходов и использование модели на ранних этапах проектирования в целях обеспечения соответствия разрабатываемой системы требованиям к производительности или своевременной корректировки требований в случае невозможности или нецелесообразности их соблюдения.

2.1. Метод Software Performance Engineering

Термин Software Performance Engineering (SPE) был впервые введен К.Ю. Смит в 1981 г. SPE представлен как набор методов, сопровождающих разработку ИС с учетом производительности на протяжении всего ЖЦ разработки для обеспечения соответствующего качества ИС с точки зрения производительности. Необходимо отметить, что использование SPE на ранних стадиях разработки позволяет избежать «fix-it-later» подхода (в переводе с английского – «исправь это позже», слэнг). Таким образом, если проблема производительности обнаружена на ранней стадии, будет легче и дешевле принять соответствующее проектное решение для ее устранения.

В рамках метода SPE сначала необходимо определить, каких целей или количественных показателей необходимо достичь. Очевидно, что они изменяются от одной стадии разработки к другой, а также у систем различных классов. Производительность системы определяется временем ее отклика на запросы пользователя и пропускной способностью. Сбор данных завершается определением типичных сценариев использования системы и их вероятностей. Построение и оценка модели производительности системы – это одна из основ SPE. В зависимости от типа модели она может анализироваться либо интерпретироваться.

Основными показателями производительности в SPE обычно являются утилизация ресурсов, время ожидания и время отклика (response time).

Основным положением данного подхода является разделение модели (ПО) и технического обеспечения (ТО). Такое деление, с одной стороны, позволяет по отдельности специфицировать модели ПО и ТО, а с другой – улучшает переносимость этих моделей.

Модель ПО содержит в себе существенную информацию о поведении ПО и основывается на графах выполнения (графах, узлы которых представляют собой нагрузочные компоненты ПО, а дуги – переходы управления). Нагрузочный компонент ПО – это набор инструкций или процедур, выполняющих определенную задачу. Графы выполнения включают несколько типов узлов: основные, циклические, условные, разделяющие и объединяющие. Каждому узлу присваивается весовой коэффициент с использованием вектора требований, который представляет использование ресурсов узла (т.е. требования для каждого ресурса).

Модель аппаратного обеспечения основана на расширенных моделях массового обслуживания (РММО). В соответствии со спецификацией РММО необходимо определить: компоненты (т.е. сервисные центры), топологию (т.е. соединения между центрами) и параметры (такие как классы задач, маршруты задач между центрами, расписание сервиса в центрах, требования сервисов в центрах сервиса). Спецификация компонентов и топологии выполняется в соответствии с описанием системы, в то время как спецификация параметров получается на основе информации, содержащейся в модели ПО, и знаний о возможностях ресурсов. Параметризованная РММО представляет собой модель программно-аппаратного комплекса всей системы.

2.2. Необходимость модификации метода SPE для WБИС

Исходя из того, что в рамках SPE содержатся методы по обеспечению производительности на всех стадиях разработки ИС, его можно использовать и для разработки WБИС. Согласно проведенным исследованиям, для учёта специфики web-систем необходимо внести в этот метод следующие изменения: ввести основной показатель производительности WБИС – время отклика, и на этапе сбора требований формировать каталог функций системы с учетом требуемых значений времени отклика для любого запроса в рамках реализации каждой функции; на этапе анализа структуры системы в качестве модели выполнения использовать навигационную модель из web-инжиниринга, а также ввести сценарии выполнения функций, которые строятся на основании навигационной модели и представлены в виде диаграмм последовательностей; расширить диаграммы последовательностей, полученные на предыдущем этапе, данными о производительности каждого

участвующего в ней компонента путем измерения производительности существующих модулей системы и экспертной оценки создаваемых модулей; использовать модель массового обслуживания для численных расчетов и получения значения критерия производительности.

3. Решение проблемы. Модификация метода Конни Смит

С учетом предлагаемых изменений модифицированный подход состоит из следующих пяти этапов.

1. Установка требований к производительности WБИС и исследование функций системы. Формирование каталога бизнес-функций и расширение его за счет внесения требований к времени отклика каждого запроса в рамках реализации каждой функции.

2. Исследование структуры системы, выделение web-компонентов и связи между ними. Дополнение каталога бизнес-функций сценариями их реализации путем анализа навигационной модели. Разработка диаграмм последовательностей для каждой функции, которые описывают процедуру реализации каждой функции с участием web-компонентов.

3. Оценка ресурсоемкости компонентов модели (web-компонентов) в контексте каждой бизнес-функции и расширение полученных на предыдущем этапе диаграмм последовательностей за счет внесения значений производительности.

4. Построение модели массового обслуживания на основании диаграмм последовательностей и проведение численных расчетов. Сравнение полученных результатов с заявленными требованиями к производительности.

5. Если полученные результаты не удовлетворительны, – анализ модели и вынесение предложений по улучшению производительности. Повторное проведение оценки.

3.1. Установка требований к производительности WБИС и исследование функций системы. Формирование каталога бизнес-функций

Для различных систем сложность установления требований к производительности может существенно отличаться. Так, для встраиваемых систем требования четко задаются требованиями внешней системы. Для систем с участием человека выделить требования к производительности сложнее, поскольку удовлетворенность производительностью – это субъективное и растяжимое понятие, а люди постоянно хотят все более быстрых реакций системы.

Сбор требований к производительности информационных систем, которые используются людьми, является сложным, и общепринятый подход для этого не разработан. В первую очередь, трудности связаны с тем, что пользователи ожидают, что новая ИС существенно изменит их работу и не могут достаточно точно спрогнозировать, как именно они будут использовать новую систему. В то же время разработчики, которые имеют навыки грамотного оформления требований, могут формулировать сами требования неточно и делать неточные оценки, что приведет к неудовлетворенности пользователей или необоснованным затратам на излишние работы по повышению производительности системы.

Также слабо развиты практические методики сбора требований к производительности ИС, что характерно и для других нефункциональных требований.

Как было установлено, основным показателем производительности WБИС является время отклика, т.е. время, которое проходит с момента отправки запроса до момента получения ответа от сервера. В зависимости от характера запросов длительность выполнения запроса, а следовательно, и время отклика, существенно разнятся. Поскольку одни и те же запросы могут осуществляться в рамках реализации различных функций и отличаться при этом только параметрами запроса, то длительность выполнения каждого запроса также может существенно отличаться в зависимости от контекста его вызова (значений параметров) и от бизнес-функций, к реализации которых относится выполнение запроса. Таким образом, требования к времени отклика должны устанавливаться для каждой функции с учётом передаваемых параметров запроса и контекста вызова функции.

Такую оценку можно производить на основании каталога бизнес-функций системы, который регламентирован в рамках метода SPE [2]. Пользователям системы предлагается оценить максимальную комфортную задержку по каждой бизнес-функции с учетом периодичности ее выполнения и характера самой функции.

В результате сбора требований должна быть составлена модифицированная таблица каталога требований [2]: помимо столбцов, регламентируемых SPE (подсистема, наимено-

вание бизнес-функции, дата составления, составитель), для каждой бизнес-функции должно быть указано требуемое максимальное время отклика в миллисекундах.

3.2. Исследование структуры системы, построение модели выполнения и диаграмм последовательностей для бизнес-функций

На данном этапе создается модель выполнения, которая представляет функции системы и описывает сценарии типичного использования этих функций пользователями [4]. Модель может иметь различные виды представления:

- в соответствии с подходом Software Performance Engineering (SPE) создаются графы выполнения, которые представляют последовательности операций, включая предшествование, циклы, выбор, разделение и объединение потоков выполнения. Это семейство моделей включает в себя диаграммы задач, диаграммы деятельности [5,6];

- конечные автоматы применяются в сочетании с такими языками моделирования как UML, SDL и ROOM [7,8];

- сети Петри и алгебра стохастических процессов тесно связаны с конечными автоматами и представляют поведение системы через глобальные состояния и переходы. Решаются путем анализа цепей Маркова в области состояний или путем имитационного моделирования.

Модель производительности расширяет модель выполнения за счет внесения данных о производительности каждого компонента системы [2]. Данные, полученные с помощью модели производительности, могут быть использованы при задании параметров модели массового обслуживания или его многоуровневой модели. Модели, базирующиеся на состояниях, могут быть трансформированы в цепи Маркова или псевдоцепи Маркова для их численного решения. Кроме того, любые модели могут быть трансформированы в имитационные модели.

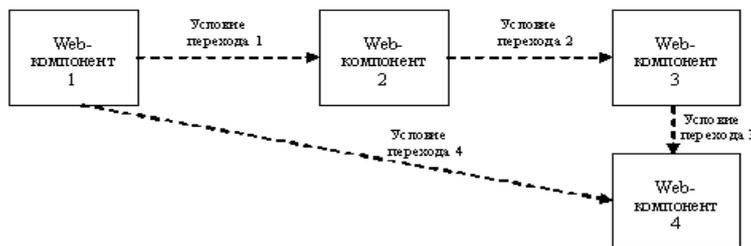
Все приведенные выше подходы доказали свою состоятельность на практике, однако при разработке различных систем на первый план выходит стоимость и длительность построения и анализа модели.

Этот этап модифицируется заменой модели выполнения навигационной моделью. Эта замена обусловлена тем, что состояние работы пользователя с web-системой в каждый момент времени характеризуется следующими параметрами:

- страница, отображаемая в текущий момент времени;
- событие, которое инициирует переход на следующую страницу;
- собственно переход, т.е. страница, на которую попадает пользователь при выполнении события [12].

Таким образом, навигационная модель включает в себя web-компоненты и взаимосвязи между ними. Каждый web-компонент обрабатывает запрос, после чего возвращает результат пользователю или передает запрос для дальнейшей обработки следующему web-компоненту, что отражено

на рисунке.



Навигационная модель WБИС в общем виде

На основании навигационной модели необходимо создать сценарии выполнения бизнес-функций системы. Каждый сценарий описывает, какие web-компоненты и в какой последовательности участвуют в реализации бизнес-функций. Сценарии описываются с помощью диаграмм последовательностей, каждая из которых описывает реализацию бизнес-функции, объектами выступают web-компоненты, участвующие в сценарии, а вызовы содержат информацию о параметрах запроса.

Каждая из которых описывает реализацию бизнес-функции, объектами выступают web-компоненты, участвующие в сценарии, а вызовы содержат информацию о параметрах запроса.

3.3. Оценка ресурсоемкости компонентов модели (web-компонентов) в контексте каждой бизнес-функции и внесение их как параметров модели

После построения структуры системы возможна оценка ресурсоемкости ее компонентов по отношению к таким ресурсам как время выполнения ее центральным процессором,

длительность дисковых операций, сетевые службы и т.д. В ряде случаев компоненты могут быть сразу оценены на этапе построения структуры системы [4]. Это даст возможность на основе численных характеристик ТО получить время использования каждого ресурса, что непосредственно влияет на численные характеристики производительности ВБИС.

Можно выделить два основных источника фактических значений для ресурсоемкости компонентов в период выполнения:

1. Проведение измерений на таких частях системы, которые уже реализованы, таких как базовые сервисы, существующие компоненты, прототипы или более старые версии [9].

2. Оценки ресурсоемкости, базирующиеся на суждении экспертов.

Оценка существующих компонентов системы, естественно, предполагает, что фактические значения ресурсоемкости будут получены только для части системы, а значит, оценка производительности всей системы на ранних этапах проектирования должна выполняться с применением более сложных моделей, методов и математического аппарата, что существенно усложняет применение данного подхода.

Хорошо зарекомендовало себя применение метода экспертных оценок с участием экспертов-проектировщиков и специалистов в области оценки производительности [5]. Обычно проектировщики стремятся создавать и оценивать прототипы, избегая работы с неточными понятиями и значениями. Кроме того, оценки, полученные на ранних этапах проектирования, являются полезными даже в тех случаях, если они весьма приблизительны.

Оценка параметров – важный этап, однако трудно выработать универсальный план, который бы позволил преодолеть все сложности, возникающие в процессе его проведения.

Для построения модели производительности необходимо оценить производительность каждого web-компонента навигационной модели в контексте каждой функции. Для этого с использованием сценариев (диаграмм последовательностей), составленных на предыдущем этапе, формируются расширенные диаграммы последовательностей, содержащие информацию об оценочной производительности каждого участвующего в сценарии web-компонента в каждый момент его активности.

Ресурсоемкость различных web-компонентов может существенно варьироваться в зависимости от подаваемых на вход параметров и данных для обработки. Исходя из проведенных исследований и практических экспериментов установлено, что при оценивании необходимо принять для каждой бизнес-функции значения входных параметров, которые создают максимальную нагрузку на web-компонент, и производить оценку исходя из максимально возможного объема данных для обработки.

Оценку производительности web-компонентов осуществляют методом экспертных оценок – для вновь разрабатываемых компонентов, а значения производительности существующих (переиспользуемых) компонентов берут исходя из экспериментальных данных.

При проведении оценки конфигурация технического обеспечения сервера считается фиксированной, поэтому оценку можно производить, например, в секундах.

3.4. Получение численных значений на основании модели производительности и сравнение полученных результатов с требованиями к производительности

На этапе получения численных значений производительности на основании анализа ее модели в связи с неточностью значений параметров, возможно, придется производить оценку несколько раз. Это дает преимущество моделям, на использование которых затрачивается меньше времени.

Например, модели на базе сетей массового обслуживания сравнительно легковесные, предоставляют возможность использовать базовые аналитические модели и быстро решаются [7,10]. Однако основное ограничение моделей на базе сетей массового обслуживания заключается в том, что они применимы только к системам, использующим лишь один ресурс одновременно. Расширенные модели массового обслуживания, описывающие несколько ресурсов, являются более сложными и дольше решаются [11].

Основной проблемой моделей массового обслуживания является их ограниченность в представлении системы, как очереди задач, реализующих бизнес-функции. Эти модели не могут представить, к примеру, логику взаимодействия задач.

Сети Петри и другие модели, базирующиеся на состояниях, могут использоваться для моделирования систем с логически запутанными алгоритмами, которые в принципе не могут быть представлены в виде моделей массового обслуживания. Они могут описать абсолютно любое поведение системы [11].

Отрицательным качеством моделей, базирующихся на состояниях, является то, что они требуют хранения значений вероятностей по пространству состояний, количество которых может насчитывать миллионы и быстро растет с увеличением системы. Таким образом, эти модели не являются масштабируемыми и могут успешно применяться только для малых систем.

Подход, основанный на имитационном моделировании, позволяет формализовать любые модели. Имитационные модели являются тяжеловесными с точки зрения времени выполнения, необходимого для получения точных результатов, и могут давать бессмысленные результаты в случае применения их неспециалистами, задающими неадекватные значения точности и погрешности. Однако имитационное моделирование не накладывает на проектировщика никаких ограничений, что является основным его преимуществом.

Для того чтобы получить параметры для выбранного математического аппарата, на основании расширенных диаграмм последовательностей, полученных на предыдущем этапе, создается формализованная модель производительности WБИС. Она представляет собой кортеж множеств:

$$M_{\text{произ}} = \{F, Q, C, P_{\text{треб}}, P_{\text{факт}}, PC_{\text{ф}}, PQ_{\text{факт}}, oFQ, oQC, oFPT\}, \quad (1)$$

где F – множество бизнес-функций системы; Q – множество запросов к WБИС; C – множество web-компонентов; $P_{\text{треб}}$ – множество значений максимально допустимого времени отклика для каждой бизнес-функции; $P_{\text{факт}}$ – множество фактических максимальных значений времени отклика для каждой бизнес-функции; $PC_{\text{ф}}$ – множество фактических значений времени отклика каждого запроса для бизнес-функции; $PQ_{\text{факт}}$ – множество фактических значений времени отклика для каждого web-компонента в рамках каждого запроса для бизнес-функции; oFQ – связи между бизнес-функциями и запросами; oQC – связи между запросами и web-компонентами; $oFPT$ – множество отношений между бизнес-функциями системы и значениями максимального требуемого времени отклика в каждой из них.

В результате проведения расчётов, предложенных в рамках данного метода, получают фактические значения производительности для каждого запроса в рамках каждой бизнес-функции и учитываются в расширенном каталоге требований отдельной графой «Фактическое максимальное время обработки».

3.5. Повторное проведение оценки

Результаты, полученные в ходе оценки производительности системы, могут быть неудовлетворительными, т.е. требуемые значения времени отклика могут быть превышены. В таком случае проводится анализ причин превышения времени отклика: проводится измерение задержки сети, загрузки дисковой системы и процессоров. В конечном счете, модификация сводится к внесению изменений в техническое обеспечение, программное обеспечение или в требования к системе.

В случае, если получены неудовлетворительные результаты по каким-то бизнес-функциям, необходимо путем анализа сценариев реализации таких функций выделить web-компоненты, которые являются наиболее ресурсоемкими. Затем необходимо оценить возможность и стоимость внесения изменений в программное обеспечение (изменить сами компоненты или реорганизовать структуру системы таким образом, чтобы исключить или минимизировать влияние этих компонентов на ресурсоемкость всей бизнес-функции) или повысить производительность технического обеспечения. Исходя из стоимости вносимых изменений, необходимо выбрать один из этих вариантов или, если стоимость обоих изменений слишком высока, внести корректировки в каталог требований к системе.

После внесения изменений, если они имели место, необходимо повторить процедуру оценки.

4. Выводы

Представленная методика позволяет отслеживать соответствие основных метрик производительности web-базируемых информационных систем требованиям заказчика на всех этапах жизненного цикла разработки ИС – от проектирования до внедрения и сопровождения.

За основу был взят подход Software Performance Engineering, в рамках которого определены методы для обеспечения качества с точки зрения производительности на всех этапах разработки ИС. Было проведено исследование особенностей WБИС и с учётом этих особенностей в метод SPE были внесены модификации.

Использование данного подхода позволяет снизить риск невозможности или слишком большой стоимости реализации функциональности с учётом требований к производительности, а также позволяет получать более точные прогнозы производительности системы благодаря ориентации на WБИС.

Список литературы: 1. *Douglis F., Kaashoek M.F.* Scalable Internet Services. IEEE Internet Computing, 2001. P. 36-37. 2. *Performance Validation at Early Stages of Software Development.* 3. *Connie U. Smith, Murray Woodside.* 3. *Menasce D.A., Goma H.* A Method for Design and Performance Modeling of Client. IEEE Software Engineering, 11/2006. P. 1066-1085. 4. *Smith C.U.* . Performance Engineering of Software Systems, Addison-Wesley, 1990. 5. *Smith U., Williams L.G.* Performance Engineering Evaluation of CORBA-based Distributed Systems with SPE•ED. Proc. 10th Int. Conf. On Modelling Tools and Techniques, Palma de Mallorca, 1998. 6. *Woodside C.M., Hrischuk C., Selic B., Bayarov S.* . A Wideband Approach to Integrating Performance Prediction into a Software Design Environment. Proc. of First International Workshop on Software and Performance (WOSP98). Oct. 1998. P. 31-41. 7. *El-Sayed H., Cameron D., Woodside M.* . Automated Performance Modeling from Scenarios and SDL Designs of Telecom Systems. Proc. of Int. Symposium on Software Engineering for Parallel and Distributed Systems (PDSE98), Kyoto, April 1998. 8. *Vetland V.* Measurement-Based Composite Computational Work Modelling of Software, PhD. thesis, University of Trondheim, Aug 1993. 9. *Menasce D.A., Goma H.* On a Language Based Method for Software Performance Engineering of Client/Server Systems. Proc. of First International Workshop on Software and Performance (WOSP98). Oct. 1998. P. 63-69. 10. *Rolia J.A., Sevcik K.C.* The Method of Layers. IEEE Trans. on Software Engineering. Aug. 1995. V. 21. N. 8. P. 689-700. 11. *Woodside C.M., Neilson J.E., Petriu D.C., Majumdar S.* The Stochastic Rendezvous Network Model for Performance of Synchronous Client-Server-like Distributed Software. IEEE Transactions on Computers. Jan. 1995. V. 44. N. 1. P. 20-34. 12. *Gorshkova E., Novikov B.* Exploiting UML Extensibility in The Design of Web Information Systems.

Поступила в редколлегию 30.06.2008

Петриченко Александр Вячеславович, канд. техн. наук, доцент кафедры ИУС ХНУРЭ. Научные интересы: проектирование web-базируемых информационных систем, управление проектами. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-451.

Мантула Дмитрий Вадимович, частный предприниматель. Научные интересы: проектирование web-базируемых информационных систем, управление проектами. Адрес: Украина, 61064, Харьков, ул. Камская, 1, кв. 64, тел. +38 (093) 419-18-27, +38 (057) 777-80-10.

Малафеев Евгений Евгеньевич, аспирант заочной формы обучения кафедры менеджмента ХАИ. Научные интересы: проектирование web-базируемых информационных систем, управление проектами. Адрес: Украина, 61166, Харьков, ул. Есенина, 1, кв. 10.