

©2012 г.

Жигалов И.Е.,
Озерова М.И.,
Шевченко Д.В.

АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ МЕТОДАМИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Рассмотрен подход к анализу производительности телекоммуникационных систем с использованием раскрашенных сетей Петри. Имитационные методы обеспечивают возможность исследования характеристик поведения существующих систем в различных режимах работы.

An approach to performance analysis of telecommunication systems using colored Petri nets. Simulation techniques provide an opportunity to study the characteristics of the behavior of existing systems in the various modes.

Ключевые слова: анализ производительности, раскрашенные сети Петри, телекоммуникационные системы.

Key words: capacity analysis, color Petri networks, telecommunication systems.

Производительность часто является основной проблемой при проектировании, разработке и конфигурировании телекоммуникационных систем (ТС). Параметры производительности ТС часто определяются производительностью подсистем, отвечающих за передачу данных между программными приложениями, для расчета которых применяют аналитические расчеты и моделирование. Для моделирования информационных компонент ТС может быть использован аппарат раскрашенных сетей Петри (РСП), который предоставляет возможность создания моделей с небольшим количеством упрощений по сравнению с другими методами моделирования [1]. При этом, по сравнению с использованием аналитических моделей, РСП позволяют анализировать производительность систем, предъявляя существенно меньшие требования к ресурсам.

Для детального анализа производительности ТС был разработан ряд РСП-моделей для проведения имитационного моделирования. Основными показателями, влияющими на производительность

информационных подсистем, являются количество передаваемых сообщений, скорость их передачи и время обработки сообщений подсистемой. В приложении данные запрашиваются клиентами асинхронно по мере необходимости. Пусть архитектура клиента была разработана таким образом, что обрабатываемая информация группируется в виде представлений. Подсистема может одновременно просматривать только одно представление. Представления в свою очередь сформированы из графических элементов интерфейса, данные для которых асинхронно запрашиваются у ТС [2].

Примером такого представления может послужить группа элементов, отвечающих за отображение в графическом виде информации из базы данных системы - кортежи таблиц. Когда пользователь выбирает нужный кортеж из таблицы, все остальные элементы графического интерфейса запрашивают необходимую им информацию у ТС. Таким образом, каждое представление - это набор запросов к ТС. Информация, полученная от ТС, обрабатывается асинхронно и отображается в соответствующих

графических элементах. При выборе нового кортежа в таблице все элементы полностью обновляются. В силу того, что это распределенное приложение, ответ ТС на запрос элемента представления подсистема может не получить. Тогда через определенный интервал времени (тайм-аут) элемент представления информирует подсистему, что данные не получены, и подсистема может обновить представление, чтобы повторно загрузить информацию.

В модели на основе РСП представления отображаются с помощью меток. Соответственно начальная разметка позиции определяет представления, имеющиеся в системе. Позиция «REQUESTS» отвечает за отображение представлений на набор запросов ТС. Переход «VIEW TO REQUEST» формирует запросы ТС. Позиция «VIEWS WAIT» содержит информацию о частоте выбора подсистемой тех или иных представлений, здесь может быть задан любой закон выбора подсистемой представлений, вплоть до полной имитации реального сценария работы подсистемы.

Сформированные запросы помещаются в очередь запросов подсистемы, которая работает по принципу FIFO. В данной модели важными параметрами запроса являются: идентификатор запроса, который обеспечивает уникальность запроса; идентификатор представления, элемент которого инициировал запрос; команда (функция), результат выполнения которой должна отослать ТС в виде ответа. Все остальные параметры запроса являются несущественными и в данной модели не рассматриваются. В случае передачи запроса серверу появляется еще один важный параметр - время передачи. Для этого запрос преобразуется в значение типа «REQUEST_TIMED».

Максимальное время ожидания подсистемой ответа ТС (тайм-аут) задается на основе требований, сформулированных в техническом задании на систему. Переход «SEND» выполняет отправку запроса, помещая его в буфер сети (позиция «REQUEST NET BUFFER»), откуда запрос может быть взят через определенное время (время передачи запроса по сети). Переход «SEND» также помещает этот запрос в позицию «REQUEST WAIT», в которой содержатся запросы, ожидающие ответ ТС, а также устанавливает признак того, что запрос еще не получил ответ ТС, помещая запрос в позицию

«FLAG». Запросы из позиции «REQUEST WAIT» могут быть извлечены по истечении тайм-аута. После чего они помещаются в позицию «REFRESH REQUEST». Если к этому времени ответ на запрос не получен, тогда переход «REFRESH REQUESTS» формирует новый набор запросов ТС, имитируя действия подсистемы, которая обновляет представления, пытаясь обновить информацию. Переход «REFRESH REQUESTS» сбрасывает признак того, что запрос еще не получил ответ от ТС, и помещает вновь сформированные запросы в очередь запросов подсистемы.

Ответы ТС помещаются в буфер ответов. Переход «TURN OFF FLAG» на основе ответов, находящихся в этой позиции, сбрасывает для них признак того, что ответ ТС не получен. Таким образом, позиция «REFRESH REQUESTS» в ходе моделирования будет накапливать запросы, ответы для которых получены за интервал времени меньший, чем тайм-аут, а позиция «RESPONSE CLIENTQUEUE» будет аккумулировать просроченные ответы.

Время обработки сообщений ТС зависит от времени, затраченного ТС на обработку конкретного сообщения, а также от количества свободных потоков, выделяемых ТС для обработки сообщений. Выделяемые ТС потоки обработки сообщений образуют пул потоков ТС (позиция, в которой количество меток говорит о количестве свободных потоков). В случае непустой очереди запросов переход «CAPTURE THREAD» занимает один из свободных потоков и помещает его в позицию занятых (выполняемых) потоков «EXECUTED», откуда поток может быть изъят по истечении интервала времени, затрачиваемого на выполнение текущего задания. Информация о времени выполнения сообщений хранится в позиции «COMMAND WAIT». После освобождения потока он помещается в позицию «THREAD POOL», а ответ - в позицию «RESPONSE QUEUE». Таким образом проходит общий цикл получения информации.

Основными параметрами указанной модели являются количество приемников сообщений (выражается в количестве представлений), максимальное время ожидания ответа, задержка в сети передаваемых запросов, задержка в сети передаваемых ответов, количество запросов на каждое представление, время, затрачиваемое на обработку сервером

каждого запроса, количество выделенных сервером потоков для обработки запросов. Эти параметры можно определить на основе статистических данных, которые получены в ходе тестовой эксплуатации системы с помощью ведения журналов действий подсистем и журналов событий в системе. Эти же данные позволяют провести верификацию и аттестацию моделей после проведения имитационного моделирования.

Верифицированная модель представляет интерес для дальнейших исследований возможностей системы с точки зрения производительности. Например, для исследования пиковой нагрузки на ТС были установлены следующие параметры модели. Количество подсистем: 600 одновременных обращений (пиковая нагрузка) и затем 400 постоянных подключений; тайм-аут: 4 секунды; задержка в сети передаваемых запросов и ответов: 0,2-0,3 секунды; время, затрачиваемое ТС на обработку запросов: 0,3-1 секунда; количество выделенных ТС потоков для обработки запросов - 150. Результаты имитационного моделирования представлены в виде графиков, приведенных на рис. 1 - 4.

Можно отметить, что максимальная нагрузка ТС приходится на момент одновременного обращения 600 подсистем. При этом количество запросов, нуждающихся в обработке, составляет 1100. Пиковая нагрузка спадает на 3 секундах, работа ТС постепенно нормализуется, но часть запросов не была принята в течение 4 секунд после отправки, подсистемы, получившие сообщение о том, что информация не получена, обновляют представления, посылая еще 230 запросов (около 90 повторных обновленных представлений) ТС. Эти запросы создают очередную пиковую нагрузку на ТС, которая

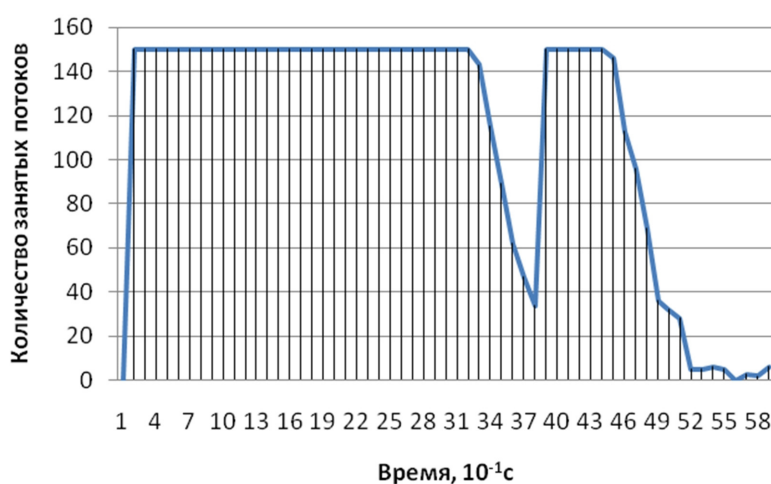


Рис. 1. Распределение количества занятых потоков ТС во времени

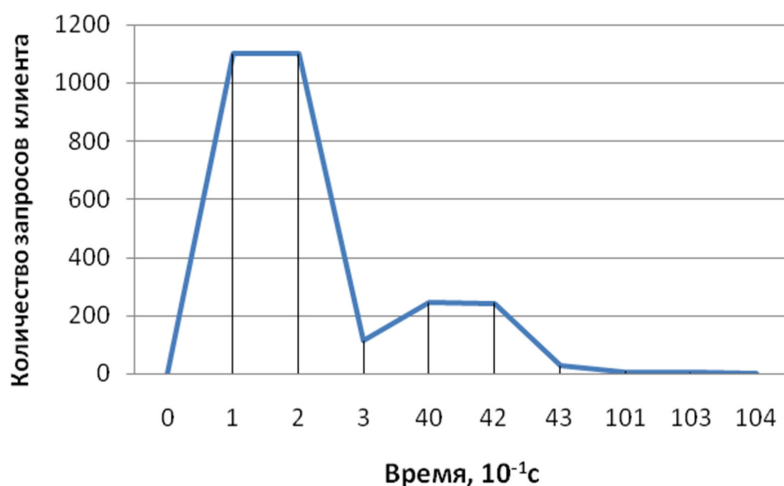


Рис. 2. Распределение количества запросов подсистемы во времени

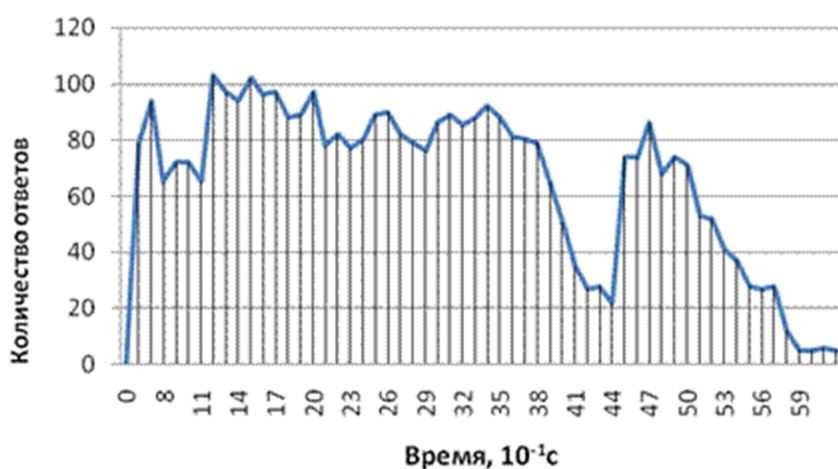


Рис. 3. Распределение количества ответов ТС во времени

продолжается 1 секунду, после работа ТС окончательно нормализуется. Все оставшееся время моделирования ТС занимала до 41 потока для обработки запросов, обеспечивая бесперебойную работу 400 подсистем.

Можно сделать вывод, что текущие настройки системы не обеспечивают бесперебойную работу при пиковой нагрузке (600 подсистем), система восстанавливается после таких нагрузок в течение 5 секунд. В обычном режиме (400 подсистем) система работает, используя не более 30% выделенных ресурсов.

Такие выводы после анализа результатов моделирования говорят о некоторых диспропорциях в конфигурировании системы и о неверно принятых решениях при ее проектировании. Так, например, второй пиковой нагрузки можно избежать при обновлении не всего представления, а только тех элементов, данные для которых не были получены. Это сократит количество вторично отправляемых запросов до 90. 150 выделенных ТС потоков не справляются с нагрузкой, т.к. 90 подсистем получают неполную информацию. Самым простым решением является увеличение максимального времени ожидания ответа. Тем не менее в обычном режиме ТС используется всего на 30%.

В этом случае можно уменьшить общее количество выделенных потоков, а пиковую нагрузку уменьшить за счет подключения средств кэширования на уровне запросов и кэширования на уровне выполнения. Такой подход позволит минимизировать время выполнения запросов за счет дополнительных затрат на ресурсы для хранения кэшированных запросов. Все перечисленные подходы можно опробовать на имеющейся РСП-модели и, варьируя различные параметры РСП-модели, добиться оптимальной работы системы с точки зрения производительности.

В качестве вывода, необходимо отметить, что имитационное моделирование обеспечивает большие возможности как при исследовании характеристик поведения существующей системы в различных режимах работы, так и при исследовании еще



Рис. 4. Зависимость количества занятых потоков от количества запросов подсистемы и количества ответов ТС

не реализованных решений, чтобы выбрать наиболее выгодные по заданным критериям (качество, время, цена) решения еще до практической реализации. С помощью имитационного моделирования обеспечивается предсказуемость производительности системы. В течение всего жизненного цикла информация о системе постоянно уточняется, что позволяет получать более точные результаты моделирования и соответственно более точную информацию о последствиях принимаемых решений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Шахов В.А. Моделирование программно-аппаратных реактивных систем раскрашенными сетями Петри. - RSDN Magazine. - 2006. - №3. - С. 3-7.
2. Гома Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений. Пер. с англ. - М.: ДМК Пресс. - 2002. - 704 с.

Поступила 18.07.2012