

ПОСТРОЕНИЕ СИСТЕМЫ АГРЕГИРОВАНИЯ ИНФОРМАЦИИ

Васюков А.Е., Крюков Д.А.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет информационных технологий, радиотехники и электроники» (МИРЭА), 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail: wasalev@mail.ru, dm.bk@bk.ru

Рассмотрен подход к построению систем агрегирования информации в контексте архитектуры «клиент-сервер», с целью поиска наиболее эффективной конфигурации системы предложена имитационная модель системы агрегирования информации. На основе полученной имитационной модели собраны экспериментальные данные и построена математическая модель системы.

Ключевые слова: горизонтально масштабируемая система, система агрегирования информации, имитационное моделирование.

BUILDING OF MASHUP INFORMATION SYSTEM

A. Vasyukov, D. Kryukov

Federal State Educational Institution of Higher Education “Moscow State University of Information Technologies, Radio Engineering and Electronics” (MIREA), 119454, Russia, Moscow, Vernadscogo avenue, 78 e-mail: wasalev@mail.ru, dm.bk@bk.ru

The paper considers a method of constructing information aggregation systems on the way of the client-server architecture. Proposed a simulation model of the information aggregation system with the goal the most efficient system configuration. Experimental data are collected and a mathematical model of the system is constructed based on the obtained simulation model.

Key words: horizontally scalable system, the system of agricultural information, math modeling.

В современном мире, где объемы данных, поступающие в информационные системы, имеют тенденцию к постоянному росту, а отдельное клиентское устройство зачастую не способно совершать выборку и обработку всех необходимых данных режиме реального времени, вопрос построения систем агрегирования информации встает все более остро.

Что же представляет из себя система агрегирования информации в сети Интернет? Согласно [4], система агрегации это способ создания новых web-приложений за счет объединения уже существующих интернет-ресурсов.

Зачастую, подобного рода системы проектируются как горизонтально масштабируемые системы [1] и служат надстройкой над API некоторых web-сервисов с целью получения больших выборок данных за время меньшее, чем если бы это делалось средствами отдельно взятого устройства [5]. Под масштабируемостью принято понимать способность системы увеличивать свою производительность при добавлении в систему ресурсов.

Различают понятия горизонтальной и вертикальной масштабируемости. Вертикальная масштабируемость – способность системы увеличить общую производительность при увеличении производительности отдельных ее элементов. Горизонтальная масштабируемость – способность системы увеличивать производительность при увеличении количества элементов, участвующих в обработке информации [3].

В данной статье содержатся положения касаясь построения модели взаимодействия компонентов горизонтально масштабируемой системы доступа к удаленным информационным ресурсам в контексте использования данной системы в качестве агрегирующего сервиса. На построенной модели проведено исследование оптимальной конфигурации горизонтально масштабируемого слоя системы. Данный подход позволяет в рамках одной модели отразить как взаимодействие пользователя с данной системой, так и взаимодействие отдельных компонентов системы между собой.

Системы агрегации информации (в западной технической литературе [4,5] устоялся термин «mashup-системы») строятся на основе горизонтально масштабируемых систем [2]. В рамках архитектуры «клиент-сервер» подобная система может соответствовать архитектуре, представленной на рисунке 1.

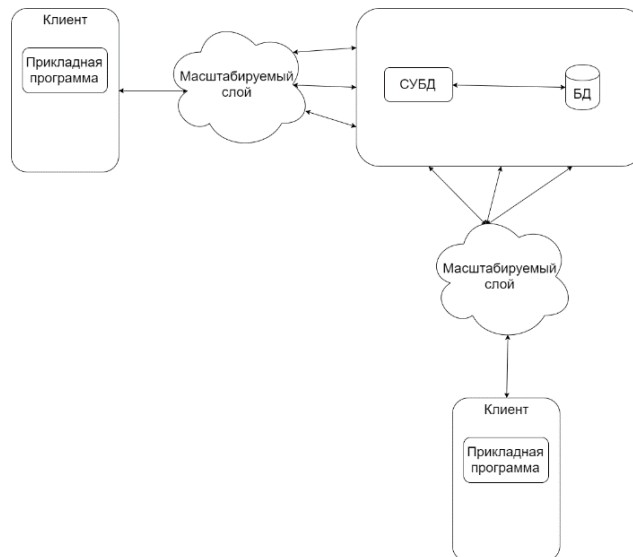


Рис. 1. Архитектура «клиент-сервер» с горизонтально масштабируемым слоем

Один из вариантов архитектуры горизонтально масштабированного слоя представлен на изображении (рис. 2). Пользователь формирует абстрактный запрос к контролеру горизонтально масштабированного слоя, затем контролер горизонтально масштабируемого слоя обрабатывает запрос, производя разложение на некоторое количество подзапросов (в случае необходимо), которые могут быть удовлетворены API сервиса и переданы в виде задач другим серверам системы. Связанные с контролером серверы выполняют запрос к удаленному серверу и, получив ответ, передают его контролеру горизонтально масштабируемого слоя, который объединяет ответы, полученные от подчиненных серверов, передает агрегированный результат клиенту.



Рис. 2. Архитектура горизонтально масштабируемого слоя

Для выявления устойчивой к заявленным нагрузкам конфигурации горизонтально масштабируемого слоя в среде имитационного моделирования AnyLogic разработана модель, представленная на рисунке 3. Модель представляет собой набор логических элементов. Элемент source выступает в качестве генератора запросов, delay и delay1 используются для моделирования временной задержки. Сущность resourcePool является хранилищем ресурсов (в данном случае количества потоков), сущность seize хранит заданное количество запросов и при достаточном количестве ресурсов, передает запрос далее.

В случае невозможности обработки запроса, запрос уничтожается в сущности sink. Сущность sink1 также используется для уничтожения запросов, по завершению работы цикла.

Для моделирования предложена ситуация, когда горизонтально масштабируемый слой используется в качестве сервиса, а интенсивность запросов, заложенная в source, составляет случайное значение согласно треугольному распределению [6] от 1 до 20 с наиболее вероятным значением, составляющим 5 запросов в минуту.

Моделирование включает три этапа:

1. Моделирование работы контроллера и удаленного сервера. В рамках этапа моделируется обработка

запроса контроллером и удаленным сервером, работа контроллера по разделению первичного запроса. Время на обработку запроса контроллером ($delay_1$) и удаленным сервером ($delay$) заложено как случайное значение согласно треугольному распределению от 0.5 до 5 с наиболее вероятным значением 1 секунда на обработку одного запроса. Блок `split` используется для моделирования работы контроллера по разделению первичного запроса к горизонтально масштабируемому слою на запросы, которые необходимо совершить к API удаленного сервиса. После разделения, оригинальный запрос перемещается в `sink`.

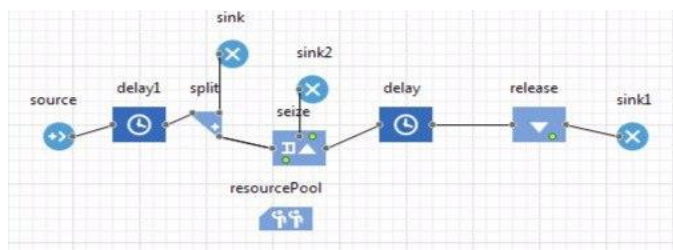


Рис. 3. Модель горизонтально масштабируемого слоя как сервиса

2. Моделирование работы горизонтально масштабированного слоя. Работа серверов горизонтально масштабируемого слоя, которым контроллер ставит в соответствие задачи, реализуется блоками `resourcePool`, `release` и `seize`. Элемент `resourcePool` представляет собой набор потоков, доступных в горизонтально масштабируемом слое для запроса к API удаленного сервера и рассчитывается как $Servers * 10$, где `Servers` это количество серверов в горизонтально масштабируемом слое используемых для решения задач, полученных от контроллера. Каждый сервер обладает потенциалом обработки 10 одновременных потоков, поэтому размер очереди блока `seize` рассчитывается аналогично.

3. Моделирование потоков. Освобождение потоков происходит после получения ответа от удаленного сервера. Возвращение ресурсов (свободных потоков) в `resourcePool` моделируется блоком `release`. Если блок `seize` не может обработать поступивший запрос или поместить его в очередь, то запрос поступает в `sink2`.

Для данной модели, приняты следующие допущения: оптимальным количеством серверов считается такое количество серверов в горизонтально масштабируемом слое, при котором не один из запросов, назначенных контроллером горизонтально масштабируемого слоя, не будет отклонен. Также, в рамках модели, все серверы в горизонтально масштабируемом слое считаются одинаково сконфигурированными и имеющими десять потоков.

В результате моделирования получена таблица, которая содержит значения оптимального количества серверов в горизонтально масштабируемом слое:

Таблица 1.

Коэффициент деления первичного запроса	Оптимальное количество серверов в горизонтально масштабируемом слое
10	2
50	10
100	20
500	87
1000	149

Как можно видеть из таблицы, оптимальное количество серверов в горизонтально масштабируемом слое нелинейно зависит от коэффициента изначального запроса. Проведем интерполяцию степенным многочленом. Известно, что через $n + 1$ точек на плоскости можно провести кривую, являющуюся графиком степенного многочлена (полинома) степени n , причем такой полином единственный. Воспользуемся следующей формулой:

$$L(x) = \sum_{i=0}^n y_i l_i(x)$$

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

За x_i примем коэффициент деления первичного запроса, за y_i – оптимальное количество серверов в горизонтально масштабируемом слое при заданной величине коэффициента деления первичного запроса.

Подставив значения в формулы получаем следующую зависимость:

$f(x) = 1.76571 \times 10^{-8}x^3 - 0.0000767385x^2 + 0.208204x - 0.122419$, где x это количество запросов, на которые будет разделен первичный запрос, а $f(x)$ – количество серверов, которое потребуется, чтобы все запросы были обработаны. Убедимся в корректности полученной для модели формулы на точках, не участвовавших в выявлении зависимости. Так, при разделении первичного запроса на 25 подзапросов получаем:

$$1.76571 \times 10^{-8}x^3 - 0.0000767385x^2 + 0.208204x - 0.122419 \approx 5.035.$$

Проведя моделирование при наличии пяти серверов, получающих задания от контролера горизонтально масштабированного слоя, убеждаемся в том, что модель работает корректно и все запросы были обработаны. При расщеплении изначального запроса на 127 запросов получаем:

$1.76571 \times 10^{-8}x^3 - 0.0000767385x^2 + 0.208204x - 0.122419 \approx 25.1179$. Проведя моделирование на двадцати пяти серверах, получающих задания от контролера горизонтально масштабированного слоя, убеждаемся в том, что модель работает корректно и все запросы были обработаны.

Таким образом, в работе построена аппроксимирующая функция для системы агрегирования информации и математическая зависимость коэффициента деления первоначального запроса и количества серверов в горизонтально масштабируемом слое, необходимого для их обработки. Использование данной модели и математической зависимости позволит оценить устойчивость подобных систем к заявленным нагрузкам до непосредственного введения их в эксплуатацию.

Список литературы

-
1. Косяков М.С. Введение в распределенные вычисления / Учебное пособие. – Санкт-Петербург: НИУ ИТМО, 2014. - 155 с.
 2. Радченко Г.И. Распределенные вычислительные системы / Г.И. Радченко. – Челябинск: Фотохудожник, 2012. – 184 с.
 3. Шамакина А. В. Обзор технологий распределенных вычислений // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2014. №3 С.51-85.
 4. Baker M., Buyya R., Laforenza D. Grids and Grid Technologies for Wide-area Distributed Computing // Journal of Software-Practice & Experience. 2002. Vol. 32, No. 15. P. 1437–1466
 5. Benslimane D., Dustdar S., Sheth A. Services Mashups: The New Generation of Web Applications // IEEE Internet Computing. 2008. Vol. 12. doi: 10.1109/MIC.2008.110
 6. Help - AnyLogic Simulation Software [Электронный ресурс] / triangular. – <https://help.anylogic.ru/index.jsp?topic=%2Fcom.anylogic.help%2Fhtml%2Ffunctions%2Ftriangular.html> (дата обращения: 20.12.2017)

References

-
1. Kosyakov M.S. Introduction to distributed computing / Tutorial. – St. Petersburg: Research University of ITMO, 2014, p. 155
 2. Radchenko G.I. Distributed Computing Systems / G.I. Radchenko, Chelyabinsk : Photographer, 2012, p. 184
 3. Shamakina A.V. Overview of distributed computing technologies / Bulletin of South Ural State University. Series: Computational Mathematics and Computer Science, 2014, no. 3 pp. 51-85
 4. Baker M., Buyya R., Laforenza D. Grids and Grid Technologies for Wide-area Distributed Computing // Journal of Software-Practice & Experience. 2002. Vol. 32, No. 15. P. 1437–1466
 5. Benslimane D., Dustdar S., Sheth A. Services Mashups: The New Generation of Web Applications // IEEE Internet Computing. 2008. Vol. 12. doi: 10.1109/MIC.2008.110
 6. Help - AnyLogic Simulation Software / triangular available at: <https://help.anylogic.ru/index.jsp?topic=%2Fcom.anylogic.help%2Fhtml%2Ffunctions%2Ftriangular.html> (accessed: 20 December 2017)