после сбоя, обеспечивающие незначительное снижение атрибутов масштабируемости и производительности.

Современная разработка программных систем зачастую сводится к решению одной из следующих задач: ручной или машинный сбор данных, очистка и преобразование данных, анализ данных. Одной из задач, которую необходимо решать в рамках всех указанных задач является проверка качества и целостности данных. Так, как ручной, так и машинный ввод данных может вносить ошибочные данные. Процессы очистки и преобразования данных так же могут вносить значительные ошибки, которые, однако зачастую могут быть обнаружены только на этапе финального анализа данных.

Следует заметить, что проверка качества данных в зависимости от методов может занимать значительное время, при этом, длительность проверки может сильно варьироваться от нескольких секунд, до нескольких часов (при этом, в зависимости от предметной области, не всегда можно дать априорную оценку длительности такого анализа).

Исходя из указанных выше требований, одним из способов решения задач такого долговременного и сложно прогнозируемого анализа может выступать разработка распределенных асинхронных систем на основе архитектурной модели сервисной шины (в английской литературе EnterpriseServiceBus, ESB). Такая архитектура позволяет обеспечить необходимый уровень производительности системы вцелом за счет легкой горизонтальной масштабируемости (в терминах облачных вычислений - эластичности) системы, когда по запросу в систему могут легко быть добавлены новые вычислительные мощности. Так, вычислительный кластер может быть резко увеличен с 3-4 вычислительных узлов, до 30-40 в случае пиковых нагрузок.

Однако такой подход в значительной степени усложняет процедуры поддержки системы вцелом, при этом значительно повышая вероятность сбоя того или иного вычислительного узла. Поэтому задачи мониторинга, централизованного отслеживания состояния всех вычислительных узлов системы, возможности централизованного обновления конфигураций всех элементов системы становятся одними из определяющих при разработке таких систем. Под централизованностью функционала управления системой при этом не подразумевается наличие единых точек отказа системы (singlepointoffailure), наличие которых резко бы понизило атрибуты отказоустойчивости и высокой доступности системы.

В качестве одной из наиболее интересных задач, подлежащих решению при разработке таких систем можно выделить задачу централизованного обновления конфигураций системы «на лету». Необходимость отсутствия единой точки отказа ограничивает возможность добавления единого узла конфигурации, которых бы хранил все системные настройки. При этом, все другие элементы системы: файловая система, база данных, очередь не могут выступать в качестве хранилища указанных настроек конфигурации, так как параметры доступа к ним сами являются настройками конфигурации. Оптимальным способом организации таких конфигураций нам видится реализация подхода master-slave с автоматическим выбором нового master-а из доступных slave при отказе предыдущего master-а.

Таким образом, задачи мониторинга состояния сильно распределенных систем являются ключевым функционалом, в значительной мере упрощающим процедуры поддержки работоспособности системы. При этом, добавление указанного функционала не должно в свою очередь снижать производительность, масштабируемость, отказоустойчивость и доступность системы.

Список использованных источников:

1. Басс Л. Архитектура программного обеспечения на практике. 2-е издание. / Басс Л., Клементс П., Кацман Р. СПб.: Питер, 2006. – 575 с.: ил.

АЛГОРИТМ СОЗДАНИЯ АВТОМАТИЧЕСКОГО АГЕНТА ДЛЯ ПОДДЕРЖКИ ПОЛЬЗОВАТЕЛЕЙ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Терещук А.В.

Серебряная Л.В. − к. т. н., доцент

В современном мире интернет оказывает все большее влияние на жизнедеятельность человека. Создается множество сложных программных продуктов, которые доступны клиентам по глобальной сети. Порог вхождения в такие системы довольно высок, поэтому для его снижения используются разные подходы. Одним из решений является поддержка пользователей в режиме онлайн чата. Для снижения затрат и увеличения прибыли поддержку пользователей можно организовать с использованием автоматического агента. В данной работе описан алгоритм реализации такого агента.

Целью применения автоматизации поддержки пользователей является уменьшение количества рутинных и однообразных действий, которые выполняют работники службы поддержки. Это достигается за счет того, что для решения большого количества типичных проблем создаются определенные ответы, которые передаются пользователям в соответствии с их запросами. Для соотнесения запросов и ответов подойдет алгоритм латентно-семантического анализа.

Латентно-семантический анализ отображает документы и отдельные слова в так называемое

«семантическое пространство», в котором и производятся все дальнейшие сравнения. В контексте поставленной задачи документом будет являться ответ, а слова – извлекаться из запроса пользователя. На первом шаге необходимо составить частотную матрицу индексируемых слов. В этой матрице строки соответствуют индексируемым словам, а столбцы – ответам. В каждой ячейке указано, какое количество раз слово встречается в соответствующем ответе. На данном шаге необходимо учитывать при подсчете слова-синонимы. Это позволит значительно улучшить исходный алгоритм, однако для этого требуется иметь в наличии словарь синонимов. На следующем этапе мы проводим сингулярное разложение полученной матрицы. Это математическая операция разложения матрицы на три составляющие:

$$M = U * W * V^t.$$

Где U и V^t – ортогональные матрицы, а W – диагональная. Диагональные элементы матрицы W отсортированы в порядке убывания. Диагональные элементы матрицы W называются сингулярными числами. Основное достоинство сингулярного разложения заключается в том, что оно позволяет выделить ключевые элементы матрицы и проигнорировать шумы. Поскольку меньшие сингулярные числа вносят незначительный вклад в итоговое произведение, размерность матрицы можно уменьшить. Единого решения в выборе результирующей размерности нет, т.к. при маленькой размерности снижается возможность обнаруживать семантические группы, а при большой размерности на результат начинают оказывать влияние шумы. После того как матрица была сформирована, необходимо найти наименьшее расстояние от набора слов запроса до ответов. Значениями слов для представления их в многомерном пространстве будут являться строки матрицы U, а ответов – столбцы матрицы V^t. Поскольку пространство отображения зачастую получается многомерным, выбор способа вычисления расстояния становится очень важным для производительности алгоритма. Среди всех вариантов было решено использовать манхэттенское расстояние, т.к. оно обладает высокой точностью и низкой сложностью алгоритма.

Применение данного алгоритма позволяет реализовать автоматического агента, который будет способен решить большое количество проблем, ответить на многие вопросы пользователей, уменьшить нагрузку на службу поддержки, в которой задействованы люди.

Список использованных источников:

1. Landauer T. K., Foltz P. W., Laham, D. Introduction to Latent Semantic Analysis // Discourse Processes. 1998. No 25. P. 259–284
2. Golub G. H., Luk F. T., Overton M. L. A block Lanczoz method for computing the singular values and corresponding singular vectors of a matrix // ACM Transactions on Mathematical Software 1981. No 7. P. 149–169.

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ MAPREDUCE ДЛЯ ОРГАНИЗАЦИИ РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ СИГНАЛЬНЫХ ДАННЫХ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Казимирчик Д.В.

Бранцевич П.Ю. − к. т. н., доцент

Обработка больших массивов экспериментальных данных с использованием стандартных алгоритмов и методов может занимать достаточно продолжительное время. В таких случаях представляется целесообразным использование методов, которые позволяют ускорить обработку данных распределяя задачи обработки между множеством машин.

Сегодня стоимость хранения информации настолько низка, что зачастую представляется целесообразным постоянное накопление “сырых” экспериментальных данных получаемых от датчиков, систем мониторинга событий и т.д. в реальном времени. В будущем над собранными в процессе наблюдения данными можно проводить различные виды анализа для получения наиболее полной картины происходящего, выявления закономерностей и аномалий. В результате этого появляется необходимость в организации эффективной и быстрой обработки таких данных. Так как ресурс повышения производительности отдельно взятых процессоров давно исчерпан, постоянно ведётся поиск путей ускорения вычислительных процессов с использованием методов распараллеливания и распределения обработки данных между множеством машин.

MapReduce – это модель организации вычислений предназначенная для использования при обработке и генерации больших объёмов данных. При использовании MapReduce выбирается map-функция, которая обрабатывает исходную пару параметров ключ/значение и в результате генерирует промежуточную пару ключ/значение, и reduce-функция, которая объединяет все промежуточные значения ассоциированные с одним промежуточным ключом [1].

Модель MapReduce может быть применена при обработке большого количества сигнальных данных для вычисления таких параметров сигнала, каксреднее квадратическое значение (СКЗ), пик-фактор, размах колебаний, а также для проведения более сложных видов анализа и обработки. Базовым подходом при реализации MapReduce в этом случае является разбиение массива сигнальных значений на части, которые в дальнейшем будут отданы на обработку множеству независимо исполняемых map-функций, которые