

УДК 371.69:004.738

ПОВЫШЕНИЕ СКОРОСТИ ОБРАБОТКИ ДАННЫХ В СИСТЕМАХ ЭЛЕКТРОННОГО ОБУЧЕНИЯ



А.Д. Тюменцев
Инженер-программист



П.В. Камлач
Доцент кафедры ЭТТ
БГУИР, кандидат
технических наук,
доцент



В.М. Бондарик
Декан ФДПиПО
БГУИР, кандидат
технических наук,
доцент



В.И. Камлач
Ассистент
кафедры ИПиЭ
БГУИР

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: lexatum@gmail.com, kamlachpv@bsuir.by, bondarik@bsuir.by

А.Д. Тюменцев

Окончил Белорусский государственный университет информатики и радиоэлектроники, специальность «Программное обеспечение информационных технологий». Работает инженером-программистом. Область научного интереса – обработка данных с использованием технологий Big Data.

П.В. Камлач

Окончил Белорусский государственный университет информатики и радиоэлектроники, специальность «Медицинская электроника». Доцент, кандидат технических наук, доцент кафедры электронной техники и технологии БГУИР. Область научного интереса – проектирование медицинских электронных систем.

В.М. Бондарик

Декан факультета доуниверситетской подготовки и профессиональной ориентации БГУИР. Окончил с отличием Минский радиотехнический институт, специальность «Конструирование и производство радиоаппаратуры», квалификация - инженер-конструктор-технолог радиоаппаратуры, кандидат технических наук, доцент. Область научного интереса – проектирование медицинских электронных систем, внедрение дистанционных образовательных технологий.

В.И. Камлач

Окончила магистратуру Белорусского государственного университета информатики и радиоэлектроники, специальность «Психология труда, инженерная психология, эргономика». Ассистент кафедры инженерной психологии и эргономики БГУИР. Область научного интереса – исследование психофизиологического состояния человека.

Аннотация. Описан один из способов повышения скорости работы системы электронного обучения за счет масштабирования базы данных с использованием модели MapReduce, снижения нагрузки на сервера и повышения производительности системы в целом. Рассмотрены положительные и отрицательные аспекты предложенного подхода к построению системы электронного обучения.

Ключевые слова: система электронного обучения, масштабирование, конвейер, большой объём данных, база данных, производительность.

Введение. В настоящее время электронное образование во всем мире активно развивается. При этом организация учебного процесса невозможна без использования систем электронного обучения (СЭО). Использование СЭО предполагает анализ большого потока информации, поступающего в информационную среду электронного обучения от участников

образовательного процесса. Существует множество программных приложений для интеллектуального анализа больших данных, которые можно использовать для классификации, кластеризации, регрессионного и сетевого анализа учебной информации. Их применение в электронном образовании позволит педагогам своевременно получать информацию об обучающихся, оперативно реагировать на любые изменения процесса обучения, своевременно вносить изменения в учебный контент.

В СЭО существует большое количество различных типов данных, как структурированных, так и неструктурированных, обработку которых трудно осуществить традиционными статистическими методами. Современные СЭО используют данные различных типов (мультимедийный контент, системы тестирования, таблицы данных и т.п.), расположенные на различных сетевых ресурсах, в том числе на удаленных. В периоды пиковой активности обучающихся возможно значительное замедление работы СЭО, что снижает эффективность образовательного процесса. Для разработки и внедрения успешных систем электронного обучения необходимо использовать новые технологии, которые позволили бы хранить и обрабатывать большие потоки данных. Использование традиционных методов анализа информации не всегда дает возможность контролировать огромные объемы постоянно растущих и обновляемых данных. Применение технологий Big Data позволяет эффективно обрабатывать структурированные и неструктурированные данные больших объемов и значительного многообразия [1].

Результаты. Для повышения скорости обработки данных в СЭО предложено воспользоваться горизонтально масштабируемыми программными инструментами. Горизонтальное масштабирование предполагает увеличение количества вычислительных ресурсов за счет увеличения количества серверов. Если при работе с приложением пользователь обратился к большому объему данных и вычислительный сервер не справляется с нагрузкой, то время отклика может заметно увеличиться. Для решения проблемы быстрого доступа к большим объемам информации предложена модель MapReduce.

MapReduce – это модель распределенной обработки данных, предложенная компанией Google для обработки больших объемов данных на компьютерных кластерах [2].

Данные в СЭО организованы в виде некоторых записей, поэтому их обработка происходит в две стадии:

1. Стадия Map. На этой стадии данные предобрабатываются при помощи функции Map. Работа этой стадии заключается в предобработке и фильтрации данных. Функция Map, примененная к одной входной записи, выдает множество пар ключ-значение. Множество может выдать только одну запись, может не выдать ничего, а может выдать несколько записей.

2. Стадия Reduce. На данной стадии данные, полученные из функции Map, будут обработаны и возвращены в соответствии с изначально сформулированной задачей.

Модель MapReduce может быть применена при разработке приложений для систем электронного обучения, позволяющих автоматизировать процесс усвоения знаний, текущей, промежуточной и итоговой аттестации с заполнением соответствующих документов. Эффективность использования предложенной модели при создании и модернизации СЭО обусловлена большим количеством данных, которые необходимо фильтровать и возвращать, что занимает довольно значительную часть времени при обработке запроса [3].

При реализации проекта на стороне WCF-сервисов был создан класс, в котором реализованы две функции map и reduce. Функция map принимает в качестве параметров: искомое значение – ключ и коллекцию данных, которую необходимо обработать. Коллекция данных имеет динамический тип, поэтому может быть использована для любого набора данных. На выходе функция возвращает коллекцию ключ-значение. Далее данные передаются в функцию Reduce. Данная функция реализует в себе фильтрацию данных в зависимости от поставленной задачи. В нашем случае – это разбиение данных на части для корректного представления их в таблице, применение фильтров, выбранных в таблице, поиск заданного

значения. Использование функций `map` и `reduce` уменьшило время обработки запроса практически вдвое.

Для асинхронного использования функций `map` и `reduce` реализован конвейер `asp.net` с помощью средств виртуальных серверов IIS. Встроенный веб-сервер IIS очень легко масштабируется. Он использован в качестве механизма конвейера общего назначения, доступного через `http`. Шаги конвейера `asp.net` выполняются последовательно, но каждый шаг может выполняться асинхронно. Веб-сервер IIS можно настроить для запуска нескольких конвейеров `asp.net`, обслуживающих `http`-запросы. Исходя из этого наши MapReduce запросы могут выполняться асинхронно, тем самым снижая время на обработку данных.

Для реализации асинхронной обработки данных созданный нами класс был перемещён в библиотеку, которую можно использовать во всех проектах, а WCF-сервис был помещен на несколько пулов (процессов `w3wp.exe`). Библиотека была подключена как модуль, вызываемый с помощью атрибута при обработке определенных `rest`-запросов, которые мы сами указываем при определении сигнатур функций для сервисов. Этого достаточно, чтобы запустить конвейер `asp.net` и обрабатывать запросы асинхронно, разделяя данные между несколькими процессами и агрегируя их уже на уровне бизнес-логики. Чтобы добиться полноценной работы модели MapReduce необходимо реализовать так называемую «сетку». Для этого было добавлено ещё несколько виртуальных серверов. Чем больше «сетка», тем серьезней проблему можно решить при обработке данных, разбивая её на более мелкие составляющие, которые нужно решить, и тем выше уровень параллелизма, который потенциально может быть достигнут. Асинхронный конвейер `asp.net`, объединенный с несколькими конвейерами на сервер, обеспечивает параллелизм в ядрах одного сервера. Поскольку серверы IIS – это просто автономные серверы, дополнительная настройка не требуется. Ещё одно преимущество архитектурной серверной «сетки» заключается в том, что она не полагается на главный узел для работы. В таких продуктах, как `Nadoop`, главный узел управляет кластером серверов и расположением данных в этом кластере. И это главный узел, который был бы источником сбоя. В предложенной нами структуре в «сетке» серверов нет главного узла. Любой серверный узел может инициировать запрос MapReduce [4].

После настройки конвейера, время на обработку запроса было снижено вдвое.

Основное преимущество применения MapReduce – возможность наращивать количество виртуальных серверов, а не закупать новое более дорогое оборудование для одного основного узла. Соответственно, при сбое любого из серверов, обработка данных может быть передана на любой другой сервер при условии, что входные данные для проводимой операции доступны.

Одним из достоинств проекта является масштабируемость, полученная с использованием конвейера `asp.net` в качестве конвейера MapReduce. Поскольку конвейер `asp.net` работает последовательно, он подходит для выполнения шагов `Map` и `Reduce`. Преимуществом предложенной архитектуры является то, что хоть последовательный конвейер и не будет двигаться к следующему шагу, пока не завершится предыдущий шаг, но каждый шаг может выполняться асинхронно. Это позволяет конвейеру продолжать получать и обрабатывать новые запросы MapReduce, даже когда он заблокирован, ожидая, что вызовы `Map` возвратятся с других распределенных серверных узлов.

Заключение. Использование функции MapReduce при построении СЭО позволило увеличить скорость обработки данных приблизительно в четыре раза без каких-либо серьезных изменений в архитектуре проекта и без значительных вложений в серверное оборудование. Применение предложенной технологии в электронном образовании позволяет достигнуть гибкости, масштабируемости, доступности, качества обслуживания и простоты использования информации.

Список литературы

- [1.] Бондарик, В.М. Увеличение скорости обработки данных в системе электронного обучения / В.М. Бондарик [и др.] // Дистанционное обучение – образовательная среда XXI века : материалы XI Междунар. науч.-метод. конф. (Республика Беларусь, Минск, 12–13 декабря 2019 года) / редкол. : В. А. Прытков [и др.]. – Минск : БГУИР, 2019. – С. 70.
- [2.] Горизонтальное масштабирование базы данных реального проекта с помощью SQL Azure Federations [Электронный ресурс] – Режим доступа: <https://msdn.microsoft.com/ru-ru/dn458578.aspx>.
- [3.] Принцип работы с большими данными, парадигма MapReduce [Электронный ресурс] – Режим доступа: <https://msdn.microsoft.com/en-us/magazine/mt147240.aspx>.
- [4.] Тюменцев, А.Д. Масштабирование базы данных с использованием модели Mapreduce / А.Д. Тюменцев [и др.] // BIG DATA and Advanced Analytics. BIG DATA и анализ высокого уровня : материалы IV Междунар. науч.-практич. конф. (Республика Беларусь, Минск, 3-4 мая 2018 года) / редкол. : М. П. Батура [и др.]. – Минск : БГУИР, 2018. – С. 211-213.

INCREASING DATA PROCESSING SPEED IN E-LEARNING SYSTEMS USING THE MODEL MAPREDUCE

A.D. TUMENTSEV
Software Engineer

P.V. KAMLACH,
PhD
*Associate Professor of
the Electronic
Technology and
Engineering BSUIR*

V.M. BONDARIK,
PhD
*Dean of the faculty of
pre-university training
and vocational
guidance BSUIR*

V.I. KAMLACH,
*Assistant, Department
of Engineering
Psychology and
Ergonomics BSUIR*

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
Email: lexatum@gmail.com, kamlachpv@bsuir.by, bondarik@bsuir.by

Abstract. A method is proposed for increasing the speed of the e-learning system by scaling the database using the MapReduce model. The MapReduce model reduces server load and improves overall system performance. The positive and negative aspects of using the MapReduce model in the construction of e-learning systems are considered.

Keywords: e-learning system, big data, scaling, conveyor, database, performance.