Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.453

Удовин
Иван Анатольевич

Модель распределенных вычислений MapReduce в режиме реального времени

АВТОРЕФЕРАТ
на соискание степени магистра
по специальности 1-40 80 04 – Информатика и технологии программирования

Научный руководитель
Волорова Н. А.
к.т.н., доцент

Минск 2022

ВВЕДЕНИЕ

Сегодня способность обрабатывать «большие данные» стала критически важной для информационных потребностей многих предприятий, научных приложений и правительств. В последнее время возрастает потребность в обработке данных не только «больших», но и «быстрых». Здесь «быстрые данные» относятся к высокоскоростным потокам данных в режиме реального времени или почти реального времени, таким как ленты Twitter, потоки поисковых запросов, потоки кликов, показов и системные журналы. Для обработки как исторических данных, так и данных в реальном времени многим компаниям приходится поддерживать несколько систем. Однако недавние практические исследования показывают, что обслуживание нескольких систем приводит не только к дублированию кода, но и к интенсивной ручной работе по разделению аналитических рабочих нагрузок и определению того, какие данные обрабатываются какой системой. Эти проблемы указывают на потребность в общей унифицированной структуре обработки данных с различными требованиями к задержкам.

Модель распределенных вычислений MapReduce, представленная компанией Google, зарекомендовала себя как эффективное решение задачи обработки больших объемов данных. Она позволяет построить масштабируемое решение, способное непрерывно работать даже в случае возникновения множественного отказа машин в кластере. В то же время, классическое представление данной модели основано на предположении, что обрабатываемые данные заранее известны и доступны целиком. Однако в реальном мире, данные в большинстве случаев не доступны в полном объеме, более того, новые данные могут поступать непрерывно, в реальном времени. В такой ситуации классическая интерпретация модели MapReduce не способна обрабатывать непрерывные потоки данных и требует некоторой доработки.

Таким образом, применение методов обработки больших объемов данных в реальном времени является перспективным направлением, на что направлено основное внимание данной диссертационной работы.

В данной работе описаны методы обработки больших объемов данных, а также описано построение комплексной системы на их основе. Разработаны алгоритмы обработки больших объемов данных в реальном времени с использованием модели распределенных вычислений MapReduce. Разработанная система была развернута в облачной среде и протестирована на задаче Word Count.

Магистерская диссертация проверена в системе «Атиплагиат». Процент оригинальности соответствует норме, установленной кафедрой информатики. Цитирования обозначены ссылками на публикации, указанные в «Списке использованных источников».

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

Целью диссертационной работы является изучение модели распределенных вычислений для обработки больших данных MapReduce, алгоритмов и архитектур, основанных на данной модели, а также разработка программной системы, позволяющей использовать модель MapReduce на данных, поступающих в реальном времени.

Для достижения поставленной цели необходимо решить следующие задачи:

а) Исследовать модель распределенных вычислений MapReduce.

б) Проанализировать алгоритмы обработки больших объемов данных, лежащие в основе MapReduce.

в) Проанализировать алгоритмы обработки потоков данных, поступающих в реальном времени.

г) На основании произведенного анализа, выбрать необходимые технологии и инструменты для решения поставленной задачи.

д) Обобщить и адаптировать модель распределенных вычислений MapReduce для случая обработки данных в реальном времени.

е) Спроектировать архитектуру и разработать программную систему обработки больших объемов данных, поступающих в реальном времени с использованием модели MapReduce.

ж) Спроектировать модель базы данных.

з) Развернуть систему в облачном сервисе.

Объектом исследований в представленной диссертации являются распределенные вычислительные среды.

Предметом исследования являются модели и методы распределённых вычислений для обработки больших объемов данных в реальном времени, основанные на модели распределенных вычислений MapReduce.

Основной гипотезой, положенной в основу диссертационной работы, является возможность построения программной системы обработки больших объемов данных в реальном времени на основе модели распределенных вычислений MapReduce. Построенная система разворачивается в облаке, что повышает её доступность и позволяет использовать её в режиме реального времени, непрерывно обрабатывая поступающие данные.

Связь работы с приоритетными направлениями научных исследований и запросами реального сектора экономики

Работа выполнялась в соответствии научно-техническими заданиями и планами работ кафедры «Информатика».

Личный вклад соискателя

Результаты, приведенные в диссертации, получены соискателем лично.

Вклад научного руководителя Волоровой Н. А. заключается в формулировке целей и задач исследования.

Структура и объем диссертации

Диссертация состоит из введения, общей характеристики работы, четырех глав, заключения, списка использованных источников и приложений. В первой главе представлен обзор предметной области, дан обзор распределенных систем, модели распределенных вычислений MapReduce, а также осуществлен обзор существующих аналогов систем real-time MapReduce. Вторая глава посвящена изучению Docker-контейнеров, системы оркестрации Kubernetes, а также изучению базы данных PostgreSQL и распределенной файловой системы CephFS. В третьей главе подробно разобран процесс проектирования и разработки программной системы real-time MapReduce. В четвертой главе описан процесс подготовки инфраструктуры программной системы и её развёртывания в облачном сервисе.

Общий объем работы составляет 80 страниц, 23 рисунка, список использованных источников из 14 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** определена область и указано основное направление исследования, показана актуальность темы диссертационной работы, рассмотрены основные проблемы классической модели распределенных вычислений MapReduce, обозначены методы решения поставленной задачи, обозначена практическая ценность работы.

В **первой главе** произведён обзор предметной области задачи, решаемой в рамках диссертационной работы; рассмотрены вопросы о сущности распределенных систем обработки больших объемов данных, в том числе, поступающих в реальном времени.

Распределенные системы имеют специфические характеристики которые не присущи обычным системам. К ним относятся:

– параллельность – программные компоненты, выполняющие распределенную обработку, могут работать параллельно;

– независимые отказы – аппаратные и программные компоненты могут отказывать независимо друг от друга;

– отсутствие глобального времени – у каждого компонента распределенной системы имеются свои часы, которые могу показывать разное время;

– коммуникационные задержки – на передачу данных между компонентами распределенной системы необходимо определенное время;

– несогласованное состояние – компоненты могут быть в разных состояниях (сна, занятости и т.д).

Также, была рассмотрена сущность модели MapReduce — это модель и связанная с ней реализация системы для обработки и создания больших объемов данных, предложенная разработчиками из Google в 2004 году.

Если говорить о модели программирования MapReduce, то система принимает на вход наборы пар ключ-значение, и создает выходной набор пар ключ-значение. Пользователь системы представляет вычисления в качестве двух функций Map и Reduce:

– Функция Map (операция Map), написанная пользователем, принимает на вход пару ключ-значение и создает набор промежуточных пар ключ-значение. Система MapReduce группирует все промежуточные значения, связанные одним и тем же промежуточным ключом, и передает их в функцию Reduce.

– Функция Reduce, так же написанная пользователем, принимает промежуточный ключ и набор значений для этого ключа. Она объединяет данные значения, чтобы сформировать объединенный набор пар ключ-значений (обычно меньшего размера). Зачастую используется лишь ноль или одно выходное значение для каждого вызова функции Reduce. Промежуточные значения пере-

даются пользовательской функции Reduce посредством итератора. Это позволяет обрабатывать списки значений, которые занимают слишком много места, чтобы поместиться в памяти.

Во **второй главе** рассматриваются технология запуска процессов в изолированной среде, система управления кластерами Kubernetes, особенности базы данных PostgreSQL и методика ее развертывания для достижения высокой доступности с использованием технологии Stolon.

В качестве технологии запуска процессов в изолированной среде используется Docker, который для своей работы инициализирует следующие пространства имен операционной системы Linux:

– пространство `PID` – для изоляции дерева процессов;

– постранство `NET` – для изоляции сети и ограничения доступа к сетевым интерфейсом;

– пространство `IPC` – для управления доступом к ресурсам межпроцессного взаимодействия (Inter Process Communication);

– пространство `MNT` – для изоляции дерева файловой системы;

– пространство `UTC` – для изоляции ядра и идентификаторов версий (Unix Timesharing System).

При рассмотрении системы управления кластерами Kubernetes, были рассмотрены следующие главные компоненты:

– `kube-apiserver` – компонент, предоставляющий API Kubernetes. Это `front-end` компонент для панели управления Kubernetes. Он предназначен для горизонтального масштабирования – то есть масштабирования путем развертывания большего количества экземпляров;

– `etcd` – согласованное и высокодоступное хранилище ключ-значение, используемое в качестве резервного хранилища Kubernetes для всех данных кластера;

– `kube-scheduler` – компонент, который наблюдает за вновь созданными подами, не имеющими назначенного узла, и выбирает узел для их запуска;

– `kube-controller-manager` – компонент, который запускает контроллеры. Логически каждый контроллер является отдельным процессом, но для уменьшения сложности все они компилируются в один бинарный файл и запускаются в одном процессе.

– `cloud-controller-manager` – запускает контроллеры, взаимодействующие с базовыми поставщиками облака.

Третья глава посвящена проектированию архитектуры и разработке программной системы для обработки больших объемов данных.

При проектировании архитектуры необходимо было учитывать главные требования к системе: масштабируемость, отказоустойчивость. Для достиже-

ния необходимой масштабируемости, система была разделена на следующие компоненты:

– Работник (далее Worker) — компонент, выполняющий операции map и reduce.

– Планировщик (далее Planner) — компонент, осуществляющий распределение операций по Worker-нодам.

– Веб-интерфейс (далее Interface) — компонент, осуществляющий доступ к системе с помощью пользовательского интерфейса.

– База данных.

– Распределенная файловая система.

Далее рассматриваются особенности работы с таблицами, операциями Map и Reduce. Приводятся примеры реализации операций Map и Reduce. Приводятся особенности устройства и реализации каждого компонента.

В **четвертой главе** описывается инфраструктура программной системы. Подробно описывается порядок развертывания кластера Kubernetes с использованием K3s на виртуальных машинах облачной платформы.

K3s — это полностью совместимый дистрибутив Kubernetes со следующими улучшениями:

– Упакован в один исполняемый бинарный файл.

– Поддержка сервера хранения etcd3.

– Обернут в простой лаунчер, который справляется со многими сложностями TLS и различных опций.

– Безопасный по умолчанию с разумными значениями по умолчанию для облегченных сред.

– Наличие локальный поставщика хранилища, балансировщика нагрузки служб, контроллера Helm и входного контроллера Traefik.

– Работа всех компонент плоскости управления Kubernetes инкапсулирована в один двоичный файл. Это позволяет K3s автоматизировать и управлять сложными кластерными операциями, такими как распространение сертификатов.

– Внешние зависимости сведены к минимуму (требуется только современное ядро и монтирование cgroup).

Далее описывается порядок развертывания и настройки файловой системы CephFS, а также настройка базы данных PostgreSQL, необходимой для хранения состояния кластера.

После настройки и запуска всех необходимых компонентов, необходимо развернуть систему real-time MapReduce.

В заключительной части четвертой главы производится тестирование работоспособности полученной системы.

Для тестирования работоспособности системы был построен простейший план выполнения для решения задачи Word Count. Решение задачи было получено с помощью следующей последовательности операций:

а) Операция Map, генерирующая строки с текстами, состоящими из случайного набора слов. Каждый текст состоит из 300 случайных слов, соединенных пробелом.

б) Операция Map, получающая на вход сгенерированные строки и осуществляющая разделение текста на слова. После разделения текста, подсчитывается количество вхождений слова в заданную строку и записывается пара из слова и количества вхождений данного слова.

в) Операция Reduce по ключу слова, получающая на вход пары слов и количеств вхождений слов и записывающая на выходе кумулятивную сумму вхождений каждого слова.

Нагрузочное тестирование системы осуществлялось при разных настройках операций и таблиц. На рисунке 1 представлены графики скорости записи при разделении всех таблиц на одну, две и четыре партиции.

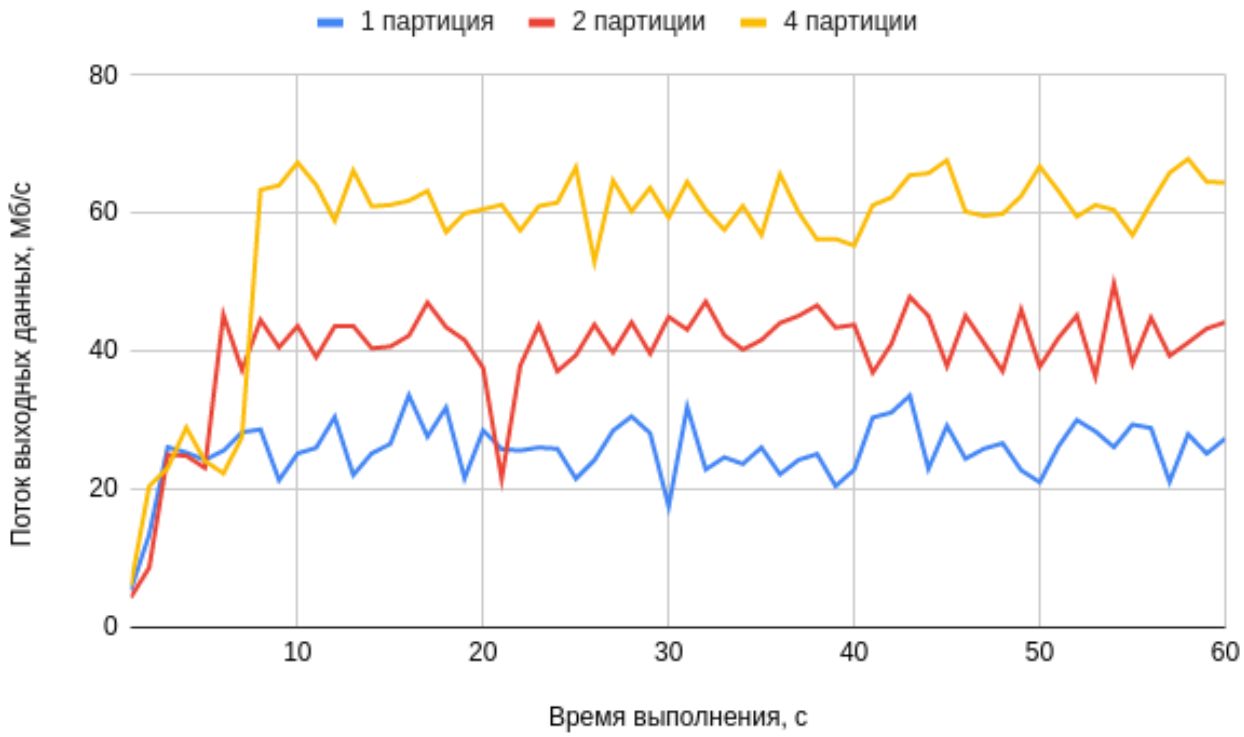

Рисунок 1 – Скорость записи выходных данных

Исходя из этого графика можно сделать вывод, что при росте количества партиций, пропорционально возрастает скорость чтения и записи в таблицы. Стоит заметить, что при увеличении количества партиций с двух до четырех не наблюдается двукратного увеличения скорости записи данных. Это связано с

тем, что в тестируемой инсталляции системы используется только три виртуальные машины для хранения данных в файловой системе CephFS.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

а) Дан обзор модели распределенных вычислений MapReduce. Произведен анализ алгоритмов обработки больших объемов данных, лежащих в основе MapReduce. Были рассмотрены существующие реализации алгоритмы обработки потоков данных, поступающих в реальном времени.

б) Модель распределенных вычислений MapReduce была обобщена и адаптирована для обработки данных, поступающих в реальном времени. На основе полученных знаний была разработана система обработки больших объемов данных, поступающих в реальном времени. Спроектирована модель базы данных.

в) Спроектирована и развернута в облачном сервисе инфраструктура. Серверная часть была реализована с использованием языка программирования Go и библиотеки «`runc/libcontainer`». Клиентский веб-интерфейс был разработан на языке TypeScript с использованием библиотеки React.

Рекомендации по практическому использованию

а) Полученные результаты формируют теоретическую и практическую базу для разработки программных систем обработки больших объемов данных в реальном времени.

б) Разработанная система может использоваться для решения реальных задач обработки больших объемов данных.

в) Предложенная инфраструктура позволяет разворачивать систему как SaaS продукт (т.е. на своих серверах), так и устанавливать её on-premise (на серверах клиентов-организаций).

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1-А. Удовин, И. А. Разработка тестирующей системы с использованием современных технологий изоляции процессов / Удовин И. А., Воронова В. В. // Информационные технологии и системы 2020, Минск, 2020. – С. 203-204.

2-А. Удовин, И. А. Применение алгоритмов для определения плагиата в программном коде / Удовин И. А., Воронова В. В. // Информационные технологии и системы 2021, Минск, 2021. – С. 101-102.