

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра Системного анализа и автоматического управления

МОДЕЛИРОВАНИЕ ОТКАЗОУСТОЙЧИВЫХ СИСТЕМ ПОТОКОВОЙ
ОБРАБОТКИ ДАННЫХ

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 271 группы
направления 09.04.01 — Информатика и вычислительная техника
факультета КНиИТ
Федоренко Михаила Антоновича

Научный руководитель

к. ф.-м. н.

И. Е. Тананко

Заведующий кафедрой

к. ф.-м. н.

И. Е. Тананко

Саратов 2022

ВВЕДЕНИЕ

Актуальность темы. Системы потоковой обработки данных в настоящее время стремительно развиваются, большинство крупных компаний интегрируют их в свои системы, ученые активно исследуют область и публикуют новые исследования, с каждым годом появляются новые инструменты для работы с потоками и все указывает на то, что данная область продолжит развиваться. Поэтому моделирование системы потоковой обработки данных является актуальным направлением на сегодняшний день.

Цель данной работы. Целью данной работы является рассмотрение потоковых систем обработки данных, их составляющих, отказоустойчивости, рассмотрение самых актуальных программных продуктов области потоковой обработки, а также моделирование сети потоковой обработки данных с ненадежными приборами с последующей реализацией модели при помощи языка программирования для подсчета характеристик системы.

В соответствии с поставленной целью определены следующие задачи:

- Изучение систем потоковой обработки информации и их составляющих
- Изучение актуальных решений в области потоковой обработки данных
- Изучение параметров сетей массового обслуживания, их параметров и характеристик
- Создание модели системы потоковой обработки данных
- Написание программы для моделирования и подсчета характеристик заданной пользователем системы потоковой обработки информации
- Проведение экспериментов с целью выявления зависимостей в модели системы потоковой обработки информации

Методологические основы. В текущей работе использовались теории сетей массового обслуживания и теории массового обслуживания

Практическая значимость магистерской работы. Практическая значимость магистерской работы заключается в использовании созданной программы для моделирования системы потоковой обработки данных с целью определения характеристик системы. Рассмотренная модель, программа для моделирования и полученные в результате экспериментов данные и зависимости могут быть использованы в учебном процессе Саратовского Государственного Университета.

Структура и объем работы. Магистерская работа состоит из Введения,

9 разделов и списка использованных источников. Общий объем работы - 51 страница, в том числе 25 рисунков, список источников - 20 наименований.

1 Краткое содержание работы

Первый раздел, "Обзор публикаций по теме исследования" посвящен исследованиям в области систем потоковой обработки и сетей массового обслуживания. В частности, рассмотрена статья о требованиях, которые были предъявлены системам потоковой обработки данных более десяти лет назад, большинство из которых остаются актуальными и по сегодняшний день. Также рассмотрена статья с исследованием тенденций в области систем потоковой обработки данных среди крупнейших компаний, показывающая, что данная область является актуальной в той или иной степени практически среди всех крупнейших компаний, принявших участие в исследовании. Помимо этого, был рассмотрен ряд статей по теме сетей массового обслуживания, именно при помощи теории сетей массового обслуживания была построена математическая модель системы потоковой обработки информации в практической части работы.

Второй раздел, "Системы потоковой обработки знакомит с общим устройством подобных систем, их отличиями от более привычных систем пакетной обработки, а также предоставляет обобщенную схему представления системы потоковой обработки данных. Системы потоковой обработки, как правило, состоят из звена сбора данных, звена очереди сообщений, звена анализа данных и звена доступа к данным (рис.1). Каждое из этих звеньев подробно рассматривается в следующих разделах.

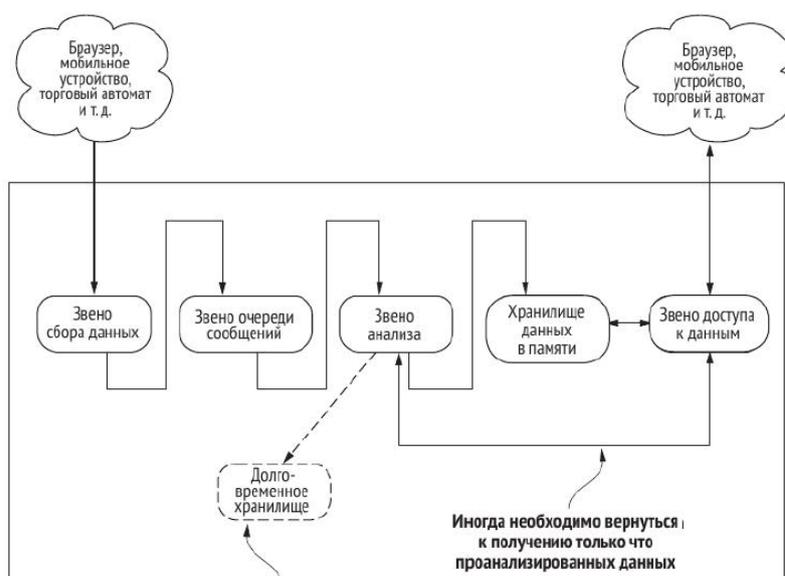


Рисунок 1 – Архитектурная диаграмма системы потоковой обработки

Третий раздел, "Звено сбора данных рассматривает первую составля-

ющую системы потоковой обработки данных - звено сбора данных. Поступающий в систему поток может сильно отличаться в зависимости от типа решаемой задачи, иметь постоянную нагрузку или сильно меняться с течением времени, к примеру, потоки от датчиков электромобиля или высоконагруженного приложения сильно отличаются от потоков датчиков погоды. В данном разделе рассмотрены такие вопросы, как схемы взаимодействия с источником, особенности представления потоковых событий. Также в разделе подробно рассмотрены механизмы отказоустойчивости звена сбора данных в системах потоковой обработки, реализующиеся при помощи протоколирования на стороне отправителя (SBML), протоколирования на стороне получателя (RBML) и гибридного протоколирования (HML). Помимо этого, в разделе рассматривается промежуточное звено между сбором данных и их анализом - звено очереди сообщений. Рассмотрена работа одного из самых популярных решений на данный момент - Apache Kafka, цикл его работы и преимущества использования.

В четвертом разделе, "Звено анализа данных" рассматривается следующий этап работы систем потоковой обработки данных, в котором происходит основная работа - фильтрация, преобразование, агрегация результатов. В обобщенном виде это звено можно описать следующим образом: Поточковый диспетчер несет ответственность за передачу потоковой задачи одному или нескольким потоковым процессорам. В потоковом процессоре выполняется самое интересное - выполняется задача. Источники данных представляют входные данные потоковой задачи.

В силу того, что область потоковой обработки данных постоянно меняется, существует множество позиций насчет того, как должен выглядеть современный потоковый фреймворк. Раздел состоит из шести подразделов, каждый из которых повествует о важном аспекте работы потокового фреймворка, являющегося реализацией звена анализа данных в реальных системах.

В первом подразделе рассматриваются различные модели обработки данных - микро-пакетная и естественная, во втором подразделе обсуждаются модели программирования - императивная и декларативная, история моделей программирования (и языков) в контексте потоков и самые популярные решения на сегодняшний день. Если несколько лет назад системы потоковой обработки данных реализовывались на узкоспециализированных

языках программирования (таких, как Scala), то сейчас весь его интерфейс продублирован на более широко распространённых языках для того, чтобы привлечь более широкий круг разработчиков к разработке систем потоковой обработки.

В третьем подразделе рассматриваются методы обработки сообщений, являющиеся важной характеристикой выбранного потокового фреймворка. Рассмотрены плюсы и минусы всех трех методов ("Строго один раз", "По крайней мере один раз" и "Максимум один раз"), а также альтернативные микро-пакетным методы обработки - журнал транзакций и механизм контрольных точек. Выбранный метод влияет на производительность обуславливается необходимостью согласования данных на разных узлах кластера.

В четвертом подразделе рассматривается такой важный аспект, как сохранение состояния, ведь если потоковый алгоритм анализа становится сложнее, чем операции с одним лишь текущим сообщением без учета зависимостей от других сообщений или внешних данных, то возникает необходимость в запоминании состояния и, вполне вероятно, понадобится служба управления состоянием. В отношении оконных операций с данными следует отметить то, что в зависимости от фреймворка, поддерживается несколько типов окон: окна на фиксированное количество сообщений или на время. Время может быть системным или событийным, то есть отсчитываться на основе времени поступления сообщений в обработку или на основе времени, записанного внутри сообщений.

В пятом подразделе рассматривается аспект распределения данных по узлам кластера и модель управления распараллеливанием. Аспект распределения данных по узлам кластера и модель управления распараллеливанием операций является одним из существенных аспектов выбора фреймворка. Большинство современных продуктов способны выполнять операции распараллеливания автоматически. Фреймворк не предоставляет никаких средств передачи данных на уровне операций. Если требуется обмен данными с другими операторами, это означает, что данные следует выгрузить в очередь Kafka с необходимой функцией распределения данных, и должно быть написано ещё одно приложение, принимающее эти данные и продолжающее их обработку. То есть Kafka становится средством, обеспечивающим гарантированность обработки данных и их доставки по графу операторов, однако

проблемой программиста является создание нужного количества приложений, их распределение по узлам кластера, а также запуск и мониторинг их состояния.

В шестом подразделе рассматривается немаловажный аспект восстановления после сбоев. Распределённая кластерная система состоит из множества взаимодействующих узлов, каждый из которых может выйти из строя. Проверка потокового фреймворка на устойчивость работы в случае отключения отдельных узлов важна потому, что в отличие от пакетной обработки, потоковая задача запускается однократно и долговременно. Потоковая задача не может быть завершена без прекращения обработки запросов вообще. Если потоковый фреймворк не обеспечивает автоматическое восстановление узлов после сбоя, это ведёт к деградации производительности задачи.

В пятом разделе, "Актуальные фреймворки" рассмотрены самые популярные инструменты в быстрорастущем сегменте фреймворков потоковой обработки, такие как Apache Storm, Twitter Heron, Apache Spark Streaming, Apache Flink, Apache Kafka Streams, Apache Samza, Apache Apex и Amazon Kinesis Streams, их ключевые особенности и преимущества.

В шестом разделе, «Звено доступа к данным» рассматриваются возможные варианты действий с полученными данными и работа звена, которое отвечает за доступ к ним. После завершения анализа данных существует четыре варианта действий: проанализировать и отбросить данные, проанализировать данные и переправить их потоковой платформе, проанализировать и сохранить их для использования в режиме реального времени и проанализировать данные и сохранить их для доступа пакетных программ. В случае записи данных в хранилище может понадобиться промежуточное звено очереди сообщений и отлично подойдет ранее рассмотренный продукт Apache Kafka.

В седьмом разделе, «Модель системы потоковой обработки данных» составлена математическая модель системы в виде открытой экспоненциальной сети массового обслуживания с одним классом требований, которая состоит из L звеньев обработки данных, одного звена очереди сообщений, одного звена анализа данных, одного хранилища данных в памяти и одного звена доступа к данным. Из браузеров, мобильных устройств и других внешних источников в систему поступает поток данных, который необходимо обрабо-

тать.

Каждому звену обработки данных i в модели поставлена в соответствие система массового обслуживания S_i типа $M|M|1$ с интенсивностями обслуживания $\mu_i, i = 1, \dots, L$.

Средняя длительность пребывания каждого из звеньев обработки данных в работоспособном состоянии намного больше средней длительности пребывания звена в состоянии восстановления.

Если прибор системы S_i находится в работоспособном состоянии, то требования в этой системе обслуживаются с интенсивностью $\mu_i > 0$. В неработоспособном состоянии обслуживание требований не производится, т. е. $\mu_i = 0$. При этом требования продолжают поступать в систему из источника в очередь и ожидают восстановления прибора системы. Источники требований представлены пуассоновским источником S_0 требований одного класса. Требования из источника поступают в систему $S_i, i = 1, \dots, L$ с вероятностью θ_{0i} .

Звену очереди сообщений и звену анализа данных поставлены в соответствие системы массового обслуживания S_a типа $M|M|1$ с интенсивностью обслуживания μ_a . После завершения обслуживания в системе $S_i, i = 1, \dots, L$ требования с вероятностью θ_{i0} переходят в источник, и с вероятностью $1 - \theta_{i0}$ в систему S_a . Вероятности $\theta_{i0}, i = 1, \dots, L$ показывают доли сообщений, которые не представляют интереса для пользователя.

Звено хранилища данных в памяти и звено доступа к данным отображены в модели системой массового обслуживания S_d типа $M|M|1$ с интенсивностью обслуживания μ_d . После завершения обслуживания в системе S_a требования с вероятностью 1 переходят в систему обслуживания S_d , а после завершения обслуживания в системе S_d требования с вероятностью 1 покидают сеть и возвращаются в источник S_0 .

Таким образом, структурная схема модели системы потоковой обработки данных имеет вид, показанный на рис. 2

После нахождения $\lambda_1, \lambda_2, \lambda_3, \lambda_a, \lambda_d$, проверяется условие существования стационарного режима, для которого необходимо, чтобы выполнялись неравенства $\lambda_1 < \mu_1, \lambda_2 < \mu_2, \lambda_3 < \mu_3, \lambda_a < \mu_a, \lambda_d < \mu_d$, если все неравенства верны, то стационарный режим в сети существует.

Если стационарный режим существует, то определяются для каждой

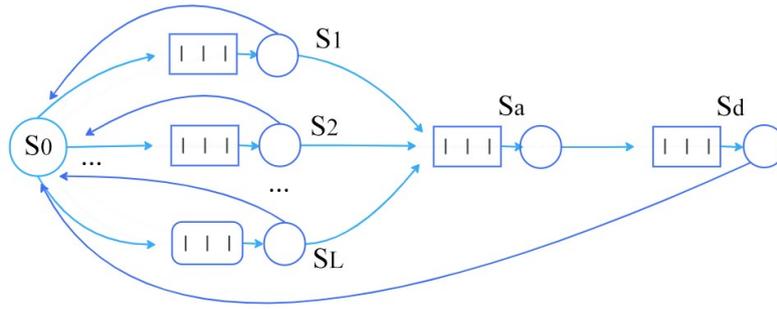


Рисунок 2 – Структурная схема модели системы потоковой обработки данных

СМО следующие характеристики:

$P_i(0) = 1 - \lambda/\mu$ – вероятность того, что в i -й СМО нет требований.

$\bar{q} = \lambda/(\mu - \lambda)$ – м.о. числа требований в системе обслуживания

$\bar{u} = \bar{q}/\lambda$ – м.о. длительности пребывания требований в СМО

$\bar{b} = \lambda^2/(\mu(\mu - \lambda))$ – м.о. числа требований в очереди СМО

$\bar{w} = \bar{b}/\lambda$ – м.о. длительности пребывания требований в очереди СМО

В восьмом разделе, «Описание программы» рассмотрена написанная на языке Java программа для создания модели с произвольным числом звеньев, с выбранными пользователем характеристиками. Программа составляет модель системы потоковой обработки согласно введенным пользователем па проверяет существование стационарного режима в сети и вычисляет ключевые характеристики для каждой СМО (вероятность, что в СМО нет требований, м.о числа требований в системе обслуживания, м.о длительности пребывания требований в СМО, м.о числа требований в очереди в СМО и м.о длительности пребывания требований в очереди в СМО) и сети в целом (м.о длительности пребывания, интенсивность требований, м.о числа требований), а также выводит полученные данные в удобном для пользователя виде.

В девятом разделе, «Результаты экспериментов» рассмотрены полученные в результате экспериментов результаты, а также выведены зависимости (и представлены в виде графиков) между некоторыми параметрами моделируемой системы, в частности, между м. о длительности пребывания требований от интенсивности потока требований, м. о числа требований от интенсивности потока, м.о длительности пребывания требований от вероятности

отказа приборов и м.о длительности пребывания требований от числа звеньев.

ЗАКЛЮЧЕНИЕ

В современных, быстроразвивающихся системах с ежегодно возрастающей нагрузкой и требованиями к скорости обслуживания невозможно недооценить важность систем потоковой обработки данных. Как было показано в данной работе, этот класс систем помогает обеспечить минимальную задержку поступления необходимой информации, а также увеличивает скорость обработки информации из-за того, что данные находятся в постоянном движении, в отличие от классических систем пакетной обработки. Это помогает системе достичь наибольшего быстродействия, что является критически важным фактором для целого ряда областей, как с точки зрения компании - создателя сервиса, так и с точки зрения пользователя. Помимо имеющихся плюсов, в работе были рассмотрены и некоторые минусы систем потоковой обработки, сдерживающие темпы их повсеместного распространения, среди которых главным является сложность разработки, поддержки и интеграции систем потоковой обработки в текущие проекты. Рассмотренные минусы нивелируются по сравнению с преимуществами использования систем потоковой обработки и предполагается, что со временем, с разработкой более удобных высокоуровневых систем, с накоплением опыта в области потоковой обработки, минусы связанные со сложностью разработкой и использованием подобных систем уйдут, а частота использования будет только расти.

В данной работе были рассмотрены системы потоковой обработки данных, их построение, функционирование, отказоустойчивость на всех этапах и самые современные реализации систем и их элементов. Также был проведен анализ материалов и статей по теме исследования, изучена актуальность области на данный момент. В заключительной части работы была построена математическая модель системы потоковой обработки данных с ненадежными элементами, для которой разработана программа на языке Java. Программа позволяет смоделировать работу системы с производным количеством звеньев и различными параметрами, которые выбирает пользователь, а также вычисляет (при условии работоспособности в условиях выбранных параметров) ключевые характеристики смоделированной системы. На основе полученных результатов были выведены зависимости между некоторыми характеристиками систем.

Также, в данной работе были рассмотрены последние и наиболее популярные технологии в системах потоковой обработки данных. Какие-то продукты позиционируются как универсальные для решения различных задач, какие-то - являются специализированными средствами решения определенного класса задач, и, судя по темпам появления новых продуктов, актуальности и наличию интереса крупных компаний к данной области, нет сомнений, что системы потоковой обработки будут продолжать развиваться стремительными темпами

Результаты магистерской работы были представлены на ежегодной конференции СУИТиММ в 2022 году. Все цели, поставленные в ходе магистерской работы, были выполнены.