Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Фирсов Михаил Александрович

Реализация статистик к Desbordante

Учебная практика

Научный руководитель:
ассистент кафедры ИАС Чернышев Г. А.

Санкт-Петербург
2023

Оглавление

Введение

Профилирование данных – процесс, заключающийся в нахождении закономерностей, исследовании характера распределения, оценке качества данных [1, 8], это поиск метаданных в данных.

Что такое метаданные?

- В первом приближении это данные о данных: размер файла, время создания, авторство.
- В более общем смысле, это различные закономерности, сокрытые в данных.

В этой работе будет рассматриваться только профилирование табличных данных. Профилирование делиться на две группы: простое и наукоёмкое. Для табличных данных к первой группе можно отнести подсчет простых статистик: число строк, столбцов в таблице, минимальное и максимальное значения в каждой колонке и ряд других. Ко второй можно отнести поиск функциональных зависимостей, который не будет здесь рассмотрен.

Desbordante — высокопроизводительный профилировщик данных, нацеленный на извлечение наукоемких паттернов. Система предоставляет как консольный, так и веб интерфейс. Изначально Desbordante создавался как профилировщик наукоёмких данных. Из-за этого он сильно проигрывал аналогам, которые предоставляли гораздо больше информации о данных, за счёт простого профилирования.

1. Постановка задачи

На момент начала учебной практики в проекте были написаны простые статистики в виде алгоритмов на C++, к которым были претензии по производительности и коду, которые не были исправлены.

Целью работы являлась доработка простых статистик в алгоритмической части Desbordante и их имплементация в серверной части web приложения. Для достижения данной цели были поставлены следующие задачи:

1. Доработать простые статистики.
2. Добавить распараллеливание их вычислений, как это сделано для некоторых других алгоритмов Desbordante.
3. Написать для простых статистик тесты для проверки производительности.
4. Сделать обзор возможностей аналогов и дополнить список статистик более сложными, которые планируется реализовывать в дальнейшем.
5. Реализовать запросы подсчета статистик на back-end части web приложения.

2. Обзор особенностей простого профилирования в других системах

Был проанализирован ряд востребованных на рынке профайлеров данных. В этом разделе будет приведен список поддерживаемых ими простых статистик.

2.1. Data Profiler

Data Profiler [4] — python библиотека для удобного анализа данных разных форматов. Поддерживает следующие колоночные статистики [5]:

1. Для колонок целочисленных типов:

    - data_type — тип данных колонки
    - column_name — имя колонки
    - categorical — принимает значение 'true', если колонка категориальная
    - order — принимает значения 'ascending', 'descending', 'random'
    - samples — небольшой срез значений в колонке
    - min — минимум
    - max — максимум
    - sum — сумма

- mean — среднее
- median — медиана
- quantiles — перцентили от 0 до 100
- variance — дисперсия
- stddev — стандартное отклонение (корень из дисперсии)
- skewness — коэффициент асимметрии (третий центральный момент)
- kurtosis — коэффициент эксцесса (четвёртый центральный момент)
- median_absolute_deviation — среднее абсолютное отклонение
- num_zeros — число нулей
- num_negatives — число отрицательных значений
- null_count — число значений NULL
- unique_count — число уникальных значений
- sample_size — общее число значений
- bias_correction — применяет коррекцию смещения к дисперсии, коэффициенту асимметрии и коэффициенту эксцесса
- histogram — информация, относящаяся к гистограммам
  (a) bin_counts — число значений в каждом интервале

(b) bin_edges — порог каждого интервала

- histogram_and_quantiles — функция построения интервалов гистограммы от метода их построения. Если метод явно не указан, то среди встроенных выбирается оптимальный. Поддерживаются следующие: ‘fd’, ‘doane’, ‘scott’, ‘rice’, ‘sturges’, ‘sqrt’
- categorical_count — число вхождений для каждого уникального значения, если колонка категориальная
- unique_ratio — отношение числа уникальный значений к общему числу значений
- categories — список всех уникальных значений, если колонка категориальная

2. Колонки типа float поддерживают помимо выше перечисленных операций, следующие:

   - precision — число значащих цифр
   - sample_ratio — процент строк, отличающихся от заданного значения на некоторую погрешность

3. Строковый колонки поддерживают все статистики целочисленного типа кроме num_zeros и num_negatives. Есть дополнительные параметры is_case_sensitive — отвечает за чувствительность к регистру;
   stop_words — список игнорируемых слов. Для колонок строкового типа также доступны следующие статистики:

- vocab — символы во всех строках
- words — все слова
- top_k_chars — k самых часто встречающихся символа
- top_k_words — k самых часто встречающихся слова

4. Статистики всей таблицы:

   - row_count — число строк
   - row_has_null_ratio — число колонок, в которых есть значение NULL
   - row_is_null_ratio — число колонок, в которых есть только значение NULL
   - unique_row_ratio — число колонок с уникальными значениями во всех строках
   - duplicate_row_count — число колонок, которые встречаются дважды
   - file_type — тип файла (например '.csv')
   - encoding — кодировка (например 'UTF−8')
   - correlation_matrix — матрица корреляции всех колонок между собой
   - chi2_matrix — матрица Хи-квадрат статистик между всеми колонками таблицы
   - profile_schema — имена всех колонок и их индексы

2.2. Data Cleaner

Data Cleaner [2] — мощный движок для профилирования данных и смены формата их хранения. В git репозитории [3] удалось найти следующие виды колоночных статистик:

1. Для колонок целочисленных типов:

    - ROW_COUNT — число строк
    - NULL_COUNT — число значений типа NULL
    - HIGHEST_VALUE — максимум
    - LOWEST_VALUE — минимум
    - SUM — сумма
    - MEAN — среднее
    - GEOMETRIC_MEAN — среднее геометрическое
    - STANDARD_DEVIATION — стандартное отклонение (корень из дисперсии)
    - VARIANCE — дисперсия (второй центральный момент)
    - SUM_OF_SQUARES — сумма квадратов
    - SECOND_MOMENT — второй момент
    - MEDIAN [9] — медиана (процентиль 50)
    - PERCENTILE25 — процентиль 25
    - PERCENTILE75 — процентиль 75

- KURTOSIS — коэффициент эксцесса (четвёртый центральный момент)
- SKEWNESS — коэффициент асимметрии (третий центральный момент)

2. Статистики колонок строкового типа:

   - MIN_WORDS — количество минимальных слов
   - MAX_WORDS — количество максимальных слов
   - WORD_COUNT — число слов
   - NON_LETTER_CHARS — число не буквенных символов
   - DIACRITIC_CHARS — число диакритических знаков
   - DIGIT_CHARS — число цифр
   - LOWERCASE_CHARS — число символов в нижнем регистре
   - UPPERCASE_CHARS_EXCL_FIRST_LETTERS — число символов в верхнем регистре, кроме первых букв в слове
   - UPPERCASE_CHARS — число символов в верхнем регистре
   - AVG_WHITE_SPACES — среднее число пробелов
   - MIN_WHITE_SPACES — минимальное число пробелов

- MAX_WHITE_SPACES — максимальное число пробелов
- AVG_CHARS — среднее число символов
- MIN_CHARS — минимальное число символов
- MAX_CHARS — максимально число символов
- TOTAL_CHAR_COUNT — общее число символов во всех строках
- ENTIRELY_LOWERCASE_COUNT — число слов в нижнем регистре
- ENTIRELY_UPPERCASE_COUNT — число слов в верхнем регистре

3. Статистики колонок типа DATETIME:

- HIGHEST_TIME — максимальное время
- LOWEST_TIME — минимальное время
- HIGHEST_DATE — максимум
- LOWEST_DATE — минимум
- NULL_COUNT — число значений NULL
- ROW_COUNT — число строк
- MEAN — среднее
- MEDIAN — медиана
- PERCENTILE25 — процентиль 25

- PERCENTILE75 — процентиль 75
- KURTOSIS — коэффициент эксцесса (четвёртый центральный момент)
- SKEWNESS — коэффициент асимметрии (третий центральный момент)

4. Статистики колонок типа BOOL:
   - FALSE_COUNT — число значений 'false'
   - TRUE_COUNT — число значений 'true'
   - NULL_COUNT — число значений NULL
   - ROW_COUNT — число строк

2.3. Great Expectations

Great Expectations [6] — это инструмент для проверки, документирования и профилирования данных, который позволяет автоматизировать эти процессы. Принцип его работы немного отличается от обычного поиска набора статистик. Основным понятием здесь является ожидание верности какого-либо утверждения о данных. Примеры ожиданий [7]:

1. expect_column_values_to_not_be_null — ожидание того, что в колонке нет значений NULL
2. expect_column_values_to_be_unique — ожидание того, что в колонке все значения уникальны

3. expect_table_row_count_to_be_between — ожидание того, что в колонке все значения из заданного диапазона

4. expect_column_median_to_be_between — ожидание того, что среднее колонки из заданного диапазона

3. Реализация

Desbordante состоит из двух частей: алгоритмической[1] и web[2]. Web в свою очередь также состоит из двух частей: back-end и front-end.

В начале учебной практики была реализация только алгоритмической части статистик. В ходе настоящей работы она была доработана, а также была реализована поддержка статистик на back-end в web-приложении. Разработкой front-end-а занимался другой член команды Desbordante.

3.1. Общие изменения в алгоритмической части

Алгоритмическая часть состояла из одного класса CSVStats, который включал в себя методы для подсчета простых статистик датасета, расположение которого являлось одним из параметров конструктора этого класса. Список всех изменений в методах этого класса:

1. Теперь класс CSVStats принимает один параметр в конструкторе — экземпляр специального класса-конфига.

2. Ранее статистики возвращались в виде типа std::optional<std::pair<std::byte const*, mo::Type const*>>. Были выделены отдельные классы одной статистики, а так-

[1]https://github.com/Mstrutov/Desbordante
[2]https://github.com/vs9h/Desbordante

же статистик колонки. Это позволило работать с результатом подсчета статистик в объектно-ориентированном стиле, сократило объём кода.

3. Добавлен метод подсчета всех статистик сразу, с возможностью распараллеливания по отдельным колонкам таблицы.

4. Много изменений в стиле кода: ссылки и указатели переделаны в east-const style, имена переменных — в snake_case, удалены лишние include. Часть методов переименована. Написаны краткие комментарии ко всем методам подсчета статистик.

5. Найдена и исправлена ошибка в методе distinct: сравнение велось по первому байту данных, а не в соответствии с типом колонки. Для смешанных типов сделан отдельный метод для подсчета этой статистики.

6. Где возможно, удалены дублирования кода.

7. В класс Type добавлен метод CloneType для клонирования типа, а не данных этого типа. В дальнейшем необходимо сделать полноценную систему приведения типов. На данном этапе сделано только приведение int и double к double. Найдены и исправлены ошибки в клонировании данных смешанного типа и типа string. Нужно добавить тип bool. На данный момент, время чтения всего датасета с одновременным определением типа колонок занимает в некоторых слу-

чаях (датасеты EpicMeds.csv и EpicVitals.csv) больше времени, чем подсчет в несколько потоков всех статистик. Эта часть кода написана другим членом команды Desbordate.

Сейчас Desbordante поддерживает следующие простые статистики:

- NumberOfValues — число непустых и не NULL значений в колонке.
- GetNumberOfColumns — число колонок в таблице.
- Distinct — число уникальных значений в колонке.
- IsCategorical — проверяет, является ли колонка категориальной. Вычисляется по формуле: $Distinct \leq min($ $NumberOfValues - 1, 10 + NumberOfValues/1000)$.
- ShowSample — возвращает тип std::vector<std::vector<std::string≫ — двумерный массив — срез таблицы от некоторой начальной до некоторой конечной колонок и строк. Эта статистика ещё не реализована в web-версии.
- GetAvg — среднее значение в колонке, если она числовая.
- GetCorrectedSTD — скорректированное стандартное отклонение колонки, если она числовая.
- GetSkewness — коэффициент асимметрии (третий центральный момент) колонки, если она числовая.

- GetKurtosis — коэффициент эксцесса (четвёртый центральный момент) колонки, если она числовая.
- GetCentralMomentOfDist — центральный момент колонки, если она числовая.
- GetStandardizedCentralMomentOfDist — нормированный момент колонки, если она числовая.
- GetMin — минимум в колонке.
- GetMax — максимум в колонке.
- GetSum — сумма в колонке, если она числовая.
- GetQuantile — возвращает нужный процентиль (25, 50, 75 для веб-части).
- DeleteNullAndEmpties — удаляет пустые и NULL значения из колонки, возвращает полученную сокращенную колонку в виде вектора.

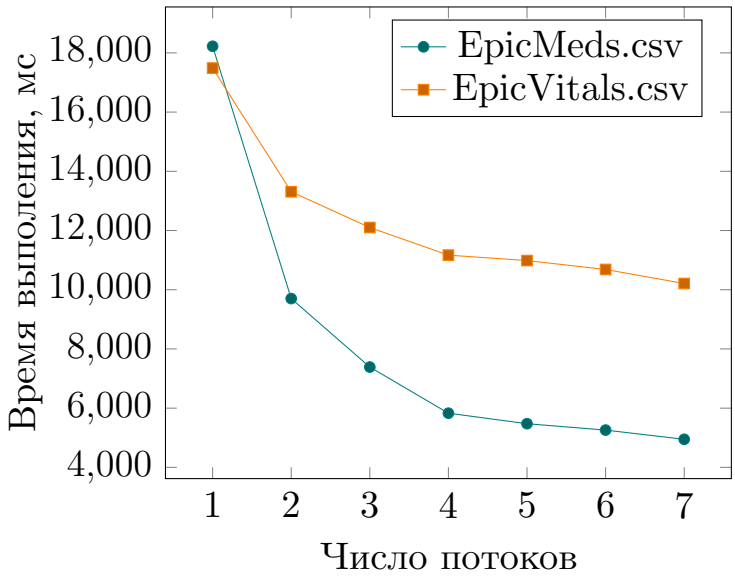

Рис. 1: Изменение времени подсчета статистик при разном числе потоков

3.2. Тестирование

На начало учебной практики к статистикам уже были написаны тесты, в них велось обращение к одному специально написанному датасету небольшого размера. Каждый тест проверял одну статистику на нескольких (не всех) колонках этой таблицы.

3.2.1. Тесты на производительность

Для тестирования производительности были выбраны датасеты EpicMeds.csv (1282 тысячи строк, 10 столбцов) и EpicVitals.csv (1246 тысяч строк, 7 столбцов), так как все их колонки числовые, а значит поддерживают одни и те же статистики. В итоговое время входит время подсчета всех статистик для датасета, время

чтения данных с файла и время их преобразования в специальный тип, который хранит уже типизированные колонки. Последние две части написаны другим разработчиком. Результат можно увидеть на Рис. 1.

3.2.2. Тесты на корректность

В уже написанных тестах были произведены те же изменения в стиле, что и в алгоритмической части. Также был произведён ряд небольших изменений в соответствии с новыми структурами данных для хранения статистик. Добавлены два теста на подсчет всех статистик вместе для пустого датасета и основного тестового датасета статистик, в который была добавлена колонка со смешанным типом.

3.3. Поддержка статистик на back-end уровне

На Рис.2 схематично изображена архитектура web части Desbordante. После завершения доработки алгоритмической части статистик требовалось добавить их поддержку в web. Эта задача была разделена между двумя людьми. Мной была добавлена возможность получать с front-end-а информацию об уже посчитанных статистиках и, в случае её отсутствия запускать подсчет всех статистик для данного датасета в заданном числе потоков с поддержкой запросов на получение прогресса этого процесса.

1. Был сделан rebase git-репозитория веб Desbordante в соот-

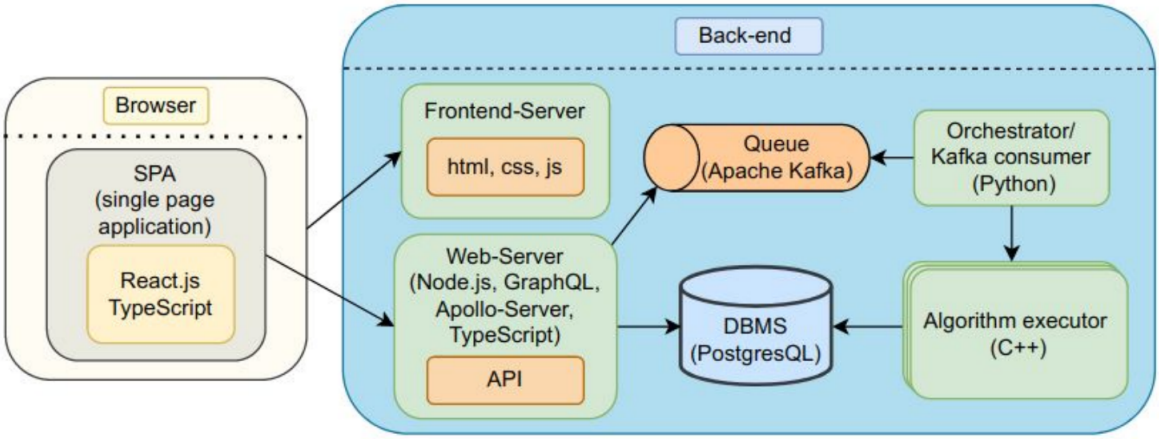

Рис. 2: Устройство веб-версии Desbordante

ветствии с последними изменениями в алгоритмической части.

2. В схему базы данных добавлена новая таблица: ColumnStats для хранения статистик колонок датасета. Она содержит следующие поля:

   - type — тип колонки
   - fileID — идентификатор датасета
   - columnIndex — индекс колонки
   - И следующие статистики: distinct, isCategorical, count (NumberOfValues), avg, STD, skewness, kurtosis, min, max, sum, quantile25, quantile50, quantile75. Для некоторых типов ряд из этих статистик устанавливается в NULL, если их нельзя посчитать для данного типа колонки.

3. Дописан cpp-consumer (algorithm executor на Рис. 2). Он запускает вычисление всех статистик датасета в нужном числе потоков и после завершения сохраняет результат в БД в таблицу ColumnStats, описанную выше.

4. Дописана GraphQL схема БД для запуска подсчета статик данного датасета и получения подсчитанных статистик для front-end-а.

4. Итоговая таблица статистик

Результатом приведённого анализа аналогов и уже реализованных статистик можно считать Таб. 1. Её можно использовать для дальнейшей разработки этого направления, например, в качестве списка статистик на реализацию.

Таблица 1: Поддерживаемые статистики

Тип данных	Статистика	Data Profiler	Data Cleaner	Desbordante
Строковые	vocab	+	−	−
	words	+	−	−
	topKChars	+	−	−
	topKWords	+	−	−
	minWords	−	+	−
	maxKWords	−	+	−
	wordCount	−	+	−
	nonLetterChars	−	+	−
	diacriticChars	−	+	−
	digitChars	−	+	−
	lowercaseChars	−	+	−
	uppercaseChars	−	+	−
	uppercaseChars−ExclFirstLetters	−	+	−
	avgWhiteSpaces	−	+	−
	minWhiteSpaces	−	+	−

	maxWhiteSpaces	−	+	−
	avgChars	−	+	−
	minChars	−	+	−
	maxChars	−	+	−
	totalCharCount	−	+	−
	entirelyLowercaseCount	−	+	−
	entirelyUppercaseCount	−	+	−
Общие	dataType	+	−	+
	columnName	+	−	−
	categorical	+	−	+
	samples	+	−	+
	min	+	+	+
	max	+	+	+
	quantiles/median	+	+	+
	nullCount	+	+	−
	uniqueCount	+	−	+
	sampleSize	+	+	+
	categoricalCount	+	−	−
	uniqueRatio	+	−	−
	categories	+	−	−
Float	precision	+	−	−
	sampleRatio	+	−	−
DateTime	highestTime	−	+	−
	lowestTime	−	+	−
	sum	+	+	+

Числовые

	mean	+	+	+
	geometricMean	−	+	−
	variance	+	+	+
	stdDev	+	+	+
	centralMoment	−	−	+
	standardizedCentralMoment	−	−	+
	skewness	+	+	+
	kurtosis	+	+	+
	medianAbsoluteDeviation	+	−	−
	numZeros	+	−	−
	numNegatives	+	−	−
	biasCorrection	+	−	−
	histogram	+	−	−
	histogramAndQuantiles	+	−	−
	sumOfSquares	−	+	−
Bool	trueCount	−	+	−
	falseCount	−	+	−
Вся талица	columnCount	+	−	+
	rowHasNullRatio	+	−	−
	rowIsNullRatio	+	−	−
	uniqueRowRatio	+	−	−
	duplicateRowCount	+	−	−
	fileType	+	−	−
	encoding	+	−	−
	correctionMatrix	+	−	−

	chi2Matrix	+	−	−
	profileSchema	+	−	−

Заключение

В процессе работы были достигнуты следующие результаты:

- Проанализированы документации аналогов, выписаны всевозможные статистики, которые будут реализованы в будущем.
- Устранены все ошибки, неэффективности в реализации простых статистик. Добавлен параллелизм по колонкам таблицы.
- Дописаны тесты, проверяющие корректность работы алгоритмов.
- Написаны нагрузочные тесты для разного числа потоков.
- Реализована поддержка статистик для back-end в web части Desbordante.

Список литературы

[1] Abedjan Ziawasch, Golab Lukasz, and Naumann Felix. Data Profiling: A Tutorial // Proceedings of the 2017 ACM International Conference on Management of Data. — New York, NY, USA : Association for Computing Machinery. — 2017. — SIGMOD '17. — P. 1747–1751. — Access mode: https://doi.org/10.1145/3035918.3054772.

[2] Data Cleaner. — 2022. — Online; accessed 29 November 2022. Access mode: https://datacleaner.github.io/.

[3] Data Cleaner GitHub repository. — 2022. — Online; accessed 29 November 2022. Access mode: https://github.com/datacleaner/DataCleaner.git.

[4] Data Profiler. — 2022. — Online; accessed 29 November 2022. Access mode: https://capitalone.github.io/DataProfiler/docs/0.7.7/html/index.html.

[5] Data Profiler Readme File. — 2022. — Online; accessed 29 November 2022. Access mode: https://github.com/great-expectations/great_expectations/blob/develop/contrib/capitalone_dataprofiler_expectations/README.md.

[6] Great Expectations. — 2022. — Online; accessed 29 November

2022. Access mode: https://greatexpectations.io/expectations/.

[7] Great Expectations Examples. — 2022. — Online; accessed 29 November 2022. Access mode: https://docs.greatexpectations.io/docs/terms/expectation/.

[8] Ilyas Ihab F. and Chu Xu. Data Cleaning. — New York, NY, USA : Association for Computing Machinery, 2019. — ISBN: 9781450371520.

[9] Шоргин С. Я. КВАНТИЛЬ // Большая российская энциклопедия. Электронная версия (2016). — 2016. — Online; accessed 29 November 2022. Access mode: https://bigenc.ru/mathematics/text/2055717.