Санкт-Петербургский Государственный Университет
Математико-механический факультет
Кафедра системного программирования

Шабанов Владимир Сергеевич

Визуализация данных тестирования storage-систем

Курсовая работа

Научный руководитель:
ст. преп. Немешев М.Х.

Санкт-Петербург
2018

Оглавление

1.Введение

Чтобы унифицировать процесс анализа данных автоматического тестирования был спроектирован и реализован фреймворк “PerfStat”, который позволяет генерировать различные отчеты, в человекочитаемом виде, из файлов, содержащих лог результатов тестирования. Использование “PerfStat” позволяет разработчикам внедрить процесс получения отчетов в автоматизированный процесс тестирования, упростить коммуникацию между разработчиками и другими сотрудниками компании.

2.Цели и задачи

Мы с коллегами разработали интуитивный фреймворк “PerfStat”. Я отвечал за проектирование расширяемой архитектуры для взаимосвязи внутреннего представления данных и сторонних библиотек для визуализации, и реализации отчетов в формате html Чтобы достичь этих целей я выделил несколько подзадач:

- Изучение предметной области:
  - изучение процесса автоматизированного тестирования
  - изучение библиотеки ChartJS
- Спроектировать и реализовать масштабируемое преобразование внутреннего представления данных в форматы различных отчетов[3]
- Реализовать конструирование html отчетов с различными элементами:
  - график (ChartJS)
  - таблица (html)
  - текст (html)

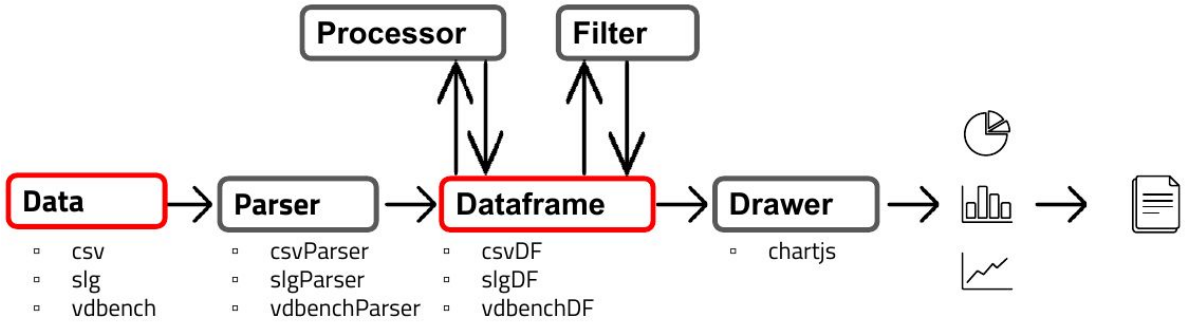

Рис. 1 Общая архитектура фреймворка “PerfStat”

На рисунке 1 представлен общий поток данных в фреймворке “PerfStat”. Сначала данные обрабатываются, и из входных форматов (csv, slg, vdbench) мы получаем внутренней представление (csvDF, slgDF, vdbenchDF). При необходимости, к этим данным применяются более сложные математические функции обработки. После этого получившиеся на прошлом этапе данные преобразуются в формат, заданный начальными настройками, для отображения конечному пользователю

3.Обзор предметной области

В последние десятилетия особенно востребованы облачные хранилища цифровой информации. Эта технология обладает нескольмими преимуществами, по сравнение с локальным хранилищем данных, таких как: быстрота развертывания, дешевизна, облегченная конфигурация. Dell-EMC, в числе прочего, занимается оценкой качества таких storage-систем. Сначала проходит тестирование системы с разными нагрузками, потом специалисты по обработке данных анализируют результаты тестирования.

Сейчас каждый специалист сам выбирает технологию для анализа таких данных, один из самых популярных методов: скрипты, заточенные под нужды конкретных разработчиков. Это создает трудности при обмене информацией между сотрудниками.

Специфическая семантика данных и корпоративная тайна не позволяет адаптировать под эти нужды крупные продукты для анализа данных, такие как QlikView или Tableau.

4.Реализация

4.1 Разработка процесса преобразования данных

Для того, чтобы поддерживать проект в будущем, в том числе, расширять его для новых форматов, я решил реализовать корневой класс, который будет задавать интерфейс для будущих преобразований данных из внутреннего представления в новые типы[1].

4.2 Реализация преобразования внутреннего представления данных в отчеты формата html

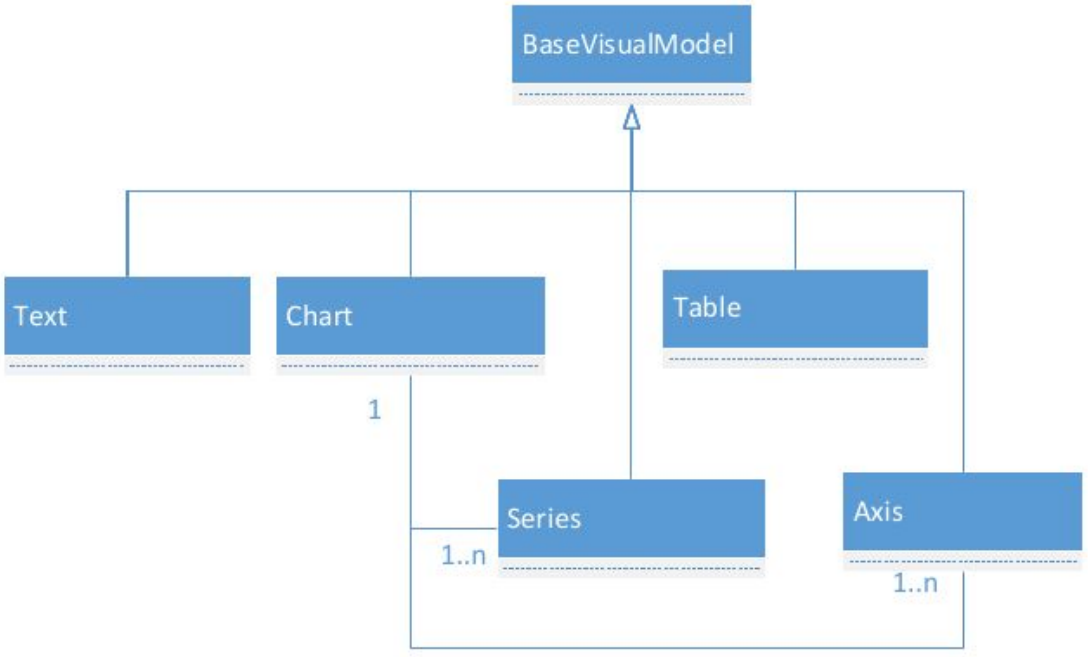

Рис. 2 Диаграмма классов данных, подготовленных к выгрузке в определенный формат

К архитектуре, разработанной на прошлой итерации, для сохранения консистентности данных, пришлось добавить несколько классов для задания графика, как самого сложного способа представления отчетов (по сравнению с просто текстом и таблицей). [5] Эти классы, задают отдельные составляющие графиков, такие как: вертикальные и горизонтальные оси, способ отображения информации на графике (линейный график,

столбцовая диаграмма) и характеристики сетки и масштаба графика. Соответственно, в реализацию класса Chart была добавлена функция для агрегации всех этих классов. Классы для табличного и текстового формата устроены проще [2]. Данные, проходящие через эти классы претерпевают лишь незначительные изменения, перед выгрузкой в html.

5.Заключение

На текущем этапе разработки достигнуты ниже описанные цели:

- Спроектировано и реализовано масштабируемое преобразование внутреннего представления данных в форматы различных отчетов
- Реализовано конструирование html отчетов с различными элементами:
  - график (ChartJS)
  - таблица (html)
  - текст (html)

Список литературы

[1] Software Requirements (3rd Edition) (Developer Best Practices) - https://www.amazon.com/Software-Requirements-Developer-Best-Practices/dp/0735679665/

[2] Operating System Concepts - https://www.amazon.com/Operating-System-Concepts-Abraham-Silberschatz/dp/1118063333/

[3] Information Storage and Management: Storing, Managing, and Protecting Digital Information in Classic, Virtualized, and Cloud Environments (2nd Edition) - EMC - 2012

[5]CharJS documentation - http://www.chartjs.org/docs