Реализация библиотеки для потоковой записи файлов формата XLSX

Свитков Сергей
группа 344
научный руководитель Ю.В. Литвинов
консультант М.В. Заведеев

СПбГУ
кафедра системного программирования

20 мая 2018 г.

Введение

- Веб-приложения в сфере биллинга, телекоммуникаций
- Различные отчеты, статистика
- Формат XLSX
- Потоковое формирование файла для экономии используемой RAM

Постановка задачи

Цель: разработка библиотеки для потоковой записи файлов формата XLSX и сравнение её с существующими реализациями. Для достижения этой цели были поставлены следующие задачи:

- Сформулировать алгоритм, который будет использоваться для генерации файлов формата XLSX в потоковм режиме
- Реализовать предложенный алгоритм в виде переиспользуемой библиотеки с открытым исходным кодом, документацией, примерами использования и артефактом в Maven
- Провести апробацию полученной реализации
- Сравнить полученную реализацию с существующими библиотеками по потреблению RAM (оперативной памяти) и скорости работы при создании документа

Обзор существующих решений

- Apache POI
  - До версии 3.8 — только in-memory
  - Начиная с 3.8 — Stream-API для записи, Event-API для чтения
  - Часть операций всё равно только in-memory
  - Отсутствие полных и подробных примеров работы с Stream-API
  - Отсутствие поддержки автоматического разбиения документа на страницы
- SJXLSX
  - Не поддерживается с 2015 года
  - Отсутствие документации
  - Не поддерживается потоковая запись файлов

Итоги обзора

- Apache POI поддерживает потоковую генерацию файлов, но всё же имеет ряд недостатков
- Реализовать свою библиотеку
- Сравнить её с рассмотренными реализациями

XLSX

- zip-архив с xml файлами
- Структура:
  - `Content_Types.xml` — типы контента в архиве и пути к ним
  - `_rels` — зависимости между файлами внутри архива
  - docProps — метаданные: имя автора, дата создания, ...
  - xl — директория с основными файлами архива: workbook, страницы, стили, таблицы

XLSX

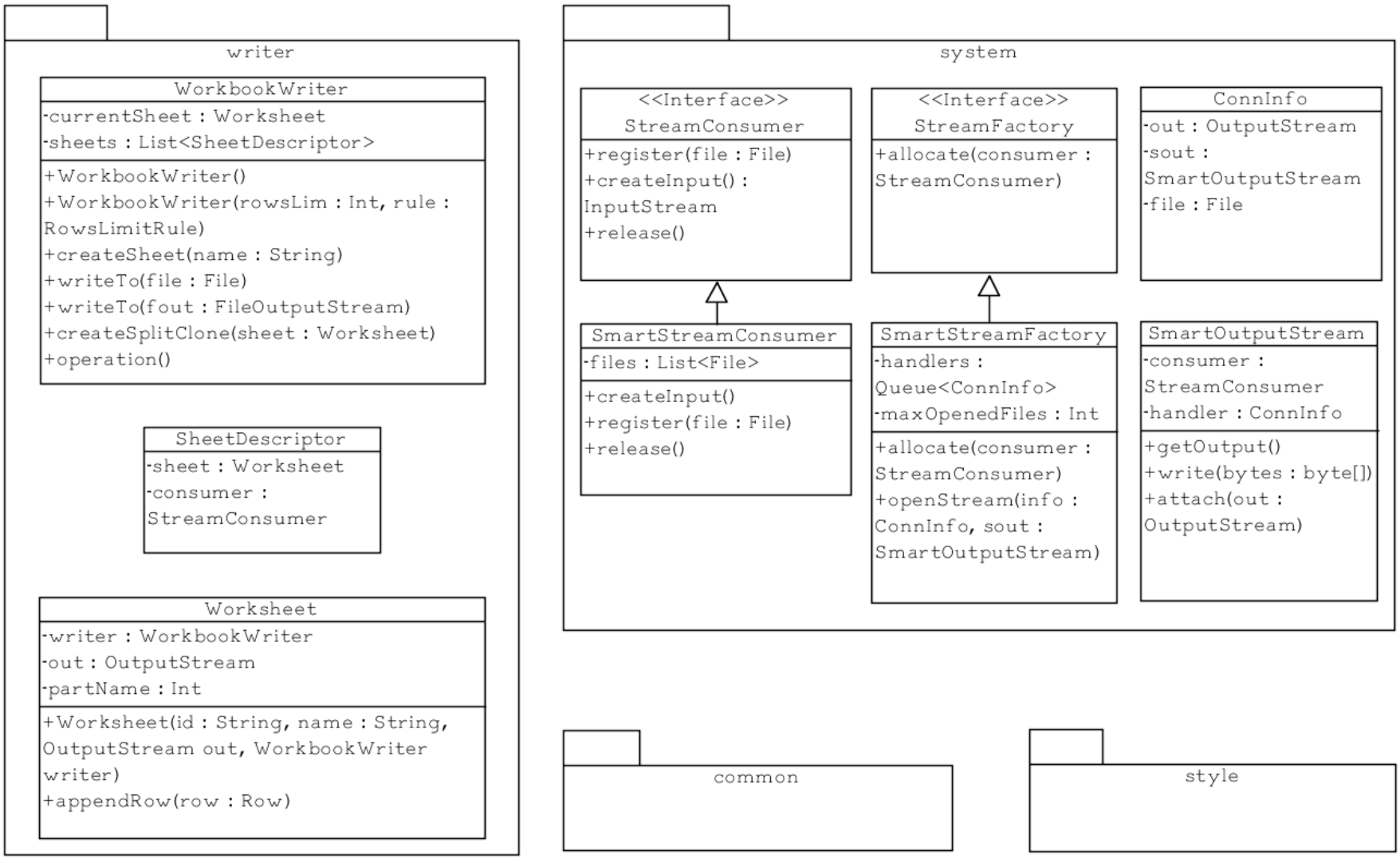
‘

- Workbook:
  - Метаданные
  - Ссылки на страницы с данными
- Worksheet:
  - Содержат данные
  - 3 формата представления данных: Grid, Chart, Dialog
- Grid:
  - Данные разбиты на ряды;
  - Каждый ряд состоит из ячеек, в которых хранятся значения
  - В каждой ячейке хранятся номер ряда и тип значения

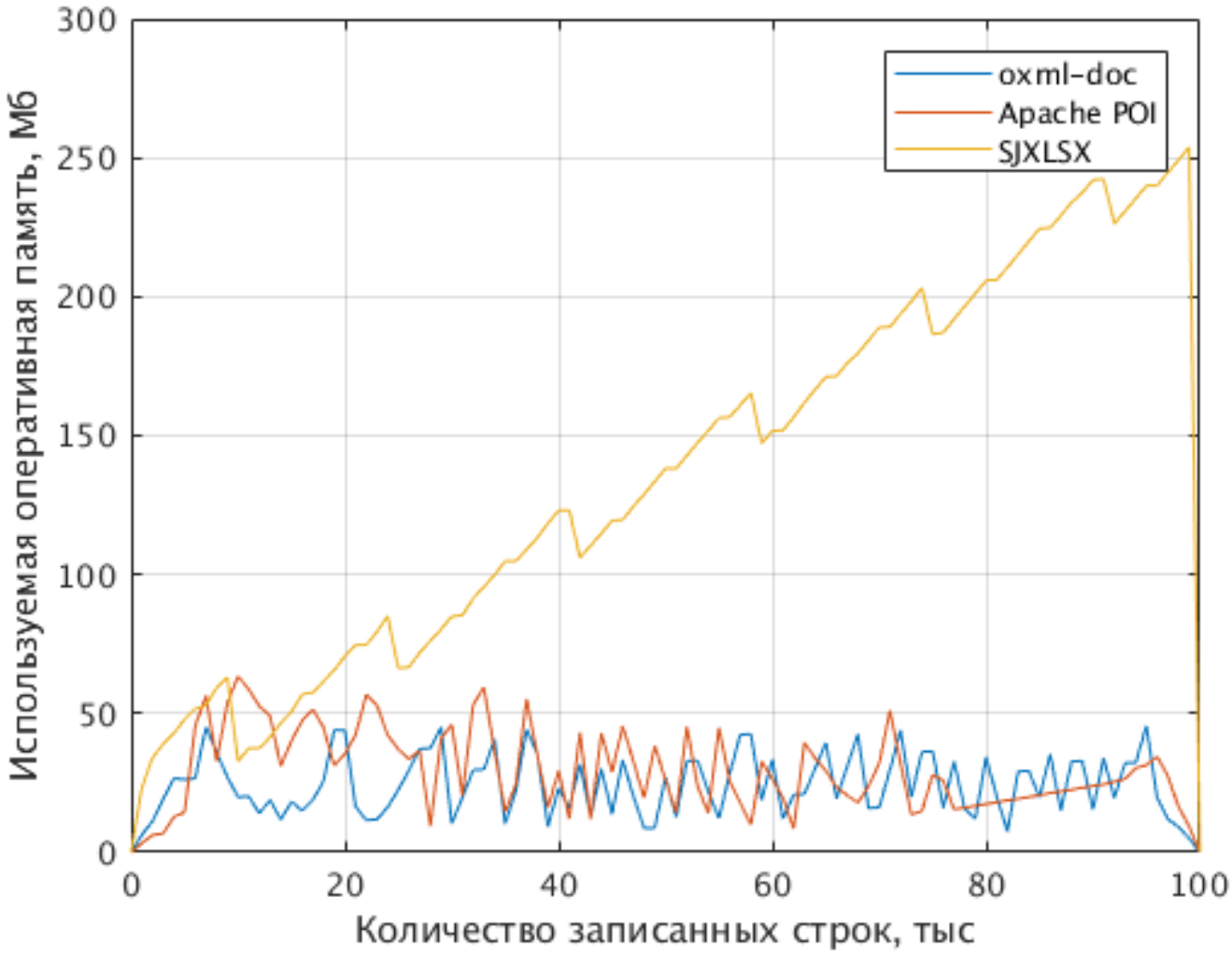
Реализация: алгоритм

- Для каждой страницы создавать временный файл
- Хранить в RAM только один ряд (во время создания)
- После создания добавлять ряд во временный файл страницы
- После завершения формирования документа — записывать данные из временных файлов в основной файл
- Для экономии дискового пространства временные файлы

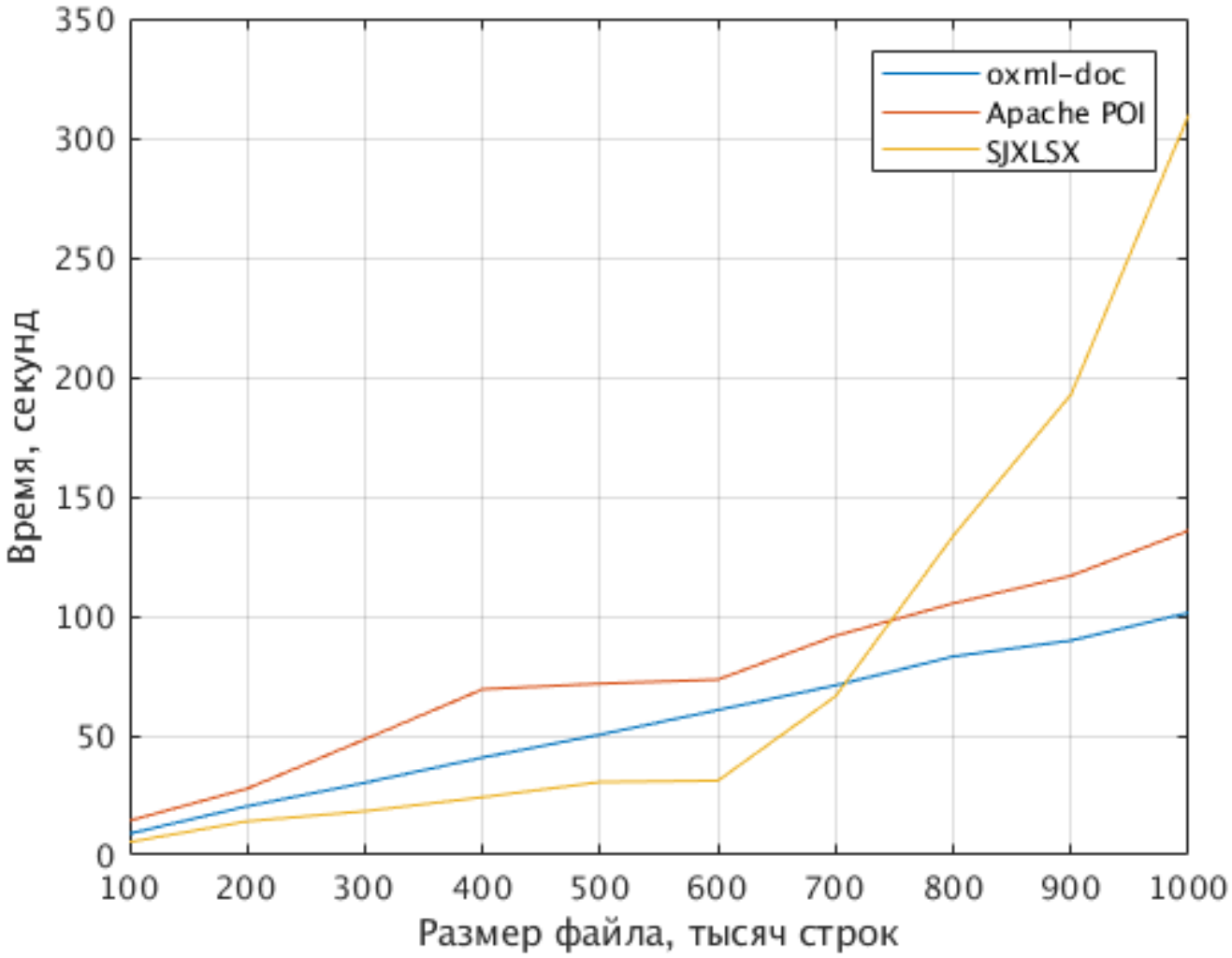
Реализация: архитектура



Эксперименты: количество используемой RAM



Эксперименты: скорость записи



Результаты

В ходе данной работы были достигнуты следующие результаты:

- Реализован предложенный в рамках работы алгоритм записи файлов формата XLSX
- Реализация оформлена в виде библиотеки для потоковой записи файлов формата XLSX с открытым исходным кодом и артефактов в Maven
- Проведено тестирование реализации
- Проведено сравнение реализации с существующими библиотеками по количеству используемой RAM и скорости работы при создании документа