

# ПАРАЛЛЕЛИЗАЦИЯ СБОРКИ WEB-ПРИЛОЖЕНИЯ ДЛЯ ЕЕ ОПТИМИЗАЦИИ

Рассматривается метод оптимизации сборки web-приложения. Предлагается использование параллелизации для ускорения процессов, выполняющихся во время сборки.

## ВВЕДЕНИЕ

Сборщики решают задачи упаковки, компиляции и организации приложения. По мере развития и роста приложения увеличивается и время его сборки – от нескольких минут при пере-сборке в development-режиме до десятков минут при production-сборке. Это оказывает влияние на продуктивность работы и в дальнейшем может значительно увеличить время разработки приложения. Для ускорения процесса сборки предлагается использовать параллелизацию.

## I. СБОРКА WEB-ПРИЛОЖЕНИЯ С ПОМОЩЬЮ WEBPACK

Рассматривается сборка web-приложения с помощью сборщика модулей Webpack, для которого существует инструмент Webpack bundle analyzer [1], позволяющий визуализировать содержимое сборки и оценить размеры включенных файлов (см.рис.1.)

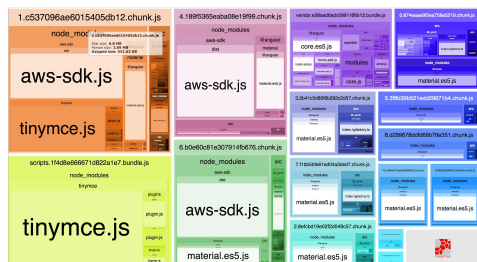


Рис. 1 – Отчет анализатора сборки

В данном примере на процесс сборки было затрачено около 20 секунд и скомпилировано 944 модуля в 1 файл фрагмента. Это означает, что Webpack должен обрабатывать 944 модуля при каждой компиляции.

## II. HAPPYPACK ДЛЯ ПАРАЛЛЕЛИЗАЦИИ СБОРКИ

Библиотека HappyPack ускоряет первоначальную сборку web-пакетов за счет параллельного преобразования файлов. HappyPack предоставляет как плагин, так и загрузчик для выпол-

нения своей работы, поэтому необходимо использовать оба, чтобы включить его. С HappyPack разработчик переключает логику так, что загрузчики передаются в плагин HappyPack и вместо этого сообщает Webpack использовать happypack/loader (см.рис.2.)

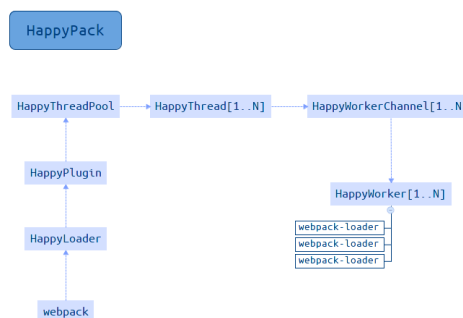


Рис. 2 – Схема выполнения

HappyPack находится между web-пакетом и первичными исходными файлами, где происходит основная часть преобразований загрузчика. Каждый раз, когда Webpack обрабатывает модуль, HappyPack берет его и все его зависимости и распределяет эти файлы по нескольким рабочим «потокам». Эти потоки на самом деле являются простыми узловыми процессами, которые вызывают преобразователь. Когда скомпилированная версия получена, HappyPack передает ее своему загрузчику.

## III. ВЫВОДЫ

Согласно результатам анализа сборки web-приложения до и после оптимизации, использование параллелизации позволит в некоторых случаях ускорить процесс в несколько раз, что позволит сократить время, затрачиваемое разработчиками на сборку web-приложения.

1. Frank Zammetti. Modern Full-Stack Development. Using TypeScript, React, Node.js, Webpack, and Docker. – 2020. – с. 396.
2. Juho Vepsäläinen. SurviveJS - Webpack and React. – 2016. – с. 40-43.

Мазура Анастасия Александровна, магистрант кафедры информационных технологий автоматизированных систем БГУИР, mazura.anastasiya@gmail.com.

Научный руководитель: Гуринович Алевтина Борисовна, заместитель декана факультета информационных технологий и управления БГУИР, кандидат физ.-мат. наук, доцент, gurinovich@bsuir.by.