Chapter 4 – Performance Testing

While the implementation works and can help in detecting the features of requesting UA’s, its inherent goal is to improve the performance of web pages across both existing and future devices and UA’s. This chapter presents how we conducted our performance tests and their results.

# Method

Measuring performance of web sites is a whole research field in its own right. There are many ways of doing it, each aimed at specific parts. Some might target the performance on the backend, while others target the frontend exclusively, looking at the execution time of JS and the size of files sent in the response. Others may not look at response- or execution times at all, but rather do analysis of the content of a web page as it loads to determine the web page’s performance as experienced by the user.

Because our plugin is situated primarily on the backend we will focus on that as the common case, but we must also consider the case where the system encounters an unknown UA and must do tests on the frontend. These two cases are quite different and measuring their performance must thus be approached differently.

Enonic CMS also has its own device detection system built into it. Since our system is meant to replace it we also need to look at it and our system comparatively, to establish the performance impact of using our plugin as a replacement. This needs to be done for both cases mentioned above. Even though our system detects more features than the built-in system, a severe performance hit might be grounds to argue against using our plugin.

## Measuring backend performance

## Measuring frontend performance

## Comparing the built-in system with our plugin

# Results