

EXAMENSARBETE Analyzing front-end performance using Webassembly**STUDENTER** Jacob Nilsson, Andreas Trattner**HANDLEDARE** Lars Bendix (LTH)**EXAMINATOR** Per Andersson (LTH)

En utvärdering av prestanda i webbläsare

POPULÄRVETENSKAPLIG SAMMANFATTNING **Jacob Nilsson, Andreas Trattner**

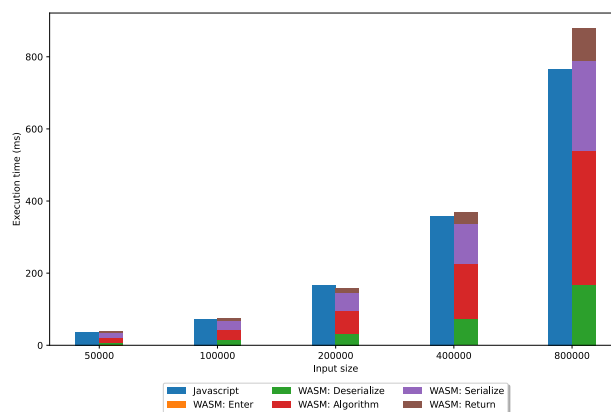
Komplexa webbapplikationer ställer höga krav på prestanda. WebAssembly är en ny teknologi som ska klara av att köra tunga processer mer effektivt än befintliga lösningar.

Hela 98% av dagens webbsidor rapporteras använda sig av programspråket JavaScript för att driva logiska processer. Samtidigt har klimatet online förändrats sedan dess begynnelse, och idag är det inte helt oväntat att en hemsida inte skiljer sig väsentligt från en traditionell applikation.

I vårt examensarbete på IKEA har vi utvärderat hur WebAssembly förhåller sig till JavaScript, med avseende på exekveringstid.

För att utforska detta utförde vi en litteraturstudie för att hitta skillnader på WebAssembly och JavaScript som påverkar exekveringstid. Sedan intervjuade vi anställda på IKEA för att se vilka algoritmer som är viktiga för dem, som också är intressanta givet skillnaderna mellan WebAssembly och JavaScript. Vi skapade sedan ett ramverk för att kunna jämföra dessa algoritmer i både JavaScript och WebAssembly. De algoritmer som användes var bland annat sortering, maskininlärning och generering av bild-data. Dessa implementerades i både WebAssembly och JavaScript.

Figuren visar exekveringstiden för de två teknikerna när en samling texter sorteras i lexikografiskt stigande ordning. De blå kolumnerna visar exekveringstid för JavaScript. De röda kolumnerna visar exekveringstid för WebAssembly. Övriga färger visar tiden det tar för We-



WebAssembly att översätta data då den tas emot och sedan skickas tillbaka till JavaScript. Resultatet visar att WebAssembly är bättre på att utföra den tunga algoritmen. Samtidigt är tiden för att skicka data mellan JavaScript och WebAssembly ofta stor, vilket resulterar i att exekveringstiden för WebAssembly växer. Detta resultat gäller för samtliga processer där komplex data ska utbytas mellan två kontexter. För problem som kan uttryckas med mer enkla strukturer, så som maskininlärning och generering av bilddata, försvinner detta problem, och WebAssembly visar sig då vara en bättre lösning.