Afklaring af muligheder for at lave profilering af JavaScript under Java

Umiddelbart er der følgende tilgange til profilering:

- Profilering via Rhinos debug-api
- Instrumentering af køretidsgenereret kode
- Statistisk sampling i Java, som derefter mappes tilbage til JavaScript-koden.

Profilering via Rhinos debug-api

Rhino har et debug-api, hvor det er muligt at følge med i kørslen af programmet, og derved også lave profiling.

- implementerbarhed - let, skal blot implementere og kalde ind i debug-api'et, lavede en simpel proof-of-concept prototype i løbet af en eftermiddag...
- virker på fortolket (ikke-JIT'et) kode
- tracking kan føres ned på linjeniveau
- højt impact på køretid - kræver at at koden kører fortolket, samt overhead til timing
- kan slås til og fra under kørslen, dog skal koden altid være fortolket.

Instrumentering af genererede class-filer

Vi kan få rhino til at inkludere profilingskode når den jit-kompilere fra JavaScript til Java klasser.

- implementerbarhed - sandsynligvis moderat, kræver ændringer i
- virker på JIT-kompileret kode
- tracking vil nok blive på funktionsniveau
- moderat impact på køretid - der kommer overhead til timing
- kan nok ikke slås fra/til under kørslen, ihvertfald ikke unden genoversættelse

Java statistisk sampling

Går essentielt ud på at lave statistisk sampling på den kørende Java kode. Kørende JavaScript kode identificeres på kaldstakken, og udfordringen her er så at mappe tilbage fra de genererede JavaScript-klasser til JavaScript kildekode.

- implementerbarhed - sandsynligvis let/moderat - kræver en (ser ud til at være lille) ændring i rhinos køretidsgenerering af klasser, så de får navne der kan føres tilbage til kildekoden.
- virker på JIT-kompileret kode
- tracking kun på functionsniveau.
- lavt impact på køretid, bør også kunne køres på drift.
- samplingen kan slås til og fra under kørslen, tildeling af navne der kan køres tilbage til kildekoden burde ikke koste noget i praksis.

Konklusion

Profilering via debug api'et kan implementeres hurtigt hvis det haster meget at vi skal profile noget her og nu.

Ellers giver det nok mest mening at ændre rhinos klassenavngivning så det kan føres tilbage til JavaScript kildekoden, hvorved vi kan lave statistisk sampling med lavt overhead.