QQ邮箱 mail.qq.com

[quickjs-devel] Re: A possible performance issue

发件人: Fabrice Bellard<fabrice@bellard.org>
时　间: 2019年10月30日(星期三) 晚上11:10
收件人: QuickJS<quickjs-devel@freelists.org>

Hi,

The code is slow because each array element access involves a float64 to string conversion which is not optimized in QuickJS. You can see the difference by just using "NISLParameter2 = 1" (0.2 seconds) instead of "NISLParameter2 = 1.541384329975526" (33 seconds).

float64 to string conversion will be optimized at some point (mainly to be independent from the libc printf implementation) but I am not sure it is critical for the performance of real code.

Best regards,

Fabrice.

On 10/30/19 1:59 AM, 姚厚友 wrote:
> Hi,
> When running the testcases bellow, QuickJS takes about 10 times
> execution time of other engines. Maybe, it is an issue of QuickJS.
>
> Best Regards,
> Houyou Yao
>
>
> # Version: quickjs-2019-09-18
>
> # Testcase:
> var NISLFuzzingFunc = function(array, from, to) {
>   var counter = 0;
>   while (to < from) {
>   if (array[to] === array[to]) {
>   to++;
>   }
>   }
> };
> var NISLParameter0 = [true];
> var NISLParameter1 = 4077282;

```
> var NISLParameter2 = 1.541384329975526;
> NISLFuzzingFunc(NISLParameter0, NISLParameter1, NISLParameter2);
>
> # Command:
> quickjs-2019-09-18/qjs testcase.js
>
>
> # Execution time:
> V8: 4098 ms
> spiderMonkey: 3355 ms
> chakra: 2439 ms
> javascriptCore: 3315 ms
> QuickJS: 32380 ms
>
```