

How to evaluate the execution time in high-resolution realtime, nanoseconds

井民全, Jing, mqijing@gmail.com

github: [view](#)

Script 的執行效能需要特別注意, 雖然早就知道了, 可是在瀏覽器中

執行, 更增加了效能評估的 issue, 使用 hrtime 可以 nano-seconds 等級

來量化程式執行效能, 評估不同的優化策略.

Enjoy.

By Jing

Table of contents

[1. Quick](#)

[2. NodeJS](#)

[2.1. Install](#)

[2.2. Code](#)

[2.3. Run](#)

[2.4. Result](#)

[3. Run in Brower](#)

[3.1. Install](#)

[3.2. Code](#)

[3.2.1. Script \(Quick\)](#)

[3.2.2. React](#)

[3.3. Run](#)

[3.4. Result](#)

[4. References](#)

1. Quick

Nodejs version

```
const start:bigint = process.hrtime.bigint();  
...  
const end:bigint = process.hrtime.bigint();  
console.log(`Benchmark took ${end - start} nanoseconds.`);
```

Browser version

Script

```
let start:[number, number] = hrtime();  
  
benchmarking_target_function();  
  
const diff:[number, number] = hrtime(start);  
let dur:bigint = BigInt(diff[0] * 1e9 + diff[1]);  
console.log(`Benchmark tool diff = `, diff);  
console.log(`Benchmark took ${dur/BigInt(Math.pow(10, 6))} milliseconds.`);  
console.log(`Benchmark took ${dur/BigInt(Math.pow(10, 9))} seconds.`);
```

Recact

```
this.__start = hrtime();
```

```
...
const diff:[number, number] = hrtime(this.__start);
this.setState({diff:diff});

// render
<h2>Benchmark took {this.state.diff[0] * 1e9 + this.state.diff[1]} nanoseconds</h2>
```

2. NodeJS

2.1. Install

```
yarn add @types/node
yarn add npm-run-all    # optional
```

2.2. Code

File: index.ts

```
function main() {
    const start:bigint = process.hrtime.bigint(); // 191051479007711n
    setTimeout(() => {
        const end:bigint = process.hrtime.bigint(); // // 191052633396993n
        console.log(`Benchmark took ${end - start} nanoseconds.`); // Benchmark took
1154389282 nanoseconds
        console.log(`Benchmark took ${(end - start)/BigInt(Math.pow(10, 6))}
milliseconds.`);
        console.log(`Benchmark took ${(end - start)/BigInt(Math.pow(10, 9))}
seconds.`);
    }, 1000);
}

main();
```

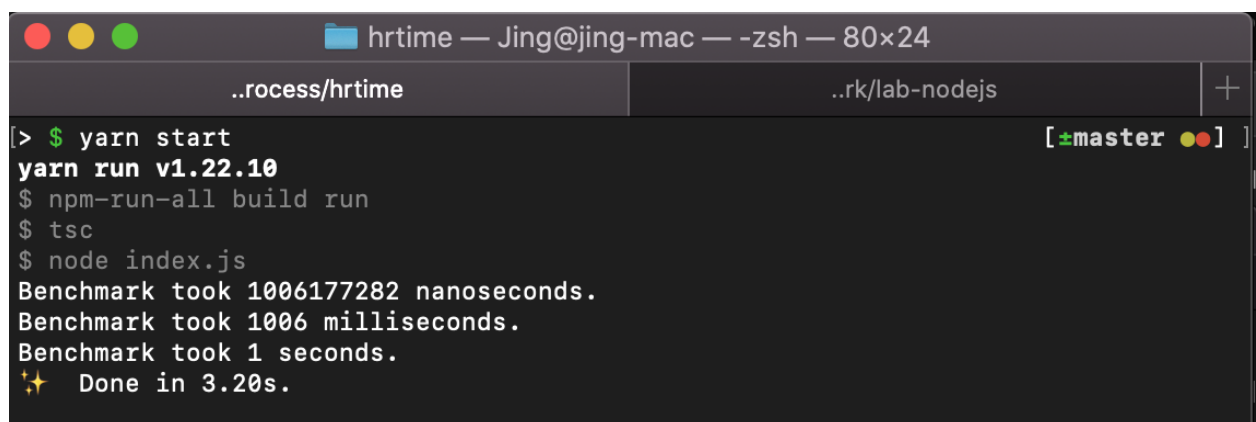
File: package.json

```
{
  "name": "index",
  "version": "1.0.0",
  "main": "index.js",
  "license": "MIT",
  "dependencies": {
    "@types/node": "^14.14.2",
    "npm-run-all": "^4.1.5"
  },
  "scripts": {
    "start": "npm-run-all build run",
    "run": "node index.js",
    "build": "tsc"
  }
}
```

2.3. Run

```
yarn start
```

2.4. Result



A terminal window titled "hrttime — Jing@jing-mac — -zsh — 80x24" with two tabs: "..rocess/hrttime" and "..rk/lab-nodejs". The terminal shows the command `yarn start` being executed, which runs `yarn run v1.22.10`. This triggers a sequence of commands: `$ npm-run-all build run`, `$ tsc`, and `$ node index.js`. The output displays three benchmark results: "Benchmark took 1006177282 nanoseconds.", "Benchmark took 1006 milliseconds.", and "Benchmark took 1 seconds.". The process concludes with a star icon and the text "Done in 3.20s." The terminal's status bar at the bottom right shows "[±master ●●]".

```
[> $ yarn start
yarn run v1.22.10
$ npm-run-all build run
$ tsc
$ node index.js
Benchmark took 1006177282 nanoseconds.
Benchmark took 1006 milliseconds.
Benchmark took 1 seconds.
🌟 Done in 3.20s.
```

3. Run in Brower

3.1. Install

```
npx create-react-app my-app --template typescript  
yarn add browser-hrtime
```

3.2. Code

3.2.1. Script (Quick)

```
let start:[number, number] = hrtime();  
  
benchmarking_target_function();  
  
const diff:[number, number] = hrtime(start);  
let dur:bigint = BigInt(diff[0] * 1e9 + diff[1]);  
console.log('Benchmark tool diff = ', diff);  
console.log(`Benchmark took ${dur/BigInt(Math.pow(10, 6))} milliseconds.`);  
console.log(`Benchmark took ${dur/BigInt(Math.pow(10, 9))} seconds.`);
```

3.2.2. React

File: App.tsx

```
import React, {Component} from 'react';  
import hrtime from 'browser-hrtime';  
  
type MyState = {diff: [number, number]};  
class App extends React.Component<{}, MyState> {  
  private __start:[number, number];  
  
  constructor(props:any) {  
    super(props);  
    this.state = {diff: [0, 0]};  
    this.__start = hrtime();  
  }  
}
```

```

    setTimeout(() => {
      const diff:[number, number] = hrtime(this.__start);
      this.setState({diff:diff});
      let dur:bigint = BigInt(diff[0] * 1e9 + diff[1]);
      console.log(`Benchmark took ${dur/BigInt(Math.pow(10, 6))} milliseconds.`);
      console.log(`Benchmark took ${dur/BigInt(Math.pow(10, 9))} seconds.`);
    }, 1000);
  }

  render() {
    let duration:number = this.state.diff[0] * 1e9 + this.state.diff[1];
    return (
      <div>
        <h1>Hello, world!</h1>
        <h2>Benchmark took {this.state.diff[0] * 1e9 + this.state.diff[1]}
nanoseconds</h2>
        <h2>Benchmark took {duration/Math.pow(10,6)} milliseconds </h2>
        <h2>Benchmark took {duration/Math.pow(10,9)} seconds </h2>
      </div>
    );
  }
}
export default App;

```

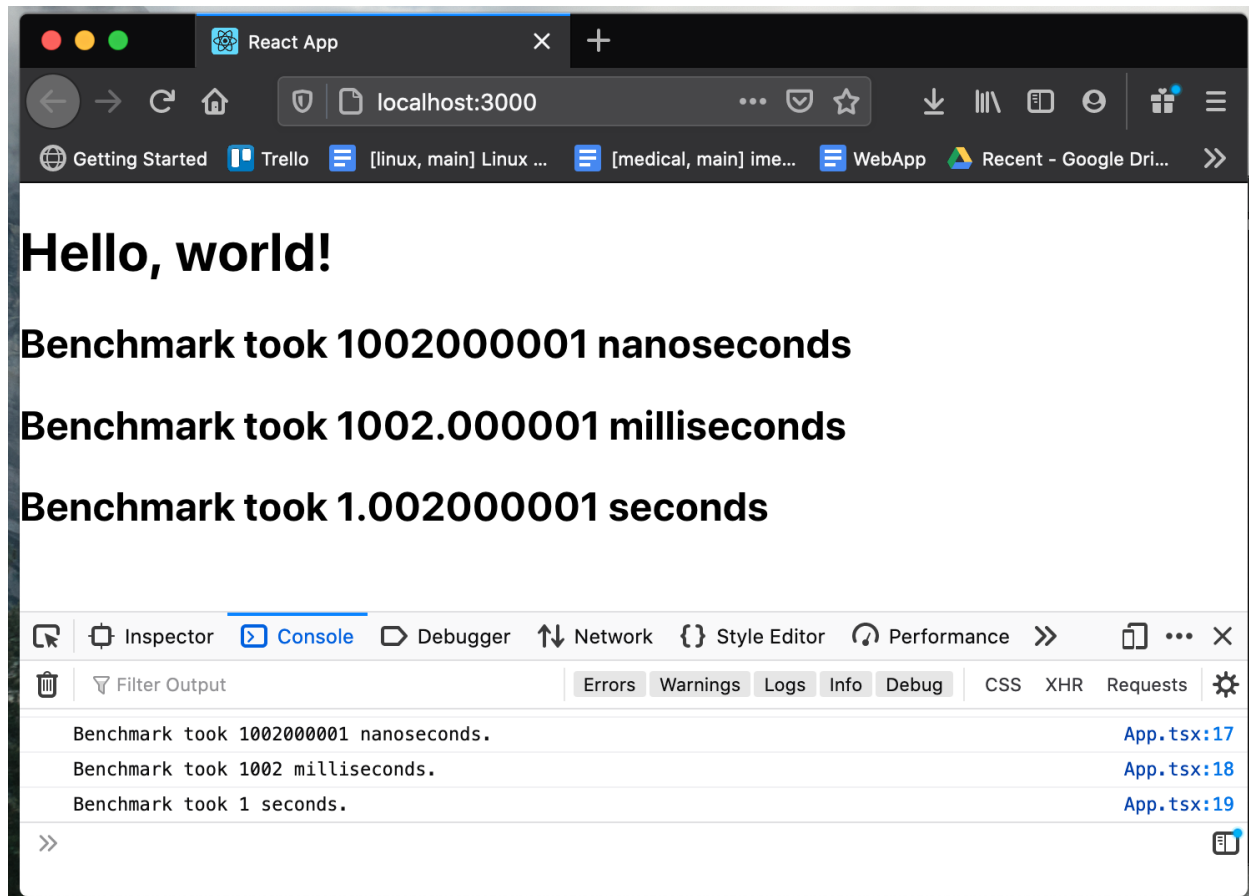
3.3. Run

```

yarn add brower-hrtime
yarn start

```

3.4. Result



4. References

1. https://nodejs.org/api/process.html#process_process_hrtime_bigint
2. <https://www.typescriptlang.org/docs/handbook/release-notes/typescript-3-2.html>
3. <https://github.com/jsdom/w3c-hr-time/pull/13>
4. <https://github.com/cabinjs/browser-hrtime>