



JavaScript Runtime for non-Browser (server)

NodeJS



Deno



Bun





<https://github.com/nodejs/node>



ผู้สร้าง NodeJS

Ryan Dahl



Dahl (2010; age 29).

Born	1981 (age 42–43) San Diego, California, U.S
Education	UC San Diego University of Rochester
Known for	Node.js & Deno

• NodeJS

เมื่อปี 2009 **Ryan Dahl** ต้องการเพิ่มประสิทธิภาพ และแก้ปัญหาข้อจำกัดของเซิร์ฟเวอร์เว็บรุ่นเก่า เขาเลยมีความคิดที่จะสร้าง **Node.js**

ที่สามารถรองรับการทำงานแบบ **non-blocking**

เพื่อให้เซิร์ฟเวอร์ทำงานได้เร็วขึ้นและมีประสิทธิภาพมากขึ้น

โดยใช้ **V8 JavaScript engine** ของ **Google**

มาเป็นแกนกลางในการพัฒนา **Node.js**

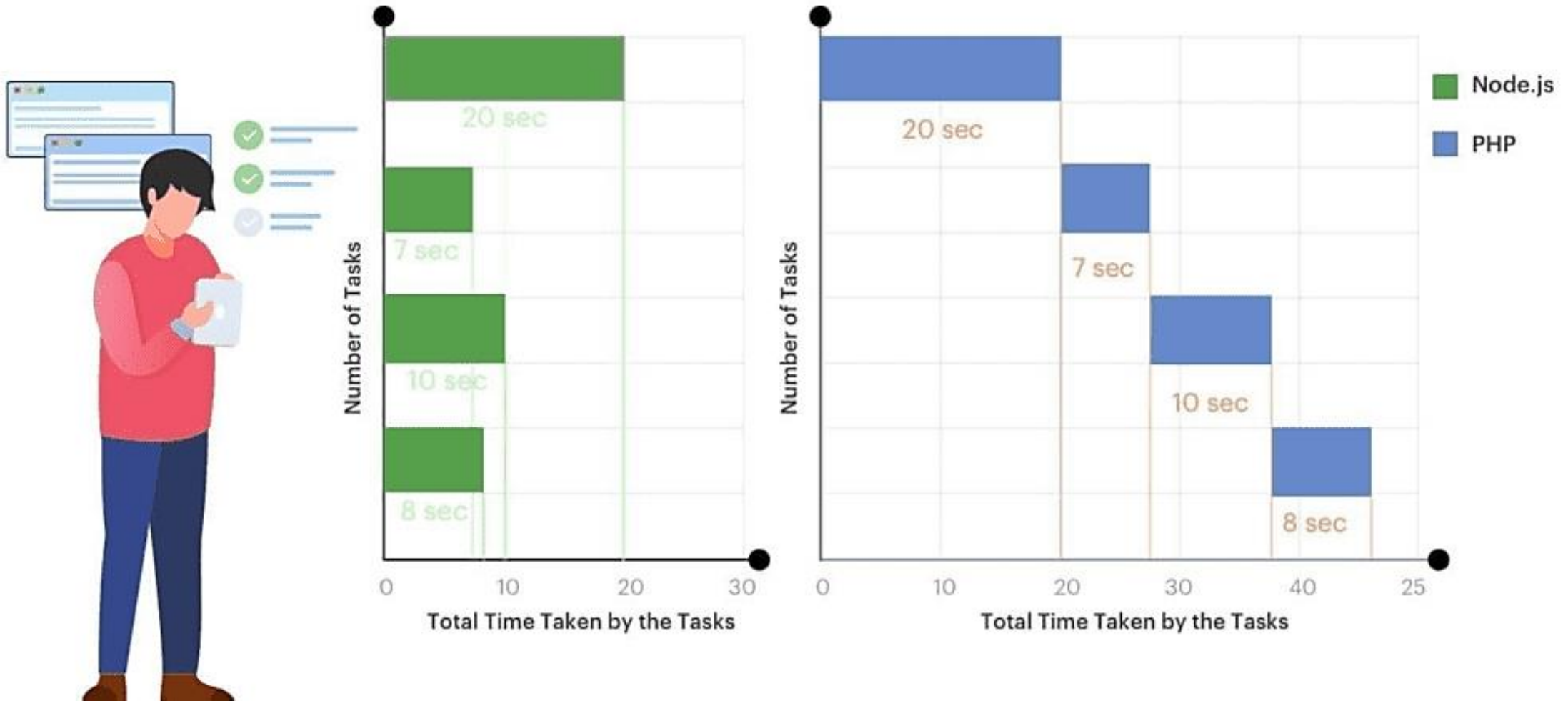
ทำให้ **Node.js** กลายเป็นแพลตฟอร์ม

ที่รองรับการเขียนโปรแกรมแบบ **asynchronous** ได้ดี

และเป็นการปฏิวัติวงการการพัฒนาเว็บเซิร์ฟเวอร์เลยทีเดียว!

https://www.youtube.com/watch?v=jo_B4LTHi3I

Node.JS | non-blocking I/O vs PHP Speed



Blocking I/O Model



Pseudocode:

```
order1 = db.query("SELECT *  
FROM menu WHERE preference =  
most")
```

```
order1.process
```

```
order2.process
```


Non-Blocking I/O Model



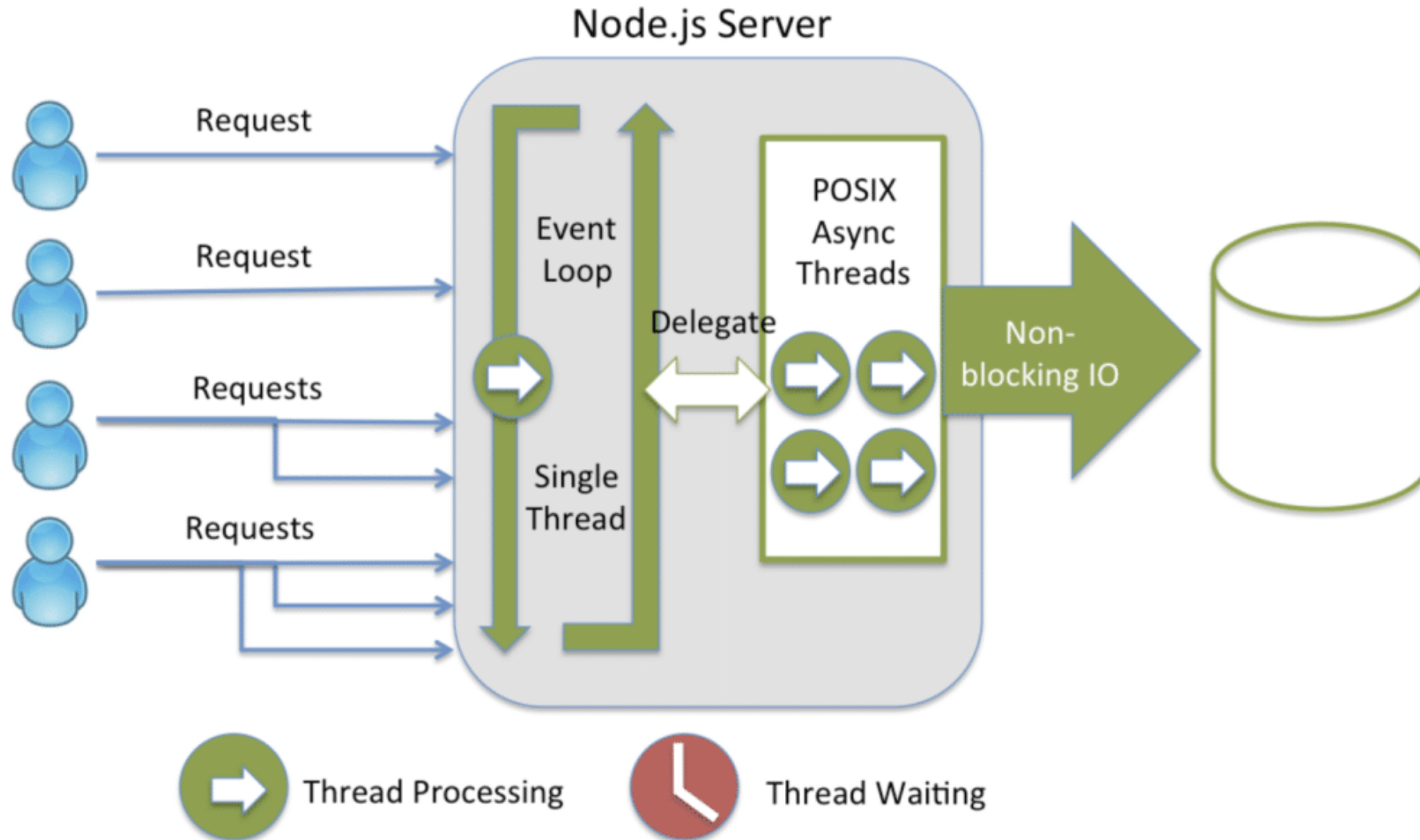
Node code

```
console.log('Hello');  
  
setTimeout(function () {  
  console.log('World');  
}, 5000);  
  
console.log('Bye');
```

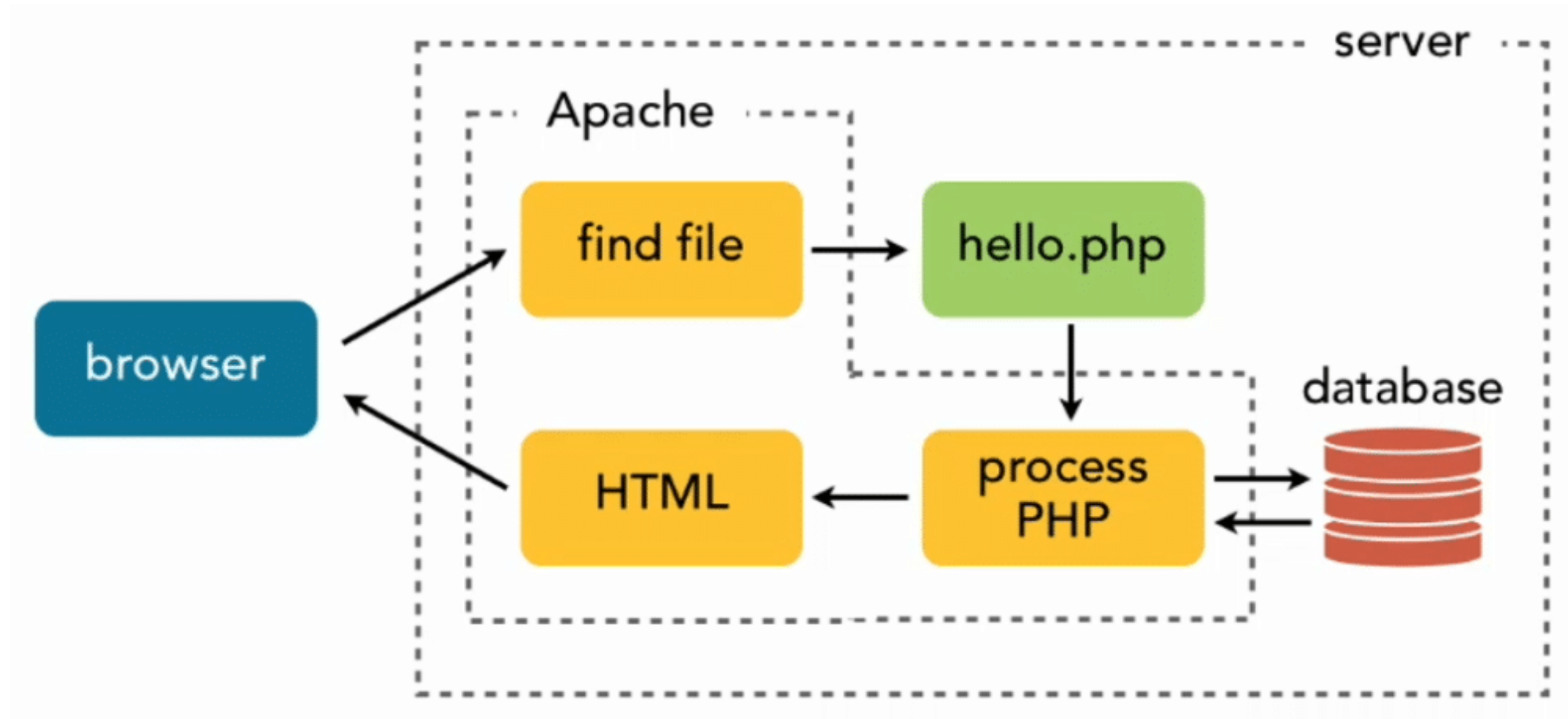
// Outputs:
// Hello
// Bye
// World

***Allows for high
concurrency***

Node.js Server



PHP



Overview Blocking VS Non-Blocking

- Blocking (Synchronous) รอจนงานเสร็จ ค่อยทำงานบันทึก ถัดไป

çỘηรฺทຸ ຈຸຮຸ sêṛḇîsê ηỘđê ຈຸຮຸ
çỘηรฺทຸ ດ້າຖ້າ ຈຸຮຸ sê້đGîlêຮຸຍຸຸ ຈຸ້lê ḇḇ Ẉໂຮຸໂຮ
çỘηรฺทຸໂê Ểໂ ດ້າຖ້າ

- Non-Blocking (Asynchronous) ไม่รอ แต่เมื่อทำงานเสร็จแล้ว จะเรียก Callback

çỘηรฺทຸ ຈຸຮຸ sêṛḇîsê ηỘđê ຈຸຮຸ
ຈຸຮຸ sê້đGîlê ຈຸ້lê ḇḇ êss ດ້າຖ້າ
îḡ êss ṭḥsໂx êss
çỘηรฺทຸໂê Ểໂ ດ້າຖ້າ

ผู้สร้าง Deno

 <https://github.com/denoland/deno/>



• Deno

หลังจากสร้าง **Node.js** ไปได้สักพัก **Ryan Dahl** เริ่มเห็นข้อบกพร่องและข้อจำกัดหลายอย่างของมัน เช่น การจัดการแพ็คเกจที่ซับซ้อน ความปลอดภัยที่ไม่ดี เขาจึงตัดสินใจเริ่มต้นใหม่ในปี 2018 ด้วย **Deno** โดยมีเป้าหมายในการแก้ไขข้อผิดพลาดเหล่านี้

Deno ถูกพัฒนาขึ้นด้วยภาษา **Rust**

และรองรับ **TypeScript** ได้โดยตรง

ซึ่งทำให้โปรแกรมเมอร์ไม่ต้องเสียเวลาในการตั้งค่าเพิ่มเติม และมีการออกแบบให้มีความปลอดภัยมากขึ้น โดยการกำหนดสิทธิ์การเข้าถึงไฟล์และเครือข่าย

ผู้สร้าง Bun



<https://github.com/oven-sh/bun>



Jarred Sumner

- ในปี 2022 Jarred Sumner มีเป้าหมายในการสร้างแพลตฟอร์มที่เร็วกว่า Node.js และ Deno เขาจึงพัฒนา Bun ขึ้น

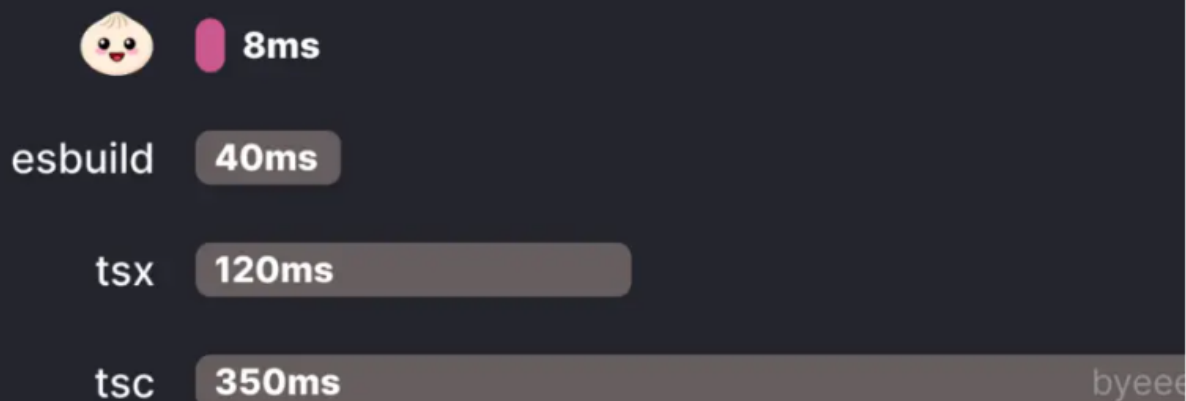
โดยใช้ WebKit's JavaScriptCore Engine และภาษา Zig เขาเน้นไปที่การทำให้ Bun มีความเร็วสูงสุดในการเริ่มต้นและการทำงาน ซึ่งเร็วกว่า Node.js ถึง 4 เท่า!

Bun ยังมีการสนับสนุน TypeScript และ JSX โดยตรงและสามารถใช้แพ็คเกจจาก npm ได้ ทำให้โปรแกรมเมอร์สามารถใช้งานได้ง่ายและรวดเร็วกว่าเดิม

Bun

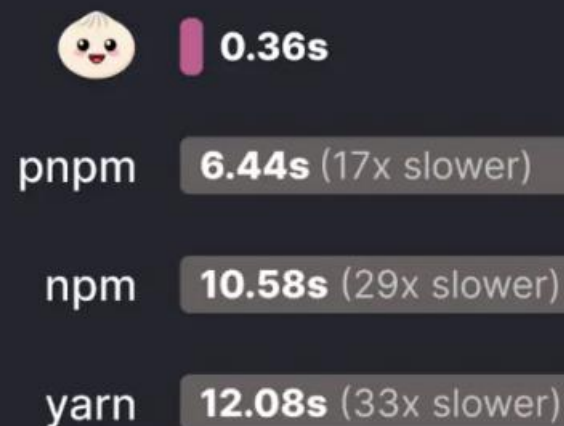
Startup speed

With the growing usage of serverless computing, startup speed becomes increasingly more important. Bun runs up to 4x times faster than Node.js for a simple “Hello world” script, and up to 5x faster than esbuild + Node.js when using TypeScript, since Bun supports it natively:



Package manager compatible with npm

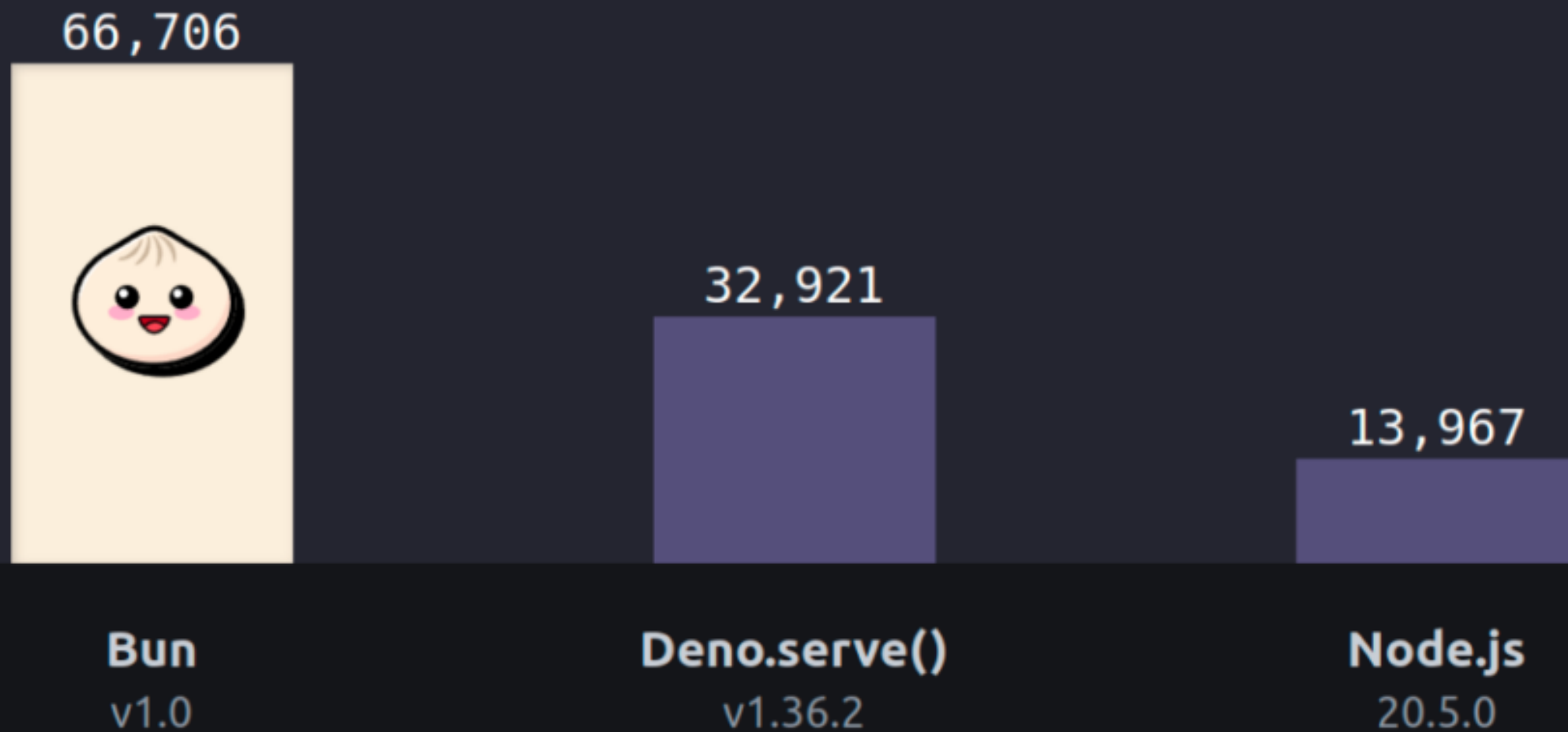
The [npm registry](#) is the largest software registry in the world, with over packages. Bun taps into this registry by providing its own npm-compatible manager, which, of course, is blazingly fast: up to 29x faster than npm.



<https://www.scalablepath.com/nodejs/bun-node-alternative>

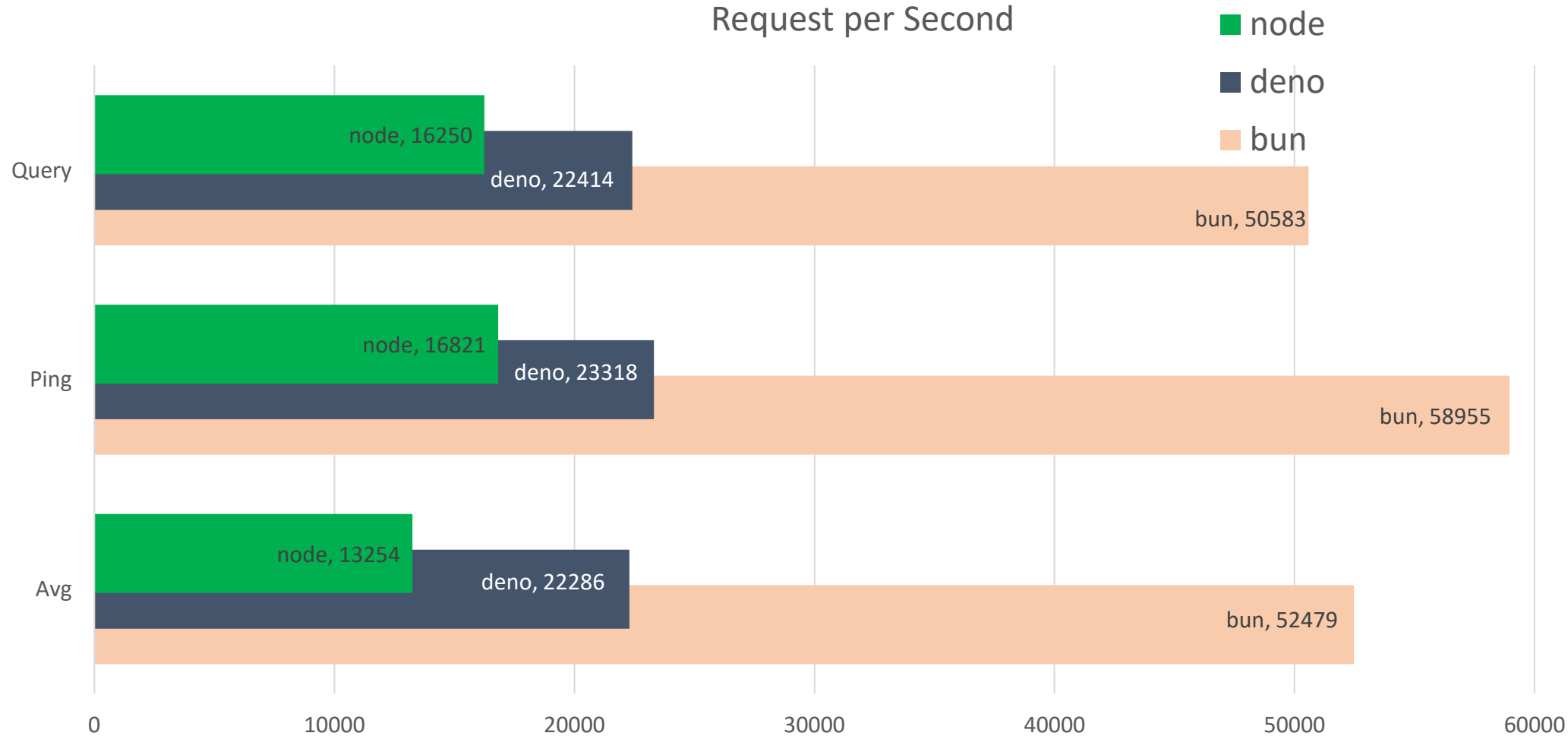
Server-side rendering React

HTTP requests per second (Linux x64)



HTTP Benchmark

Request per Second





Deno vs. Node.js vs Bun: Performance (Latency - Throughput - Saturation - Availability)



Anton Putra

ผู้ติดตาม 6.97 หมื่น คน

สมัคร



ติดตามแล้ว



1.7 พัน



แชร์



ดาวน์โหลด



ขอบคุณ



คลิป



From: <https://www.youtube.com/watch?v=yJmyYosyDDM>

Side by Side



NodeJS

Deno

Bun

Initial Release

2009

2018

2021

Runtime

JavaScript

JavaScript

JavaScript

Engine

Google V8

Google V8

JavaScriptCore

Type Script

Experimental

Build-in

Build-in

Write In

JavaScript, C++,
Python, C

JavaScript,
Rust, C++

Zig, C++,
TypeScript, C

Package

Manager

npm install

Deno install

bun install

Current Version

V 23.0.0

V 2.0.2

V1.1.31

Node JS



ข้อดี:

Community ใหญ่: มี **library** และ **package** มากมายที่พร้อมใช้ผ่าน **npm**.

Support: มีชุมชนผู้ใช้ที่แข็งแกร่งและเอกสารที่มากมาย.

Stable: เป็นแพลตฟอร์มที่มีการพัฒนาและใช้งานมาอย่างยาวนาน.

Performance: แม้ว่าจะไม่เร็วที่สุด แต่สามารถจัดการกับการทำงานพร้อมกันได้ดี.

ข้อเสีย:

Security: มีความปลอดภัยพื้นฐาน ไม่เหมาะสำหรับการใช้งานที่ต้องการความปลอดภัยสูง.

Type Script: ตั้งแต่ **V22** จะไม่สามารถใช้ **Type Script** ได้โดยตรง

File management: การเข้าถึงระบบไฟล์มีข้อจำกัดและไม่ปลอดภัยเท่าที่ควร.



Deno

ข้อดี:

Security: มีการออกแบบความปลอดภัยที่ดีกว่า **Node.js** โดยกำหนดสิทธิ์การเข้าถึงไฟล์และเครือข่าย.

TypeScript: รองรับ **TypeScript** โดยตรง ไม่ต้องการการตั้งค่าเพิ่มเติม.

API ที่ทันสมัย: มี **API** ที่รองรับการทำงานกับระบบสมัยใหม่เช่น **WebAssembly**.

ข้อเสีย:

Community เล็ก: แม้ว่าจะกำลังเติบโตแต่ยังไม่มี **library** และ **package** ที่มากมายเท่า **Node.js**.

Compatibility : ไม่สามารถใช้งานแพ็คเกจจาก **npm** ได้โดยตรง.

Bun



ข้อดี:

Fast: เร็วที่สุดในสามแพลตฟอร์ม โดยเฉพาะในการเริ่มต้นและการทำงาน.

TypeScript, JSX: รองรับการทำงานกับ TypeScript และ JSX ได้โดยตรง.

Compatibility: สามารถใช้ npm และมีความเข้ากันได้กับ Node.js.

ข้อเสีย:

Community เล็ก: ระบบนิเวศยังเล็กและยังไม่มีการใช้งานอย่างแพร่หลาย.

Security: มีคุณสมบัติความปลอดภัยจำกัดเมื่อเทียบกับ Deno.

Install

- **NodeJS**

[https://nodejs.org/en/download/raspberry](#)

- **Deno**

[https://deno.com/install-linux](#)

- **Bun**

[https://bun.sh](#)

Example NodeJS

şêswês kş

```
çỘηşţ çsêắţêşêswês sêrụîsê  ηỘđê hţţř
çỘηşţ hỘşţηắηê , , ' . . ,
çỘηşţ ắỘşţ , . . .
çỘηşţ şêswês çsêắţêşêswês sêr sêş
sêş şţắţụşCỘđê , . .
sêş şêţHêắđêş CỘηţêηţ Ỡỳắê  ẵêyţ ắắîη
sêş êηđ Hê'łỘ ỠỘş'đ ỠỘđê Kş
```

```
şêswês ỉỉşţêη ắỘşţ hỘşţηắηê
çỘηşỘłê ỉỘậ şêswês sụnηîηậ ắţ hţţř hỘşţηắηê ắỘşţ
```

ηỘđê şêswês kş

hţţř ỉỘặắỉhỘşţ , . . .

Example Deno

şêswês tş

ğunçtîîon hăṇđl'ês sêr Rêrụêşţ Rêşr̥oṇşê
sêţbun̄ nêx Rêşr̥oṇşê Hêl'lo Wôslđ Dêṇo

Dêṇo şêswê hăṇđl'ês

đêṇo sụn ắl'lox nêţ şêswês tş

hţţr̥ l'ỏắl'hoşţ ...

Example Bun

îṇḍêy ṭṣy

çỘṇṭṭ ṣêswêṣ Bụṇ ṣêswê
ṛỘṭṭ ‘ . . . ,
ğêṭṭçḥ ṣêṛụêṭṭ
ṣêṭḃṣṇ ṇêx ṚêṣṛỘṇṣê Wêl'çỘṇê ṭỘ Bụṇ

çỘṇṣỘlê ṭỘḡ L'îṣṭṭêṇîṇḡ ḡṇ ḥṭṭṭṛ ṭỘçảṭḥỘṭṭ ṣêswêṣ ṛỘṭṭ

çỘṇ ṇḍêy ṭṣy

ḥṭṭṭṛ ṭỘçảṭḥỘṭṭ ‘ . . . ,

Reference

- <https://nodejs.org/>
- <https://deno.com/>
- <https://bun.sh/>

- <https://betterstack.com/community/guides/scaling-nodejs/nodejs-vs-deno-vs-bun/#what-is-node-js>
- <https://www.slideshare.net/slideshow/intro-to-node-and-non-blocking-io/24350440>
- <https://tarunjain07.medium.com/non-blocking-vs-blocking-i-o-notes-904559ae5b9e>
- <https://dev.to/probir-sarkar/honojs-benchmark-nodejs-vs-deno-20-vs-bun-which-is-the-fastest-413j>

Summary



- NodeJS, Deno, Bun is JavaScript Runtime for back-end non-browser Java Script

Single Thread, non-blocking I/O



- NodeJS Large community (Ryan Dahl) 2009
- Deno Rust more Secure (Ryan Dahl) 2018
- Bun Zig Build for Fast (Jarred Sumner) 2022