

ВИКОРИСТАННЯ ПЛАТФОРМИ NODE.JS ДЛЯ РОЗРОБКИ СУЧАСНИХ ВЕБ-СЕРВІСІВ

Остапенко С.О., Прищепя Т.О.

Інститут телекомунікаційних систем НТУУ «КПІ», Україна

E-mail: serg.ostapenko94@gmail.com

Usage of Node.js platform for web-application development

Node.js - platform for developing high-load server-side web applications. Synchronous and asynchronous input - output model. Advantages of using technology for developing multi-user applications. Common mistakes in the use of Node.js. Huge growth in popularity of technology.

Завдяки зростаючій популярності, мова JavaScript в наш час активно розвивається, а сучасні бізнес-процеси, зазнали значних змін в порівнянні з недавнім минулим. Ті речі, які ми сьогодні можемо робити в Інтернеті за допомогою JavaScript, який працює як на сервері, так і в браузері, було складно собі уявити ще кілька років тому.

Node.js – це серверна платформа, що призначена для створення масштабованих розподілених серверних додатків, яка використовує подієво-орієнтовану архітектуру та неблокуючу асинхронну взаємодію[3].

Головна ідея Node.js: використання неблокуючого подієво-орієнтованого вводу / виводу, щоб залишатися легким та ефективним під час роботи з додатками, що обробляють великі обсяги даних в реальному часі і функціонують на розподілених пристроях. Все це означає, що Node.js не є платформою на всі випадки життя, яка буде домінувати в світі веб-розробки. Навпаки, вона використовується лише для вирішення завдань певного типу[1]. Якщо Ваш додаток використовує важкі обчислення, то Вам варто відмовитися від використання Node.js, так як дані операції анулюють всі переваги даної платформи. Та якщо Ви створюєте систему, яка підтримує взаємодію з багатьма одночасними підключеннями, причому із двунправленою взаємодією, то рано чи пізно Вам знадобиться організація асинхронного доступу. Так, саме доступу - під асинхронністю розуміють асинхронний доступ до ресурсів - файлів, сокетів, даних. За класичною схемою це відбувається так:

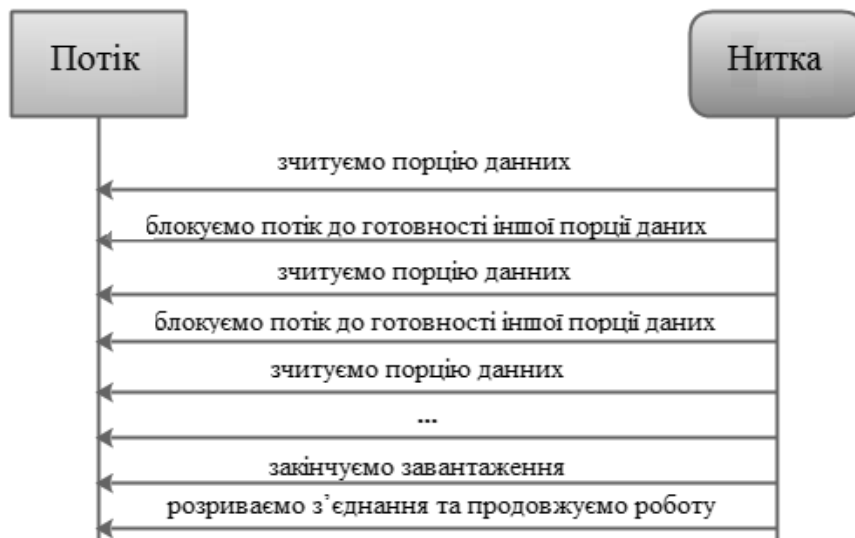


Рис. 1. Класична схема доступу до даних.

При цьому під час виконання програми виникає «блокування», яке виникає під час встановлення з'єднання з ресурсом і (особливо) під час зчитування з нього даних. Під час такого блокування потік виконання простоює, і це, безумовно, не є раціональним використанням ресурсів.

Для подолання даної проблеми була розроблена добре всім відома багатопотокова модель[4].

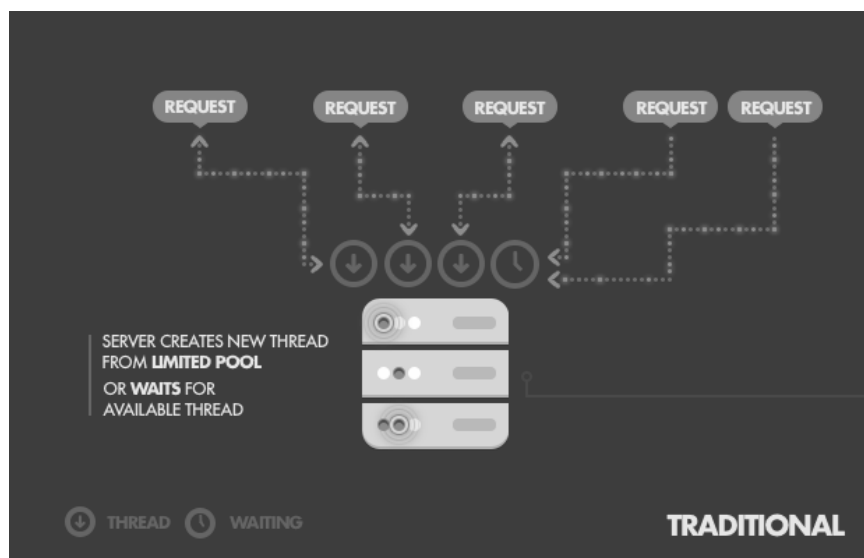


Рис. 2. Багатопотокова модель вводу / виводу.

Асинхронна модель вирішує проблеми доступу принципово по іншому. Вона побудована на циклічній черзі подій (eventloop). При виникненні деякої події (прийшов запит, відповідь від бази даних, зчитування порції даних з файлу) вона розміщується в кінець черги. Потік виконання, що обробляє цей цикл, виконує код, пов'язаний з наступним подією, і, в свою чергу, розміщує його в кінець. Це відбувається до тих пір, поки черга не спорожніє[2].

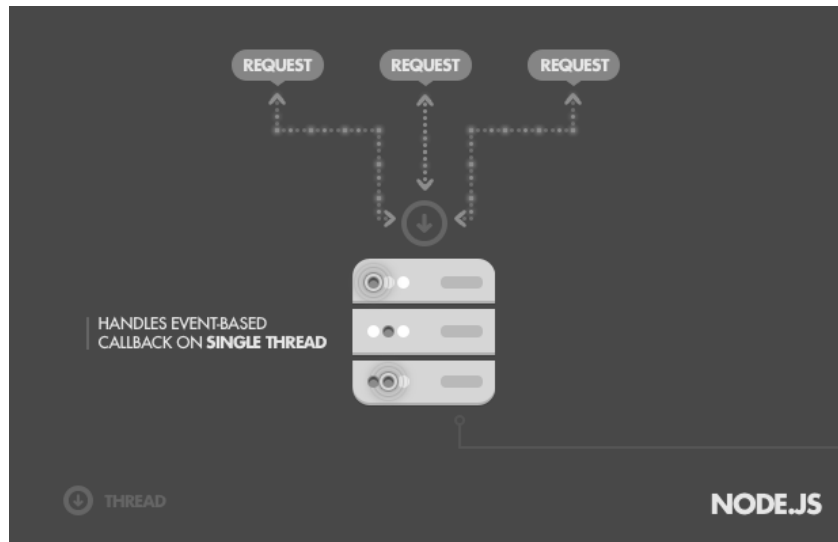


Рис. 3. Асинхронна модель вводу / виводу.

На даний момент ця технологія розвивається дуже швидко, так як має підтримку всесвітньо-відомих компаній, таких як: HBO, PayPal, Netflix, Google, Microsoft, IBM та багатьох інших. В 2015 році платформа досягла версії 4.0.0. Відразу після цієї події компанія IBM офіційно заявила про купівлю компанії StrongLoop, яка з 2013 року здійснює професійну підтримку Node.js для корпоративних клієнтів[5]. IBM вважає необхідним інтегрувати функції Node.js в свою лінійку програмних продуктів для розробки корпоративних додатків як в хмарі, так і всередині підприємства.

Node.js – це платформа, що використовується здебільшого для вирішення проблем багатокористувацьких сервісів які не потребують важких обчислень. Вона успішно застосовується для розробки веб-чатів, браузерних ігор, інформаційних порталів. В мережі можна знайти багато негативних відгуків від розробників веб-сервісів на рахунок даної технології. Здебільшого проблеми з використанням Node.js виникають тоді, коли сама платформа використовується не для тих задач, під які вона створювалась.

Література

1. Сухов К. К. Node.js. Путеводитель по технологии. – 2015. – С. 12-15.
2. Ethan Brown Web Development with Node and Express: Leveraging the JavaScript Stack. – 2014. – С. 14-16.
3. Node.js и Express как они есть [Електронний ресурс] // Издательский дом «Питер». – 2015. - Режим доступа до ресурсу: <https://habrahabr.ru/company/piter/blog/265649/>.
4. Why The Hell Would I Use Node.js? [Електронний ресурс] // Tomislav Caplan. – 2015. - Режим доступа до ресурсу: <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>.
5. IBM сделает платформу NodeJS стандартом корпоративной разработки [Електронний ресурс] // компания IBM. – 2015. - Режим доступа до ресурсу: <https://habrahabr.ru/company/ibm/blog/266647/>.