Keivan Damirchi • Following
Software Developer
19h •

Handling High Traffic In ASP.NET Core

The following are tools and techniques that will help you handle high traffic in a .NET Core app:

1 - Caching:

Caching stores frequently accessed data in memory or on disk to reduce database load and improve response times.

Tools & Techniques:

- Redis [https://lnkd.in/ex7agR2g]
- ASP.NET Core Response Caching Middleware [https://lnkd.in/egf5yw-B]
- In-Memory Cache [https://lnkd.in/eXFUDAnu]

2 - Load Balancing:

Load balancing distributes incoming network traffic across multiple servers to improve application performance and availability.

Tools & Techniques:

- Nginx [https://lnkd.in/ewKW36Xu]
- HAProxy [https://lnkd.in/eM9pcRu6]
- Azure Load Balancer [https://lnkd.in/eSwSZ4C2]

3 - Asynchronous Programming:

Asynchronous programming runs multiple tasks concurrently, allowing the application to handle more requests and improve performance.

Tools & Techniques:

- Task Parallel Library (TPL) [https://lnkd.in/eZ4D8Mpn]
- async/await [https://lnkd.in/e_AZEewx]

- Reactive Extensions (Rx) [**https://lnkd.in/e9QVJUgU**]

4 - Message Queues:

Message queues decouple tasks and process them asynchronously, improving application performance and scalability during traffic spikes.

Tools & Techniques:

- RabbitMQ [**https://lnkd.in/e9naBiiR**]
- Azure Service Bus [**https://lnkd.in/en-CZRye**]
- Kafka [**https://lnkd.in/eSQ4NZDm**]

These are just some of the tools and techniques for handling high traffic on your apps. Note that sometimes it is not necessary to use these tools and methods and better performance can be achieved with simpler methods.

For further reading:

My Telegram: **https://t.me/keivandev**

My Medium: **https://lnkd.in/esQhC2nP**

My GitHub: **https://lnkd.in/eHsjJNh4**

#scalability #dotnetcore

Handling High Traffic In

ASP.NET Core

Tools and techniques to manage high traffic in .NET Core apps

Caching

Caching stores frequent data in memory or disk to reduce database load & improve response times.

Tools & Techniques

- **Redis**
- **Response Caching Middleware**
- **In-Memory Cache**

Asynchronous Programming

Asynchronous programming runs tasks concurrently, boosting performance and handling more requests.

Tools & Techniques

- **Task Parallel Library (TPL)**
- **async/await**
- **Reactive Extensions (Rx)**

Message Queues

Message queues decouple tasks & processing them asynchronously to improve performance during traffic spikes.

Tools & Techniques

- **RabbitMQ**
- **Azure Service Bus**
- **Kafka**

Load Balancing

Load balancing spreads traffic across servers to boost performance & availability.

Tools & Techniques

- **Nginx**
- **HAProxy**
- **Azure Load Balancer**

Quick Tips

Keivan Damirchi t.me/keivandev

158 4 comments 28 repos...