

کد این بخش اول در فایل `multithread.py` آورده شده است. زمان هایی که به دست آمد، به شکل زیر است:

3.4, 2.2, 4.6, 2.4, 2.2, 2.5, 4.1, 4.2, 3.2, 2.5

که میانگین آن ها برابر با 3.13 ثانیه است.

حال کد آماده ی گذاشته شده را ۱۰ بار ران می کنیم که اعداد آن به شکل زیر است:

3.6, 1.6, 2.1, 4.2, 1.7, 2.9, 1.9, 1.5, 1.8, 2.1

که میانگین آن برابر با 2.34 است.

وقتی که مقایسه را انجام دهیم، می بینیم که در حالت `async` بهتر شده است و منطقی هم بوده چون که همانطور که در بخش تئوری گفتیم، برای زمان هایی که می خواهیم چند کار را به شکل کاملاً مستقل انجام دهیم، `async` گزینه بهتری است و در این مثال هم که درخواست زدن هر کدام ربطی به دیگری ندارد و می تواند به شکل مستقل انجام شود و در ضمن در `vertical scalability` این روش `async` بهتر جواب می دهد که این مثال هم در این مورد جای دارد. و در کل در `async` ما یک `thread` می سازیم و در آن همه `request` ها را `dispatch` می کنیم ولی در روش `multithread` هر ترد برای هر `request` ساخته می شود و این ساخت و پاک کردن ترد زمان بر است.