**Part2:**

In the second part of the assessment, imagine that size of the JSON input has increased significantly, APIs will not work for this approach, and we decided to use a messaging system. Please implement a solution based on messaging if time permits. If not, at least share a design with a short paragraph on how you would solve the issue of scale and timeouts present in APIs.

The solution for that is to use multiprocessing or multithreading or asyncio

**Multiprocessing** uses two or more CPUs to increase computing power. [1]

**Multithreading** creating computing threads of a single process to increase computing power which It allows you to execute multiple processes concurrently. Multiple threads of a single process are executed concurrently [2]

**Asyncio** helps making foundation for multiple Python asynchronous frameworks that provide high-performance network and webservers, database connection libraries, distributed task queues [3]

Choosing between these three methods depends on working scenarios for each function

CPU Bound => Multiprocessing

I/O Bound, Fast I/O, Limited Number of Connections => multi-Threading

I/O Bound, Slow I/O, Many connections => Asyncio

**Reference**:

[1]"multiprocessing | Definition & Facts", *Encyclopedia Britannica*, 2022. [Online]. Available: https://www.britannica.com/technology/multiprocessing. [Accessed: 14- May- 2022].

[2]"What Is Multithreading: A Guide to Multithreaded Applications | TotalView by Perforce", *TotalView by Perforce*, 2022. [Online]. Available: https://totalview.io/blog/multithreading-multithreaded-applications. [Accessed: 14- May- 2022].

[3]"Implementing Async Features in Python - A Step-by-step Guide", *Velotio.com*, 2022. [Online]. Available: https://www.velotio.com/engineering-blog/async-features-in-python. [Accessed: 14- May- 2022].