**So sánh bất đồng bộ và đồng bộ**

1. **Thử nghiệm demo chương trình nhỏ trên về hiệu suất giả lập 2 chương trình ngoài**
2. **Bất đồng bộ**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

* **1000 request 1 lúc chỉ xử lí trong 1,2189981 giây.**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Đồng bộ**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

* **10 request nhưng xử lí trong 10,1634899 s**
* **A screenshot of a computer

  AI-generated content may be incorrect.**

1. **Thử nghiệm trực tiếp với VN System khi có nhiều request trên**

* Hiện tại hàm ExcuteQuery đang dùng chung cho cả hệ thống mỗi lần chạy thì trung bình chỉ mất 0.01 mili giây. => Công ty hiện tại khoảng 3000 người nhưng số lượng sử dụng hệ thống trong 1 lúc thì khoảng nhiều nhất cỡ 500 – 1500 người là nhiều.
* Giả sử 3000 người cùng sử dụng 1 lúc thì 3000\*0.01 = 30 mili giây => Nhìn chung thì người thứ 3000 phải đợi cỡ 30 mili giây ( Không đáng kể) chưa thể hiện được sự khác biệt giữa đồng bộ và bất đồng bộ.

Nhìn xa hơn 1 chút trong tương lai khi nhân viên nhiều lên hoặc số lượng dữ liệu nhiều lên: Khi dữ liệu lên 1 triệu dòng hoặc xuất báo cáo nếu như mất 1s để câu select có thể được chạy xong thì người thứ 3000 sẽ phải đợi 3000 giây = 50 phút để có được dữ liệu ( Hướng xử lí thì có thể quay lại **dùng bất đồng bộ** hoặc **tối ưu hóa câu query**)

**Demo với việc sử dụng bất đồng bộ và đồng bộ trên VN System**

Thay thế một câu query bất kì có sleep(10): Mục đích giả sử câu query với thời gian 5s rồi xem độ treo hệ thống giữa sử dụng đồng bộ và bất đồng bộ bị ở trên UI.

1. **Giải pháp debug khi dùng bất đồng bộ**

* Thêm delay theo số giây cần quan sát **request 1,** giúp dễ theo dõi **request 2** hơn và đặt nhiều breakpoint và debug như bình thường. Có thể test bằng postman hoặc 1 hàm test hoặc tool nếu muốn.
* Dùng thêm "Threads Window" để xem request nào chạy trên thread nào. **Debug** → **Windows** → **Threads** (Ctrl + Alt + H).