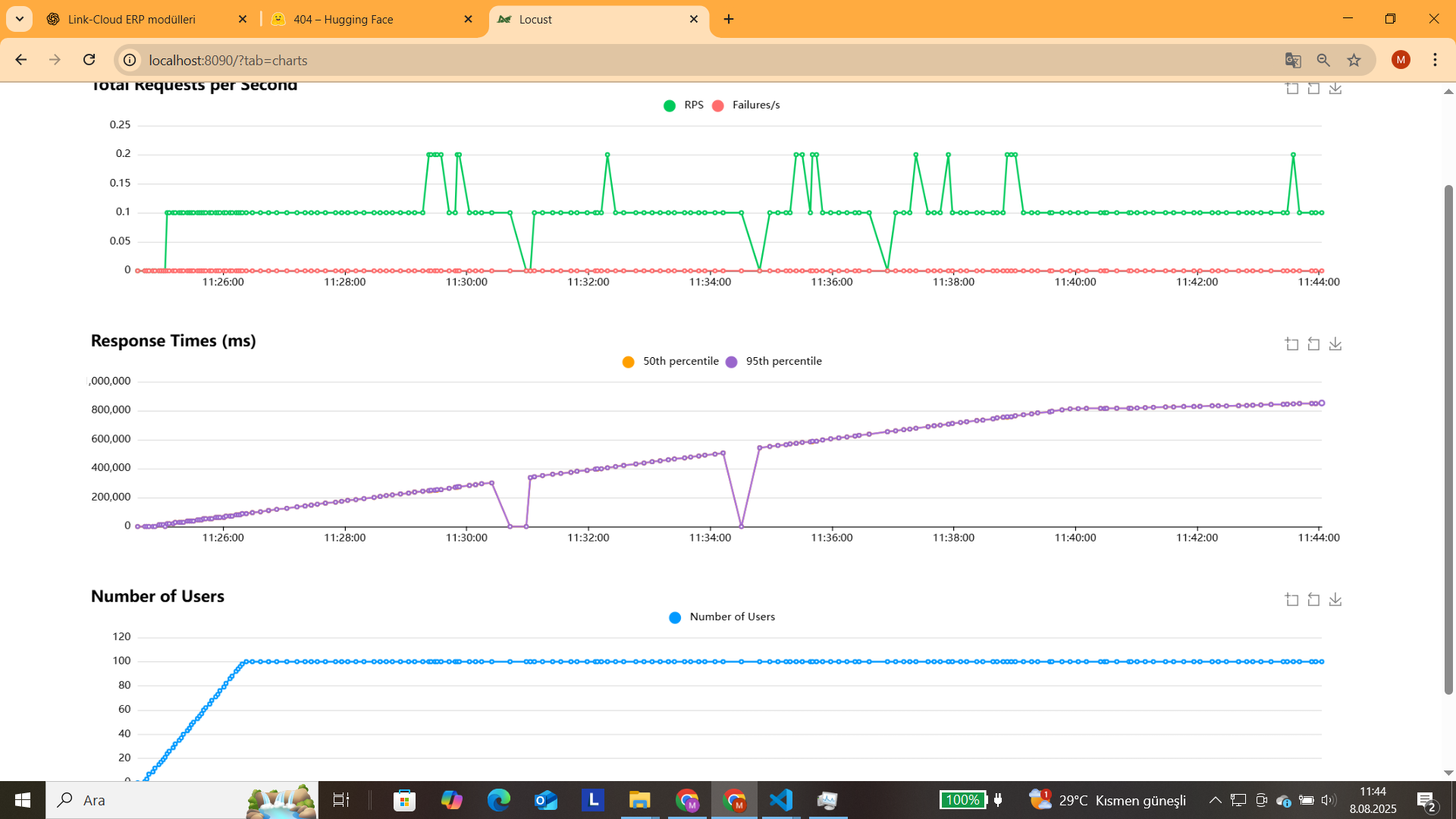
# Locust Performans Testi Raporu (gguf’u doğrudan c++ tools ları ile çalıştırma )

## 1. Kullanılan Yapı ve Teknoloji

Bu performans testi, Flask ile oluşturulan bir API üzerinde çalıştırılmıştır. API, `llama-cpp-python` kütüphanesi üzerinden GGUF formatında quantize edilmiş bir LLaMA modeli kullanarak cevaplar üretmektedir. Model, 8 thread kullanılarak CPU üzerinde çalışmaktadır. Performans testi ise Locust aracı kullanılarak gerçekleştirilmiştir. Locust, aynı anda çok sayıda kullanıcı simüle ederek sistemin yanıt süresini, istek başına düşen yanıt sayısını ve sistem hatalarını ölçmek için tercih edilmiştir.  
  
Kodlar incelendiğinde:  
- `Flask` sunucusu `llama\_cpp` ile entegre çalışmaktadır.  
- Model GGUF formatında quantize edilmiştir (Q4\_0).  
- Model `n\_ctx=32768`, `n\_batch=256` parametreleri ile başlatılmıştır.  
- 8 thread kullanılmıştır (OMP ve GGML için ayarlanmıştır).  
- Locust ile yapılan testte 100 kullanıcı eş zamanlı çalıştırılmıştır.

## 2. Test Sonuçlarının Analizi

Aşağıda Locust arayüzünden alınan performans testi ekran görüntüsü yer almaktadır. Bu grafik, sistemin farklı boyutlardaki metriklerine dair bilgi sunmaktadır:



## 3. Grafik Analizi

Grafikler üç farklı metriği göstermektedir:  
  
\*\*Toplam İstek / Saniye (Total Requests per Second):\*\*  
- Ortalama istek sayısı saniyede 0.05 ila 0.2 arasında dalgalanmıştır.  
- Bu oran, sabit kullanıcı yüküne rağmen modelin stabil şekilde çalıştığını göstermektedir.  
- Kırmızı çizgi (Failures/s) neredeyse sıfırdır, bu da sistemin hata vermediğini gösterir.  
  
 \*\*Yanıt Süreleri (Response Times):\*\*  
- Yanıt süreleri (95. percentile) zamanla artış göstermiştir.  
- Başlangıçta ~100.000ms civarında olan yanıt süresi, zamanla ~1.000.000ms seviyesine yaklaşmıştır.  
- Bu durum sistemin CPU sınırına yaklaşmasıyla oluşan kuyruklardan kaynaklı gecikmeleri göstermektedir.  
  
 \*\*Kullanıcı Sayısı (Number of Users):\*\*  
- Kullanıcı sayısı hızlıca 100’e ulaşmış ve test boyunca sabit kalmıştır.  
- Bu durum testin eş zamanlı yük altında gerçekleştirildiğini gösterir.  
  
Genel olarak sistemin hata vermediği ancak zamanla tepki süresinde ciddi artış yaşandığı görülmektedir.

## 4. Sonuç ve Öneriler

Bu test sonucuna göre:  
- Flask + LLaMA GGUF model mimarisi 100 eş zamanlı kullanıcıyı işlemede başarılıdır.  
- Ancak, model CPU üzerinde çalıştığı için tepki süresi artmaktadır.  
- 8 thread kullanımı makul olmakla birlikte, daha yüksek verim için aşağıdaki adımlar önerilir:  
 • Modelin GPU ile çalıştırılması.  
 • Thread sayısının çekirdek sayısına göre optimize edilmesi.  
 • Flask yerine üretim ortamına uygun bir WSGI sunucusu (örneğin `gunicorn` veya `waitress`) kullanılması.  
- Ayrıca modelin cevap süresini iyileştirmek için prompt boyutları ve `n\_batch` parametresi optimize edilmelidir.