Merhaba,

7 yıl sonra bloga döndüğüm son yazımda, OpenCode Go aboneliğinde DeepSeek limitimi 13 günde %91 tükettiğimi anlatmıştım.O yazının sonunda bütçem ile tempom arasındaki açığı konuşmuştum. Bu yazı, o açığı kapatmak için denediğim farklı bir yolun hikâyesi: **modeli kendi bilgisayarımda çalıştırmak.** Kısa özet: denedim, bir yere kadar işe yaradı, sonunda vazgeçtim. Nedenlerini rakamlarla anlatayım.

Neden yerel modeli düşündüm?

İki sebebim vardı:

Limit. Her agent her adımda istek atıyor ve bunlar hızla birikiyor. Yerel çalışan bir modelin kotası yok.
Gizlilik. Çalıştığım projenin kodunun dışarı çıkması fikri beni hâlâ düşündürüyor. Önceki yazıda DeepSeek için resmî sayfada veri saklamama ve eğitimde kullanmama taahhüdünü okuduğumu yazmıştım. Ama kod hiç makineden çıkmasa, bu konuyu düşünmem bile gerekmezdi.
Elimde 48 GB belleğe sahip bir MacBook Pro (M4 Pro) vardı. “Neden olmasın?” dedim.

Kurulum: Şaşırtıcı derecede kolay

Burası iyi haber. Ollama’yı masaüstü uygulaması olmadan, sadece komut satırı sürümüyle kurdum. Tüm modelleri ve ayarları tek bir klasörde topladım, böylece yedek almak klasörü kopyalamaktan ibaret oldu. Model olarak Qwen3-Coder 30B’yi seçtim. Diskte yaklaşık 18 GB tutuyor, 32K bağlamla çalıştırınca bellekte yaklaşık 20 GB yer kaplıyor ve tamamı GPU’ya sığıyor. OpenCode’a yerel bir sağlayıcı olarak tanımladım. Kısa bir Java sorusuna anında ve doğru cevap verdi. Bu noktada gayet umutluydum.

Deneme 1: Hepsini yerele çektim, proje yöneticisi takıldı

İlk reflekslerim “hepsini yerele alıp bakayım” oldu. Proje yöneticisi agent’ım tekrar tekrar takıldı. Aynı adımı yineledi, ilerlemedi. Bağlamı 64K’ya çıkarmıştım ve bir noktada bilgisayarım donmaya başladı. `ollama ps` hâlâ “%100 GPU” diyordu, yani ilk bakışta her şey normaldi.

Sonradan 32K'ya dönüp bellek ölçümlerine baktım: 48 GB'ın yaklaşık 37,7 GB'ı kullanılıyordu, takas (swap) kullanımı sıfırdı, bellek baskısı grafiği yeşildi. Yani 32K'da sistem rahattı. Donmanın büyük bağlam ve döngünün birleşiminden geldiğini tahmin ediyorum, ama bunu kesin olarak kanıtlayamam. Proje yöneticisini hemen DeepSeek'e geri aldım.

Deneme 2: Test agent'ı 25 dakika döndü durdu

Sıra kalite güvence agent'ındaydı. Proje yöneticisinden ona küçük, salt okunur bir denetim verdirdim: üç dosyaya ve bir veritabanı betiğine bakıp üç net soruya cevap verecekti. Normalde birkaç dakikalık bir iş.

25 dakika sonra hâlâ bitmemişti. Log'a baktım: üç dosyaya neredeyse hiç dokunmamış, tek bir dosyanın etrafında dönüyordu. Kendi kendine görevde hiç yer almayan bir soruya kaymış ("şu çeviri eksik mi?") ve neredeyse aynı cümleleri tekrarlıyordu. Görevde yazdığım istek ve okuma sınırlarını çoktan aşmıştı. Görev gayet net yazılmıştı. Sorun, modelin kuralları aklında tutamamasıydı.

Deneme 3: Ön yüz agent'ı, asıl şansı buydu

Yerel modelin en çok işe yarayacağını düşündüğüm yer burasıydı: kod yazmak. Sonucu kendi notumla paylaşıyorum:

> Model 40-60 satırlık, tek konulu, spesifikasyonu kod örneğiyle verilmiş paketlerde güvenilir. Zor/çok konulu refactor'da (370 satır, 3 dosya, en az 8 bağımlılık) bağlamı taşıyamıyor ve doğrulanmamış "DONE" veriyor.

Bu son cümle beni en çok düşündüren kısım oldu. Çalışmayan kod bir sorun, ama "bitti" diyen ve hiç doğrulamamış bir agent daha tehlikeli: hatayı sessizce bana devrediyor. Bunu yakalamak için her çıktıyı tek tek kontrol etmem gerekir ki bu, limit tasarrufu için kullandığım zamandan çok daha pahalıya gelir.

Ne öğrendim?

Yerel model dar ve net işlerde çalışıyor. Küçük, tek konulu ve örnekle tarif edilmiş görevlerde güvenilir.

Koordinasyon ve hüküm verme zayıf kalıyor. Uzun bağlamı taşımak, kuralları hatırlamak ve “gerçekten kontrol ettim” demek benim denemelerimde en çok orada koptu.
Agent ekibinde en zayıf halka belirleyici. Her rol için ayrı model seçmeden hepsini yerele çekmek, bütün zinciri en zayıf halkanın seviyesine indiriyor.
Bellek dengesi önemli. 48 GB rahat görünüyor ama bağlam büyüdükçe model ve sistem birbirini sıkıştırıyor.
Zaman da bir maliyet. Limitten kurtulmak için harcadığım saatleri bir dolar karşılığı olarak hesaplasam, tasarruf rakamı pek kâr çıkarmıyor.

Bu bir “yerel model kötü” yazısı değil

Bu yazı bir yargı değil, kendi deneyimimin kaydı: tek bir makine (48 GB), tek bir model (30B sınıfı), tek bir iş yükü (çok agent’lı bir yazılım akışı). Daha büyük bellek, daha güçlü bir model ya da daha dar bir iş yükü farklı sonuç verebilir. Küçük, yalıtılmış ve gizli tutmak istediğiniz görevlerde yerel model hâlâ mantıklı bir seçenek olabilir.

Peki şimdi ne yapıyorum?

Yerel modelleri iş akışımdan çıkardım ve ilk yazıdaki yola döndüm: kullanımı kısmak. Gizlilik tarafında ise hassas dosyaları agent’ların erişiminden dışarıda tutmak ve bulut agent’lara ham kod yerine görev özeti vermek gibi daha basit önlemleri öne çıkaracağım.Ücretli modeller ve abonelik seçenekleri üzerindeki araştırmalarım hâlâ devam ediyor. Command Code dahil birkaç alternatifi karşılaştırıyorum. Birinde karar kılıp o yolla devam edeceğim, kararımı deneyimimi bir sonraki yazıda paylaşacağım.

Sizden gelen yorumları merak ediyorum

Yerel modelle agent kurgusu deneyenler oldu mu? Hangi model ve donanımla?
Yerel modelin işe yaradığı rol sizde hangisi: kod yazmak mı, test mi, yoksa başka bir şey mi?
Gizlilik ile maliyet arasında nasıl bir denge kuruyorsunuz?
Büyük bellekli bir Mac’te (64 GB ve üzeri) farklı sonuç aldınız mı?