

Go'da concurrency

İlk bilgisayarlar için programlama dilleri geliştirildiğinde bilgisayarlar tek işlemcili (çekirdekli) bilgisayarlar için geliştirildi.

Zaman geçtikler tek işlemcinin yetersiz kalmaya başladı ve büyük ve kompleks problemleri çözmek için Cloud Sistemleri ortaya çıktı. Bu Sistemler birden fazla bilgisayarı bağlayarak işlemci güçlerini aynı zamanda kullanmak için geliştirildiler ama programlama dillerinin tek işlemcili yapısı buna engel oluyordu.

Bu sorunu aşmak için Concurrency kavramı geliştirildi.

Bilgisayarlar hala tek çekirdek çalışırken geliştirelen programa dillerine Library , framework yada methodology olarak eklenmeye başlandı.

Go bu durumdan farklı olarak özellikle 2010 sonrası çok çekirdekli bilgisayarların yaygınlaşmasının etkisiyle tamamen temelden concurrency yeteneği ile birlikte geliştirildi.

Concurrency nedir ? Theaded, Paralel, Asynchronous ?

Concurrency aynı anda birçok işin bir seferde yapılması anlamına gelir.

Örnek olarak 2 mektubu tek kalemle yazmak verilebilir.Paragrafı sağ elle yazdıktan sonra sol ölle yazarsın. Bu şekilde sürekli eline değiştirerek mektupları yazmayı bitirebilirsin.

Birçok işi kaynakları ve zamanı bölüşerek yapmak..

Paralellism ise işin aynı anda başlanıp aynı anda bitirilmesi anlamına gelir.

Mektup örneğinde en iyi karşılığı iki elde iki kalemle 2 mektup yazmak verilebilir. İki elde eş zamanlı 2 mektubu yazar.

Birçok işi aynı anda yapma...

Concurrency programcı tarafından kodlara eklenebilecek bir yapıdır.

Paralellism bilgisayarlar tarafından karar verilir.

Neden Önemli

Sequential programlama ilk öğrenilen ve çoğu zaman algoritma kurulurken tercih edilen ve bilgisayarların çalışma sistemine yakın yöntemdir.

Ama concurrency sayesinde bu işlemleri daha verimli yapabiliriz.

Örnek olarak graph search algoritmda algoritma graphdaki tek noktadan başlayıp son node(düğüm)e kadar gider.

Eğer birçok çekirdek işlemde kullanılırsa her çekirdek ayrı yolu kullanarak çekirdek sayısı kadar hıza bölümde bitirebilir.

Tamam ama Neden Go ?

Geleneksel olarak concurrency için programlama dilleri threadleri kullanır ve librarylerle bu dillere eklenir.

Bu tip programlama dillerinde yapmamız gereken aşamalar:

- 1) Program threadleri nasıl uygulayacak
- 2) Threadleri uygulayan kodları imlemante etmek
- 3) İstenen çıktıyı alamadığı fark etmek Nedeni deadlock, geçersiz memory access, senkronize operasyon olmaması ...
- 4) Sorunu bulmak , çözmek tekrar kontrol etmek
- 5) 3 ve 4 tekrar
- 6) Program açılanamamak kadar kompleks hale geldi

En önemlisi programlama dillerindeki Thread/Concurrency operasyonları birkaç faktöre bağlı

- 1) Dil nasıl OS thread hallediyor
- 2) Treadlerin stackleri ne kadar büyüklükte(yığınları)
- 3) Thread Pool(havuzunu) hallleden library(kütüphane) ne kadar verimli

4) Operasyon sırasında kaç çekirdek kullanılabilecek

Go nun sağladığı avantajlar

- 1) Kasmayan hafif çalışan threadler
- 2) OS thread l yok. Sanal threadler multiplexed(dağıtılıyor?) ediliyor OS threadleri üzerine
- 3) Birbirleriyle etkileşimde olan Channellar
- 4) Kolay ve temiz uygulanma
- 5) Go runtime verimlili sağlayacak şekilde çalışması

GO Concurrency Rehberi

Go da concurrency 3 basit konseptten oluşur

- 1)Go Routines
- 2)Blocking Operations
- 3)Channels(Bufferend yada UnBuffered)