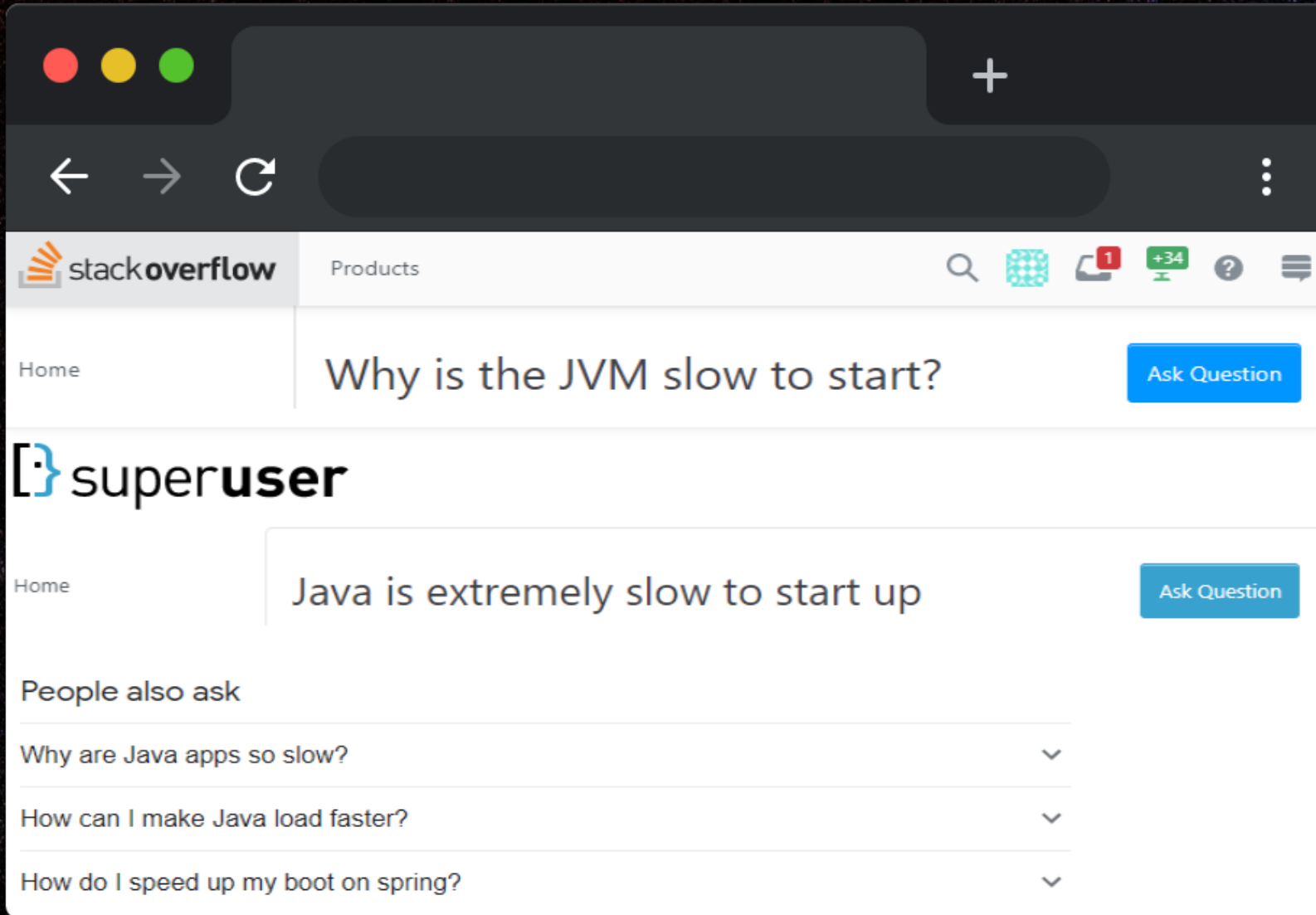


Hakkında

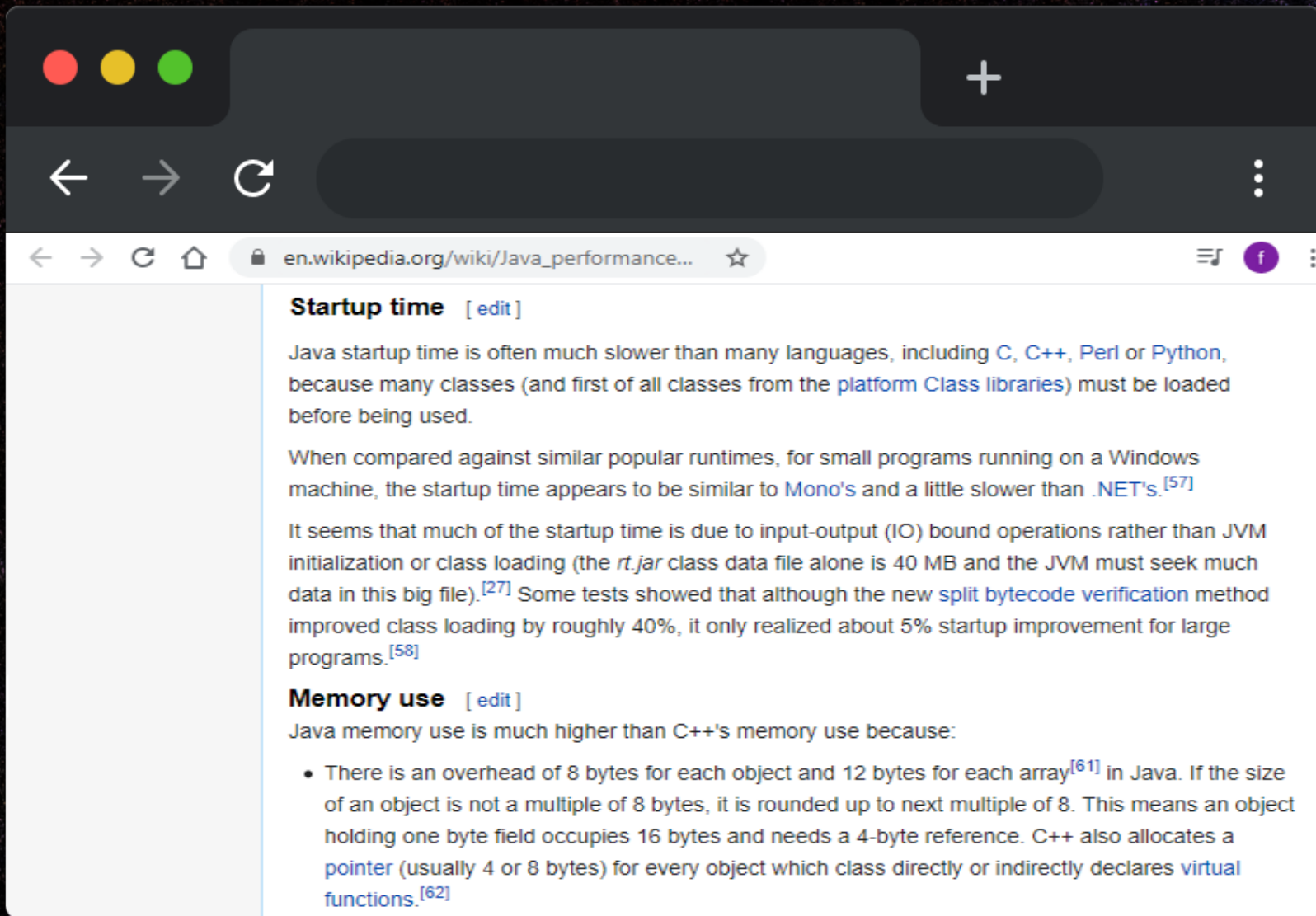
GraalVM™

adesso

Soru şuydu



Ne yapabiliriz ?

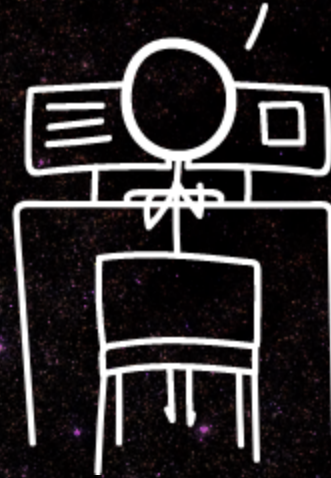


Çözüm Nedir?

Wait.
So you just save it,
And your code is running?
And it's Java?!



I know, right?
SUPERSONIC JAVA, FTW!



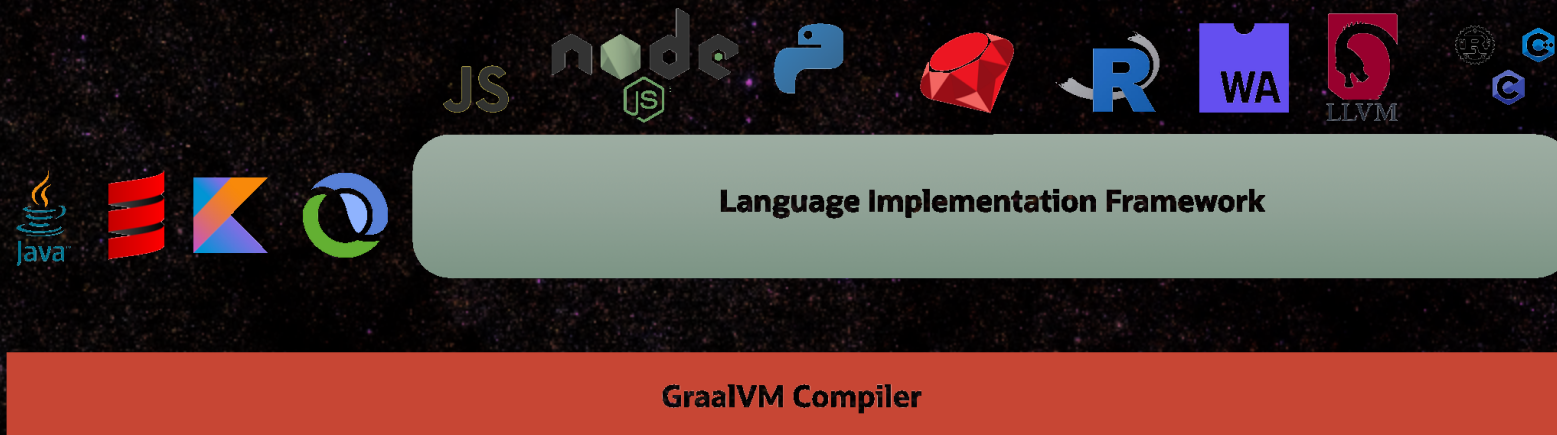
GraalVM Nedir ?

GraalVM, Java ve diğer JVM dillerinde yazılmış uygulamaların yürütülmesini hızlandırmak için tasarlanmış yüksek performanslı bir JDK dağıtımıdır ve JavaScript, Ruby, Python ve *bir dizi başka popüler dil*. GraalVM in çok dilli yetenekleri, *yabancı dil arama maliyetlerini ortadan kaldırırken* birden çok programlama dilini tek bir uygulamada karıştırmayı mümkün kılar.

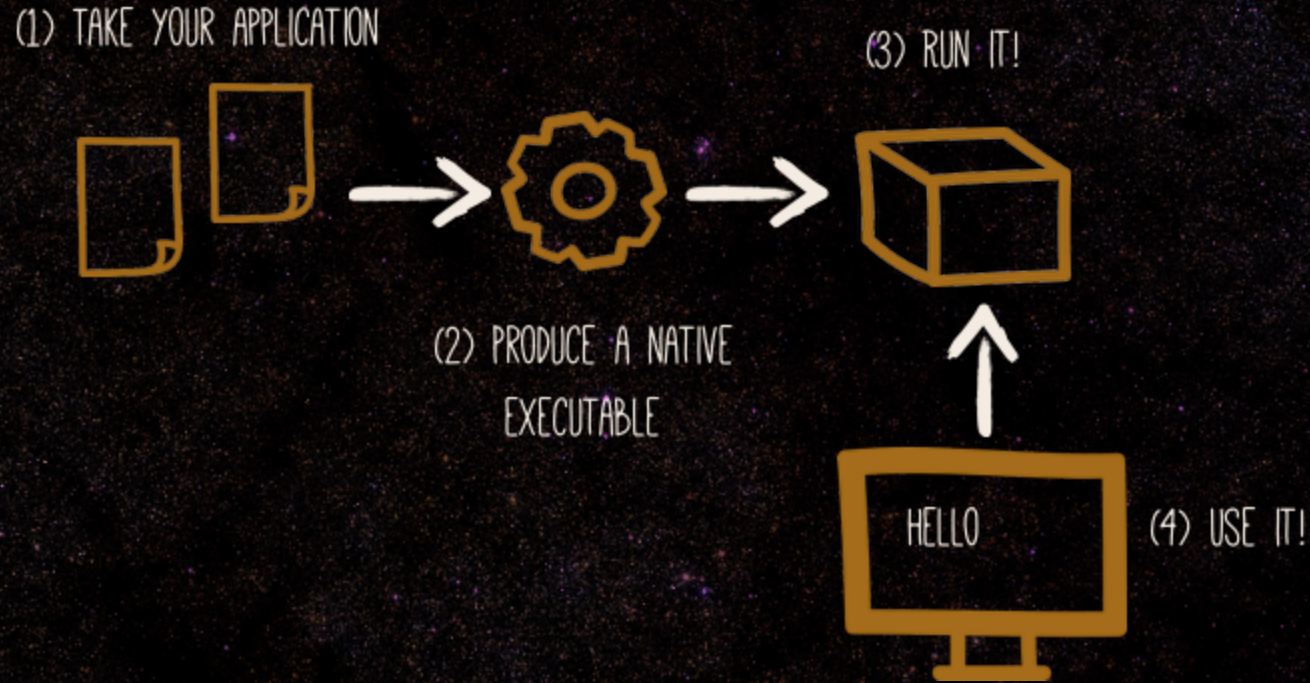
GraalVM bana ne verir ?

- Tek bir uygulamada birden fazla programlama dilini karıştırın
- Yerel yürütülebilir dosyalar
- **İnanılmaz** hızlı önyükleme süresi
- İnanılmaz derecede **düşük RSS belleği** (yalnızca yığın boyutu değil!)
- **Kubernetes** gibi kapsayıcı düzenleme platformlarında anında (nispeten) ölçek büyütme ve yüksek yoğunluklu bellek kullanımı.

GraalVM Mimarisi



Yerel Yürütülebilir Dosyalar



Uygulamamız için yerel yürütülebilir dosya; uygulama kodu, gerekli kitaplıklar, Java API'ları ve bir sanal makinenin küçültülmüş sürümü nü içerecektir. Daha küçük VM tabanı, uygulamanın başlangıç zamanını iyileştirir ve minimum disk ayak izi üretir.

Truffle dil uygulama çerçevesi

Truffle dili uygulama çerçevesi (bundan böyle "Truffle"), kendi kendini değiştiren Soyut Sözdizimi Ağaçları için yorumlayıcılar olarak araçlar ve programlama dilleri uygulamaları oluşturmak için açık kaynaklı bir kitaplıktır. Açık kaynak GraalVM derleyicisi ile birlikte,

Truffle, mevcut dinamik diller çağında programlama dili uygulama teknolojisinde önemli bir adımı temsil ediyor.

Desteklenen Çerçeveler ve Teknoloji

- Spring Framework (Experimental)
- Quarkus
- Play Framework
- Camel
- Prometheus
- JavaFX

...

Dezavantajları

- Bu yaklaşımın ana dezavantajı, platforma bağlı yerel koddur.

Bu, linux/windows vb. için kaynak kodunu derlemeniz gerektiği anlamına gelir.

Yani ??

SystemAdmin01: server_api.so sunucumda çalışmıyor !

Customer\$\$\$__: customer_api.exe bilgisayarımda çalışmıyor!

DevGuy42_____: api.jar iş istasyonumda çalışıyor!

Fırat_____: Bazı durumlarda kullanılan kitaplıklardan veya derleme komut dosyalarından sorunlar çıkabilir. Yani **testlere** dikkat etmek gerekiyor.

SystemAdmin01: hımm, server_api.so benim sunucumda çalışmıyor !

.....

DevGuy42 yazıyor ...

Örnek Uygulama

Oracle indirmelerinden GraalVM'i indirin ve kurun. Bir `Example.java` dosyası oluşturun.

```
public class Example {  
  
    public static void main(String[] args) {  
        System.out.println("Merhaba GraalVM");  
    }  
}
```

```
#Bu komutları çağırın :  
javac Example.java  
native-image Example
```


Testler : Tabiki

```
package org.acme.quickstart;

import io.quarkus.test.junit.NativeImageTest;

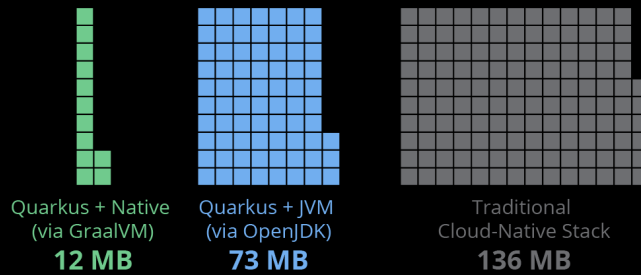
@NativeImageTest
public class NativeGreetingResourceIT extends GreetingResourceTest {
    // Run the same tests
}
```


Performans

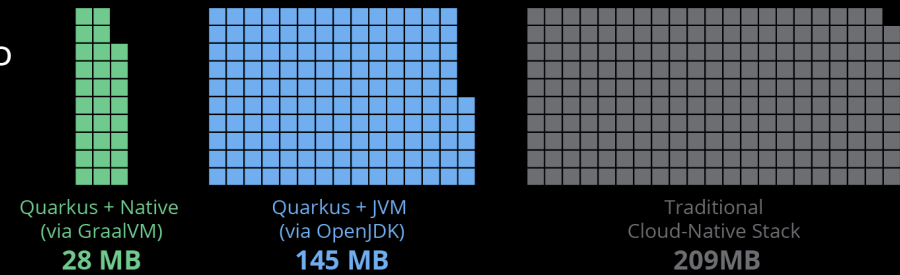
Memory (RSS) in Megabytes*

*Tested on a single-core machine

REST



REST + CRUD



BOOT + First Response Time

REST



REST + CRUD



Quarkus Testleri

Bu deneyimden sonra ihtiyacım için bir demo uygulaması yaptım.
Başlangıç ve bellek ayak izini gerçekten önemli ölçüde azaltıyor. :)



Destek

GraalVM teknolojileri, üretime hazır ve deneysel olarak dağıtılır.

GraalVM'nin gelecekteki sürümleri için deneysel özellikler değerlendirilmektedir ve üretimde kullanılması amaçlanmamıştır.

Geliştirme ekibi, deneysel özelliklerle ilgili geri bildirimleri memnuniyetle karşılar, ancak kullanıcılar, deneysel özelliklerin hiçbir zaman son sürüme dahil edilmeyebileceğini veya üretime hazır kabul edilmeden önce önemli ölçüde değişebileceğini bilmelidir.

Teşekkürler

- Oracle
- Quarkus
- Spring
- JavaFX / GluonHQ
- Google and Wiki

