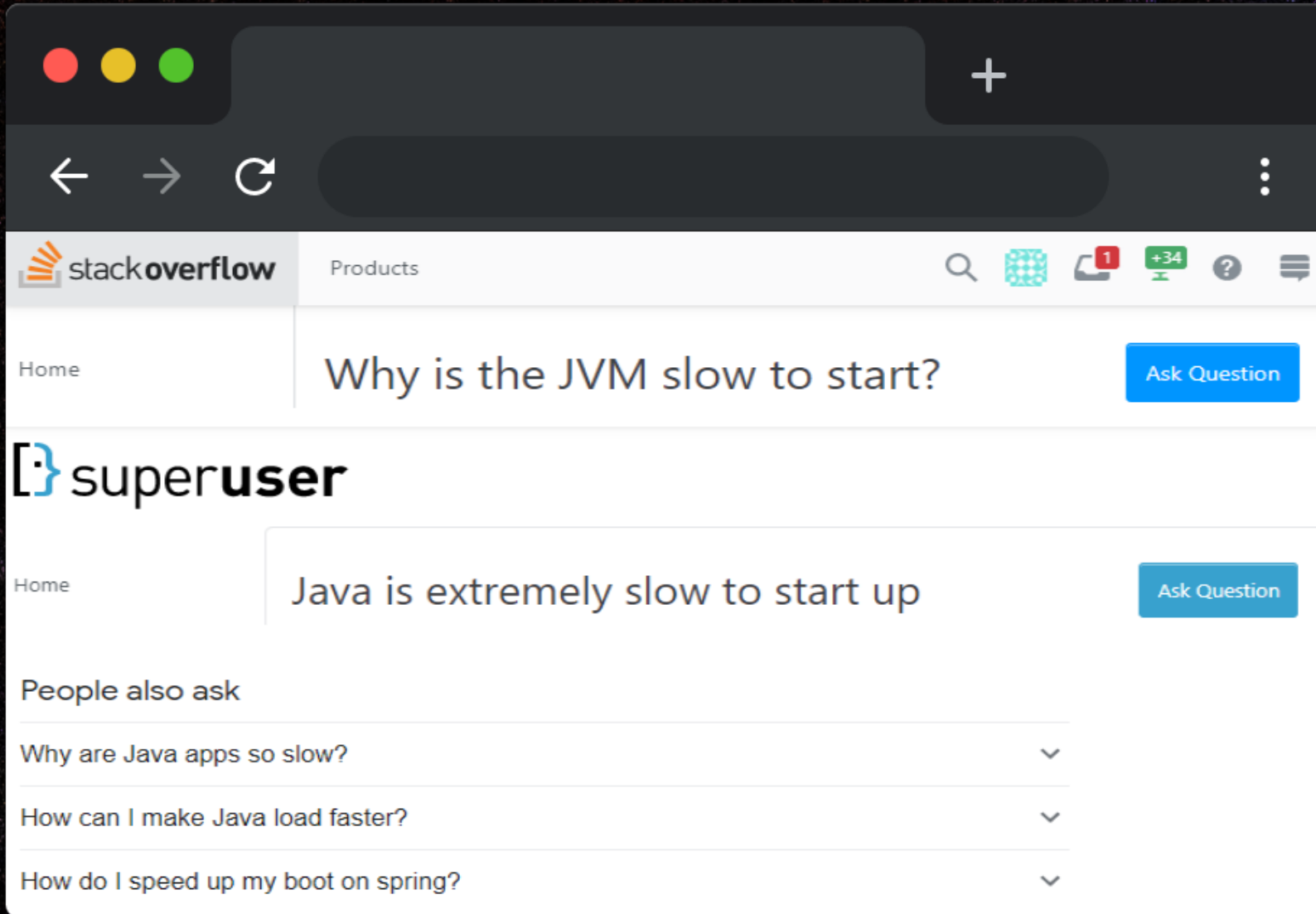


Hakkında

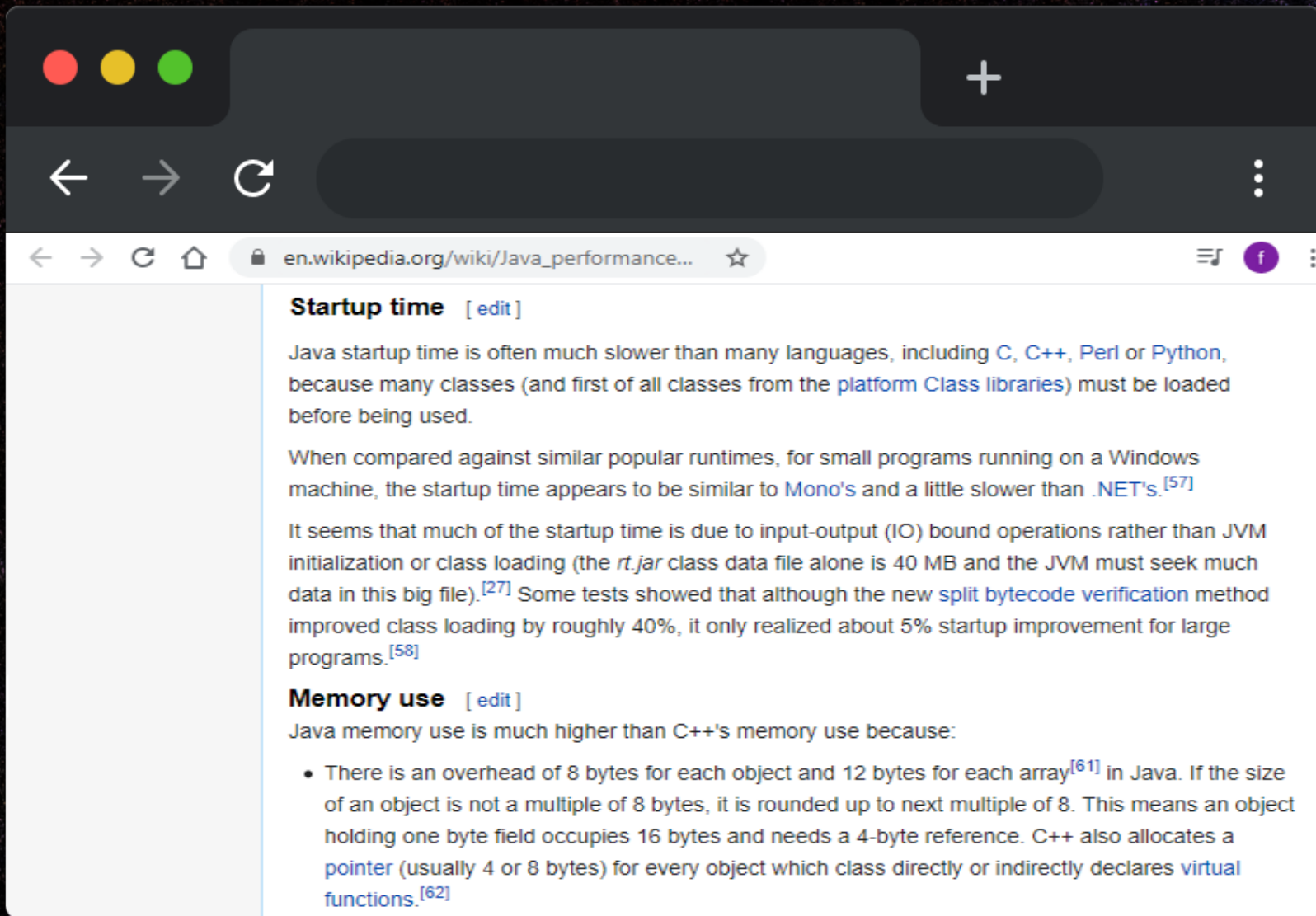
GraalVM™

adesso

Soru şuydu



Ne yapabiliriz ?



Startup time [\[edit\]](#)

Java startup time is often much slower than many languages, including [C](#), [C++](#), [Perl](#) or [Python](#), because many classes (and first of all classes from the [platform Class libraries](#)) must be loaded before being used.

When compared against similar popular runtimes, for small programs running on a Windows machine, the startup time appears to be similar to [Mono's](#) and a little slower than [.NET's](#).^[57]

It seems that much of the startup time is due to input-output (IO) bound operations rather than JVM initialization or class loading (the *rt.jar* class data file alone is 40 MB and the JVM must seek much data in this big file).^[27] Some tests showed that although the new [split bytecode verification](#) method improved class loading by roughly 40%, it only realized about 5% startup improvement for large programs.^[58]

Memory use [\[edit\]](#)

Java memory use is much higher than C++'s memory use because:

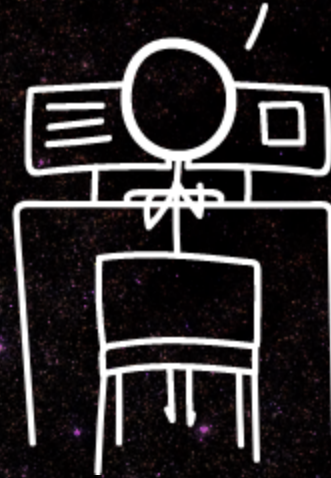
- There is an overhead of 8 bytes for each object and 12 bytes for each array^[61] in Java. If the size of an object is not a multiple of 8 bytes, it is rounded up to next multiple of 8. This means an object holding one byte field occupies 16 bytes and needs a 4-byte reference. C++ also allocates a [pointer](#) (usually 4 or 8 bytes) for every object which class directly or indirectly declares [virtual functions](#).^[62]

Çözüm Nedir?

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



I know, right?
SUPERSONIC JAVA, FTW!



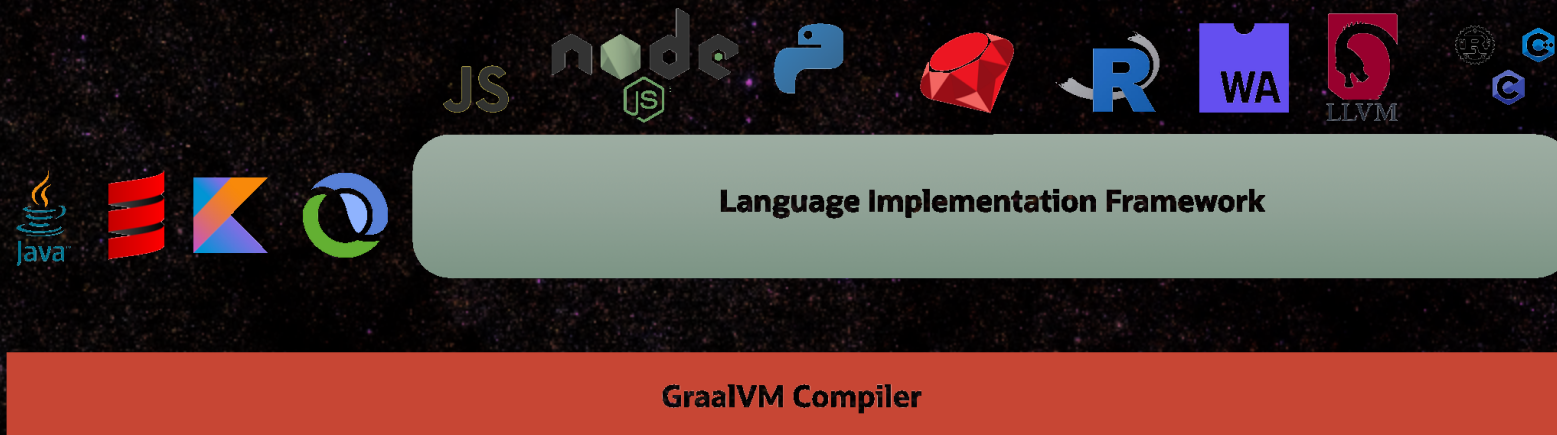
GraalVM Nedir ?

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.

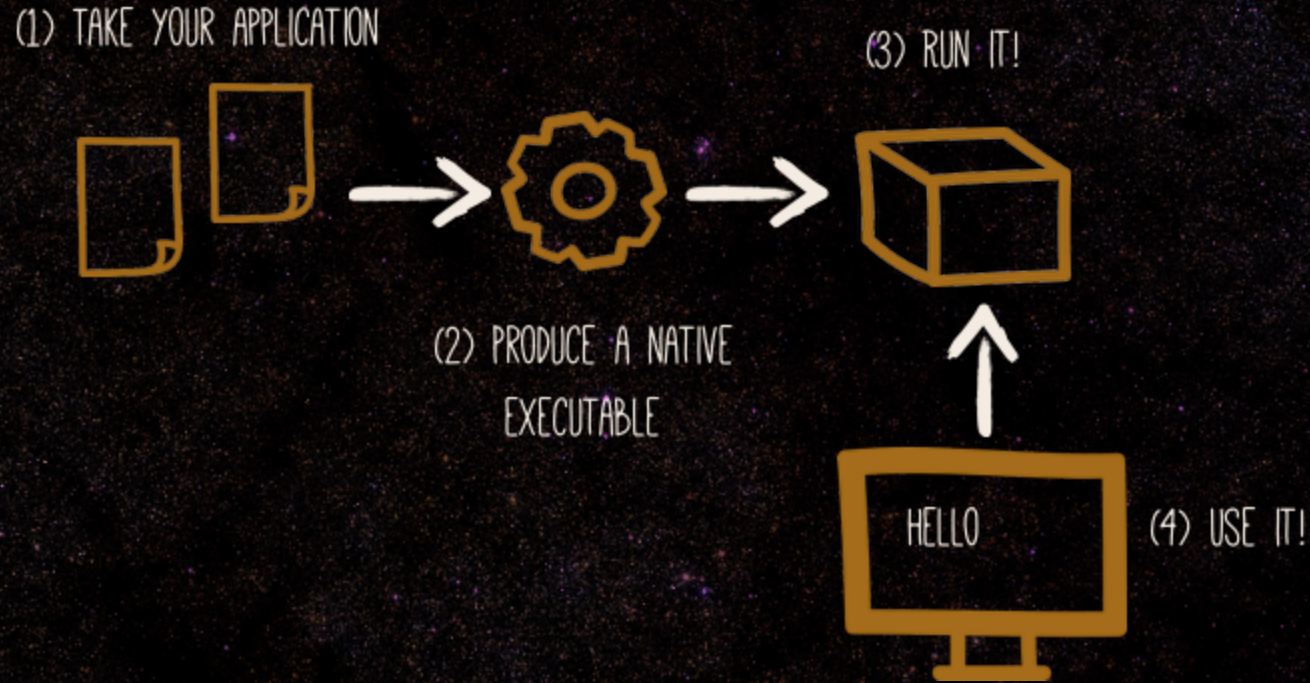
GraalVM bana ne verir ?

- 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 ...

Testler : Tabiki

```
package org.acme.quickstart;

import io.quarkus.test.junit.NativeImageTest;

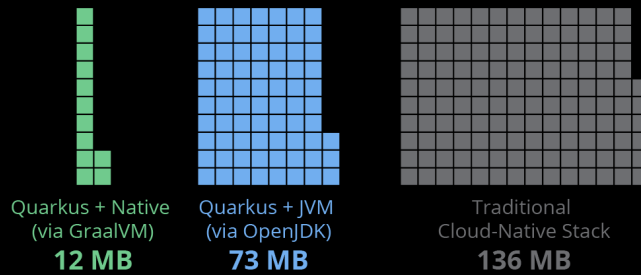
@NativeImageTest
public class NativeGreetingResourceIT extends GreetingResourceTest {
    // Run the same tests
}
```


Performans

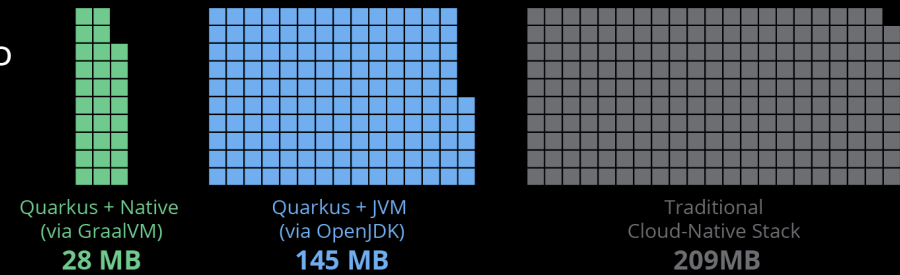
Memory (RSS) in Megabytes*

*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. :)



Teşekkürler

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

