**VERT.X PROJECTS**

After a very thorough research into what sort of tools or toolkit or platform for some API I have been planning for some time, vertx came up as the final and possible immediate solution or option for the level of performance I was aiming for.

In my research I used the following websites for my Benchmarking checks and confirmations :

|  |
| --- |
| **Other Benchmarking websites :** |
| https://dev.to/tuananhpham/popular-backend-frameworks-performance-benchmark-1bkh |
| <https://www.tiobe.com/tiobe-index/> |
| https://pypl.github.io/PYPL.html#google\_vignette |
| https://trends.builtwith.com/framework |
| https://benchmarksgame-team.pages.debian.net/benchmarksgame/index.html |
| https://programming-language-benchmarks.vercel.app/ |
| https://programming-language-benchmarks.vercel.app/java |
| https://programming-language-benchmarks.vercel.app/java-vs-go |
| <https://www.quora.com/What-are-the-best-alternatives-to-Java-for-high-performance-backend-development> |
| https://news.ycombinator.com/item?id=17254152 |
| https://github.com/rwf2/Rocket/issues/710 |
| https://just.billywhizz.io/blog/on-javascript-performance-01/ |
| [https://www.techempower.com](https://www.techempower.com/) |

Although, one could argue that the way benchmarking is done or executed, it might not be the same as what you might experience in a production environment. And for that I say : If the framework or platform or tool or toolkit or language of choice performed at this level for a very simple hello world app or sample project or in applying a. very simple hello world json structure, then , that is just it. All the other technologies are been benchmarked with same. You better believe it is just what it is. If you did pull the sample technologies and tested them your results would not be too far away nor apart from what has already been done.  
  
Below is a composite framework scoring ( source : techempower.com )

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **­­­** | | | |  | **Source** : *techempower.com* | | | | |  |  |  |  |
| *Each framework's peak performance in each test type (shown in the colored columns below) is multiplied by the weights shown above. The results are then summed to yield a weighted score. Only frameworks that implement all test types are included. 159 total frameworks ranked, 10 visible, 149 hidden by filters. See filter panel above.* | | | | | | | | | |  | **Hardware :** Citrine Dell R440 Xeon Gold + 10 GbE | | |
| Rnk | Framework | JSON | 1-query | 20-query | Fortunes | Updates | Plaintext | Weighted score | | Min | Max | Average |  |
| 5 | redkale | 1,210,086 | 457,935 | 32,272 | 413,537 | 22,900 | 6,981,831 | **7,096** | 87.80% | 22,900 | 6,981,831 | 1,519,760 |  |
| 7 | vert.x | 1,194,427 | 581,542 | 32,505 | 453,406 | 17,122 | 5,624,745 | **6,847** | 84.70% | 17,122 | 5,624,745 | 1,317,291 |  |
| 11 | jooby | 1,162,248 | 515,795 | 32,196 | 427,682 | 15,889 | 4,801,653 | **6,370** | 78.80% | 15,889 | 4,801,653 | 1,159,244 |  |
| 16 | vertx-web | 1,121,889 | 566,804 | 30,669 | 388,512 | 16,598 | 3,774,206 | **6,107** | 75.50% | 16,598 | 3,774,206 | 983,113 |  |
| 19 | inverno | 1,042,404 | 467,759 | 32,441 | 309,684 | 15,864 | 4,043,496 | **5,566** | 68.80% | 15,864 | 4,043,496 | 985,275 |  |
| 38 | quarkus | 903,185 | 318,897 | 17,610 | 214,275 | 6,697 | 2,861,479 | **3,637** | 45.00% | 6,697 | 2,861,479 | 720,357 |  |
| 50 | actframework | 964,004 | 231,641 | 16,942 | 124,422 | 1,985 | 3,273,101 | **2,911** | 36.00% | 1,985 | 3,273,101 | 768,683 |  |
| 55 | javalin | 512,495 | 211,243 | 16,582 | 161,275 | 10,405 | 897,788 | **2,755** | 34.10% | 10,405 | 897,788 | 301,631 |  |
| 57 | revenj.jvm | 527,667 | 290,147 | 14,330 | 177,853 | 5,206 | 730,158 | **2,543** | 31.40% | 5,206 | 730,158 | 290,894 |  |
| 125 | wicket | 378,043 | 24,624 | 1,426 | 25,882 | 551 | 516,369 | **679** | 8.40% | 551 | 516,369 | 157,816 |  |

So basically after trying out a number of the technologies listed thereof, I settled on vert.x.



Starting a vertx project is as simple as getting your preferred IDE eg. IntelliJ Community Package version or Eclipse or Visual Studio Code and creating a maven or gradle project and ensuring a dependency to and for io.vertx is set in place.  
  
Or you can start from the vertx starter link to generate a downloadable package or project to start from after successfully importing into the/an IDE as a project.

You can use the Vert.x starter web application at https://start.vertx.io and generate a project skeleton to download



After successfully importing the project,

You can then run the following command :

$ mvn clean install

You can expect some <Build Successful> kind of message at the end to be sure that everything worked out perfectly.

To install the application and the respective dependencies.

**Caution :**

I tried the following commands and it did not work out as expected :

$ vertx run com.arieceive.firstvertxapp.starter.MainMerticle

And the application did not run but rather I had the following error :

Unrecognized VM option 'UseBiasedLocking'

Error: Could not create the Java Virtual Machine.

Error: A fatal exception has occurred. Program will exit.

I also tried the same command and later this error :

Can not find io.netty.resolver.dns.macos.MacOSDnsServerAddressStreamProvider in the classpath, fallback to system defaults. This may result in incorrect DNS resolutions on MacOS. Check whether you have a dependency on 'io.netty:netty-resolver-dns-native-macos'

Failed in deploying verticle

java.lang.ClassNotFoundException: com.arieceive.firstvertxapp.starter.MainMerticle

at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:581)

at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:178)

at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:527)

at io.vertx.core.impl.JavaVerticleFactory.createVerticle(JavaVerticleFactory.java:41)

at io.vertx.core.impl.VerticleManager.doDeployVerticle(VerticleManager.java:217)

at io.vertx.core.impl.VerticleManager.doDeployVerticle(VerticleManager.java:193)

at io.vertx.core.impl.VerticleManager.doDeployVerticle(VerticleManager.java:180)

at io.vertx.core.impl.VerticleManager.deployVerticle(VerticleManager.java:156)

at io.vertx.core.impl.VertxImpl.deployVerticle(VertxImpl.java:794)

at io.vertx.core.impl.VertxImpl.deployVerticle(VertxImpl.java:800)

at io.vertx.core.impl.launcher.commands.VertxIsolatedDeployer.deploy(VertxIsolatedDeployer.java:42)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.base/java.lang.reflect.Method.invoke(Method.java:566)

at io.vertx.core.impl.launcher.commands.ClasspathHandler.deploy(ClasspathHandler.java:169)

at io.vertx.core.impl.launcher.commands.RunCommand.deploy(RunCommand.java:398)

at io.vertx.core.impl.launcher.commands.RunCommand.run(RunCommand.java:260)

at io.vertx.core.impl.launcher.VertxCommandLauncher.execute(VertxCommandLauncher.java:248)

at io.vertx.core.impl.launcher.VertxCommandLauncher.dispatch(VertxCommandLauncher.java:383)

at io.vertx.core.impl.launcher.VertxCommandLauncher.dispatch(VertxCommandLauncher.java:346)

at io.vertx.core.Launcher.main(Launcher.java:45)

So the next thing was to try running the generated directly.

I tried : ( from the root folder of the project )

$ java -jar target/starter-1.0.0-SNAPSHOT.jar

And I got this feedback :

no main manifest attribute, in target/starter-1.0.0-SNAPSHOT.jar

What worked out find in getting application running :

But at last, after trying this :

$ java -jar target/starter-1.0.0-SNAPSHOT-fat.jar

I was then able to get the application running with a successful test using a web browser.  
  
Nov 04, 2024 2:34:20 PM io.netty.resolver.dns.DnsServerAddressStreamProviders <clinit>

WARNING: Can not find io.netty.resolver.dns.macos.MacOSDnsServerAddressStreamProvider in the classpath, fallback to system defaults. This may result in incorrect DNS resolutions on MacOS. Check whether you have a dependency on 'io.netty:netty-resolver-dns-native-macos'

HTTP server started on port 8888

Nov 04, 2024 2:34:20 PM io.vertx.core.impl.launcher.commands.VertxIsolatedDeployer

INFO: Succeeded in deploying verticle