

1. GraalVM et Graal Native image

1.1. GraalVM

GraalVM est une Machine Virtuelle (VM), Open Source, issue d'un projet de recherche commencé il y a plus de 10 ans chez Oracle Labs (anciennement Sun Labs) .

Cette nouvelle VM est maintenue par une communauté d'acteurs majeurs du net (Oracle, Amazon, Twitter, RedHat notamment avec Quarkus, VMWare pour l'intégration de son framework Spring, ...).

C'est une **nouvelle génération** de VM, **polyglotte**, c'est à dire qu'elle supporte de nombreux langages, même ceux qui ne génèrent pas de bytecode. A terme, elle pourrait remplacer l'actuelle VM HotSpot.

Techniquement :

- La VM **GraalVM** est couplée à un nouveau compilateur, **Graal**, écrit entièrement en Java (ce qui permet une compilation cyclique) :
- Il vise à remplacer le compilateur C2 utilisé pour le **JIT** de la VM **HotSpot** et qui est arrivé en fin de vie car trop complexe à faire évoluer (mélange d'assembleur, C, Java)
- Le compilateur Graal peut aussi faire de la compilation **AOT** (Ahead-Of-Time, à l'avance) aussi appelée compilation anticipée.

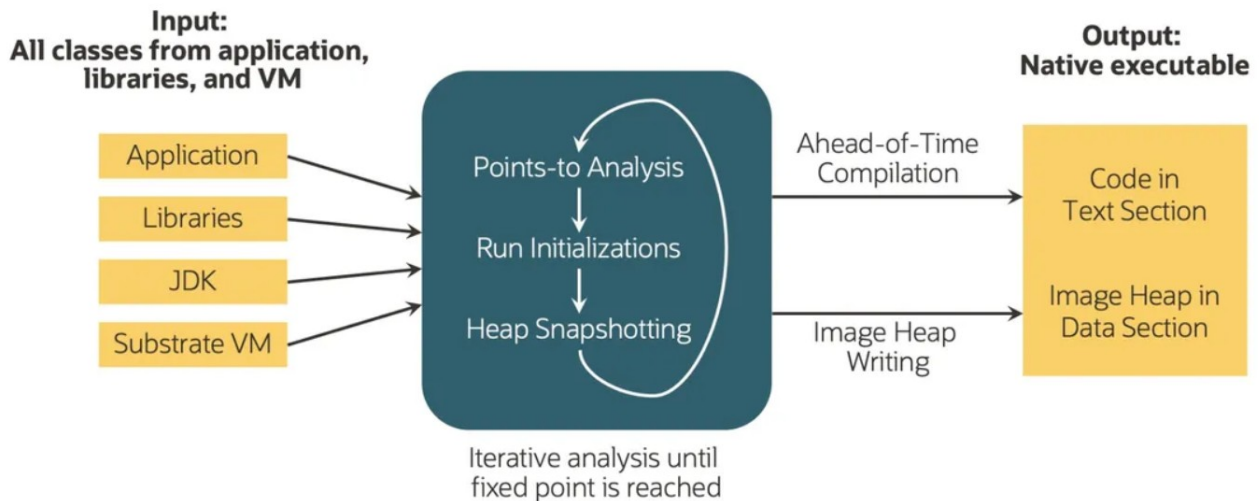
1.2. GraalVM Native Image

- **GraalVM Native Image** est une technologie de compilation ahead-of-time qui **génère des exécutables natifs**.
- **Les exécutables natifs sont idéaux pour les conteneurs et les déploiements cloud car ils sont petits, démarrent très rapidement et nécessitent beaucoup moins de CPU et de mémoire.**
- GraalVM Native Image bénéficie d'une adoption importante avec le support des principaux frameworks Java tels que Spring Boot, Micronaut, Quarkus, Gluegun Substrate, etc.

Dans les grandes lignes , avec une JVM classique , la compilation et l'exécution se produisent en même temps (lors du démarrage) avec une longue période de chauffe .

Avec GraalVM Native Image, en mode AOT, le compilateur effectue toutes les compilations (jusqu'au stade binaire) pendant la construction et avant l'exécution .

L'utilitaire GraalVM 'native-image' prend le bytecode Java en entrée et produit un exécutable natif. Pour ce faire, l'utilitaire effectue une analyse statique du bytecode sous une hypothèse de monde fermé. Lors de l'analyse, l'utilitaire recherche tout le code que votre application utilise réellement et élimine tout ce qui est inutile(un peu comme un bundle javascript généré par webpack et angular).



Pour approfondir le sujet :

<https://scalastic.io/graalvm-microservices-java/>

<https://www.infoq.com/fr/articles/native-java-graalvm/> (source de l'image ci-dessus)

1.3. Spring Native

En 2022, le framework spring est en train d'intégrer "GraalVM Native Image" au sein des dernières versions Spring5/SpringBoot2 et des nouvelles versions Spring6/SpringBoot3 .

Article :

<https://www.infoq.com/fr/articles/native-java-spring-boot/>

<https://docs.spring.io/spring-native/docs/current/reference/htmlsingle/>

1.4. Quarkus (de RedHat)

<https://quarkus.io/>

<https://www.redhat.com/fr/topics/cloud-native-apps/what-is-quarkus>

<https://www.ionos.fr/digitalguide/serveur/configuration/quest-ce-que-quarkus/>