GraalVM, New Technology in Java

Java™ Champions

@javaoracle

NIA 한국지능정보사회진흥원

A long time ago…..

Enterprise is java world!!!

But now….

kubernetes

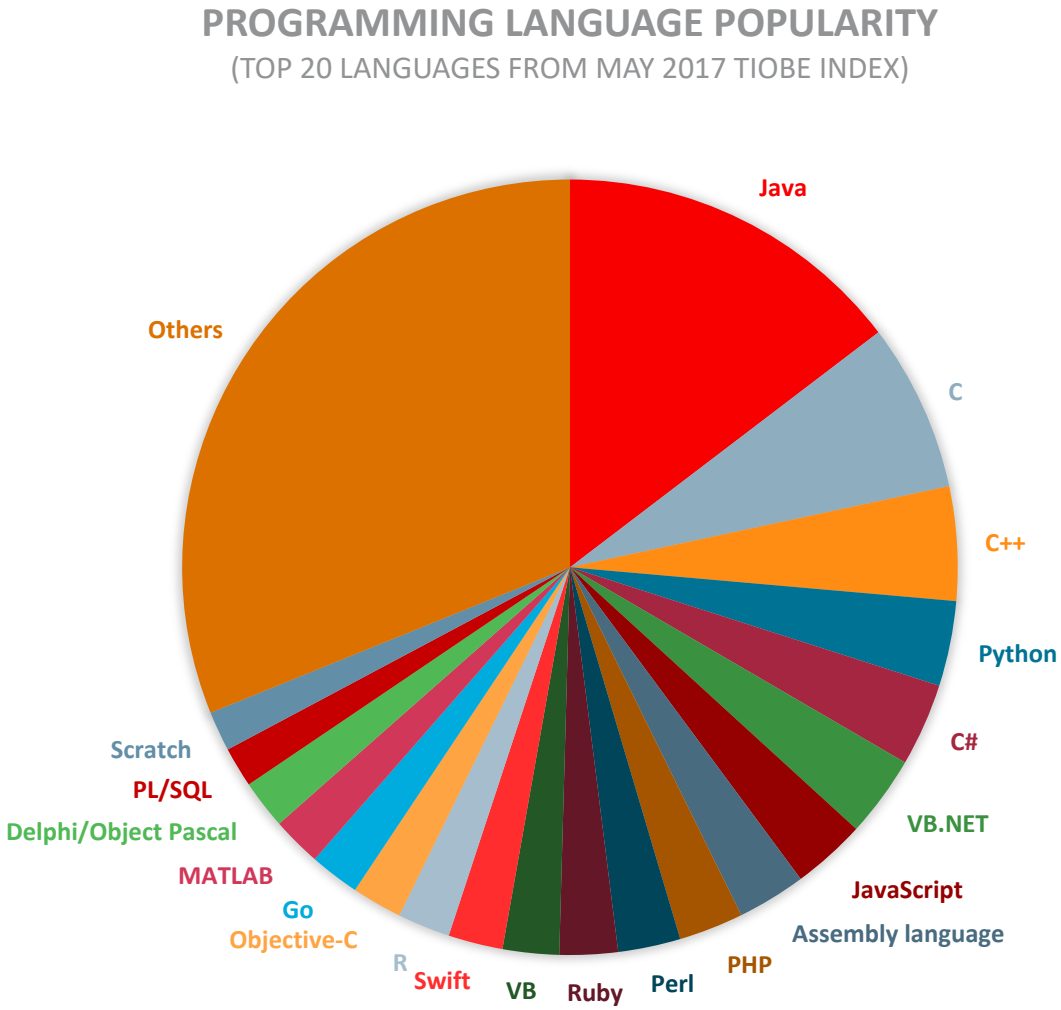
docker

Most Popular Cloud Computing Languages

컴퓨팅 월드는 원래부터 다양한 프로그래밍 언어들이 존재.

Long tail of programming languages

- 몇 개 이상의 앱을 실행하면 몇 개 이상의 언어로 된 어플을 실행.
- 각각의 작업에 맞는 서로 다른 언어를 사용.
- 데이터 사이언스를 위한 R 또는 파이썬
- 프론트엔드에 자바 스크립트
- 백엔드에는 Java 또는 C/C++
- 불행히도 대부분의 런타임은 대부분의 언어를 잘 실행할 수 없음.
- JVM은 JS, 파이썬 또는 네이티브 코드를 잘 실행하지 못함.



GraalVM는 말이죠?

빠르고

- **평균 50% 더 빠름**
  실제 벤치마크에서
- 네이티브 실행의 경우
  빠른 시작

똑똑하고

- **27개의 특허 받은 최적화**
- **네이티브**로 컴파일
- 다중언어 활용

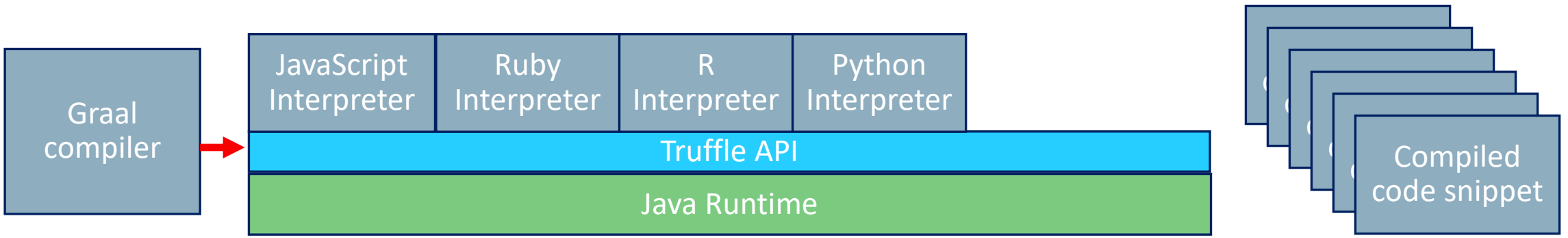
가벼운

- **낮은 메모리**
- **낮은 CPU 사용량**으로 처리량 증가
- **가비지 컬렉션 오버헤드 감소**

OracleGraalVM

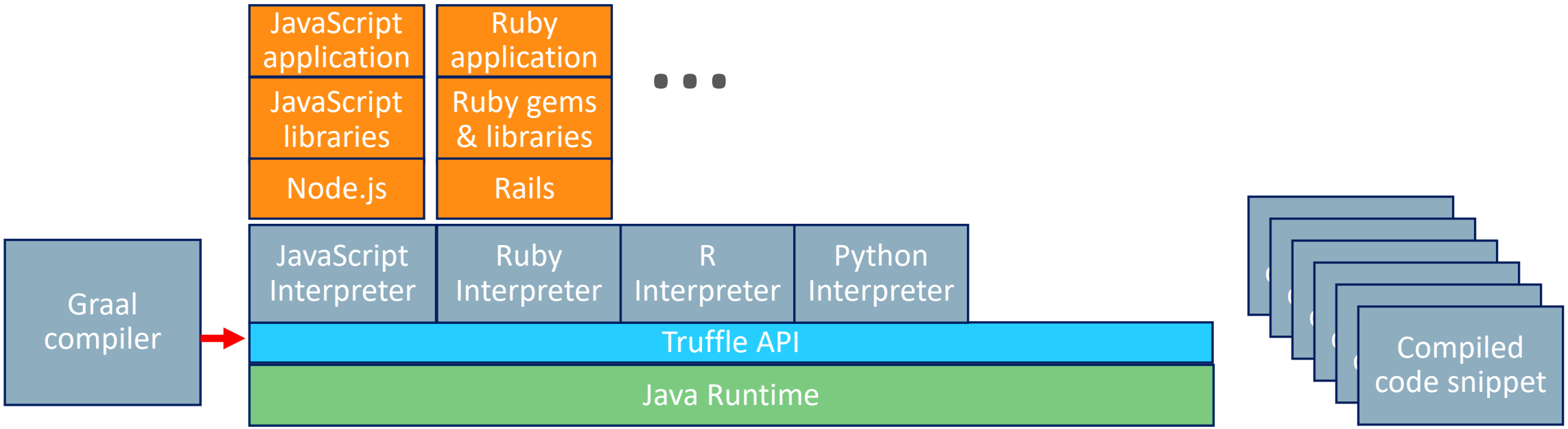
GraalVM은 언어 가상화를 위한 메타 런타임

- GraalVM 언어는 인터프리터를 구성하기 위한 "트러플" API(Java)에 내장.
- 인터프리터는 언어의 의미 체계를 정의하며 프로그램 동작을 프로파일하여 자주 사용하는 코드(예: 루프)의 편집을 안내하는 데도 사용.
- GraalVM은 여러 언어 인터프리터를 포함하고, 해당 인터프리터를 기반으로 효율적인 컴파일러를 생성하기 위해 그들이 하는 일에서 타입 시스템과 언어를 학습합니다.
- 인터프리터가 모두 JVM위에서 구현되기 때문에 단일 엔진으로 병합하면 GraalVM과 단일(매우 복잡한) 언어(언어 교차 비용이 전혀 없는)와 동일.



GraalVM architecture (I)

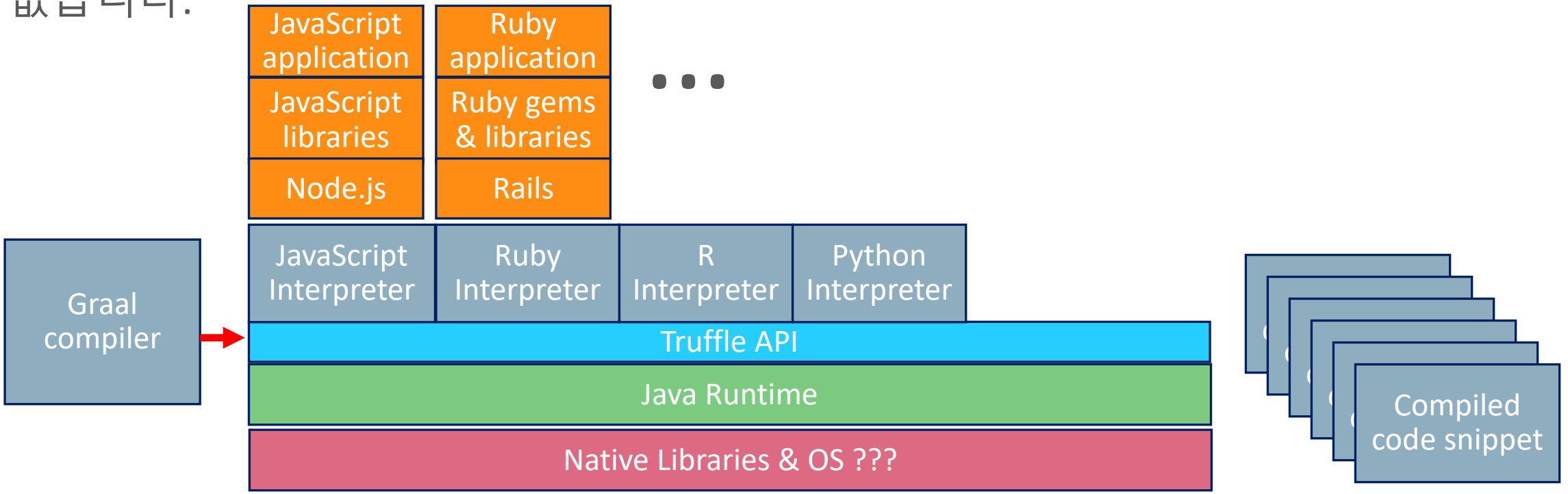
Step one: 인터프리터에서 각 언어의 응용 프로그램, 라이브러리 및 프레임워크 실행



그러나 많은 언어 런타임에는 많은 네이티브 라이브러리가 있기 마련

이들은 core나 시스템 기능의 성능을 높이기 위해 네이티브 코드를 사용

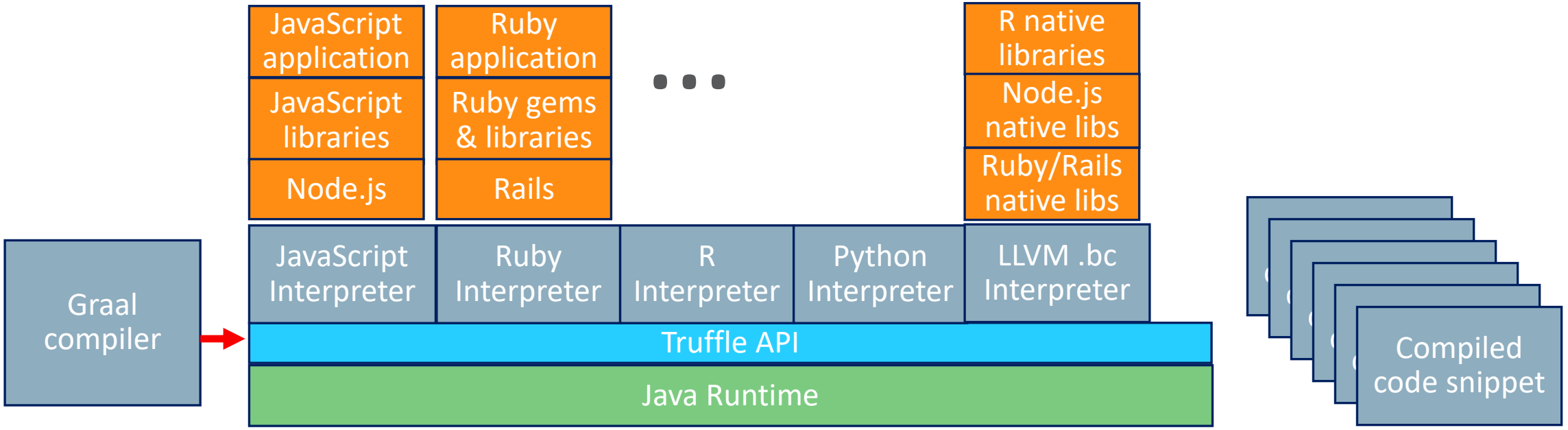
- GraalVM에서 네이티브 코드를 실행하는 경우 가상 컴퓨터에서 원하는 모든 추상화가 중단되지 않습니까?
- 응용 프로그램을 실행하는 안전한 방법입니까? 기본 코드는 안전한 샌드박스에 없습니다.
-



또다른 트러플 인터프리터라는 해결책.

LLVM (“low level virtual machine”) bitcode 인터프리터

- LLVM은 수십개의 정적언어를 지원 (C, C++, FORTRAN, COBOL, Rust, Julia, Swift, Objective-C, …)
- LLVM 컴파일러는 모든 언어에 대해 중간 단계로 "비트 코드"(즉. .bc 파일)를 생성.
- 우리는 LLVM 비트 코드에 대한 트러플 인터프리터를 작성하고 VM 샌드 박스에서 실행



Why is GraalVM Different?

- 다언어 지원
- 쉽게 내장가능.
- 높은 보안수준
- 효율성

Goldman Sachs

" 언어 경계를 넘어 프로파일을 만들 수 있고, 언어 경계를 넘어 디버깅할 수 있으며, 이는 엄청난 효율성의 승리입니다. "

- Zach Fernandez
- Goldman Sachs

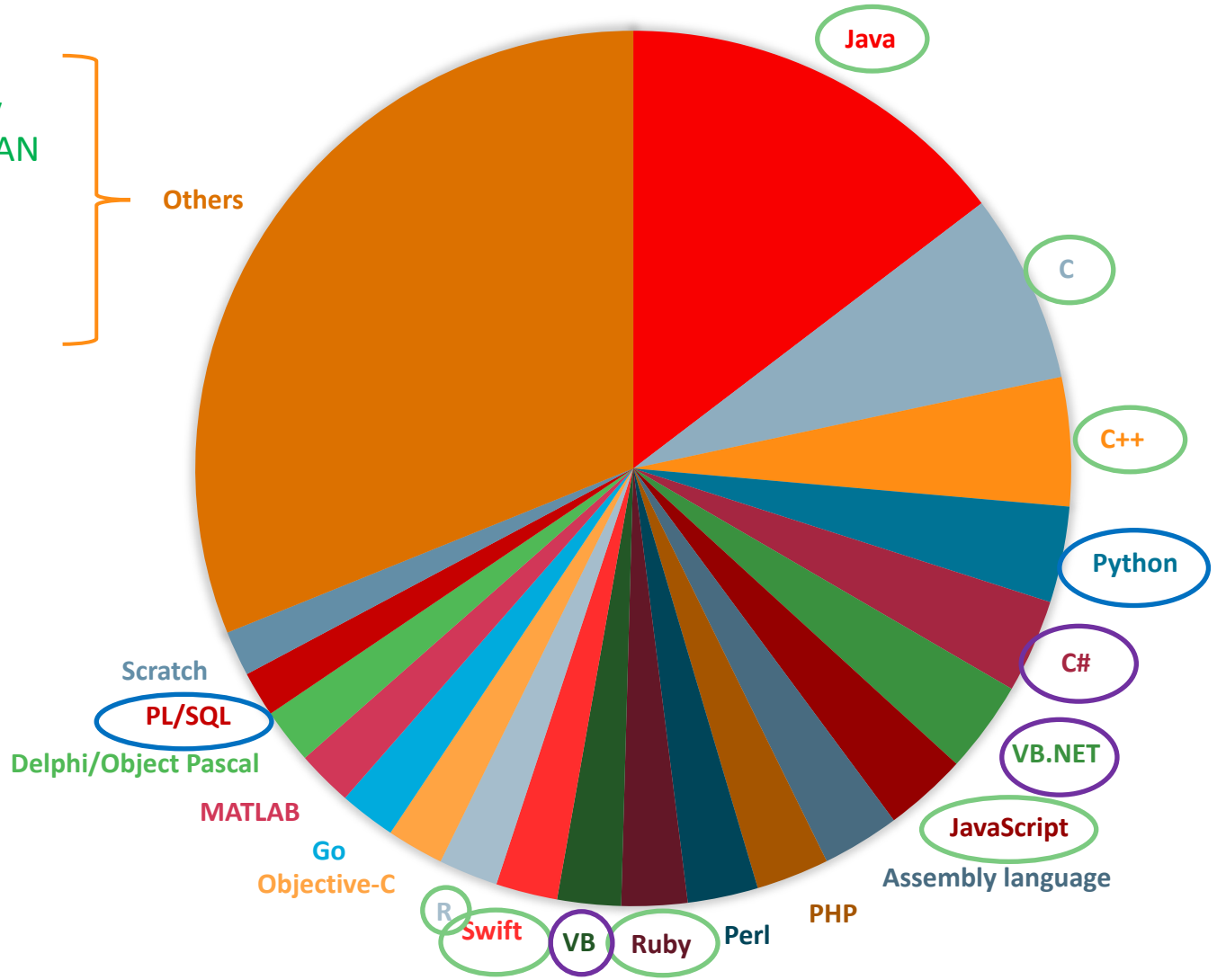
GraalVM Language Status

Available 2017

Available 2018

Proposed / Prototyping

Go를 제외한 나머지 언어를 서버에서
어플리케이션 개발할 때 얼마나 쓸까요?



대다수 랭귀지 런타임은 다른 언어지원이 시원치않…

- 일반적으로 하나 또는 두 개의 언어를 염두에 두고 설계된 언어 수준 런타임
- 언어 수준 VM에서 실행되는 이종언어는 일반적으로 높은 수준의 호환성을 달성하는 데 어려움.
- 다국어 런타임의 성능은 베이스 언어 이외의 경우는 좋을 수 없음.
  - JDK Nashorn은 구글 V8, 애플 사파리보다 3배 ~ 5배 느림.
- 일반적으로 언어 장벽을 넘어선 것에 대한 성능 페널티 (e.g. JNI)
- 여러 개의 별도 VM을 실행한다는 것은 두 개의 개별 힙을 크기로, 조정 및 관리하기 위해 엔진을 분리하고, 언어 경계를 넘어 10x-1000x 더 높은 오버헤드를 의미.

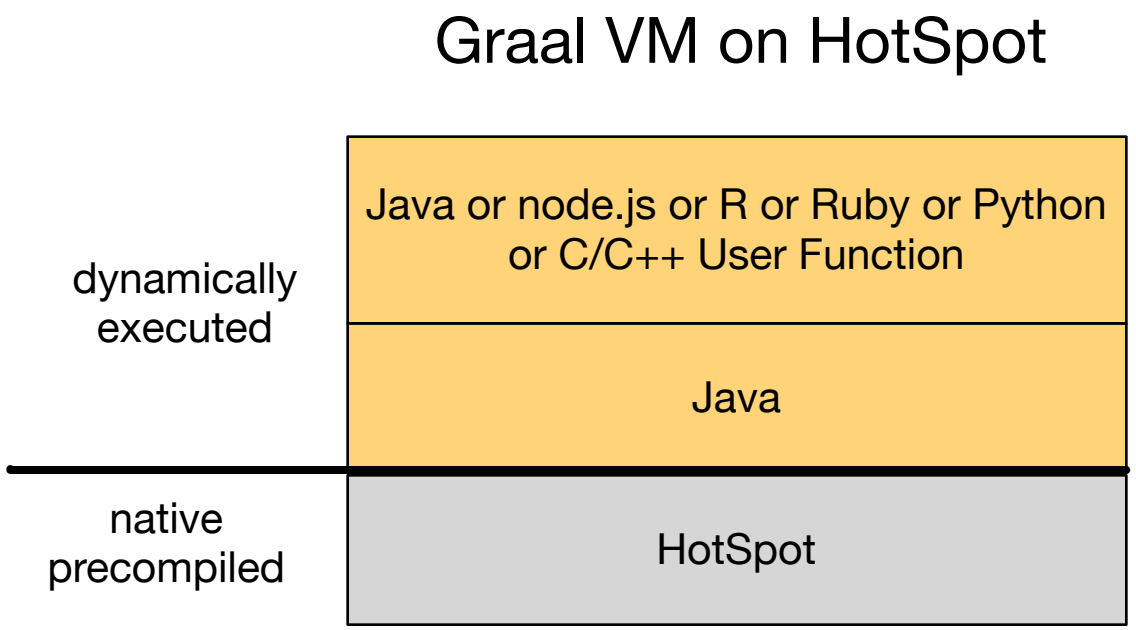
Embeddability

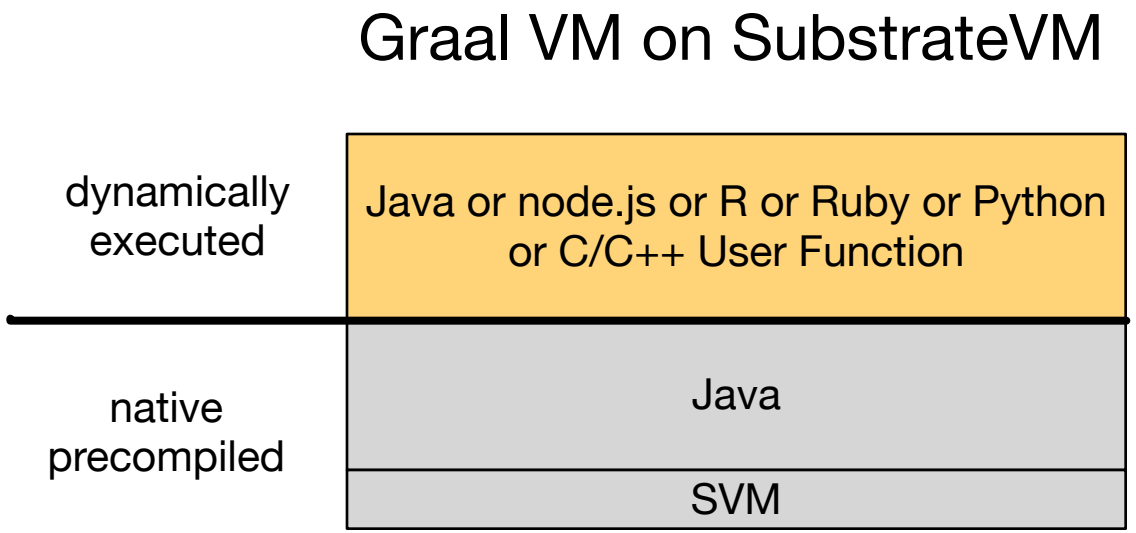
- GraalVM은 런타임(메모리 관리, 스레드 스케줄링, 코드 캐시 등)의 모든 부분을 포함 엔진에서 공급.

GraalVM is a hybrid of static & dynamic runtimes

SubstrateVM 은 GraalVM을 실행하는 독립실행형 및 포함 형 인프라를 제공합니다.

- 응용 프로그램 코드는 SVM을 통해 미리 컴파일되거나 동적으로 로드.
-





유연성과 보안을 위한 GraalVM two-layer design

Host Language (Java + native)

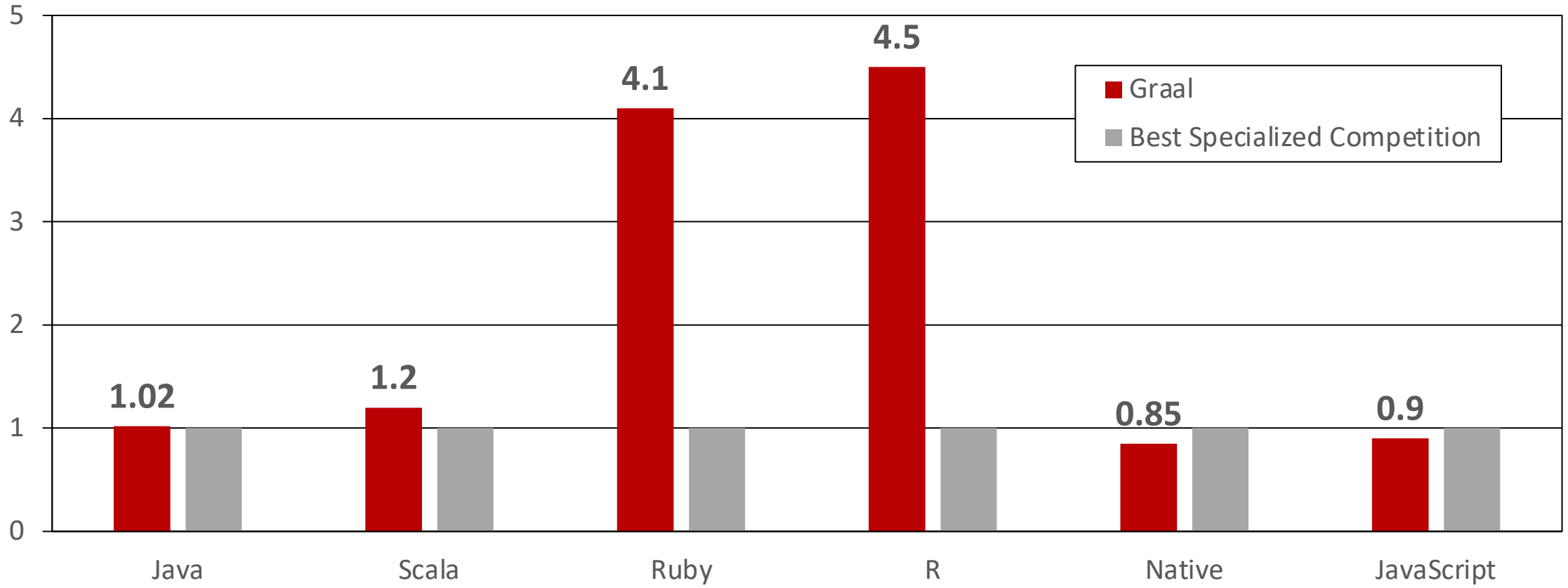
- Ahead-of-time compiled
  - 100x 빠른 실행속도
  - 2x 낮은 리소스 소모
- 관리(garbage collected, bounds-checked, & secured against stack overflows)
- 신뢰할 수 있는 코드는 여기에서 실행
- 화이트리스트 코드만 게스트에 액세스.
  - Reflection은 white lists로 관리

Guest Language (Ruby, JavaScript, Python, R, Java, native)

- Dynamically compiled
  - 최대 성능을 극대화하기 위해 실제 데이터 shape에 대해 런타임 프로파일
  - 실행 중인 시스템에 새 코드 추가
- 다이나믹 해서 유연하고 생산성 있는 환경구성
- 신뢰할 수 없는 코드는 Memory-boxed & time-boxed 상에서 구동
- 게스트 언어 라이브러리에 대한 네이티브 확장도 안전하게 실행 가능.
  - 확장에 대한 언어 교차 오버헤드 없음

High-Performance for **Each** Language

Speedup, higher is better



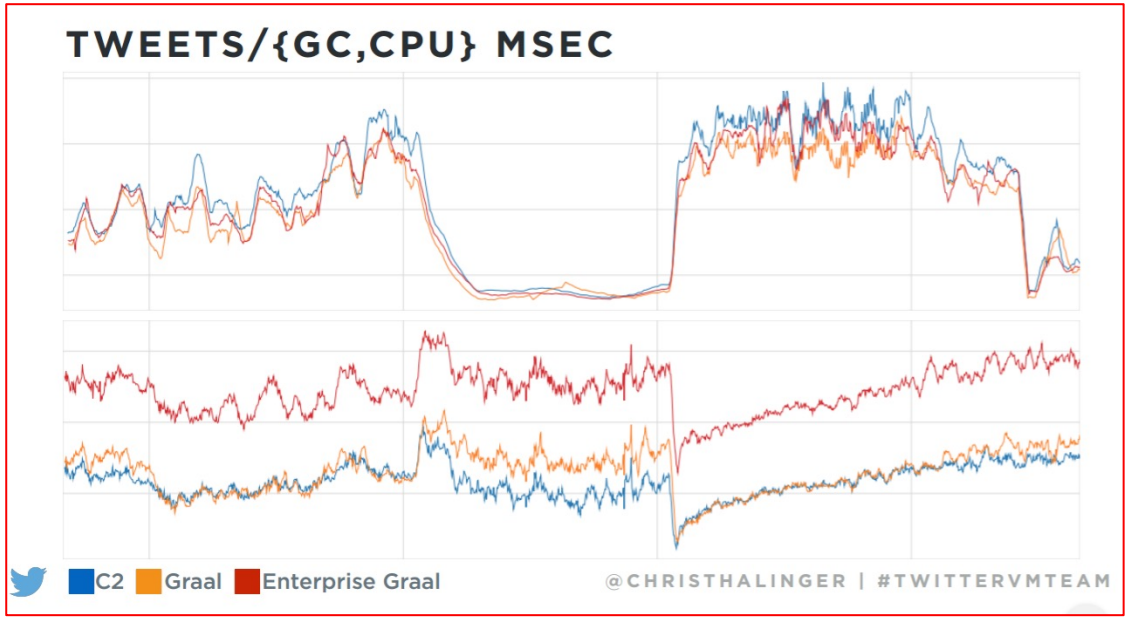
64비트 x86의 성능이 잘 알려진 벤치마크 제품군:

HotSpot/Server, JRuby, GNU R, LLVM AOT compiled, V8

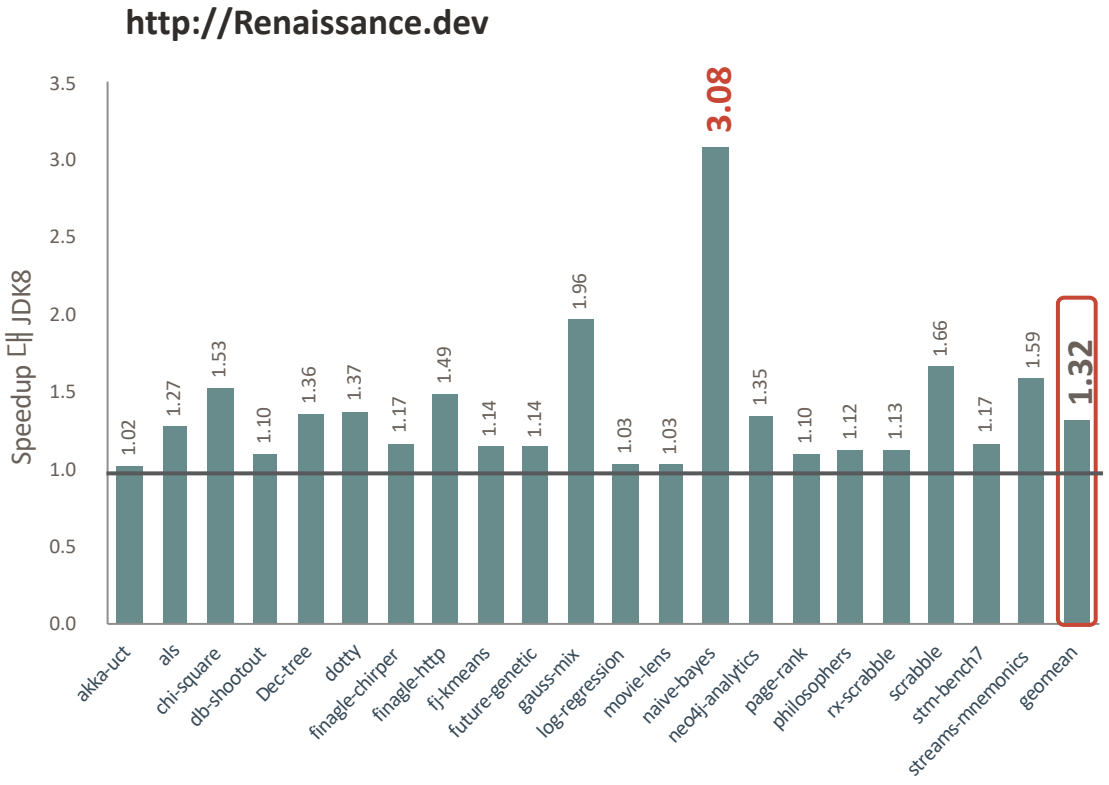
Application Level Performance

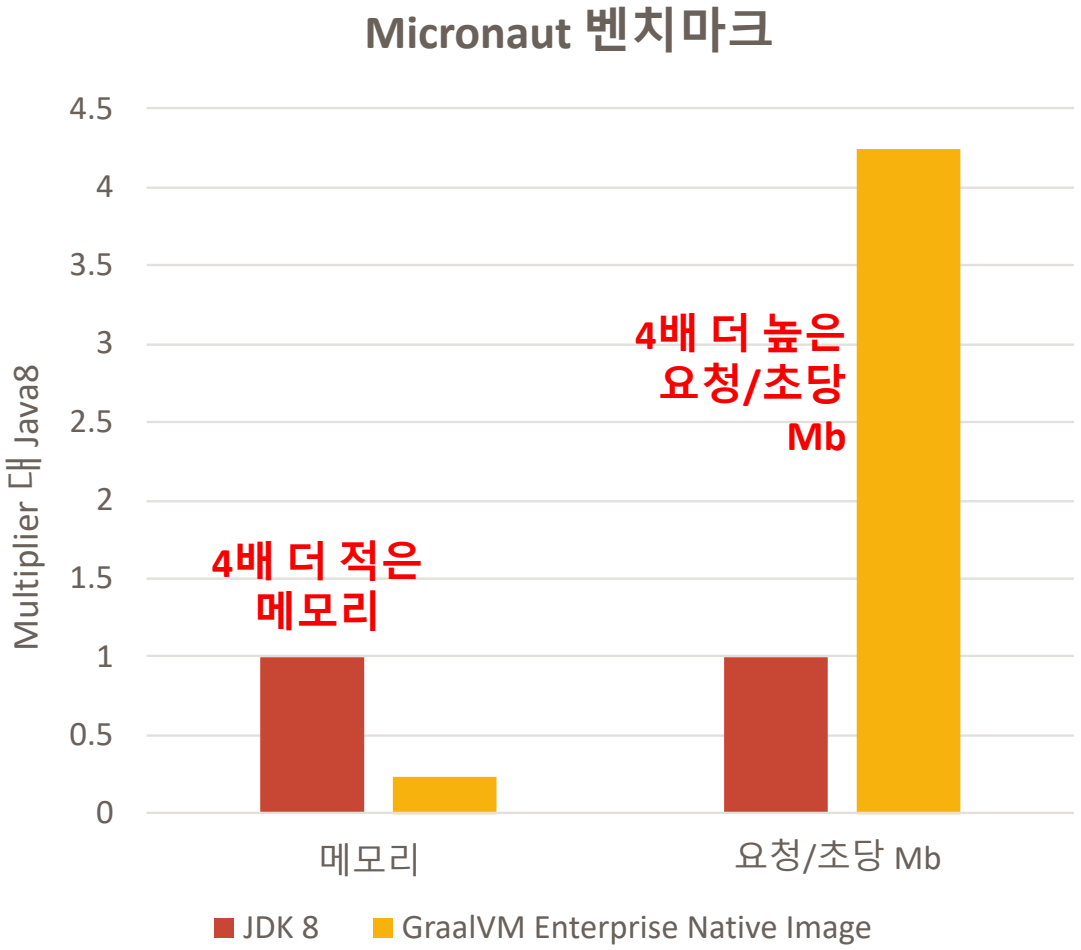
벤치마크 뿐만 아니라 실제 응용 분야는 어떻습니까?

- 트위터에 따르면, Graal은 자바8보다 트위터 25% 빠른 실행
- WebLogic Server는 성능이 10% 향상

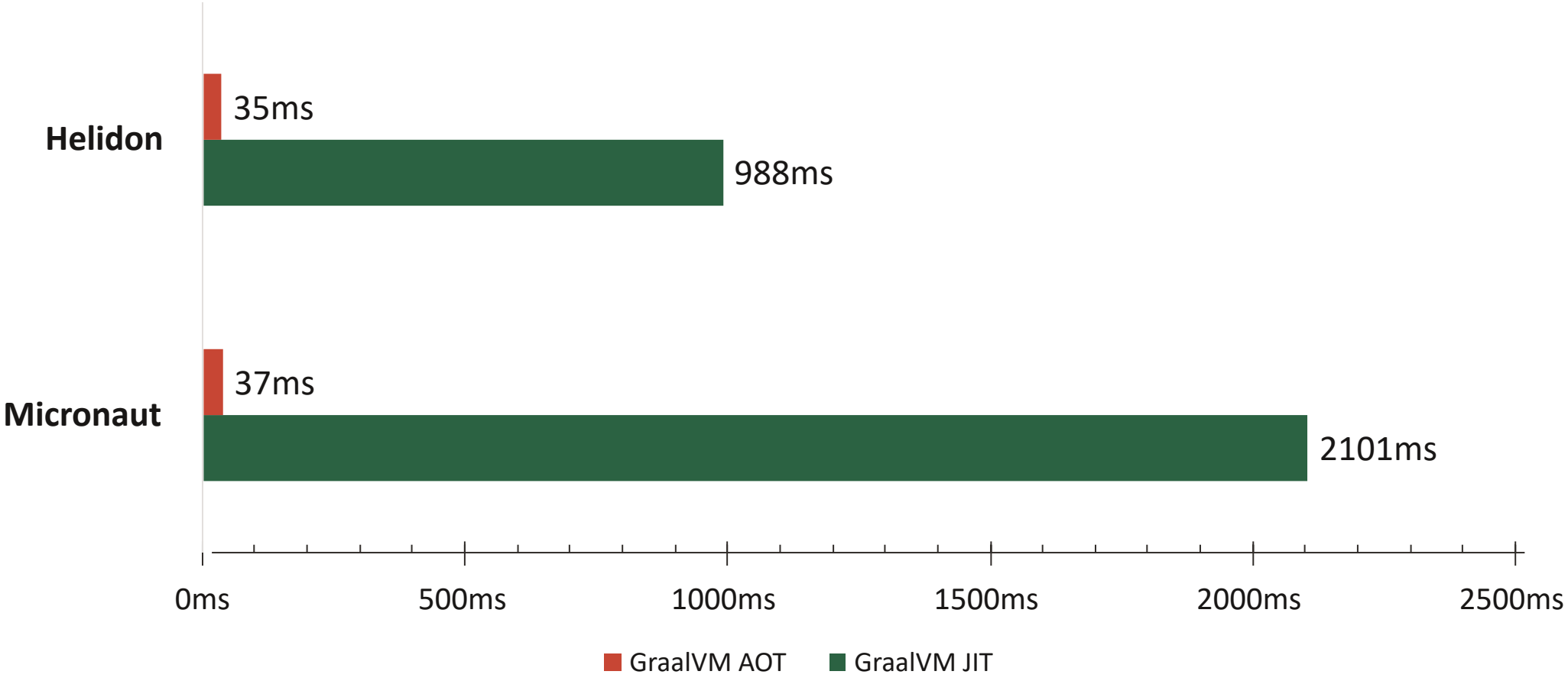


실제 애플리케이션 벤치마크에서 성능 향상

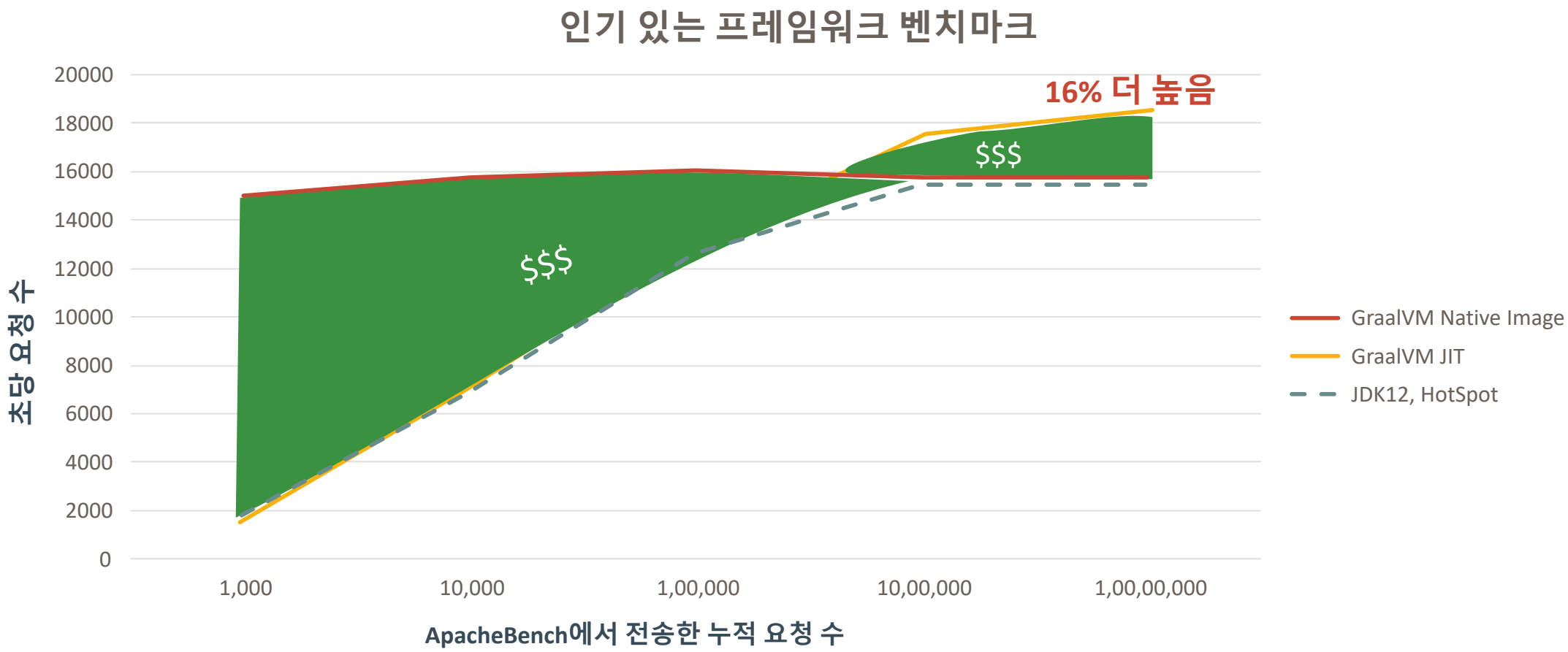




시간 단축
거의 즉각적인 시작



GraalVM Enterprise 처리량

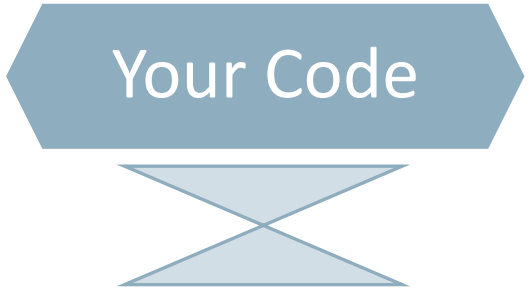


"우리는 많은 비용과 CPU 사이클을 절약할 수 있었습니다."

10% 성능 향상
20% 대기시간 감소

- Chris Thalinger
- Twitter 책임 시스템 엔지니어

Graal Ecosystem



Languages

Java
JavaScript
Ruby
R
Python
C/C++. FORTRAN, …

*

Graal

Optimizations
Tooling
Interoperability

*

Embeddings

HotSpot JVM
MySQL
Oracle RDBMS
Spark
nginx
…

부담 없는 GraalVM으로
Free Lunch
즐겨보시기 바랍니다.

감사합니다