Shank

A Simple and Fast Programming Language

Theodore Willis, Mendel Rubin, Wyatt Bushman and Timothy Gudlewski

University at Albany
State University of New York

Background

- In programming languages, **ease of use** and **speed** are thought of as **at odds** with each other.
- Easy-to-use languages are typically **interpreted**, and therefore **slower** than **compiled** languages.

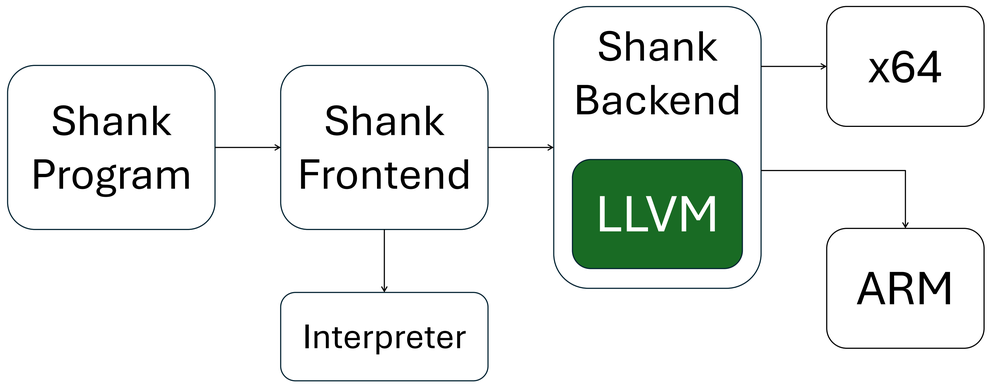
Our research focused on writing a programming language that can overcome this tradeoff – Shank.

- **Shank** is a programming language that provides language constructs that can **simply express** programs but can also compile down to **fast code**.
- It is one of the few compiled ***and*** interpreted programming languages.
- Shank cares about the **developer** while still valuing **machine efficiency**.
- Shank was already an interpreted language before our research began. **The goal of this project was to make it both a compiled and interpreted language, and to demonstrate its efficiency compared to industry standards.**

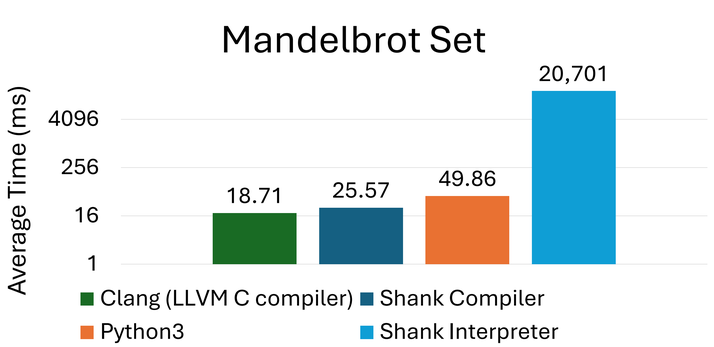
LLVM

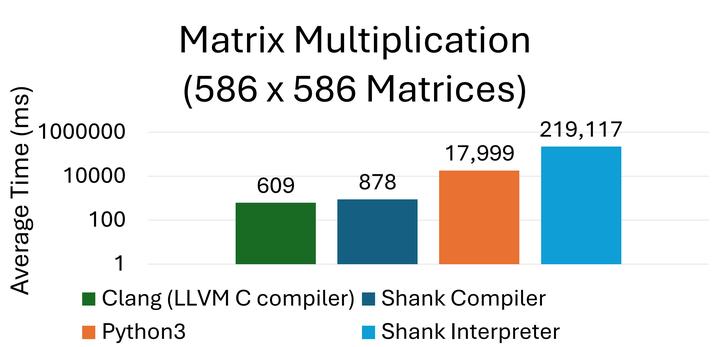
- Designed by University of Illinois graduate student Chris Lattner for a research project, LLVM streamlines **writing compilers** by generating a kind of generic assembly code, called **IR** (Intermediate Representation).
- LLVM will perform hundreds of common **optimizations** and output **machine code** for many different architectures; this makes writing a **cross-platform** compiler much easier.
- LLVM compilers are widely used throughout the industry.
  - Industry standard compilers that use LLVM include **clang, rustc, and swiftc**.
  - Companies that use LLVM include **Apple, IBM, Sony, Nvidia, Intel, AMD, and ARM**.
- The **Shank Compiler** uses the **LLVM framework** which is what makes it portable and efficient enough to be a fair competitor to most **C compilers.**

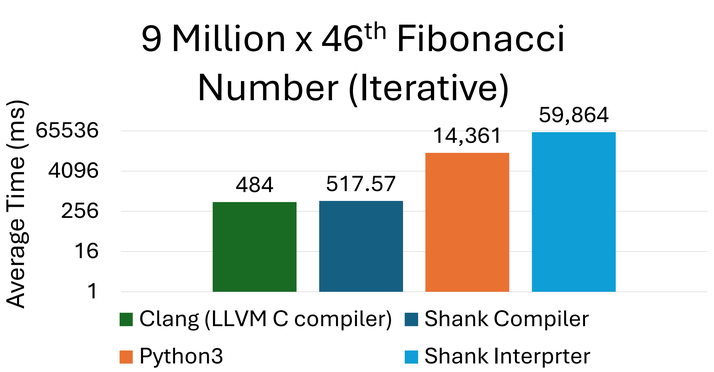
Project Architecture



Data







Data Synopsis/Conclusions

Average times were computed across 7 trials.

Clang : LLVM C compiler (Control)
Python3 : Industry standard Interpreter (Control)
Shank Compiler : Shank LLVM Compiler (Experimental)
Shank Interpreter : Shank's interpreter (Experimental)

Conclusion : A language that can compile its programs down to "executables" that target—and run "natively" on—a particular system gains a marked advantage in speed over purely interpreted languages.