A Presentation on *A Performance Survey on Stack-based and Register-based Virtual Machines*

Alex Fang, Tyler Liu

LA SALLE COLLEGE HIGH SCHOOL

November 30, 2016

Problem Statement

Fact

Every programmer deals with interpreted languages (Java, Python, etc.) every day.

1. Arguments arise when discussing the performance of interpreted languages: WHICH is faster? WHY and HOW is it faster?
2. As programmers we want to provide a better understanding of this issue using a SCIENTIFIC METHOD

Problem Boil-Down

Definition

The performance of an interpreted programming language = The runtime performance of its virtual machine

Fact

Two most popular types of virtual machines: The Stack-based Virtual Machine and The Register-based Virtual Machine

Through comparing and contrasting the performance between a stack-based virtual machine and a register-based virtual machine, we can find out:

1. Which one has the best performance
2. Different best-case applications of the two architectures

Hypothesis

Formula on the theoretical performance differences between a stack-based machine and a register based machine, proposed by Davis et. al:

$$\mathrm{T}_{VRM} \approx T_{VSM} - \#dispatches \times T_{dispatch} + \#fetches \times T_{fetch}$$

- *VRM*: Register-based Virtual Machine
- *VSM*: Stack-based Virtual Machine
- *T*: Time
- #*dispatches*: Total time spent in dispatches
- #*fetches*: Total time spent in fetches

Reasoning Behind The Hypothesis

$$\mathrm{T}_{VRM} \approx T_{VSM} - \#dispatches \times T_{dispatch} + \#fetches \times T_{fetch}$$

Stack-based Bytecode requires only ONE operand per instruction, whereas...

Register-based bytecode requires THREE operands per instruction.

Result: The stack-based virtual machine has a performance advantage in FETCHING the operands (fetch time); the register-based virtual machine has a advantage in a higher instruction efficiency (DISPATCHING instructions in general)

- Estimates:
  - Stack VM can perform better in fetch performance since it has 1-2 less operands per instruction
  - Register VM can have better performance in dispatch amount since it concentrates amount of instructions by having explicitly specified memory adresses

Approach

Instrument. We constructed two new virtual machines from scratch (one stack-based, another register-based)

Benchmark. We demonstrate the performance of each machines through two different types of benchmarks:

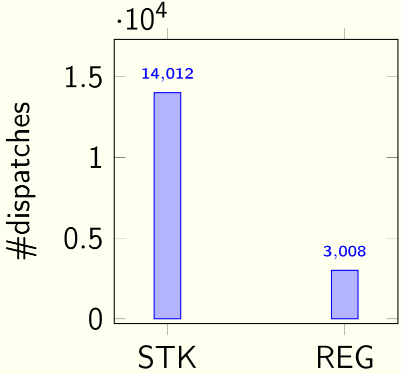
- Regular algorithm-based benchmark: Fibonacci, Recursion
- High density arithmetic benchmark: ExhaustiveCollatz, AddictiveAddition

Measurements. Using the formula of Davis et. al, we measured the four core factors of performance:

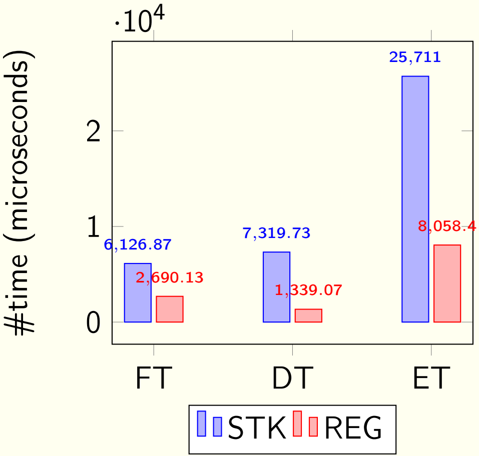
- Amount of dispatches per program
- Total time spent in instruction dispatch
- Total time spent in fetching instruction operands
- Total execution time

Benchmark Data - Regular Algorithm-based Benchmark

Fibonacci - Amount of Dispatches



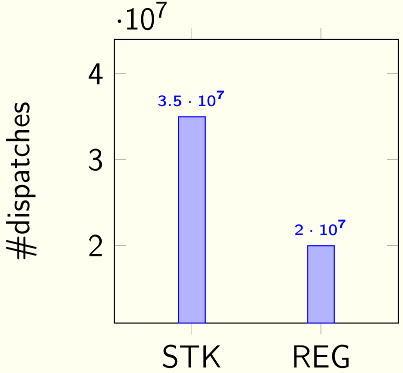
Fibonacci - Execution Time



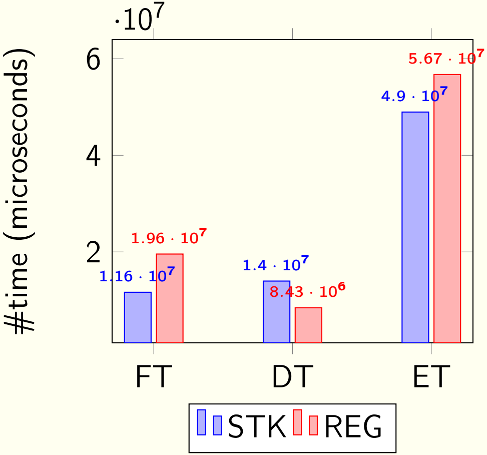
Note: STK=Stack-based Virtual Machine, REG=Register-based Virtual Machine

Benchmark Data - High Density Arithmetic Benchmark

AddictiveAddition - Amount of Dispatches



AddictiveAddition - Execution Time



Note: STK=Stack-based Virtual Machine, REG=Register-based Virtual Machine

Conclusion

- Overall, a register-based virtual machine is around 20% faster
- Stack-based VM performed better in fetch time (less)
- If you want to implement a high-performance, compact Domain-Specific Language on limited hardware, go for a register-based virtual machine!
- If you favor simplicity (both in byte code and in code for VM) over performance and want to perform dense read/write to the VM's memory space, implement a stack-based virtual machine!

Extra Works Conducted

- We pre-published our paper on the academia's largest e-print pre-publishing website, arXiv.org.
- We opensourced Everything related to this paper: virtual machine programs, bytecodes... In the hope of a better transparency and more feedback
- We plan to publish a revised paper on a Computer Science-related transaction after this conference
- Future works are planned for us to make revisions to our method of analysis to make the data gained more consistent and scientific.

References

- Brian Davis, Andrew Beatty, Kevin Casey, David Gregg, and John Waldron. The case for virtual register machines. In *Proceedings of the 2003 Workshop on Interpreters, Virtual Machines, and Emulators*, IVME '03. pages 41-49, New York, NY, USA, 2003. ACM.