目前主流的JVM，基本都是基于栈式架构的虚拟机：

基于栈的指令集架构

设计和实现更简单，适用于资源受限的系统

避开了寄存器的分配难题：使用零地址指令方式分配

指令流中的指令大部分都是零地址的指令，其执行过程依赖于操作栈。指令集更小，编译器容易实现。

不需要硬件的特殊支持，可移植性好，能够更好的实现跨平台

由于是基于栈的架构，零地址指令方式，所以也有相应的缺点：完成同一个操作需要的更多的出栈和入栈指令，这就意味着将需要更多次的指令分派（instruction dispatch）次数以及内存读写次数。

为了解决或者优化这一问题，Hotspot MV的那些大佬们开发了栈顶缓存技术

该技术将栈顶的元素全部缓存到了CPU的物理寄存器中，既能降低内存的读写次数，又能提高执行指令的效率。大概就是这么一个样子，了解记录一下

**栈顶缓存技术（Top-of-Stack Cashing）：（说白了就是相同的指令会执行很多次，仅仅只是每次执行指令操作的操作数不一样，而重复的去加载指令然后舍弃，必然严重的损耗性能，减少加载次数那就直接把常用的缓存起来。这就像操作系统和计算机组成原理中提到并使用的方法。）**

1、前面提过，基于栈式架构的虚拟机所使用的零地址指令更加紧凑，但完成一项操作的时候必然需要使用更多的入栈和出栈指令，这同时也就意味着将需要更多的指令分派（instruction dispatch）次数和内存读/写次数。

2、由于操作数是存储在内存中的，因此频繁地执行内存读/写操作必然会影响执行速度。为了解决这个问题，Hotspot JVM地设计者们，提出了栈顶缓存（TOS，Top-of-Stack Cashing）技术，将栈顶元素全部缓存在物理CPU地寄存器中，以此降低对内存地读/写次数，提升执行引擎地执行效率。