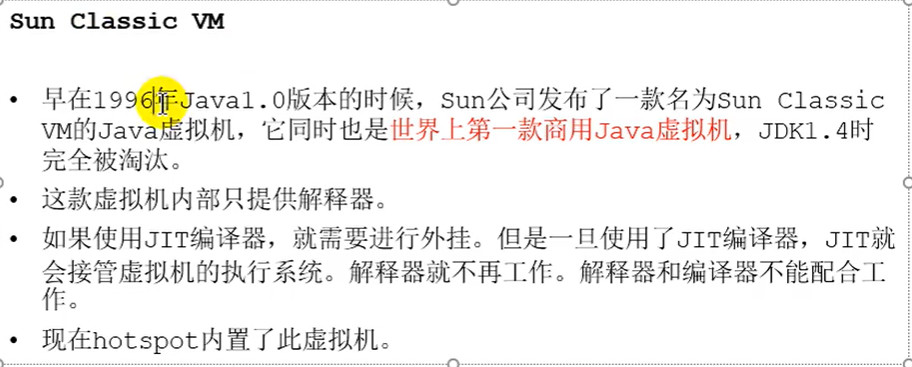
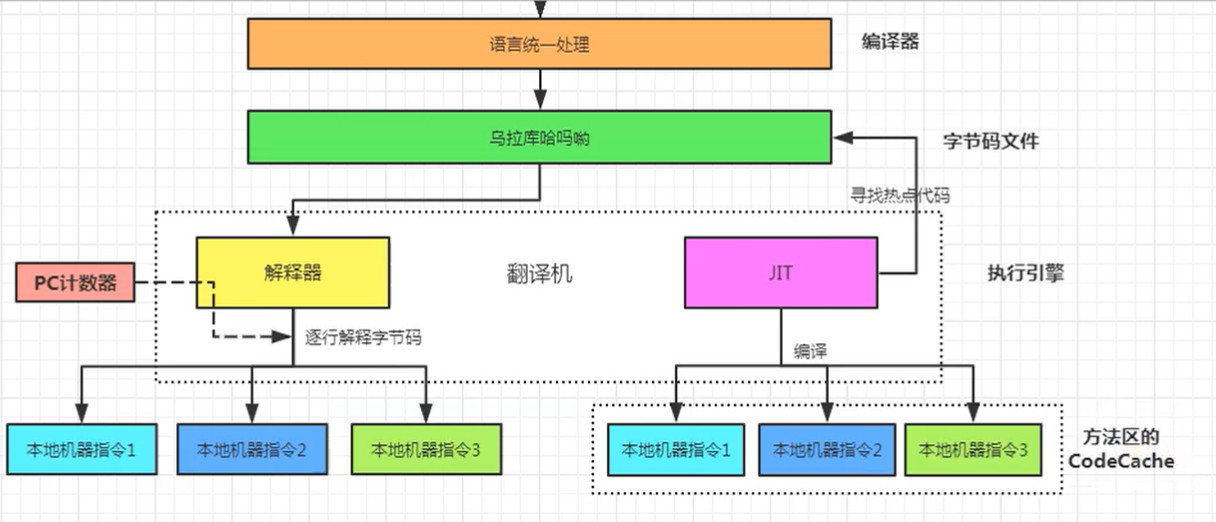
（今天的Java速度已经不亚于c与c++了）

（虚拟机执行的都是前端编译后的class）

第一台商用Java虚拟机：





两者中只要有一款就能正常执行，但是如果只有解释器，那么就会没执行一条代码都会去解释，然后执行，效率就会很低，但是如果加上了JIT（将代码即时编译为指令缓存起来），会将热点代码进行即时编译为指令然后缓存起来，这样就提升了效率。（如果只有JIT，那么我们就会将所有代码即时编译然后缓存，这样系统开销就会很大，比如时间上，会在开始时程序卡顿，这也就是为什么会有Java比C慢的感觉，所以不能用很好的JIT，要限制JIT的开销。）

（

解释器：相当于走路，响应快，执行慢

JIT：相当于坐公交，等待长，执行快

传统是两者只能选一种。为解决响应和执行效率问题，现在将两者结合，把等待的时间拿来走路，当公交到时，再上车。虽然现实是路程时间花费一样，但是放到这个上面就解决了响应和执行的问题。

）

二：第二台虚拟机：

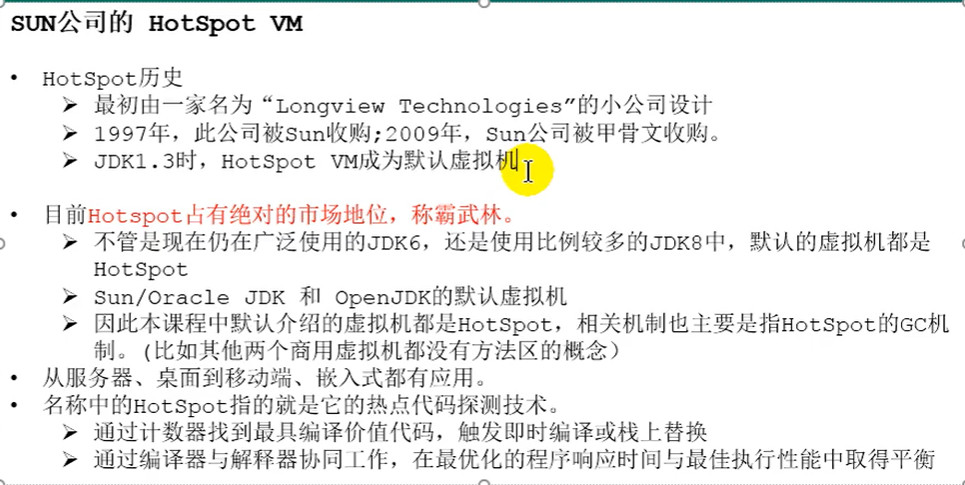
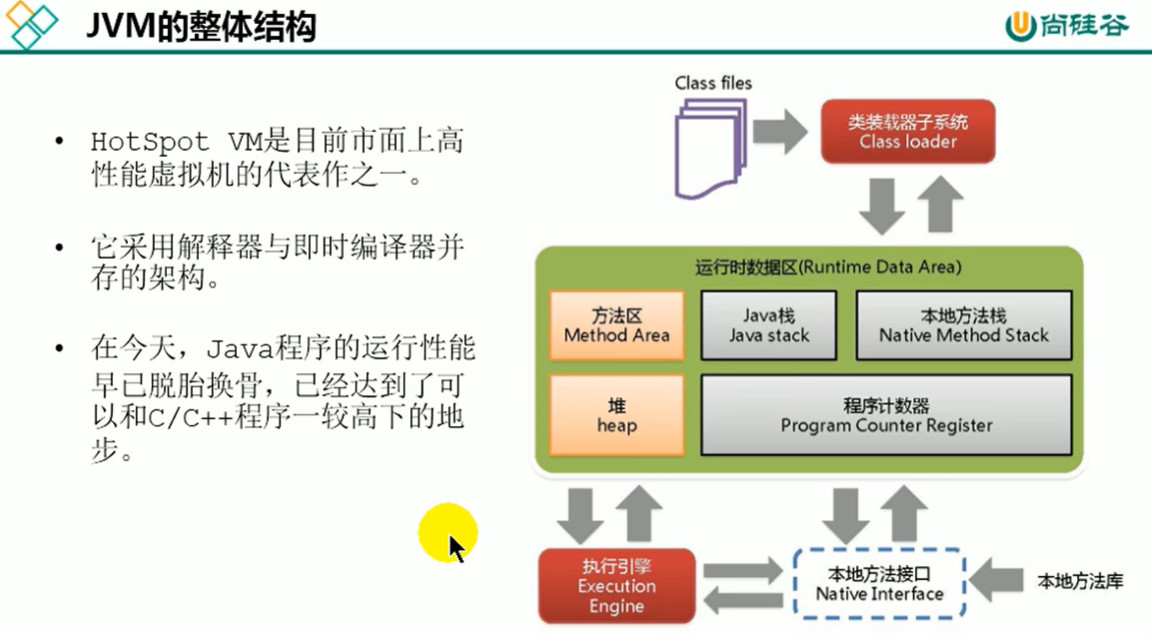


？？没懂他讲的意思！！！！！比如在jvm中，栈中有一个地址，指向堆中一个对象，当垃圾回收器开始处理堆区时，如果这个对象不是个如果无法确定这个地址的数据类型究竟是不是对象的引用，那么

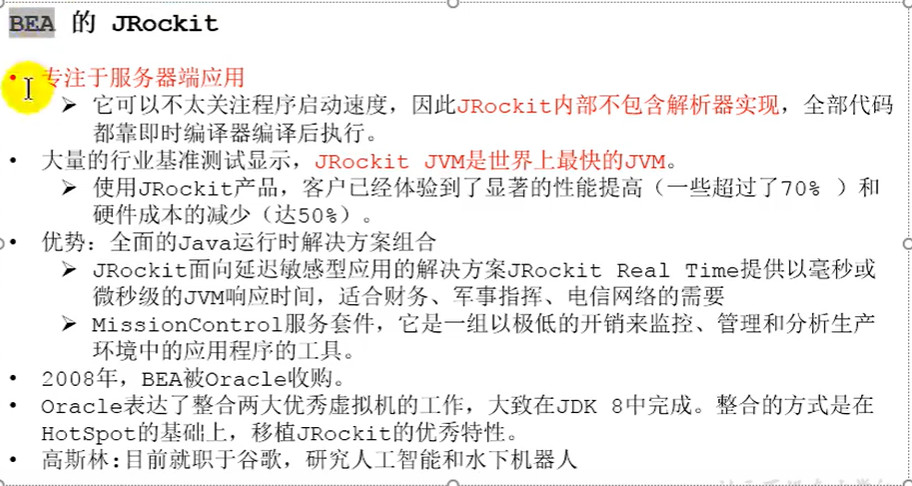
三：三大虚拟机之HotSpot虚拟机:假如我们现在需要提升我们某段代码的执行效率，我们是否就可以把它想法设定为热点代码，那我们就需要知道Hotspot是如何来判断是否为热点代码的？

（注意：解释器主要负责响应时间，编译器主要负责执行的性能。）

（编译执行加解释执行相结合）



四：三大虚拟机之JRockit虚拟机：(编译执行)

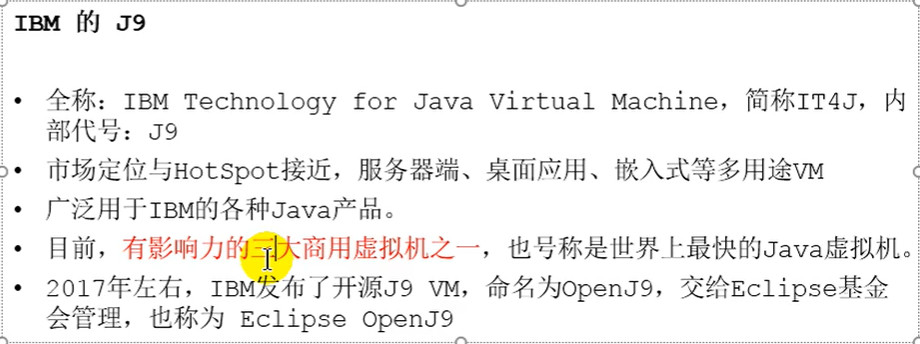


（JMC专门用来监督内存泄漏问题）

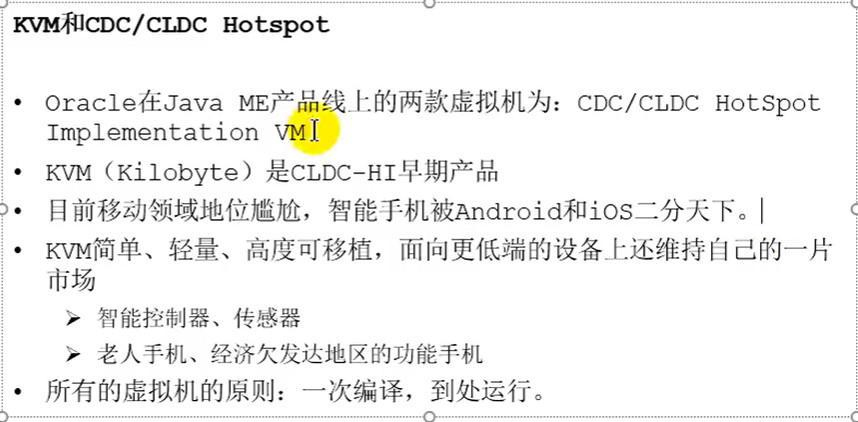
（前两大已经被Oracle收购）

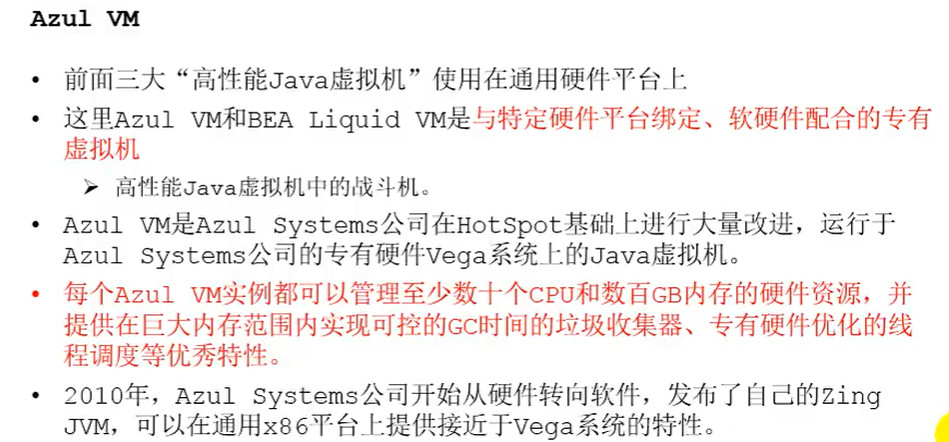
五：三大虚拟机之J9虚拟机：

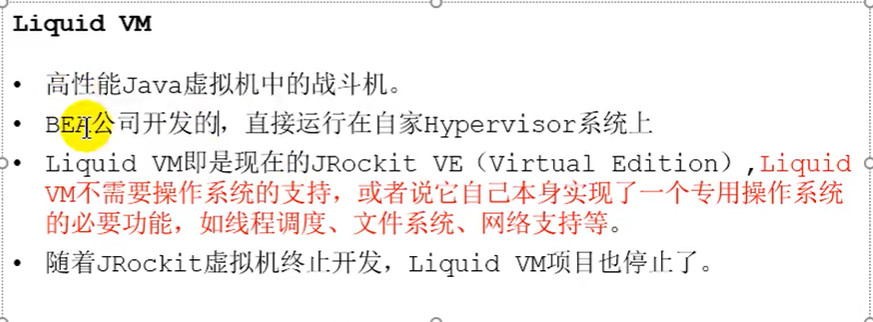
(在IBM自己的产品中使用很稳定很快，因为它有自己的架构，有着自己一系列的产品，产品耦合度大)

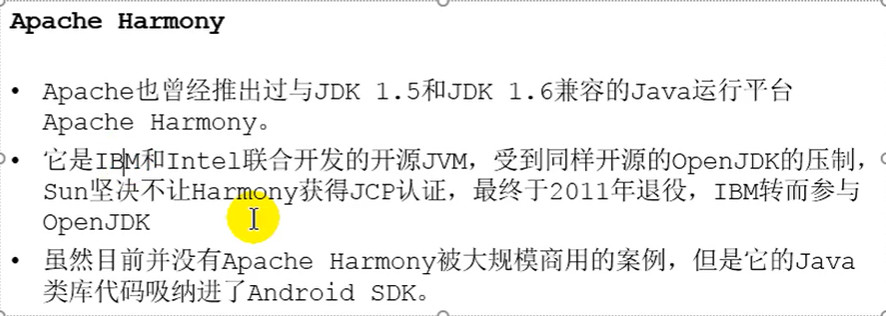


其他虚拟机：（这个编译指的是前端编译）

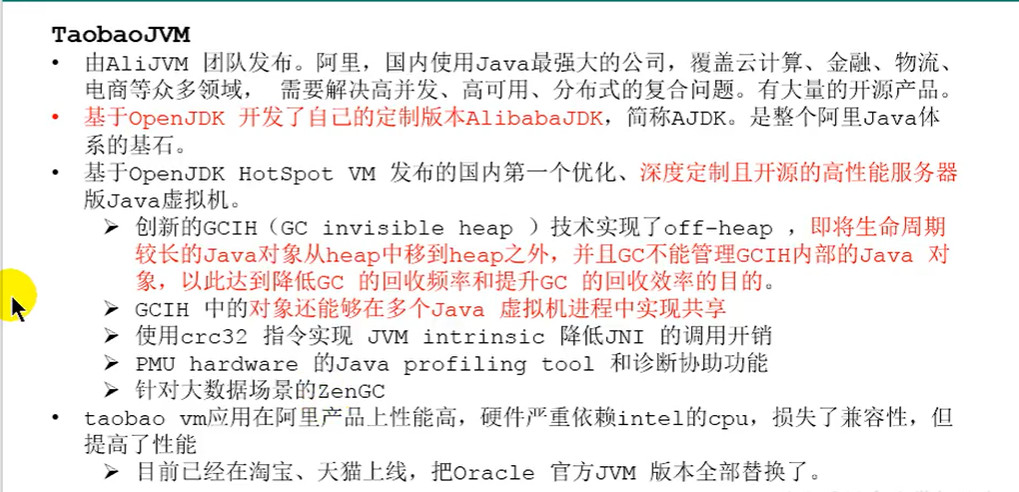








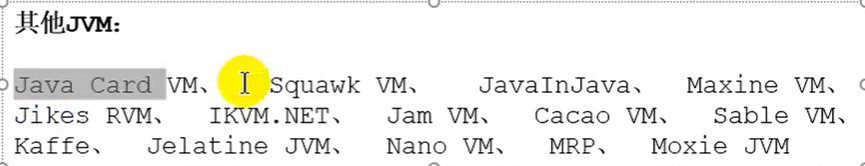


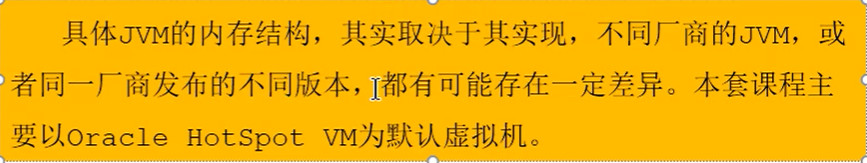


不是Java虚拟机的虚拟机：

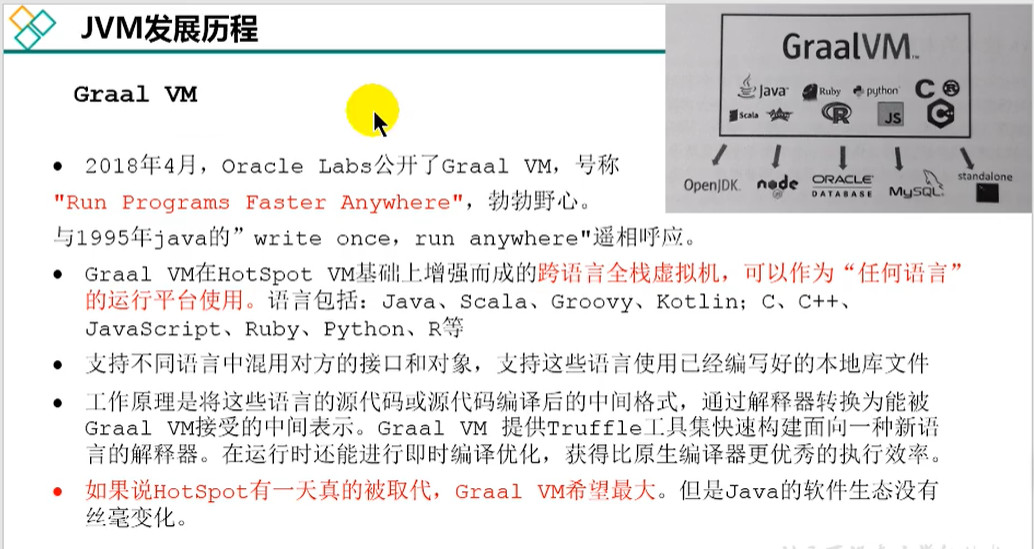
（虽然用的Java进行开发，却不能在Java虚拟机上运行）







Hotspot未来的替代品：



与具体机器指令集相关，而不再与语言相关，（也就是机器指令集，虽然不同机器可能对不同的指令实现方式不一样，但是我的语言只要能调用指令实现功能即可）