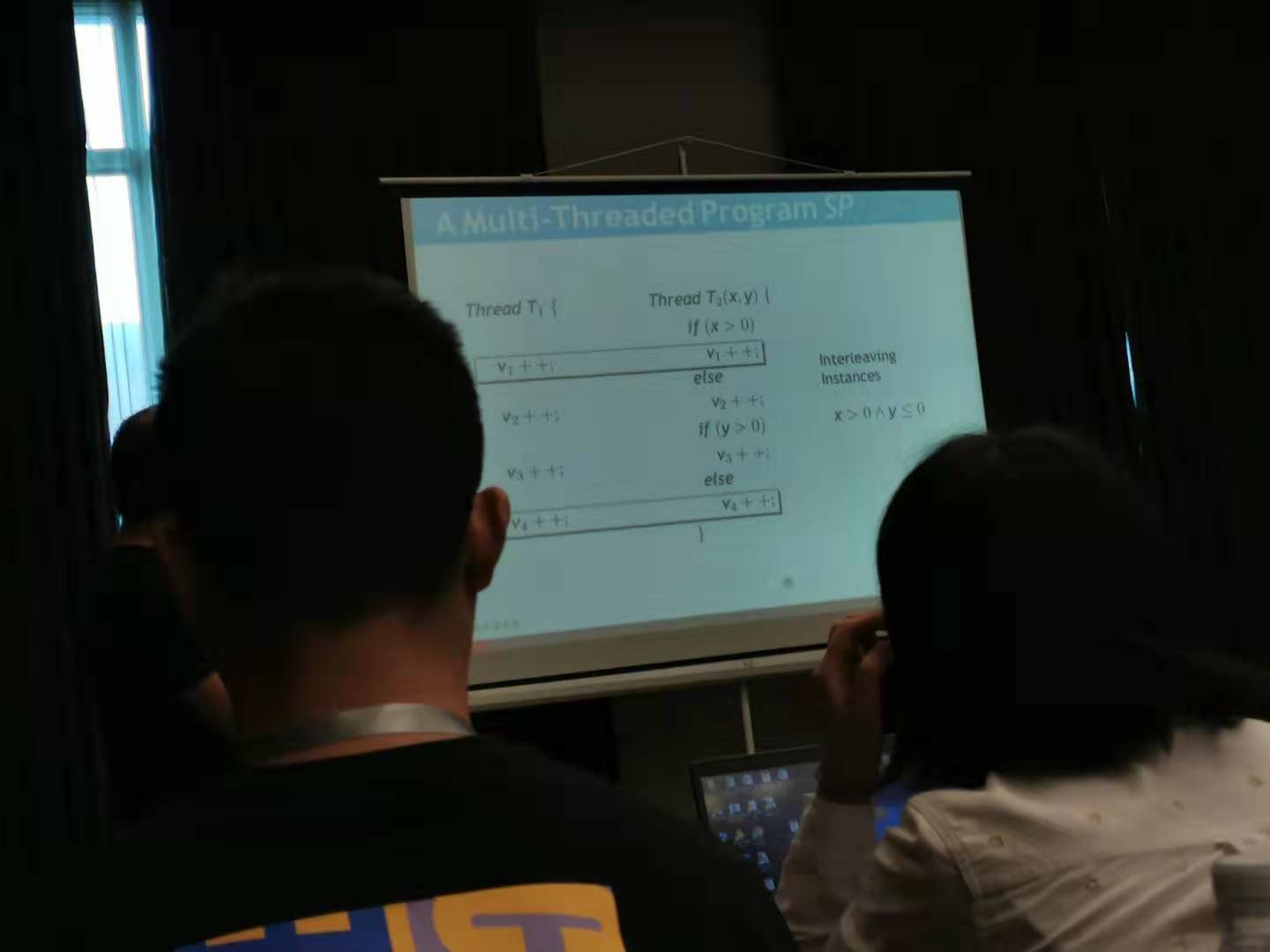
基于内存共享的多线程并发程序的测试用例的生成和oracle问题

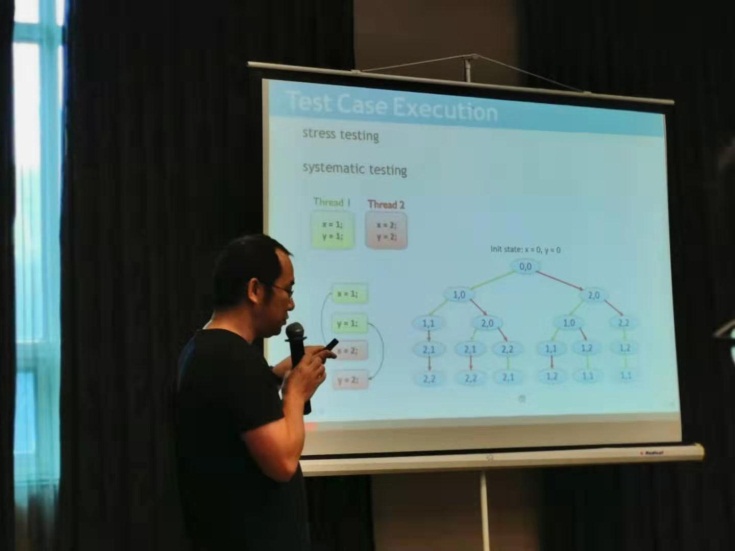
交错实例（Interleaving instance）



交错和输入相关---data dependent，所以要考虑测试输入问题。

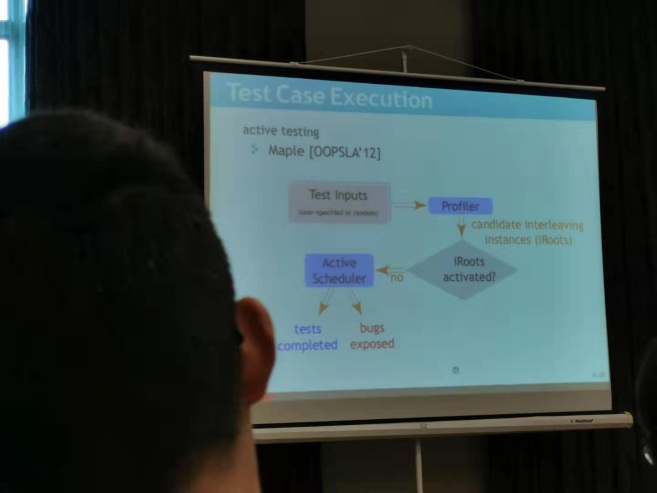
更关注的是并发测试用例是如何执行的，因为测试用例的执行需要覆盖不同的交错实例。

典型的方法：压力测试、随机测试、系统测试。



系统测试面临的问题：状态爆炸。

现在比较成熟的方法，active testing---主动测试。（非通用，特定的）：



两个阶段：

profiler，先用类似于压力测试的方法执行程序，然后收集测试执行过程中的trace，然后确定此次测试所触发的interleaving instances，根据这些已经出发的interleaving instances 去预测是不是有未知的还没有被触发到的交错实例，如果有交错实例没有被触发到，通过systematic testing方法控制线程调度的一种普通策略，来触发那些交错实例。

线程优先级

交错实例是测试用例。

执行之后收集轨迹，根据已经发生的去预测所有可能发生的，

并发程序的执行是不确定的，oracle不存在或开销大。

并发程序测试的问题：

测试输入生成，他们的工作。

执行

Oracle问题

Maple：profiler相当于动态，

去执行不同的没触发到的交错。

强制优先级调度，基于linux底层的。

两个维度的diversity？

Active testing+ART

