1.

老师们，同学们大家好，我今天汇报的工作是虚拟化环境下基于平均淘汰时间模型的内存工作集预测

2.

汇报首先会介绍工作的研究背景，前人的相关工作，然后介绍内存工作集预测系统的设计与实现，验证的实验，最后总结本文的工作和对未来工作的展望

3.

我的工作是在虚拟化平台下内存工作集预测，所以首先要了解的问题是虚拟化技术是什么？对虚拟化技术有一个基本的了解，然后在虚拟化平台上为什么需要动态内存调度以及如何去做调度的问题？

4.1

虚拟化技术分为三类：平台虚拟化、资源虚拟化以及应用程序虚拟化，平台虚拟化是针对计算机和操作系统的虚拟化，资源虚拟化是针对特定的系统资源的虚拟化，比如内存、存储、网络资源等，应用程序虚拟化包括仿真、模拟、解释技术等。

4.2

我们通常所说的虚拟化主要是指平台虚拟化技术，通过使用控制程序也被称为 **Virtual Machine Monitor** 或 **Hypervisor**，隐藏特定计算平台的实际物理特性，为用户提供抽象的、统一的、模拟的计算环境（称为**虚拟机**）。虚拟机中运行的操作系统被称为客户机操作系统（Guest OS），运行虚拟机监控器的操作系统被称为主机操作系统（Host OS）

5

平台虚拟化细分为三类，全虚拟化是指虚拟机模拟了完整的底层硬件，包括处理器、物理内存、时钟、外设等，使得为原始硬件设计的操作系统或其它系统软件完全不做任何修改就可以在虚拟机中运行；半虚拟化是一种修改 Guest OS 部分访问特权状态的代码以便直接与 VMM 交互的技术；硬件辅助虚拟化是指借助硬件（主要是主机处理器）的支持来实现高效的全虚拟化。

6

6.1

了解了虚拟化的定义和分类，现在需要了解本文工作的研究目的，在虚拟化平台下为什么做动态内存调度基于下面三个原因，虚拟化平台下通常在一个物理机上会运行多个虚拟机，而每个虚拟机上运行的工作负载对内存的使用是不断变化着的，

6.2

而物理资源的内存总数是有限的，平均分配显然不是一种好的资源分配方式，而在实际使用中往往虚拟化运营商会使用内存超卖，这意味着售出的内存总和超过物理限制，如果没有一个合适的调度方式势必会让某些内存使用量激增的虚拟机内存不足。

6.3

在现代虚拟化平台下，内存的方式都是静态分配，即在虚拟机启动的时候分配固定大小的内存，在超卖情况下分配的内存显然会小于实际购买的内存数

7

虚拟化下动态内存调度主要通过三种方式实现

7.1

气球驱动方式是通过在虚拟机里添加气球模块，要减少虚拟机的内存大小则充气，虚拟机管理器回收气球驱动充气用到的页面，要增加虚拟机的内存则放气

7.2

远程内存通过将本地的内存页面缓存到远程空闲虚拟机的内存中来释放内存使用压力

7.3

当所有虚拟机均面临内存压力的时候就要把内存资源不足的虚拟机迁移到相对空闲物理机上去

8

有了调度手段，现在还缺乏调度指标，虚拟机内存不足时到底需要多少额外内存，空闲虚拟机目前在使用多少内存有多少内存空闲能够拿出来调度这些问题最终都是虚拟机内存工作集预测准确度问题。

工作集定义为一个进程在t 时刻的工作集W(t,τ)为进程在(t- τ,t)时间段里访问到的内存大小

通过定义计算工作集的问题是无法刻画内存大小和页面失效率的关系，失效率的含义是在给定内存情况下，页面访问没有在内存中命中所占的比例，所以失效率越高，内存紧缺程度越大，额外的内存访问开销也就越大

9

总结来看，要做到虚拟化动态内存预测，对虚拟机的工作集精确预测有很高的要求，MRC传统算法时间复杂度较高不适宜在线预测，接下来会介绍相关的对于MRC优化的算法以及关于内存工作集预测的一些研究。

10

图2.1是一个失效率曲线的示意图，当缓存大小为0的时候失效率为100%，随着缓存的增大，失效率降低，有更多的页面在缓存中命中，表2.1和2.2是LRU算法执行过程示例，维护一个页面访问的LRU栈，每一次访问获得该页面在栈里的位置，更新重用距离分布表，并将该页面移动到LRU栈头部。在运行足够长时间之后会得到一个重用距离频率分布表，MRC曲线就通过重用距离分布计算。

11

MRC算法优化主要有两种手段，一种是优化LRU算法模拟的时间，用链表维护LRU的代价为O(NM)，N是页面访问数，M是LRU栈的深度，用平衡树优化的时间复杂度为O(NlogM)。另外一种优化方法针对N的优化，通过采样的方法只截获部分页面访问获得近似MRC。

12

另外还有一些别的关于内存工作集预测的方法，第一种方法是系统参数法，利用操作系统提供的参数得到内存使用估计值，第二种方法是采样标记法，标记一部分页面，一段时间扫描统计标记页面的比例估算内存使用量。第三种方法页面截获法可分为软件方法和硬件方法，硬件方法增加额外部件截获内存访问，通过截获的内存访问计算MRC，软件方法则是需要人为地对页面进行标记制造页面中断。

13

比较这几种方法，系统参数和采样标记方法虽然开销不大，但是精度一般，页面截获法能够获得页面的访问序列所以精度较高，但是在缺乏特定硬件帮助情况下用缺页中断的方式截获页面访问会增加内存访问的延时，会给应用程序增加额外巨大的开销

14

接下来介绍系统的设计与实现，我们采用页面截获的方式获得虚拟机的访存序列，在虚拟化环境下截获内存页面方法和裸机上会有所不同，需要首先了解清楚虚拟化中内存虚拟化机制，在获得访存序列之后需要实现AET模型，由于截获内存页面的巨大开销，所以系统还需要大量的优化手段。

15

15.1

我们使用的虚拟化环境是通过影子页表完成地址翻译的，由于在全虚拟化下虚拟机感知不到虚拟器管理器的存在，虚拟机会把虚拟机虚拟地址转换为从零开始的虚拟机物理地址，而实际上多个虚拟机在共享物理内存，所以虚拟机物理地址还需要转化为真正的物理地址，影子页表完成的工作就是将虚拟机物理地址转换为机器物理地址的工作，当虚拟机加载页表项首地址CR3的时候实际上载入CR3寄存器的是影子页表首地址，这样虚拟机就能够安全透明地使用物理内存地址。

15.2

那么影子页表是在何时同步呢，特别是在进程切换的时候影子页表如何更新到对应新的进程？第一种方法是在进程切换的时候将影子页表清空，虚拟机在访问内存时就因为页表项不存在而陷入虚拟机管理器，另一种优化方法是在进程切换时将影子页表项设置为不存在，这样没有被新进程使用的页表项得到保留。还有一个同步时机是在页表项更新access位和dirty位的时候，影子页表项被建立好之后会把页置为只读的，这样当虚拟机修改页面并且修改dirty位的时候会因为只读保护陷入，这时候同步影子页表dirty位

由影子页表的同步机制我们知道所有的页面首次访问前均会发生一次影子错误，所以我们就能够收集到所有的PTE，我们通过置位PTE的保留位来使得页面访问发生错误陷入到虚拟机管理器，这种错误称为AET错误

16

在介绍AET实现之前先简要介绍AET原理

17

18

19

20

29

fake是我们自己编写的测试程序，它首先会malloc一个g的地址空间，它的运行会分为七个阶段，每个阶段扫描固定大小的内存空间，这个值是由随机数产生，memused是系统累积很长时间使用的内存数，并不仅仅表明当前正在运行的进程所使用的页面数，休眠的进程没有被替换出内存的部分也会统计，所以这个值比当前工作集要偏大，进程的虚拟内存大小在这里是我们malloc的大小，1个g，而活跃页面数和res相对会和实际工作集接近，但是当工作集突然变小时，活跃页面和res并不能反应出这一变化，这是因为系统内存充裕，无需回收页面。

30

对于spec的测试程序来说也有类似结果，通常实际的工作集都会比系统所提供的统计值要小

31

由于我们系统采用了采样来降低开销，为了验证不同采样率下的结果，我们计算了各个测试程序的工作集曲线，工作集曲线是通过mrc里定义95%失效率的内存大小为工作集而绘制出来的一条随时间变化的工作集变化曲线图，这里只展示了两个spec测试程序在不同采样率下的工作集变化曲线，通过这个实验我们验证采样在我们在线预测上的可行性。