1. 存在的问题

现代计算机内存系统主要可以针对两个方面进行优化，一方面是降低内存访问延时；另一方面是提高内存访问带宽。而对于GPU内存系统中的则更侧重于提高访问带宽。为此，现代GPU内存系统被划分为多个可并行访问的内存通道，并且系统将内存地址通过某一特定的地址-通道映射规则映射到不同的内存通道，从而提高整个GPU内存系统的并行访存能力，进而提高GPU的带宽。

通常来说，一系列连续内存访问的的高位变化较少，相反，地址的地位变化较为频繁，因此在计算机系统结构设计时，通常倾向于选取内存地址中较低的部分位用于选择内存通道。但是在实际应用的过程中，由于特定的GPU访存的地址序列，访问的内存地址通过映射转换为通道时，大量访问的请求会集中于同一个通道，导致该通道的冲突，而其他通道处于相对空闲的状态。如此导致有效带宽在实际上仅仅为最大带宽的一部分。

1. 伪随机交织方法

为了解决上述访问请求大量集中于同一个通道的冲突问题，曾经有学者提出内存地址交织(伪随机交织)方法对此问题进行改进。其中主要思想为：在选取用于确定内存通道的若干位的同时，这里称为channel bit，也选取内存地址中的其他位，这里成为bit——用于与channel bit进行异或操作，从而得到新的用于确定内存通道的地址。在统计学的基础上，此处的异或操作是的每次访存的请求所映射到的内存通道呈现伪随机性，是的GPU原始访问模式映射为在内存通道中分布得更加均匀的访问模式，从而保证最大程度地提高每个内存通道的利用率。

1. 伪随机交织方法不好在哪

但是随着现代GPU执行的任务类型逐渐增多且复杂，上述固定的内存地址交织方法无法满足所有任务类型的需求。在大多数情况下，上述方法使得在一段时间内的内存访问请求在不同通道之间是相对均匀的，但是对于不同的内存地址交织方式可能导致内存地址的重叠，从而产生访存错误；并且GPU一旦启动，固定的内存地址交织方式便无法修改。

1. 基于扩展页表的优化方法，根据不同任务标识选择不同映射方式，选择最佳的那个信息熵最大的那个

因此有学者提出一种基于扩展页表的GPU访存自适应优化方法。该方法可以解决同时存在多种映射、或针对多种应用选取不同映射方式时，内存地址重叠所导致的错误。其核心思想为：通过扩展页表内容，加入任务类型标记，使每一个应用可获得定制化的访存映射方式，可以在保证访存的正确性的同时，实现在GPU运行时候根据应用进行针对性的访存优化。

通体来说，解决思路包括一下步骤

1. 逻辑运算单元ALU接收GPU应用所发送的完成任务所需的指令，注意到其中需要包含任务类型的相关信息；然后根据指令完成相应的操作，并在访问内存时向内存管理单元发起虚拟地址请求。
2. 内存管理单元根据所接受的虚拟地址请求，通过扩展页表将虚拟地址转换为物理地址，然后将物理地址发送至地址转换单元，基于任务类型信息查找GPU应用对应的所属任务类型标识，将所述任务类型标识发送至映射方案配置寄存器，其中所属任务类型标识标记在内存管理单元的扩展页表中。
3. 映射方案配置寄存器根据所述任务类型标识查找对应的最佳访存映射方式，将所述最佳访存映射方式发送至地址转换单元
4. 地址转换单元根据所述最佳访存映射方式，将所述物理地址映射为新的映射后地址，将所述的新的映射后地址发送至内存子系统，基于所述新的映射后地址访问内存子系统。
5. 具体来说
   1. 页表为了兼容不同应用程序在同一个GPU上运行而不产生地址冲突，GPU软件在访问内存时统一使用从0开始的虚拟地址，然后再通过页表转换为物理地址。其中，物理内存以通常以4KB的大小分为若干页，然后通过页表将虚拟地址上4KB的地址空间映射到物理地址上的某个4KB的地址空间。
   2. 同时，程序在GPU启动时，将获得一个任务类型标识。内存管理单元需要根据任务类型信息查询出访存请求所对应的任务类型标识，并且记录于分配给该程序的页表项中，从而区分不同类型的任务的映射方式。于是在映射为物理地址的同时，任务类型标识也被传入地址伪随机交织模块，从而将原始访存模式映射为该应用的最佳访存模式。
   3. 这里通过利用不同的访存映射方式，将GPU访问请求进行不同的内存地址映射，然后记录GPU最大等待队列的长度作为一个时间段，基于这个时间段确定多个统计时间段，在每个时间段中统计GPU访问请求在不同访问通道上的分布信息熵，然后取得平均值，从而确定某种映射后GPU访问请求对应的信息熵，最后将最大信息熵对应的访存映射方式作为最佳访存映射方式。这里信息熵的衡量了一组消息携带信息的量：E = sum(-pi\*log2(pi))，即消息传输中每种可能的消息i的概率pi的负对数之和。