听《Software-hardware co-design for data-dependent parallelism and register coalescing in GPUs》有感

2019年4月29日，学校邀请了Daniel Wong教授关于《GPU中数据并行同步与寄存器合并的软硬件协同设计》开展技术报告。

Daniel Wong教授是加利福尼亚大学电气与计算机工程学院的助理教授。他的研究领域覆盖了计算机架构设计、高性能计算、数据中心计算等等。他关于数据中心节能相关的研究在2012年在电气电子工程师学会计算机架构领域被认定为IEEE Micro Top Picks。他目前工作专注于如何能够提升计算机的性能、开发效率以及基于加速器的计算机系统。

报告的开始首先介绍了GPU的相关信息。GPU目前是普遍的设备，不仅仅用在超级计算机或者是云计算中心中，他存在于我们身边各种各样的设备中，包括手机以及各种嵌入式设备。随着这几年的飞速发展，GPU不断发展成了一个通用的计算设备。在过去的十年里，GPU的性能和规模都在持续增长。执行单元的数量一直在稳步增加，从而增加了保持这些单元的使用所需的并发线程上下文的数量。为了支持大型活动线程组之间的快速上下文切换，GPU投资于大型寄存器文件以允许每个线程维护其上下文。这种设计允许在执行线程组之间进行细粒度的切换，这是隐藏数据访问延迟所必需的。例如，Nvidia VoltaGPU有80个流式多处理器，每个处理器有256KB的寄存器文件（64K寄存器，每个32位宽），总共有20MB的寄存器文件空间。由于它的连续存取，寄存器文件是维持性能的关键结构。寄存器文件是模具上最大的SRAM结构，也是GPU上最耗电的组件之一。2013年，据估计，寄存器文件占GPU芯片总功耗的18%，这一百分比可能有所增加。随着射频尺寸的不断增大。本文通过引入寄存器合并1来提高GPU寄存器文件的性能和能源效率。类似于将连续内存访问合并为单个内存的内存合并

即便如此，软件与硬件之间仍然存在着一些不高效性。报告的主题围绕着如何联合设计软件与硬件，从而能够消除这些不高效性，并提出了两个重要的挑战。

首先，GPU的设计缺少了从数据相关并行与同步的支持，这就导致了算法需要重新设计并映射到有限的GPU中已经存在的且有限的同步元语中。针对此问题，报告中提到了Wireframe，其能够使应用自然地表达线程之间的依赖关系，并将这些依赖关系构建成一个依赖关系图。在硬件的实时运行中，依赖关系图可以直接用于调度之中，保证了数据之间并行的顺利执行。

第二，GPU寄存器文件在存储、移动窄带数据时十分低效。针对此问题，报告中提到了CORF，一个编译器辅助的合并操作数寄存器文件，其能够通过组合多寄存器的读取操作组合成一个指令，并变成单次的物理读取。他本质上重设计了寄存器文件框架，使得物理上不同的寄存器能够同时地进行数据的读取。同时，他需要利用编译器的帮助，通过编译器的提示，达到最优的组合效率。

CORF的设计本质上提供了一种新的设计思路，打破了常规物理上独立的寄存器独立工作的限制，让我受益匪浅。