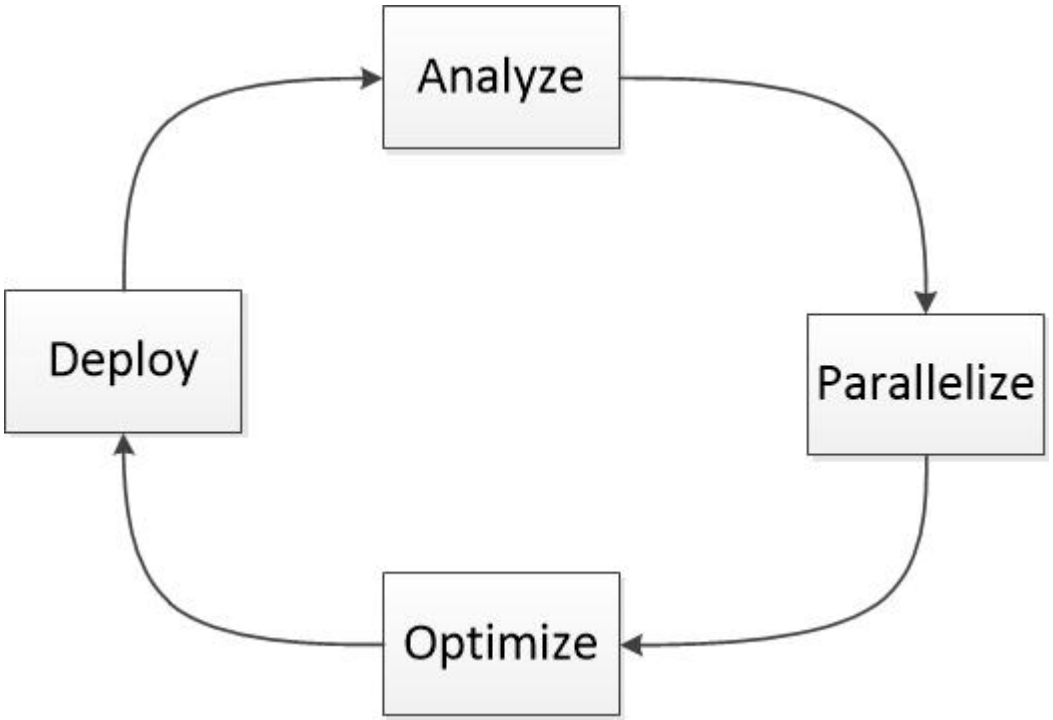
GPU并行平台程序优化性能评估学习报告

报告人：黄一智
指导老师：李仁发教授、刘彦老师
2017-06-15

背景提出

- 性能评估是程序优化的一个关键步骤，既是程序优化的论证(为什么好)也是进一步优化(还能不能更好)的依据（例APOD方法）。



- 性能评估的关键问题，测试方法（怎么测试是合理的）以及测试内容（哪些内容是有效的）

大纲

- 性能测量方法论文阅读

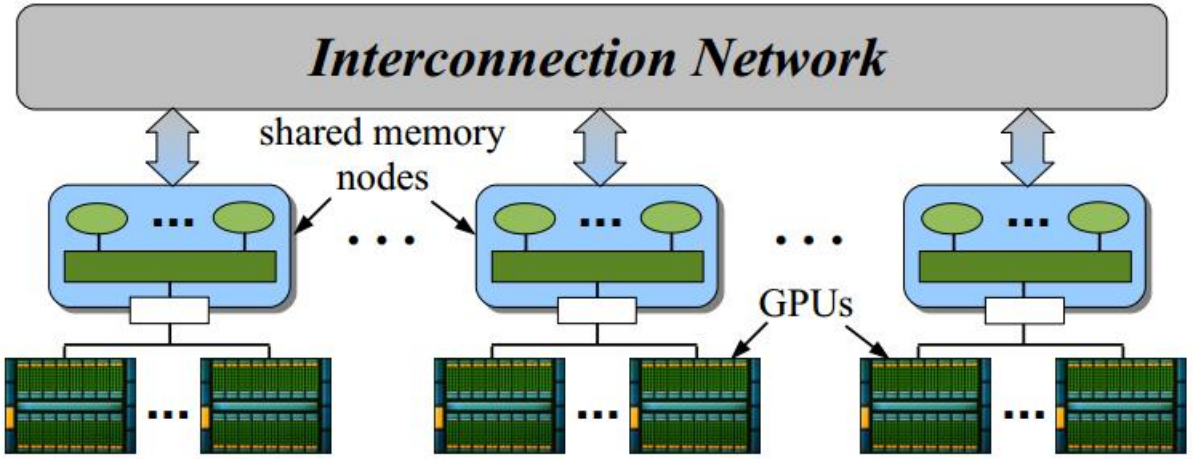
- Merge-CSR论文相关性能测试实验

Parallel Performance Measurement of Heterogeneous Parallel Systems with GPUs

- 论文来源：International Conference on Parallel Processing(ICPP'11, CCF B类)
- 论文作者：University of Tennessee, Knoxville, Innovative Computing Laboratory (ICL) ; Technische Universitat Dresden, Center for Information Services and High Performance Computing (ZIH), Germany ; NVIDIA Corporation, Santa Clara, CA ;
- 论文的主要贡献：在GPGPU计算发展的早期，提出了GPU异构系统的性能测试方法，并依据方法给出了3种性能测试工具：PAPI,Vampir,TAU
- 关注论文原因：学习其测试方法。

目标平台

- 多节点
- 多核处理器
- 多GPU



- 三种交互类型：

1、节点级：进程间通信发生节点间交互（消息传递、全局地址空间）

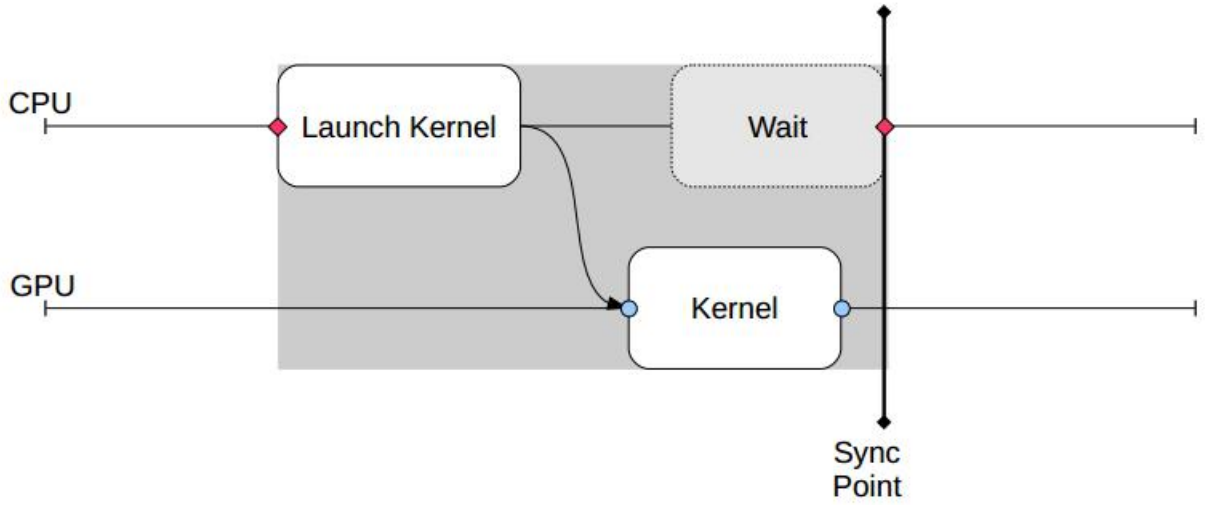
2、节点内线程级：节点内多核CPU的线程执行（涉及共享内存和多线程运行时系统）

3、节点内CPU与GPU：涉及Host和Device间传输、GPU的启动

评估目标的关注点：

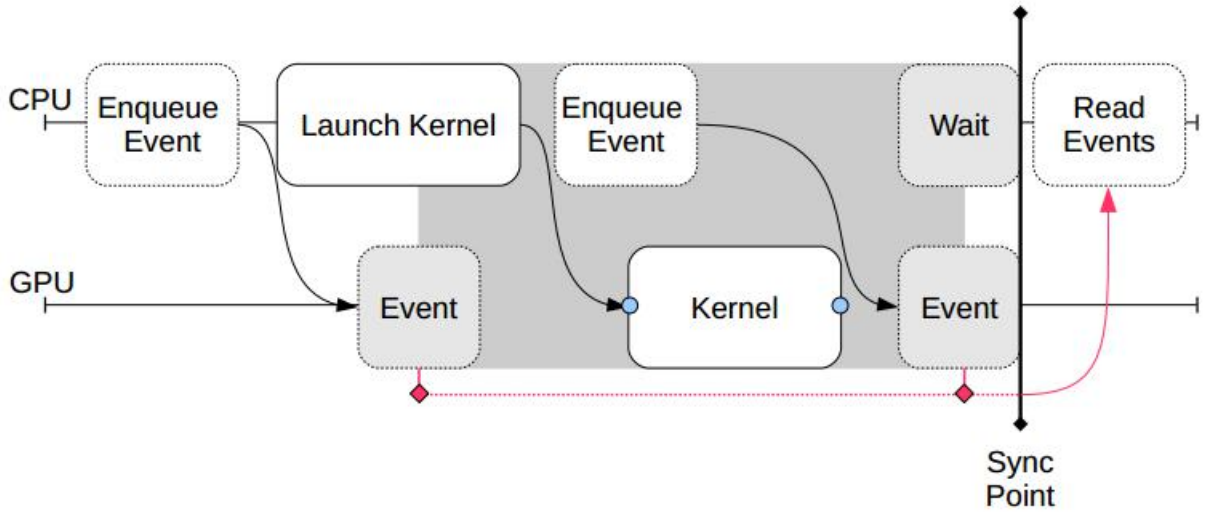
- GPU内核执行
- Host-Device的交互：GPU内核的配置与启动的度量、内存传输时间、并发重叠、主机的异步等待开销
- 节点内线程执行
- 节点间通信

性能度量方法一：Synchronous method



- CPU开始配置内核之前到CPU和GPU同步结束。
- 同步法假设了GPU内核在配置完成后立即执行（对于内核执行的弹性模式下不适用）

性能度量方法二：Event queue method

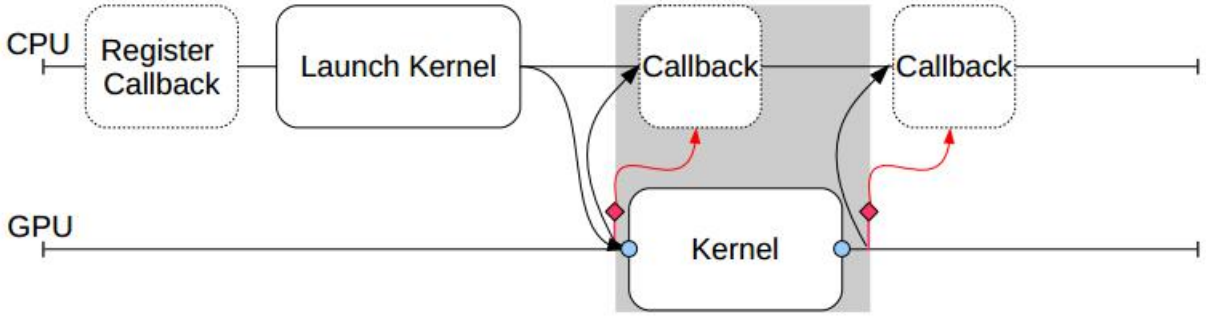


- 事件是特殊内核，在计算核函数发生的前后立即注入。
- CUDA和Opencl典型方法(cudaEvent_t)
- 问题：

1、完全依赖设备制造商是否提供

2、需要Host端显示参与，设置和读取事件。

性能度量方法三：Callback method



- 尽早注册回调，在核函数开始前和结束后调用回调
- 优势：更及时，避免配置前的代码修改。
- 回调方法依赖于设备制造商在设备层乃至GPU硬件上提供支持

Merge-csr论文实验性能测试

- The International Conference for High Performance Computing, Networking, Storage and Analysis文中没有数学证明线程处理item的绝对平衡性

- 性能测试练习并希望能看到改进的余地

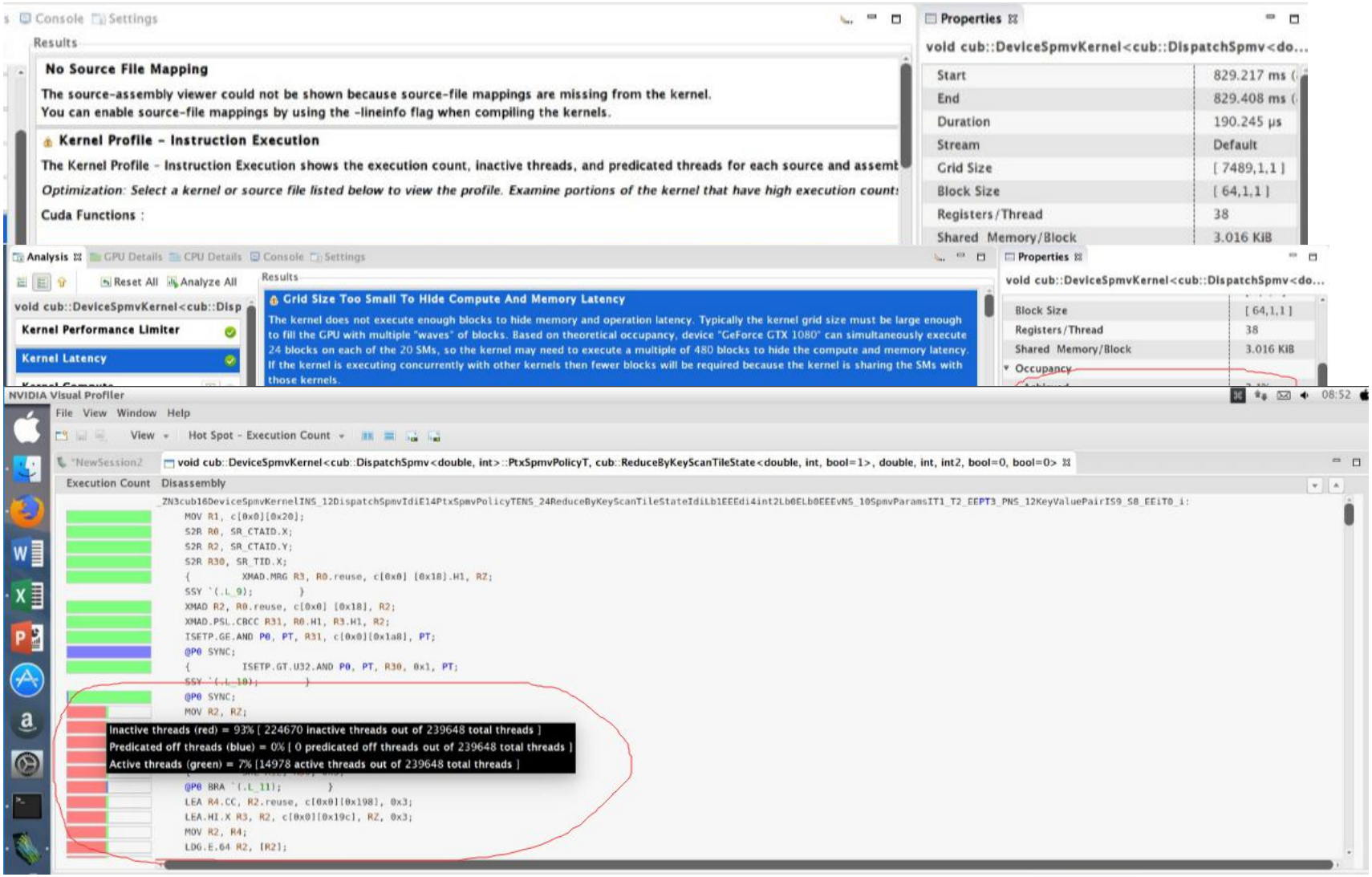
实验环境

- CPU：Intel Core 4770K @ 3.5GHZ 4C8T
- 内存：1333MHz DDR3 32GB
- GPU：Nvidia Geforce GTX1080 @ 1.73GHz
- CUDA core: 2560
- 显存： 5000MHz GDDR5X 8GB
- 数据集：University of Florida Sparse Matrix Collection，选取了4765个矩阵。
- 测试方法： Event queue method
- 评价指标：吞吐量和实际显存带宽

论文实验测试结果

- 确实如论文所述，性能不会随row_length_variation的改变而发生波动。
- 问题一：输入数据量较小时，性能是cuSPARSE库的一半。
- 问题二：输入数据量较大时（带宽343/360 = 95.3%），峰值性能仅1/10。

Nvvp(NVIDIA Visual Profiler)工具测试



后续工作

- 总结并绘制CPU-GPU平台的结构
- 根据merge-csr源码和平台结构总结指令流向和数据流向，试图分析warp分歧的原因，并提供解决方法。