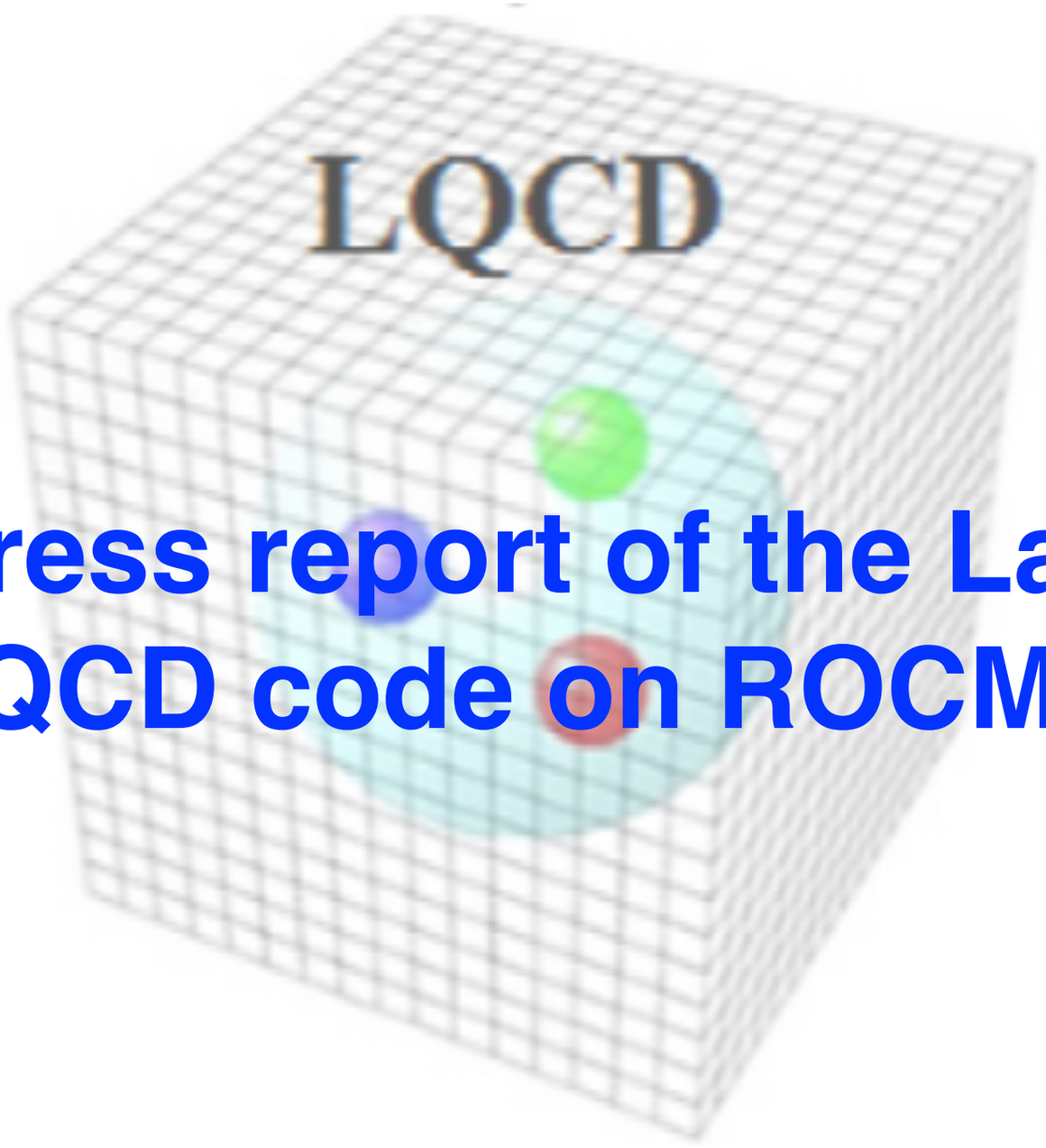


Progress report of the Lattice QCD code on ROCM

杨一玻
（代表开发团队）

计算机实现

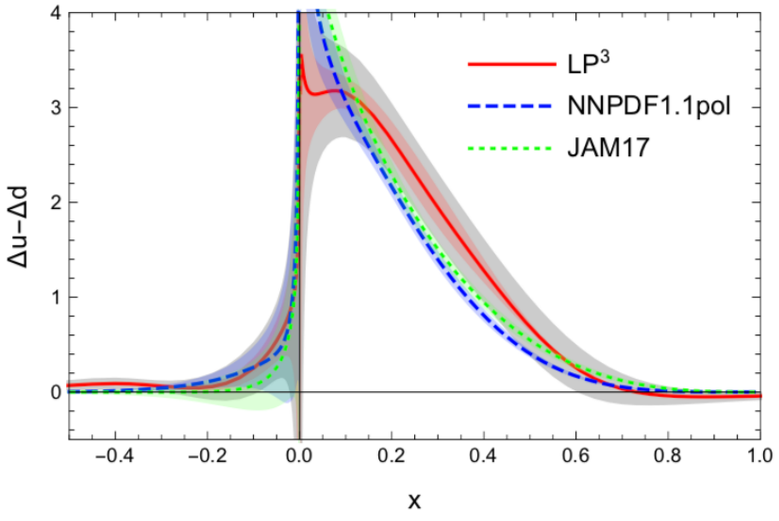
移植中的CUDA软件包

- **Chroma+QUDA**：
- JLab+Nvidia牵头带领社群开发
- QUDA实现GPU上的基本计算单元，Chroma提供CPU上的基本计算单元和整合的应用界面
- 支持MPI+OpenMP+CUDA, 处理产生组态以及clover相关问题

- **GWU package**：
- χQCD合作组开发
- 通过SSE实现CPU上的基本计算单元，通过thrust库提供对GPU的支持
- 支持MPI+OpenMP+CUDA，处理overlap相关问题

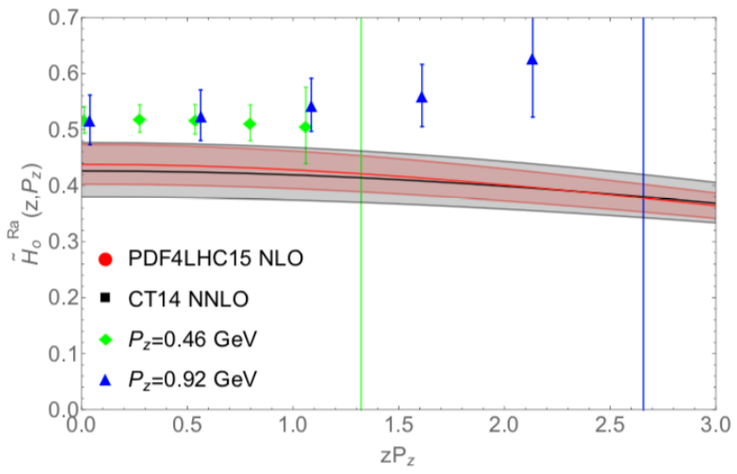
应用范例

上下夸克极化分布函数的差别



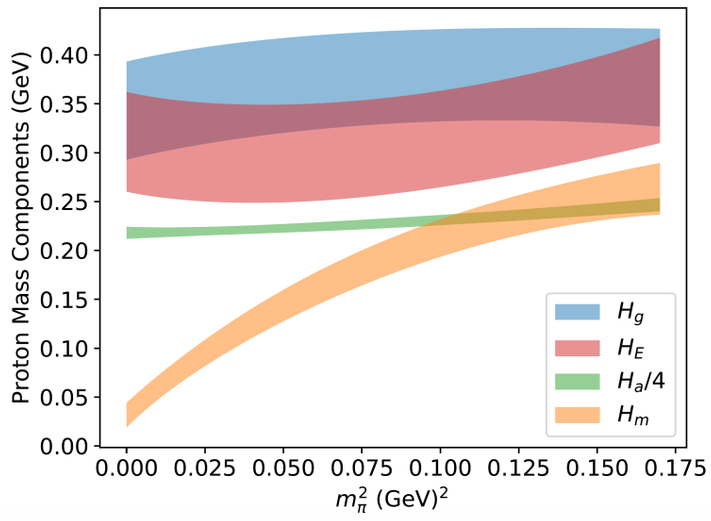
H. Lin, et al, LP³ Collaboration, PRL121(2018)242003

胶子非极化分布函数的傅立叶变换



Z. Fan, YBY, et al, PRL121(2018)242001

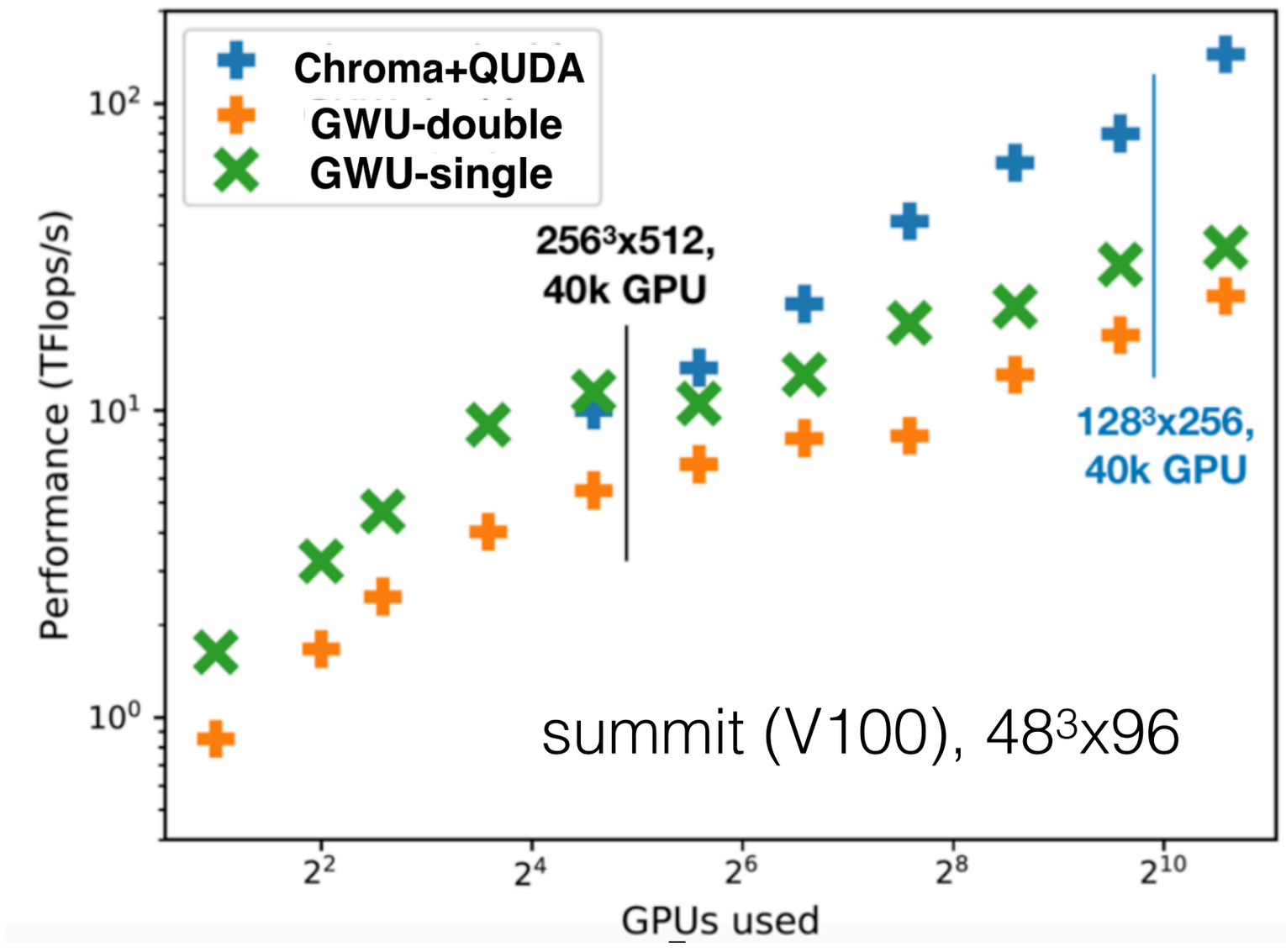
质子质量分解



YBY, J. Liang, et al, χQCD Collaboration, PRL121(2018)212001, Viewpoint and Editor's suggestion

计算机实现

强scaling: 85%



- **强scaling:** (问题总规模固定) 规模每增加一倍性能提高85%，1536GPU时大约理论峰值1%；
- **假设有良好的弱scaling** (单节点上的问题规模固定，性能完全正比于问题规模):
  - 使用128³x256的格子有望在40k GPU时达到**单任务**峰值性能**1%**。
  - 使用256³x512的格子有望在40k GPU时达到**单任务**峰值性能**3%**。
- 小规模多任务时可以达到峰值性能10%以上。

ROCm上的格点QCD

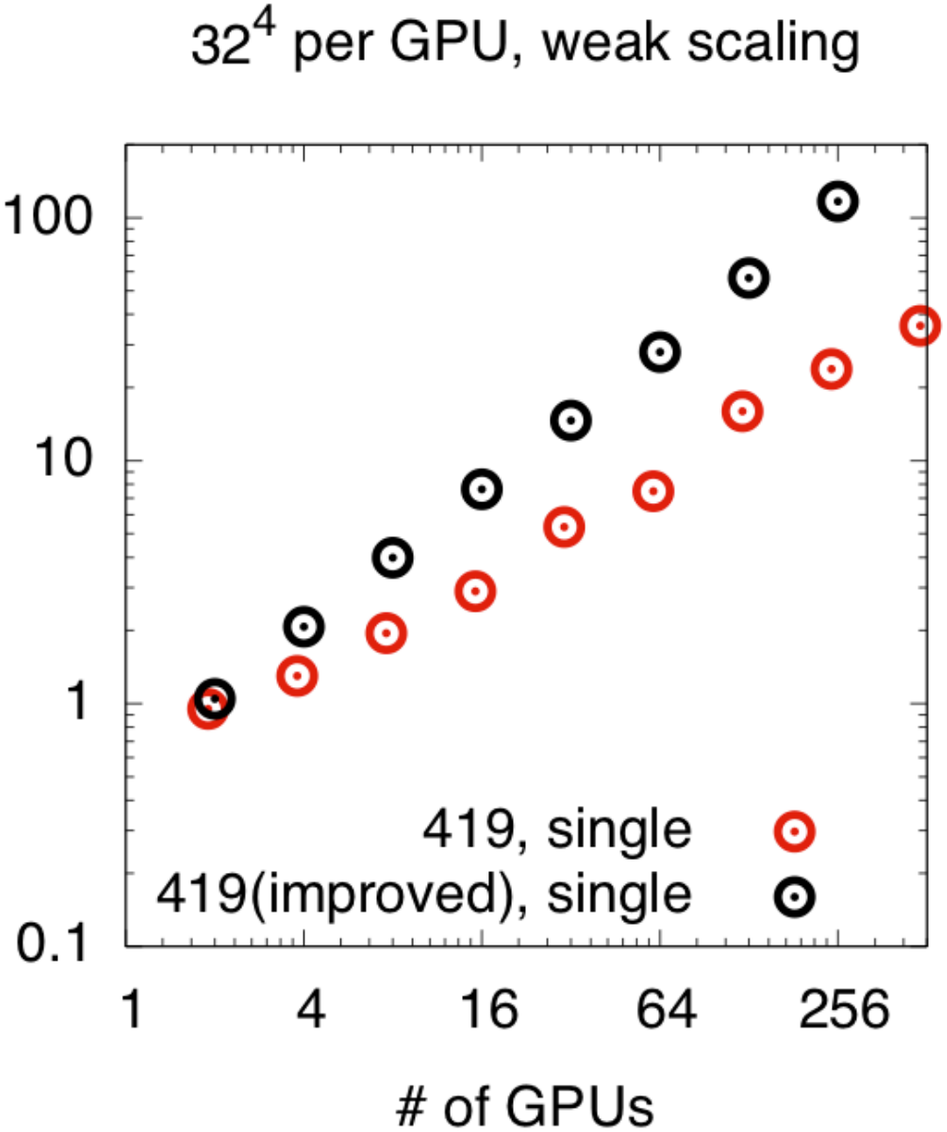
GWU package移植完成

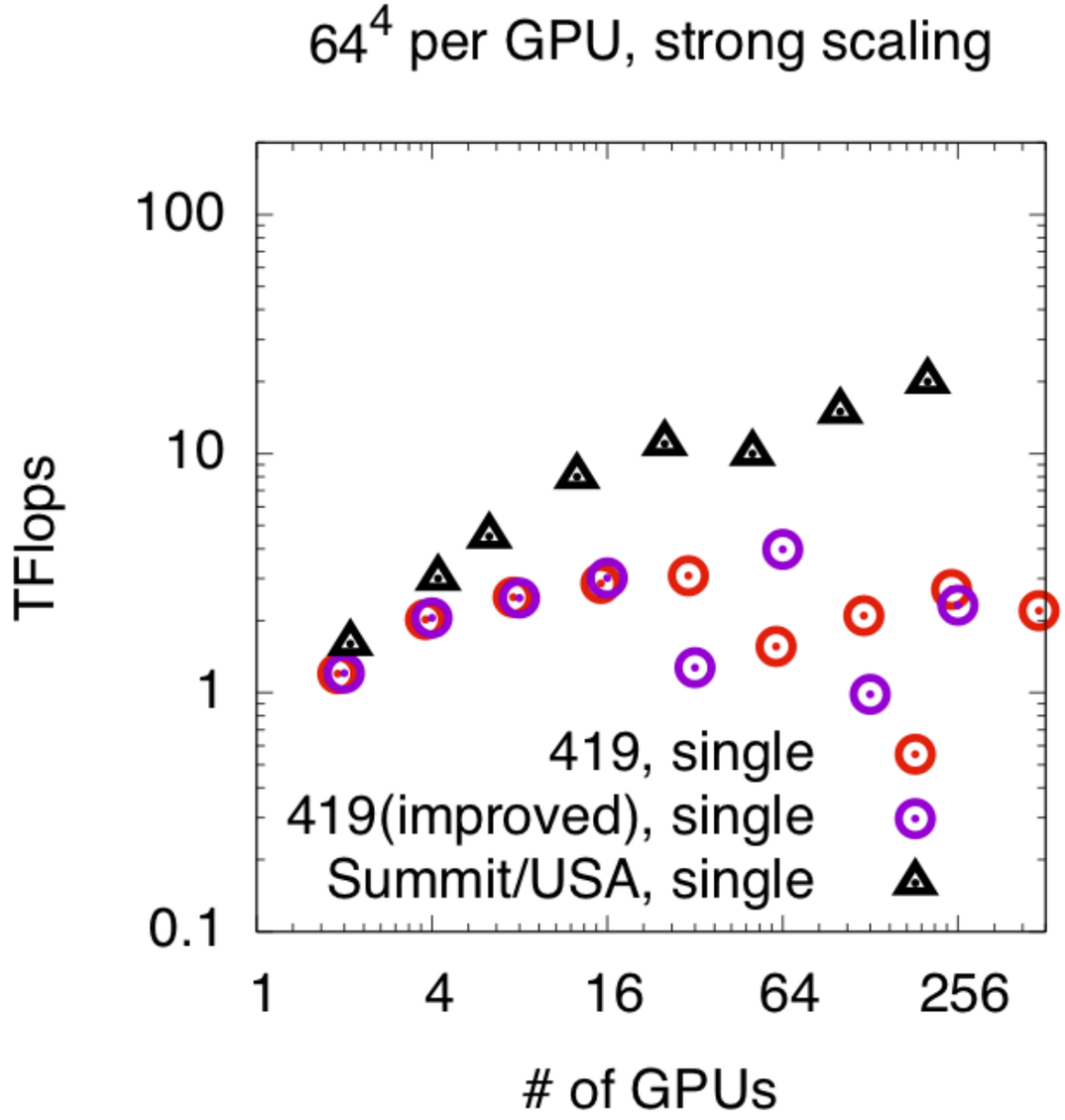
- **原始代码基于CUDA和第三方CUDA库Thrust**
- **约10万行代码**

- 转码
  - 使用Hipify转码，并使用HIP版Thrust库
- 编译
  - 手动实现导致编译器崩溃的OpenMP reduction宏
- 链接
  - 使用GNU版OpenMP编译，Intel版OpenMP链接
- 单节点测试
  - 绕过各种bug，确认**单DCU稀疏矩阵操作性能峰值约540GFlops，单节点4GPU性能峰值约2.0TFlops。**
  - 相当于CUDA版本约70%的性能。
  - 强scaling增长幅度约85%，和CUDA版本相当
- **大规模应用**
  - **进行48^3x96及以上规模的强scaling测试，与summit结果对比分析**

DCU上的格点QCD

GWU package大规模性能测试

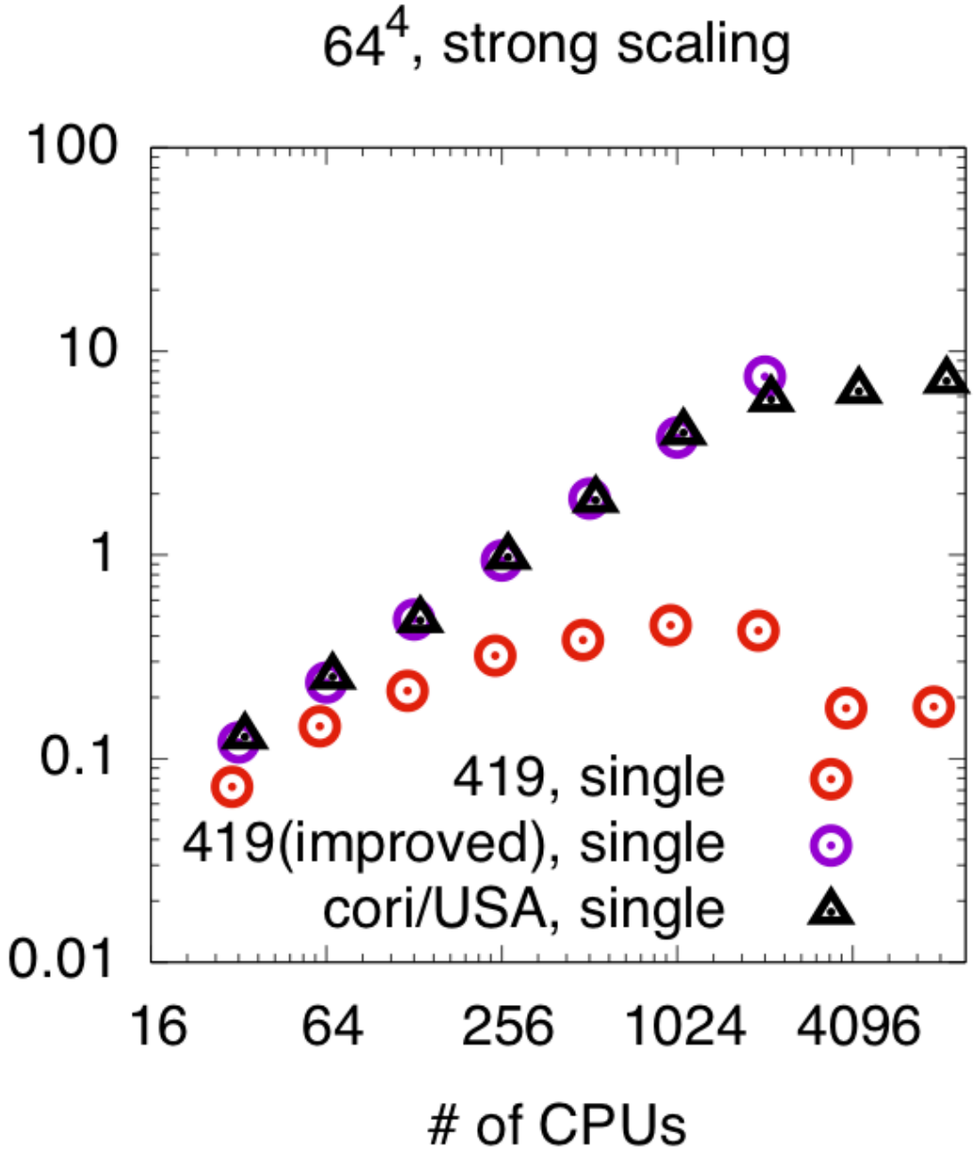


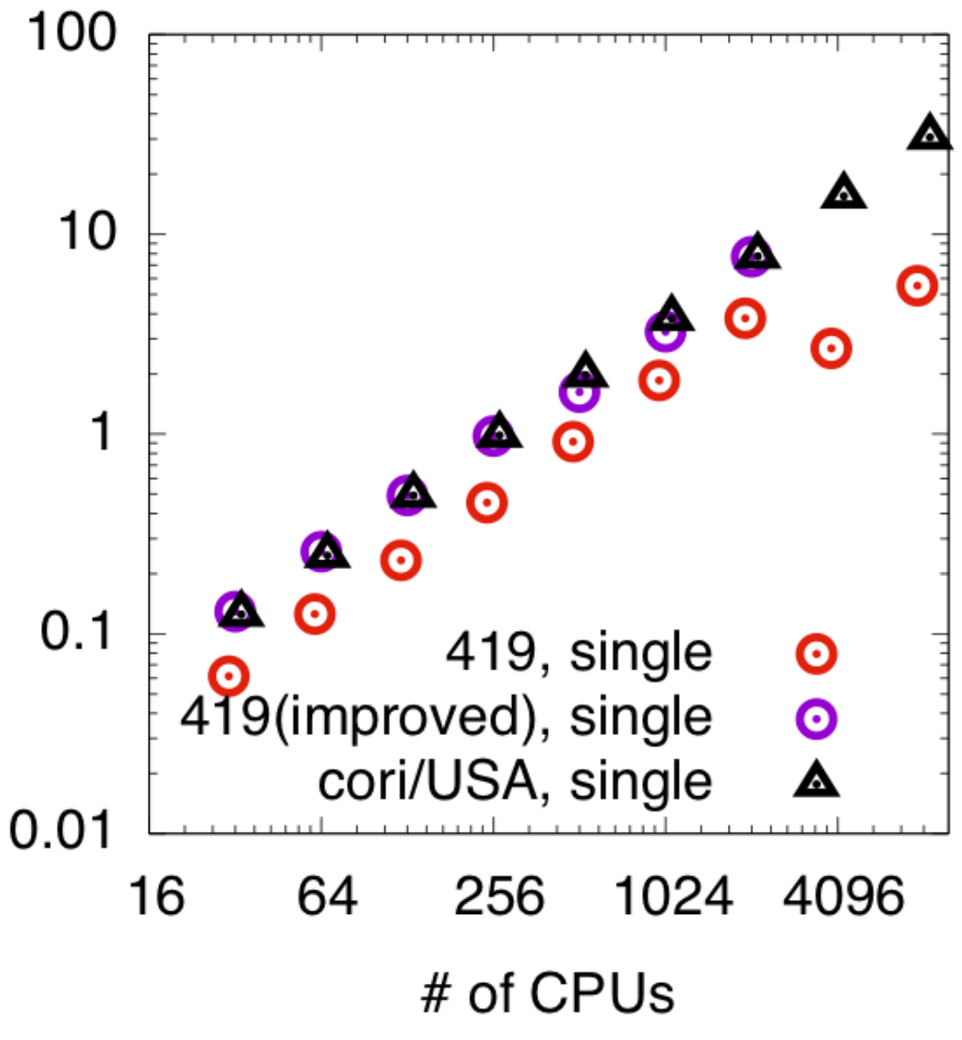


- 剔掉部分节点并改变测试方式之后weak scaling令人满意；
- 对strong scaling影响不大，与Summit的性能相比有差距。

DCU上的格点QCD

GWU package网络互联性能测试





- 剔掉部分节点，并且每节点只用30核(而不是用满32核)时强弱scaling性能都和美国Cray机器相当；
- 到2048核为止，更大规模需要发展系统地剔除节点的策略；
- GPU强scaling下降可能主要是单卡规模太小以及通信性能的问题。

ROCm上的格点QCD

GWU package需要解决的问题

- 稀疏矩阵操作反复迭代时随机出错
- 程序使用三个流，流1负责格子内部计算，流2收集和分发通信数据，默认流负责PCIE异步拷贝
  - 只使用一个流时计算不会出错
  - PCIE拷贝使用同步拷贝时不会出错
  - 加上HSA_ENABLE_SDMA=0和HIP_MEMCPY2D_FORCE_SDMA=0也不能保证迭代不出错，尽管可以降低出错概率
- 需要在单GPU上计算32^4的格子才能达到最大性能，是否能减弱这一限制?
- 其他帮助单GPU性能和强scaling性能提升的优化可能性?

ROCm上的格点QCD

移植Chroma+QUDA的挑战

- **原始代码基于CUDA和第三方库Eigen、cuBLAS、Thrust和cub等**
- **约22万行代码**

- 整体代码结构(基于最新发布的v1.0)
  - include目录: 83个头文件，约27,000行
  - include/kernels目录: 绝大部分的核函数实现，35个cuh文件，约9,000行
  - include/external目录: 约100个复制自各种库的文件，约52,000行
  - lib目录: 197个.cpp/.cu文件，cpu代码和调用核函数的代码，约75,000行
  - tests目录: 34个test程序和28个辅助.h/.cpp文件，约60,000行
  - cmake目录: cmake脚本文件，约500行
  - CMakeLists.txt文件：cmake脚本主文件，约900行

- 依赖外部CPU库qmp和qio完成MPI通行和IO
- 依赖外部CPU/GPU库eigen完成部分稠密矩阵操作
- 依赖外部CPU/GPU库cuBLAS批量计算3x3矩阵的逆

ROCm上的格点QCD

移植Chroma+QUDA的挑战

- **原始代码基于CUDA和第三方库Eigen、cuBLAS、Thrust和cub等**
- **约22万行代码**

毕玉江、徐顺、宫明、肖熠、YBY

- 转码/编译（基于v1.0，**除LU分解以外**已基本完成）
  - 使用Hipify转码QUDA，并手动修改上百个文件，上千行代码

1. 传递参数时需要显示转换传递kernel函数类型为void *(void **)类型，并显示指明kernel参数。
2. 修改Eigen库上百处，为代码追加EIGEN_DEVICE_FUNC关键字
3. 把全局__shared__变量挪到函数体内，
4. 修改cuda/hip上相似函数中不一致的参数类型和格式
5. 将内联汇编代码暂时改写为C代码
6. ……

- **multigrid应用中需要重新实现DCU版BLAS库的LU分解和求逆**
- 有L^3xTx4个n x n的矩阵，L和T的取值范围是100以内(最常用的是20到32)，n在10以内(最常用的是3)，然后先调用cublasCgetrfBatched批量计算这些矩阵的LU分解，然后调用cublasCgetriBatched批量计算这些矩阵的逆。需要实现的就是cublasCgetrfBatched和cublasCgetriBatched这两个函数。

ROCm上的格点QCD

移植Chroma+QUDA的挑战

- **原始代码基于CUDA和第三方库Eigen、cuBLAS、Thrust和cub等**
- **约22万行代码**

毕玉江、徐顺、宫明、肖熠、YBY

- 链接（已完成）
  - 19年3月上旬：实现了以动态库方式链接，以及chroma以动态库调用QUDA
  - 19年3月下旬：怀疑DCU对texture支持有限，重新移植无texture版本
  - 19年4月上旬：运行时找不到大量__global__函数，发现动态链接版本没有成功编译和链接到所有的函数；静态链接可以在链接时发现这些问题。
  - 19年4月下旬：测试环境换成新编译器，重新修改各种有兼容性问题的编译器flag，修改devtool版本混乱的问题，最终链接完成，准备开始调试
  - 19年5月上旬：测试环境再次更新，程序在加载时崩溃。重新修改各种有兼容性问题的编译器flag后，重新编译问题依旧。
  - 19年5月下旬：再次重新移植最终开发版的quda，程序可以正常启动运行，动态库方式链接也可以正常工作，开始调试。

ROCm上的格点QCD

移植Chroma+QUDA的挑战

- **原始代码基于CUDA和第三方库Eigen、cuBLAS、Thrust和cub等**
- **约22万行代码**

孙鹏、肖熠、YBY

- **单卡调试（基本操作）**
- **结构体作为global函数参数直接拷贝到寄存器的问题**
- 将超过约2000字节的结构体作为global函数参数，从cpu拷贝到寄存器后，部分线程读取的结果错误
- 目前通过将结构体复制到显存，然后在global函数中引用该结构体加以回避
- **autotuning过程中崩溃的问题**
- 部分被重载的函数需要针对特定类型手动展开，原因未知
- 限制tuning的线程数到512而不是1024
- **D-slash稀疏矩阵操作性能低下**
- 性能大约只有GWU code类似操作的10%以下，可能的原因包括频繁访问显存以及结构体实现方式等，未解决

ROCm上的格点QCD

移植Chroma+QUDA的挑战

- **原始代码基于CUDA和第三方库Eigen、cuBLAS、Thrust和cub等**
- **约22万行代码**

- **单卡调试（Inversion）**
  - **通用inverter**
  - BICG可用
  - CG/GCR可看到residual下降，但是在达到目标精度前约一个数量级后无法下降
  - **Multigrid inverter**
  - 生成subspace所需的gpu端并行随机数生成器不可用
  - hipMemcpy到constant内存有bug会崩溃，目前用普通显存替代constant显存
  - cublasCgetrfBatched和cublasCgetriBatched未实现

- **多卡调试**
  - **通用inverter**
  - 通过chroma调用quda进行mpi测试
  - 完全关闭quda对多卡通信的优化（QUDA_ENABLE_P2P=0 ）之后可以正确运行
  - 格子较大时结果会变成nan

孙鹏、肖熠、YBY

ROCm上的格点QCD

移植Chroma+QUDA的挑战

- 原始代码基于CUDA和第三方库Eigen、cuBLAS、Thrust和cub等
- 约22万行代码

- 需要解决的问题
  - D-slash操作性能问题
  - 多卡环境下正确性问题；
  - cublasCgetrfBatched和cublasCgetriBatched的移植，或者暂时使用cpu版blas?
  - multigrid inverter的完整测试
  - 大规模测试
  - 产生组态测试

总结

- **GWU-code已经初步完成移植，需要进一步优化性能：**

- 多节点性能问题

- D-slash操作使用多个流并反复迭代时随机出错的问题（仅使用单个流有近50%的性能损失）

- **QUDA已经初步完成移植，需要进一步调试：**

- D-slash操作性能极端低下；

- 为回避编译器的大量bug，引入了各种不安全的hack，需要深度测试

- multigrid inverter还未能完整跑通

- **感谢参与本项目各单位的各位同事！**