



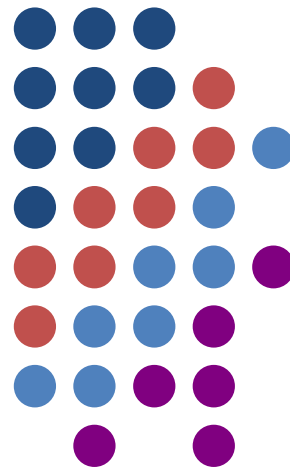
中国科学院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences

四月季度考核报告

学生：肖鹏飞

导师：季筱璐

2024年4月29日



主要内容

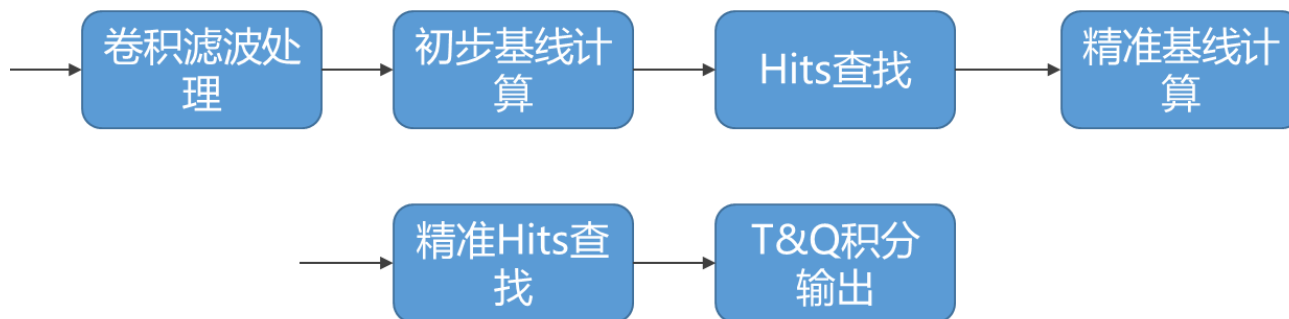


- ◆ 基于CPU的JUNO在线波形重建优化研究
- ◆ 基于CPU/GPU异构的JUNO在线波形重建优化研究
- ◆ 2024RT会议参会
- ◆ 当前计划



基于CPU的JUNO在线波形重建优化研究

- JUNO波形重建算法研究简介

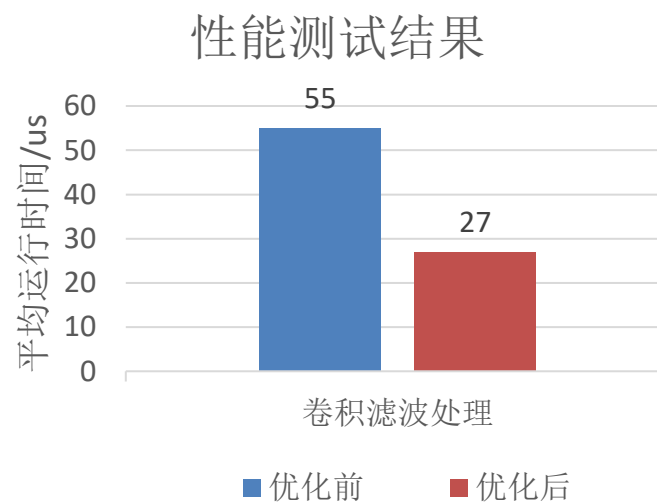
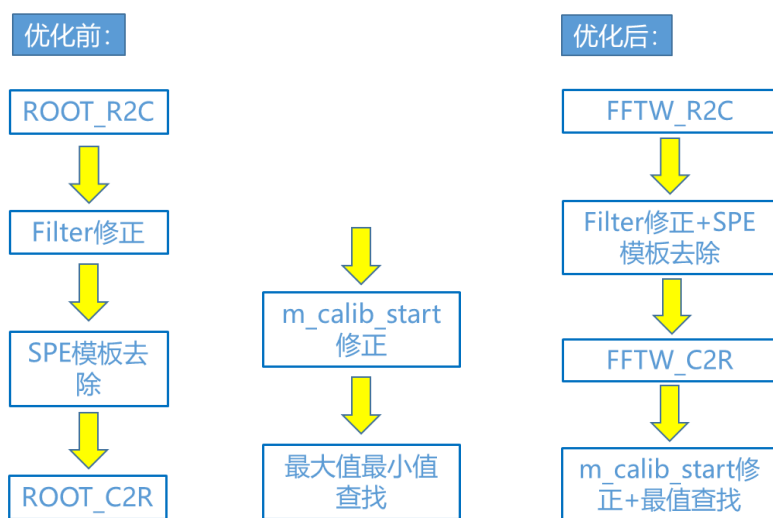


- JUNO的波形重建中，原始波形数据经过卷积滤波处理去噪后进行两次基线计算和扣除，找到目标时域区间积分计算并输出，同时也是高能物理实验中常用的
一种波形数据处理方法

基于CPU的JUNO在线波形重建优化研究



卷积滤波处理优化

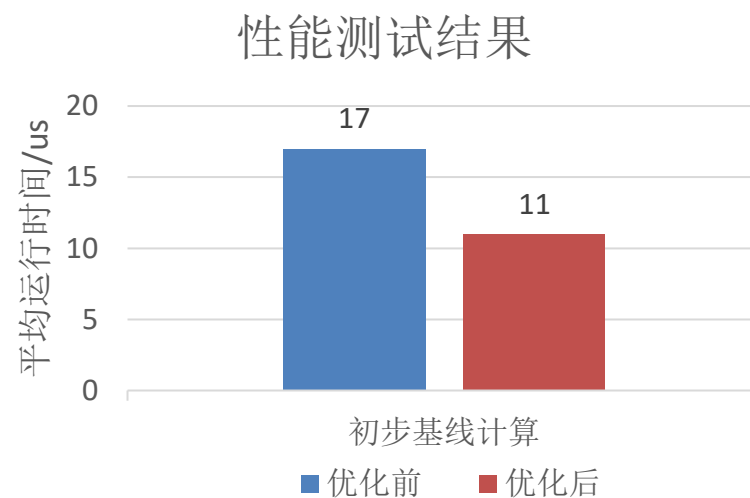
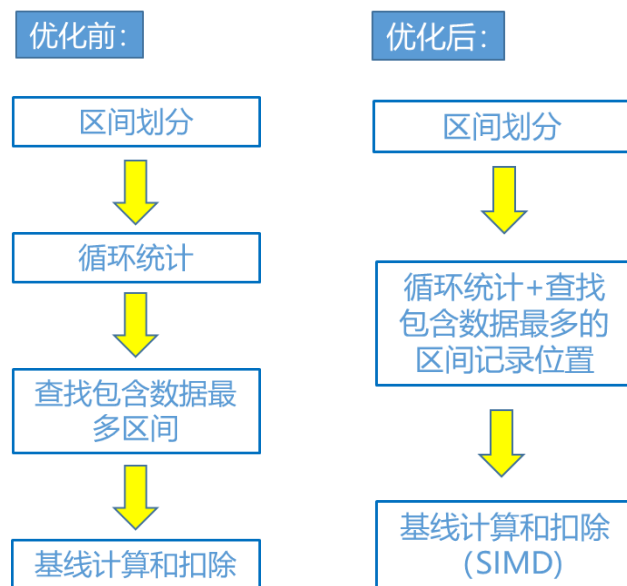


- 用最新的算法库代替ROOT进行原始波形数据的傅里叶变换和逆变换→应用更快算法；并在循环中分别合并波形修正和波形模板去除，刻度起始修正和最大值最小值查找过程→一次访问进行多次处理，避免后续遍历的额外开销



基于CPU的JUNO在线波形重建优化研究

● 初步基线计算优化

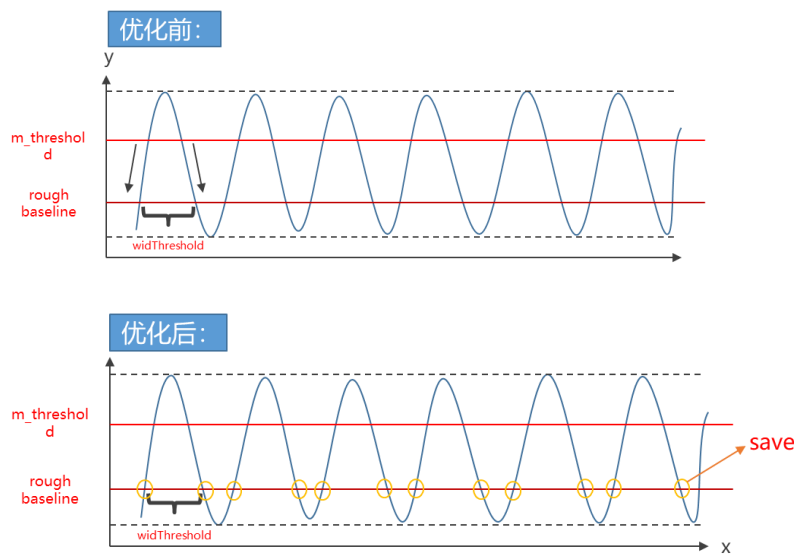


- 在循环中将数据向Y轴投影统计时记录包含数据最多的划分区间位置→避免后续遍历的额外开销；循环统计结束即可找到含有多数区间位置，并采用SIMD技术进行基线扣除→加速迭代过程

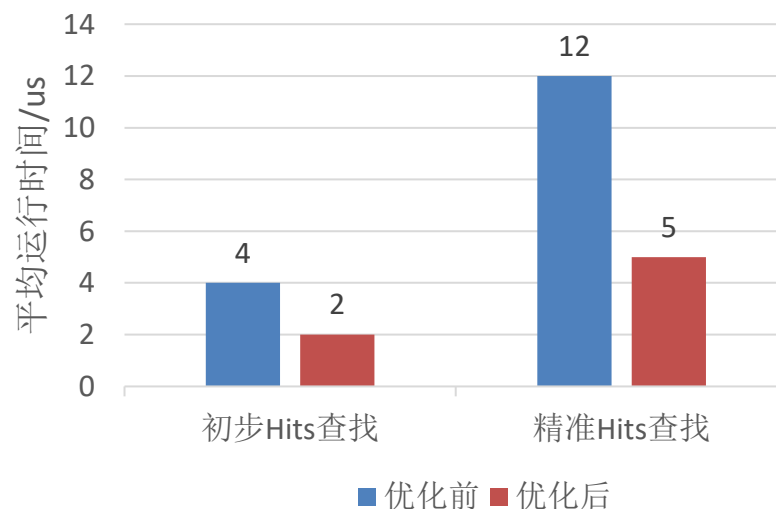
基于CPU的JUNO在线波形重建优化研究



● 初步Hits查找/精准Hits查找优化



性能测试结果

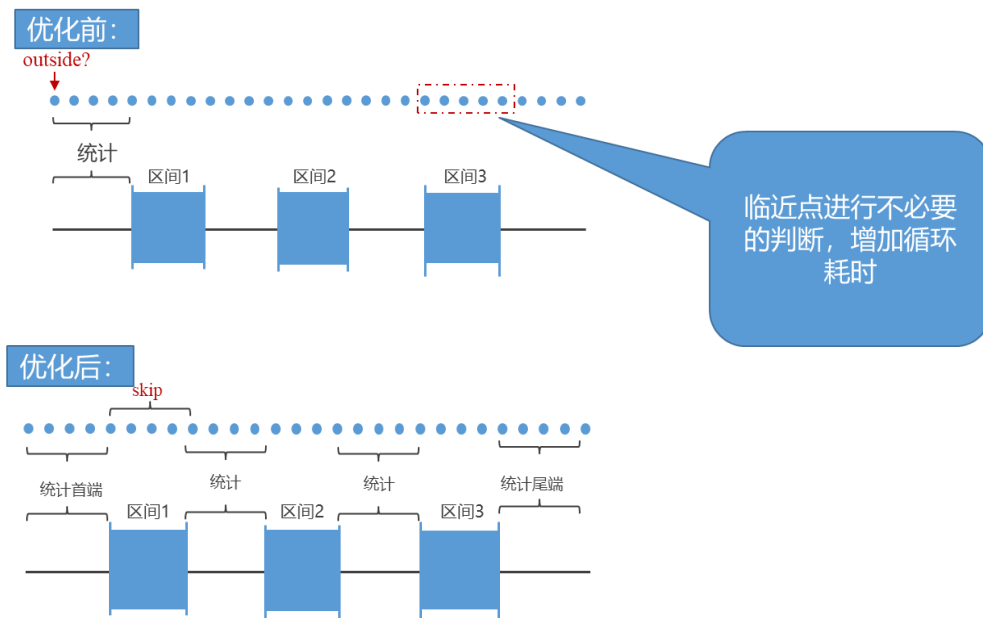


- 将以大于阈值为起始不断向左右查找满足条件区间的策略变为先存储可能的区间，再在中进行筛选→减少重复的判断

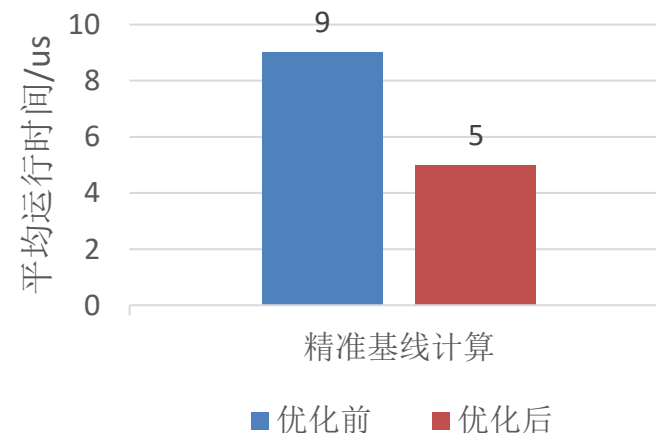


基于CPU的JUNO在线波形重建优化研究

● 精准基线计算优化



性能测试结果



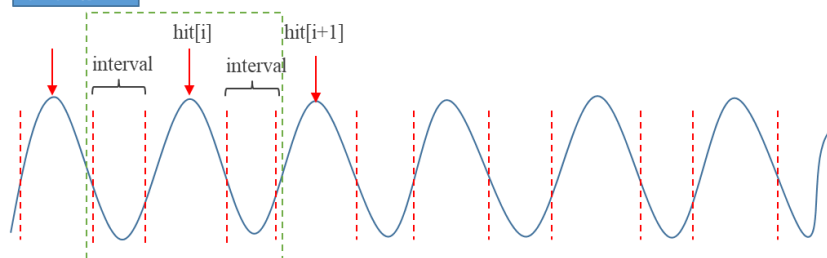
- 利用时间点的空间局部性, 直接跳过邻近点判断, 统计区间外的数据点计算基线, 用SIMD进行基线扣除→减少分支判断从而加速迭代过程

基于CPU的JUNO在线波形重建优化研究

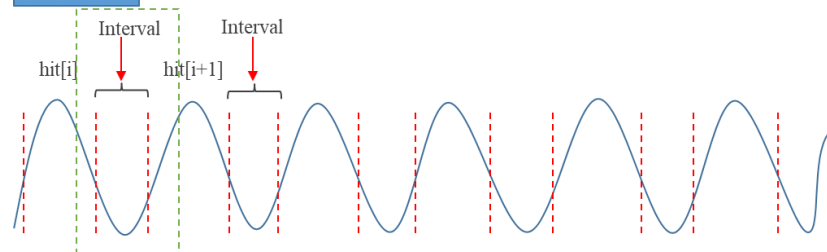


● T&Q积分输出

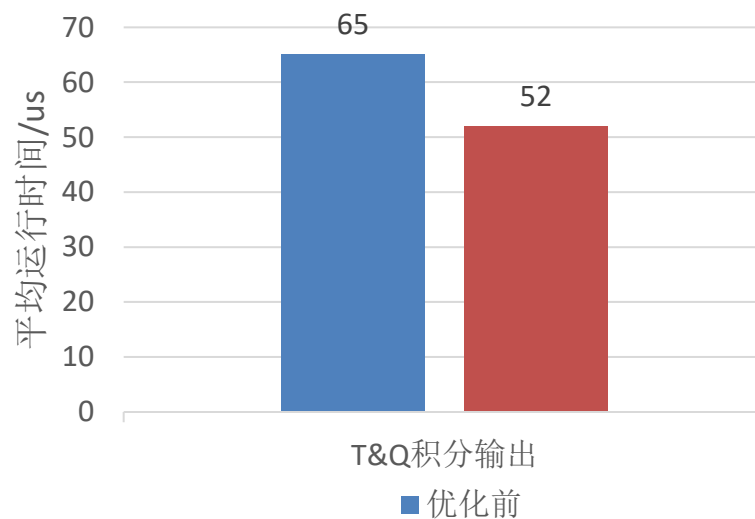
优化前:



优化后:



性能测试结果

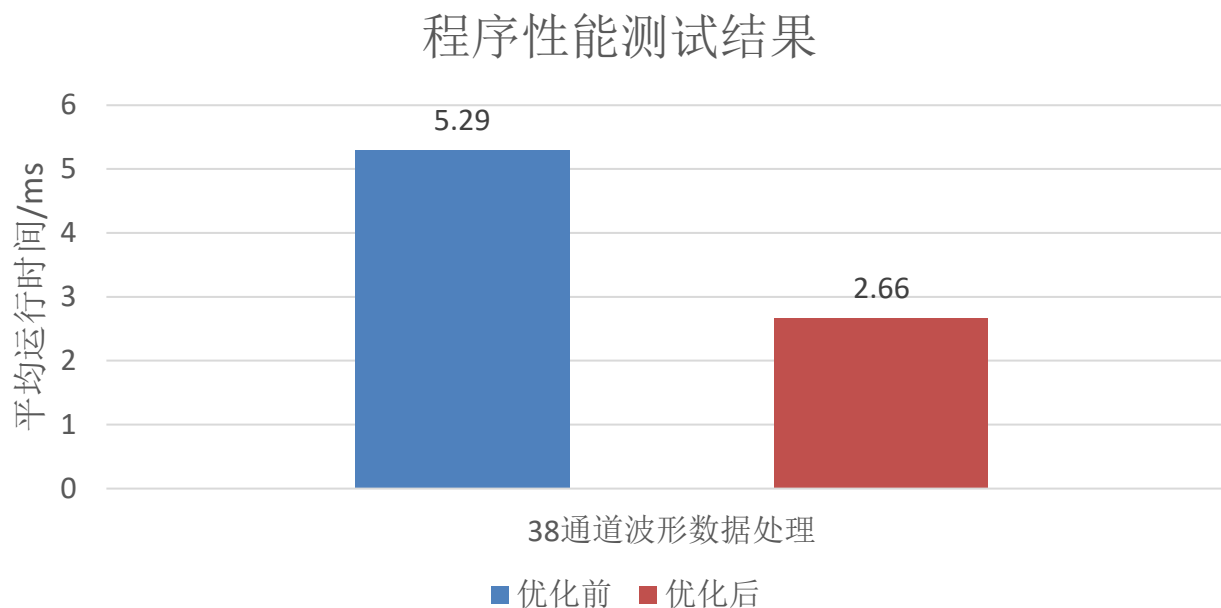


- 将以hit为基准进行遍历判断左右间隔确定积分区间优化为以interval为基准遍历确定左右hit积分区间的边界→避免重复判断；减少if语句使用并在积分计算的同时进行结果输出→减少额外循环输出



基于CPU的JUNO在线波形重建优化研究

- 整体测试结果

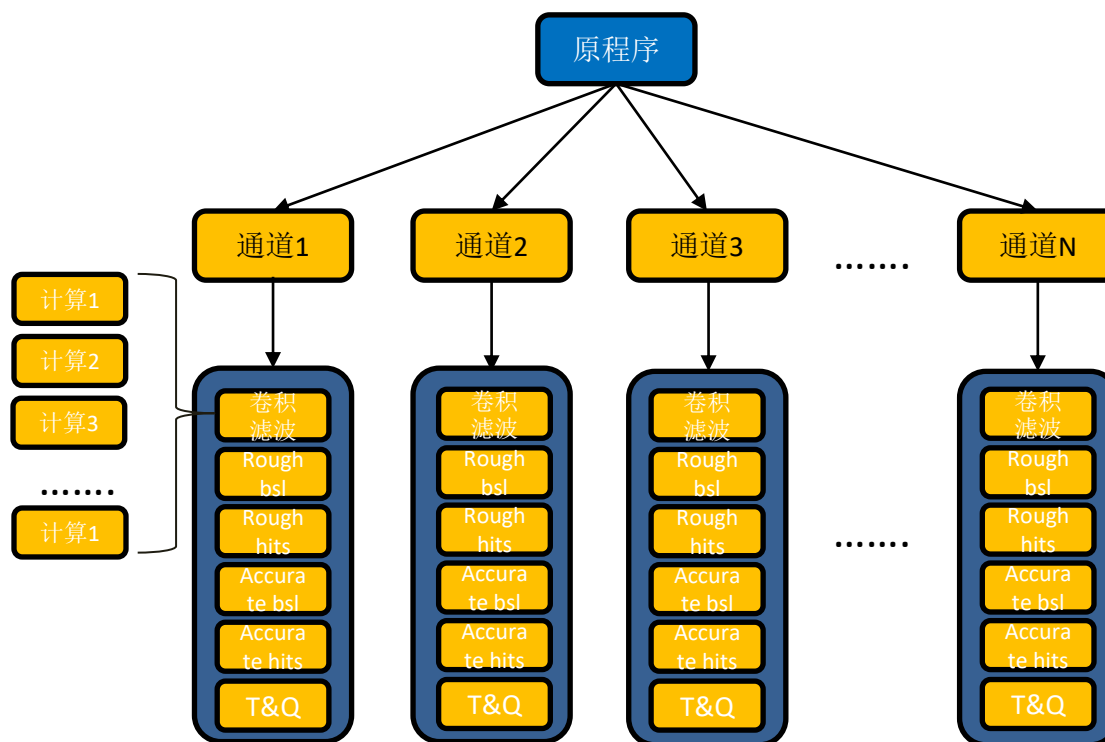


- 在整个优化研究中，采用算法优化，减少循环开销，一次访问多次处理的优化方法，最终使38通道波形数据处理的测试性能提升至198.87%



基于CPU/GPU异构的JUNO在线波形重建优化研究

- 基于GPU的并行设计

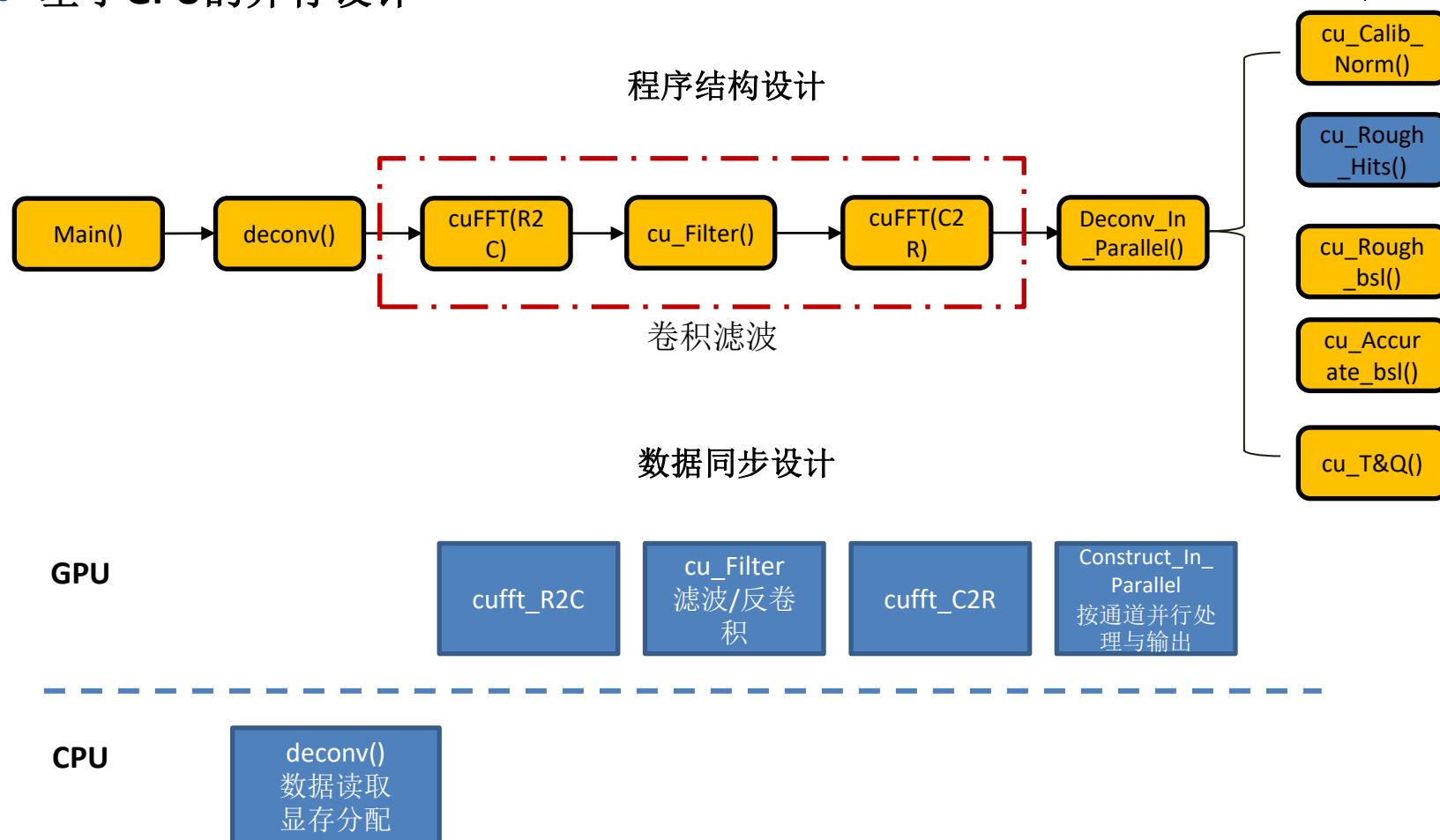


- 利用CUDA进行波形重建代码各部分重写，基于GPU动态并行（核函数递归调用）实现通道级并行及通道内计算并行

基于CPU/GPU异构的JUNO在线波形重建优化研究



● 基于GPU的并行设计



- 在核函数的父网格中利用动态并行的隐式同步实现通道内各子网格并行计算的同步过程

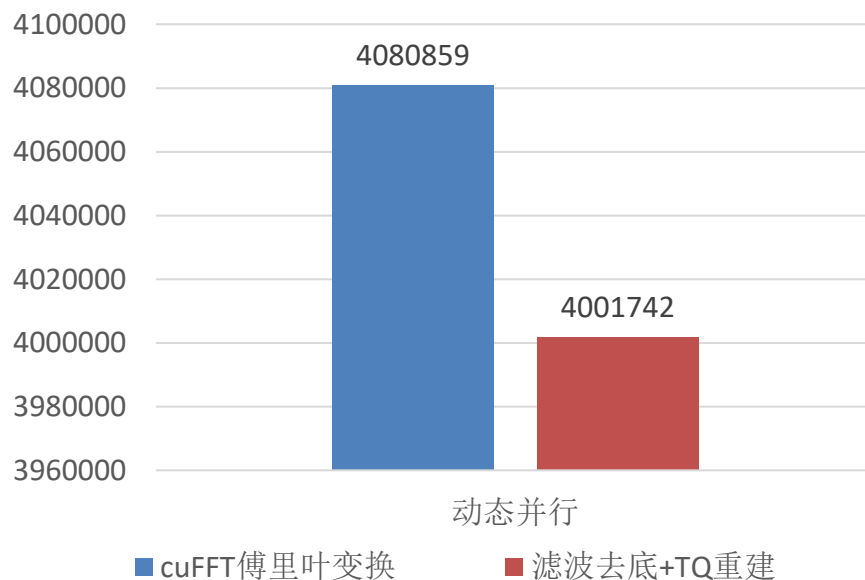


基于CPU/GPU异构的JUNO在线波形重建优化研究

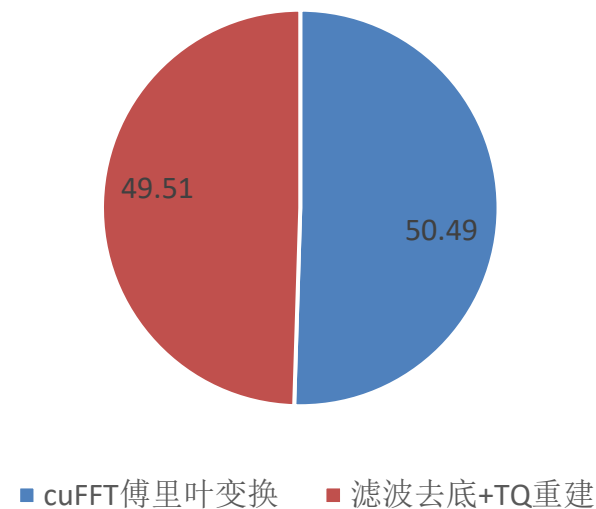
● 瓶颈分析测试

1800通道×1000数据点处理程序性能测试

平均运行时间/us



运行总耗时占比

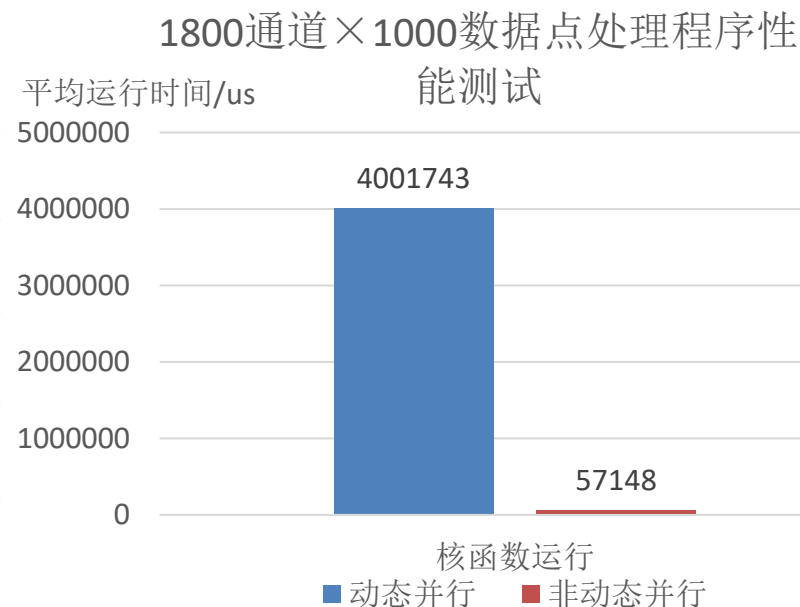
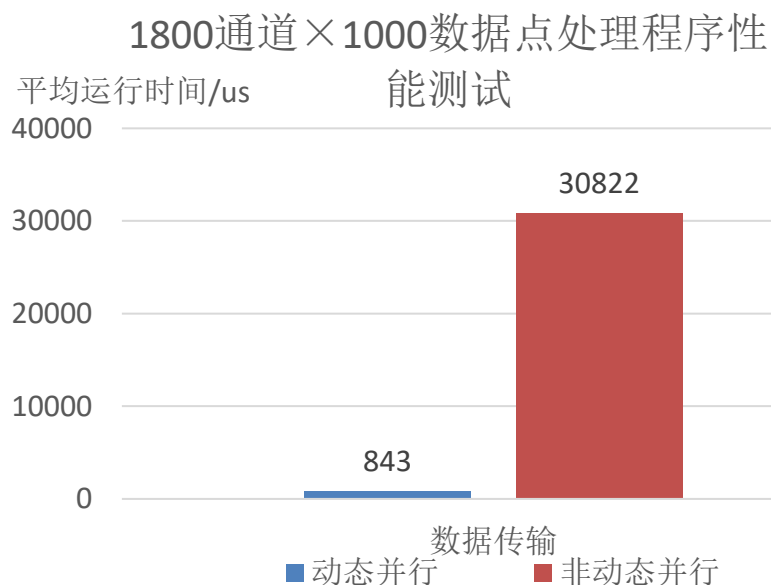


- 动态并行实现方式中的cuFFT库傅里叶变换部分成为程序的瓶颈部分



基于CPU/GPU异构的JUNO在线波形重建优化研究

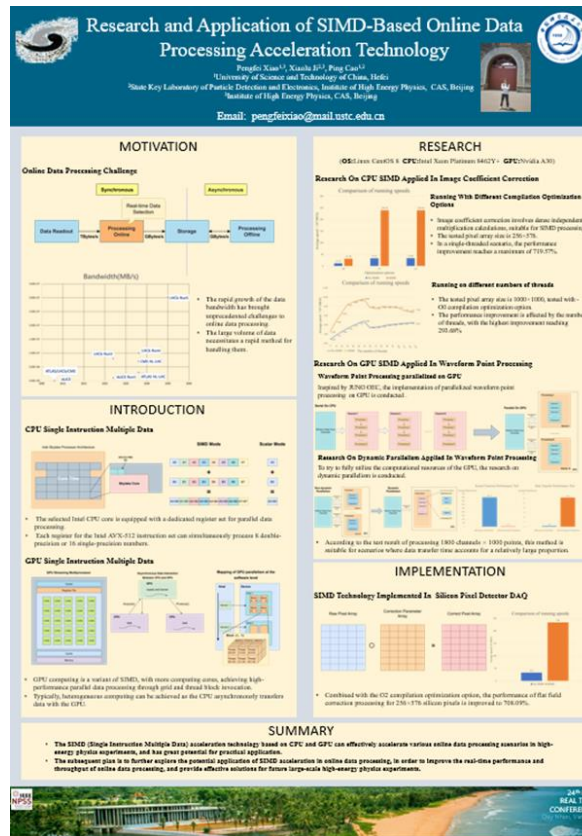
● 动态并行与非动态并行优势分析



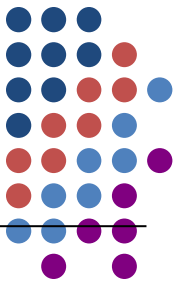
- 对比两种实现方式，动态并行因其在GPU中递归调用的实现方式其核函数运行耗时明显；
减少了与CPU的数据传输次数，数据传输上更具优势

2024RT会议参会

● Poster + Mini Oral



■ 准备2024 RT海报展示和Mini Oral分享演讲



当前计划

- ◆ 完善基于CPU/GPU异构的波形重建优化的后续研究
- ◆ 撰写RT论文并针对会议内容展开一些相关研究调研

THANK YOU !

