



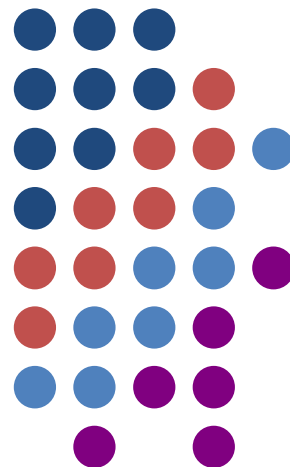
中国科学院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences

十二月季度考核报告

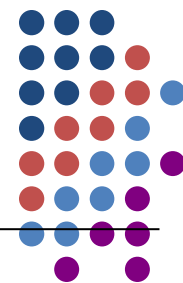
学生：肖鹏飞

导师：季筱璐

2023年12月28日



主要内容

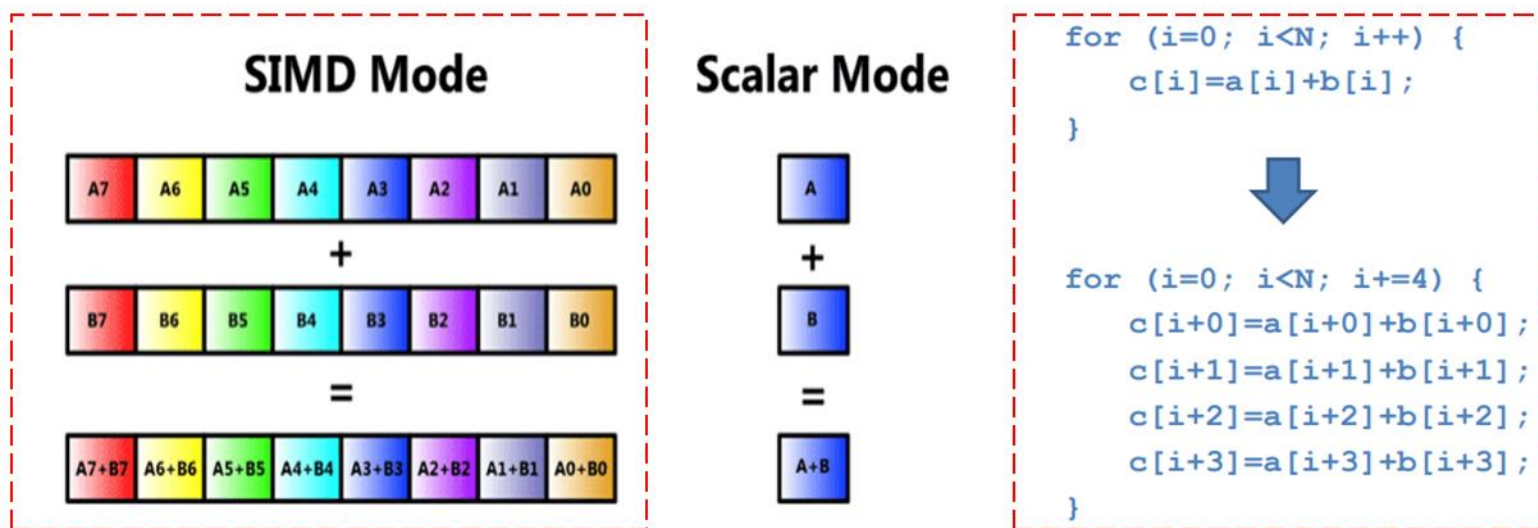


- ◆ 基于CPU扩展指令集SIMD技术的研究与应用
- ◆ 国外高能物理实验异构计算应用的调研
- ◆ 开题
- ◆ 当前计划



基于CPU扩展指令集SIMD技术的研究与应用

- 基于CPU扩展指令集的单指令多数据(SIMD)并行技术的研究

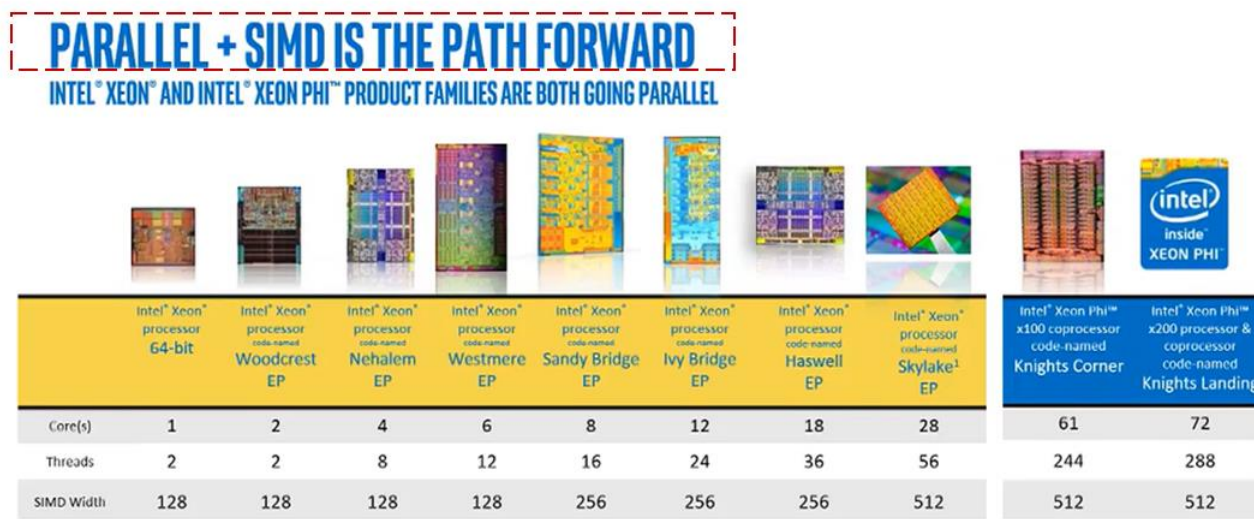


- 基于CPU使服务器 → 无需依赖额外设备资源
- 基于CPU专用寄存器 → 实现高效的并行计算
- 有效加速程序中数据的迭代处理



基于CPU扩展指令集SIMD技术的研究与应用

- OpenMP并行技术模型



More cores → More Threads → Wider vectors

OpenMP* is one of most important vehicles for the parallel + SIMD path forward

- OpenMP+SIMD推动了并行技术的发展，有较大的应用潜能
- 高性能且具有较好的可移植性

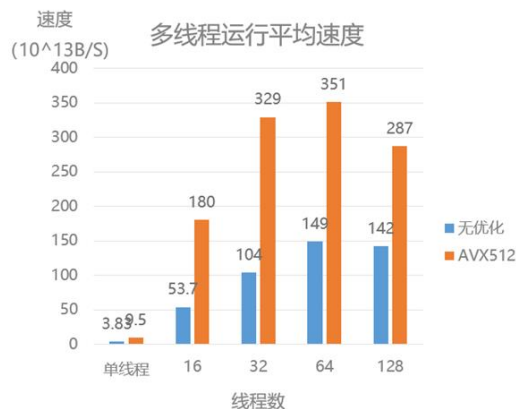


基于CPU扩展指令集SIMD技术的研究与应用

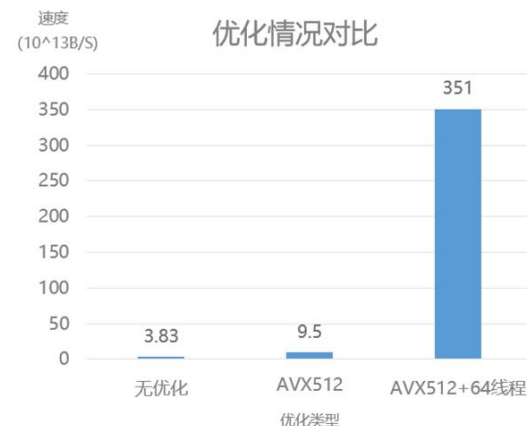
● 前期研究实验测试



单线程测试性能最高优化至
357.58%

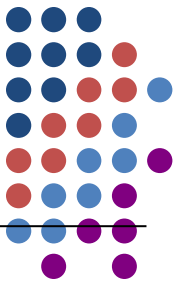


多线程测试性能最高优化至
316.35%



结合OpenMP多线程总体性能较
无优化最高提升至**9164.49%**

- 单线程不同大小矩阵计算性能测试
- 与OpenMP结合不同线程场景下性能测试



基于CPU扩展指令集SIMD技术的研究与应用

● BPIX DAQ图像处理-平场校正应用

```
for (int i = 0; i < pixel_count - 16; i += 16)
{
    //类型转换
    __m256i vec_tmp_0 = _mm256_loadu_si256((__m256i*)&raw_mat[i]);
    __m512 vec_tmp_1 = _mm512_cvtepi32_ps(_mm512_cvtepi16_epi32(vec_tmp_0));
    __m512 params = _mm512_loadu_ps(&params_arr[i]);

    // 乘法计算
    __m512 vec_tmp_2 = _mm512_mul_ps(vec_tmp_1, params);

    // 比较操作
    __mmask16 cmp_result = _mm512_cmp_ps_mask(vec_tmp_2, _mm512_set1_ps(POWER_2_16), _CMP_LT_OQ);

    // 使用掩码进行条件选择
    __m512 vec_tmp_3 = _mm512_mask_blend_ps(cmp_result, _mm512_set1_ps(POWER_2_16), vec_tmp_2);

    // 存储结果
    __m256i vec_tmp_4 = _mm512_cvtepi32_epi16(_mm512_cvtps_epi32(vec_tmp_3));
    __mm256_mask_storeu_epi16(&raw_mat[i], 0xFFFF, vec_tmp_4);
}

// 处理剩余的不足 16 个元素
for (int i = pixel_count - pixel_count % 16; i < pixel_count; ++i)
{
    float correction_raw_mat_tmp = raw_mat[i] * params_arr[i];
    if (correction_raw_mat_tmp > POWER_2_16)
    {
        correction_raw_mat_tmp = POWER_2_16;
    }
    raw_mat[i] = uint16_t(correction_raw_mat_tmp);
}
```

优化前测试耗时:

```
use time : 0.871476
use time : 0.928754
use time : 0.965618
use time : 1.01007
use time : 0.875985
use time : 0.882933
use time : 0.882784
use time : 0.883437
use time : 1.15156
use time : 0.871262
use time : 0.871631
use time : 0.871451
use time : 0.871931
use time : 0.909996
use time : 0.884277
use time : 0.871583
use time : 0.887508
use time : 0.871154
use time : 0.873174
use time : 0.895691
use time : 0.874698
use time : 0.929394
use time : 0.871474
use time : 0.899893
```

优化后测试耗时:

```
use time : 0.12284
use time : 0.123
use time : 0.122904
use time : 0.123282
use time : 0.122886
use time : 0.123114
use time : 0.12363
use time : 0.122976
use time : 0.122994
use time : 0.122649
use time : 0.122894
use time : 0.12338
use time : 0.122754
use time : 0.122964
use time : 0.122866
use time : 0.123095
use time : 0.123124
use time : 0.122952
use time : 0.122992
use time : 0.122744
use time : 0.122646
use time : 0.123038
```

- 完成平场校正部分优化部署
- 代码性能提升至约**700%**符合测试预期

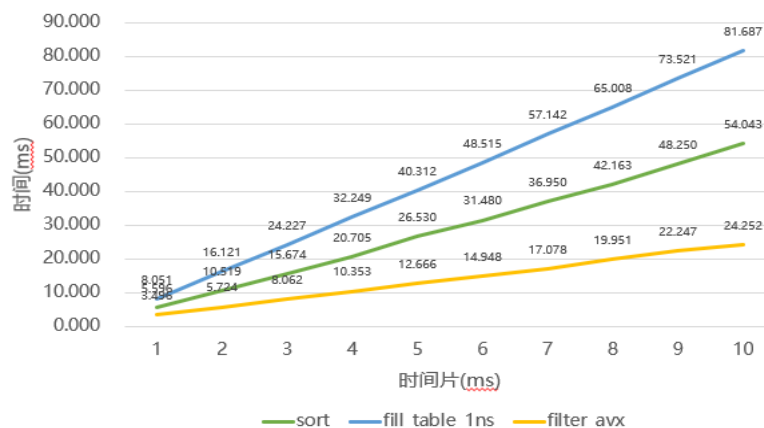


基于CPU扩展指令集SIMD技术的研究与应用

● Nhit软件触发算法加速

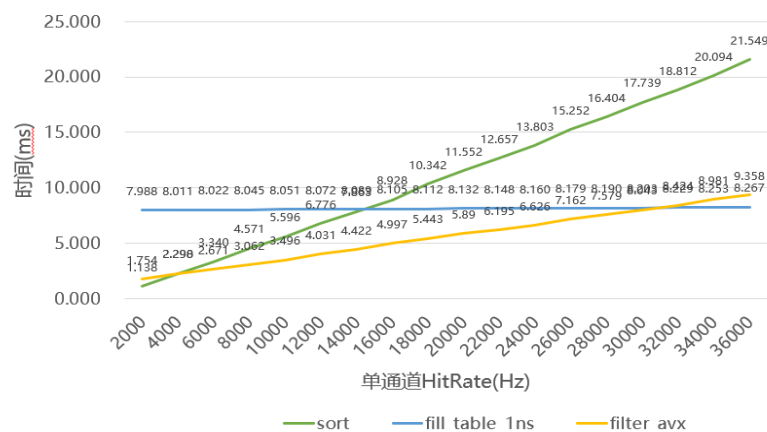
```
__m512i vTriggeerCount = __mm512_setzero_si512();
__m512i vRank2 = __mm512_setzero_si512();
uint64_t* backEndHits = new uint64_t[simdwidth];
for (int i = 0; i < Hits.size(); i+=simdwidth)
{
    //加载数据
    __m512i vFilter = __mm512_loadu_si512((__m512i*)&filter[i]);
    //进行条件判断
    __mmask8 mask_e = __mm512_cmpeq_epi64_mask(vFilter, __mm512_set1_epi64(0));
    __mmask8 mask_l = __mm512_cmple_epi64_mask(vFilter, __mm512_set1_epi64(Filter_In_Start));
    __mmask8 mask_g = __mm512_cmpgt_epi64_mask(vFilter, __mm512_set1_epi64(Filter_In_Start + 6));
    __mmask8 mask = mask_e & mask_l & mask_g;
    //统计
    vTriggeerCount = __mm512_mask_add_epi64(vTriggeerCount, mask, vTriggeerCount, __mm512_set1_epi64(1));
    //查找起始
    __mmask8 mask1 = __mm512_cmpeq_epi64_mask(vFilter, __mm512_set1_epi64(Filter_In_Start));
    //更新起始
    if (mask1 != 0)
    {
        for (int j = 0; j < simdwidth; j++)
        {
            if (Hits[i+j].size())
            {
                backEndHits[j] = Hits[i + j].back().time;
                Hits[i + j].pop_back();
            }
        }
        __m512i vBackEndHits = __mm512_loadu_si512((__m512i*)&backEndHits[0]);
        vFilter = __mm512_mask_blend_epi64(mask1, vFilter, vBackEndHits);
        __mm512_storeu_si512((__m512i*)&filter[i], vFilter);
    }
}
//更新_rank_2
vRank2 = __mm512_max_epi64(vRank2, vFilter);
```

1000通道数10000Hz单通道频率算法执行时间



- 应用SIMD优化测试了一种新提出的软件触发算法实现
- 性能优势主要体现在**低通道数**以及**大时间片**场景

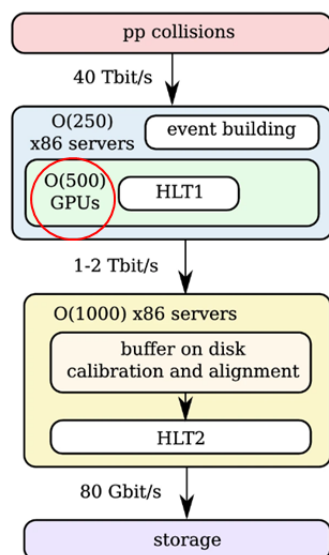
1000通道数算法执行时间



国外高能物理实验异构计算应用的调研



● LHCb/ALICE



LHCb

RORC 1	C-RORC	CRU
		
2 ch @ 2 Gb/s PCIe gen.1 x4 (1 GB/s)	12 ch @ up to 6 Gb/s PCIe gen.2 x 8 (4 GB/s)	24 ch @ 5 Gb/s PCIe gen.3 X 16 (16 GB/s)
Custom DDL protocol	Custom DDL protocol (same protocol but faster)	GBT
Protocol handling TPC Cluster Finder	Protocol handling TPC Cluster Finder	Protocol handling TPC Cluster Finder Common-Mode correction Zero suppression
Run 1	LS1	Run 2
		LS 2
		Run 3

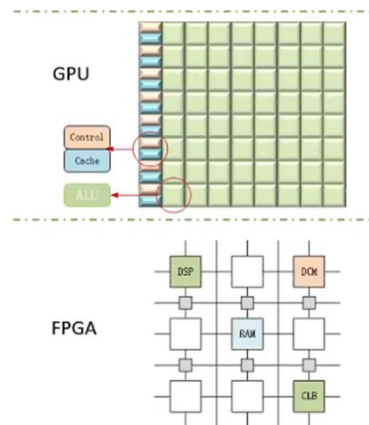
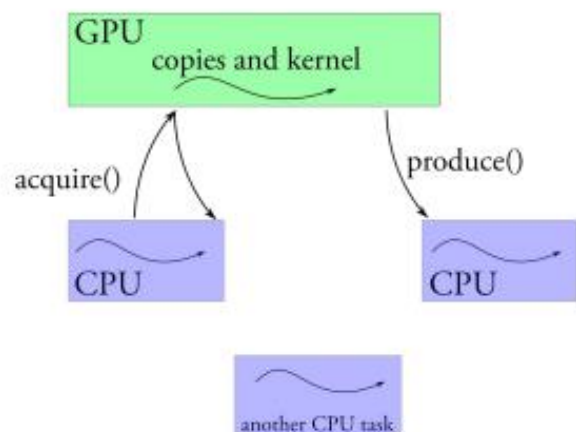
ALICE

- LHCb长期探索GPU的在线计算潜力，并有效实施
- ALICE探索了FPGA在线数据处理的可行性，并在数据处理模型中进行应用

国外高能物理实验异构计算应用的调研



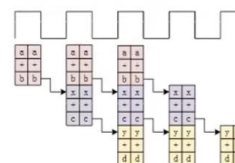
- GPU和FPGA异构计算



支持大量数据的并行计算 适用于SIMD



支持数据的并行与流水线的定制化设计 适用于MIMD

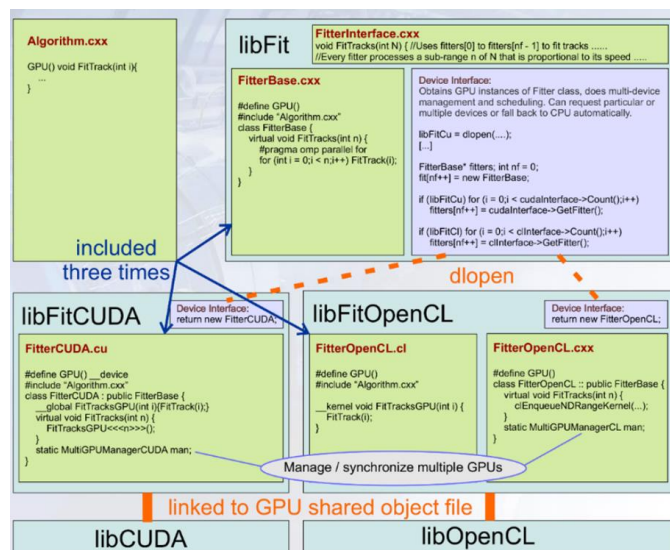


- 充分利用不同设备的并行优势加速CPU计算负载

开题



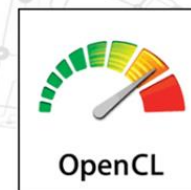
- 完成开题报告的撰写
- 进行开题答辩
- 明确前期研究相关技术路线（异构平台测试：CUDA+OpenCL）



CUDA

Code compare OpenCL

```
1 // ACL kernel for adding two input vectors
2
3 #define __attribute__((packed)) __attribute__((aligned(8)))
4
5 struct data_in {
6     float a;
7     float b;
8 };
9
10 struct data_out {
11     float c;
12 };
13
14 #define __attribute__((num_simd_work_items(1)))
15 #define __attribute__((num_compute_units(1)))
16
17 __kernel void rich(__global const struct data_in* restrict dataIn,
18                  __global struct data_out* restrict dataOut)
19 {
20     // get index of the work item
21     private int index = get_global_id(0);
22
23     dataOut[index].c = dataIn[index].a + dataIn[index].b;
24 }
25
```



OpenCL

当前计划



- ◆ 开展后续异构计算技术的调研和学习
- ◆ 尝试对JUNO的波形重建部分进行加速优化

THANK YOU !

