



12月研究生考核报告

报告人: 张叙

指导老师: 李飞

专业: 计算机应用技术

触发与数据获取组

2023年12月28日

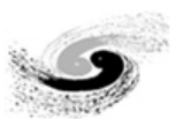


◆ JUNO相关

- ◆ 数据检查系统升级
- ◆ TQ模拟数据源开发
- ◆ 数据流高可用框架测试验证
- ◆ 直方图显示程序开发

◆ CEPC相关

- ◆ 并行技术调研
- ◆ 软件触发方案研究



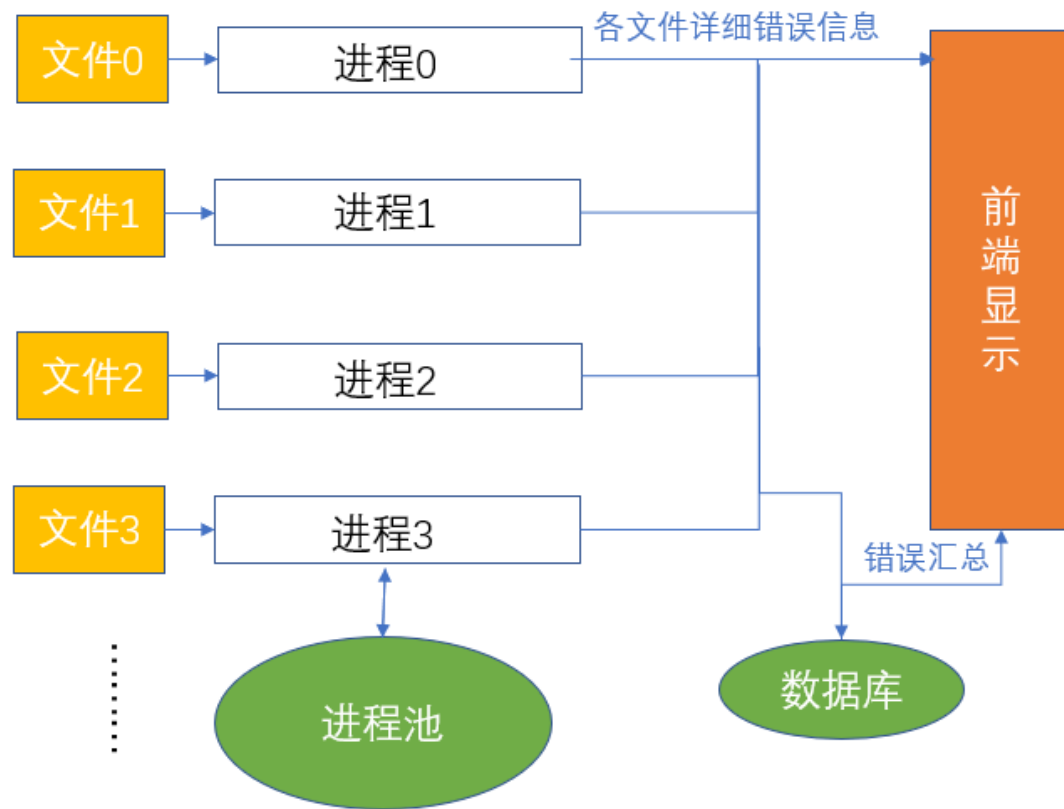
数据检查系统升级

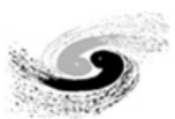
➤ 数据检查系统需求

- 数据检查系统可以从**数据**的角度验证DAQ是否功能正常。
- 测试我们所用的**测试数据**本身有没有问题。

➤ 主要工作

- 参照之前版本的数据检查系统重写了数据包的**检查逻辑**。
- 由之前的只能检查**单层**数据包升级到了可以检查**三层**数据包。
- 通过代码的重写以及**模块化**实现了一定程度的**通用性**。

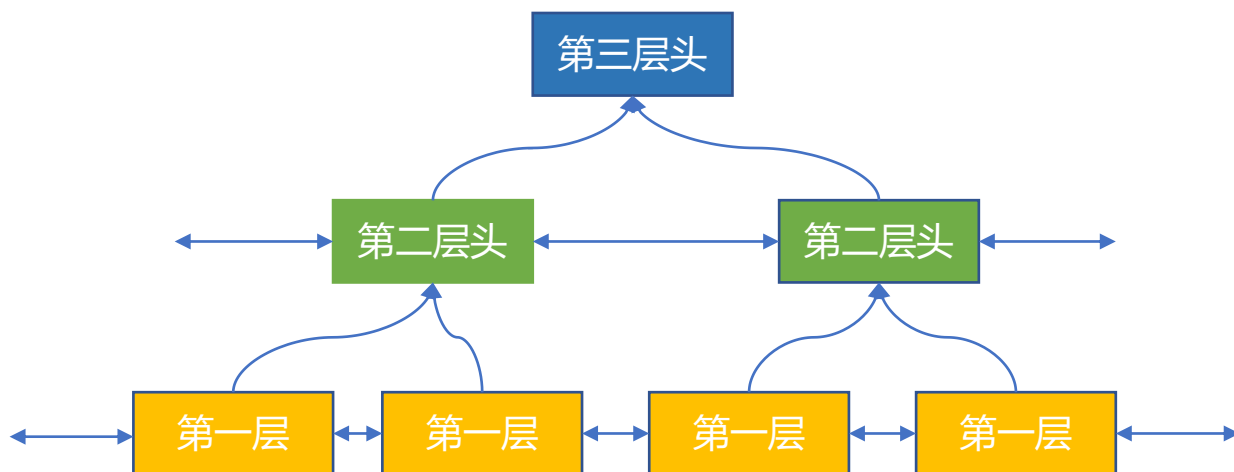




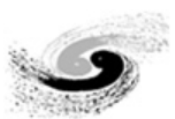
数据检查系统升级

➤ 具体方法

- 基于链表与树设计了一个网状树，便于快速定位错误。
- 通过解析用户所写的json文件满足不同用户的要求。



```
"head1": "0x805a",
"tail1": "0x8069",
"packetSize": "0x7f0",
"head1sumSize": 16,
"tail1sumSize": 16,
"head1num": 5,
"head1size": 16,
"tail1num": 6,
"tail1size": 16,
"head2": "0xee1234ee",
"head2size": 32,
"head2num": 15,
"head3": "0xff1234ff",
"head3size": 32,
"head3num": 5,
"level1size": [16, 16, 16, 16, 48, 16, 16, 16, 16, 32, 16],
"level2size": [32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32],
"level3size": [32, 32, 32, 32, 32],
"first_level_first_choice": [1, 2],
"first_level_second_choice": [1, 2],
"first_level_function_choice": [1, 8],
"second_level_first_choice": [0],
"second_level_second_choice": [0],
"second_level_function_choice": [0],
"third_level_first_choice": [0],
"third_level_second_choice": [0],
"third_level_function_choice": [0]
```



数据检查系统升级

选择检查的文件

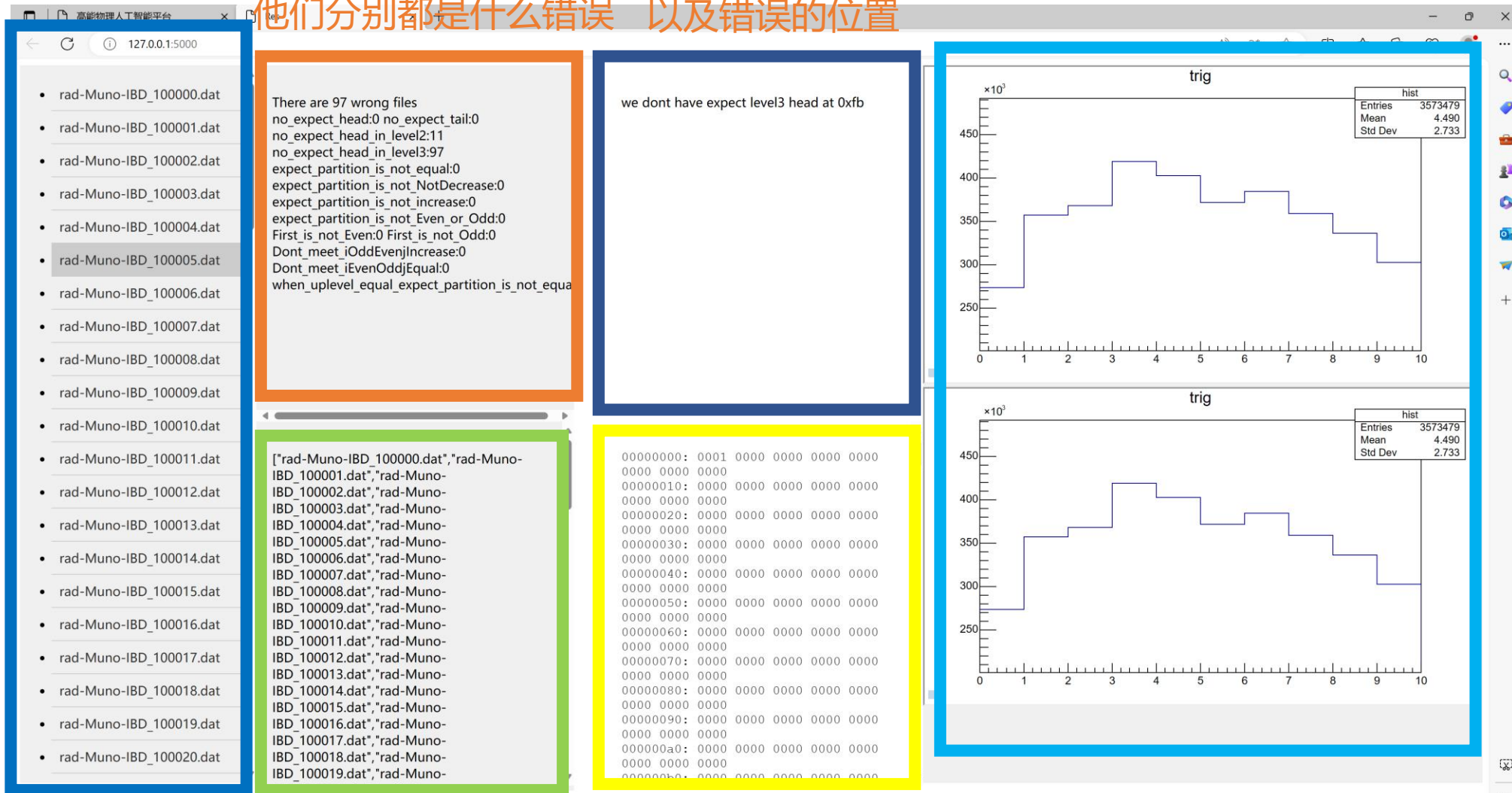
总共多少个文件错误

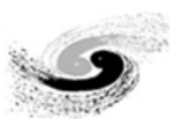
所选择的文件的错误

一些需要画的直方图

他们分别都是什么错误

以及错误的位置





TQ模拟数据源开发

➤ **需求：**DAQ需要TQ模拟数据源对其TQ处理的相关功能进行测试以及验证。

➤ **主要工作：**

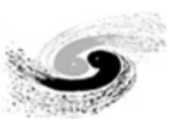
- ① 开发TQ模拟数据源，模拟数据源在单台服务器上可以模拟至少900个gcu以30khz的速率向DAQ发TQ数据。
- ② 编写内存池以对内存进行管理，使每次的操作尽量在一个Cache块中进行，保证数据源的性能。
- ③ 可以发送不同比例的好事例，为触发软件的调试提供测试条件。
- ④ 模拟数据源可以模拟DAQ读出通过ipbus与硬件交互的过程。

```
(base) [zhangxu@hd099 bin]$ fee_dummy_main_plus -h
Usage: fee_dummy_main [-vh] [-p 60000] [-d WAVE] [-n 30] [-b 0] [-r 100] [-g 3] [-t 30] [-a hd001:8087]
    -p, --port=60000          第一个监听端口号，如60000
    -d, --detector_type=WAVE 模拟数据类型，有WAVE、TQ、MIX三种
    -n, --connection_gcu_number=30 连接的总GCU数
    -b, --begin_gcu_id=0      第一个连接GCU号，默认为0
    -r, --event_rate=100      发送事例率，默认为100Hz
    -g, --GCU_channel_number=3 一个GCU包含的通道数，默认为3
    -t, --thread_GCU_number=30 每个线程连接GCU数，默认30，注意线程+1.则端口号相应也+1
    -a, --redis_address=hd001:8087 设置运行参数写入的redis地址，默认hd001:8087
    -v, --version              打印版本信息
    -h, --help                 打印使用帮助信息
(base) [zhangxu@hd099 bin]$
```

```
时间 2023-11-28 13:51:37 线程 4 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 0 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz

时间2023-11-28 13:51:37 平均发送事例率= 30000.91 Hz, 吞吐= 54865.67 MB/s

时间 2023-11-28 13:51:37 线程 7 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 9 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 10 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 6 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 13 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 8 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 11 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 14 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 3 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
时间 2023-11-28 13:51:37 线程 5 ,设定事例率= 30000 Hz, 发送事例率= 30004.55 Hz
```

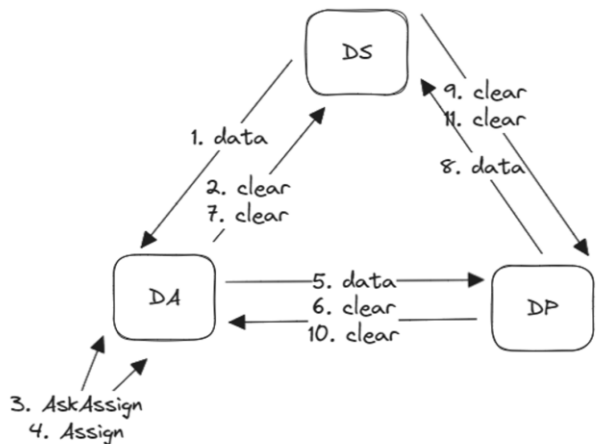


数据流高可用框架测试验证

- **需求**：为了能保证数据流在超新星爆发时正常工作，数据流需要实现高可用，高可用上架前需要先测试验证。
- **单点故障重传(单点坏掉不丢数)**模型是当前数据流高可用架构的主要组成部分，针对这一模型做了以下工作：

- 用**反证法**证明了这个模型可以保证单点故障时，有备份存在。 理论模型在DP DA同时坏时潜在问题

反证法证明单点故障(DA or DP出错)发生时有备份存在：

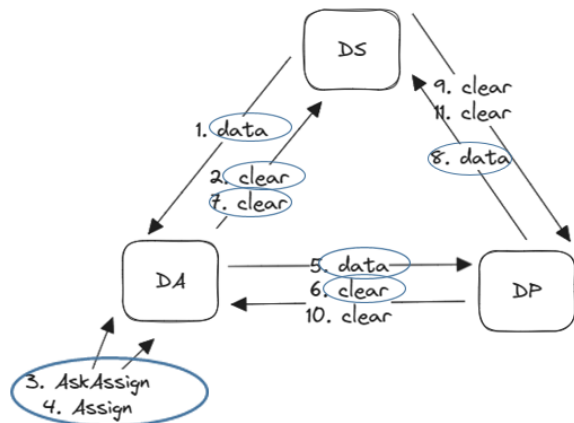


已知单个数据在发送中，流程运行的时间顺序满足 $T(1.data) < T(2.clear) < T(3.askassing) < (\dots \text{省略})$
假设在DA发生单点故障并且数据没有发送到DP时，DS中不存在备份。

$\therefore$ DS中不存在备份
 $\therefore$ 7.clear已经发生，删除了DS中存的备份
又 $\therefore$ DA发生了故障，并且数据还没发送出去
 $\therefore$ 5.data, 6.clear还没有执行
 $\therefore T(7.clear) < T(6.clear)$
与已知矛盾

假设在DP发生单点故障并且数据没有发送到DS时，DA中不存在备份。

$\therefore$ DA中不存在备份
 $\therefore$ 10.clear已经发生，删除了DA中存的备份
又 $\therefore$ DP发生了故障，并且数据还没发送出去
 $\therefore$ 8.data, 9.clear还没有执行
 $\therefore T(10.clear) < T(9.clear)$
与已知矛盾



当8.data发送时，数据已经被传递到了DS，此时DA, DP的损坏并不会对其造成影响。

- 分析理论模型在各种情况下的备份恢复情况。



- 运行两个脚本，定时kill掉结点，总运行时长169min，未出现丢数。

- 给出了一种去重方案，并进行了测试。
- 当**队列**的大小大于一定值时，可以实现重传无重复。

重复问题解决方案：建立一个全局的哈希表以及循环队列

- 由上述分析可以得知，一个数据的重复传输在一个有限的时间范围内，可以利用一个循环队列按照时间顺序存储一定时间范围内的数据的id号，再利用哈希表进行去重操作。

- 具体过程为一个数据到达该算法时，先存入循环队列中，再存入哈希表中，哈希表的value为该id数据的个数，当个数大于1，即说明该数据重复，当队列存满时，删除队头元素，在队尾加入新元素，并对哈希表同样进行插入与删除，若某个id的计数为0，则删除这一项。

- 哈希表的查找插入与删除以及队列的插入与删除的时间复杂度都是 $O(1)$ ，因此总的复杂度是 $O(n)$ 。

- 空间复杂度是常数级 $O(1)$ 。



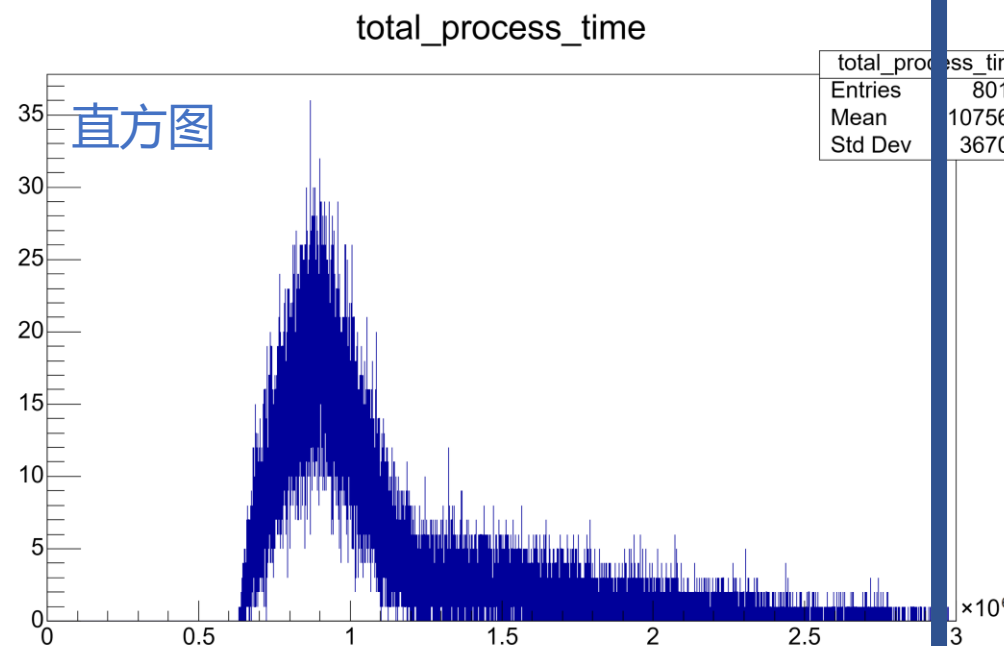
主要工作

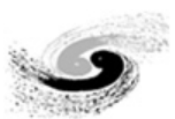
用直方图展示数据流中物理相关的信息。

- 在网页上拿到json类型的直方图相关信息，解析。
- 实现了一个队列，实现一边取数解析，一边画图合并。

JSON数据

直方图





并行技术调研

➤ **主要目的：**学习DUNE架构为CEPC做并行计算技术储备。

➤ **主要工作：**

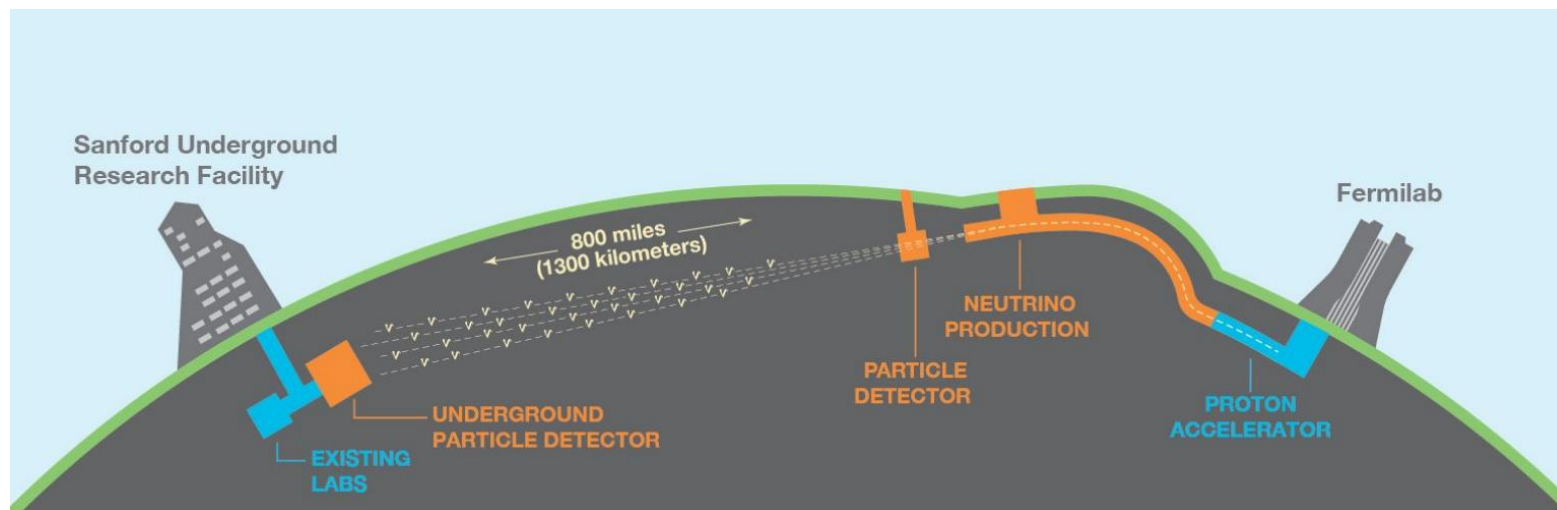
1. 整理DUNE调研文档，在组会上做调研分享报告。
2. 学习其采用的并行技术(SIMD)，在查找匹配算法上进行了应用。

SIMD技术

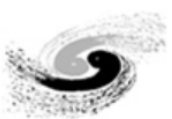
- 将**多个**数放在一个寄存器中
- 用**一次**计算来代替**多次**计算。

1		1		2
2		2		4
3	+	3	=	6
4		4		8

1		1		2
2		2		4
3	+	3	=	6
4		4		8



Pic from The DAQ for the single phase DUNE Prototype at CERN. Roland Sipos (CERN) for the DUNE Collaboration



并行技术调研

➤ 基于SIMD对遍历查找算法向量化优化

目标：查找一串字符串中的805a。

2a	85	67	80	5a	69	87	35
----	----	----	----	----	----	----	----

正常依次遍历对比

80	5a
----	----

方法1：一次遍历多个

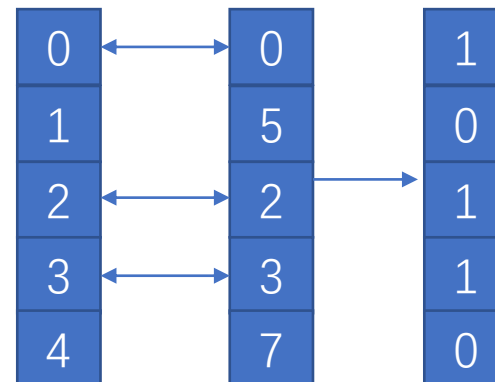
80	5a	80	5a	80	5a	80	5a
80	5a	80	5a	80	5a	80	5a

方法2：分级遍历，第一轮遍历80，找到80再看是不是805a

80	80	80	80	80	80	80	80
----	----	----	----	----	----	----	----

➤ 基于SIMD对匹配算法向量化优化

目标：匹配一段数组是否与另一段数组有x个值相同(可用于基于模式匹配的算法)。

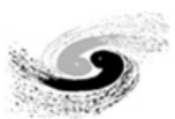


avx_way (向量化查找) 比common_way (逐个遍历) 快了约20倍

```
(base) [zhangxu@daqs1 avx_test]$ g++ -mavx512f -mavx512bw
(base) [zhangxu@daqs1 avx_test]$ ./search
3412
size is1701000
common_way Duration is 9335387 nanoseconds
avx_way Duration is 432536 nanoseconds
```

avx_way比common_way快了约5倍

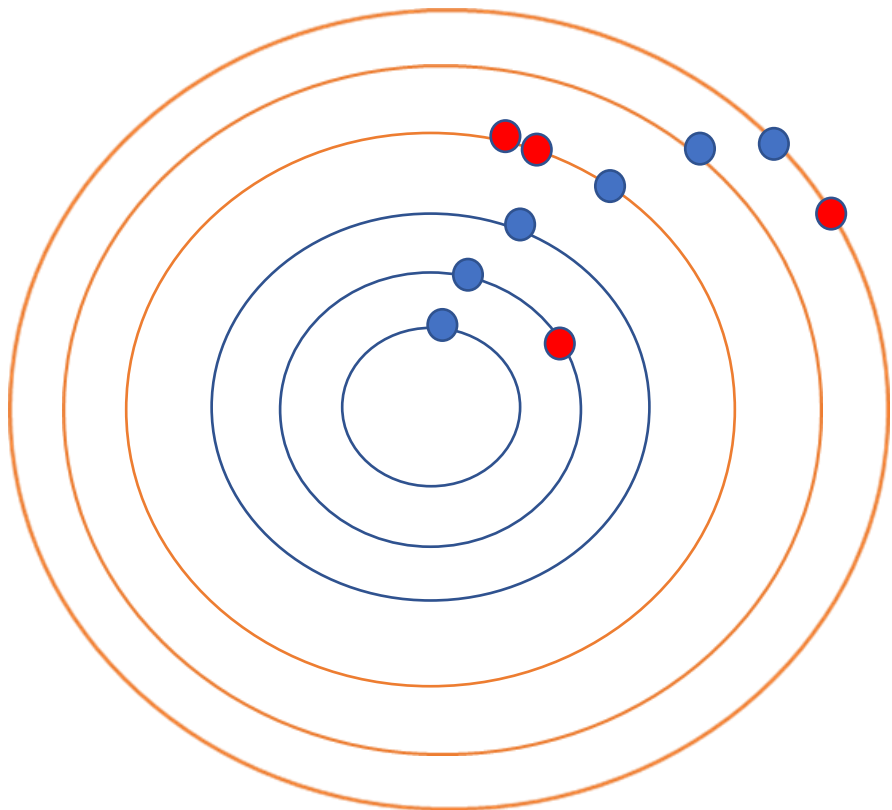
```
(base) [zhangxu@daqs1 avx_test]$ g++ -mavx512f -mavx512bw
(base) [zhangxu@daqs1 avx_test]$ ./match
size is1216512
common_wayDuration is 1747076 nanoseconds
avx_wayDuration is 320167 nanoseconds
```



软件触发方案研究

一种针对于硅径迹探测器的软件触发方案研究

内径迹探测器+外径轨探测器



➤ **主要目的：**为CEPC的软件触发方案做技术储备。

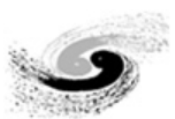
参考BESIII触发条件设置：

1. 事例判选主要根据横动量超过阈值的径迹数量。
2. BhaBha事例(本底)的判选还需要径迹的位置。



一个事例满足触发条件在理论上应该满足以下三个条件

1. 粒子的轨迹能形成半径在一定范围的圆(不同半径对应不同动量)。
2. 能形成的圆的个数多于一定值(横动量超过阈值的径迹条数)。
3. 圆心的位置是否符合要求(BhaBha事例的判选)。

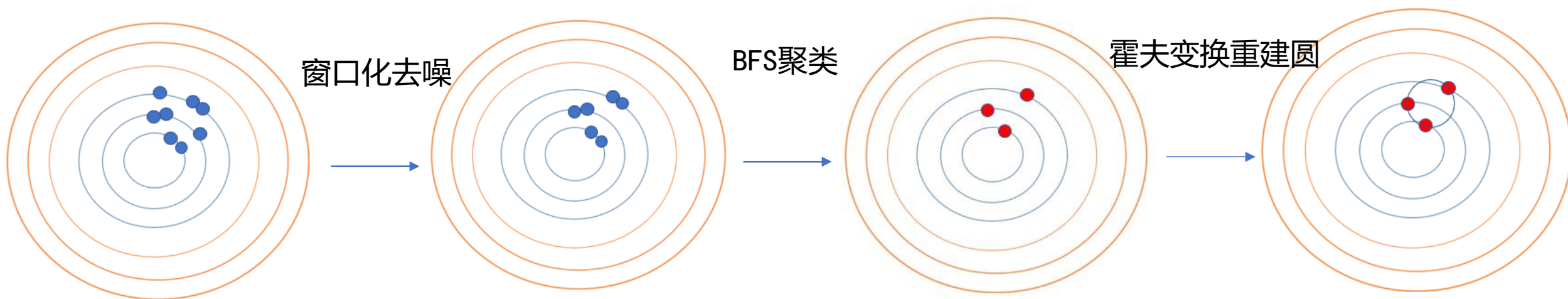


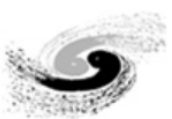
软件触发方案研究

➤ 主要工作

1. 根据大量的调研给出了一套**径迹**触发的软件方案。
2. 利用SIMD，实现了一套窗口**去噪**算法。
3. 利用**广度优先遍历**，实现了一套基于密度的聚类算法。
4. 实现了一套基于霍夫变换能够自动筛选出**半径**以及**圆心**符合要求的**触发判选算法** (belle2采用)。
5. 将霍夫变换的计算过程做成了两个**矩阵相乘**，用SIMD进行**并行优化**。

➤ **不足**：针对的具体实验还不确定，数据缺失，因此对其还需要进一步的测试。

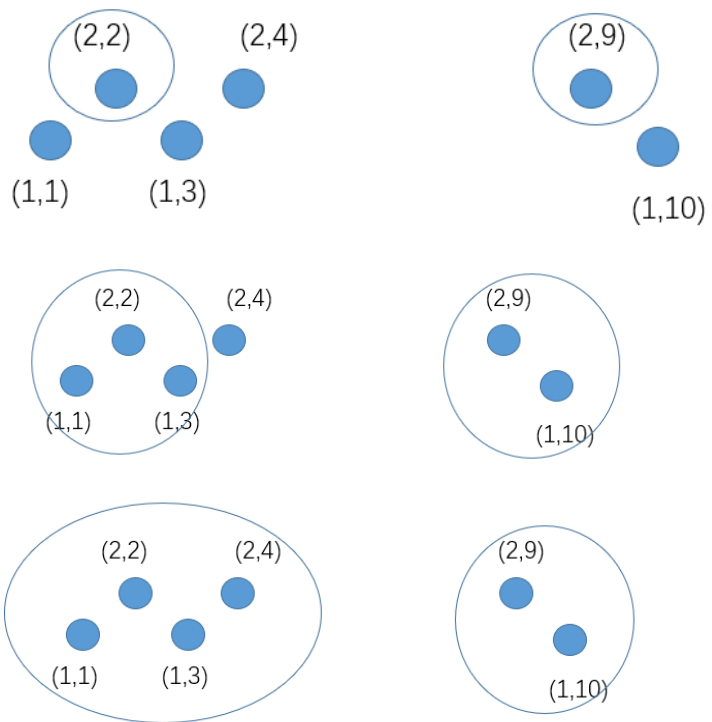




软件触发方案研究

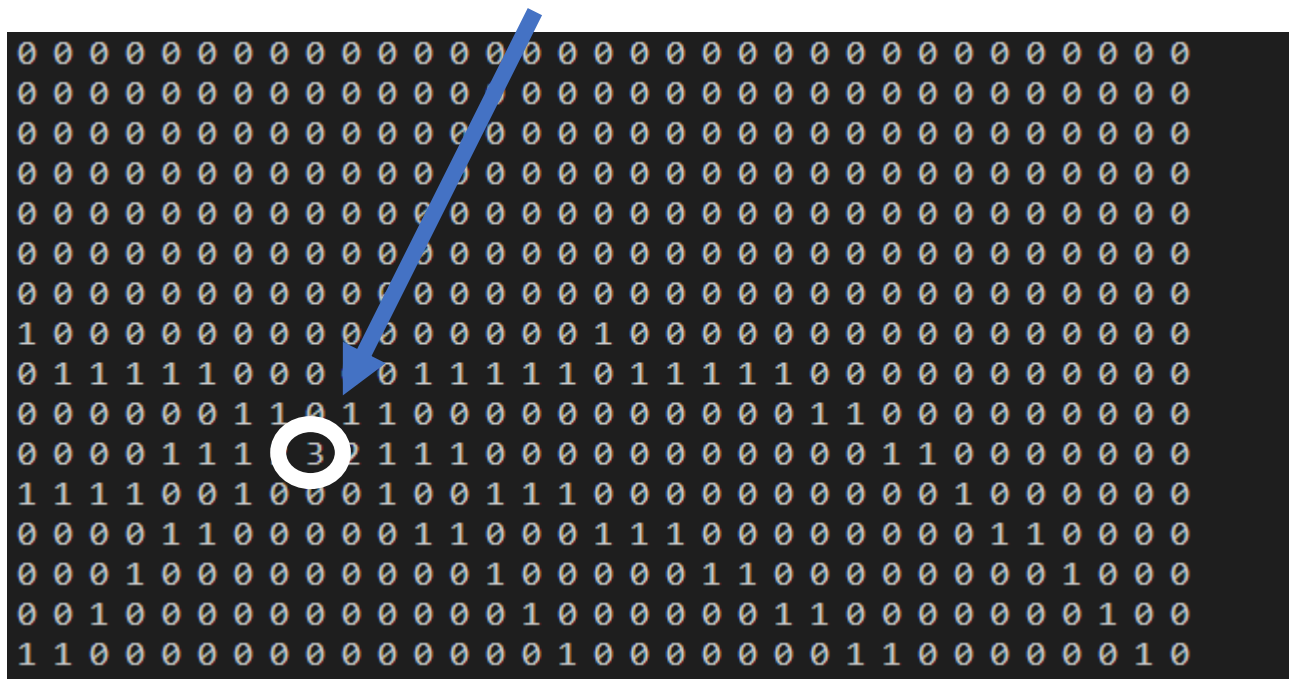
➤ 窗口化去噪 + 广度优先遍历聚类

- 窗口化去噪：看一个点**周围**是否有x个点。
- 广度优先遍历聚类：满足**一定距离**条件视为一簇。



➤ 霍夫变换

- 它通过**参数空间**曲线**交点**来判断**实际空间**中各点的关系。
- 根据**交点**有几个曲线**经过**即可判断有**几个点**在同一个**圆**上，且圆的参数是根据**交点位置值**求出。





谢谢大家