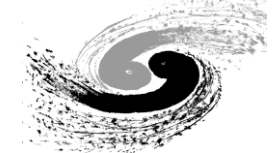


研究生12月考核报告

专业: **计算机应用技术**

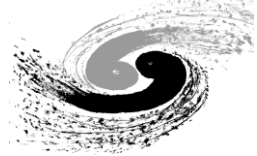
课题组: **触发与数据获取组**

考核人: **张叙**



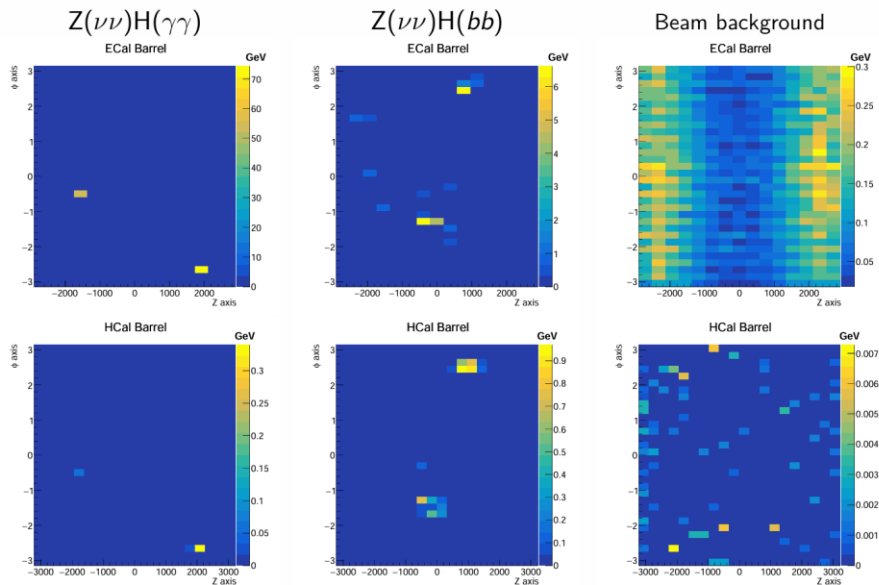
- **CEPC**
 - CEPC在线算法性能研究
 - 在线与离线框架融合研究
 - CEPC在线文档知识库构建
- **其他**
 - Progress of online process software frame for CEPC ref-detector(CEPC workshop报告)
 - JUNO Online Data Process Software(小论文)

CEPC在线算法性能研究(基于量能器)



- 算法原理: 本底事例与物理事例在量能器上的沉积能量相差较大。
- 算法过程: 基于量能器不同的区的沉积能量对事例进行判选。
- 物理效率: > 99% 物理事例, < 1% 本底事例。

- Large energy deposition(> 10 GeV) for signal (photon, and Jet)
- Very tiny energy deposition(<0.5 GeV) for beam background, mostly from pair production

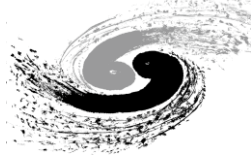


平均每个bunch crossing所需判断时间<5us

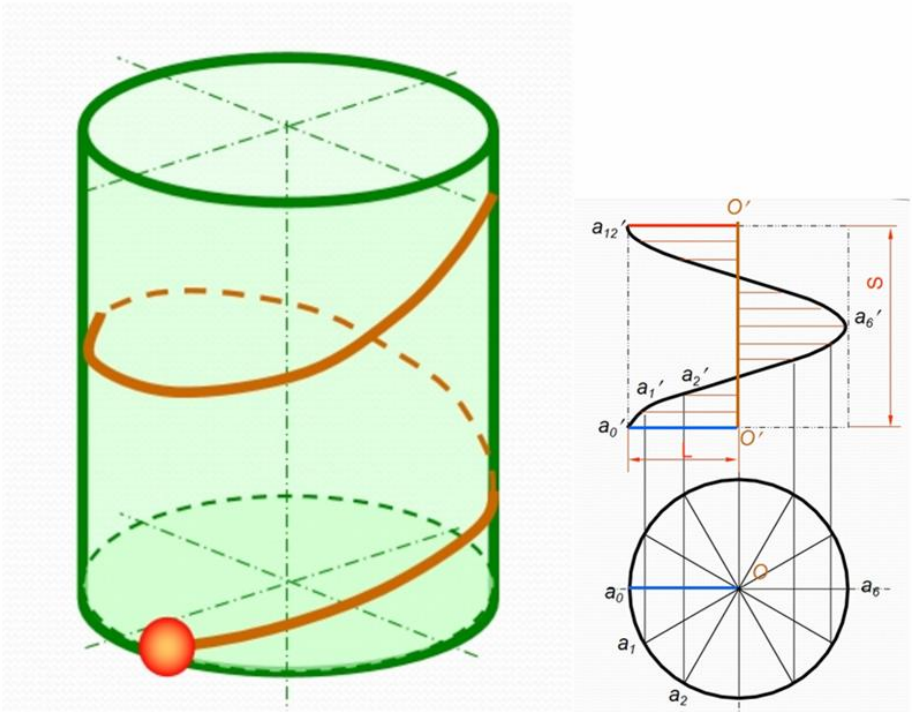
Bunch crossing	1.3MHz(Higgs)	39MHz(High Z)
CPU cores	~7	~200

Trigger simulation progress. BoPing Chen

CEPC在线算法性能研究(基于ITK OTK VTX)



- 算法原理: 在磁场的作用下, 带电粒子的轨迹应是螺旋线。
- 算法过程: 基于VTX, OTK, ITK 中的点是否能形成螺旋线进行判选。
- 物理效率: > 94.9% 物理事例, < 1% 本底事例。

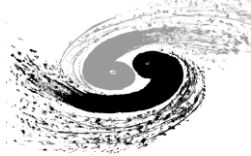


平均每个Bunch crossing所需判断时间~150ms

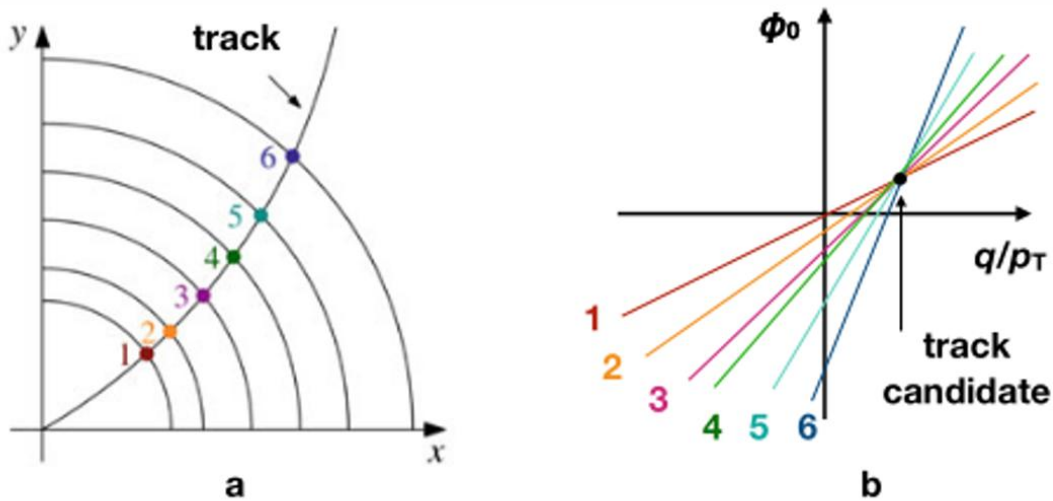
Bunch crossing	1.3MHz(Higgs)	39MHz(High Z)
CPU cores	200k	6000k

Trigger simulation progress. BoPing Chen

CEPC在线算法性能研究(基于TPC)



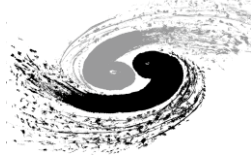
- 算法原理: 基于TPC重建出的track与其他探测器的轨迹应共同构成螺旋线。
- 算法过程: 1.基于霍夫变换, 进行二维平面上的TPC的**track**查找。
2. 重建Z方向进行与其他探测器数据的匹配(未完成)。



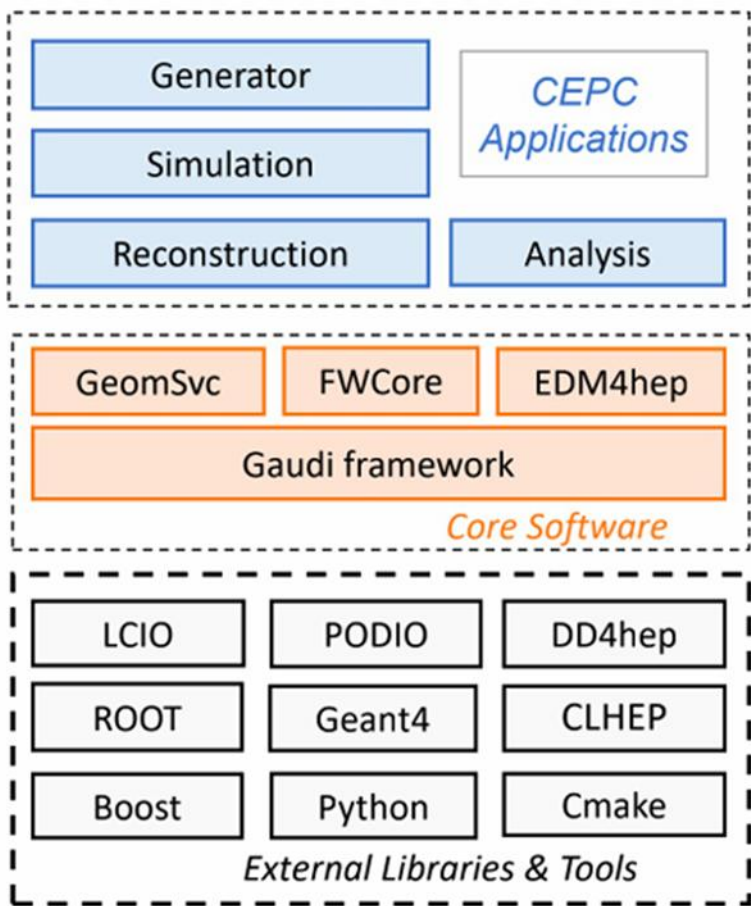
平均每个Bunch crossing track所需判断时间~10ms

Bunch crossing	1.3MHz(Higgs)	39MHz(High Z)
CPU cores	13k	390k

在线与离线框架融合研究



- CEPCSW是面向CEPC的离线算法开发软件,其中包括了探测器, 电子学的模拟以及各种服务。
- 在在线框架中引入了对CEPCSW中的数据模型模块与服务模块适配, 方便离线算法的移植。



online-offline-inegration

main online-offline-inegration /

History Find file Edit Code

delete OnlineScheduler

zhangxu00@ihep.ac.cn authored 1 week ago

23cd0584

Name	Last commit	Last update
.ci	add gitlab runner.	1 year ago
.github/workflows	Try to enable CI with the key4hep stabl...	1 year ago
.vscode	a temp version, which have demo algs' ...	3 weeks ago
Analysis	base version(includes ana digi rec sim t...	3 weeks ago
Detector	Updates on FinalPIDSvc	1 month ago
Digitization	Add option to append random noise for...	4 months ago
Reconstruction	base version(includes ana digi rec sim t...	3 weeks ago
Service	Updates on FinalPIDSvc	1 month ago
Simulation	WIP: oow (out-of-window) time window...	3 months ago
Utilities	Tracking: improve performance for circle...	4 months ago
cmake	Tracking: improve performance for circle...	4 months ago
docker	WIP: prepare docker image el9	7 months ago
docs	Add Data samples Descriptions in Docs	1 month ago
online-computing-platform @ 0b159e47	delete OnlineScheduler	1 week ago

1,632 Commits

1 Branch

0 Tags

23.9 MiB Project Storage

README

CI/CD configuration

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING

Auto DevOps enabled

Add Kubernetes cluster

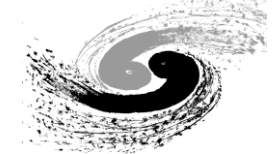
Add Wiki

Configure Integrations

Created on September 16, 2025

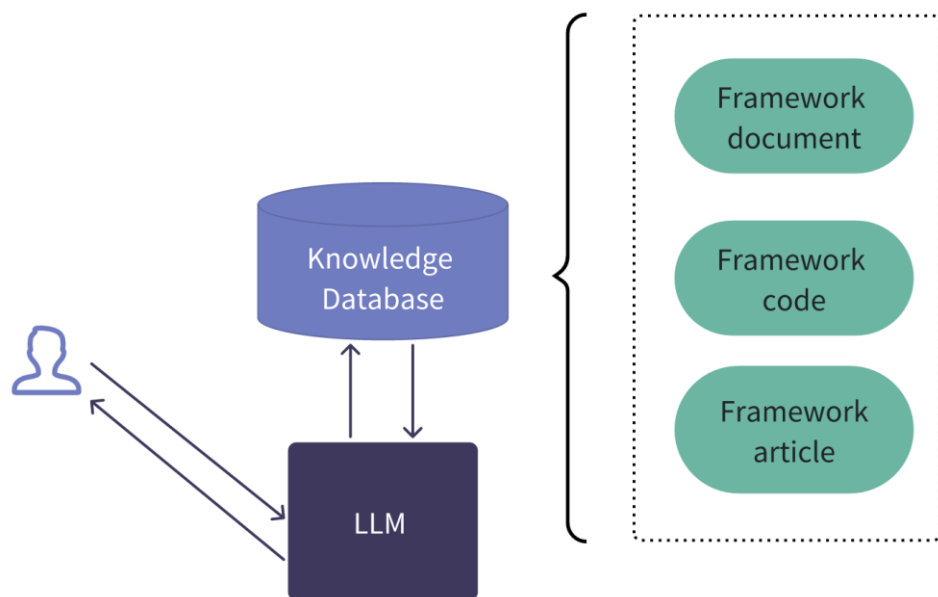
在线调度直接适配调度CEPCSW中的离线算法

CEPC在线文档知识库构建



- 部署本地大型语言模型帮助CEPC在线软件框架用户学习和理解文档。
- 通过利用知识库，大型模型能更好地回答关于CEPC在线框架的问题。

基于CEPC在线框架信息构建知识库



How to Add New Algorithm in this framework



大语言模型基于知识库里的信息给予用户相应的回答。



Okay, the user is asking "How to Add New Algorithm in this framework" after previously discussing how to run the RADAR V3.0 project. Let me analyze the context to provide a precise answer.



Looking at the provided context documents (contents.md with multiple contexts), I see detailed instructions about adding algorithms. The key points from Context 0 are:

01. Place new algorithm under project/a1gs/
02. Must define header and source files
03. Need to register with:
 - Algorithm class name
 - Stream (HLT1/HLT2)
 - Input/output (output must be a struct in ./type/type/datatype.h)
 - Configuration path
 - Include init() and run() methods

Context 2 provides a concrete example with:

在模型回答时，会给出参照的具体文献

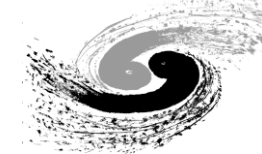
```
C++  
REGISTER_TASKTYPE(TaskType::kHlt1);  
REGISTER_CLASS(Func1);  
REGISTER_OUTPUT(Func1, Func1Output);  
REGISTER_CONFIG(Func1, "config/func1_config.json");  
REGISTER_DEVICE_TYPE(Func1, DeviceType::kCPU);
```

Context 3 discusses configuration details but isn't directly about adding algorithms.

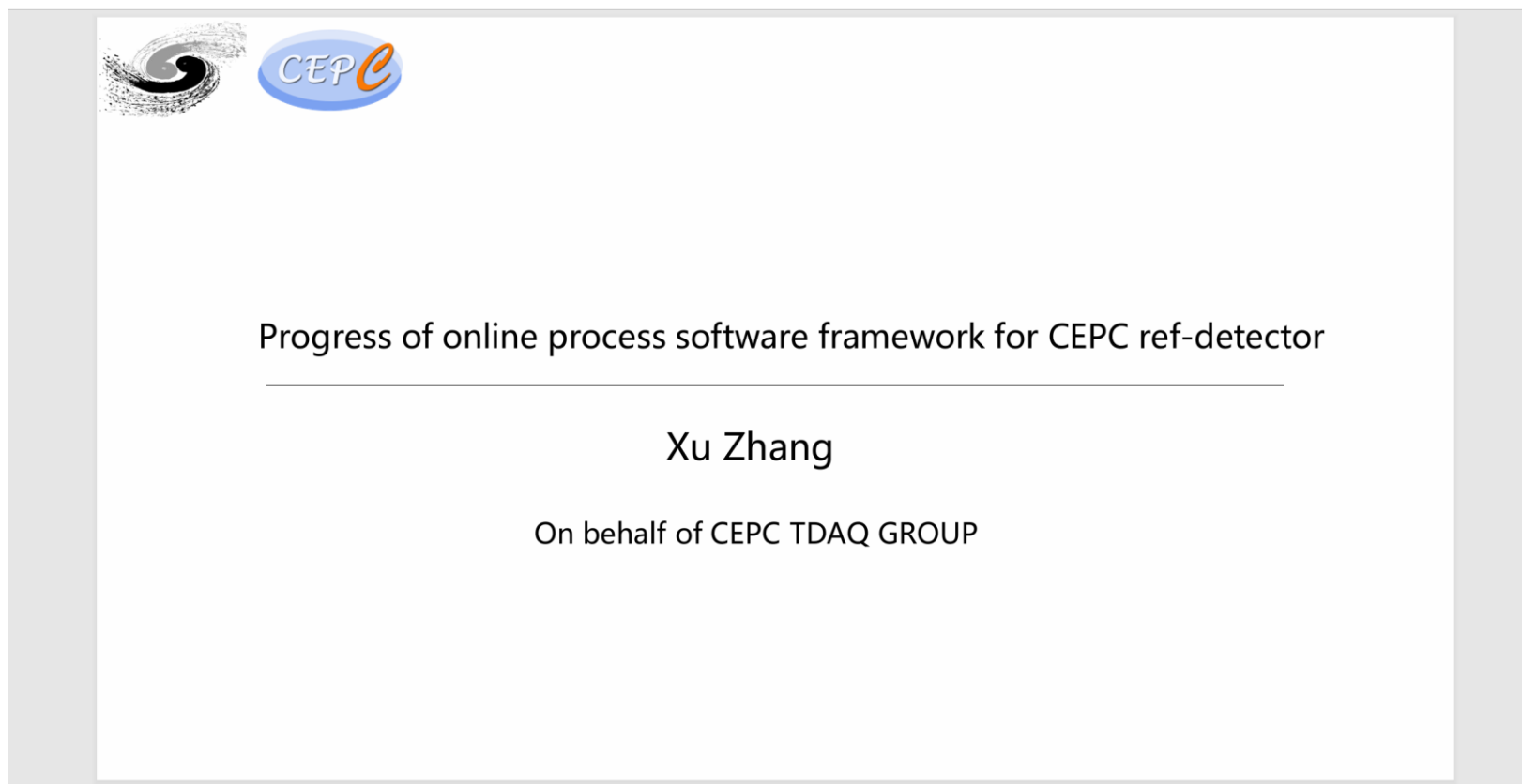
The user's question is very specific about the framework's requirements for adding algorithms. I should provide the exact steps based on the documentation, not general advice.

I'll structure the answer to cover:

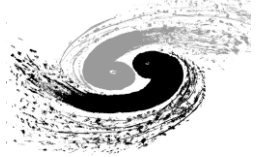
- File placement
- Required registration macros (from Context 2)
- Input/output struct definition
- Configuration setup
- Implementation of init()/run() methods



1. 在2025 CEPC workshop上作了报告: Progress of online process software frame for CEPC ref-detector.
2. 完成小论文的编写: JUNO Online Data Process Software(江门合作组审查中).



下一步计划



1. 经典高级触发算法流构建与CEPC应用。
2. GPU加速技术在CEPC在线算法中的应用。
3. 在线异构计算框架的集成策略。