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

Modern C++ and cuda based parallel fitting

丁雪峰 特聘青年研究员

中国科学院，高能物理研究所，实验物理中心

2024中微子夏令营
2024年7月9日

Warmup: Let’s play typing game.

- Go to https://dazi.kukuw.com/
- No need to login with wechat.
- Choose “competition (invitation)” and use X7JKG
- Keep correctness 100%, and try to beat me
  - 游客7987844 365 KPM

Topic 0

Basics

Hands-on 0. Hello World!

- Open windows terminal and open WSL Ubuntu or mac iTerm2
- Or ssh to the node:
  - ssh -p 19325 root@connect.yza1.seetacloud.com，
  - password 6U0I+Q1L3XvL
- Write hello world in c++
- Compile with gcc
- Run it

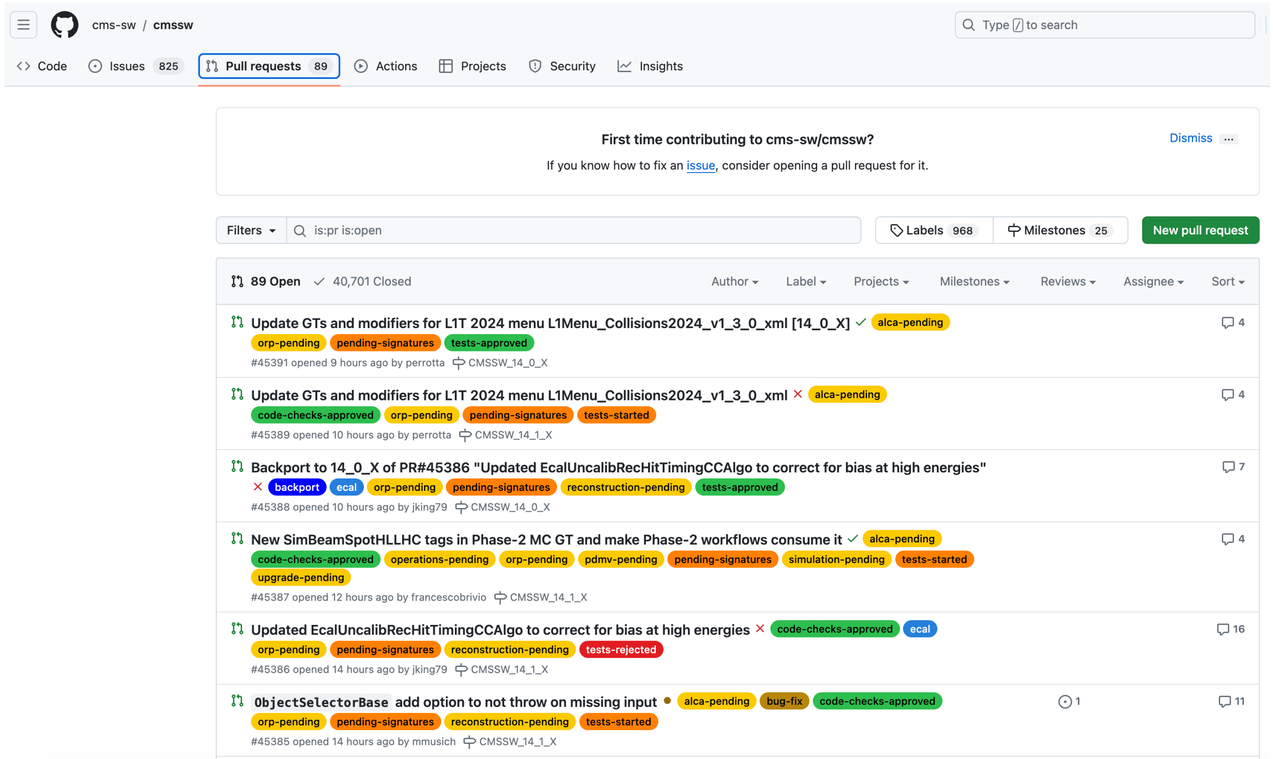
Topic 1

Split the responsibility.

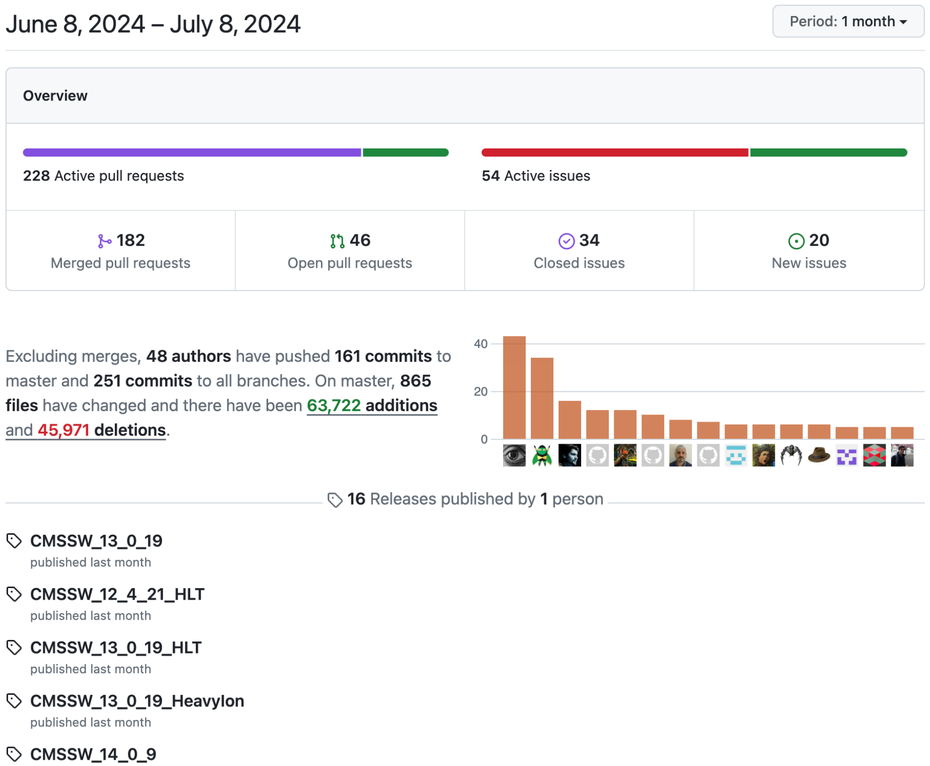
Programming in particle physics

- Complex reconstruction and signal filtering => large-scale software (100,000 lines per month)
- Software engineering method is needed

CMSSW: software used by CMS



Code frequency



The Curse

Project time does not reduce with more manpower

The Spell

Split the responsibility.

Hands-on 1. Split the responsibility (C++ or python)

- You have four random integers: A, B, C, and D and need to update their value so they decrease in order. You and GPT split the responsibility.
- Ask GPT to write two functions increase() and equalize() which will increase one number and equalize two number respectively. For example,
  - `void increase(int &a) {  a+=1; }`
- Now you write your part, use increase and equalize and update A, B, C, D. For example,
  - `void main() { int A=rand(), B=rand(); equalize(A,B);increase(A); std::cout<<A<<“ “<<B<<std::endl; }`

Hands-on 1. Solution-step 1

You

write two function increase() and equalize() which will increase one number and equalize two number respectively. For example,
void increase(int &a) { a+=1; }

ChatGPT

Certainly! Here are the two functions `**increase()**` and `**equalize()**` in C++:

cpp — Copy code

```cpp
#include <iostream>
using namespace std;

// Function to increase the value of a number by 1
void increase(int &a) {
    a += 1;
}

// Function to equalize the values of two numbers
void equalize(int &a, int &b) {
    int avg = (a + b) / 2;
    a = avg;
    b = avg;
}
```

Hands-on 1. Solution-step 2

hands-on-1 > main.cc > ...

```cpp
#include <iostream>

void increase(int &a) {
    a+=1;
}
void equalize(int &a, int &b) {
    int avg = (a+b)/2;
    a = avg;
    b = avg;
}
int main() {
    int a = 37, b = 99;
    equalize(&: a,&: b);
    increase(&: a);
    std::cout<<a<<" "<<b<<std::endl;
}

```

```
dingxf@Xuefengs-MacBook-Pro hands-on-1 % g++ -o main main.cc
dingxf@Xuefengs-MacBook-Pro hands-on-1 % ./main
69 68
```

Splitting the responsibility

Wrap one task into one bag.

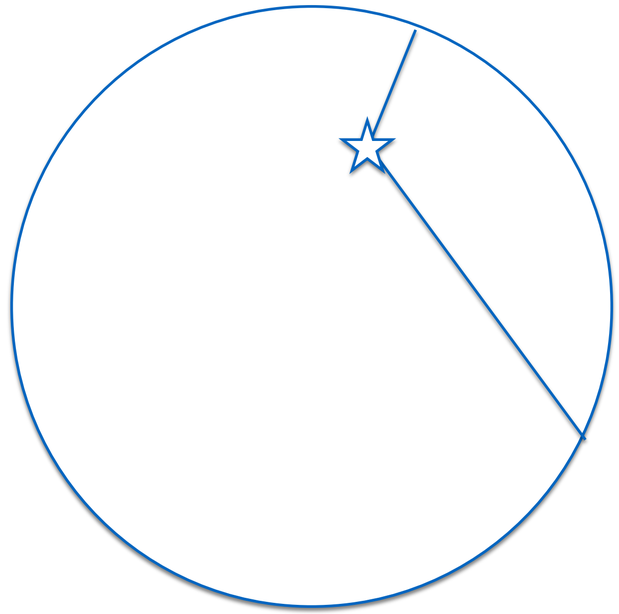
Everything is inside the bag => no worry inside

The bag know nothing outside => no worry outside

The key is “no worry”

Example: Position Reconstruction

- Time of flight => Distance to PMT => can be used to determine the event position.
- Need a function to calculate the “time of flight”.
- We can split the responsibility, take this out, and put it in a separate function.
- It’s easier for debug.



Topic 2

Use the class to package the responsibility

The Curse

Multi-department bureaucracy is a nightmare.

The Spell

Merge the department.

Hands-on 2. Use class to simplify part 1 (C++ or python)

- You have four random integers: A, B, C, and D and need to update their value so they decrease in order. You and GPT split the responsibility.
- Ask GPT to write three functions:
  - opA(int &a, int &b, int &c, int &d) increase four number by 1,3,1,2
  - opB(int &a, int &b, int &c, int &d) increase four number by 5,1,2,1
  - opC(int &a, int &b, int &c, int &d) increase four number by 1,1,3,0
- Now you write your part, update A, B, C, D.

Hands-on 2. part 1

hands-on-2 > main.cc > ...

```cpp
#include <iostream>

void opA(int &a, int &b, int &c, int &d) {
  a += 1;
  b += 3;
  c += 1;
  d += 2;
}
void opB(int &a, int &b, int &c, int &d) {
  a += 5;
  b += 1;
  c += 2;
  d += 1;
}
void opC(int &a, int &b, int &c, int &d) {
  a += 1;
  b += 1;
  c += 3;
}
void equalize(int &a, int &b, int &c, int &d) {
  a = b = c = d = (a + b + c + d) / 4;
}
```

```cpp
int main() {
  int a = 37, b = 99, c = 1, d = 1355;
  equalize(&: a, &: b, &: c, &: d);
  opA(&: a, &: b, &: c, &: d);
  opA(&: a, &: b, &: c, &: d);
  opA(&: a, &: b, &: c, &: d);
  opB(&: a, &: b, &: c, &: d);
  opB(&: a, &: b, &: c, &: d);
  opB(&: a, &: b, &: c, &: d);
  opA(&: a, &: b, &: c, &: d);
  std::cout << a << " " << b << " " << c <<
}
```

Why should I write a,b,c,d so many times?

```
24  int main() {
25    int a = 37, b = 99, c = 1, d = 1355;
26    equalize(&: a, &: b, &: c, &: d);
27    opA(&: a, &: b, &: c, &: d);
28    opA(&: a, &: b, &: c, &: d);
29    opA(&: a, &: b, &: c, &: d);
30    opB(&: a, &: b, &: c, &: d);
31    opB(&: a, &: b, &: c, &: d);
32    opB(&: a, &: b, &: c, &: d);
33    opA(&: a, &: b, &: c, &: d);
34    std::cout << a << " " << b << " " << c <
35  }
```

hands-on-2 > main-class.cc > ...

```
1  class Op {
2  public:
3    void opA();
4    void opB();
5    void opC();
6    void opD();
7
8    int m_a;
9    int m_b;
10   int m_c;
11   int m_d;
12 };
```

The class in C++ / python

- We can create an object according to a class.
- Function of an object (opA() etc.) automatically know all quantities owned by the object (m_a etc.).
- Look how elegant is opA() now.

Hands-on 2. Use class to simplify part 2 (C++ or python)

- You have four random integers: A, B, C, and D and need to update their value so they decrease in order. You and GPT split the responsibility.
- Ask GPT to write three functions:
  - opA(int &a, int &b, int &c, int &d) increase four number by 1,3,1,2
  - opB(int &a, int &b, int &c, int &d) increase four number by 5,1,2,1
  - opC(int &a, int &b, int &c, int &d) increase four number by 1,1,3,0
- Now you write your part, update A, B, C, D.
- Now rewrite with class.

Hands-on 2. part 1

hands-on-2 > main-class.cc > ...

```cpp
class Op {
public:
  void opA();
  void opB();
  void opC();
  void equalize();

  int m_a;
  int m_b;
  int m_c;
  int m_d;
};

void Op::opA() {
  m_a += 1;
  m_b += 3;
  m_c += 1;
  m_d += 2;
}
```

```cpp
void Op::opB() {
  m_a += 5;
  m_b += 1;
  m_c += 2;
  m_d += 1;
}
void Op::opC() {
  m_a += 1;
  m_b += 1;
  m_c += 3;
}
void Op::equalize() { m_a = m_b = m_c = m_d = (m_a + m_
```

```cpp
int main() {
  Op op;
  op.m_a = 37;
  op.m_b = 99;
  op.m_c = 1;
  op.m_d = 1355;
  op.equalize();
  op.opA();
  op.opA();
  op.opA();
  op.opB();
  op.opB();
  op.opB();
  op.opC();
  std::cout << op.m_a << " " << op.m_b << "
            << std::endl;
}
```

Topic 3

Reuse previous packages

Inheritance

```
1   class Animal {
2   public:
3     void eat();
4   };
5
6   class Cat : public Animal {
7   public:
8     void meow();
9   };
```

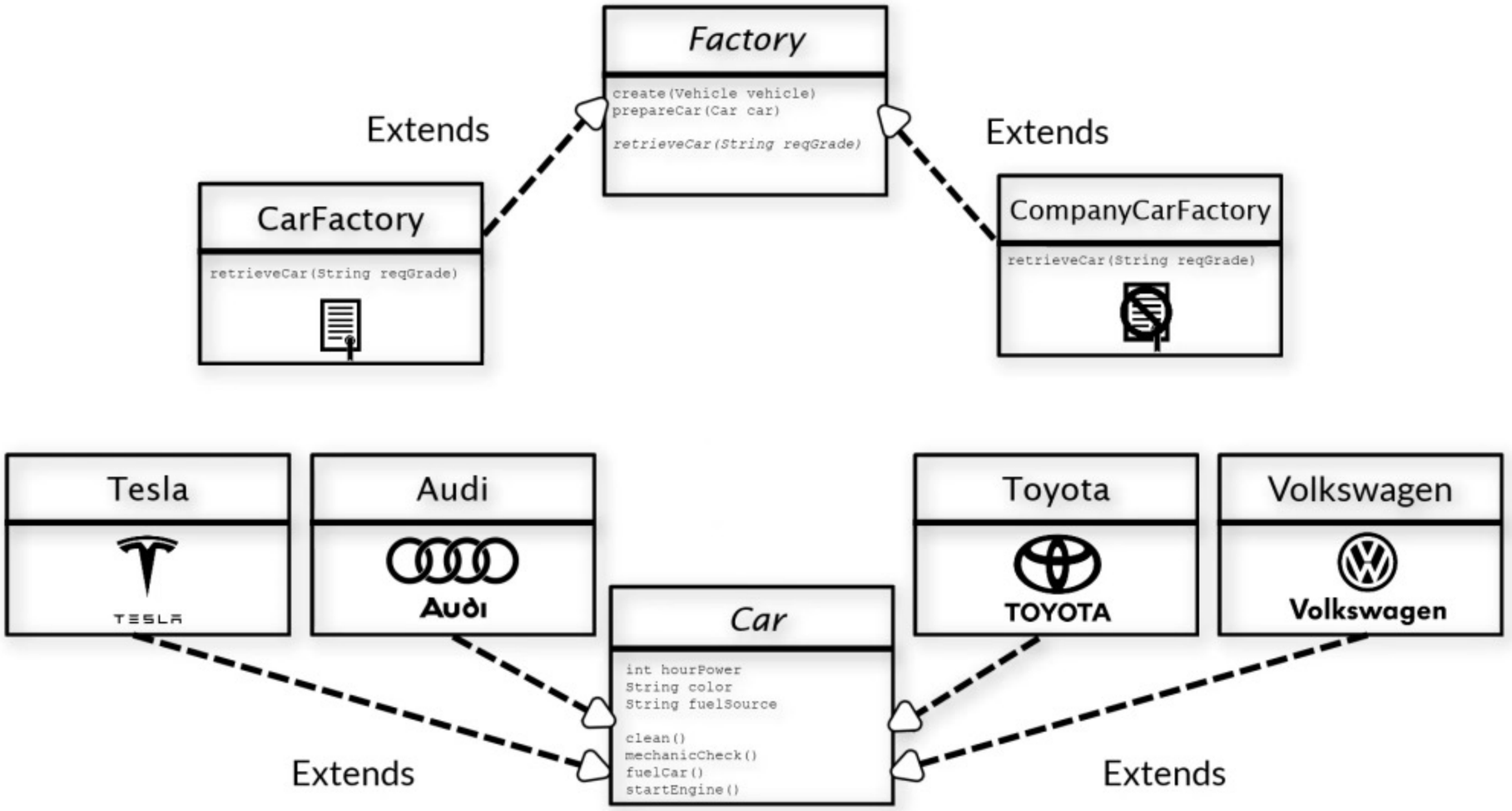
Cat inherit animal
So it eat automatically

Polymorphism

```
1   class Animal {
2   public:
3     virtual void speak();
4   };
5
6   class Cat : public Animal {
7   public:
8     void speak();
9   };
10
11  int main() {
12      Animal *a = new Cat();
13      a->speak();
14  }
```

Override the behavior without touching code

Design pattern



Credit: https://integu.net/factory-pattern/

Topic 4

C++ standard and magic spell

C++11: auto

Type less, mistake less.

```
1    class RunManager {};
2    int main() {
3      auto *run: RunManager * = new RunManager();
4      return 0;
5    }
```

C++11/14: smart pointers

```cpp
#include <iostream>
#include <memory>

class RunManager {
public:
  ~RunManager() { std::cout << "Destructed" << std::endl; }
};
int main() {
  auto run = std::make_unique<RunManager>();
  return 0;
}
```

```
dingxf@Xuefengs-MacBook-Pro build % make
[ 50%] Built target c++11
[ 75%] Building CXX object example/CMakeFiles/c++14.dir/c++14.cc.o
[100%] Linking CXX executable c++14
[100%] Built target c++14
dingxf@Xuefengs-MacBook-Pro build % ./example/c++14
Destructed
```

Automatically delete

So you program does not die due to too much memory

C++11: std::array and for(auto ..

```cpp
#include <array>
#include <iostream>

int main() {
  std::array<int, 4> arr = {[0]=1, [1]=2, [2]=3, [3]=4};
  for (auto a: value_type : arr) {
    std::cout << a << std::endl;
  }

  return 0;
}
```

Avoid C-array

Avoid segmentation fault!

C++14: super lambda

```
1  #include <cmath>
2  #include <iostream>
3  int main() {
4      auto compare: (lambda) = [](auto a, auto b) -> bool { return fabs(a) < fabs(b); };
5      std::cout << compare(-1, 2) << std::endl;
6      std::cout << compare(-1.0, 2.0) << std::endl;
7  }
8
```

C++17: filesystem

```cpp
#include <filesystem>
#include <iostream>

int main() {
  std::string pname = "./c++17-file.cc";
  std::filesystem::path p = pname;
  if (std::filesystem::exists(p)) {
    std::cout << pname << " exists" << std::endl;
  } else {
    std::cout << pname << " not found" << std::endl;
  }
  return 0;
}
```

More to read

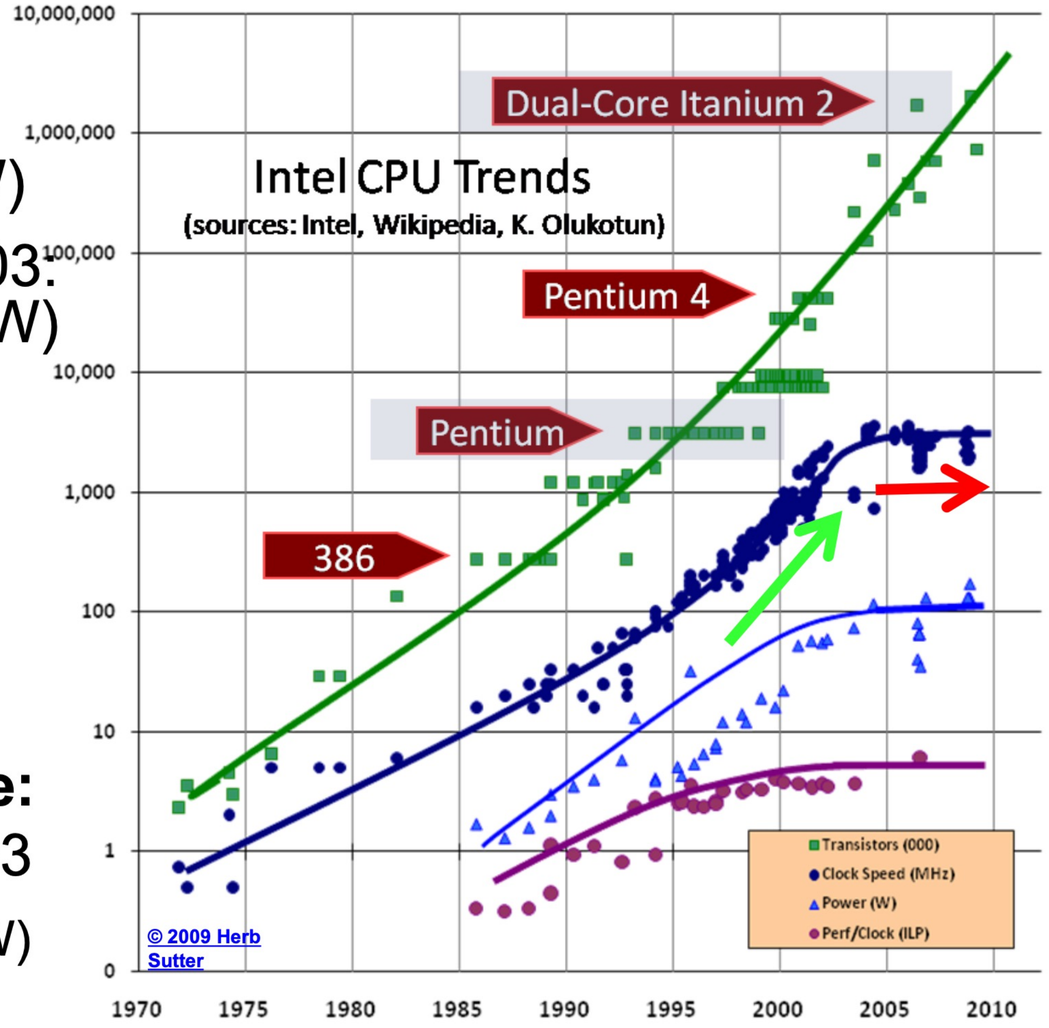
- https://www.geeksforgeeks.org/c-11-vs-c-14-vs-c-17/
- https://en.cppreference.com/w/cpp/language/lambda
- https://learn.microsoft.com/zh-tw/cpp/cpp/lambda-expressions-in-cpp?view=msvc-170
- https://cppcon.org/ C++ international conference

Topic 5

Parallel computing and GPU

Frequency scaling

- **The 7 “fat” years of frequency scaling:**
  - The Pentium Pro in 1996: 150 MHz (12W)
  - The Pentium 4 in 2003: 3.8 GHz (~25x) (115W)
- **Since then**
  - Core 2 systems:
    - ~3 GHz
    - Multi-core
- **Recent CERN purchase:**
  - Intel Xeon E5-2630 v3
    - “only” 2.40 GHz (85W)
    - 8 core



Credit: Vincenzo Innocente and Sverre Jarp

Modern computers are always many cores.

We need to update software to use many cores

Otherwise it's slow:

Europa Universalis IV (Paradox Interactive)

To simulate 1000,000,000 events,

are parallel programming necessary?

No.

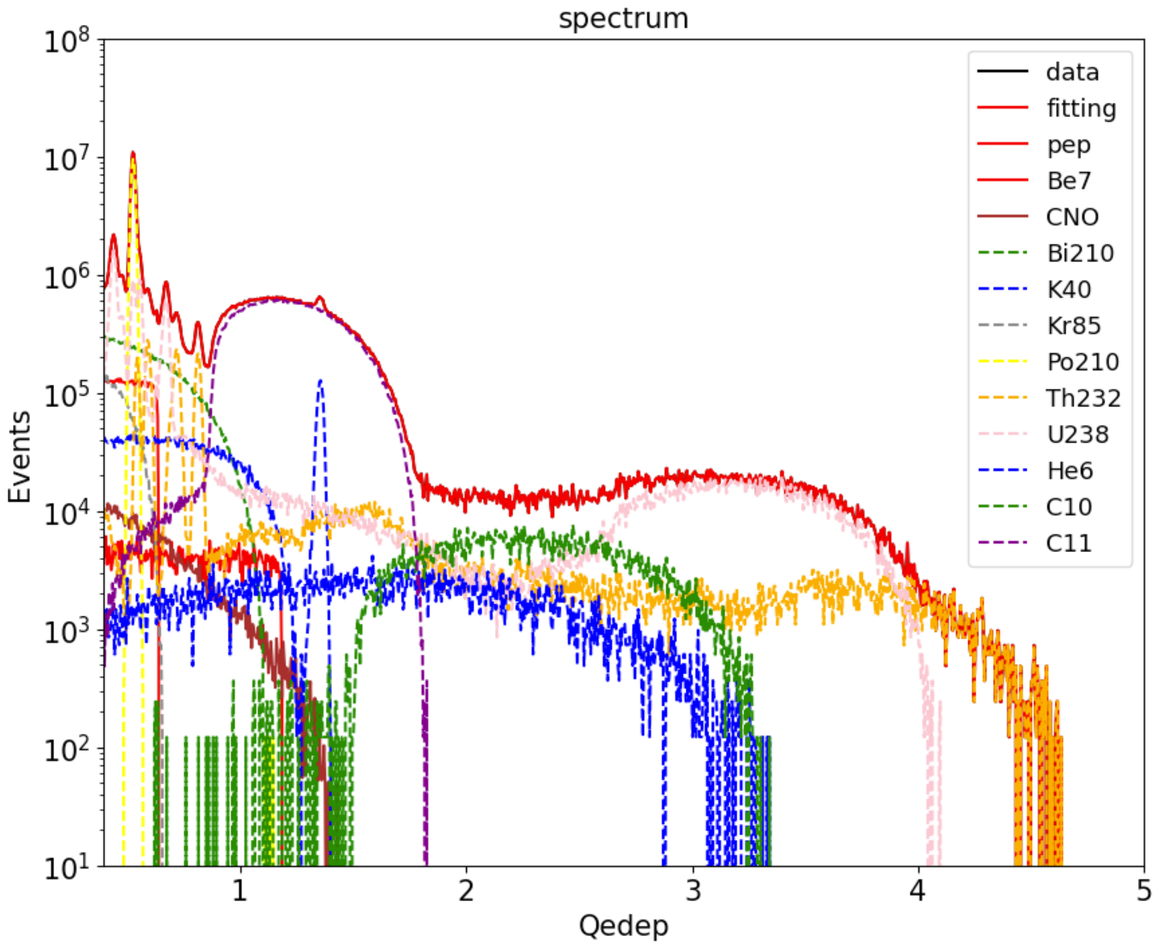
We can run 1000,000,000 jobs simultaneously.

Different from EU IV.

When is parallel necessary?

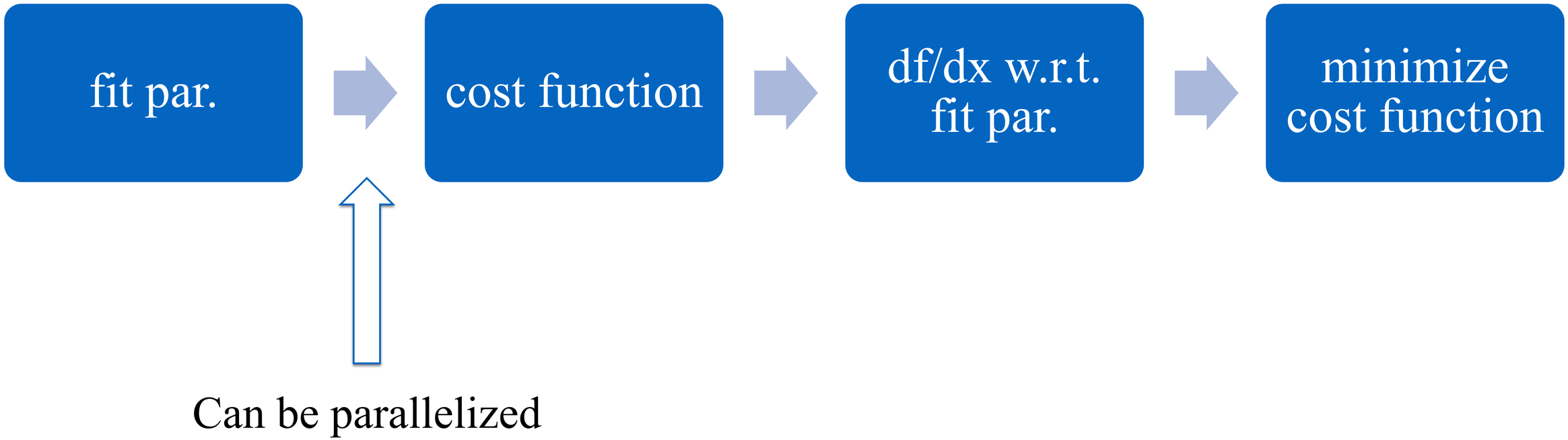
Non-repeating task, such as fitting.

The fitting



- We have model and data of histogram.
- Fit: tune the model parameters, so model matches data.
- By searching parameters that minimize the cost function, usually the log-likelihood, or χ^2.

The fitting


fit par.
cost function
df/dx w.r.t.
fit par.
minimize
cost function
Can be parallelized

What is parallelization?

TRANSFORMATION VS. REDUCTION

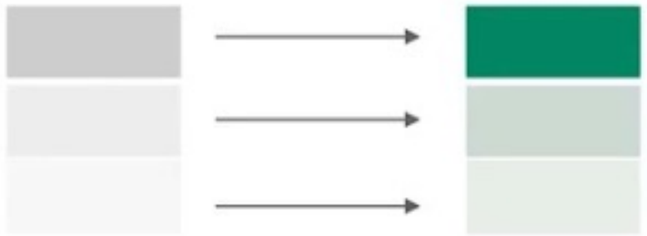
May guide the **thread strategy**: what will each thread do?

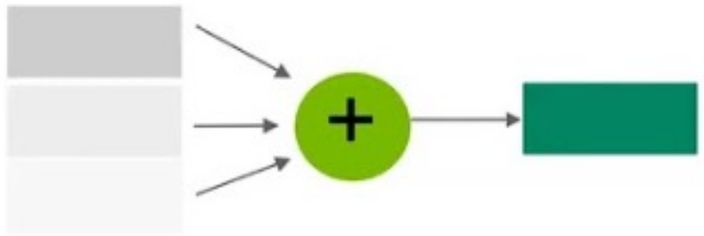
$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

χ^2 = chi squared

O_i = observed value

E_i = expected value





Transformation:

e.g. c[i] = a[i] + 10;

Thread strategy: one thread per output point

Reduction:

e.g. *c = Σ a[i]

Thread strategy: ??

Credit: https://youtu.be/D4l1YMsGNlU?si=T90h-1apP84IjLFZ

Chi-square computation can be parallelized

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

χ^2 = chi squared

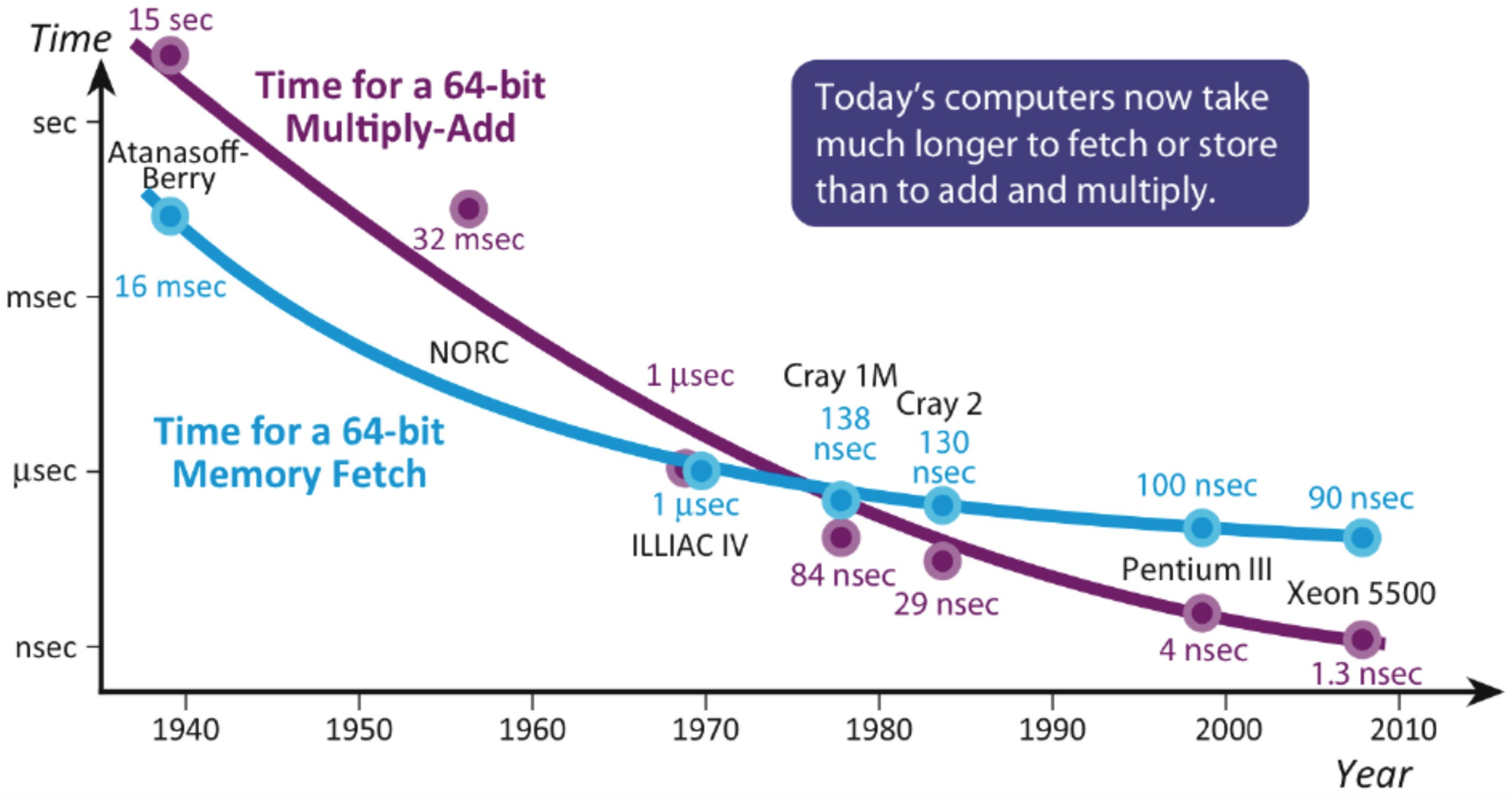
O_i = observed value

E_i = expected value

What is GPU and why GPU

- Graphical Processing Unit
- Optimized for parallel computing
- Much much cheaper : 1 GPU ~ 1000 CPU

The memory fetching barrier

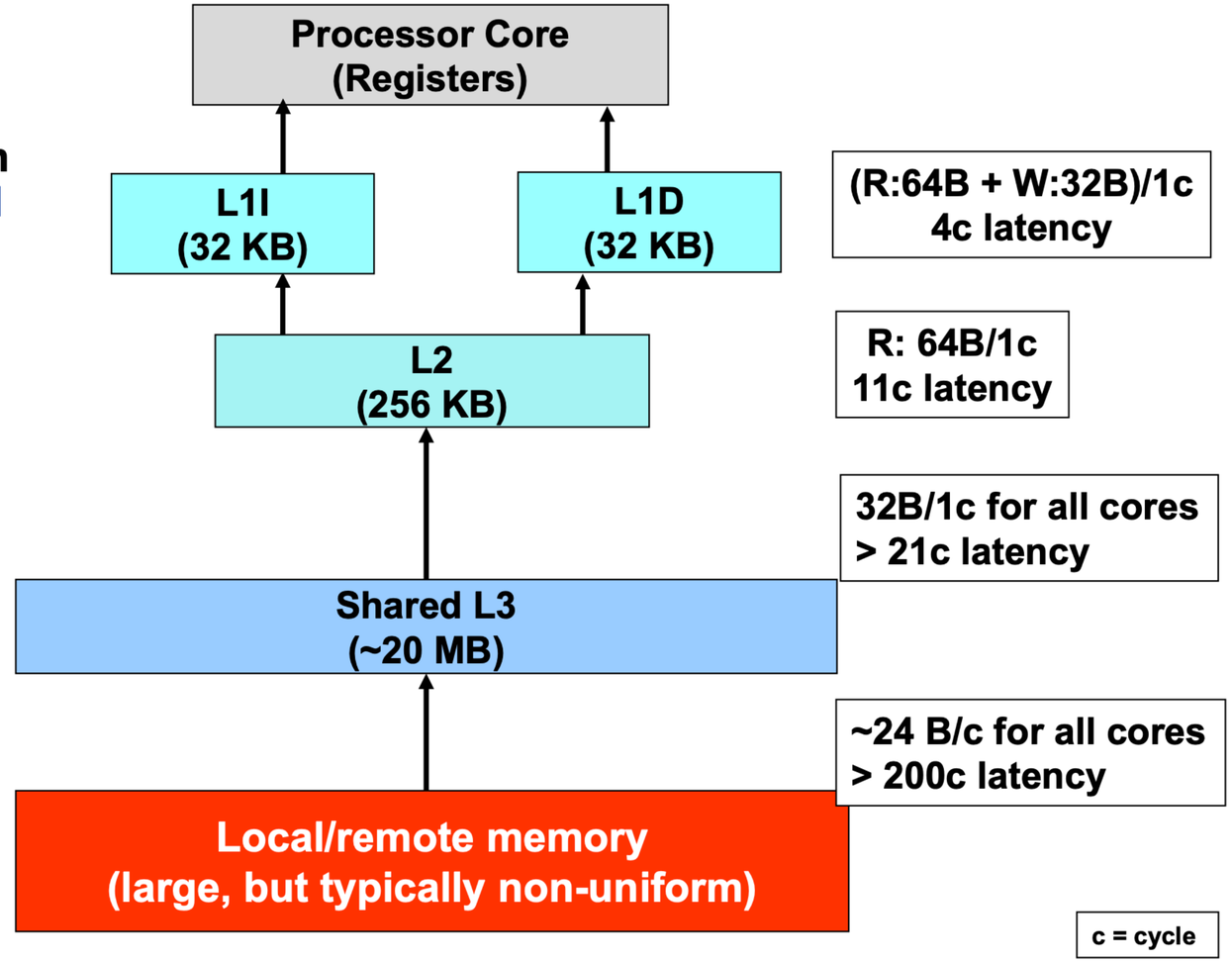


Credit: Vincenzo Innocente and Sverre Jarp

The memory fetching models (CPU)

From CPU to main memory on a recent Haswell processor

- With multicore, memory bandwidth is shared between cores in the same processor (socket)



Credit: Vincenzo Innocente and Sverre Jarp

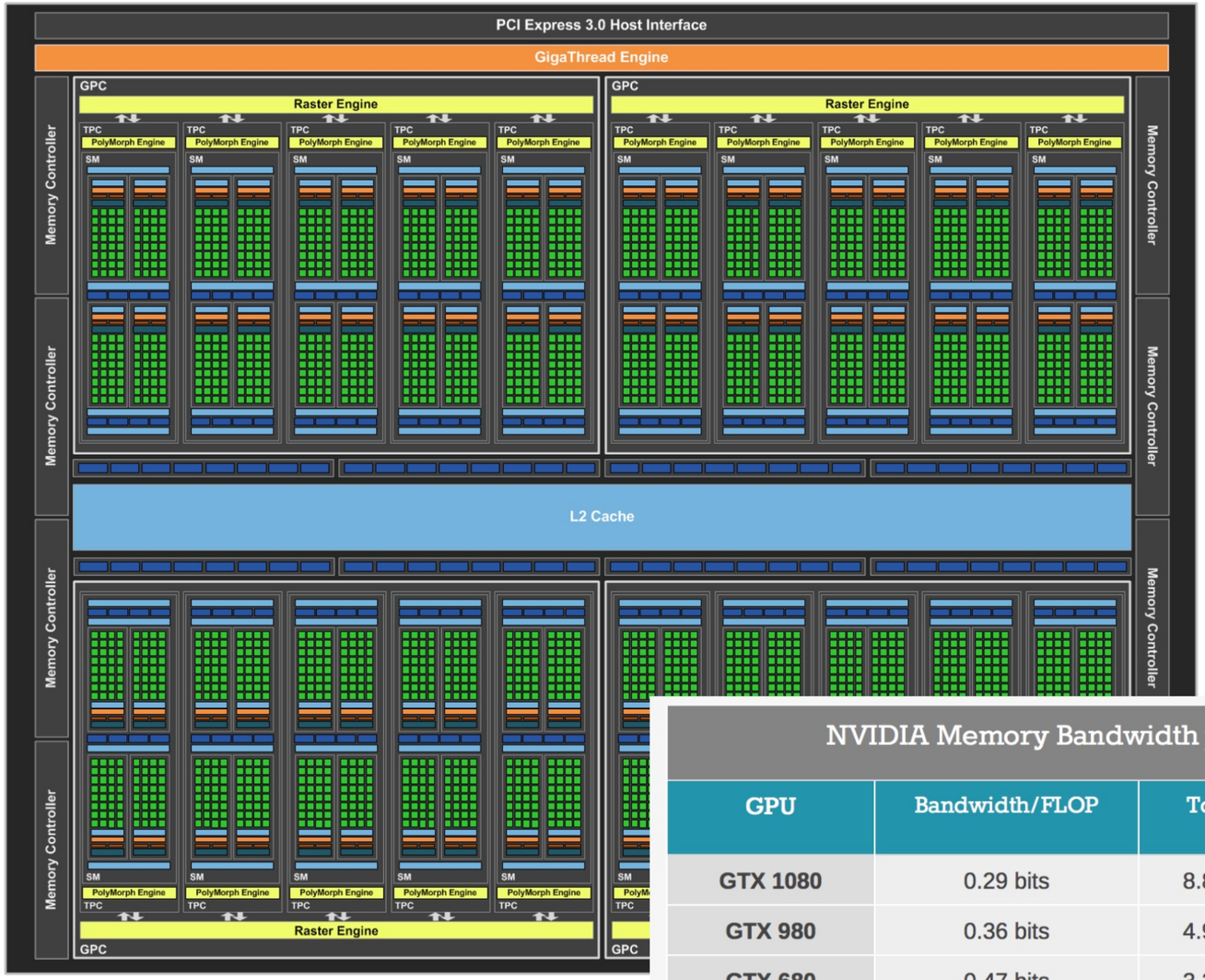
Cache lines (CPU)

- **When a data element or an instruction is requested by the processor, a cache line is ALWAYS moved (as the minimum quantity), usually to Level-1**

- **A cache line is a contiguous section of memory, typically 64B in size (8 * double) and 64B aligned**
  - A 32KB Level-1 cache can hold 512 lines
- **When cache lines have to be moved come from memory**
  - Latency is long (>200 cycles)
    - It is even longer if the memory is remote
  - Memory controller stays busy (~8 cycles)

Credit: Vincenzo Innocente and Sverre Jarp

Streaming Multiprocessor Architecture



NVIDIA Pascal
32 CUDA core
x4 x5 x4 = 2560
Floating Point
Units @1.7GHz
8GB fast memory

NVIDIA Memory Bandwidth per FLOP (In Bits)

GPU	Bandwidth/FLOP	Total FLOPs	Total Bandwidth
GTX 1080	0.29 bits	8.87 TFLOPs	320GB/sec
GTX 980	0.36 bits	4.98 TFLOPs	224GB/sec
GTX 680	0.47 bits	3.25 TFLOPs	192GB/sec
GTX 580	0.97 bits	1.58 TFLOPs	192GB/sec

Credit: Vincenzo Innocente and Sverre Jarp

Why GPU is faster

- GPU: fetch a cache line and share it with 32 SP in 1 SM. 1 fetch in total.
- CPU: 32 CPU each fetch a cache line and does not share. 32 fetch in total.

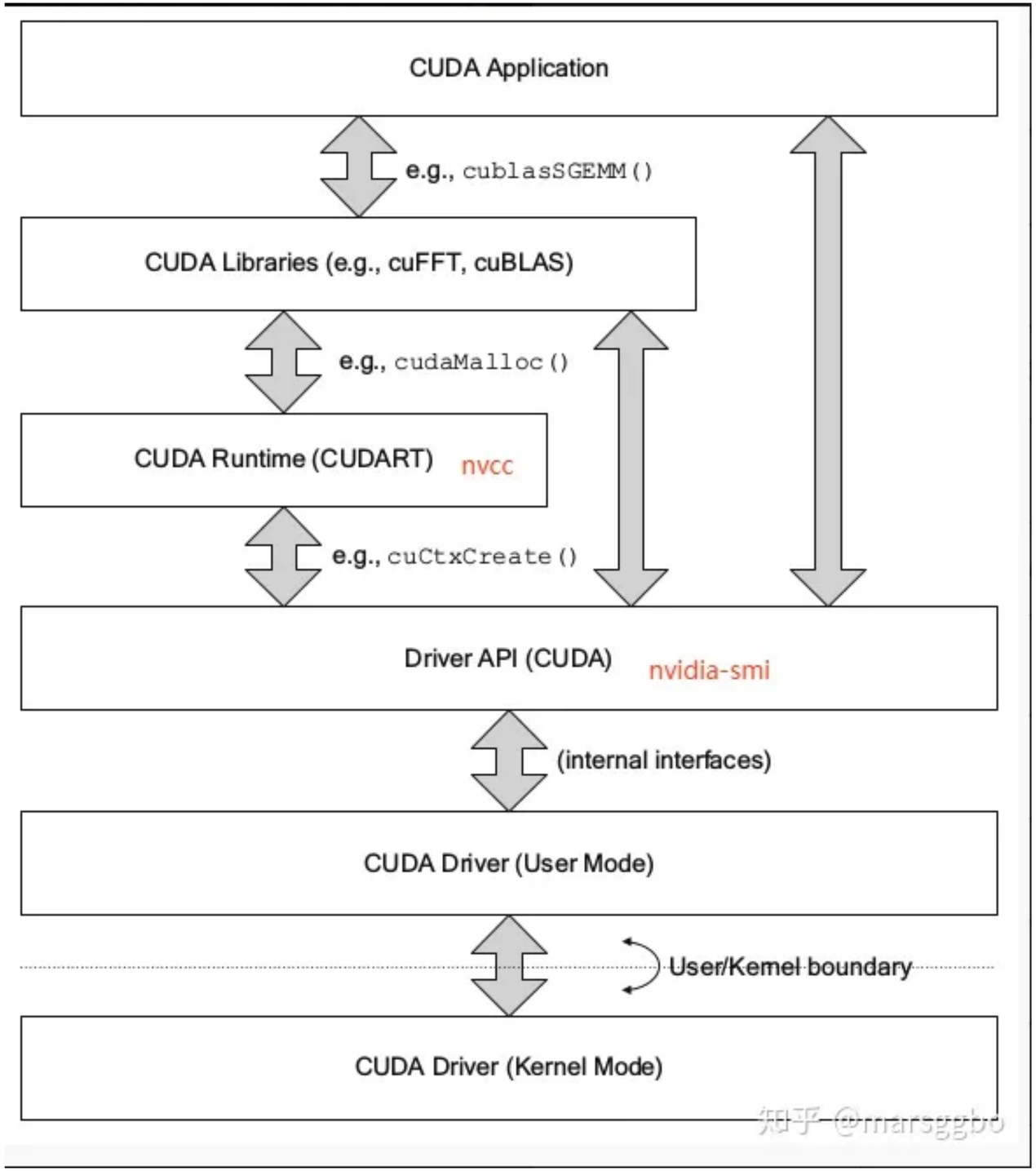
- But GPU should do same task, otherwise slower.
  - Why? Need to do 1-by-1? Or limited by the slowest?

More to read

- https://cvw.cac.cornell.edu/roadmaps
- https://codas-hep.org/
- https://indico.cern.ch/event/1384010/
- https://indico.cern.ch/event/1338689/

Topic 6

CUDA hello world.



The concepts

From driver to cuda

Credit https://zhuanlan.zhihu.com/p/91334380

Pre-requisite

- Install WSL on windows
- Install driver for nVidia GPU card
  - Verify installation: `nvidia-smi`
- Install cudatoolkit for WSL
  - Verify installation: `nvcc -V`

CUDA hello world

```cpp
#include <thrust/device_vector.h>
#include <thrust/transform.h>
int main() {
  std::vector<double> b {3,2,7,4};
  thrust::device_vector<double> a = b;
  thrust::for_each(a.begin(),a.end(),[] __device__ (const double x) {
      printf("hi %lf\n",x); });
  return 0;
}
```

```
~/neutrino-summer-camp/teacher/cuda-hello-world
autodl-container-3fa34eb518-1fabc1e7>nvcc --extended-lambda -o thrust thrust.cu
```

Hands-on 3. Use CUDA to calculate 1+2+3+…+100

- Tips: Use `reduce`
- Check the manual: https://nvidia.github.io/cccl/thrust/api/function_group__reductions_1ga52c9e815fc0bfa4b3a8bc8a2315016dc.html
- You may also ask GPT, or search zhihu, stackoverflow etc.

Hands-on 3 solution. Use CUDA to calculate 1+2+3+…+100

```
#include <iostream>
#include <thrust/iterator/counting_iterator.h>
#include <thrust/reduce.h>
int main() {
  std::cout<<"sum is "<<thrust::reduce(thrust::counting_iterator<int>(1),
      thrust::counting_iterator<int>(101))<<std::endl;
  return 0;
}
```

```
~/neutrino-summer-camp/teacher/hands-on-3
autodl-container-3fa34eb518-1fabc1e7>nvcc -o sum sum.cu
```

Topic 7

Finals: ~~Solar neutrino fit~~ Gaussian fit

Story of GPU fitter

科研 博士 理论物理 深度学习（Deep Learning） 科研成果

同组博士师兄的结果复现不出来，我应该怎么办?

今年研二，老师给了一个课题，让我接着师兄的工作继续做。但是师兄的结果我复现不出来，师兄不愿意给我代码。前段时间因为马上要硕士生开题，所以我骗了老师说我可以复现出来结果了。但是现在我继续按着我自己的仿真做下去，和师兄的现象完全不一样，我该怎么办比较好

已关注 编辑回答 邀请回答 好问题 113 5 条评论 分享 ···

查看全部 117 个回答

雪峰

普林斯顿大学 粒子天体物理博士后

杨知守、Serendipity 等 1809 人赞同了该回答

我刚进组的时候代码就是祖传了20年的拟合代码。

然后我给祖传代码写了unittest，integration test，还有 continuous integration，并且重要的的文章用的代码都会打tag。

大部分分析我都写了singularity recipe，杜绝os/compiler导致的结果的变化。

c++程序都用google sanitizer和valgrind查过memory error；用clang-tidy检查风格问题，用clang-format统一格式，用perf检查性能问题。

我所有的分析代码都用git track。

只要懂一点点软件工程的方法就可以解决重现结果的问题。科研代码是科研的一部分，应该是reproducible的。这本身就是一个科研课题。

A few numbers..

- **For analytical pmt fitting.. how much speed up?**

	original fitter	GooStats	speed up
single fit + dn	**~ 1 hour**	**~4 seconds**	x600
complementary + dn	**~ 2 hours**	**~7 seconds**	x1000
dn+comp.+MV	unknown (**> 1 week**)	**~5 minutes**	> x2000

SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore

History & path

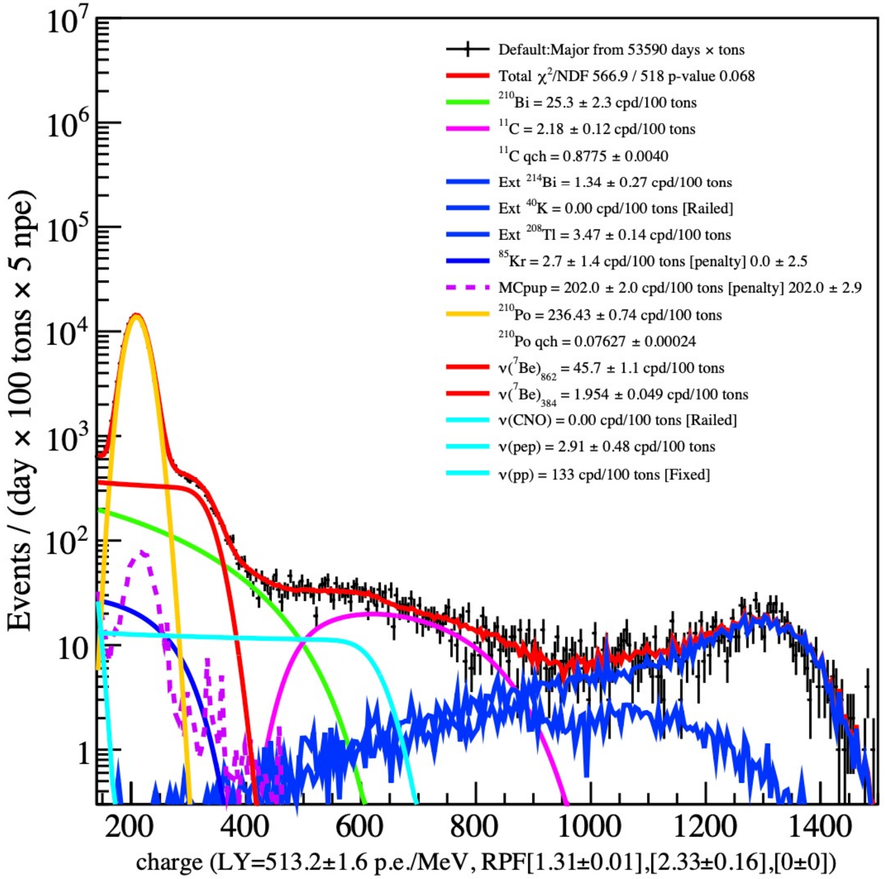
- 2016 Feb. Ilia: I added MINOS option so we can get precise error but it takes 8 hours.. me: hmm??
- 2017 New Year's Eve, GooStats v0.001
- 2017 Feb. 03 bx-GooStats-charge
- 2017 Mar. 19 bx-GooStats-MC-MV
- 2017 Mar. 23 bx-GooStats-npmt
- 2017 April Alina, Omer et al start to produce physics result with bx-GooStats

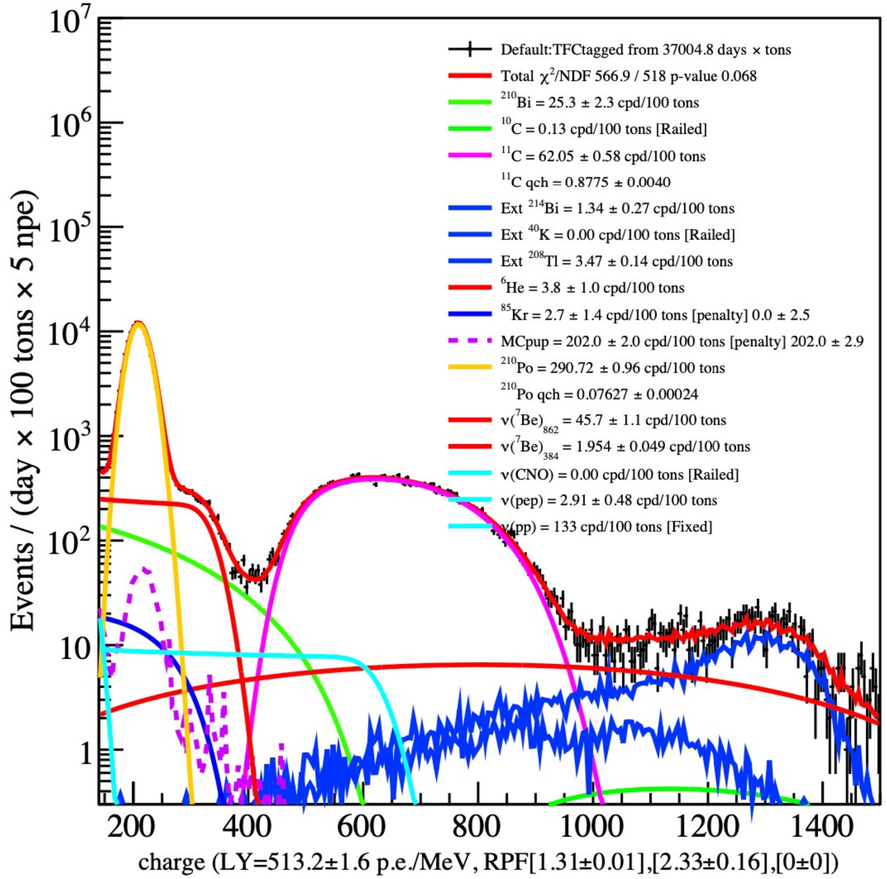
Git version tree

```
* | d73baa5 - (3 months ago) fix the logic in use_MC - Xuefeng Ding
* |   09a7ffd - (3 months ago) Merge branch 'MVfit_dev' into dingxf_dev - Xuefeng Ding
|\ \
| * | fc455c0 - (3 months ago) enable assert by building Debug. update startbin.. etc - Xuefeng Ding
| * | 4ec0eb4 - (3 months ago) bug fix: could not use static.. fix wrong start/end bin - Xuefeng Ding
| * | c8c08e4 - (3 months ago) now the logic is correct.. - Xuefeng Ding
| * | 3b6df49 - (3 months ago) bug fix: use memcp to symbol.. - Xuefeng Ding
| * | 12c2f9e - (3 months ago) bug fix: return -log(L). bug fix: MEMCPY_TO_SYMBOL is operating on symbol. you cannot pass th
* | |   c31cd18 - (3 months ago) Merge branch 'MVfit_dev' into dingxf_dev - Xuefeng Ding
|\ \ \
| |/ /
| * | 582d6e3 - (3 months ago) final bug fix. constant symbol cannot by passed. now you can run it - Xuefeng Ding
| * | 616ab96 - (3 months ago) can compile, yet cannot run. invalid device symbol - Xuefeng Ding
| * | cd67d09 - (3 months ago) add more infrustructure - Xuefeng Ding
| * |   3939786 - (3 months ago) Merge branch 'master' into MVfit_dev - Xuefeng Ding
| |\ \
| | |/
| * |   54cc966 - (3 months ago) Merge branch 'master' into MVfit_dev - Xuefeng Ding
| |\ \
| * | | 3b2c9a9 - (3 months ago) add infrustructure , yet cannot compile - Xuefeng Ding
| * | | 5085285 - (3 months ago) can compile - Xuefeng Ding
| * | |   845992f - (3 months ago) Merge branch 'master' into MVfit_dev - Xuefeng Ding
| |\ \ \
| * | | | 6fc1364 - (3 months ago) add MVPdf - Xuefeng Ding
| * | | | cb15b57 - (3 months ago) add MV fit - Xuefeng Ding
* | | | | b42b811 - (3 months ago) remove comparison with the old sun - Xuefeng Ding
| |_|_|/
|/| | |
* | | | a240d30 - (3 months ago) update ChangeLog - Xuefeng, Ding
* | | | d95969a - (3 months ago) update install_AgostiniPlot.sh - Xuefeng Ding (20170325_cvs_commit)
* | | | 95d287f - (3 months ago) update the instructions - Xuefeng Ding
```

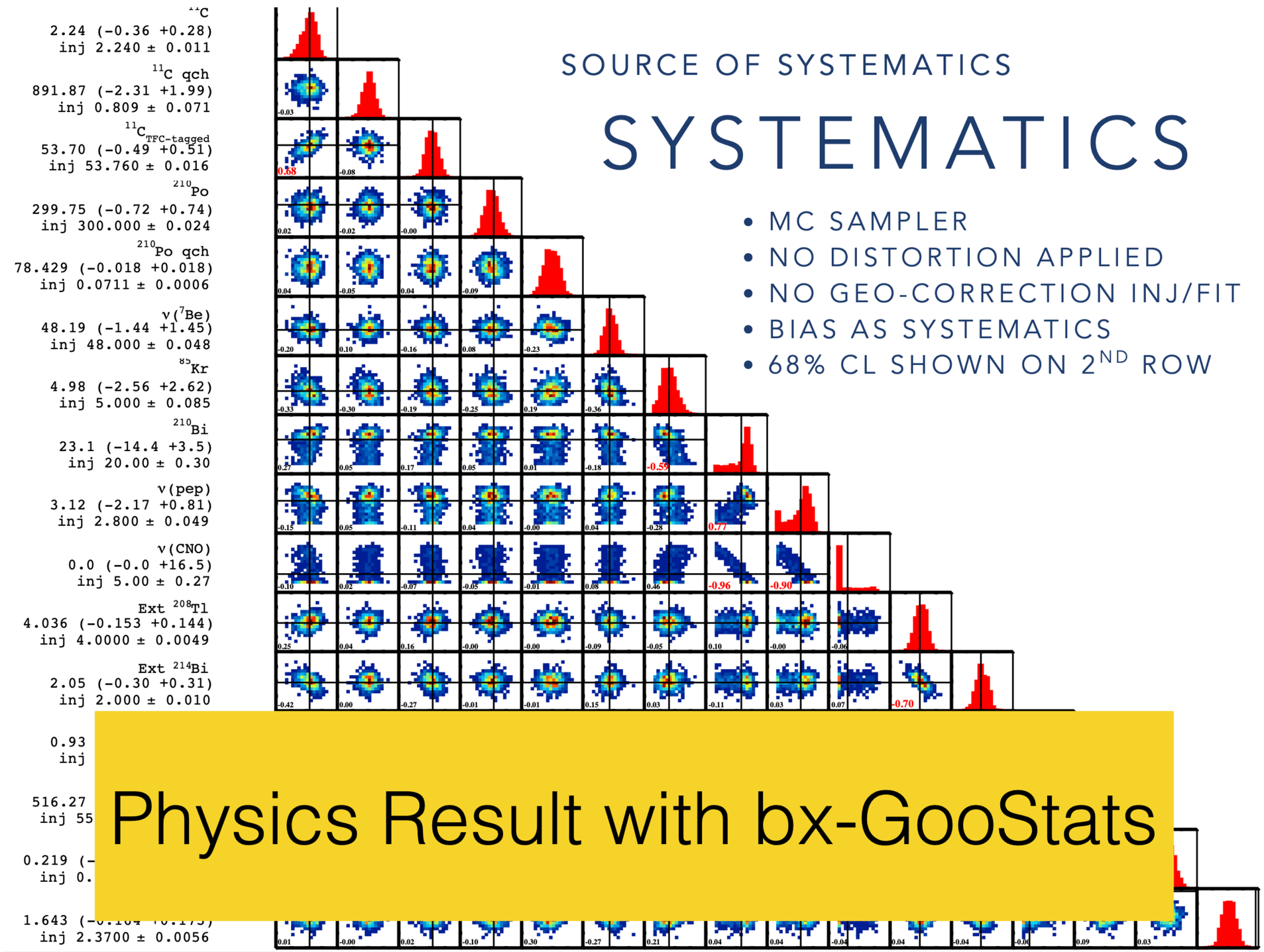
GRAN SASSO SCIENCE INSTITUTE
SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore

Physics Result with bx-GooStats



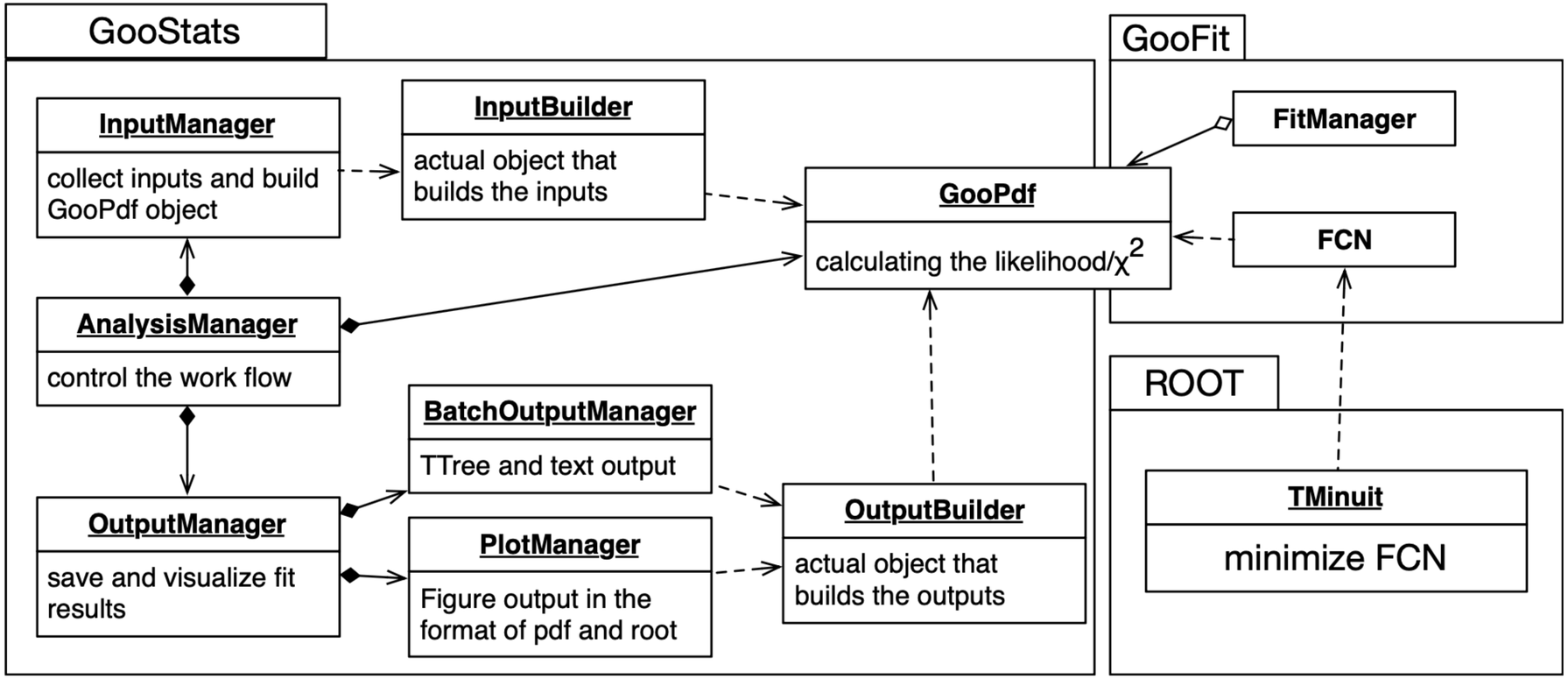


GRAN SASSO SCIENCE INSTITUTE
SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore


^{11}C
2.24 (-0.36 +0.28)
inj 2.240 ± 0.011
^{11}C qch
891.87 (-2.31 +1.99)
inj 0.809 ± 0.071
^{11}C$_{TFC-tagged}$
53.70 (-0.49 +0.51)
inj 53.760 ± 0.016
^{210}Po
299.75 (-0.72 +0.74)
inj 300.000 ± 0.024
^{210}Po qch
78.429 (-0.018 +0.018)
inj 0.0711 ± 0.0006
ν(^{7}Be)
48.19 (-1.44 +1.45)
inj 48.000 ± 0.048
^{85}Kr
4.98 (-2.56 +2.62)
inj 5.000 ± 0.085
^{210}Bi
23.1 (-14.4 +3.5)
inj 20.00 ± 0.30
ν(pep)
3.12 (-2.17 +0.81)
inj 2.800 ± 0.049
ν(CNO)
0.0 (-0.0 +16.5)
inj 5.00 ± 0.27
Ext ^{208}Tl
4.036 (-0.153 +0.144)
inj 4.0000 ± 0.0049
Ext ^{214}Bi
2.05 (-0.30 +0.31)
inj 2.000 ± 0.010
0.93
inj
516.27
inj 55
0.219 (-
inj 0.
1.643 (-
inj 2.3700 ± 0.0056
SOURCE OF SYSTEMATICS
SYSTEMATICS
• MC SAMPLER
• NO DISTORTION APPLIED
• NO GEO-CORRECTION INJ/FIT
• BIAS AS SYSTEMATICS
• 68% CL SHOWN ON 2ND ROW
Physics Result with bx-GooStats

Package Design

- **Middle Layer** between **GooFit** and **User module**



Hands-on-final: Study of impact of binning on Gaussian fit

- Produce a sample of 1000 gaussian distributed numbers
- Fill them into histogram
- Fit the histogram with Gaussian functions
- Repeat it many times
- Fill the fit result into histograms
- How fit result width changes with the binning?
- Tips: You can do it in python first, then try to move to CUDA.

Ad: Research program @ IHEP for undergraduate

- Personal homepage

  http://dingxf.cn/

  https://teacher.ucas.ac.cn/~dingxf
- Undergraduate Innovation and Practice Training Program（科创计划）

  available this year

  https://ihep.cas.cn/edu/bks/kcjh/202407/t20240702_7207509.html
- My project: Search for new physics with Solar neutrinos
  - More other projects see http://dingxf.cn/

Conclusions

- Model computing architecture urges for upgrading of computing program to parallelized.
- With CUDA, we may speed up the fitting process.