

# NeutronHP vs. ParticleHP - TARC

1. Sequential Timing of TARC vs. reference tag
2. Fluence spectra for QGSP\_BIC\_HP vs. data
3. Profiling – gmon/G4PROFILE
4. Profiling – valgrind
5. Fix? MT performance?

Alex Howard, ETH Zürich

# HP Performance

- Miguel reported in Ferrara that he saw a factor of two loss in performance with his n-tof simulations between v10.1 and v10.2
  - \*Running in sequential mode\*
  - Not seen significantly in non-HP physics lists
- Reproduced in TARC for **QGSP\_BIC\_HP**
- Timing varies significantly between v10.1ref03 and v10.2ref08.
  - Between 10.2ref06 and 10.2ref08 only time things got better, but still >50% slower than v10.1ref03
- Investigated for 9 different Geant4 configurations
  - Both MT and sequential
- My resurrected TARC validation was started with v10.1ref09, so after the performance loss occurred
- Neutron\_HP was merged with Particle\_HP during this period...
- Many changes to de\_excitation, precompound
- Not a unit test, so anything in Geant4 could also be responsible (including my application code)
- Very much a qualitative comparison

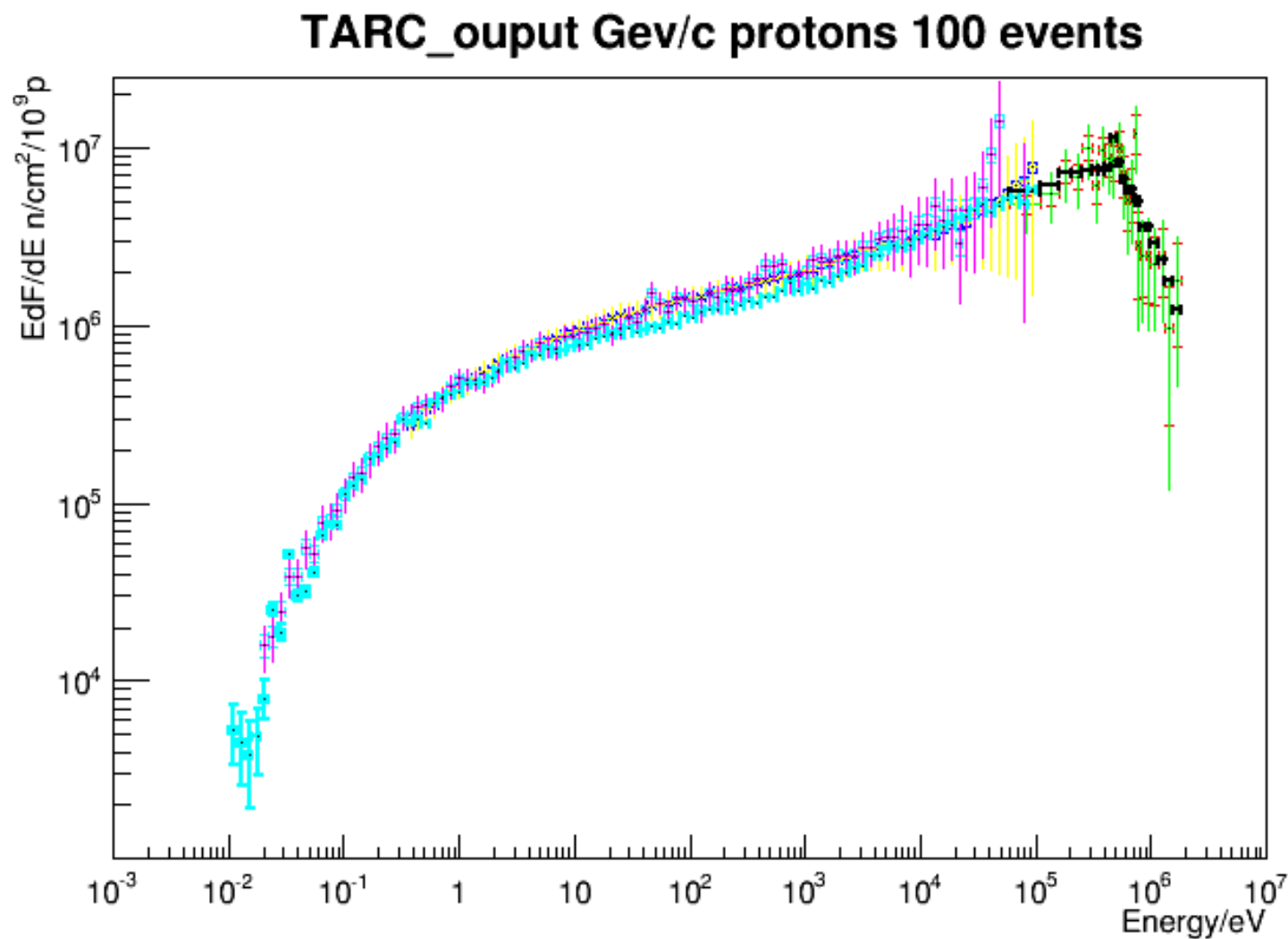
# Geant4 Versions

- 10.1:
  - Ref03, ref04, ref06, ref09
- 10.2:
  - Ref00, Ref06, ref08
  - Ref08 modified
    - particle\_hp replaced with neutron\_hp from 10.1ref03
  - Ref08 “fixed”
    - Removed worker thread flags from HPData classes

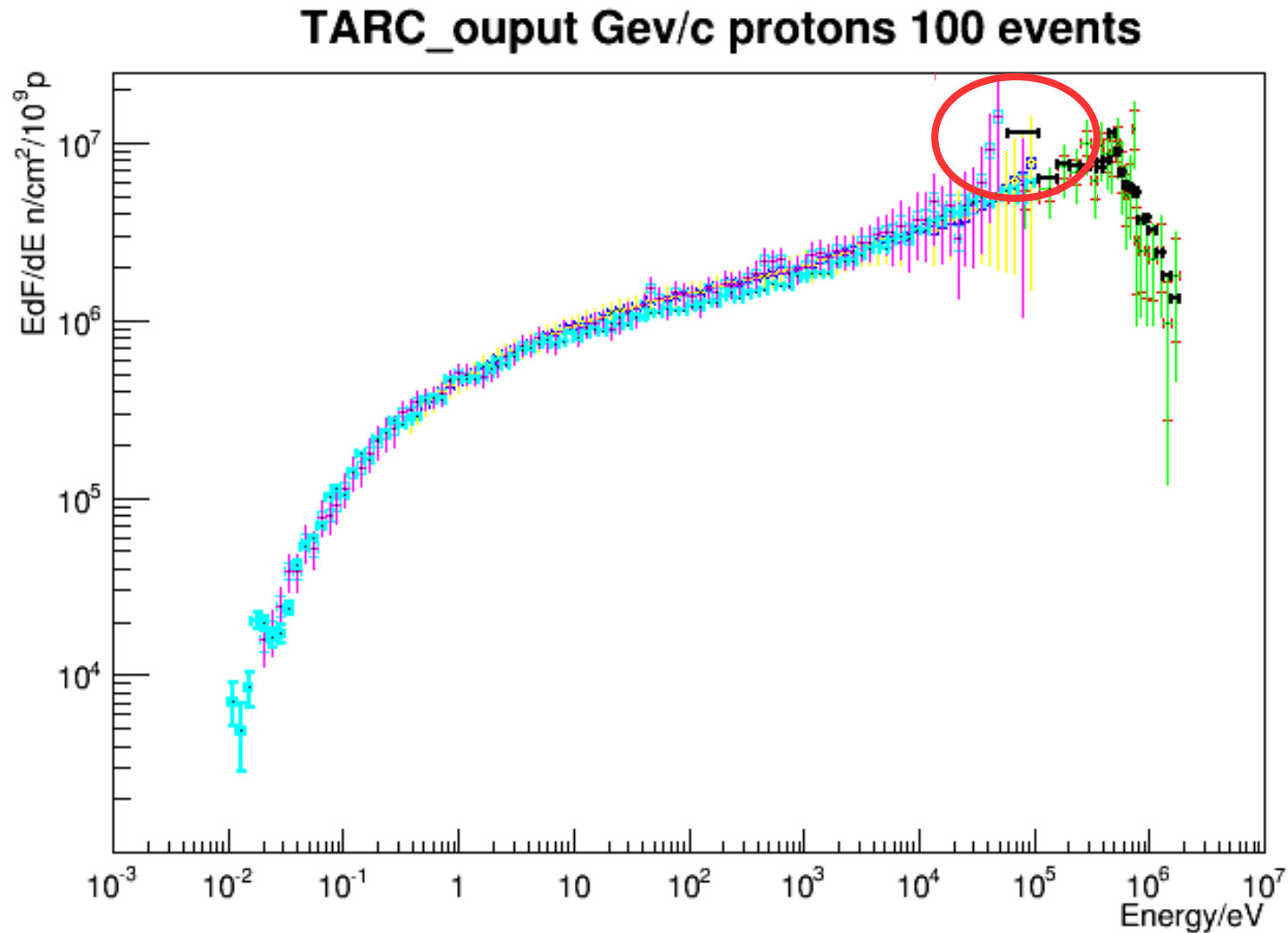
# Timing

G4	10.1ref03	10.1ref04	10.1ref06	10.1ref09	
Sequential	<b>83m10s</b>	80m35s	146m49s	148m43s	
MT (10 threads)	<b>15m15s</b> <b>141m23s</b>	15m19s 138m33s	24m5s 225m3s	25m40s 240m58s	
G4	10.2ref00	10.2ref06	10.2ref08	10.2ref08mod	10.2ref08fix
Sequential	145m43s	143m47s	<b>125m34s</b>	71m7s	<b>68m33s</b>
MT (10threads)	25m15s 237m13s	27m8s 257m8s	<b>22m31s</b> <b>216m53s</b>	16m31s 131m34s	<b>13m24s</b> <b>125m33s</b>

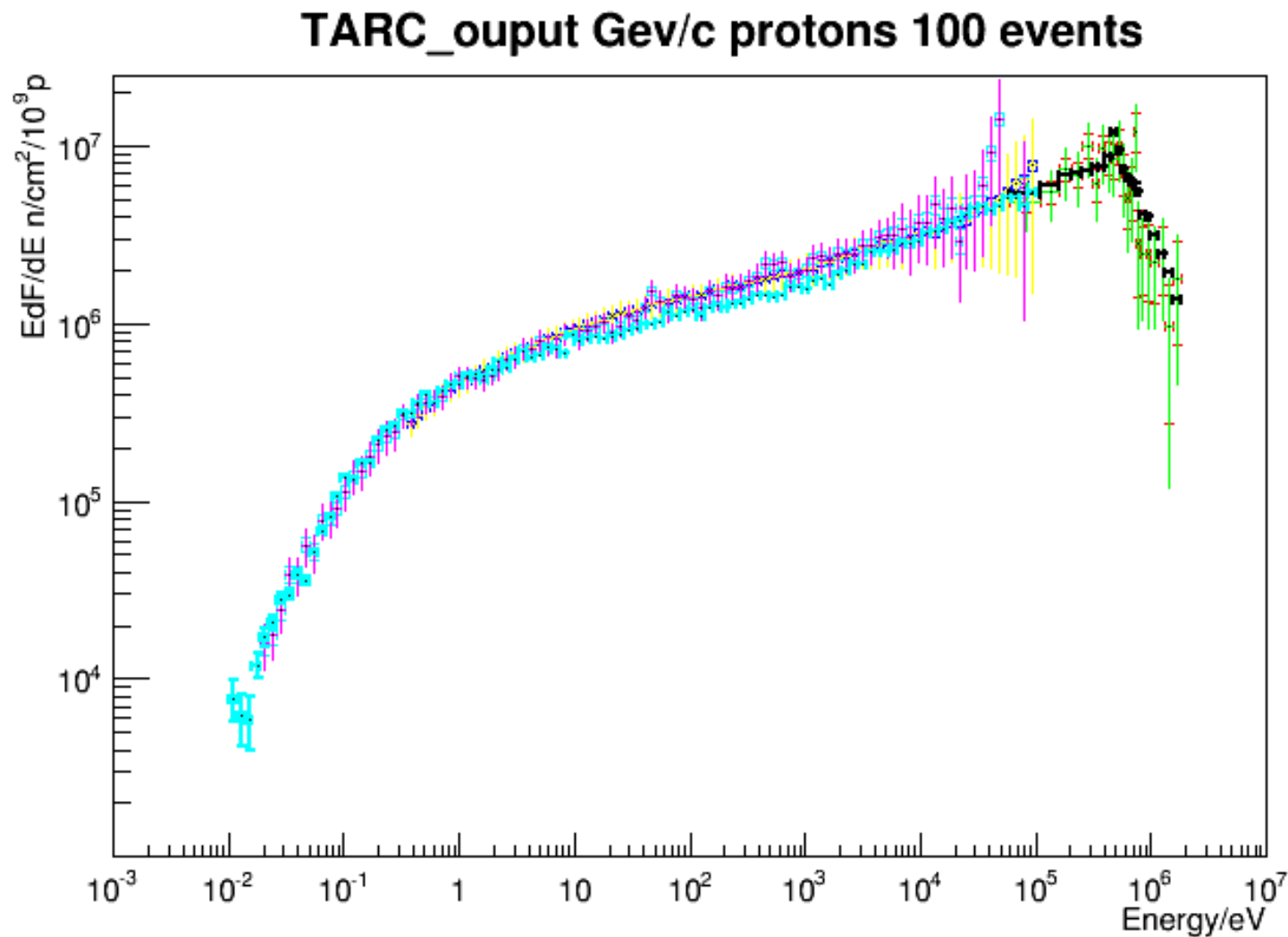
# Fluence Spectrum – 10.1ref03



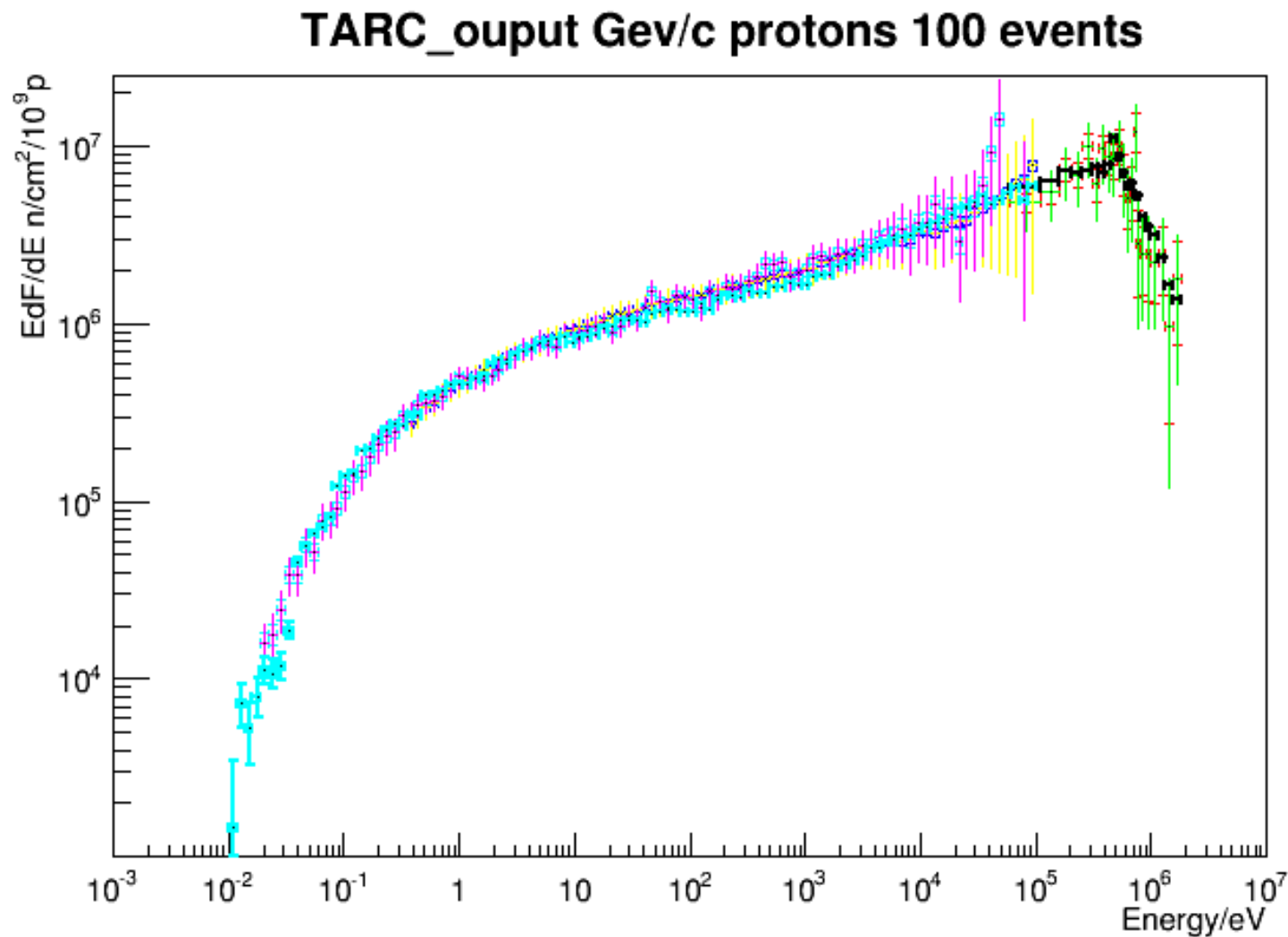
# Fluence Spectrum – 10.1ref03MT



# Fluence Spectrum – 10.1ref04

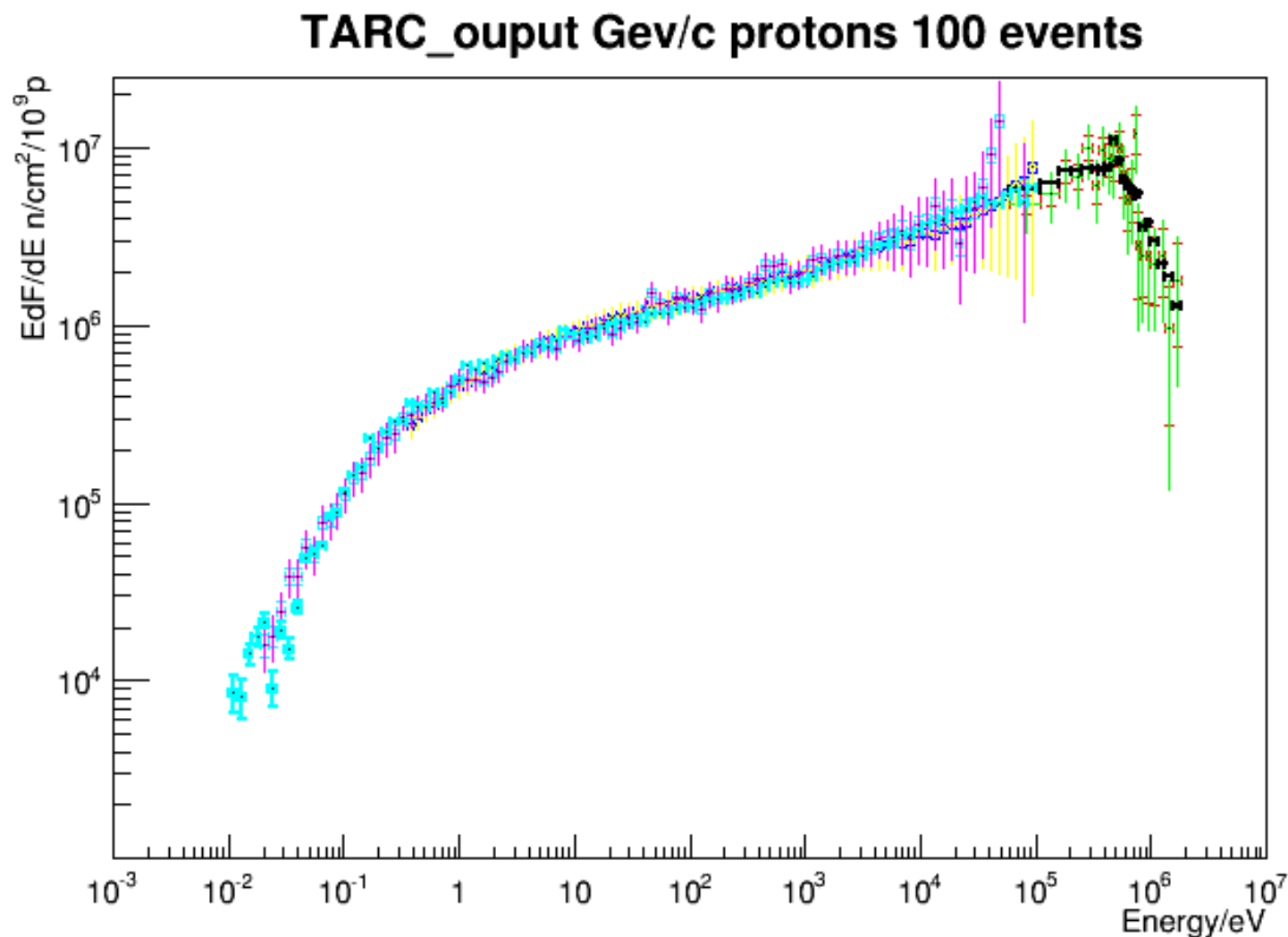


# Fluence Spectrum – 10.1ref06



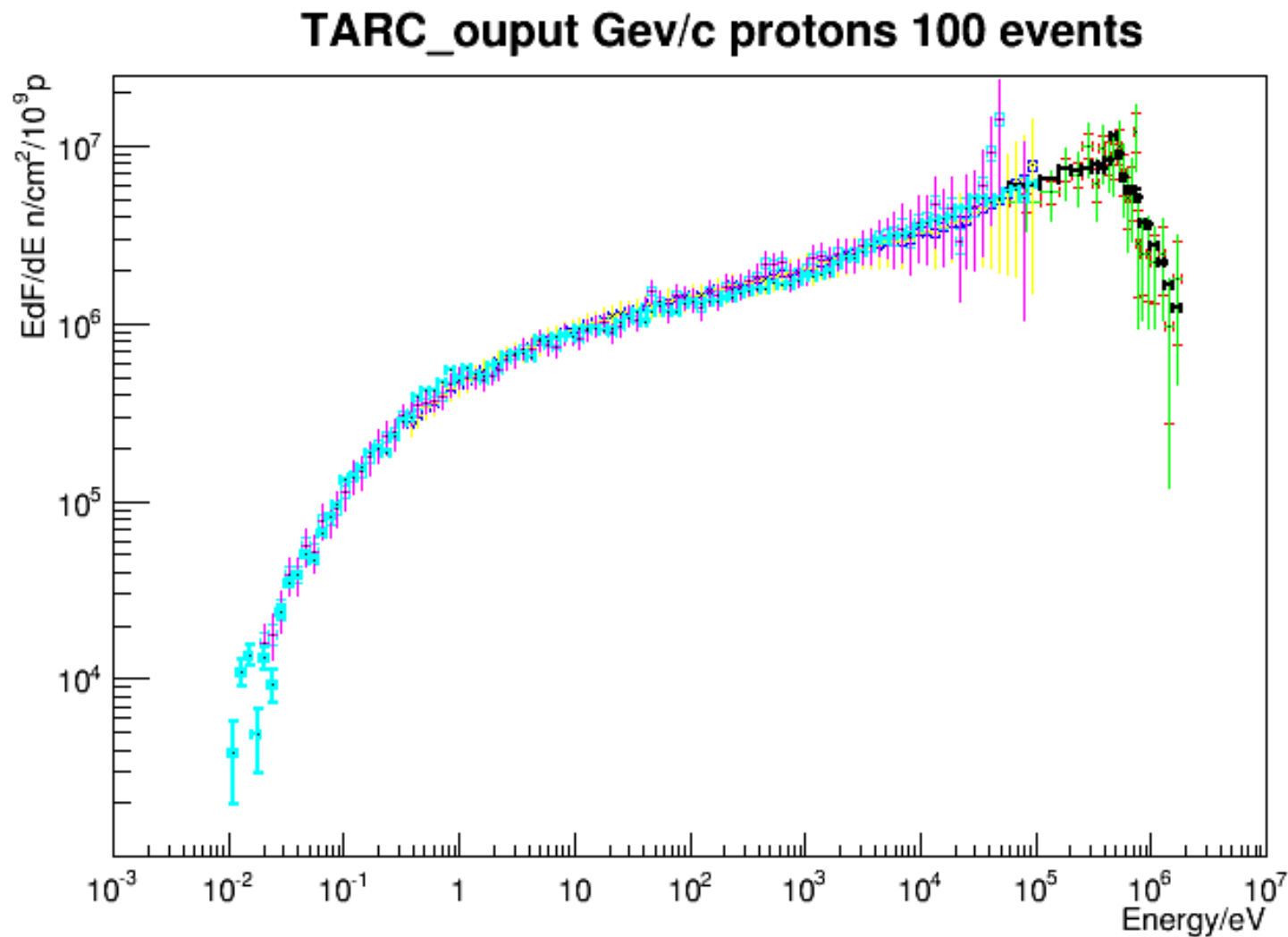


# Fluence Spectrum – 10.1ref09

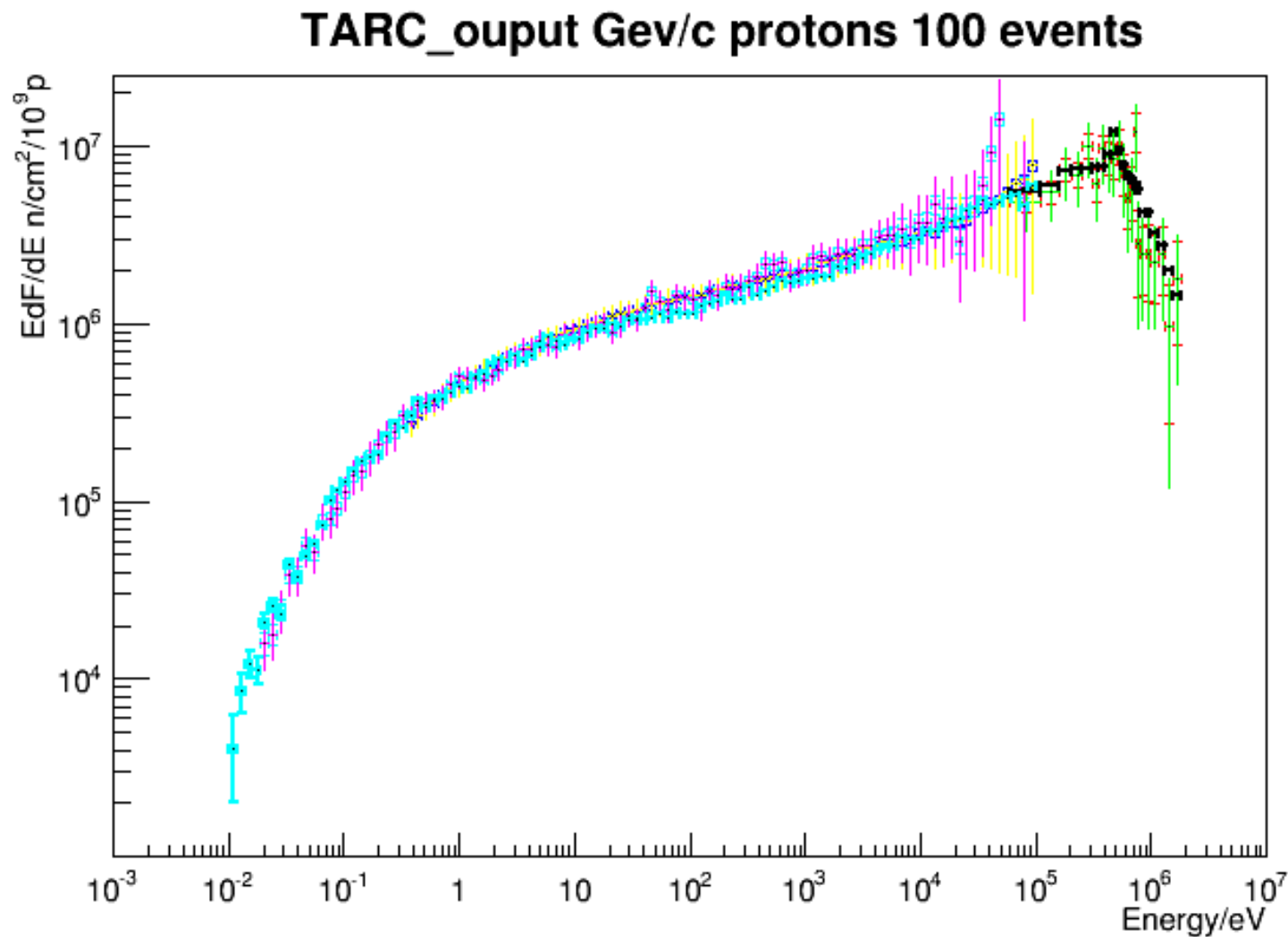


Previous validation of TARC – presentation November 2015...

# Fluence Spectrum – 10.2ref00

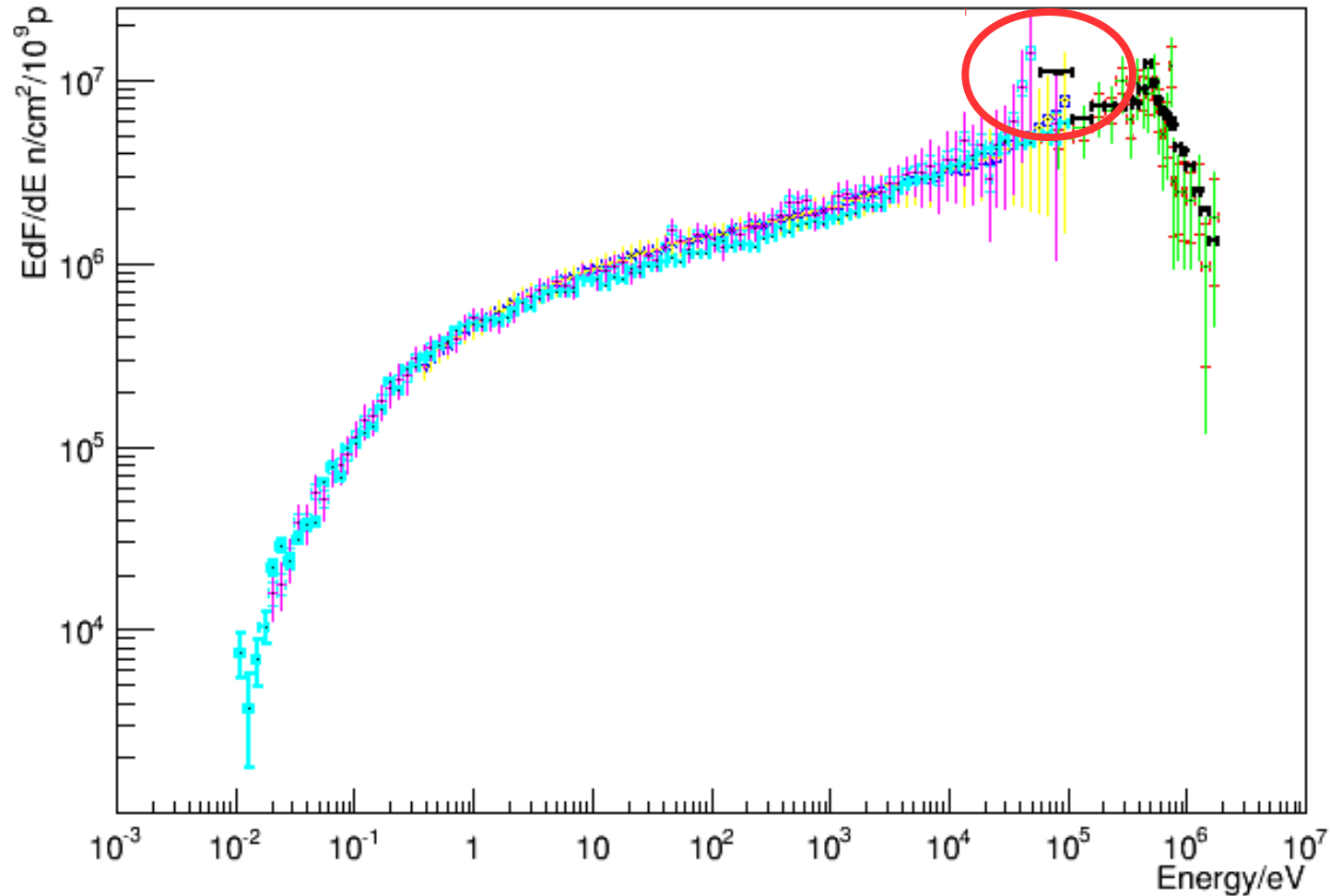


# Fluence Spectrum – 10.2ref06

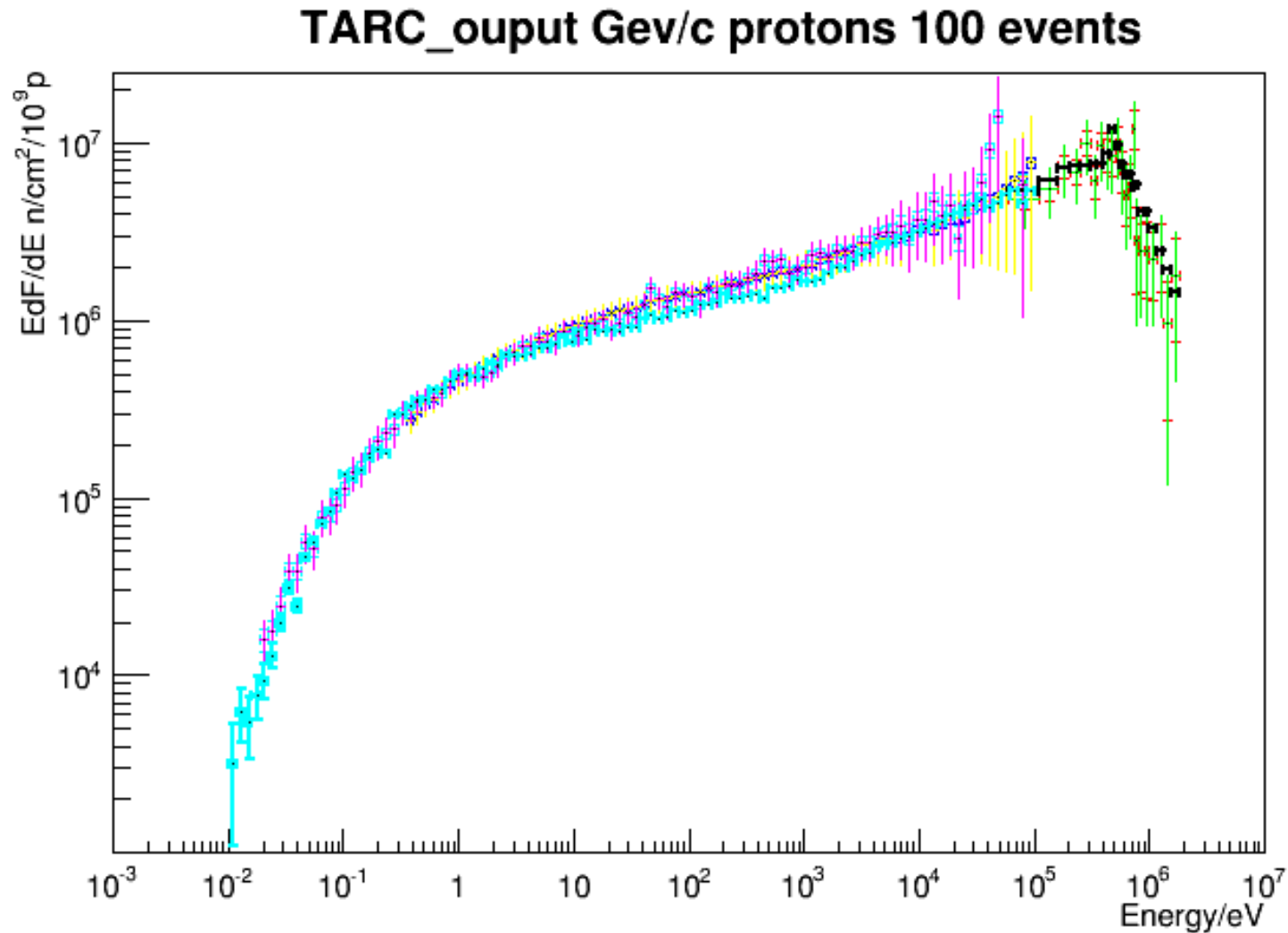


# Fluence Spectrum – 10.2ref06MT

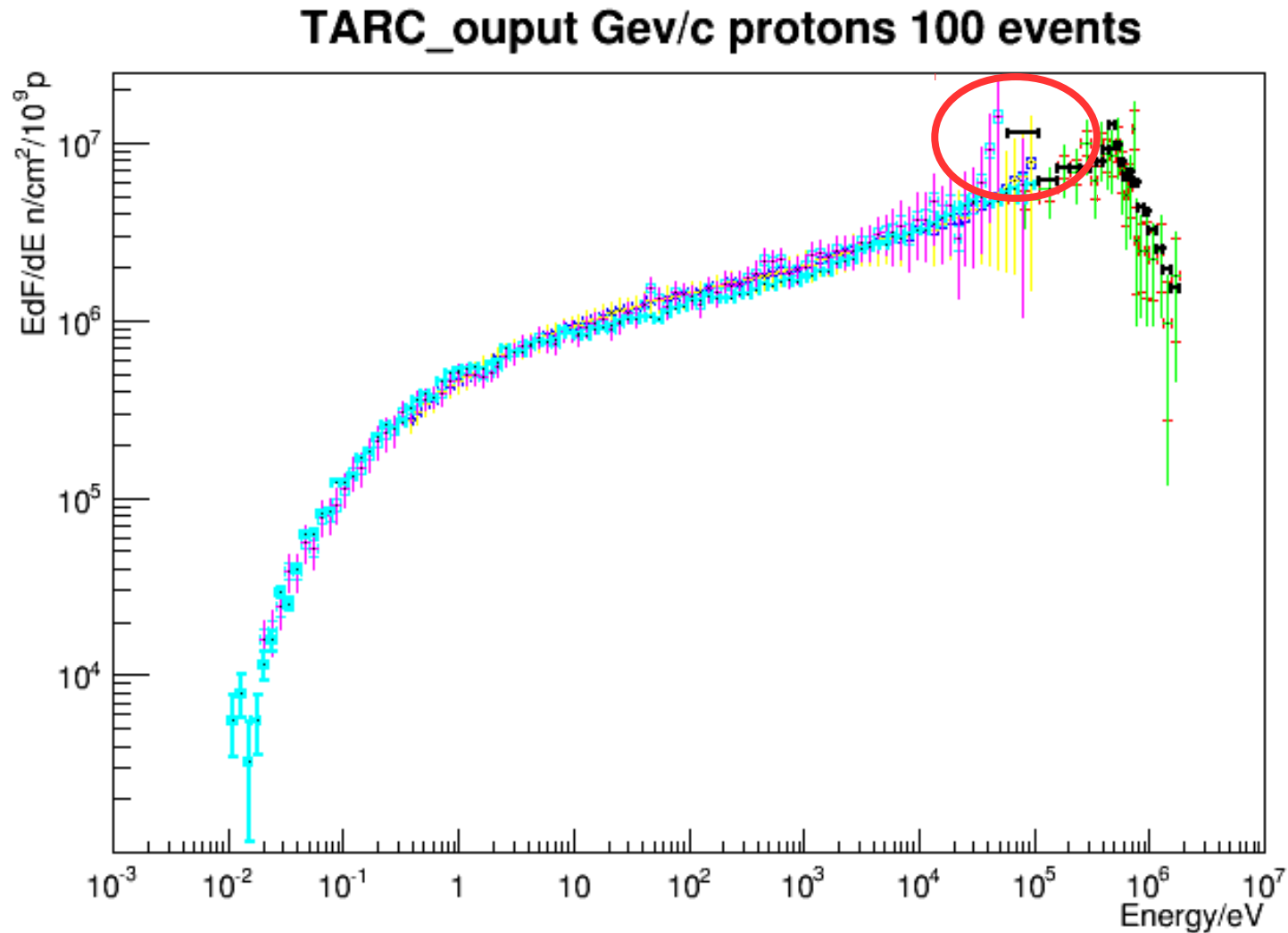
TARC\_output Gev/c protons 100 events



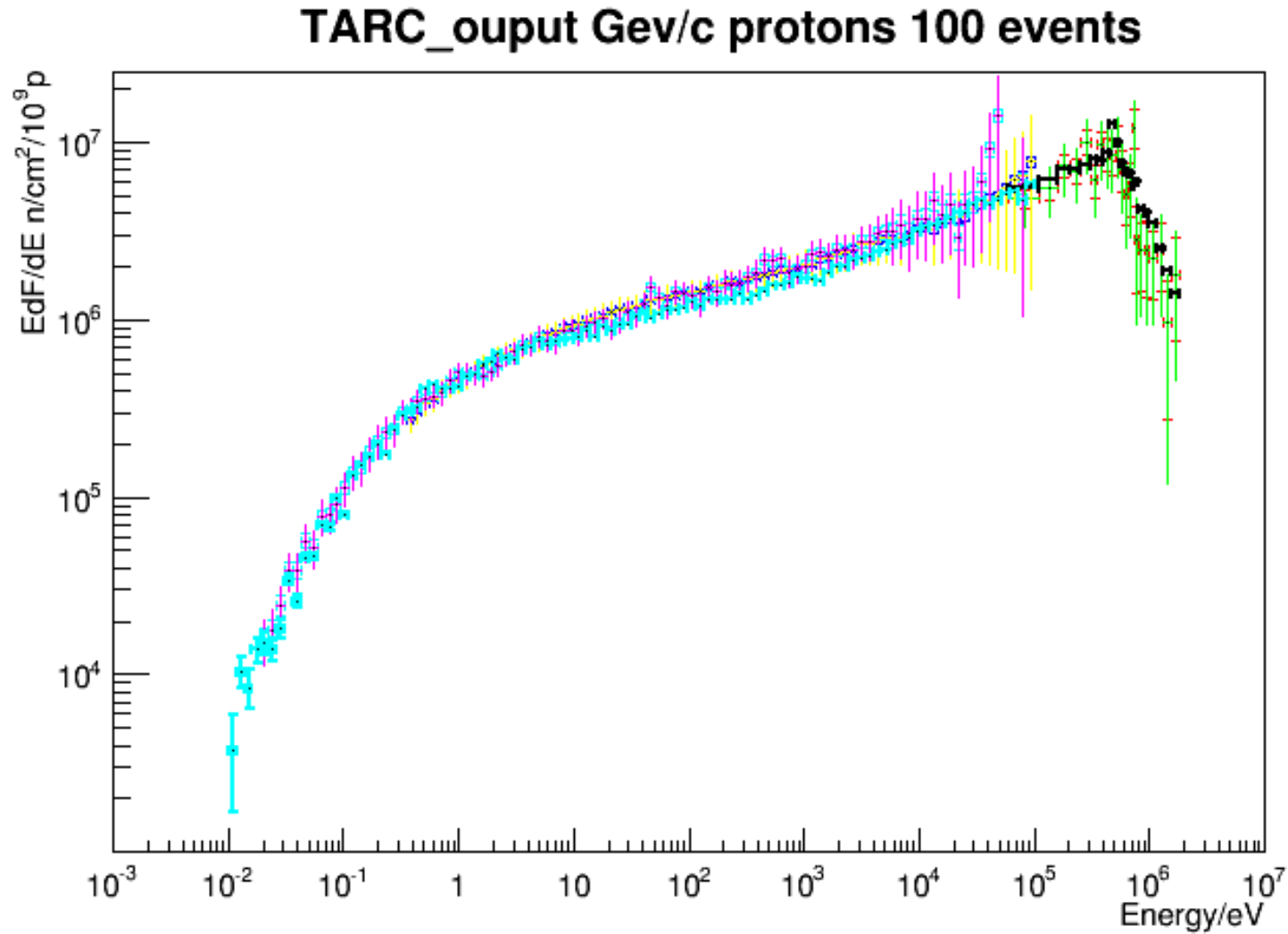
# Fluence Spectrum – 10.2ref08



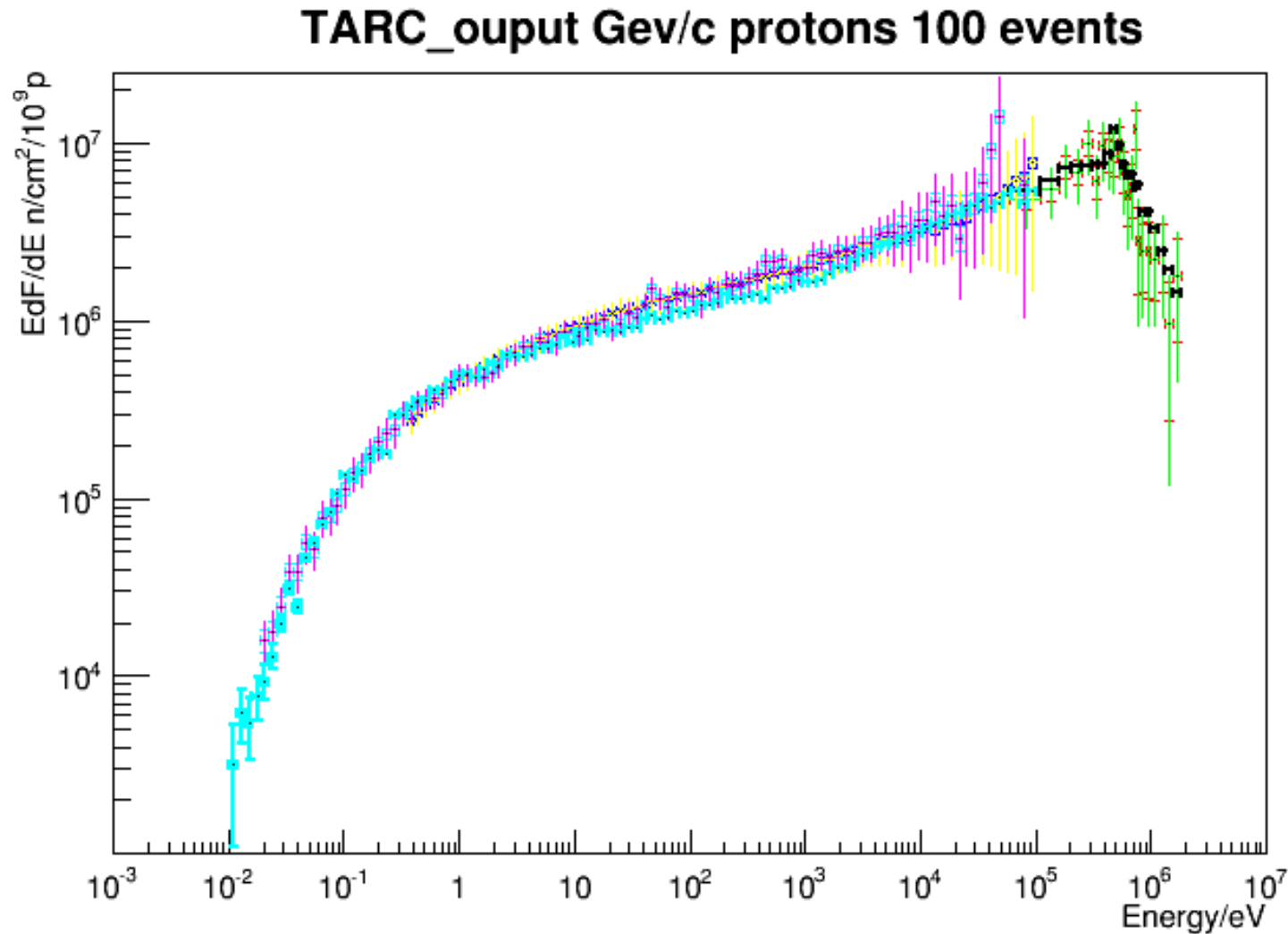
# Fluence Spectrum – 10.2ref08MT



# Fluence Spectrum – 10.2ref08mod



# Fluence Spectrum – 10.2ref08fix

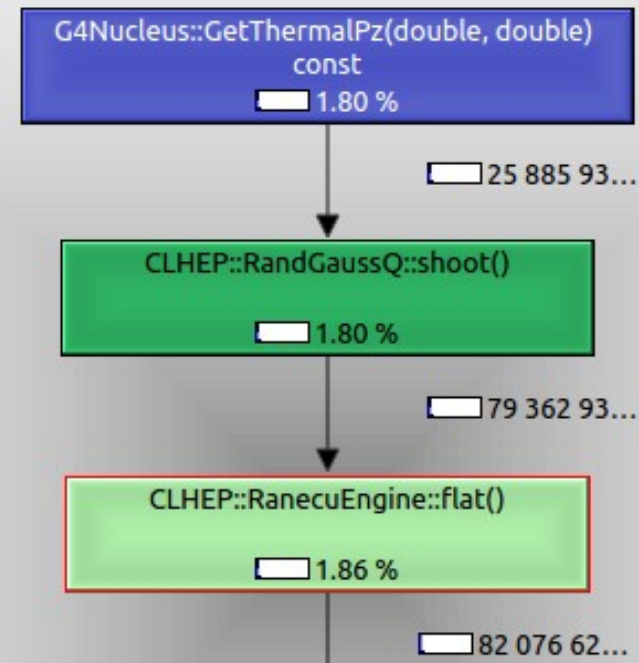




# Profiling Ref08

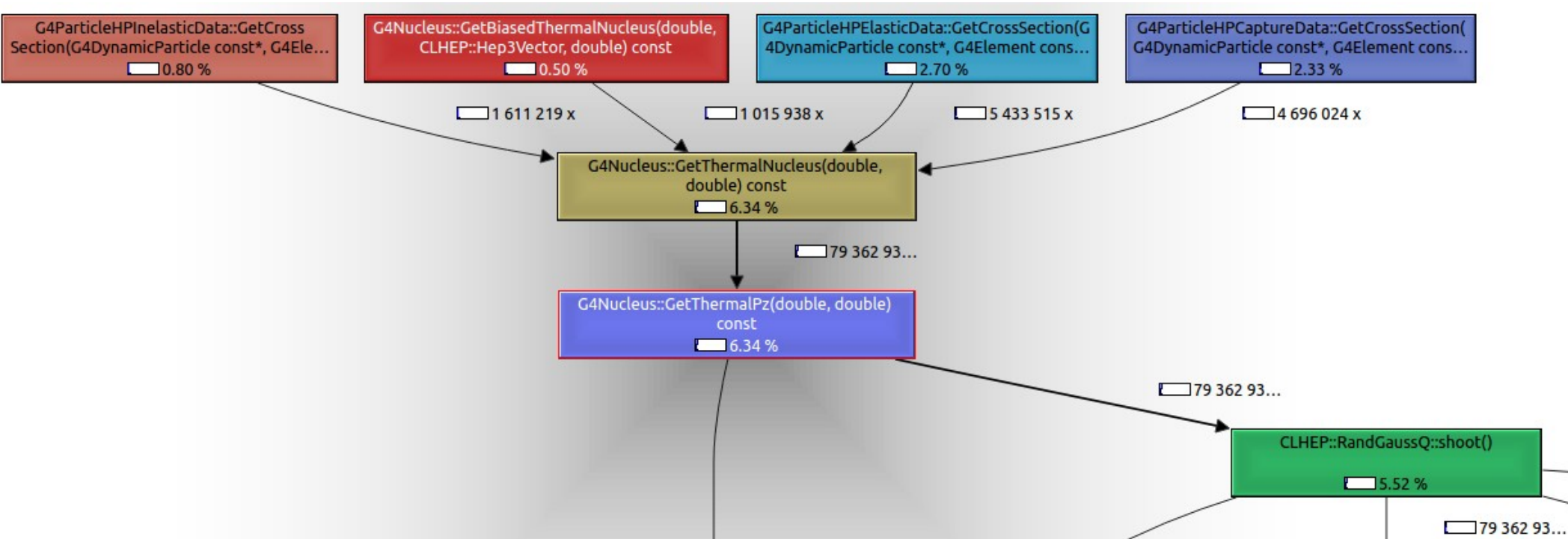
- Valgrind/callgrind → kcachegrind
- 1 event in TARC (!) - 36 hours...

3.15	1.74	259 732 644	G4ParticleHPInterpolator::Li...	libC
1.66	1.66	345 104 213	std::string::push_back(char)	libs
1.86	1.42	82 076 627	CLHEP::RanecuEngine::flat()	libC
4.93	1.18	689 230 921	G4ParticleHPDataPoint::Get...	libC
10.72	1.06	16 659 451	__gnu_cxx::__normal_iterato...	libC
1.25	0.82	79 363 554	CLHEP::RandGaussQ::transf...	libC
6.74	0.80	214 486 400	void std::advance<__gnu_cx...	libC
5.24	0.77	36 863 115	G4ParticleHPVector::Check(i...	libC
13.15	0.72	26 454 310	G4Nucleus::GetThermalNucl...	libC
2.36	0.71	303 670 498	std::abs(double)	TARF
0.68	0.68	42 328 199	str_to_mpn.isra.0	libC
0.67	0.67	3 293	adler32	libC
3.88	0.66	24 943 222	G4ReactionProduct::Lorentz...	libC
3.59	0.59	22 662 947	G4ParticleHPHash::GetMinIn...	libC
9.19	0.57	14 329 617	G4ParticleHPVector::GetXse...	libC
1.74	0.57	214 486 400	__gnu_cxx::__normal_iterato...	libC
2.31	0.55	323 156 960	G4ParticleHPDataPoint::Get...	libC
3.10	0.54	129 512 493	G4ParticleHPDataPoint::ope...	libC
1.61	0.48	206 665 827	std::vector<G4ParticleHPDat...	libC



# Profiling Ref08

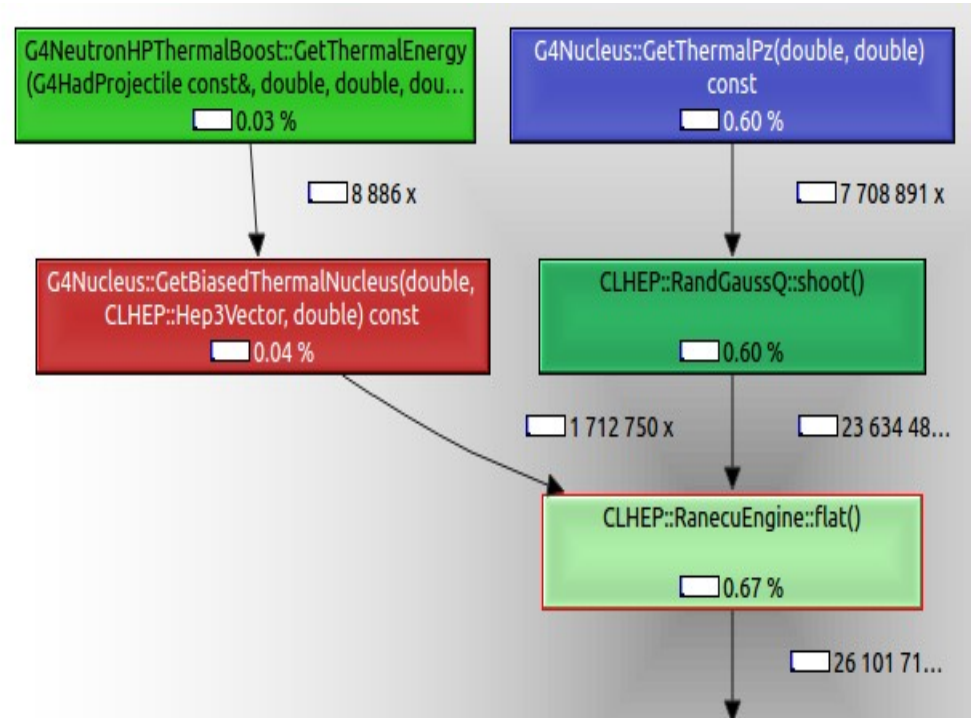
- Valgrind/callgrind → kcachegrind



# Profiling Ref08mod

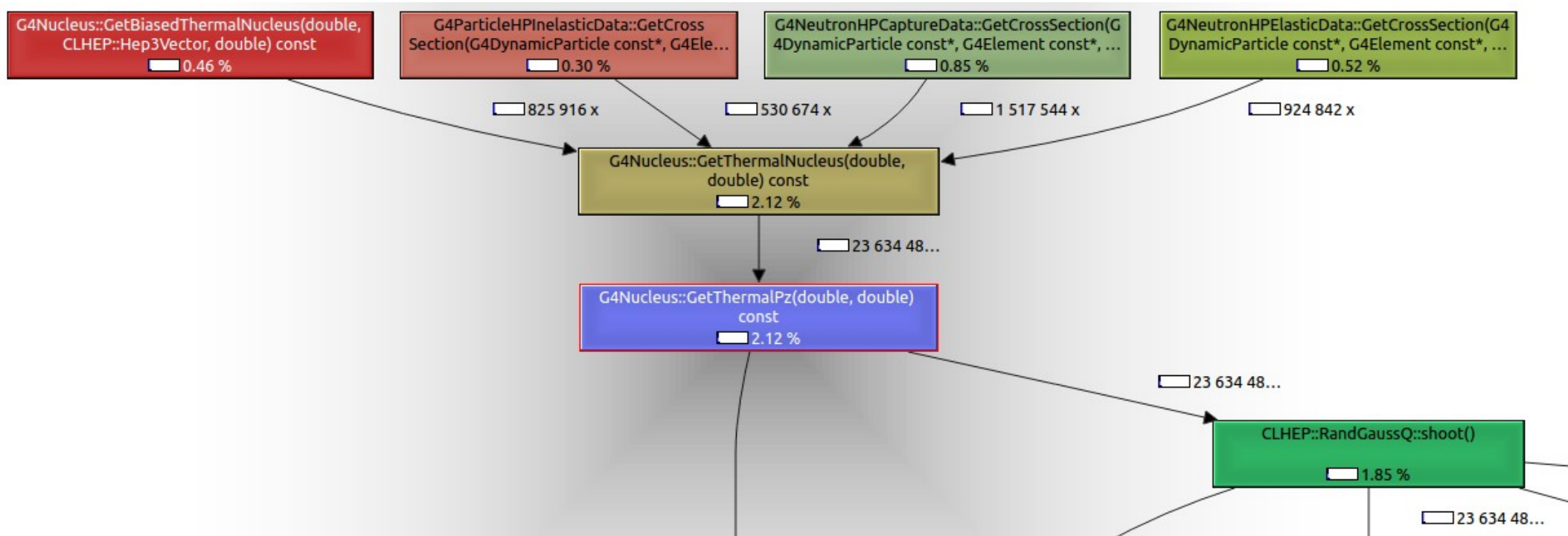
- Replaced particle\_hp with neutron\_hp from v10.1\_ref03

3.55	1.95	259 732 644	G4ParticleHPInterpolator::Li...	libG
6.66	1.55	80	G4NeutronHPVector::ThinO...	libG
3.83	1.15	438 061 292	std::abs(double)	TAR
4.71	1.13	585 254 462	G4ParticleHPDataPoint::Get...	libG
1.77	0.98	129 866 322	G4NeutronHPInterpolator::L...	libG
0.89	0.89	49 698 487	str_to_mpn.isra.0	libc
0.83	0.83	3 631	adler32	libG
3.02	0.72	375 703 892	G4NeutronHPDataPoint::Ge...	libG
4.90	0.72	29 971 137	G4ParticleHPVector::Check(i...	libG
0.60	0.60	495 955 334	0x000000000000ab980	libs
2.45	0.59	304 709 755	G4ParticleHPDataPoint::Get...	libG
3.55	0.58	19 789 629	G4ParticleHPHash::GetMinIn...	libG
2.92	0.51	108 414 119	G4ParticleHPDataPoint::ope...	libG
0.67	0.51	26 101 711	CLHEP::RanecuEngine::flat()	libG
0.50	0.50	40 098 179	_int_malloc	libc
0.49	0.49	33 621 922	std::istream::sentry::sentry(s...	libs
7.78	0.46	10 740 476	G4ParticleHPVector::GetXse...	libG
12.98	0.45	33 148 936	std::num_get<char, std::istre...	libs
0.46	0.45	36 197 600	_int_free	libc



# Profiling Ref08mod

- Replaced particle\_hp with neutron\_hp from v10.1\_ref03



# The Problem?

- MT compliance, not sequential...
- All ParticleHP data classes have unique behaviour for workers vs. master

```
G4ParticleHPCaptureData::G4ParticleHPCaptureData()  
:G4VCrossSectionDataSet("NeutronHPCaptureXS")  
{  
    SetMinKinEnergy( 0*MeV );  
    SetMaxKinEnergy( 20*MeV );  
  
    theCrossSections = 0;  
    onFlightDB = true;  
  
    instanceOfWorker = false;  
    if ( G4Threading::IsWorkerThread() ) {  
        instanceOfWorker = true;  
    }  
    //BuildPhysicsTable(*G4Neutron::Neutron());  
}  
  
G4ParticleHPCaptureData::~G4ParticleHPCaptureData()  
{  
    if ( theCrossSections != NULL && instanceOfWorker != true ) {  
        theCrossSections->clearAndDestroy();  
        delete theCrossSections;  
        theCrossSections = NULL;  
    }  
}
```

# The Problem?

- MT compliance, not sequential...
- Capture, elastic, fission, ?inelastic?

```
G4ParticleHPCaptureData::G4ParticleHPCaptureData()  
:G4VCrossSectionDataSet("NeutronHPCaptureXS")  
{  
    SetMinKinEnergy( 0*MeV );  
    SetMaxKinEnergy( 20*MeV );  
  
    theCrossSections = 0;  
    onFlightDB = true;  
  
    // instanceOfWorkers = false;  
    instanceOfWorkers = true;  
    if ( G4Threading::IsWorkerThread() ) {  
        instanceOfWorkers = true;  
    }  
    //BuildPhysicsTable(*G4Neutron::Neutron());  
}  
  
G4ParticleHPCaptureData::~~G4ParticleHPCaptureData()  
{  
    if ( theCrossSections != NULL && instanceOfWorkers != true ) {  
        theCrossSections->clearAndDestroy();  
        delete theCrossSections;  
        theCrossSections = NULL;  
    }  
}
```

# Exiting Number

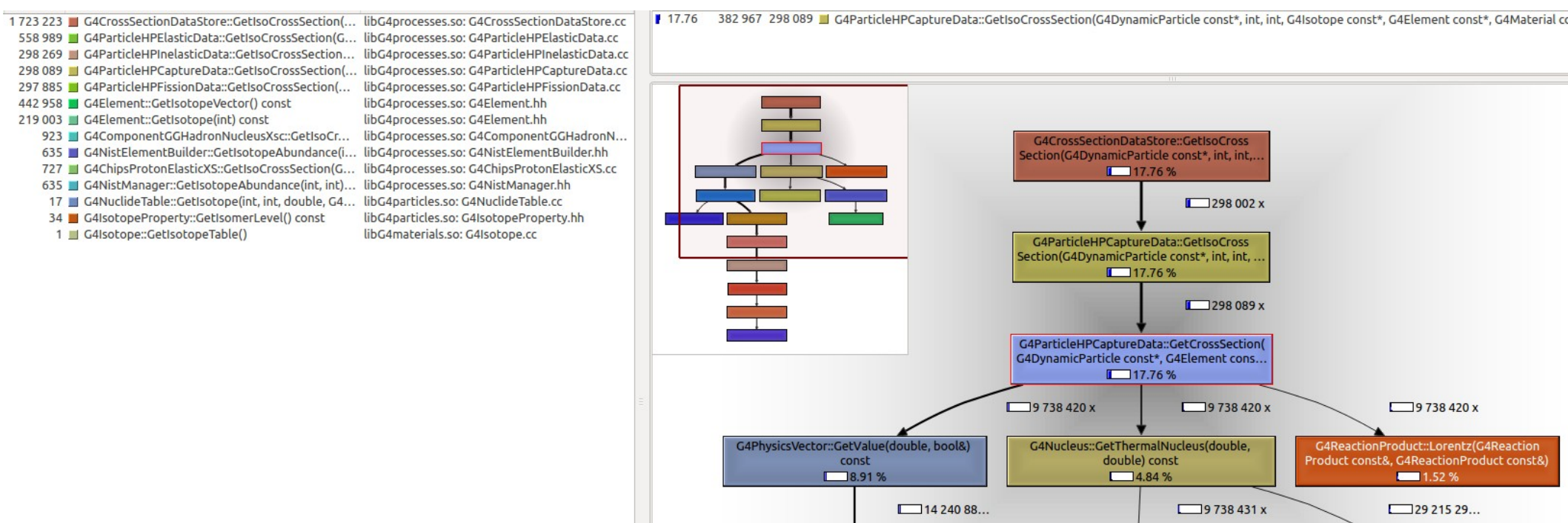
G4	10.1ref03	10.1ref04	10.1ref06	10.1ref09	
Sequential #	19782	20510	20318	20313	
Sequential MeV	1010	990	958	1185	
MT MeV	1114	1218	1075	960	
G4	10.2ref00	10.2ref06	10.2ref08	10.2ref08mod	10.2ref08fix
Sequential	20272	20784	20973	21109	20973
Sequential MeV	1361	1200	1021	<b>1063</b>	1324
MT MeV	920	980	1324	<b>1200</b>	1021

# Debug mode...

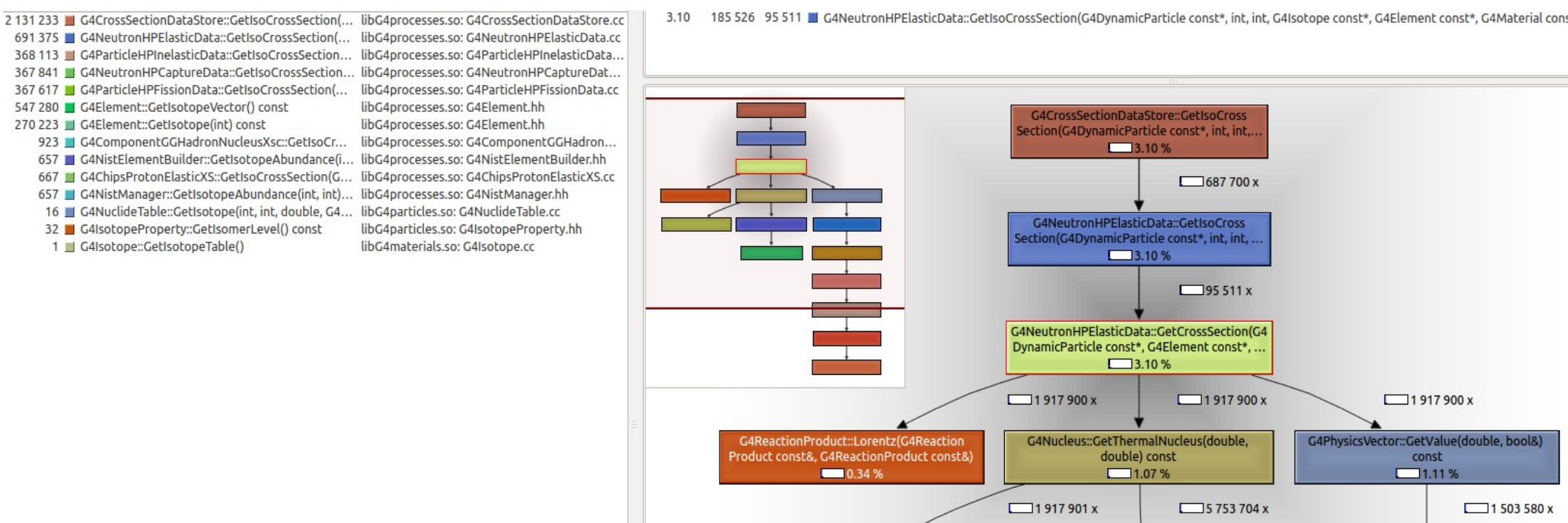
- 1 event in Ref08fixMT = 5m12s
- 1 event in Ref08mod = 4m46s
- 1 event in Ref08 = 5m21s
- 1 event in Ref08fixCache = 4m9s



# GetIsoCrossSection - ref08



# GetIsoCrossSection - ref08mod



# Capture, Elastic, Fission, Inelastic

## particle\_HP

```
G4double G4ParticleHPCaptureData::GetIsoCrossSection( const G4DynamicParticle* dp ,
                                                    G4int /*Z*/ , G4int /*A*/ ,
                                                    const G4Isotope* /*iso*/ ,
                                                    const G4Element* element ,
                                                    const G4Material* material )
{
    G4double xs = GetCrossSection( dp , element , material->GetTemperature() );
    return xs;
}
```

## neutron\_HP

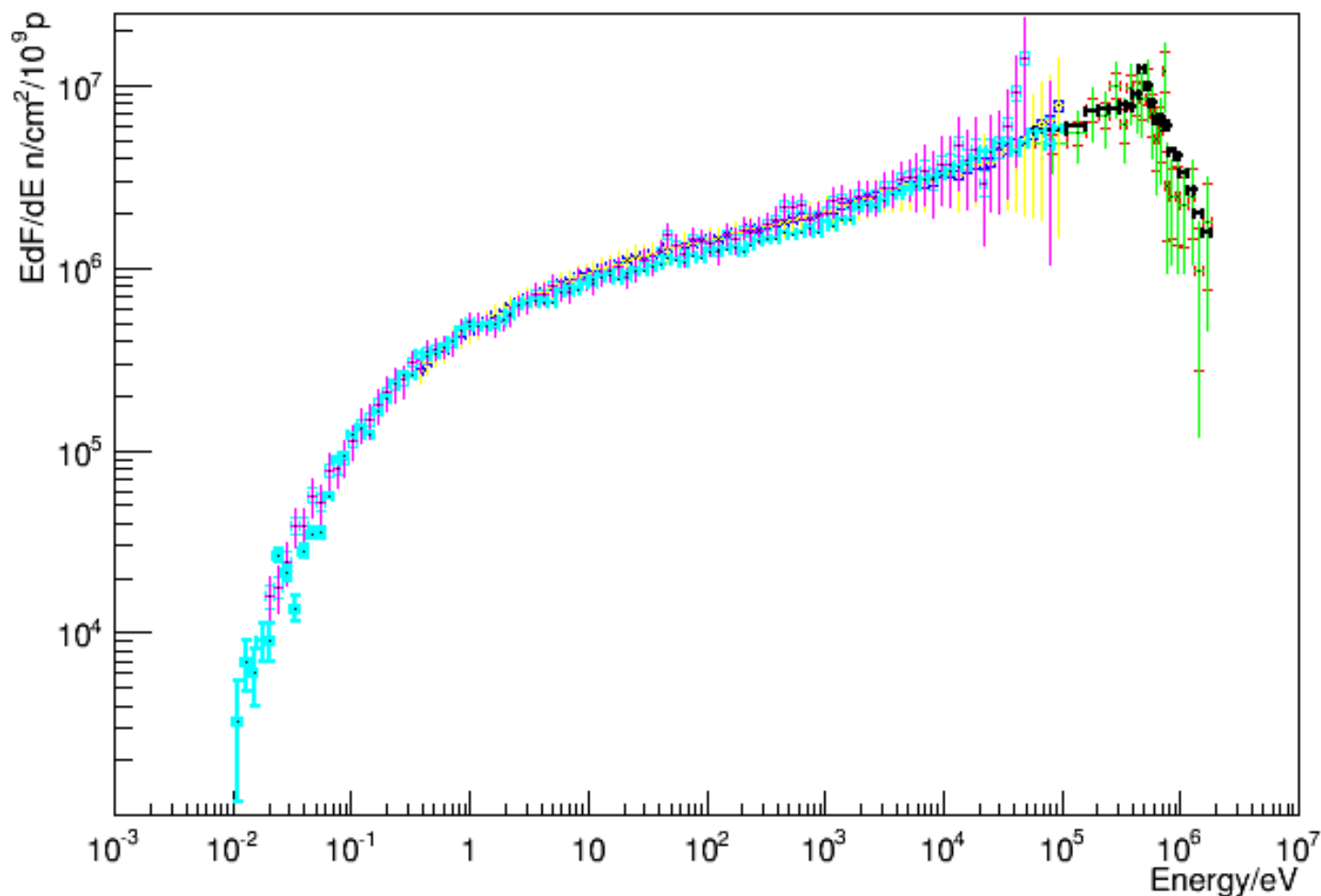
```
G4double G4NeutronHPCaptureData::GetIsoCrossSection( const G4DynamicParticle* dp ,
                                                    G4int /*Z*/ , G4int /*A*/ ,
                                                    const G4Isotope* /*iso*/ ,
                                                    const G4Element* element ,
                                                    const G4Material* material )
{
    if ( dp->GetKineticEnergy() == ke_cache && element == element_cache && material == material_cache ) return xs_cache;

    ke_cache = dp->GetKineticEnergy();
    element_cache = element;
    material_cache = material;
    G4double xs = GetCrossSection( dp , element , material->GetTemperature() );
    xs_cache = xs;
    return xs;
}
```

# Ref08 + cache fix – Sequential

- Time for 1000 events: **68m33s vs. 125m34s**
- Exiting #: 21598 vs. 20973
- Exiting MeV: 1070 vs. 1021

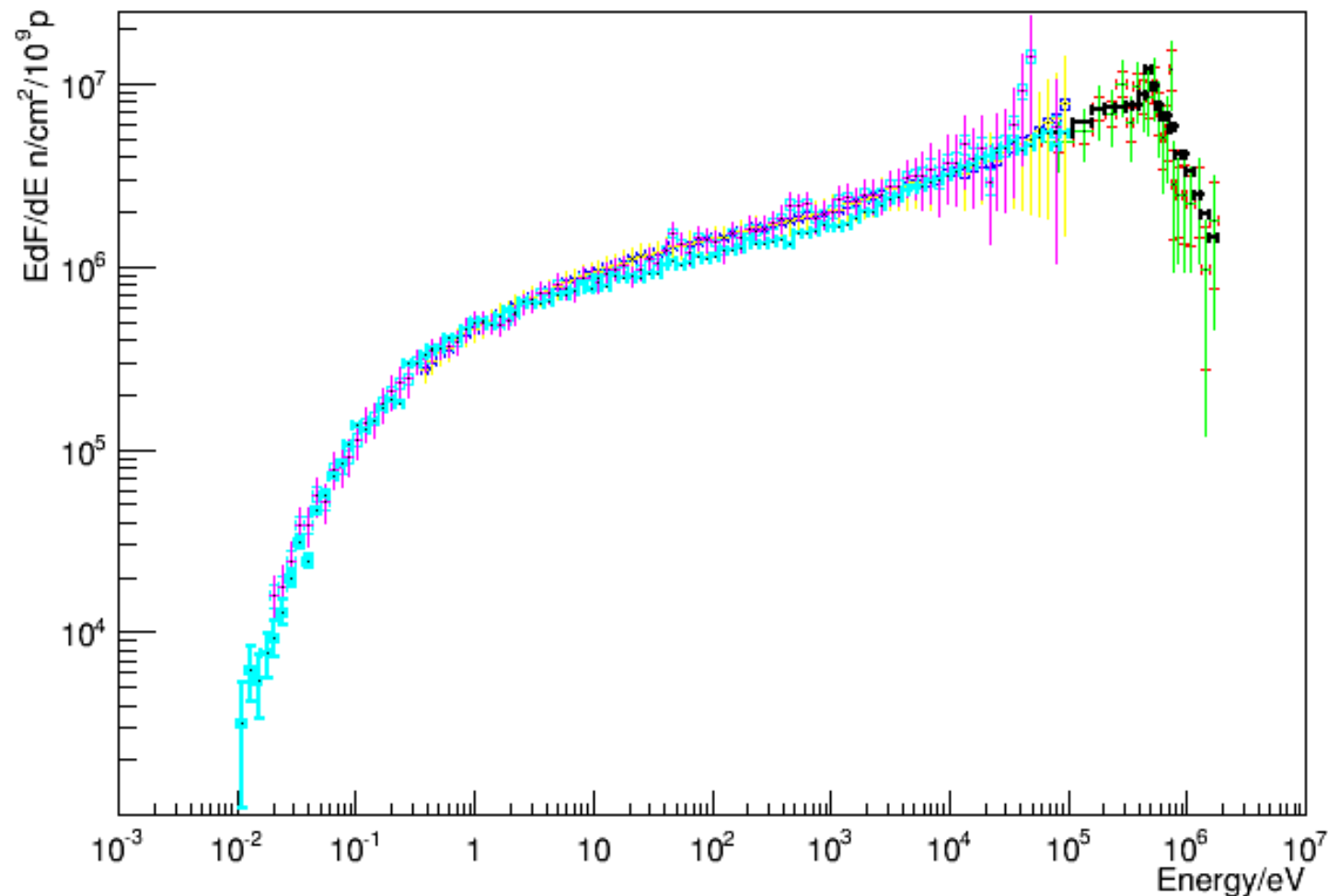
**TARC\_ouput Gev/c protons 100 events**



# cf: Ref08 normal

- Time for 1000 events: **125m34s**
- Exiting #: 20973
- Exiting MeV: 1021

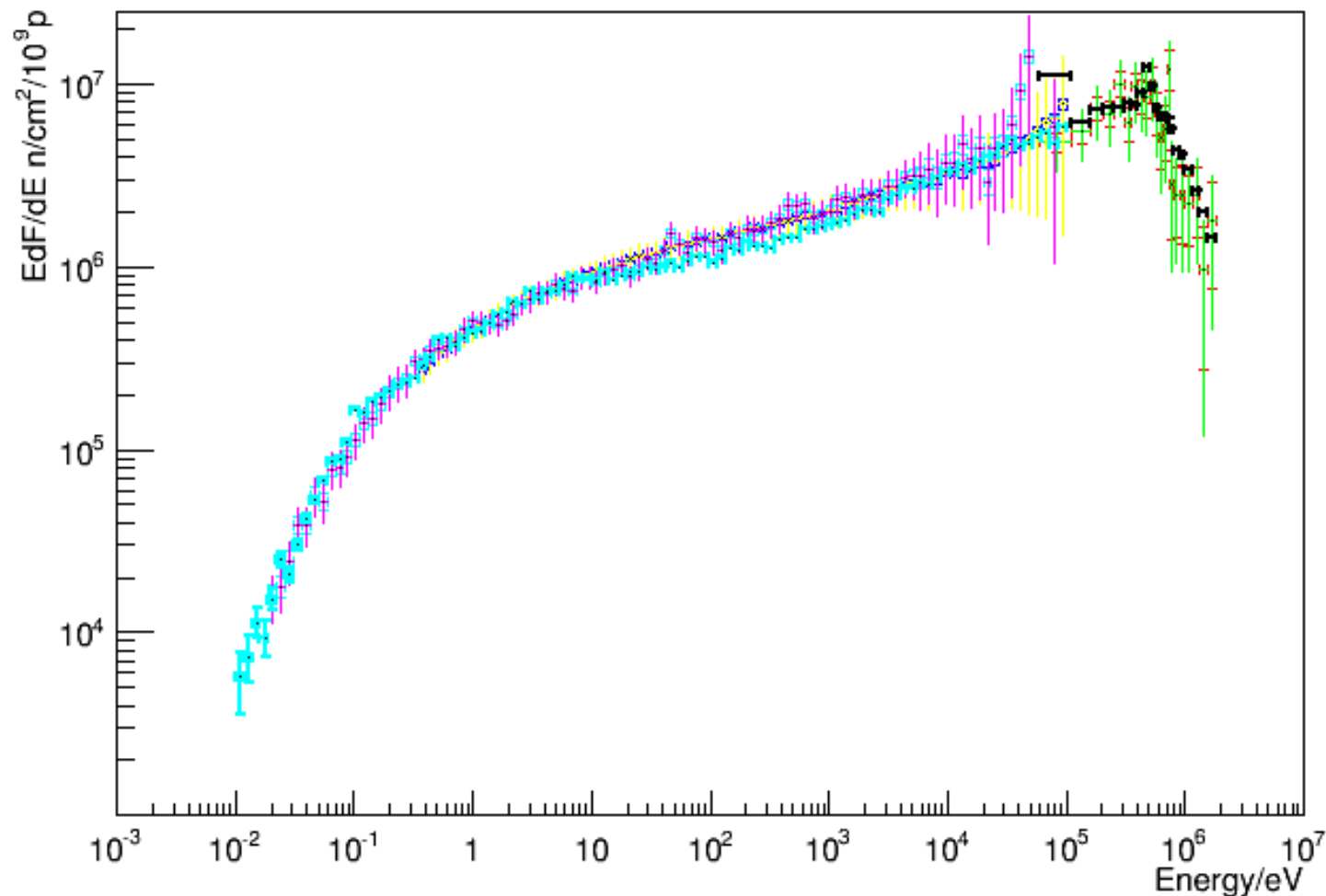
**TARC\_ouput Gev/c protons 100 events**



# Ref08 + cache fix – MT (10 threads)

- Time for 1000 events: **13m24s vs. 22m31s**
- Exiting #: 21280 vs. 21332
- Exiting MeV: 1271 vs. 1324

**TARC\_ouput Gev/c protons 100 events**



# Created Tracks and Particles

Other: NOT neutron, proton, deuteron, triton, gamma, e-, <sup>206</sup>Pb, <sup>207</sup>Pb, <sup>208</sup>Pb

Sequential Mode

G4	10.1ref03	10.1ref04	10.1ref06	10.1ref09	
# Tracks	49300000	50120000	51720000	51980000	
Neutrons	<b>139802</b>	147331	145860	146061	
Protons	<b>6006</b>	5931	5797	5949	
Other	<b>775532</b>	785445	810315	812656	
G4	10.2ref00	10.2ref06	10.2ref08	10.2ref08mod	10.2ref08fix
# Tracks	51820000	51420000	51240000	51380000	52500000
Neutrons	143711	151046	150428	150226	<b>154513</b>
Protons	5869	6010	5799	5860	<b>5951</b>
Other	809264	802192	800102	817607	<b>817291</b>

# Summary

- Cross-section caching seems to be the problem of CPU penalty between 10.1 and 10.2
  - **Was there a reason for leaving it out of particle\_hp?**
- Ref08 is faster than ref06 – expected?
- Spectrum appears to be a bit harder (more exiting flux) – expected?
- **instanceOfWorkers** flag makes no sense to me
- Some MT differences in fluence (high bin) need investigation
  - Could be my code...
- **Maybe it's time to introduce TARC to testing?!!**