Overview

Ideas explored

- **Particle bunching (G4SmartTrackStack)**
- Hard-coded stepping manager (G4SteppingManager)
- Caching of cross-sections calculations in hadronic processes (G4CrossSectionDataStore)
- Reducing branch mispredictions in Value() (G4PhysicsVector)
- Caching values of ln(Energy) (G4Track)

Particle "bunching"

Definition Process *same* particle types before switching to another particle type. E.g.,

$$e^-, e^-, \ldots, e^-, \gamma, \gamma, \ldots, \gamma, \ldots$$

Why Better *cache utilisation* due to access to the same physics list

Number of stacks we are using: 5

1. Primary particles + everything not belonging to:
2. Neutrons
3. Electrons
4. Gammas
5. Positrons

Particle "bunching" - Problems

Problems

- **Stacks can grow very large**
  - e.g., when processing electrons, the gamma stack explodes, and vice versa
- Therefore, we have to **restrict** them, which leads to another problem:
  - What is the optimal size for each one?
  - How much aggressively should we process a track, once it has hit its upper limit ?

Particle "bunching" - Problems cont.

If we allow **too large** stack sizes

- we diverge a lot in terms of geometry (it hurts)

If we allow **too small** stack sizes

- we switch too often between stacks, and we thrash (it hurts)

If we are **too aggressive** when penalizing the offending stack,

- by consuming its elements, then the other stacks will get inflated (it hurts)

Outcome very dependent on the selection of above parameters

G4SmartTrackStack - Current state of things

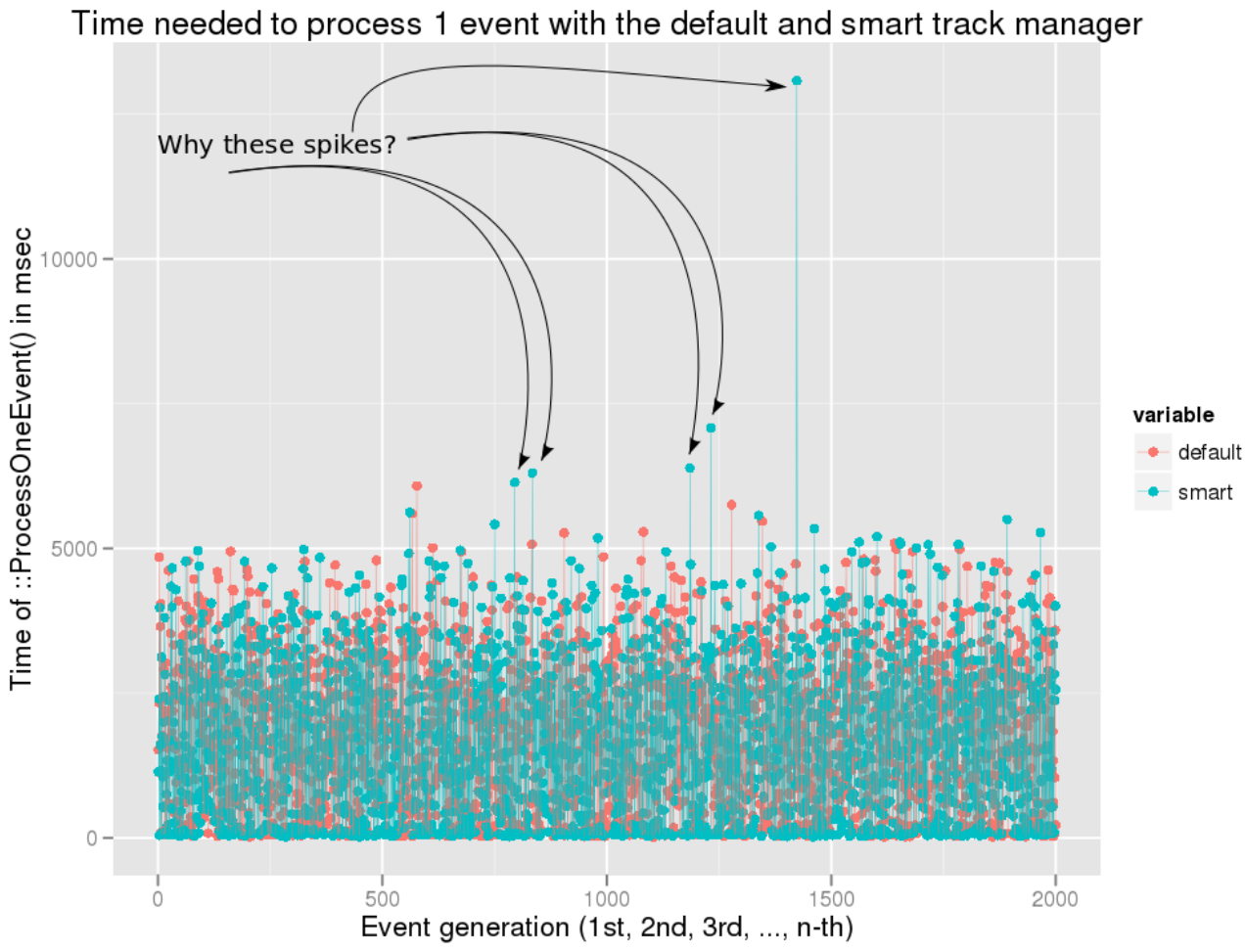
Current state The algorithm, in its current incarnation, does **not** provide any benefit in terms of performance.

Problems

- Suboptimal choice of max stack sizes
- Although G4SmartTrackStack tries to impose limits on the maximum size a stack can grow to, there is a **degenerate** case where it doesn't.

USDT probes + Speculative tracing - A real use case

Problem Some `ProcessOneEvent()` need much more than average time to complete



USDT probes + Speculative tracing - A real use case

Strategy We are going to trace all `ProcessOneEvent()` calls, but commit to our tracing buffer *only* those that behave bad.

"Trace", in this context, means to look at the stack sizes.

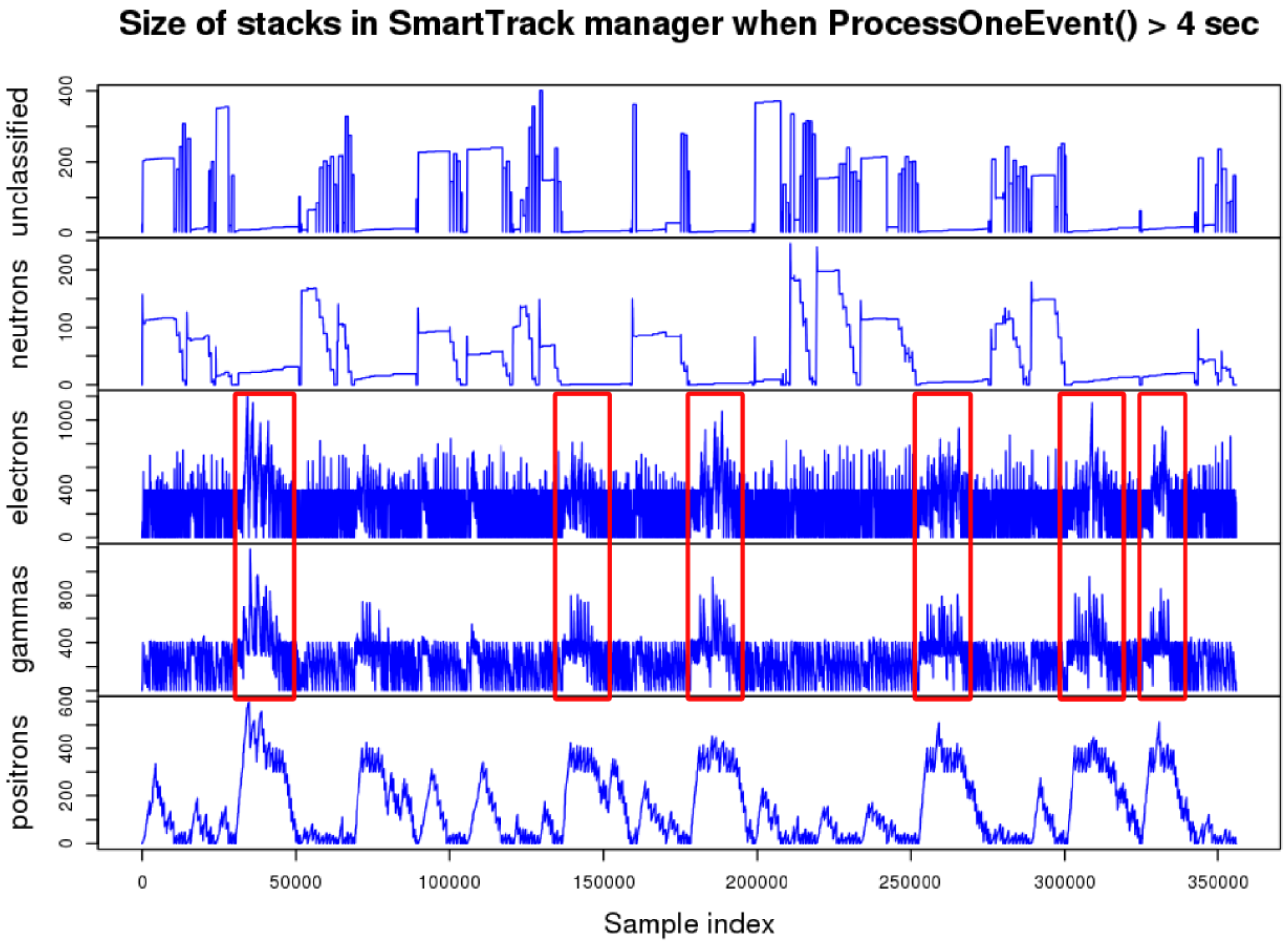
USDT probes + Speculative tracing - A real use case

```
pid$target  *G4EventManager*ProcessOneEventEP7G4Event entry
{
    self->pstart = vtimestamp
    spec = speculation()
}

simple$target
/tracing && spec/
{
    speculate(spec)
    printf("%d %d %d %d %d\n", arg0, arg1, arg2, arg3, arg4)
}

pid$target  - '$retaddr'
/self->pstart/
{
    self->t = (vtimestamp - self->pstart)/1000000
    self->pstart = 0
}

pid$target  - '$retaddr'
/spec && self->t >= 4500/
{
    commit(spec)
    spec = 0
}

pid$target  - '$retaddr'
/spec && self->t < 4500/
{
    discard(spec)
    spec = 0
}
```

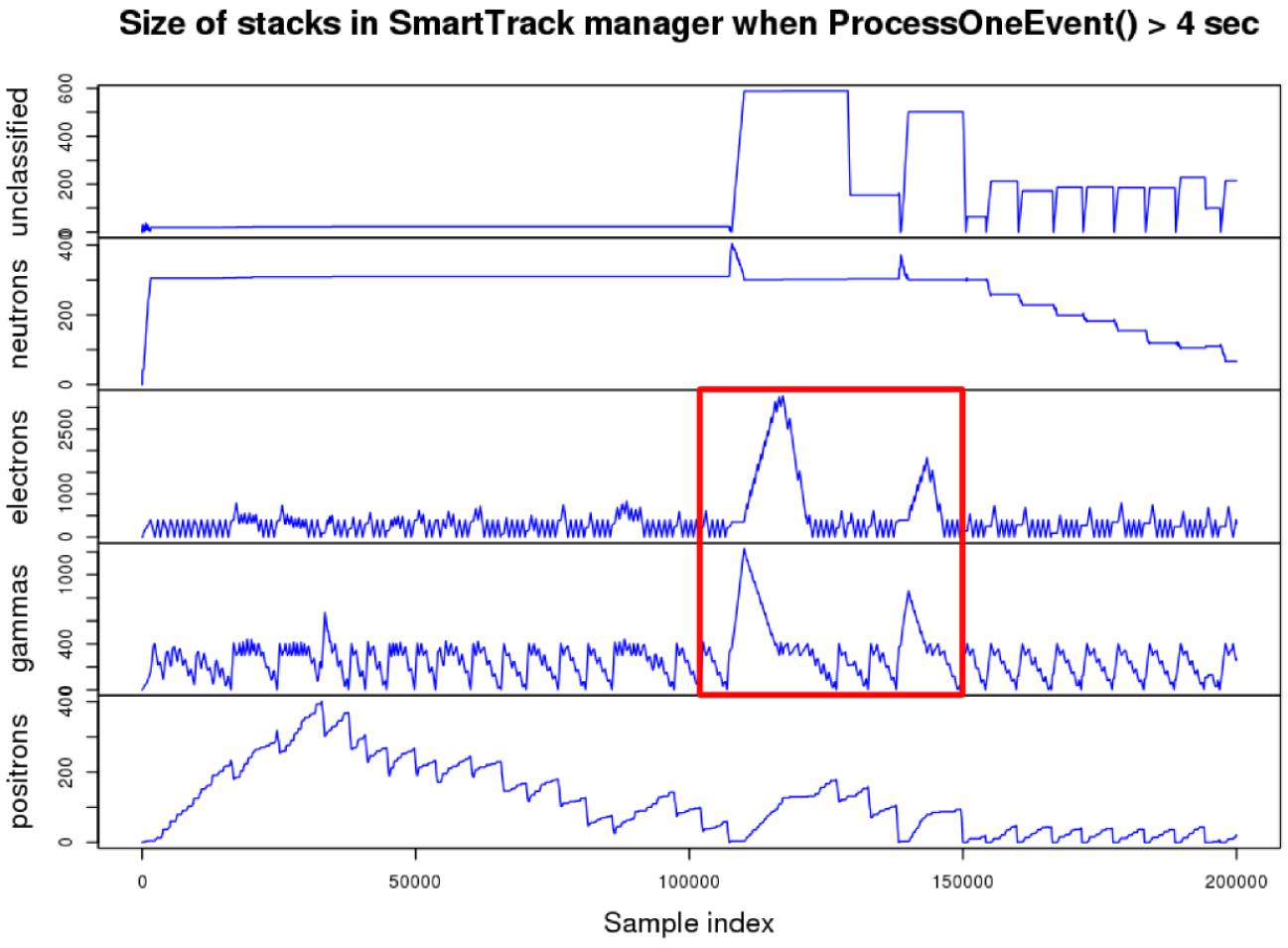
USDT probes + Speculative tracing - A real use case

Hint The maximum desired size for all stacks was requested to be 400.

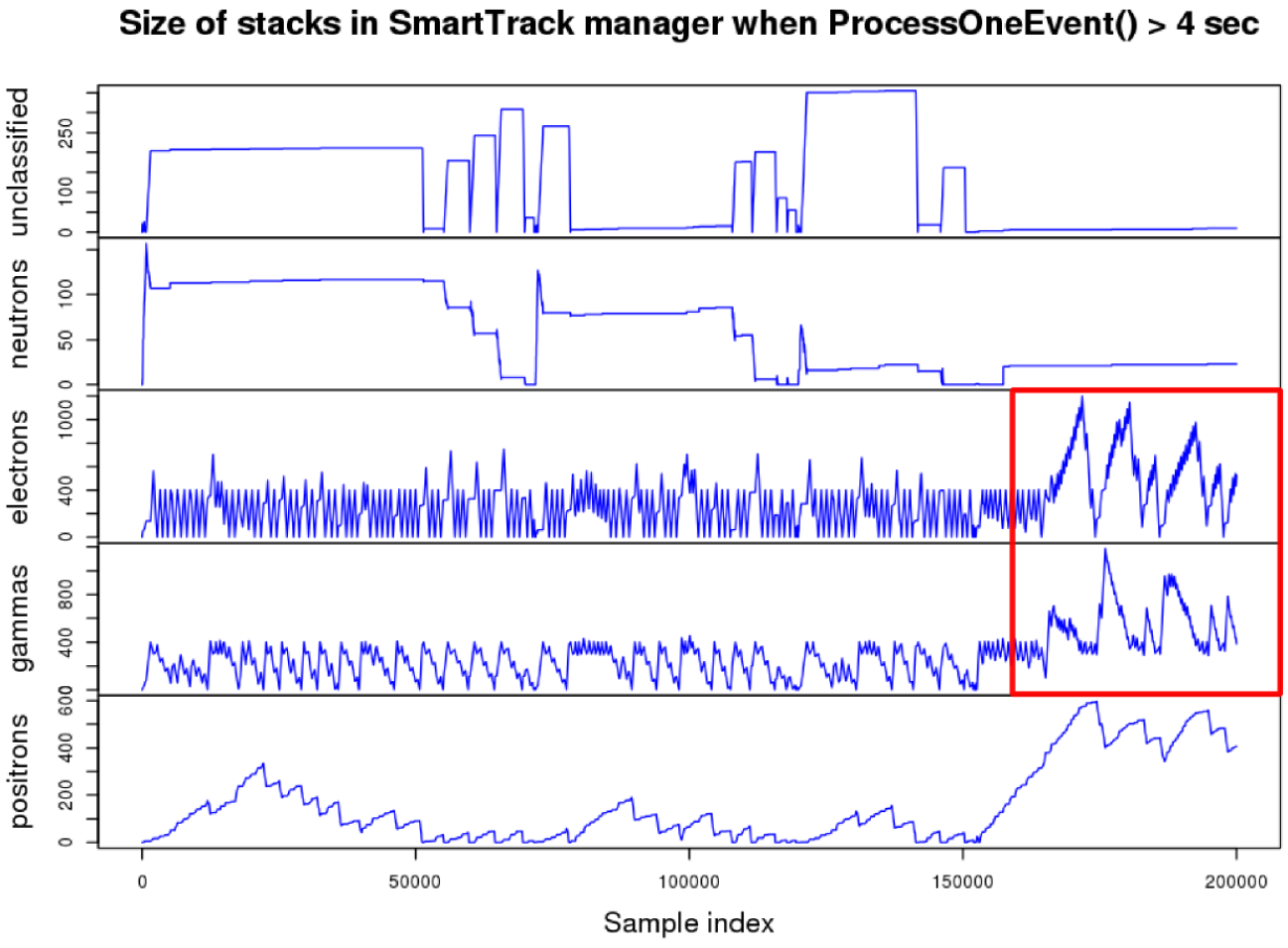
e^- and γ too often will not honour that limit.



USDT probes + Speculative tracing - Zoom 1/2



USDT probes + Speculative tracing - Zoom 2/2



G4SmartTrackStack - Modifications

Reduce size of stacks

- 5% gain in 200 events, 3% in 1k events, 1% in 2k events, cross-over at 3k events

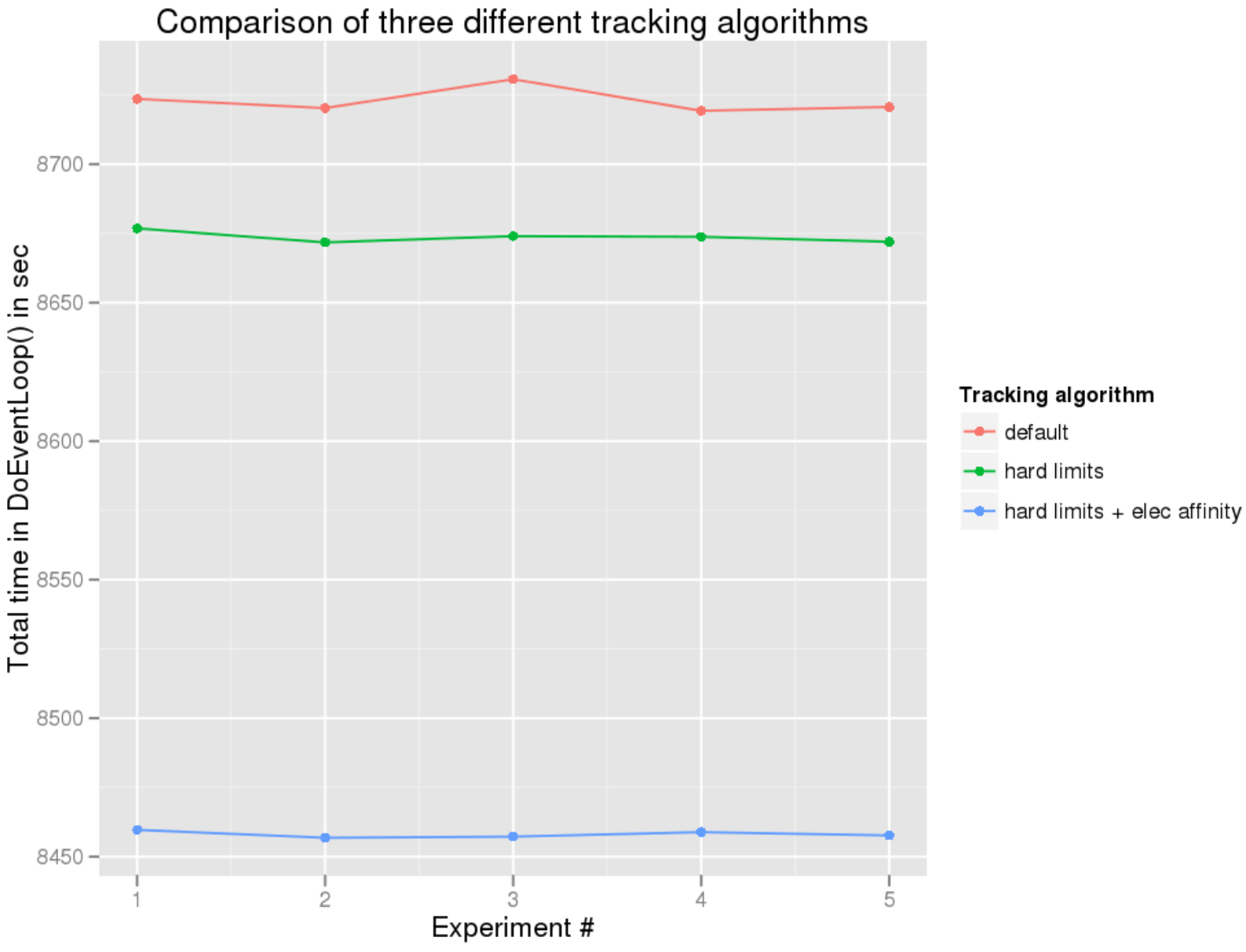
Impose **hard limits** on the size of the stacks

- First version ever to confer a **persistent** reduction in total execution time, albeit small ($< 1\%$)

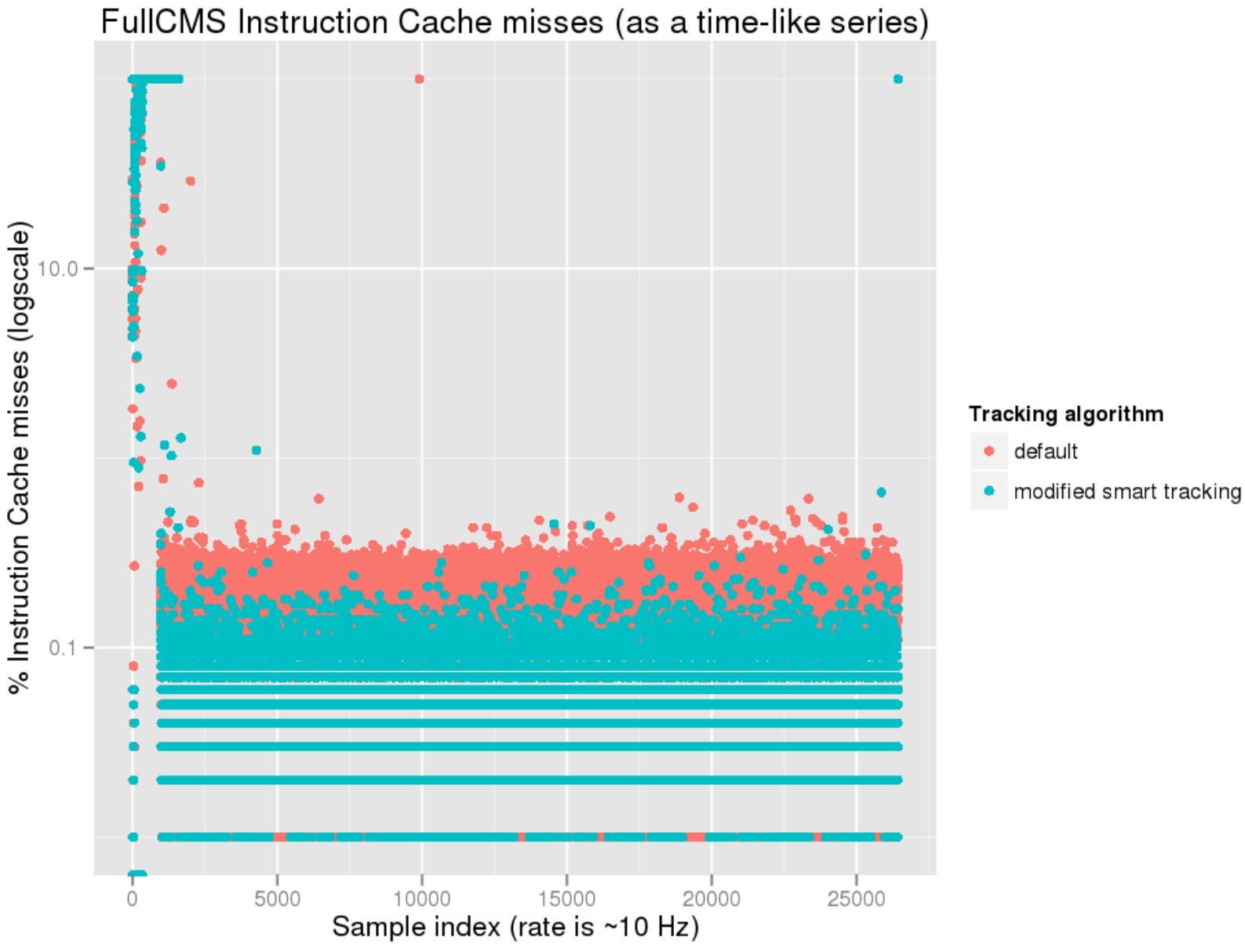
Show **affinity for low energy** e^- (+ hard limits)

- Best version of particle bunching: 4-5% **persistent** reduction in total execution time in FullCMS experiment (less in SimplifiedCalorimeter)

G4SmartTrackStack



G4SmartTrackStack - Why is it faster anyway ?



Future plans

- Document everything, either a positive or a negative result
- Break commits so that each one introduces only one feature
- Try to reproduce the results in an environment closer to CERN's

Documentation

Documentation will be appearing in the following links:
http://leaf.dragonflybsd.org/ beket/geant4/dtrace.html
http://leaf.dragonflybsd.org/ beket/geant4/solaris.html

http://island.quantumachine.net/ stathis/geant4/smartstack.html
http://island.quantumachine.net/ stathis/geant4/crosssections.html
http://island.quantumachine.net/ stathis/geant4/hardstepping.html
http://island.quantumachine.net/ stathis/geant4/lnenergy.html

The end

Thank you. Questions?