ACTS-ITk Seeding - Time Optimisation

Luis Falda Coelho

Jan 31th, 2023

CPU Comparison in Athena

CPU Comparison in Athena

Time monitor in Athena (**SiSPSeededTrackFinder::newStrategy**):

newEvent():

- Fill the SP grid looping on PPP and SSS space point collections
- Iteration 0 for SSS and 1 for PPP, iteration value used to setup parameters, e.g. Rmax of middle SP

find3Sp():

- Produces SP groups with neighbors
- Seed finding and filtering: For every bottom SP, loop on top sps and find compatible top SPs. All seeds for a given central+bottom pair are directly filtered. We don't build seeds if this is not needed (e.g. if the number of top SPs for a given central+bottom pair is below the expected minimum)

Time monitor for ACTS:

Monitoring inside **SeedingTool::createSeeds():**

- Grid initialisation, separated for PPP and SSS SPs since this happens in two separate tools looping on different collections
- Seed production

EDM time in **ActsTrk::SiSpacePointsSeedMaker()**

CPU Comparison in Athena

Time monitor in Athena **newEvent()** and **find3Sp()** separately:

```
/**
 * Set up the first pass (strip seeds), and prepare to
 * obtain a vertex Z estimate from the candidates we find
 **/

/// set up the seed maker for first pass
m_seedsmaker->newEvent(ctx, seedEventData, 0);
std::list<Trk::Vertex> vertexList;
/// and run seeding - starting with an empty list of vertices for the first pass
m_seedsmaker->find3Sp(ctx, seedEventData, vertexList);
```

→ SSS

```
/** Now set up the second seeding pass, using pixel seeds
* The seed maker will internally reconfigure itself based on
* the "1" argument for the "iteration" argument in this call.
**/
m_seedsmaker->newEvent(ctx, seedEventData, 1);

/// perform vertex Z estimation and run second seeding pass
std::pair<double,double> zBoundaries;
if (not m_ITKGeometry) {
  /// Estimate a Z vertex interval and, if running the new strategy, also a list of the HS candidates
  findZvertex(vertexList, zBoundaries, numberHistogram, zWeightedHistogram, ptWeightedHistogram);
  /// pass the Z boundary pair c-array-style to satisfy existing interfaces of the seeds maker family.
  /// Trigger second seed finding pass (PPP)
  m_seedsmaker->find3Sp(ctx, seedEventData, vertexList, &(zBoundaries.first));
} else {
  m_seedsmaker->find3Sp(ctx, seedEventData, vertexList);
}
```

→ PPP

CPU Comparison in Athena

Time monitor in ACTS **createSeeds()**:

Grid Initialisation

```
template< typename spacepoint_iterator_t >
StatusCode
SeedingTool::createSeeds(spacepoint_iterator_t spBegin,
  spacepoint_iterator_t spEnd,
  const Acts::Vector3& beamSpotPos,
  const Acts::Vector3& bField,
  std::vector<Acts::Seed< typename external_spacepoint<spacepoint_iterator_t>::type >>& seeds) const {

using external_spacepoint_t = typename external_spacepoint<spacepoint_iterator_t>::type;
using seed_t = Acts::Seed< external_spacepoint_t >;
seeds.clear();
if (spBegin == spEnd)
  return StatusCode::SUCCESS;

auto [gridCfg, finderCfg] = prepareConfiguration< external_spacepoint_t >(Acts::Vector2(beamSpotPos[Amg::x], beamSpotPos[Amg::y]),
                                                                          bField);

auto extractCovariance = [&beamSpotPos](const external_spacepoint_t& sp, float,
                                        float, float) -> std::pair<Acts::Vector3, Acts::Vector2> {
  /// Convert coordinates w.r.t. beam spot
  Acts::Vector3 position(sp.x() - beamSpotPos[Amg::x], sp.y() - beamSpotPos[Amg::y], sp.z() - beamSpotPos[Amg::z]);
  Acts::Vector2 covariance(sp.varianceR(), sp.varianceZ());
  return std::make_pair(position, covariance);
};


Acts::Extent rRangeSPExtent;

std::shared_ptr< Acts::BinFinder< external_spacepoint_t > > bottomBinFinder =
  std::make_shared< Acts::BinFinder< external_spacepoint_t > >(m_zBinNeighborsBottom, m_numPhiNeighbors);
std::shared_ptr< Acts::BinFinder< external_spacepoint_t > > topBinFinder =
  std::make_shared< Acts::BinFinder< external_spacepoint_t > >(m_zBinNeighborsTop, m_numPhiNeighbors);

std::unique_ptr< Acts::SpacePointGrid< external_spacepoint_t > > grid =
  Acts::SpacePointGridCreator::createGrid< external_spacepoint_t >(gridCfg);
Acts::BinnedSPGroup< external_spacepoint_t > spacePointsGrouping(spBegin, spEnd, extractCovariance,
  bottomBinFinder, topBinFinder, std::move(grid), rRangeSPExtent, finderCfg);

Acts::SeedFinder< external_spacepoint_t > finder(finderCfg);

// variable middle SP radial region of interest
const Acts::Range1D<float> rMiddleSPRange(std::floor(rRangeSPExtent.min(Acts::binR) / 2) * 2 +
                                          finderCfg.deltaRMiddleMinSPRange,
                                          std::floor(rRangeSPExtent.max(Acts::binR) / 2) * 2 -
                                          finderCfg.deltaRMiddleMaxSPRange);
```

CPU Comparison in Athena

Time monitor in ACTS **createSeeds()**:

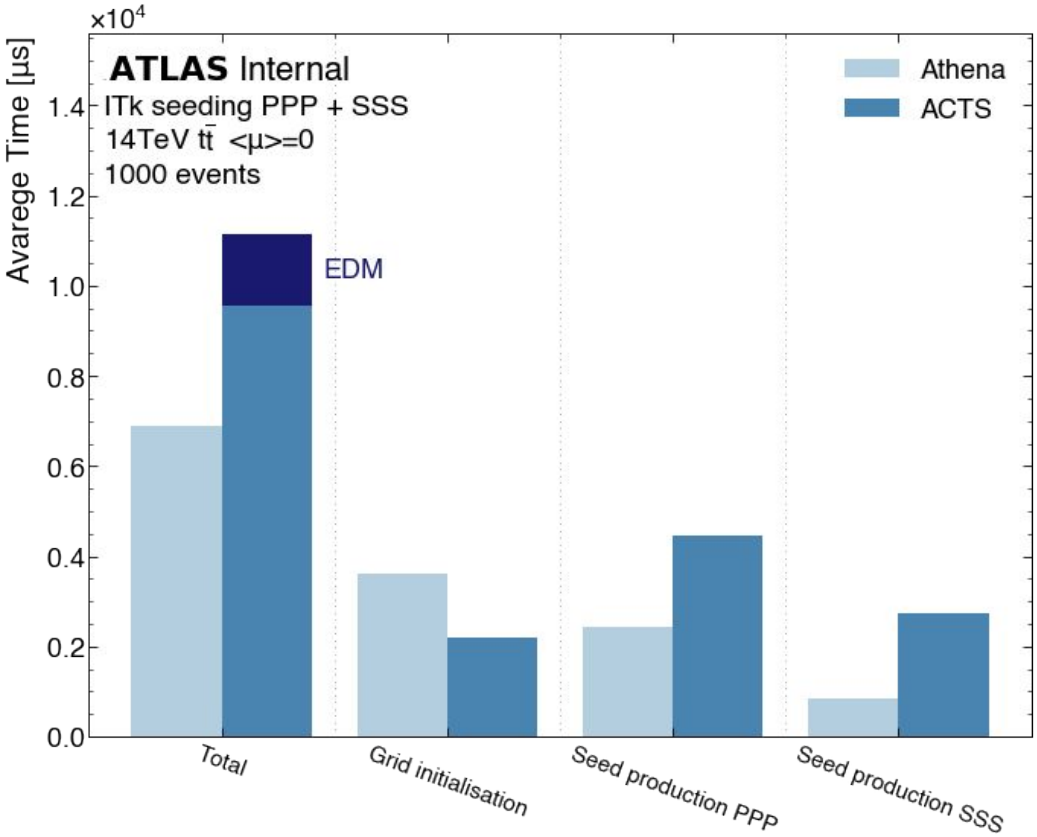
```
//TODO POSSIBLE OPTIMISATION come back here: see MR !52399 ( i.e. use static thread_local)
typename decltype(finder)::SeedingState state;

auto group = spacePointsGrouping.begin();
auto groupEnd = spacePointsGrouping.end();
for (; group != groupEnd; ++group) {
  finder.createSeedsForGroup(state, std::back_inserter(seeds), group.bottom(),
                             group.middle(), group.top(), rMiddleSPRange);
}
```

→ Seed production

EDM time in ACTS **ActsTrk::SiSpacePointsSeedMaker()**:

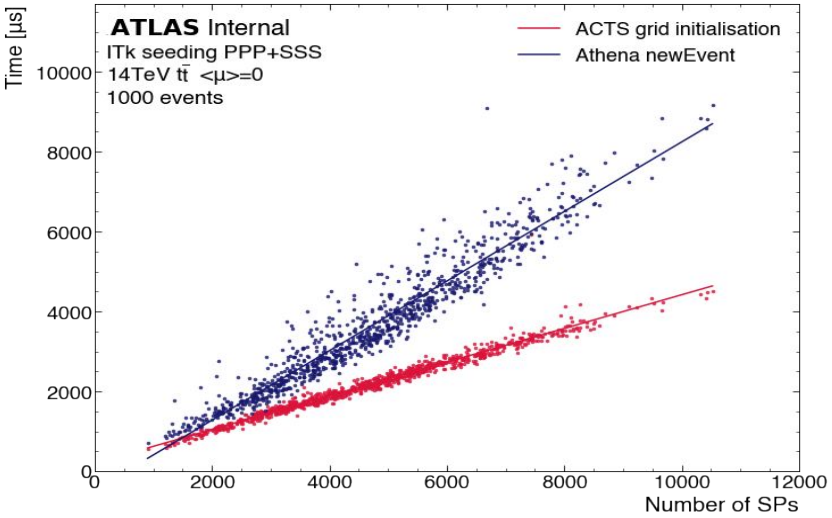
```
if (isPixel) {
  SG::ReadHandle<xAOD::PixelClusterContainer> inputClusterContainer( m_pixelClusterContainerKey, ctx );
  if (!inputClusterContainer.isValid()){
    ATH_MSG_FATAL("xAOD::PixelClusterContainer with key " << m_pixelClusterContainerKey.key() << " is not available...");
    return;
  }
  const xAOD::PixelClusterContainer inputContainer = *inputClusterContainer.cptr();

  std::vector< InDet::PixelSpacePoint* > pixelSpacePoints(inputContainer.size(), nullptr);
  std::vector< ITk::SiSpacePointForSeed* > pixelSpacePointsForSeed(inputContainer.size(), nullptr);


  for (const ActsTrk::Seed* seed : *seedPtrs.get()) {

    // creating ITk::SiSpacePointForSeed for bottom, middle and top sps
    // first we need the space points
    std::size_t bottom_idx = seed->sp()[0]->measurementIndexes()[0];
    std::size_t medium_idx = seed->sp()[1]->measurementIndexes()[0];
    std::size_t top_idx    = seed->sp()[2]->measurementIndexes()[0];

    std::array<const xAOD::PixelCluster*, 3> pixel_cluster{ inputContainer.at(bottom_idx),
                                                            inputContainer.at(medium_idx),
                                                            inputContainer.at(top_idx) };


      ...
```

PPP+SSS ttbar mu=0

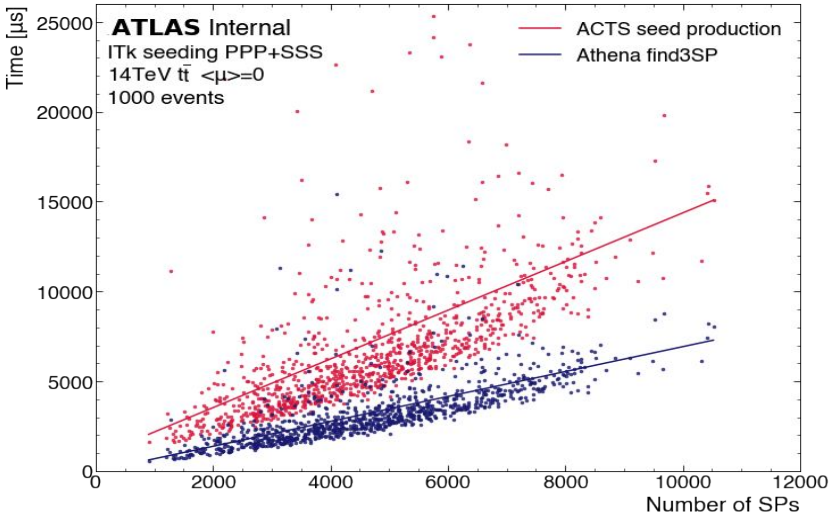


PPP+SSS ttbar mu=0

Grid Time PPP+SSS

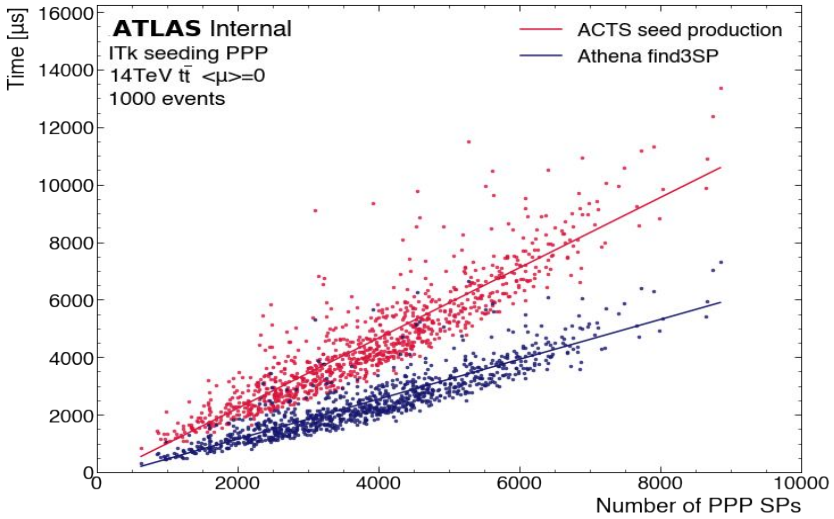


Seed Production Time PPP+SSS

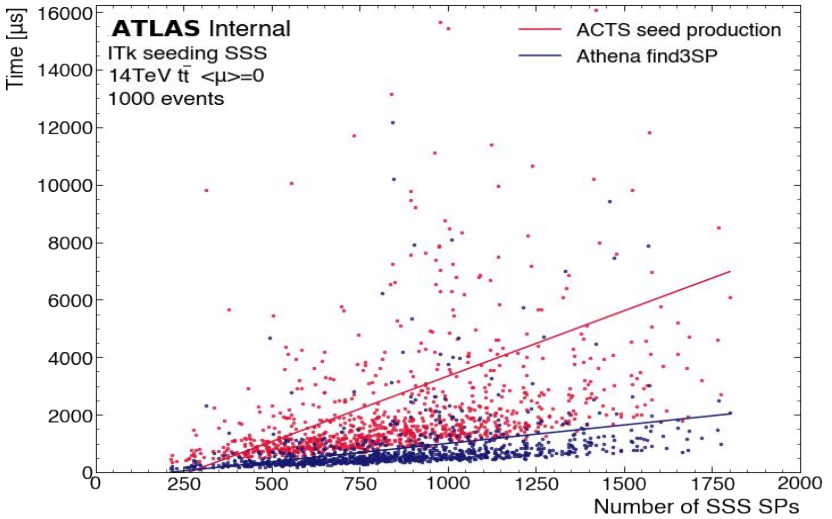


PPP+SSS ttbar mu=0

Seed Production Time PPP

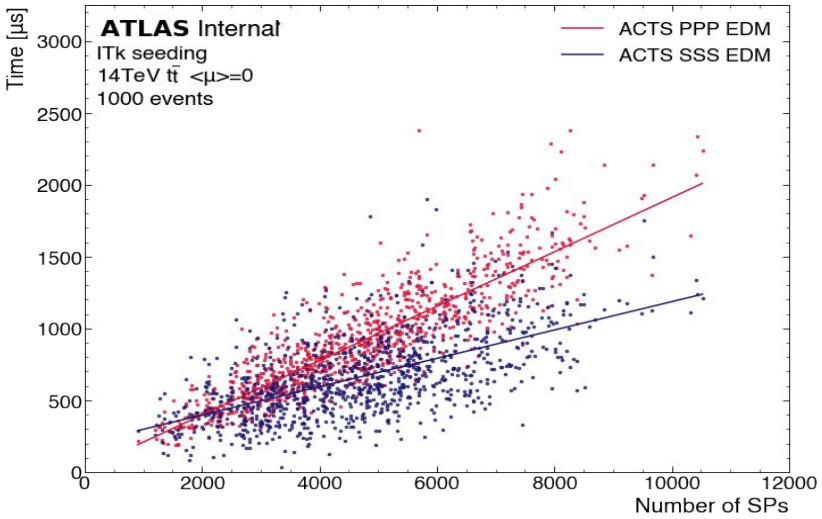


Seed Production Time SSS

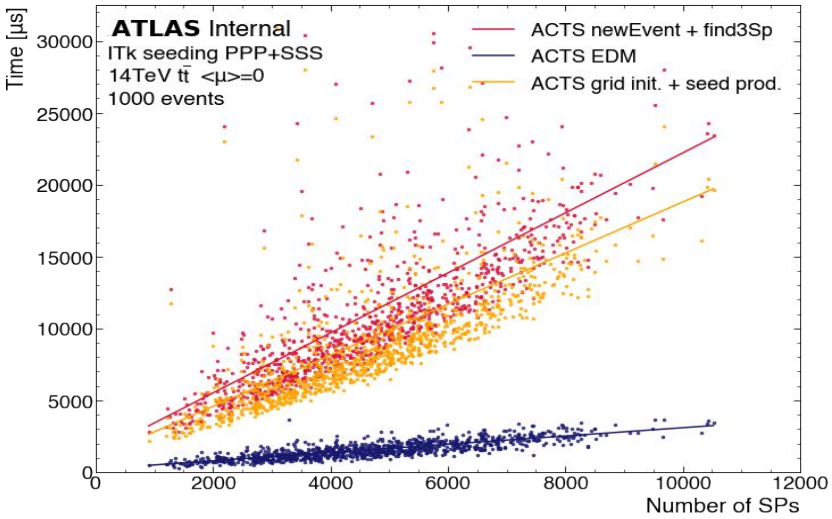


PPP+SSS ttbar mu=0 - EDM Time

EDM Time PPP vs SSS

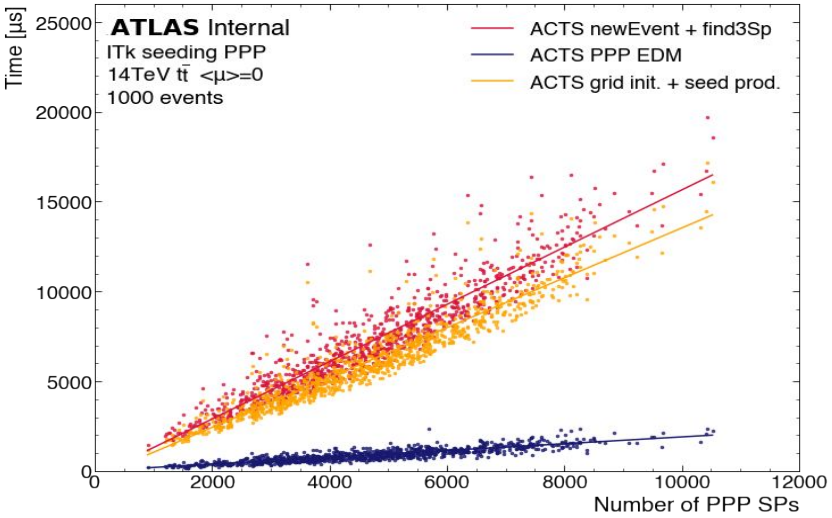


EDM Time PPP+SSS

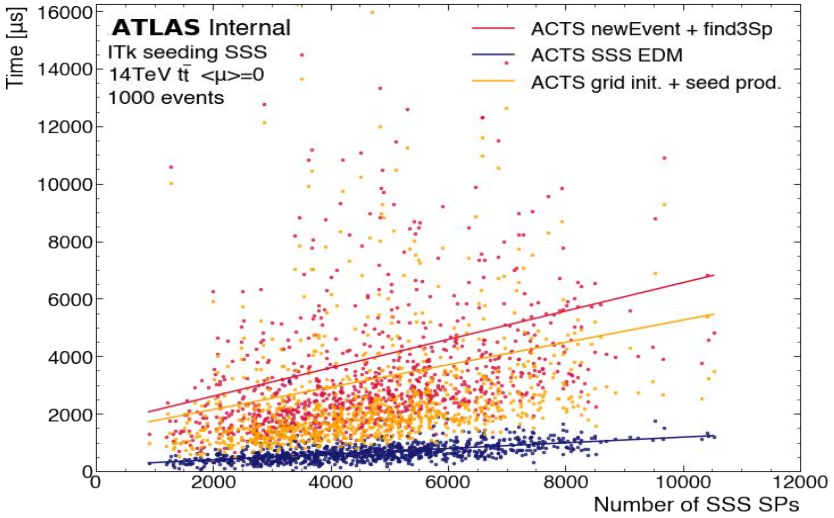


PPP ttbar mu=0 - EDM Time

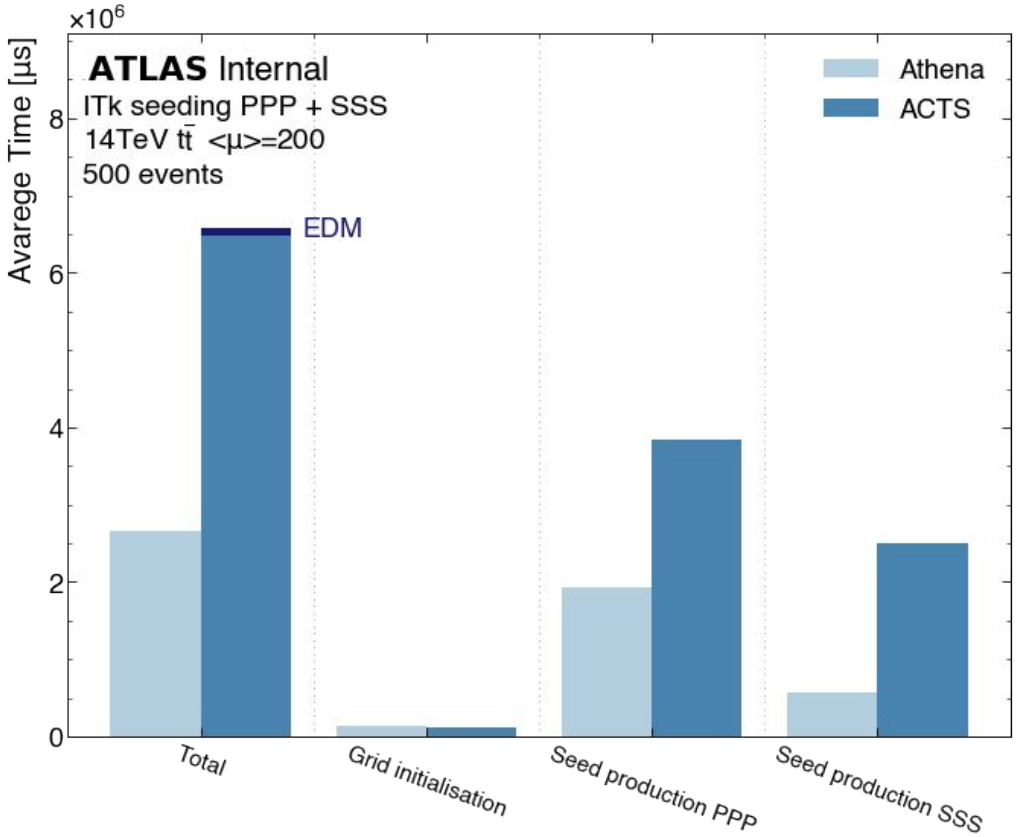
EDM Time PPP



EDM Time SSS



PPP+SSS ttbar mu=200



Situation gets worse when we look at mu=200:

- more SPs, more seed candidates, more memory

We need to work a bit on improving the seeding in ACTS:

- Find the hot spots in ACTS. Run heaptrack... check if if we allocate memory or move memory around
- Run a time profiler

More plots in backup

Time Profile

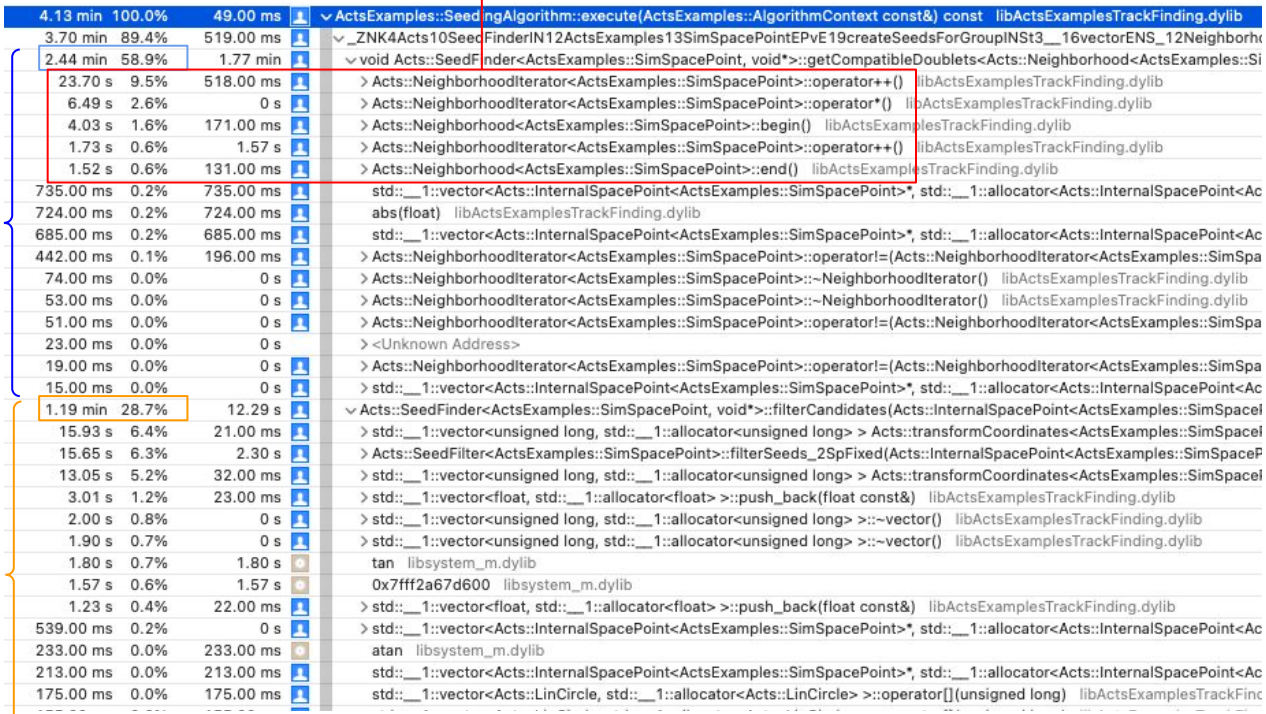
Time Profile - ACTS Standalone Reference - ttbar <μ>=200

From now on, the results include Carlo's latest changes, which slightly improve the time
PR #1764 [refactor: Managing Seed Candidates with ad-hoc container]

I ran a Time Profiler to find hot spots in ACTS: **Neighbourhood iterators** seem to take too much time

getCompatibleDoublets() - 58.9%

filterCandidates() - 28.7%



Neighborhood Iterators - New Strategy

In ACTS, when we loop over the SPs and produce the neighboring groups of SPs, we use fancy iterators, whereas in Athena we use simple for loops over the necessary bins of the SP grid:

In Athena we pass neighboring bins to the seeding and we iterate first over each bin and then over the top/bottom SPs in that cell:

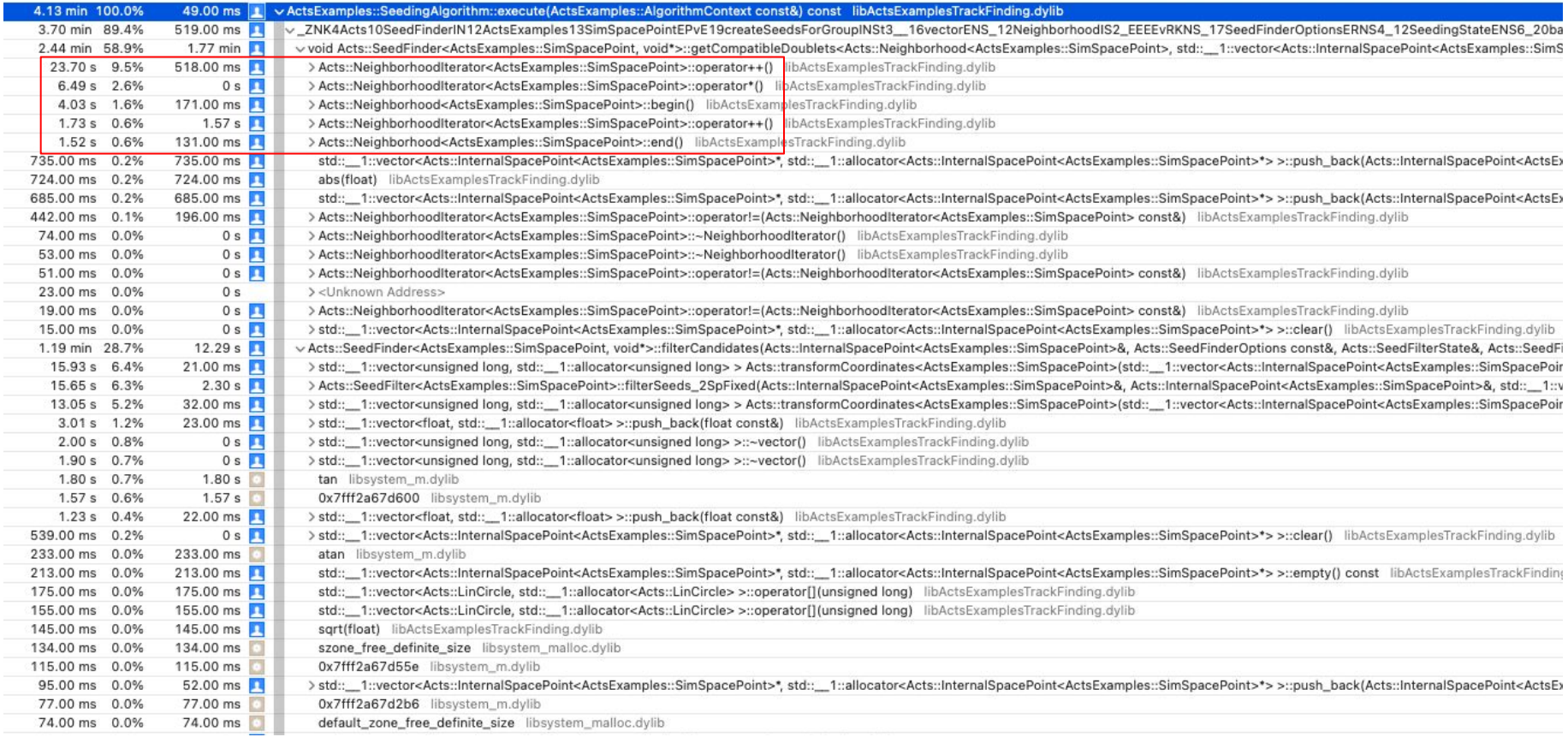
- This way we can also keep the initial sorting of the SPs in radius for each bin
- We can use this sorting to break out of the duplet loop if outside the region of interest - this avoids unnecessary iterations

I implemented the same strategy from Athena in ACTS and compared 3 scenarios in standalone:

- Main ACTS reference
- Monitored strategy without using the neighborhood iterators
- Monitored strategy without using the neighborhood iterators + breaking out of duplets loop

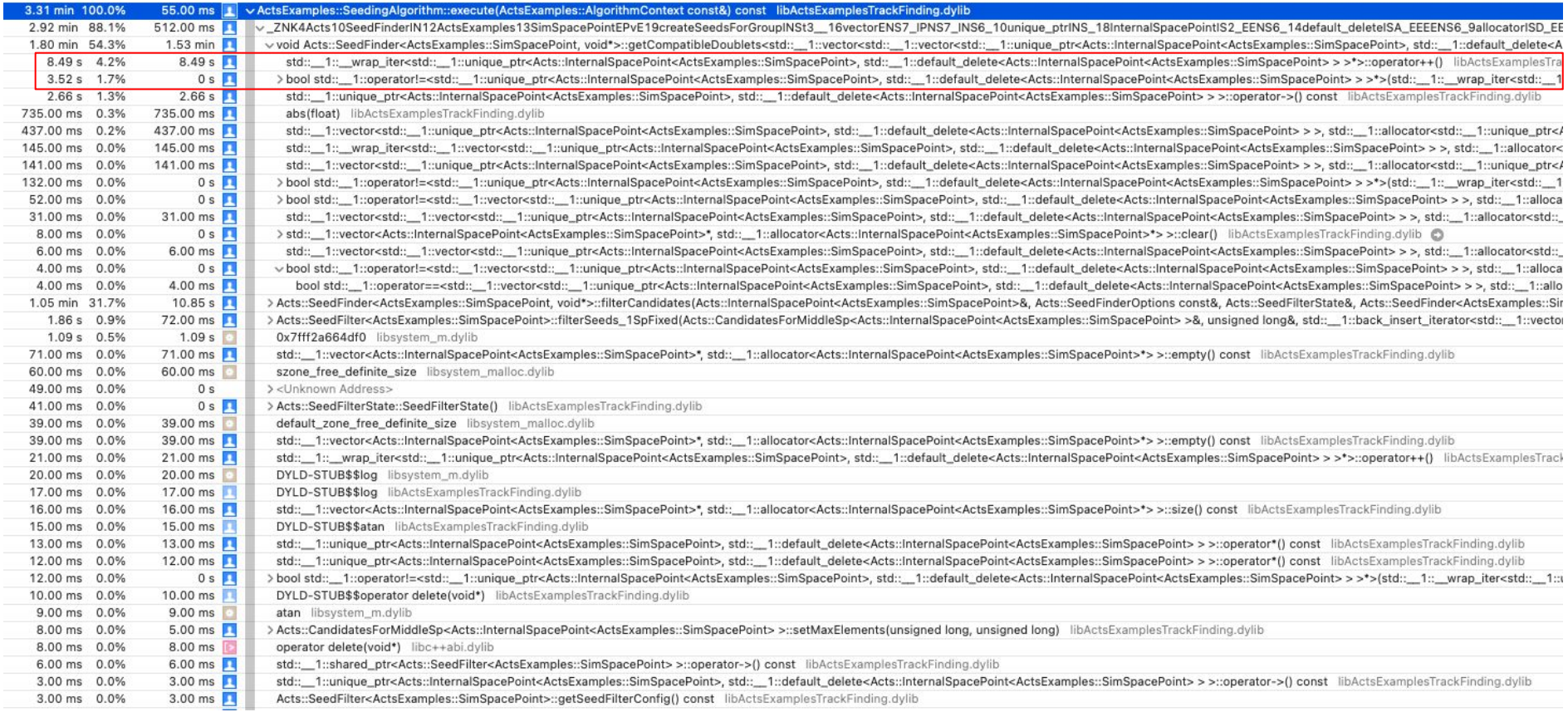
The physics performance between them is the same

Time Profile - ACTS Standalone Reference



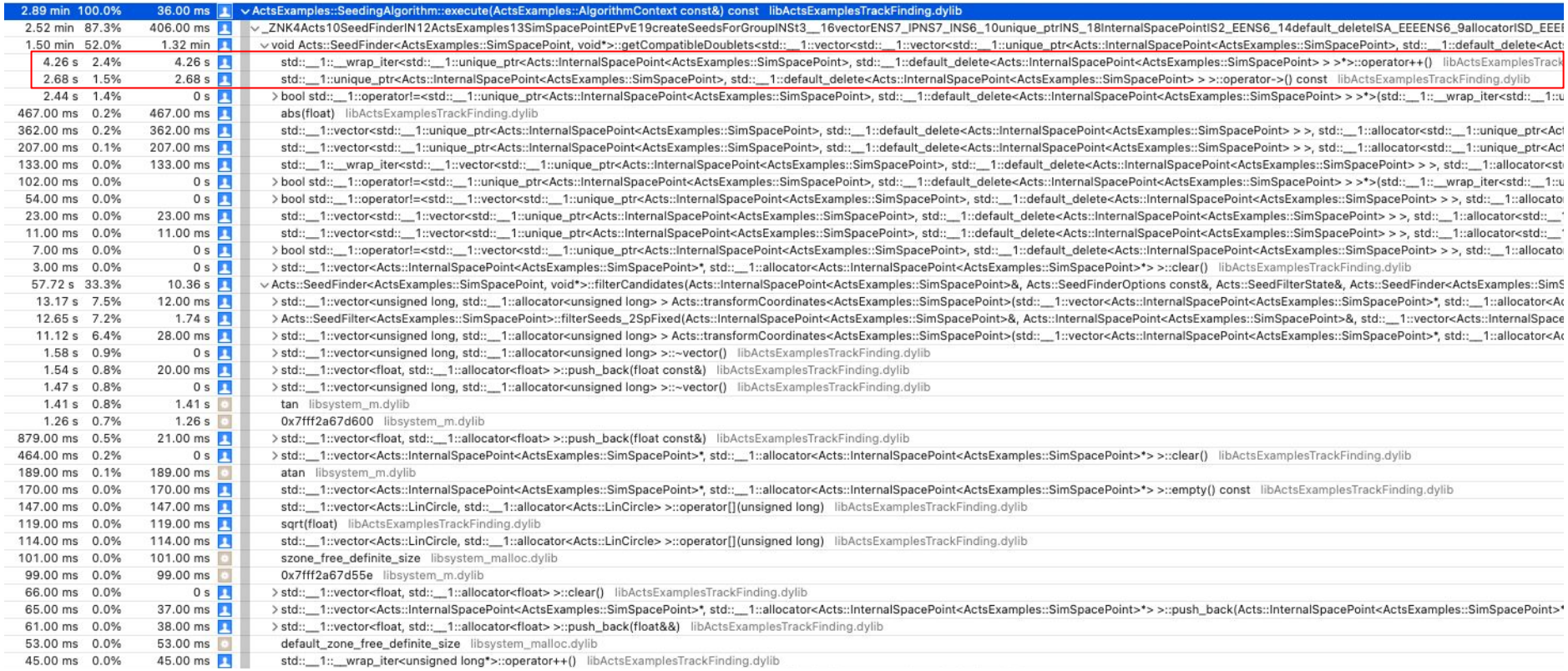
4.13 min	100.0%	49.00 ms	ActsExamples::SeedingAlgorithm::execute(ActsExamples::AlgorithmContext const&) const libActsExamplesTrackFinding.dylib
3.70 min	89.4%	519.00 ms	_ZNK4Acts10SeedFinderIN12ActsExamples13SimSpacePointEPvE19createSeedsForGroupINSt3__16vectorENS_12NeighborhoodIS2_EEEEvRKNS_17SeedFinderOptionsERNS4_12SeedingStateENS6_20ba
2.44 min	58.9%	1.77 min	void Acts::SeedFinder<ActsExamples::SimSpacePoint, void*>::getCompatibleDoublets<Acts::Neighborhood<ActsExamples::SimSpacePoint>, std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimS
23.70 s	9.5%	518.00 ms	Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint>::operator++() libActsExamplesTrackFinding.dylib
6.49 s	2.6%	0 s	Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint>::operator*() libActsExamplesTrackFinding.dylib
4.03 s	1.6%	171.00 ms	Acts::Neighborhood<ActsExamples::SimSpacePoint>::begin() libActsExamplesTrackFinding.dylib
1.73 s	0.6%	1.57 s	Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint>::operator++() libActsExamplesTrackFinding.dylib
1.52 s	0.6%	131.00 ms	Acts::Neighborhood<ActsExamples::SimSpacePoint>::end() libActsExamplesTrackFinding.dylib
735.00 ms	0.2%	735.00 ms	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::push_back(Acts::InternalSpacePoint<ActsEx
724.00 ms	0.2%	724.00 ms	abs(float) libActsExamplesTrackFinding.dylib
685.00 ms	0.2%	685.00 ms	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::push_back(Acts::InternalSpacePoint<ActsEx
442.00 ms	0.1%	196.00 ms	Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint>::operator!=(Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint> const&) libActsExamplesTrackFinding.dylib
74.00 ms	0.0%	0 s	Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint>::~NeighborhoodIterator() libActsExamplesTrackFinding.dylib
53.00 ms	0.0%	0 s	Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint>::~NeighborhoodIterator() libActsExamplesTrackFinding.dylib
51.00 ms	0.0%	0 s	Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint>::operator!=(Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint> const&) libActsExamplesTrackFinding.dylib
23.00 ms	0.0%	0 s	<Unknown Address>
19.00 ms	0.0%	0 s	Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint>::operator!=(Acts::NeighborhoodIterator<ActsExamples::SimSpacePoint> const&) libActsExamplesTrackFinding.dylib
15.00 ms	0.0%	0 s	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::clear() libActsExamplesTrackFinding.dylib
1.19 min	28.7%	12.29 s	Acts::SeedFinder<ActsExamples::SimSpacePoint, void*>::filterCandidates(Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>&, Acts::SeedFinderOptions const&, Acts::SeedFilterState&, Acts::SeedFi
15.93 s	6.4%	21.00 ms	std::__1::vector<unsigned long, std::__1::allocator<unsigned long> > Acts::transformCoordinates<ActsExamples::SimSpacePoint>(std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoin
15.65 s	6.3%	2.30 s	Acts::SeedFilter<ActsExamples::SimSpacePoint>::filterSeeds_2SpFixed(Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>&, Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>&, std::__1::v
13.05 s	5.2%	32.00 ms	std::__1::vector<unsigned long, std::__1::allocator<unsigned long> > Acts::transformCoordinates<ActsExamples::SimSpacePoint>(std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoin
3.01 s	1.2%	23.00 ms	std::__1::vector<float, std::__1::allocator<float> >::push_back(float const&) libActsExamplesTrackFinding.dylib
2.00 s	0.8%	0 s	std::__1::vector<unsigned long, std::__1::allocator<unsigned long> >::~vector() libActsExamplesTrackFinding.dylib
1.90 s	0.7%	0 s	std::__1::vector<unsigned long, std::__1::allocator<unsigned long> >::~vector() libActsExamplesTrackFinding.dylib
1.80 s	0.7%	1.80 s	tan libsystem_m.dylib
1.57 s	0.6%	1.57 s	0x7fff2a67d600 libsystem_m.dylib
1.23 s	0.4%	22.00 ms	std::__1::vector<float, std::__1::allocator<float> >::push_back(float const&) libActsExamplesTrackFinding.dylib
539.00 ms	0.2%	0 s	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::clear() libActsExamplesTrackFinding.dylib
233.00 ms	0.0%	233.00 ms	atan libsystem_m.dylib
213.00 ms	0.0%	213.00 ms	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::empty() const libActsExamplesTrackFindin
175.00 ms	0.0%	175.00 ms	std::__1::vector<Acts::LinCircle, std::__1::allocator<Acts::LinCircle> >::operator[](unsigned long) libActsExamplesTrackFinding.dylib
155.00 ms	0.0%	155.00 ms	std::__1::vector<Acts::LinCircle, std::__1::allocator<Acts::LinCircle> >::operator[](unsigned long) libActsExamplesTrackFinding.dylib
145.00 ms	0.0%	145.00 ms	sqrt(float) libActsExamplesTrackFinding.dylib
134.00 ms	0.0%	134.00 ms	szone_free_definite_size libsystem_malloc.dylib
115.00 ms	0.0%	115.00 ms	0x7fff2a67d55e libsystem_m.dylib
95.00 ms	0.0%	52.00 ms	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::push_back(Acts::InternalSpacePoint<ActsEx
77.00 ms	0.0%	77.00 ms	0x7fff2a67d2b6 libsystem_m.dylib
74.00 ms	0.0%	74.00 ms	default_zone_free_definite_size libsystem_malloc.dylib

Time Profile - ACTS Standalone Monitored Strategy



3.31 min	100.0%	55.00 ms	ActsExamples::SeedingAlgorithm::execute(ActsExamples::AlgorithmContext const&) const libActsExamplesTrackFinding.dylib
2.92 min	88.1%	512.00 ms	_ZNK4Acts10SeedFinderIN12ActsExamples13SimSpacePointEPvE19createSeedsForGroupINSt3__16vectorENS7_IPNS7_INS6_10unique_ptrINS_18InternalSpacePointIS2_EENS6_14default_deleteISA_EEEENS6_9allocatorISD_EE
1.80 min	54.3%	1.53 min	void Acts::SeedFinder<ActsExamples::SimSpacePoint, void*>::getCompatibleDoublets<std::__1::vector<std::__1::vector<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<A
8.49 s	4.2%	8.49 s	std::__1::__wrap_iter<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >*>::operator++() libActsExamplesTra
3.52 s	1.7%	0 s	bool std::__1::operator!=<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >*>(std::__1::__wrap_iter<std::__1
2.66 s	1.3%	2.66 s	std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >::operator->() const libActsExamplesTrackFinding.dylib
735.00 ms	0.3%	735.00 ms	abs(float) libActsExamplesTrackFinding.dylib
437.00 ms	0.2%	437.00 ms	std::__1::vector<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >, std::__1::allocator<std::__1::unique_ptr<
145.00 ms	0.0%	145.00 ms	std::__1::__wrap_iter<std::__1::vector<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >, std::__1::allocator<
141.00 ms	0.0%	141.00 ms	std::__1::vector<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >, std::__1::allocator<std::__1::unique_ptr<
132.00 ms	0.0%	0 s	bool std::__1::operator!=<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >*>(std::__1::__wrap_iter<std::__1
52.00 ms	0.0%	0 s	bool std::__1::operator!=<std::__1::vector<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >, std::__1::alloca
31.00 ms	0.0%	31.00 ms	std::__1::vector<std::__1::vector<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >, std::__1::allocator<std::_
8.00 ms	0.0%	0 s	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::clear() libActsExamplesTrackFinding.dylib
6.00 ms	0.0%	6.00 ms	std::__1::vector<std::__1::vector<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >, std::__1::allocator<std::_
4.00 ms	0.0%	0 s	bool std::__1::operator!=<std::__1::vector<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >, std::__1::alloca
4.00 ms	0.0%	4.00 ms	bool std::__1::operator==<std::__1::vector<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >, std::__1::allo
1.05 min	31.7%	10.85 s	Acts::SeedFinder<ActsExamples::SimSpacePoint, void*>::filterCandidates(Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>&, Acts::SeedFinderOptions const&, Acts::SeedFilterState&, Acts::SeedFinder<ActsExamples::Sir
1.86 s	0.9%	72.00 ms	Acts::SeedFilter<ActsExamples::SimSpacePoint>::filterSeeds_1SpFixed(Acts::CandidatesForMiddleSp<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> >&, unsigned long&, std::__1::back_insert_iterator<std::__1::vecto
1.09 s	0.5%	1.09 s	0x7fff2a664df0 libsystem_m.dylib
71.00 ms	0.0%	71.00 ms	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::empty() const libActsExamplesTrackFinding.dylib
60.00 ms	0.0%	60.00 ms	szone_free_definite_size libsystem_malloc.dylib
49.00 ms	0.0%	0 s	<Unknown Address>
41.00 ms	0.0%	0 s	Acts::SeedFilterState::SeedFilterState() libActsExamplesTrackFinding.dylib
39.00 ms	0.0%	39.00 ms	default_zone_free_definite_size libsystem_malloc.dylib
39.00 ms	0.0%	39.00 ms	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::empty() const libActsExamplesTrackFinding.dylib
21.00 ms	0.0%	21.00 ms	std::__1::__wrap_iter<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >*>::operator++() libActsExamplesTrack
20.00 ms	0.0%	20.00 ms	DYLD-STUB$$log libsystem_m.dylib
17.00 ms	0.0%	17.00 ms	DYLD-STUB$$log libActsExamplesTrackFinding.dylib
16.00 ms	0.0%	16.00 ms	std::__1::vector<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*, std::__1::allocator<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>*> >::size() const libActsExamplesTrackFinding.dylib
15.00 ms	0.0%	15.00 ms	DYLD-STUB$$atan libActsExamplesTrackFinding.dylib
13.00 ms	0.0%	13.00 ms	std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >::operator*() const libActsExamplesTrackFinding.dylib
12.00 ms	0.0%	12.00 ms	std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >::operator*() const libActsExamplesTrackFinding.dylib
12.00 ms	0.0%	0 s	bool std::__1::operator!=<std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >*>(std::__1::__wrap_iter<std::__1::u
10.00 ms	0.0%	10.00 ms	DYLD-STUB$$operator delete(void*) libActsExamplesTrackFinding.dylib
9.00 ms	0.0%	9.00 ms	atan libsystem_m.dylib
8.00 ms	0.0%	5.00 ms	Acts::CandidatesForMiddleSp<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> >::setMaxElements(unsigned long, unsigned long) libActsExamplesTrackFinding.dylib
8.00 ms	0.0%	8.00 ms	operator delete(void*) libc++abi.dylib
6.00 ms	0.0%	6.00 ms	std::__1::shared_ptr<Acts::SeedFilter<ActsExamples::SimSpacePoint> >::operator->() const libActsExamplesTrackFinding.dylib
3.00 ms	0.0%	3.00 ms	std::__1::unique_ptr<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint>, std::__1::default_delete<Acts::InternalSpacePoint<ActsExamples::SimSpacePoint> > >::operator->() const libActsExamplesTrackFinding.dylib
3.00 ms	0.0%	3.00 ms	Acts::SeedFilter<ActsExamples::SimSpacePoint>::getSeedFilterConfig() const libActsExamplesTrackFinding.dylib

Time Profile - ACTS Standalone Monitored Strategy + Break Duplets

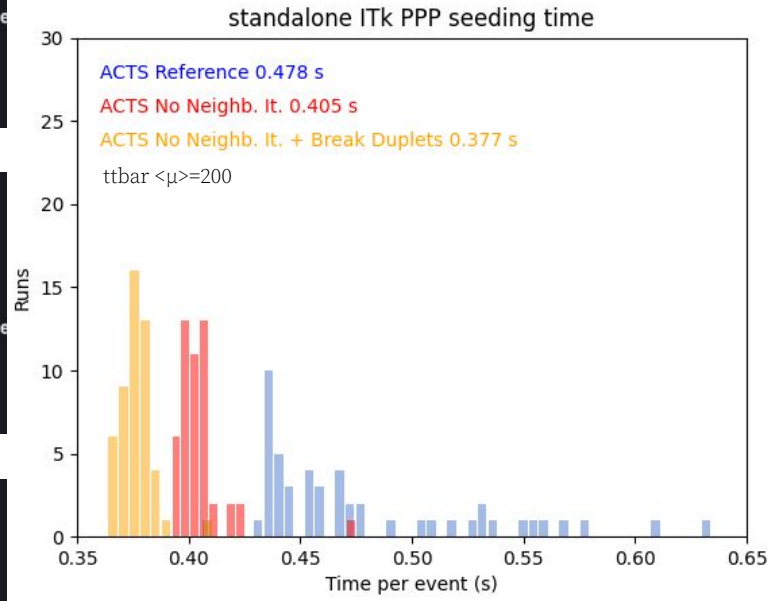


Total Seeding Time - 100 runs of ttbar <μ>=200 events

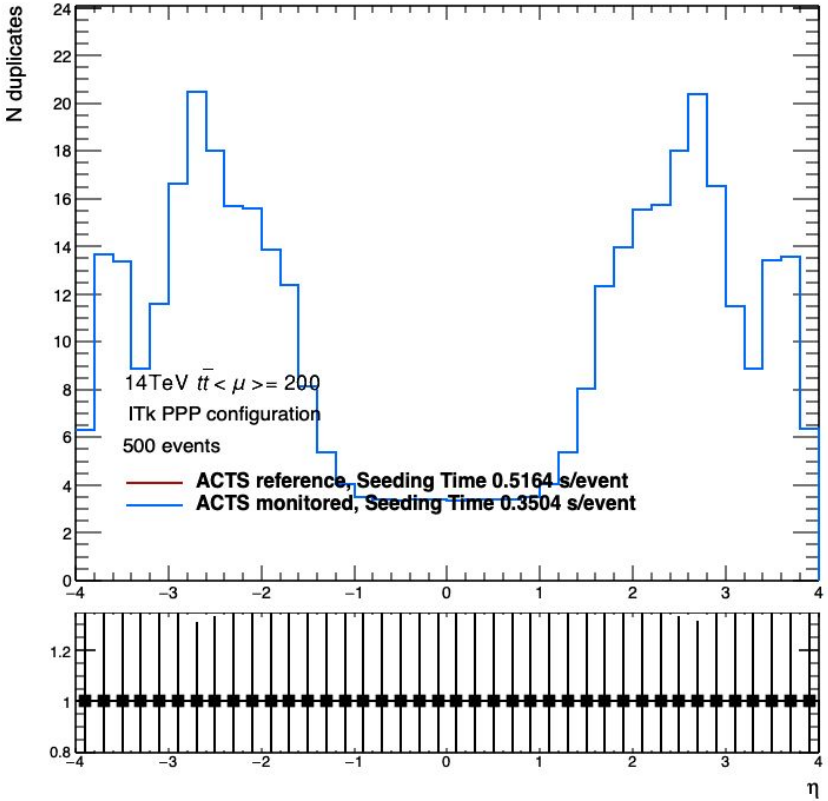
```
Reference:
500 / 500 events processed
Efficiency (nMatchedParticles / nAllParticles) = 0.96534
Fake rate (nUnMatchedSeeds / nAllSeeds) = 0.028693
Duplication rate (nDuplicatedMatchedParticles / nMatchedParticles) = 0.989692
Average number of duplicated seeds ((nMatchedSeeds - nMatchedParticles) / nMatchedParticle
Processed 500 events in 2238.503144 s (wall clock)
Average time per event: 4.454888 s/event
Algorithm:SeedingAlgorithm Total: 258.2 s, 0.5164 s/event
```

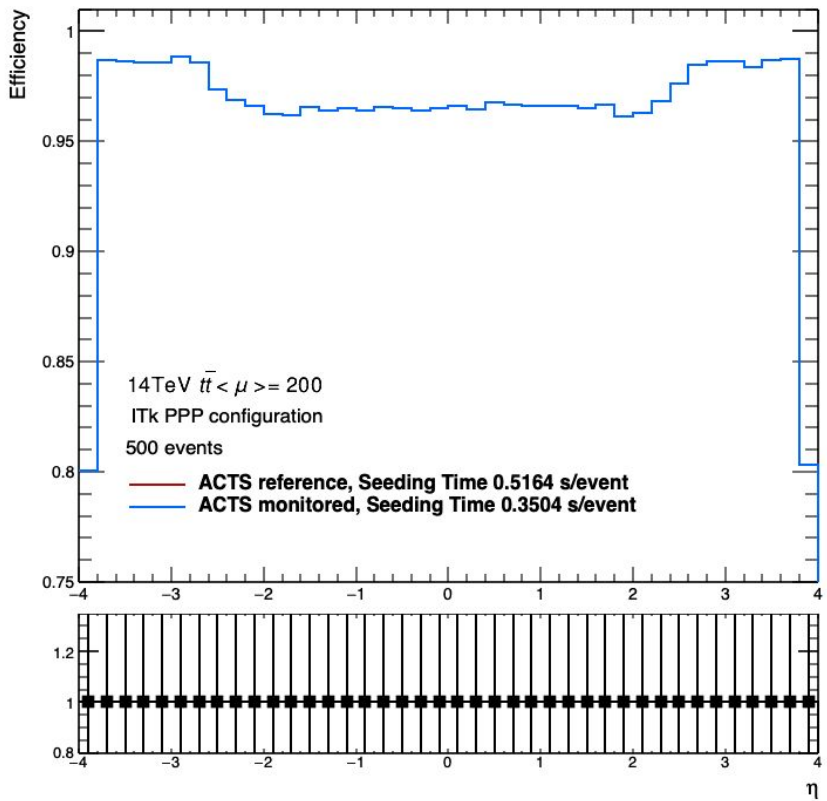
```
Monitored - No Iterators:
500 / 500 events processed
Efficiency (nMatchedParticles / nAllParticles) = 0.96534
Fake rate (nUnMatchedSeeds / nAllSeeds) = 0.028693
Duplication rate (nDuplicatedMatchedParticles / nMatchedParticles) = 0.989692
Average number of duplicated seeds ((nMatchedSeeds - nMatchedParticles) / nMatchedParticle
Processed 500 events in 1975.206154 s (wall clock)
Average time per event: 3.928311 s/event
Algorithm:SeedingAlgorithm Total: 209.4 s, 0.4188 s/event
```

```
Monitored - No Iterators + Break Duplets:
500 / 500 events processed
Efficiency (nMatchedParticles / nAllParticles) = 0.96534
Fake rate (nUnMatchedSeeds / nAllSeeds) = 0.028693
Duplication rate (nDuplicatedMatchedParticles / nMatchedParticles) = 0.989692
Average number of duplicated seeds ((nMatchedSeeds - nMatchedParticles) / nMatchedParticles) = 15.1778
Processed 500 events in 1762.167427 s (wall clock)
Average time per event: 3.504794 s/event
Algorithm:SeedingAlgorithm Total: 175.2 s, 0.3504 s/event
```



Physics Performance





Summary

Improvement in performance with the third strategy - probably not enough to make up the difference between ACTS and Athena

Should we change that in ACTS?

Next steps:

Compare the changes in Athena

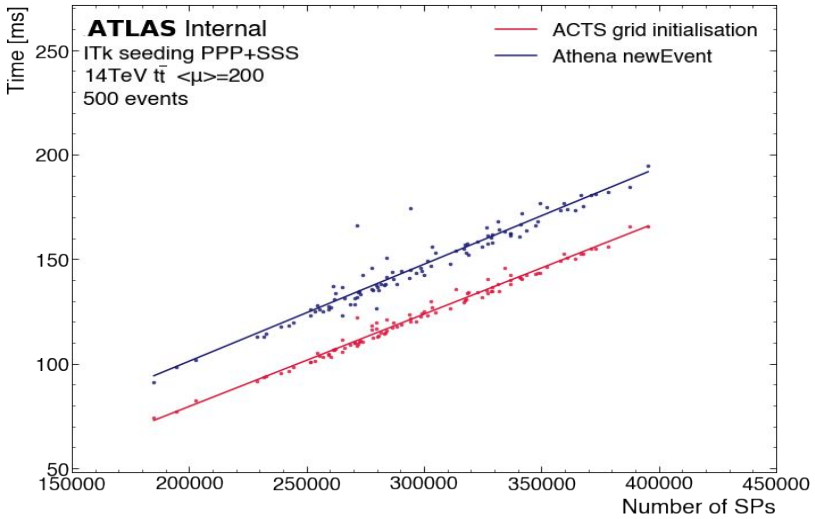
Work on improving seeding time in ACTS:

- Run Heaptrack to identify where the hot spots are in ACTS
- Implement the same approach as in Athena where we create seeds in batches

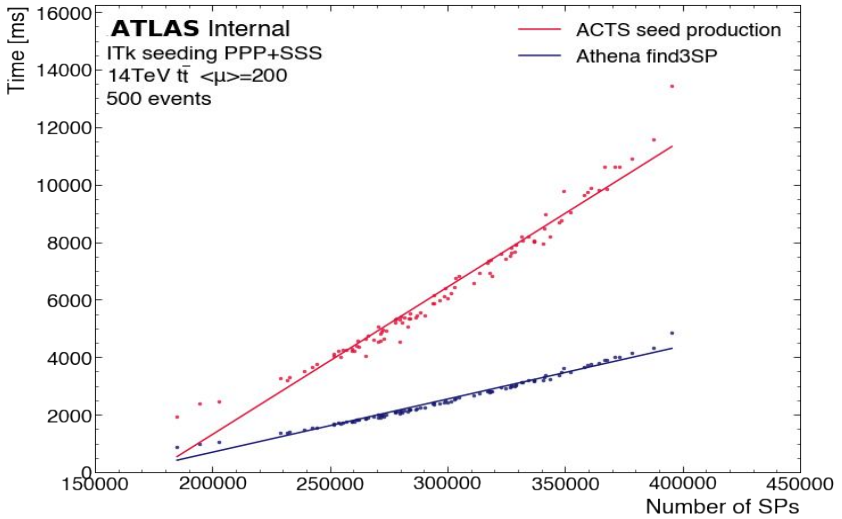
Backup

PPP+SSS ttbar mu=200

Grid Time PPP+SSS

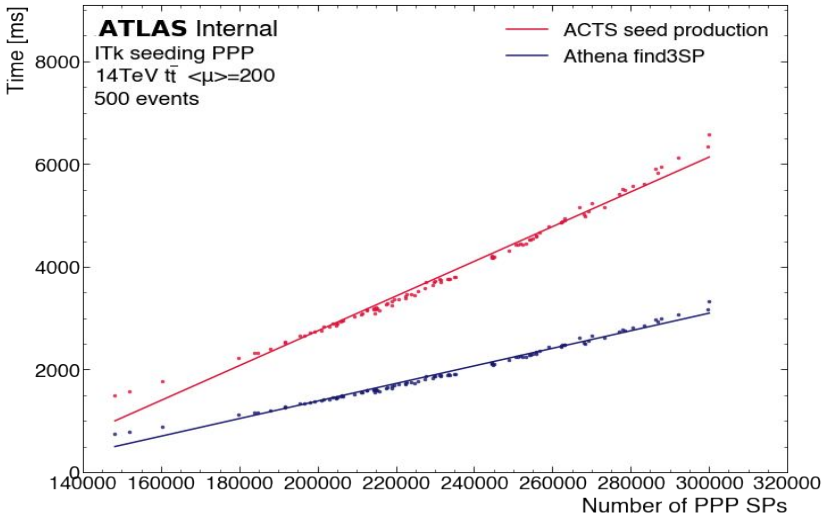


Seed Production Time PPP+SSS

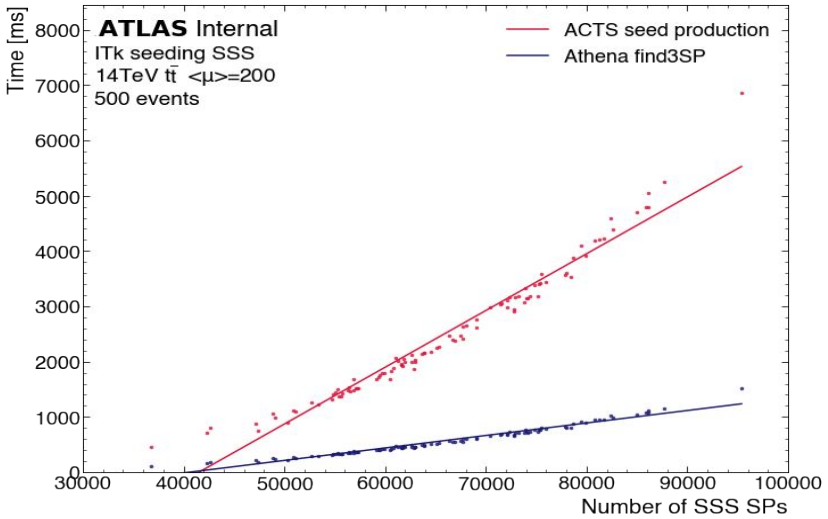


PPP+SSS ttbar mu=200

Seed Production Time PPP

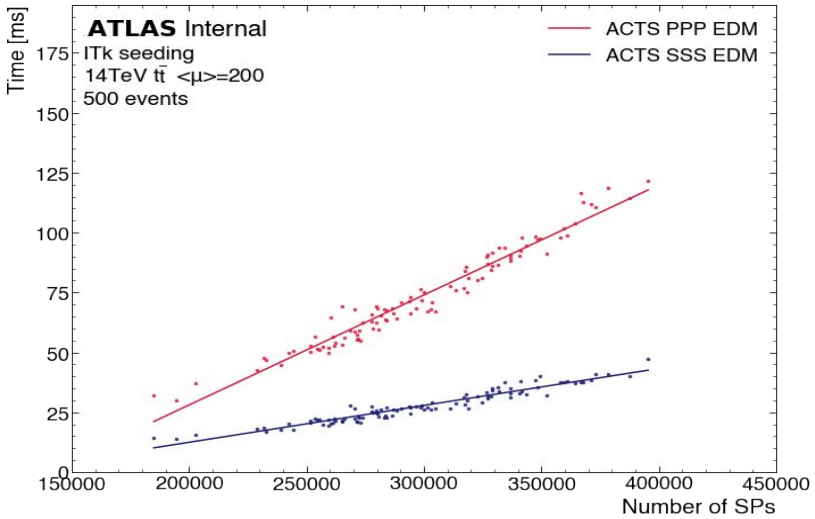


Seed Production Time SSS

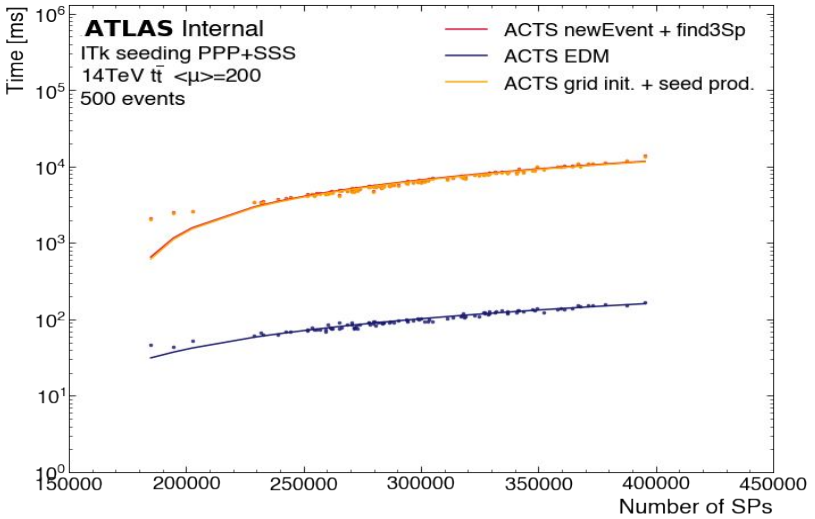


PPP+SSS ttbar mu=200 - EDM Time

EDM Time PPP vs SSS

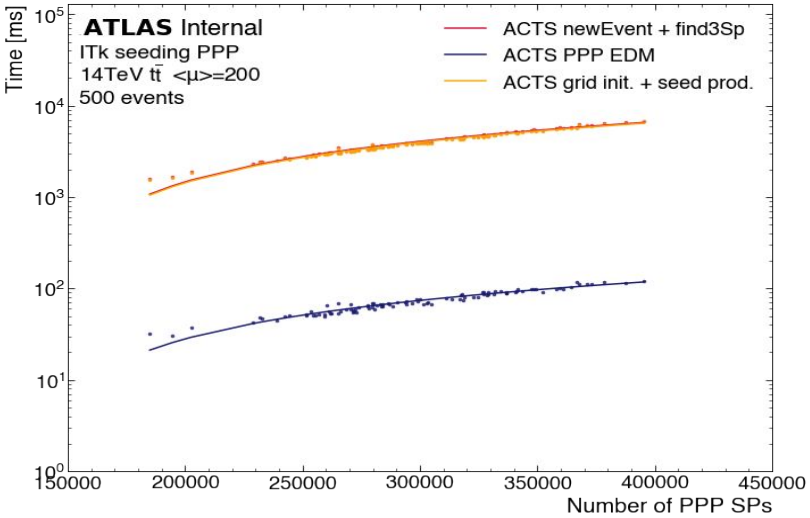
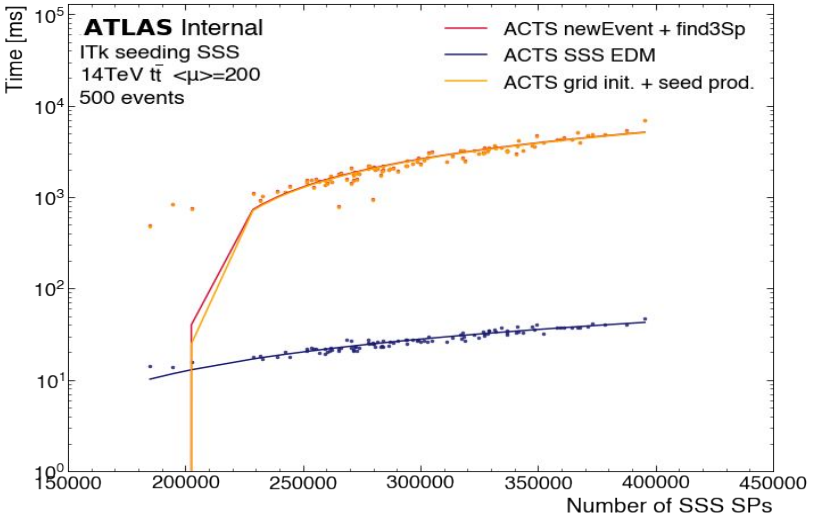


EDM Time PPP+SSS



PPP ttbar mu=200 - EDM Time

EDM Time PPP



EDM Time SSS