Some little studies

Idea

- On the basis, that probably the hot stuff happens where the CPU time is spend, I had a look into the timing.

Setup

```
basf2 -n 10000 createSimFile.py
```

```
# We might want to have particle gun(s) and EVTGen.
# Still in that case the ParticleGun modules better come first:
param_pGun = {
    'pdgCodes': [13, -13],                  # 13 = muon --> negatively charged!
    'nTracks': 20,                         # 20 tracks is a lot, but we don't use beam background in this sc
    'momentumGeneration': 'uniformPt',
    'momentumParams': [0.1, 0.15],          # 2 values: [min, max] in GeV
    'thetaGeneration': 'uniform',
    'thetaParams': [60., 85.],              # 2 values: [min, max] in degree
    'phiGeneration': 'uniform',
    'phiParams': [0., 90.],                 # [min, max] in degree
    'vertexGeneration': 'uniform',
    'xVertexParams': [-0.1, 0.1],           # in cm...
    'yVertexParams': [-0.1, 0.1],
    'zVertexParams': [-0.5, 0.5],
}

particlegun = register_module('ParticleGun')
particlegun.logging.log_level = LogLevel.WARNING
particlegun.param(param_pGun)
main.add_module(particlegun)

# Now we might want to add EvtGen:
if False:
    evtgenInput = register_module('EvtGenInput')
    evtgenInput.logging.log_level = LogLevel.WARNING
    main.add_module(evtgenInput)

# Gearbox to access stuff from the data folders, and Geometry:
gearbox = register_module('Gearbox')
main.add_module(gearbox)

geometry = register_module('Geometry')
geometry.param('components', ['BeamPipe', 'MagneticFieldConstant4LimitedRSVD',  # Important: look at B field!
                              'PXD', 'SVD'])
```

```
basf2 -n 10 -i MyRootFile.root testVXDTFRelatedModules.py
```

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	10	0	0.02	1.89 +- 0.32
EventInfoPrinter	10	0	0.00	0.00 +- 0.00
Gearbox	10	0	0.00	0.00 +- 0.00
Geometry	10	0	0.00	0.00 +- 0.00
EventCounter	10	0	0.00	0.05 +- 0.09
SectorMapBootstrap	10	0	0.00	0.00 +- 0.00
SpacePointCreatorSVD	10	0	0.01	1.18 +- 0.29
SpacePointCreatorPXD	10	0	0.00	0.21 +- 0.09
SpacePoint2TrueHitConnector	10	0	0.02	1.99 +- 0.81
GFTC2SPTCConverter	10	0	0.02	1.52 +- 0.45
SPTCReferee	10	0	0.00	0.20 +- 0.07
SpacePoint2TrueHitConnector	10	0	0.04	4.37 +- 0.65
SecMapTrainerBase	10	0	0.01	0.96 +- 0.27
Total	10	0	0.13	12.70 +- 1.98

```
basf2 -n 10 -i MyRootFile.root testVXDTFRelatedModules.py\
```

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	10	0	0.02	2.23 +- 0.41
EventInfoPrinter	10	0	0.00	0.00 +- 0.00
Gearbox	10	0	0.00	0.00 +- 0.00
Geometry	10	0	0.00	0.00 +- 0.00
EventCounter	10	0	0.00	0.14 +- 0.06
SectorMapBootstrap	10	0	0.00	0.00 +- 0.00
SpacePointCreatorSVD	10	0	0.01	1.16 +- 0.20
SpacePointCreatorPXD	10	0	0.00	0.19 +- 0.03
SpacePoint2TrueHitConnector	10	0	0.02	1.86 +- 0.20
GFTC2SPTCConverter	10	0	0.01	1.40 +- 0.20
SPTCReferee	10	0	0.00	0.21 +- 0.06
SpacePoint2TrueHitConnector	10	0	0.05	5.44 +- 1.02
RawSecMapMerger	10	0	0.00	0.00 +- 0.00
SegmentNetworkProducer	10	0	13.04	1304.29 +- 387.11
TrackFinderVXDCellOMat	10	0	0.00	0.46 +- 0.20
SPTCvirtualIPRemover	10	0	0.00	0.01 +- 0.00
QualityEstimatorVXDRandom	10	0	0.00	0.00 +- 0.00
SPTCNetworkProducer	10	0	0.00	0.08 +- 0.06
TrackSetEvaluatorGreedy	10	0	0.00	0.11 +- 0.10
SVDOverlapChecker	10	0	0.01	0.54 +- 0.09
TrackSetEvaluatorGreedyDEV	10	0	0.00	0.20 +- 0.06
TrackFinderVXDAnalizer	10	0	0.03	2.54 +- 0.38
Total	10	0	13.22	1321.69 +- 388.31

And bad finding efficiency, despite the same events are used for training!

Segment Network Producer is slow, even when there are few friend relationships between sectors (as we have used only 10 events for the training), O(30) TrackCands

Note: There are 10,000 events in the file.

```
basf2 -i MyRootFile.root testVXDTFRelatedModules.py
```

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	10001	0	20.62	2.06 +- 15.31
EventInfoPrinter	10000	0	0.02	0.00 +- 0.00
Gearbox	10000	0	0.01	0.00 +- 0.00
Geometry	10000	0	0.01	0.00 +- 0.00
EventCounter	10000	0	0.29	0.03 +- 0.04
SectorMapBootstrap	10000	0	0.02	0.00 +- 0.00
SpacePointCreatorSVD	10000	0	13.04	1.30 +- 0.21
SpacePointCreatorPXD	10000	0	1.79	0.18 +- 0.02
SpacePoint2TrueHitConnector	10000	0	17.46	1.75 +- 0.21
GFTC2SPTCConverter	10000	0	13.12	1.31 +- 0.14
SPTCReferee	10000	0	1.86	0.19 +- 0.12
SpacePoint2TrueHitConnector	10000	0	42.44	4.24 +- 0.67
SecMapTrainerBase	10000	0	11.83	1.18 +- 1.17
Total	10001	0	129.40	12.94 +- 15.43

```
basf2 -n 10 -i MyRootFile.root testVXDTFRelatedModules.py\
```

4th event “breaks” my Computer;

```
basf2 -i MyRootFile.root -n 3 testVXDTFRelatedModules.py
```

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	3	0	0.01	1.92 +- 0.18
EventInfoPrinter	3	0	0.00	0.00 +- 0.00
Gearbox	3	0	0.00	0.00 +- 0.00
Geometry	3	0	0.00	0.00 +- 0.00
EventCounter	3	0	0.00	0.26 +- 0.17
SectorMapBootstrap	3	0	0.00	0.00 +- 0.00
SpacePointCreatorSVD	3	0	0.00	1.33 +- 0.34
SpacePointCreatorPXD	3	0	0.00	0.30 +- 0.13
SpacePoint2TrueHitConnector	3	0	0.01	1.99 +- 0.26
GFTC2SPTCConverter	3	0	0.00	1.58 +- 0.17
SPTCReferee	3	0	0.00	0.27 +- 0.08
SpacePoint2TrueHitConnector	3	0	0.01	4.74 +- 0.51
RawSecMapMerger	3	0	0.00	0.00 +- 0.00
SegmentNetworkProducer	3	0	7.68	2559.42 +- 260.50
TrackFinderVXDCellOMat	3	0	0.13	43.59 +- 18.38
SPTCvirtualIPRemover	3	0	0.00	0.58 +- 0.26
QualityEstimatorVXDRandom	3	0	0.00	0.07 +- 0.02
SPTCNetworkProducer	3	0	49.25	16418.10 +-15926.02
TrackSetEvaluatorGreedy	3	0	96.15	32050.80 +-30802.89
SVDOverlapChecker	3	0	3.34	1113.93 +- 830.97
TrackSetEvaluatorGreedyDEV	3	0	0.10	32.93 +- 21.72
TrackFinderVXDAnalizer	3	0	0.01	2.42 +- 0.39
Total	3	0	156.75	52249.60 +-47335.36

Segment Network Producer gets only factor 2 slower, if there is a ton of friend relations…

New modules SVDOverlapChecker and TrackSetEvaluatorGreedyDEV scale well with large number of TrackCandidates (O(10,000));

Let’s see for Hopfield, but I’m optimistic.

```
isActive, Array Index: 11, QI: 0.825849
isActive, Array Index: 89, QI: 0.988963
isActive, Array Index: 2026, QI: 0.996657
isActive, Array Index: 4389, QI: 0.997194
isActive, Array Index: 4851, QI: 0.999532
isActive, Array Index: 6139, QI: 0.99964
isActive, Array Index: 6256, QI: 0.999548
isActive, Array Index: 6308, QI: 0.773476
isActive, Array Index: 7341, QI: 0.958889
isActive, Array Index: 7389, QI: 0.995538
isActive, Array Index: 7447, QI: 0.999933
isActive, Array Index: 7499, QI: 0.473606
Array Index: 7447, QI: 0.999933
Array Index: 6139, QI: 0.99964
Array Index: 6256, QI: 0.999548
Array Index: 4851, QI: 0.999532
Array Index: 4389, QI: 0.997194
Array Index: 2026, QI: 0.996657
Array Index: 7389, QI: 0.995538
Array Index: 89, QI: 0.988963
Array Index: 7341, QI: 0.958889
Array Index: 11, QI: 0.825849
Array Index: 6308, QI: 0.773476
Array Index: 7499, QI: 0.473606
```

First set comes from Jakob's Greedy, second is the result of my Greedy of 3rd event.
They are the same (they differ with the fitter as QI estimator, as they act differently, if there are several candidates with the same QI value).

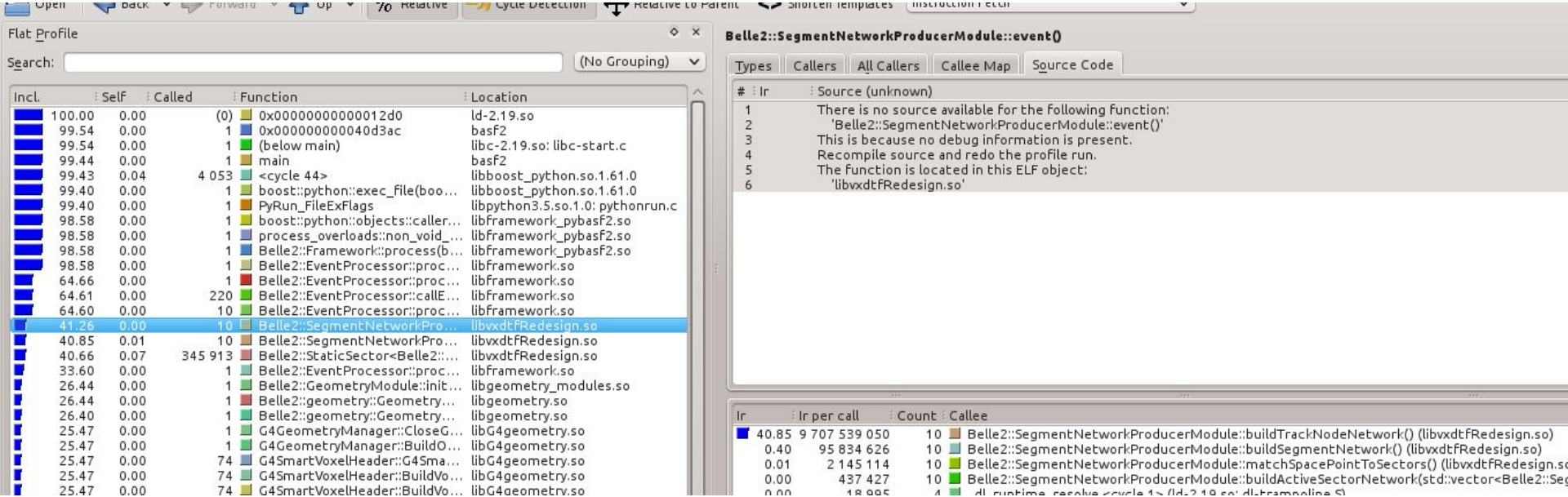
Conclusion of Study of differently trained secMaps

- Preliminary, possibly wrong: [Multi-Pass doesn’t solve our problems, as even with sparse secMap, the SegmentNetworkProducer takes fairly long for a difficult event.]
- Timing strongly dependent on implementation of stuff → Stefano will not get a good intuition on the real timing with using the new trackFinding setup.

Look into SegmentNetworkProducerModule

- Found a swamp of code, that in some parts looks very optimized, but has as well tons of pitfalls regarding:
  - Branch misprediction (“If-statements”),
  - Lack of cache locality (Pointers or vectors of pointers, where vectors of the object itself could be),
  - debugging and supervising information, that isn’t strictly needed, and
  - “Keep it save” checks on validity of data, where having non-valid data simply shouldn’t occur (in this respect, the CDC code is as well not a good example).

Study the Timing of the event loop of SegmentNet...



Belle2::SegmentNetworkProducerModule::buildTrackNodeNetwork()

Types | Callers | All Callers | Callee Map | Source Code

#	Ir	Source (unknown)
1		There is no source available for the following function:
2		'Belle2::SegmentNetworkProducerModule::buildTrackNodeNetwork()'
3		This is because no debug information is present.
4		Recompile source and redo the profile run.
5		The function is located in this ELF object:
6		'libvxdtfRedesign.so'

Ir	Ir per call	Count	Callee
40.66	279 332	345 913	Belle2::StaticSector<Belle2::SpacePoint, Belle2::Filter<Belle2::OperatorAnd, Belle2::Filter<Belle2::OperatorAnd, Belle2::Filter<Belle2::OperatorAnd, …
0.13	4 115	76 233	std::reverse_iterator<__gnu_cxx::__normal_iterator<Belle2::DirectedNode<Belle2::TrackNode, Belle2::VoidMetaInfo>**, std::vector<Belle2::Direct…
0.03	9 007	6 778	Belle2::DirectedNodeNetwork<Belle2::TrackNode, Belle2::VoidMetaInfo>::linkTheseEntries(Belle2::TrackNode&, Belle2::TrackNode&) (libvxdtfRedesi…
0.01	292	76 233	Belle2::DirectedNode<Belle2::TrackNode, Belle2::VoidMetaInfo>::addOuterNode(Belle2::DirectedNode<Belle2::TrackNode, Belle2::VoidMetaInfo>&) …
0.01	223	76 233	Belle2::DirectedNode<Belle2::TrackNode, Belle2::VoidMetaInfo>::addInnerNode(Belle2::DirectedNode<Belle2::TrackNode, Belle2::VoidMetaInfo>&) (…
0.01	90	152 466	__gnu_cxx::__normal_iterator<Belle2::DirectedNode<Belle2::TrackNode, Belle2::VoidMetaInfo>**, std::vector<Belle2::DirectedNode<Belle2::TrackN…
0.00	38	78 714	Belle2::LogSystem::isLevelEnabled(Belle2::LogConfig::ELogLevel, int, char const*) const (libframework.so)
0.00	630	2 471	Belle2::DirectedNodeNetwork<Belle2::TrackNode, Belle2::VoidMetaInfo>::addNode(Belle2::TrackNode&) (libvxdtfRedesign.so)
0.00	360	2 471	__gnu_cxx::__normal_iterator<unsigned int*, std::vector<unsigned int, std::allocator<unsigned int> > > std::find<__gnu_cxx::__normal_iterator<un…
0.00	4	78 714	Belle2::LogSystem::Instance() (libframework.so)
0.00	27	2 471	void std::vector<unsigned int, std::allocator<unsigned int> >::emplace_back<unsigned int>(unsigned int&&) (libRGL.so)
0.00	25 320	2	_dl_runtime_resolve <cycle 1> (ld-2.19.so: dl-trampoline.S)

Belle2::StaticSector<Belle2::SpacePoint, Belle2::Filter<Belle2::OperatorAnd, Belle2::Fi...t(Belle2::FullSecID const&, Belle2::SpacePoint const&, Belle2::SpacePoint const&) const

Types | Callers | All Callers | Callee Map | Source Code

#	Ir	Source (unknown)
1		There is no source available for the following function:
2		'Belle2::StaticSector<Belle2::SpacePoint, Belle2::Filter<Belle2::OperatorAnd, Belle2::Filter<Belle2::OperatorAnd, Belle2::Filter<Belle2::OperatorAnd, Belle2::Filter<Bell...
3		This is because no debug information is present.
4		Recompile source and redo the profile run.
5		The function is located in this ELF object:
6		'libvxdtfRedesign.so'

Ir	Ir per call	Count	Callee
22.34	65 257	813 545	Belle2::SelectionVariable<Belle2::SpacePoint, double>::name() const (libvxdtfRedesign.so)
9.51	66 019	342 318	TTree::Fill() (libTree.so)
8.43	58 492	342 318	Belle2::createPurityInfosVec(std::vector<Belle2::SpacePoint const*, std::allocator<Belle2::SpacePoint const*> > const&) [clone .constprop.941] (li...
0.15	217	1 627 090	std::map<std::string, bool*, std::less<std::string>, std::allocator<std::pair<std::string const, bool*> > >::at(std::string const&) (libvxdtfRedesign.so)
0.04	200	471 227	std::map<std::string, double*, std::less<std::string>, std::allocator<std::pair<std::string const, double*> > >::at(std::string const&) (libvxdtfRedesi...
0.03	100	813 545	std::string::_Rep::_M_dispose(std::allocator<char> const&) [clone .part.25] (libvxdtfRedesign.so)

OK, Problem is Observer setup

- Jakob told, me the problem are the Observers…
  - Next step: using VoidObservers and test the timing again.

```
diff --git a/tracking/modules/VXDTFHelperTools/src/RawSecMapMergerModule.cc b/tracking/modules/VXDTFHelperTools/src/RawSecMapMergerModule.cc
index 40b842a..0046637 100644
--- a/tracking/modules/VXDTFHelperTools/src/RawSecMapMergerModule.cc
+++ b/tracking/modules/VXDTFHelperTools/src/RawSecMapMergerModule.cc
@@ -457,18 +457,19 @@ template <class FilterType> void RawSecMapMergerModule::add2HitFilters(VXDTFFilt


   /// JKL Feb 2016 - big working example:
-//   VXDTFFilters<SpacePoint>::twoHitFilter_t friendSectorsSegmentFilter =
-//     (
-//       (filterCutsMap.at("Distance3DSquared").getMin() <= Distance3DSquared<SpacePoint>() <=
-//        filterCutsMap.at("Distance3DSquared").getMax()).observe(VoidObserver()) &&
-//       (filterCutsMap.at("Distance2DXYSquared").getMin() <= Distance2DXYSquared<SpacePoint>() <=
-//        filterCutsMap.at("Distance2DXYSquared").getMax()).observe(VoidObserver()) &&
-//       (filterCutsMap.at("Distance1DZ").getMin() <= Distance1DZ<SpacePoint>() <= filterCutsMap.at("Distance1DZ").getMax()).observe(
-//         VoidObserver()) &&
-//       (filterCutsMap.at("SlopeRZ").getMin() <= SlopeRZ<SpacePoint>() <= filterCutsMap.at("SlopeRZ").getMax()).observe(VoidObserver()) &&
-//       (filterCutsMap.at("Distance3DNormed").getMin() <= Distance3DNormed<SpacePoint>() <=
-//        filterCutsMap.at("Distance3DNormed").getMax()).observe(VoidObserver())
-//     );
+   VXDTFFilters<SpacePoint>::twoHitFilter_t friendSectorsSegmentFilter =
+     (
+       (filterCutsMap.at("Distance3DSquared").getMin() <= Distance3DSquared<SpacePoint>() <=
+        filterCutsMap.at("Distance3DSquared").getMax()).observe(VoidObserver()) &&
+       (filterCutsMap.at("Distance2DXYSquared").getMin() <= Distance2DXYSquared<SpacePoint>() <=
+        filterCutsMap.at("Distance2DXYSquared").getMax()).observe(VoidObserver()) &&
+       (filterCutsMap.at("Distance1DZ").getMin() <= Distance1DZ<SpacePoint>() <= filterCutsMap.at("Distance1DZ").getMax()).observe(
+         VoidObserver()) &&
+       (filterCutsMap.at("SlopeRZ").getMin() <= SlopeRZ<SpacePoint>() <= filterCutsMap.at("SlopeRZ").getMax()).observe(VoidObserver()) &&
+       (filterCutsMap.at("Distance3DNormed").getMin() <= Distance3DNormed<SpacePoint>() <=
+        filterCutsMap.at("Distance3DNormed").getMax()).observe(VoidObserver())
+     );
+/*********************************************
    VXDTFFilters<SpacePoint>::twoHitFilter_t friendSectorsSegmentFilter =
      (
        (
@@ -484,6 +485,7 @@ template <class FilterType> void RawSecMapMergerModule::add2HitFilters(VXDTFFilt
         ).observe(ObserverCheckMCPurity())
      );
+*************************************************/
```

```
diff --git a/tracking/trackFindingVXD/environment/include/VXDTFFilters.h b/tracking/trackFindingVXD/environment/include/VXDTFFilters.h
index 2306387..11058f4 100644
--- a/tracking/trackFindingVXD/environment/include/VXDTFFilters.h
+++ b/tracking/trackFindingVXD/environment/include/VXDTFFilters.h
@@ -65,7 +65,7 @@ namespace Belle2 {
     /// minimal working 2-hits-example used for redesign of VXDTF.
 //     typedef decltype((0. <= Distance3DSquared<Belle2::SpacePoint>() <= 0.).observe(ObserverPrintResults())) twoHitFilter_t;

-
+/*********************************************************************************
     /// big working 2-hits-example used for redesign of VXDTF.
     typedef decltype(
       (
@@ -76,15 +76,16 @@ namespace Belle2 {
       ).observe(ObserverCheckMCPurity())
     ) twoHitFilter_t;
+*********************************************************************************/

     // March9th2016: TODO: we want to use a big observer observing everything - Working title: MegaObserver.
-//   typedef decltype(
-//     ((0. <= Distance3DSquared<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())&&
-//     (0. <= Distance2DXYSquared<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())&&
-//     (0. <= Distance1DZ<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())&&
-//     (0. <= SlopeRZ<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())&&
-//     (0. <= Distance3DNormed<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())).observe(VoidObserver())
-//   ) twoHitFilter_t;
+  typedef decltype(
+    ((0. <= Distance3DSquared<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())&&
+    (0. <= Distance2DXYSquared<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())&&
+    (0. <= Distance1DZ<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())&&
+    (0. <= SlopeRZ<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())&&
+    (0. <= Distance3DNormed<Belle2::SpacePoint>() <= 0.).observe(VoidObserver())).observe(VoidObserver())
+  ) twoHitFilter_t;

     /// minimal working example for 3-hits:
 //   typedef decltype((0. <= Angle3DSimple<point_t>()    <= 0.).observe(Observer3HitPrintResults())) threeHitFilter_t;
```

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	10	0	0.02	2.05 +- 0.38
EventInfoPrinter	10	0	0.00	0.00 +- 0.00
Gearbox	10	0	0.00	0.00 +- 0.00
Geometry	10	0	0.00	0.00 +- 0.00
EventCounter	10	0	0.00	0.13 +- 0.05
SectorMapBootstrap	10	0	0.00	0.00 +- 0.00
SpacePointCreatorSVD	10	0	0.01	1.04 +- 0.14
SpacePointCreatorPXD	10	0	0.00	0.19 +- 0.03
SpacePoint2TrueHitConnector	10	0	0.02	1.84 +- 0.21
GFTC2SPTCConverter	10	0	0.01	1.40 +- 0.18
SPTCReferee	10	0	0.00	0.22 +- 0.05
SpacePoint2TrueHitConnector	10	0	0.05	5.32 +- 0.89
RawSecMapMerger	10	0	0.00	0.00 +- 0.00
SegmentNetworkProducer	10	0	0.04	4.02 +- 1.75
TrackFinderVXDCellOMat	10	0	0.03	2.67 +- 7.06
SPTCvirtualIPRemover	10	0	0.00	0.01 +- 0.00
QualityEstimatorVXDRandom	10	0	0.00	0.00 +- 0.00
SPTCNetworkProducer	10	0	0.00	0.08 +- 0.07
TrackSetEvaluatorGreedy	10	0	0.00	0.10 +- 0.10
SVDOverlapChecker	10	0	0.00	0.38 +- 0.04
TrackSetEvaluatorGreedyDEV	10	0	0.00	0.17 +- 0.04
TrackFinderVXDAnalizer	10	0	0.03	2.80 +- 0.89
Total	10	0	0.23	22.91 +- 7.76

With “little” SecMap. Nothing relevant...

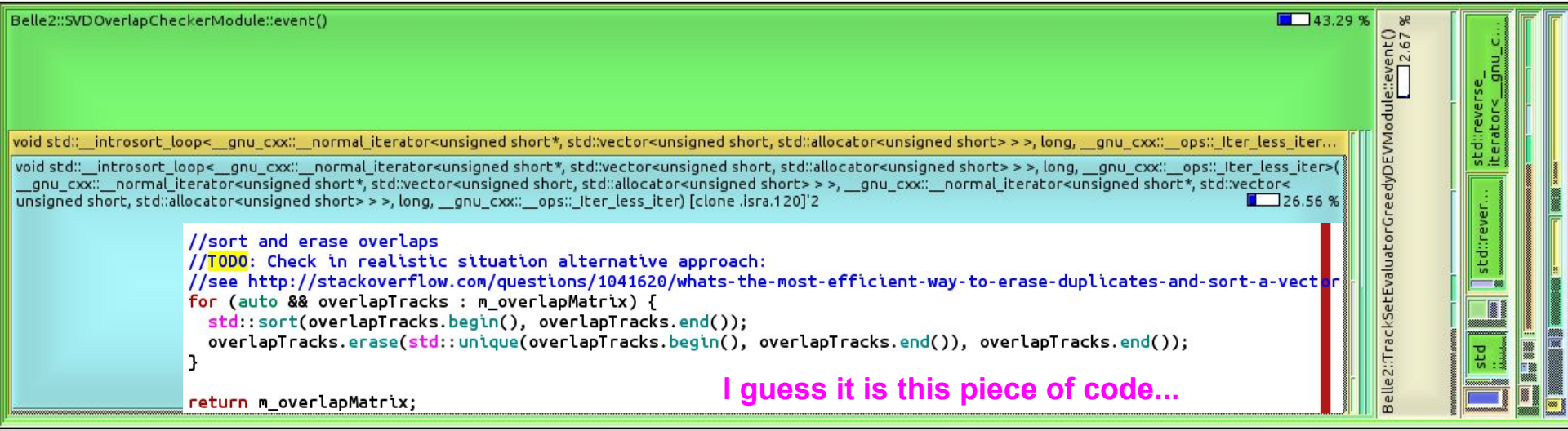
Now using the fat SecMap again

```
basf2 -i MyRootFile.root testVXDTFRelatedModules.py
```

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	3	0	0.01	1.94 +- 0.20
EventInfoPrinter	3	0	0.00	0.00 +- 0.00
Gearbox	3	0	0.00	0.00 +- 0.00
Geometry	3	0	0.00	0.00 +- 0.00
EventCounter	3	0	0.00	0.19 +- 0.07
SectorMapBootstrap	3	0	0.00	0.00 +- 0.00
SpacePointCreatorSVD	3	0	0.00	0.97 +- 0.08
SpacePointCreatorPXD	3	0	0.00	0.20 +- 0.03
SpacePoint2TrueHitConnector	3	0	0.01	1.81 +- 0.18
GFTC2SPTCConverter	3	0	0.00	1.45 +- 0.27
SPTCReferee	3	0	0.00	0.25 +- 0.09
SpacePoint2TrueHitConnector	3	0	0.01	4.69 +- 0.35
RawSecMapMerger	3	0	0.00	0.00 +- 0.00
SegmentNetworkProducer	3	0	0.28	92.83 +- 53.59
TrackFinderVXDCellOMat	3	0	0.12	40.29 +- 16.80
SPTCvirtualIPRemover	3	0	0.00	0.55 +- 0.24
QualityEstimatorVXDRandom	3	0	0.00	0.07 +- 0.02
SPTCNetworkProducer	3	0	47.30	15765.77 +-15135.49
TrackSetEvaluatorGreedy	3	0	93.29	31095.64 +-30119.97
SVDOverlapChecker	3	0	3.36	1120.39 +- 839.42
TrackSetEvaluatorGreedyDEV	3	0	0.10	32.29 +- 21.52
TrackFinderVXDAnalizer	3	0	0.01	2.61 +- 0.42
Total	3	0	144.53	48175.11 +-46135.34

OK, now we are back, that the OverlapChecker is the “problem”. Or not, as this amount of TrackCandidates is rare, so we know, the redesign is fast enogh!

```
isActive, Array Index: 11, QI: 0.825849
isActive, Array Index: 89, QI: 0.988963
isActive, Array Index: 2026, QI: 0.996657
isActive, Array Index: 4389, QI: 0.997194
isActive, Array Index: 4851, QI: 0.999532
isActive, Array Index: 6139, QI: 0.99964
isActive, Array Index: 6256, QI: 0.999548
isActive, Array Index: 6308, QI: 0.773476
isActive, Array Index: 7341, QI: 0.958889
isActive, Array Index: 7389, QI: 0.995538
isActive, Array Index: 7447, QI: 0.999933
isActive, Array Index: 7499, QI: 0.473606
Array Index: 7447, QI: 0.999933
Array Index: 6139, QI: 0.99964
Array Index: 6256, QI: 0.999548
Array Index: 4851, QI: 0.999532
Array Index: 4389, QI: 0.997194
Array Index: 2026, QI: 0.996657
Array Index: 7389, QI: 0.995538
Array Index: 89, QI: 0.988963
Array Index: 7341, QI: 0.958889
Array Index: 11, QI: 0.825849
Array Index: 6308, QI: 0.773476
Array Index: 7499, QI: 0.473606
```



Ir	Ir per call	Count	Callee
49.51	1 379 986 138	78	Belle2::EventProcessor::callEvent(Belle2::Module*) (libframework.so)
0.00	473 137	1	Belle2::EventProcessor::processBeginRun(bool) (libframework.so)
0.00	9 884	4	Belle2::RandomNumbers::initializeEvent() (libframework.so)
0.00	38 442	1	__cxa_throw (libstdc++.so.6.0.21)
0.00	306	78	Belle2::PathIterator::descendIfNecessary() (libframework.so)
0.00	196	79	__dynamic_cast (libstdc++.so.6.0.21)
0.00	1 378	10	_dl_runtime_resolve <cycle 1> (ld-2.19.so: dl-trampoline.S)
0.00	7 228	1	Belle2::EventProcessor::StoppedBySignalException::StoppedBySignalException(int) (libframework.so)
0.00	1 737	2	Belle2::ProcessStatistics::setCounters(double&, double&, double, double) (libframework.so)
0.00	21	78	Belle2::Module::evalCondition() const (libframework.so)
0.00	11	56	Belle2::EventMetaData::operator==(Belle2::EventMetaData const&) const (libframework.so)
0.00	5	79	Belle2::EventMetaData::isEndOfData() const (libframework.so)
0.00	50	4	Belle2::DBStore::updateEvent() (libframework.so)
0.00	129	1	__cxa_allocate_exception (libstdc++.so.6.0.21)
0.00	16	4	std::string::assign(std::string const&) (libstdc++.so.6.0.21)
0.00	12	4	TObject::operator=(TObject const&) (libCore.so)
0.00	5	4	Belle2::RandomNumbers::useEventDependent() (libframework.so)
0.00	4	4	Belle2::Environment::Instance() (libframework.so)
0.00	4	4	Belle2::DBStore::Instance() (libframework.so)
0.00	3	1	Belle2::EventProcessor::processEndRun() (libframework.so)

Alternative Conclusion

- Pausing work for that long on a badly documented complicated piece of code, is not a great idea.
- Speed is probably a non-issue of the VXD Track finder.
  - → Focus on efficiency.
  - Look into realistic situation, if then SVDOverlapChecker is still the problem, we can further think about not sorting...

BackUp

```
basf2 -i MyRootFile.root -n 1000 testVXDTFRelatedModules.py
basf2 -i MyRootFile.root -n 10 testVXDTFRelatedModules.py
```

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	10	0	0.02	2.00 +- 0.42
EventInfoPrinter	10	0	0.00	0.00 +- 0.00
Gearbox	10	0	0.00	0.00 +- 0.00
Geometry	10	0	0.00	0.00 +- 0.00
EventCounter	10	0	0.00	0.14 +- 0.04
SectorMapBootstrap	10	0	0.00	0.00 +- 0.00
SpacePointCreatorSVD	10	0	0.01	1.05 +- 0.14
SpacePointCreatorPXD	10	0	0.00	0.19 +- 0.03
SpacePoint2TrueHitConnector	10	0	0.02	1.77 +- 0.17
GFTC2SPTCConverter	10	0	0.01	1.36 +- 0.18
SPTCReferee	10	0	0.00	0.21 +- 0.05
SpacePoint2TrueHitConnector	10	0	0.05	5.02 +- 0.87
RawSecMapMerger	10	0	0.00	0.00 +- 0.00
SegmentNetworkProducer	10	0	21.26	2125.60 +- 678.76
TrackFinderVXDCellOMat	10	0	0.07	7.40 +- 5.52
SPTCvirtualIPRemover	10	0	0.00	0.07 +- 0.06
QualityEstimatorVXDRandom	10	0	0.00	0.02 +- 0.01
SPTCNetworkProducer	10	0	2.20	220.46 +- 449.80
TrackSetEvaluatorGreedy	10	0	3.74	374.24 +- 785.13
SVDOverlapChecker	10	0	0.46	45.59 +- 77.04
TrackSetEvaluatorGreedyDEV	10	0	0.01	1.42 +- 1.89
TrackFinderVXDAnalizer	10	0	0.02	1.96 +- 0.32
Total	10	0	27.90	2790.22 +-1541.76

With circle Fitter instead of Random

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	10	0	0.02	2.11 +- 0.56
EventInfoPrinter	10	0	0.00	0.00 +- 0.00
Gearbox	10	0	0.00	0.00 +- 0.00
Geometry	10	0	0.00	0.00 +- 0.00
EventCounter	10	0	0.00	0.14 +- 0.03
SectorMapBootstrap	10	0	0.00	0.00 +- 0.00
SpacePointCreatorSVD	10	0	0.01	1.07 +- 0.16
SpacePointCreatorPXD	10	0	0.00	0.19 +- 0.03
SpacePoint2TrueHitConnector	10	0	0.02	1.78 +- 0.20
GFTC2SPTCConverter	10	0	0.01	1.35 +- 0.18
SPTCReferee	10	0	0.00	0.20 +- 0.06
SpacePoint2TrueHitConnector	10	0	0.05	5.02 +- 0.85
RawSecMapMerger	10	0	0.00	0.00 +- 0.00
SegmentNetworkProducer	10	0	21.46	2146.11 +- 694.81
TrackFinderVXDCellOMat	10	0	0.07	7.40 +- 5.49
SPTCvirtualIPRemover	10	0	0.00	0.08 +- 0.09
QualityEstimatorVXDCircleFit	10	0	0.07	6.70 +- 14.67
SPTCNetworkProducer	10	0	2.18	217.89 +- 442.65
TrackSetEvaluatorGreedy	10	0	3.78	377.67 +- 795.30
SVDOverlapChecker	10	0	0.45	45.41 +- 77.10
TrackSetEvaluatorGreedyDEV	10	0	0.01	1.38 +- 1.85
TrackFinderVXDAnalizer	10	0	0.02	2.02 +- 0.24
Total	10	0	28.18	2818.15 +-1543.33

We should spend more time on the quality estimation!