

Profiling ACTS

Motivation

- See where the CPU is spending most of its time
- Understand the project's performance characteristics better
- Identify areas that take up more resources than expected
- Make it easier to see why this is the case
- Make performance optimisation quicker and less painful (hopefully)

gperftools

- [gperftools](#) is a software profiling package
- Popular, has a lot of documentation and support threads online
- Consists of a:
 - Sampling profiler for CPU
 - Thread-caching malloc library
 - Memory leak detection tool
 - Memory allocation profiler
 - Profile visualisation tool

Cmake and Documentation

```
if(ACTS_ENABLE_CPU_PROFILING)
  message(STATUS "added lprofiler")
  target_link_libraries(
    ActsCore
    PUBLIC -lprofiler)

  target_link_options(
    ActsCore
    PUBLIC "LINKER:-no-as-needed")
endif()
```

[PR #1274](#)

- [Documentation](#) for profiling consists of:
 - Brief overview of profiling and gperftools
 - How to install
 - How to enable profiling when building ACTS
 - How to turn on profiling when running ACTS
 - Basics on using pprof

Results – Top 10 Functions

ActsExampleCKFTracksGeneric run over 5000 events, call tree starting at ActsExamples::TrackFindingAlgorithm::execute, Eigen filtered out

```
Showing nodes accounting for 35.04s, 3.68% of 951.69s total
```

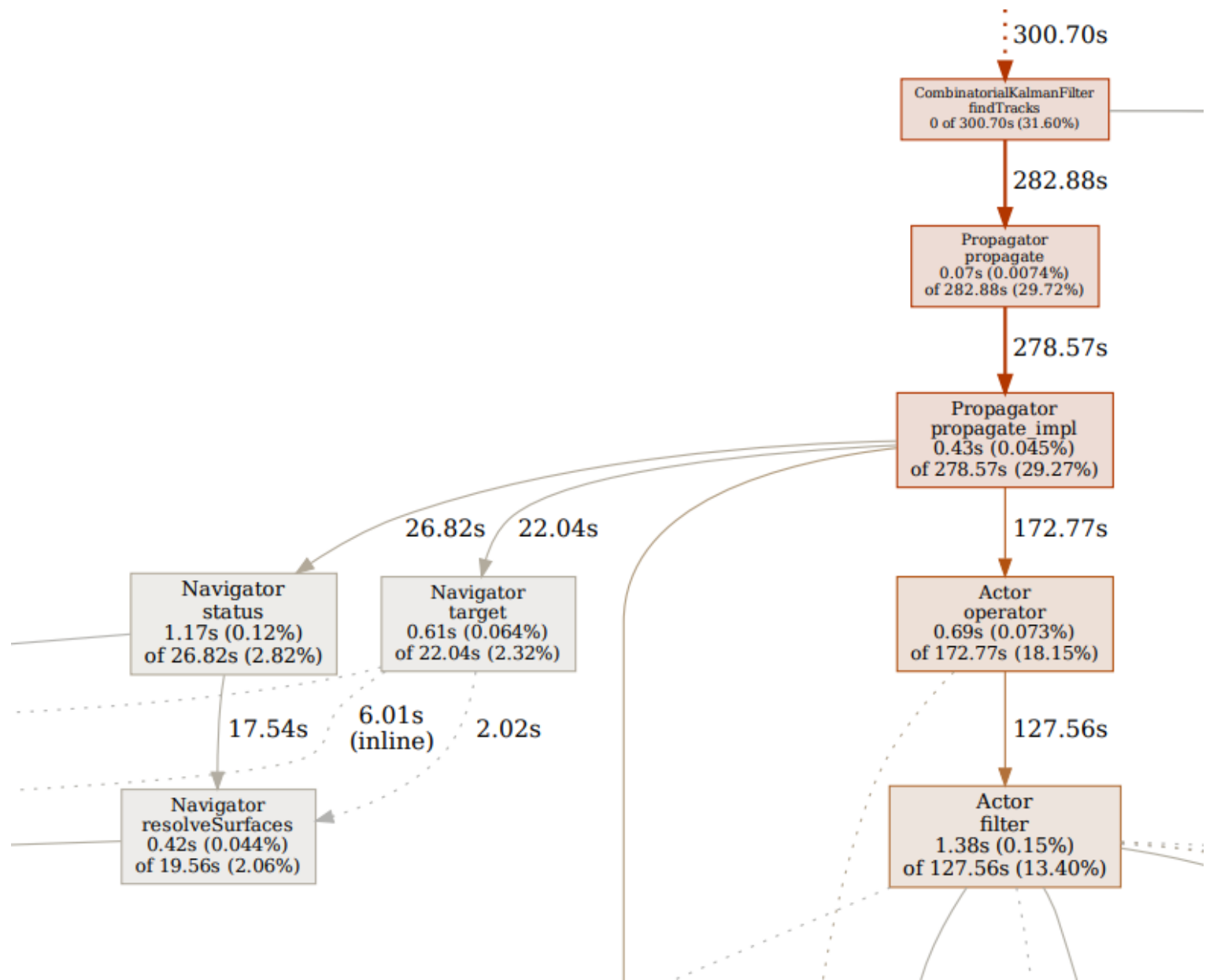
```
Dropped 483 nodes (cum <= 4.76s)
```

```
Showing top 15 nodes out of 204
```

flat	flat%	sum%	cum	cum%	
0.04s	0.0042%	0.0042%	316.06s	33.21%	ActsExamples::TrackFindingAlgorithm::execute
0	0%	0.0042%	300.70s	31.60%	(anonymous namespace)::TrackFinderFunctionImpl::operator()
0.03s	0.0032%	0.0074%	300.70s	31.60%	Acts::CombinatorialKalmanFilter::findTracks
0.62s	0.065%	0.073%	282.88s	29.72%	Acts::Propagator::propagate
26.80s	2.82%	2.89%	278.57s	29.27%	Acts::Propagator::propagate_impl (partial-inline)
0	0%	2.89%	172.92s	18.17%	Acts::ActionList::operator() (inline)
0	0%	2.89%	172.92s	18.17%	Acts::detail::action_list_impl::action (inline)
0.73s	0.077%	2.97%	172.77s	18.15%	Acts::CombinatorialKalmanFilter::Actor::operator()
1.61s	0.17%	3.13%	127.56s	13.40%	Acts::CombinatorialKalmanFilter::Actor::filter
0	0%	3.13%	75.56s	7.94%	std::__invoke (inline)
0.26s	0.027%	3.16%	75.56s	7.94%	std::__invoke_impl (inline)
0.03s	0.0032%	3.16%	66.78s	7.02%	Acts::Delegate::operator() (inline)
0	0%	3.16%	66.75s	7.01%	std::invoke (inline)
4.51s	0.47%	3.64%	48.69s	5.12%	Acts::detail::transportCovarianceToBound
0.41s	0.043%	3.68%	40.76s	4.28%	Acts::MultiTrajectory::addTrackState

Results – Function Line-by-line

```
Total: 951.69s
ROUTINE ===== Acts::detail::transportCovarianceToBound in /acts/Core/src/Propagator/detail/CovarianceEngine.cpp
  4.51s      48.69s (flat, cum)  5.12% of Total
    .        .      332:
    .        .      333: void transportCovarianceToBound(
    .        .      334:     const GeometryContext& geoContext, BoundSymMatrix& boundCovariance,
    .        .      335:     BoundMatrix& fullTransportJacobian, FreeMatrix& freeTransportJacobian,
    .        .      336:     FreeVector& freeToPathDerivatives, BoundToFreeMatrix& boundToFreeJacobian,
  270ms     270ms  337:     const FreeVector& freeParameters, const Surface& surface) {
    .        .      338:     // Calculate the full jacobian from local parameters at the start surface to
    .        .      339:     // current bound parameters
  40ms     28.42s  340:     boundToBoundJacobian(geoContext, freeParameters, boundToFreeJacobian,
    .        .      341:                          freeTransportJacobian, freeToPathDerivatives,
    .        .      342:                          fullTransportJacobian, surface);
    .        .      343:
    .        .      344:     // Apply the actual covariance transport to get covariance of the current
    .        .      345:     // bound parameters
    .        .      346:     boundCovariance = fullTransportJacobian * boundCovariance *
  4.17s     4.47s  347:                          fullTransportJacobian.transpose();
    .        .      348:
    .        .      349:     // Reinitialize jacobian components:
    .        .      350:     // ->The transportJacobian is reinitialized to Identity
    .        .      351:     // ->The derivatives is reinitialized to Zero
    .        .      352:     // ->The boundToFreeJacobian is initialized to that at the current surface
    .        15.50s  353:     reinitializeJacobians(geoContext, freeTransportJacobian,
    .        .      354:                          freeToPathDerivatives, boundToFreeJacobian,
    .        .      355:                          freeParameters, surface);
  30ms     30ms  356: }
    .        .      357:
    .        .      358: void transportCovarianceToCurvilinear(BoundSymMatrix& boundCovariance,
    .        .      359:     BoundMatrix& fullTransportJacobian,
    .        .      360:     FreeMatrix& freeTransportJacobian,
    .        .      361:     FreeVector& freeToPathDerivatives,
```

PR Feedback I'm Working On

- Allow specifying gperftools libraries location when building ACTS so elevated privileges are not required to install gperftools

Thanks