

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()
```

- 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

PR #1274

Results – Top 10 Functions

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

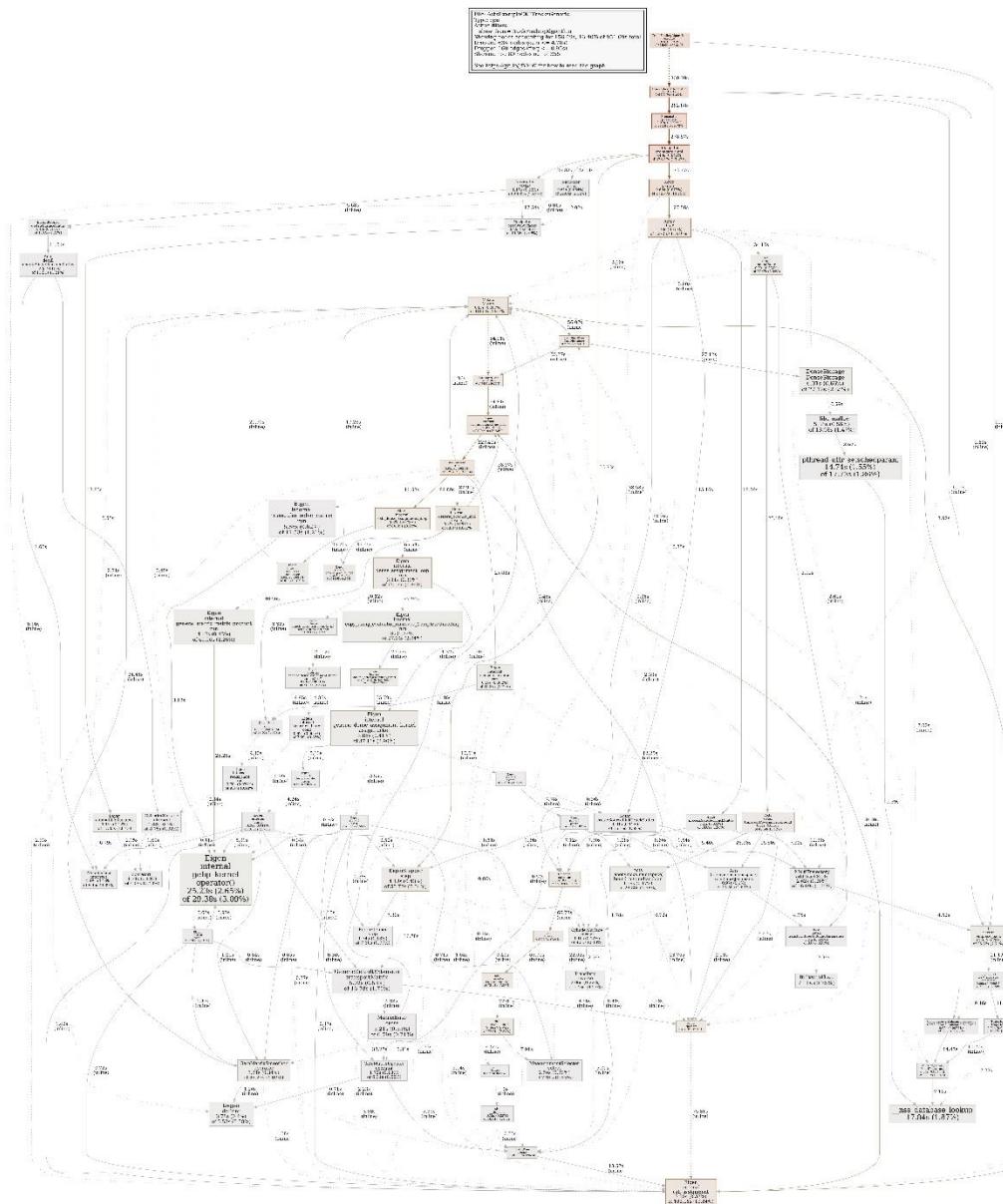
```
Showing nodes accounting for 35.04s, 3.68% of 951.69s total
Dropped 483 nodes (cum <= 4.76s)
Showing top 15 nodes out of 204

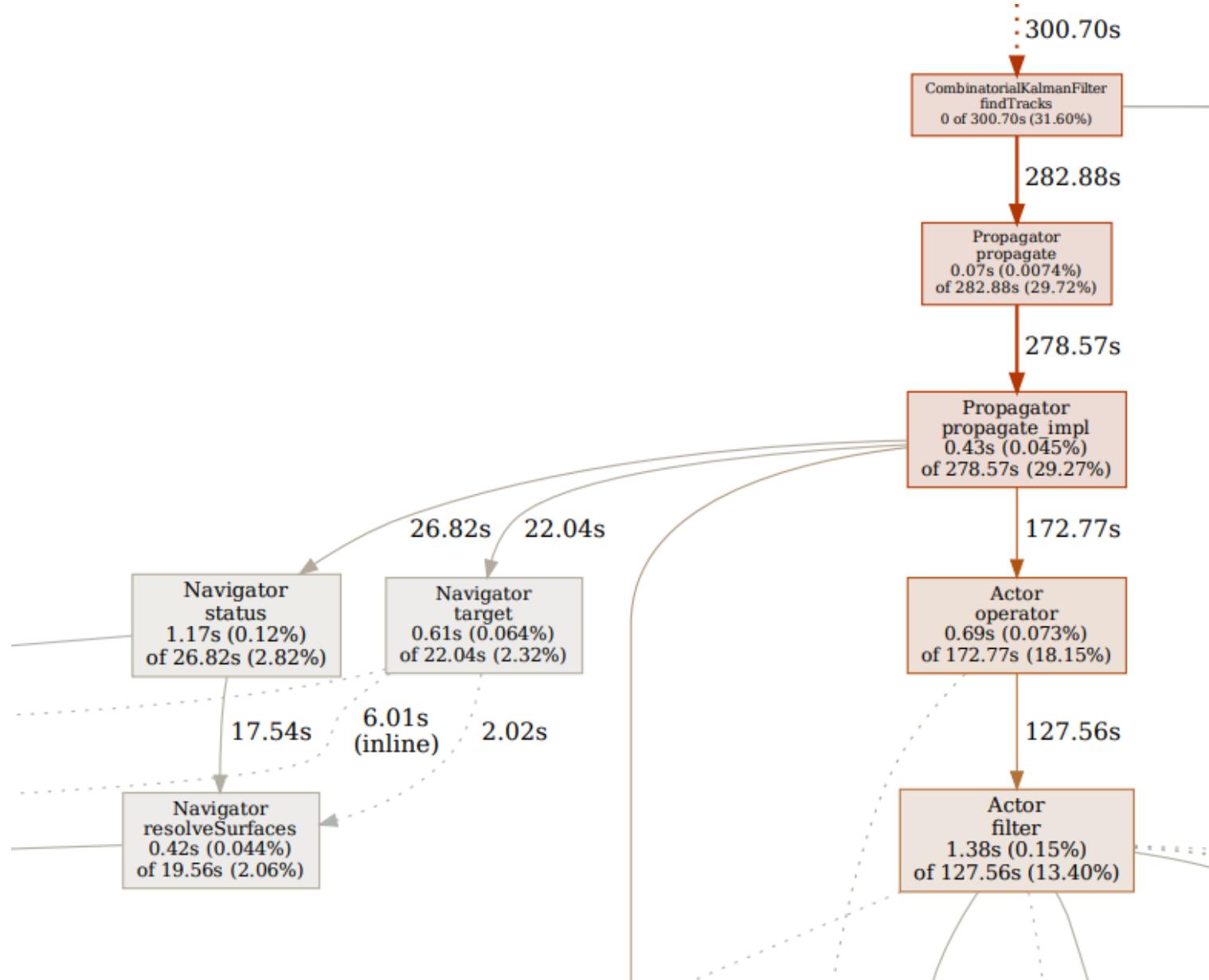
      flat   flat%   sum%      cum   cum%
0.04s 0.0042% 0.0042% 316.06s 33.21% ActsExamples::TrackFindingAlgorithm::execute
      0     0% 0.0042% 300.70s 31.60% (anonymous namespace)::TrackFinderFunctionImpl::operator()
0.03s 0.0032% 0.0074% 300.70s 31.60% Acts::CombinatorialKalmanFilter::findTracks
0.62s 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,	





Flat(s)	Flat(%)	Sum(%)	Cum(s)	Cum(%)	Function
0	0	0	303.71s	31.91	ActsExamples::TrackFindingAlgorithm::execute
0	0	0	300.7	31.6	(anonymous namespace)::TrackFinderFunctionImpl::operator()
0	0	0	284.07	29.85	Acts::CombinatorialKalmanFilter::findTracks
0	0	0	278.67	29.28	Acts::Propagator::propagate
0	0	0	172.92	18.17	Acts::ActionList::operator()
0	0	0	172.92	18.17	Acts::detail::action_list_impl::action
0.05	0.0053	0.0053	172.82	18.16	Acts::Propagator::propagate_Impl
0	0	0.0053	157.16	16.51	Acts::Propagator::propagate_Impl
0.07	0.0074	0.013	127.63	13.41	Acts::CombinatorialKalmanFilter::Actor::operator()
0.03	0.0032	0.034	66.76	7.01	Acts::Delegate::operator()
26.15	2.75	2.79	54.03	5.68	Acts::Propagator::propagate_Impl
0	0	2.79	39.76	4.18	Acts::CombinatorialKalmanFilter::Actor::operator()
0.01	0.0011	2.79	38.59	4.05	Acts::CombinatorialKalmanFilter::Actor::finalize
0	0	2.79	38.57	4.05	Acts::Delegate::connect(Acts::GainMatrixSmoothen const*):{lambda(void const*, Acts::ContextType const&, Acts::MultiTrajectory&, unsigned long, Acts::LoggerWrapper)#1}::_FUN
0	0	2.79	38.57	4.05	Acts::Delegate::connect(Acts::GainMatrixSmoothen const*):{lambda(void const*, Acts::ContextType const&, Acts::MultiTrajectory&, unsigned long, Acts::LoggerWrapper)#1}:operator()
0	0	2.79	38.31	4.03	Acts::GainMatrixSmoothen::operator()
0	0	2.79	38.27	4.02	Acts::GainMatrixSmoothen::operator()
0.15	0.016	2.81	30.91	3.25	Acts::CombinatorialKalmanFilter::Actor::filter
0.04	0.0042	2.81	28.42	2.99	Acts::detail::transportCovarianceToBound
0.06	0.0063	2.82	28.24	2.97	Acts::detail::boundState
0.04	0.0042	2.82	26.59	2.79	Acts::Propagator::propagate_Impl
0.02	0.0021	2.82	25.17	2.64	Acts::MultiTrajectory::MultiTrajectory
20.51	2.16	4.98	24.33	2.56	Acts::GainMatrixSmoothen::operator() (Acts::ContextType const&, Acts::MultiTrajectory&, unsigned long, Acts::LoggerWrapper) const:{lambda(auto:1)#1}:operator()
20.44	2.15	7.13	23.74	2.49	Acts::(anonymous namespace)::boundToBoundJacobian
6.92	0.73	7.85	21.11	2.22	Acts::detail_It::GrowableColumns::GrowableColumns
0.04	0.0042	7.86	20.55	2.16	Acts::CombinatorialKalmanFilter::Actor::filter
0.28	0.029	7.89	20.31	2.13	Acts::Propagator::propagate_Impl
0.08	0.0084	7.9	18.97	1.99	Acts::CombinatorialKalmanFilter::Actor::filter
0.14	0.015	7.91	18.89	1.98	Acts::Navigator::resolveSurfaces
0.03	0.0032	7.91	18.2	1.91	Acts::StepperExtensionList::finalize
0.03	0.0032	7.92	17.57	1.85	Acts::Navigator::status
0.08	0.0084	7.92	16.86	1.77	Acts::detail::GenericDefaultExtension::finalize
0.09	0.0095	7.93	16.02	1.68	Acts::CombinatorialKalmanFilterResult::CombinatorialKalmanFilterResult
0	0	7.93	15.74	1.65	Acts::Propagator::propagate_Impl
0	0	7.93	15.5	1.63	Acts::detail::transportCovarianceToBound
0.02	0.0021	7.94	15.18	1.6	Acts::CombinatorialKalmanFilter::Actor::filter
0	0	7.94	11.23	1.18	template_switch
0	0	7.94	11.23	1.18	visit_measurement
0	0	7.94	11.21	1.18	template_switch
0	0	7.94	11.19	1.18	ActsExamples::TrackFindingAlgorithm::execute
0	0	7.94	11.19	1.18	ActsExamples::Trajectories::Trajectories
					Flat = time spent at the location itself
					Cum = total time spent in location and all of its callees
					Sum = total time spent in program so far

PR Feedback I'm Working On

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

Thanks