6 YEARS OF OPTIMIZING WORLD OF TANKS: MAKING THE GAME A GREAT EXPERIENCE ON ALL SYSTEMS FROM LAPTOPS TO HIGH-END PCS

Denis Ishmukhametov, Wargaming
Bronislav Sviglo, Wargaming
Philipp Gerasimov, Intel

Legal Notices and Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel, Core and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

Agenda

- Introduction
- 2014 / 2015 – Optimizing for Ultrabook™
- 2016 – DX11 and Core Engine
- 2017 – World of Tanks 1.0
- 2018 – Concurrent Rendering
  - Concurrent Rendering
  - Tank Treads
  - Havok, AVX2
- 2019 – Ray Traced Shadows
- Summary
- Q&A

Introduction

- Philipp Gerasimov, Intel
  Senior Game / Graphics Application Engineer, Munich.

- Denis Ishmukhametov, Wargaming
  Rendering Engineer, World of Tanks, Minsk.

- Bronislav Sviglo, Wargaming
  Rendering Team Lead, World of Tanks, Minsk.

2014 2015 2016 2017 2018 2019

OPTIMIZING FOR ULTRABOOK™

World of Tanks

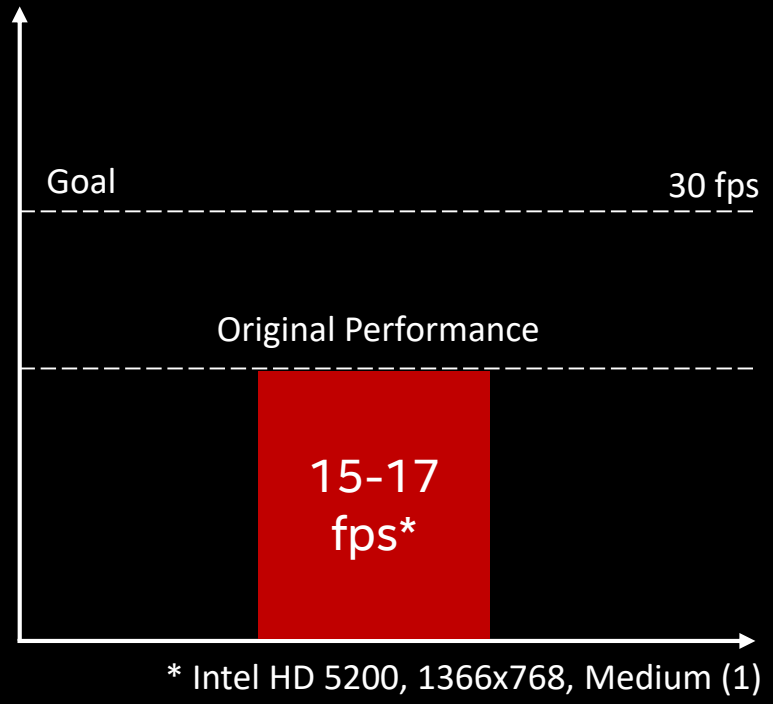
World of Tanks – online free-to-play game, developed by Wargaming

Quick Facts:

- Launched in 2010
- More than 160 millions of registered gamers
- 4 Golden Joysticks awards over the years
- Main development in Minsk, Belarus
- 15 vs. 15 players, 3rd / 1st person view tank battles

Optimizing for Ultrabook™ - Where we started?

World of Tanks has more than **160 million** registered gamers with wide range of HW – our first goal was enabling game even for mainstream PCs, including Ultrabook™

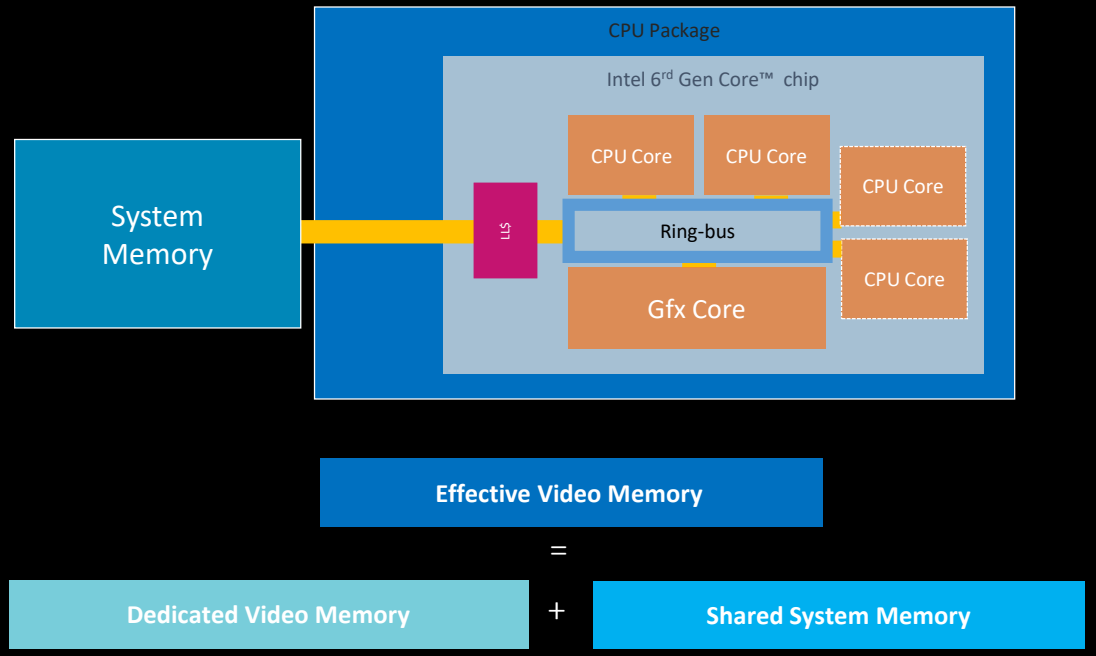


(1) For more information about performance and benchmark results, visit www.intel.com/benchmarks.

Optimizing for Ultrabook™ - Enabling Shadows

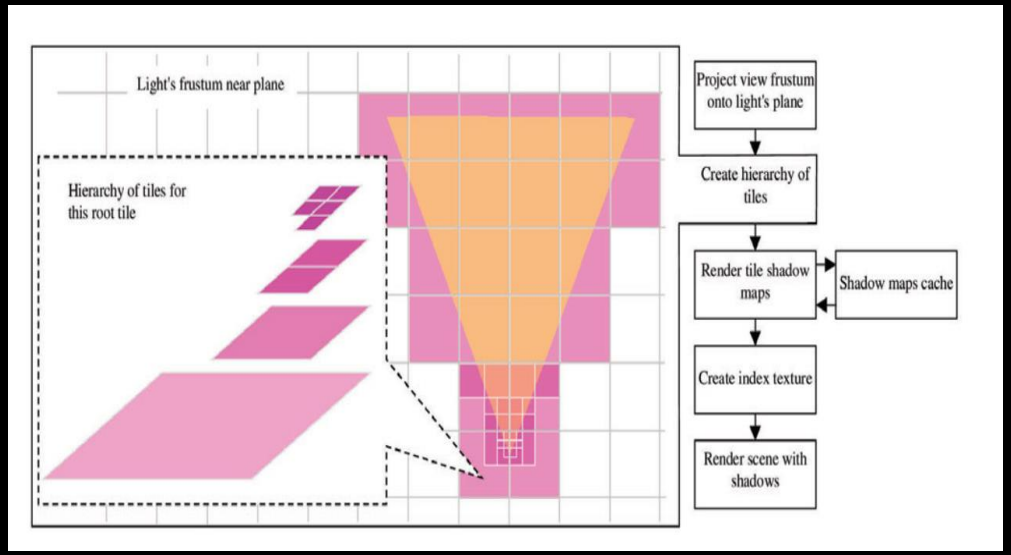
Enabling Shadows

- Correctly detecting Intel HW using *shared video memory*



Quality / Performance Optimizations

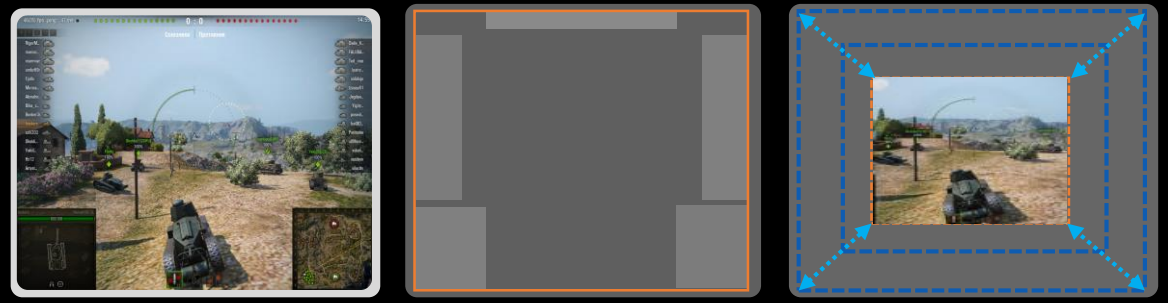
- **Adaptive Shadow Maps (ASM)** for static objects
- **Cascaded Shadow Maps(CSM)** for dynamic objects



Optimizing for Ultrabook™ - Optimizations

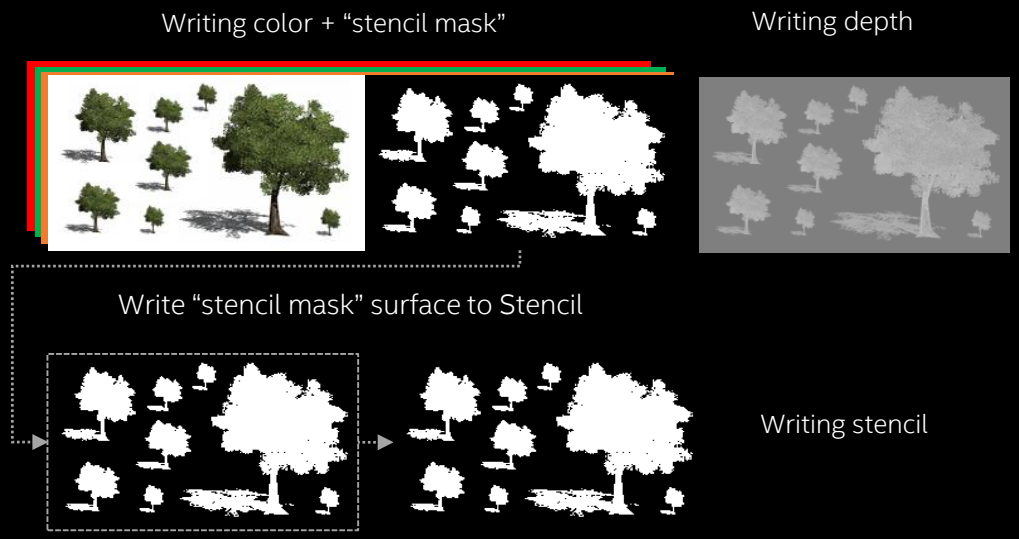
Dynamic Resolution

- Scene gets rendered with dynamic resolution and unscaled, depending on performance
- UI always rendered in full resolution
- One of the first implementation for PC games
- Allows stable 30 fps frame rate during gameplay



HW Specific Optimizations (2015)

- Two pass stencil write for vegetation rendering on Intel GPUs for optimal stencil + **clip()** performance

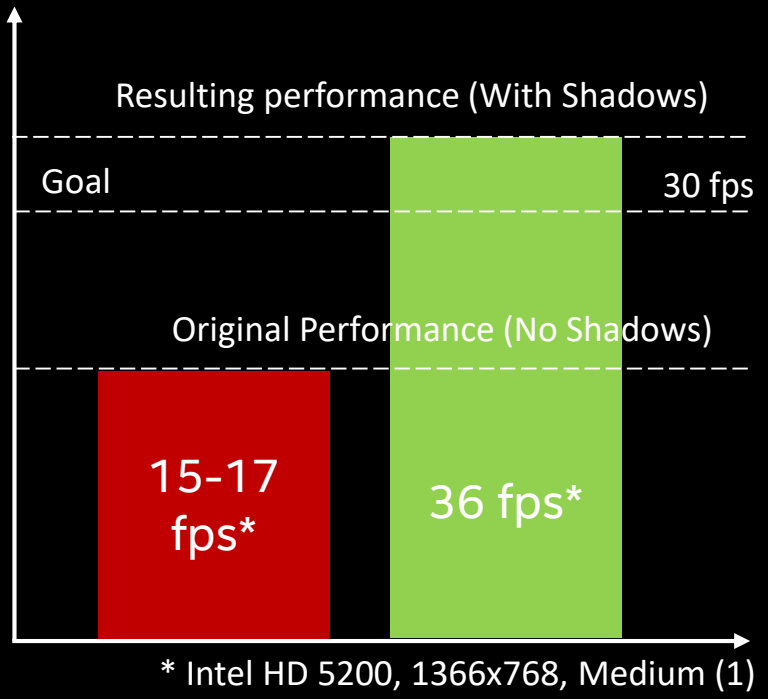


Optimizing for Ultrabook™ - Scaling Quality

* World of Tank Performance in 2019 – Skull Canyon @ 1080p

Optimizing for Ultrabook™ - Results

By end of 2015, together with World of Tanks engineering team we reached our performance goals and enabled missing rendering features



(1) For more information about performance and benchmark results, visit www.intel.com/benchmarks.


2014
2015
2016
2017
2018
2019
CORE
DX11 AND CORE ENGINE

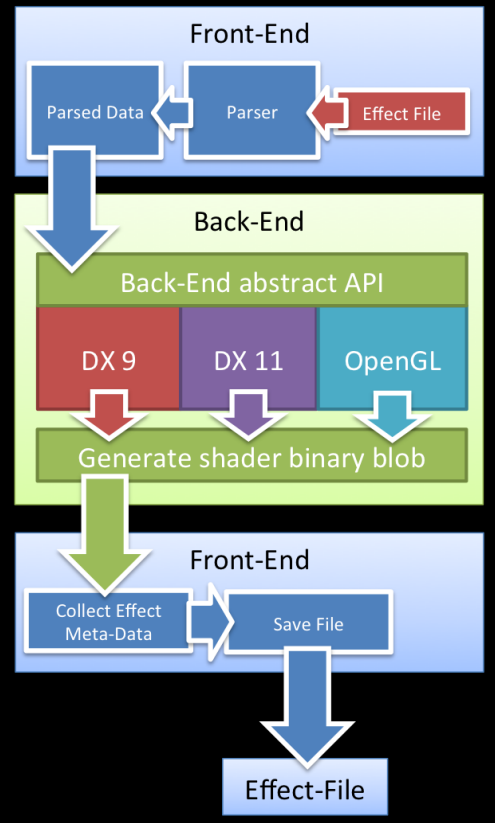
Porting to DX11

In 2016 the team started the big work on re-architecting the engine

- Add support of DirectX11
- Cleanup existing code base (about 15 years old)
- Prepare for the future graphics APIs & platforms
- Efficiently support existing platforms
  - Windows XP (D3D9)
  - Windows 7/8/10 (D3D11)

- Solution
  - Abstract rendering interface (ARI)
  - Wargaming Effect Framework (WGFX)

WGFX architecture



Core Engine 1.0

The first Core Engine was released in 2016 in World of Tanks 9.15 update

- Support wide range of hardware (DX9 + DX11)
- ARI + WGFX
- Usage of background render thread
- Temporal anti-aliasing
- KISS approach to improve performance
- Released in 2016 in World of Tanks 9.15 update

Up to 30% performance improvement!

2014 2015 2016 **2017** 2018 2019

WORLD OF TANKS 1.0

Core Engine 5.0

During 2017 this work was extended with new features, effects and performance optimizations

- Optimization is top priority
- Minimum system requirements preserved – 2004+ level hardware
- Next level graphics quality
  - Physically-based rendering
  - 32x32km Terrain with virtual texture
  - Adaptive shadow maps
  - Improved water rendering
  - Global illumination
  - Havok ® Destruction
  - Interactive vegetation

Internal **Core Engine 5.0** became known as just **Core** engine

World of Tanks 1.0

New engine allowed to deliver the biggest game update – World of Tanks 1.0

- Completely revamped version of World of Tanks
- Brand new **Core** engine
- All art assets created from scratch ~1200 3d models
- 29 maps reworked with new technology
- 1 new map Glacier + Brand new garage
- Each reworked map has x5 objects
- Added new graphics quality level - **Ultra**

And with all the changes performance remained the same for all the existing graphics quality levels!


CORE
Old vs. New


CORE
Old vs. New


CORE
Old vs. New

2014 2015 2016 2017 2018 2019

CONCURRENT RENDERING

Intel® Threading Building Blocks (TBB)

Now the team was ready for the next ambitious task – Make the engine ready for modern multi-core CPUs and the first problem was selecting a good job system. How to select a good job system?

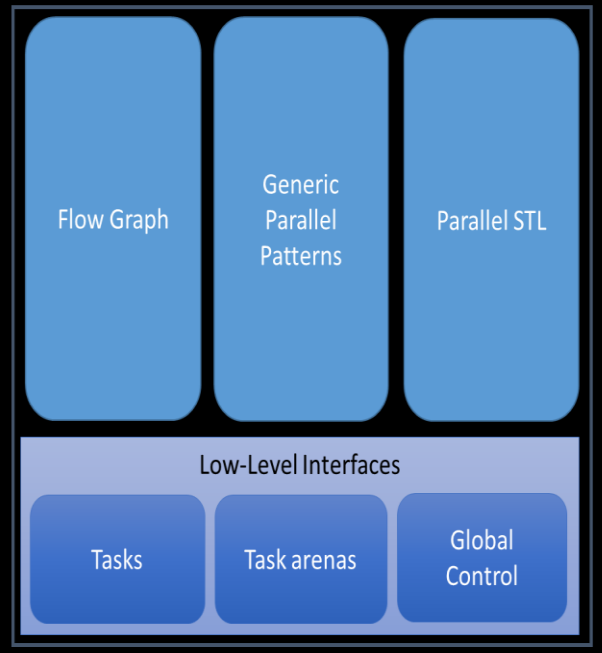
threadingbuildingblocks.org

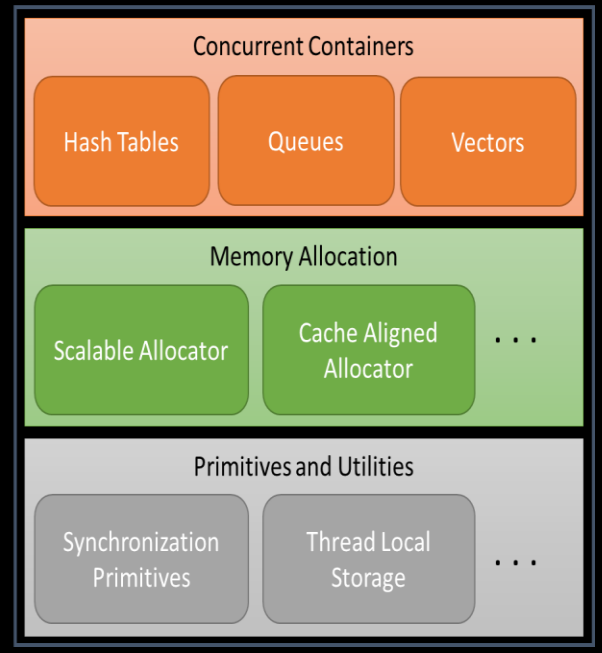
WoT Engineering team criteria:

- Easy to use
- Two types of parallelism: functional/task and data
- Feature rich and robust
- Good support

Threading Building Blocks

- Parallel algorithms and data structures, threads and synchronization
- Scalable memory allocation and task scheduling
- Is a library-only solution that does not depend on special compiler support
- Supports C++, Windows*, Linux*, OS X*, Android* and other OSes

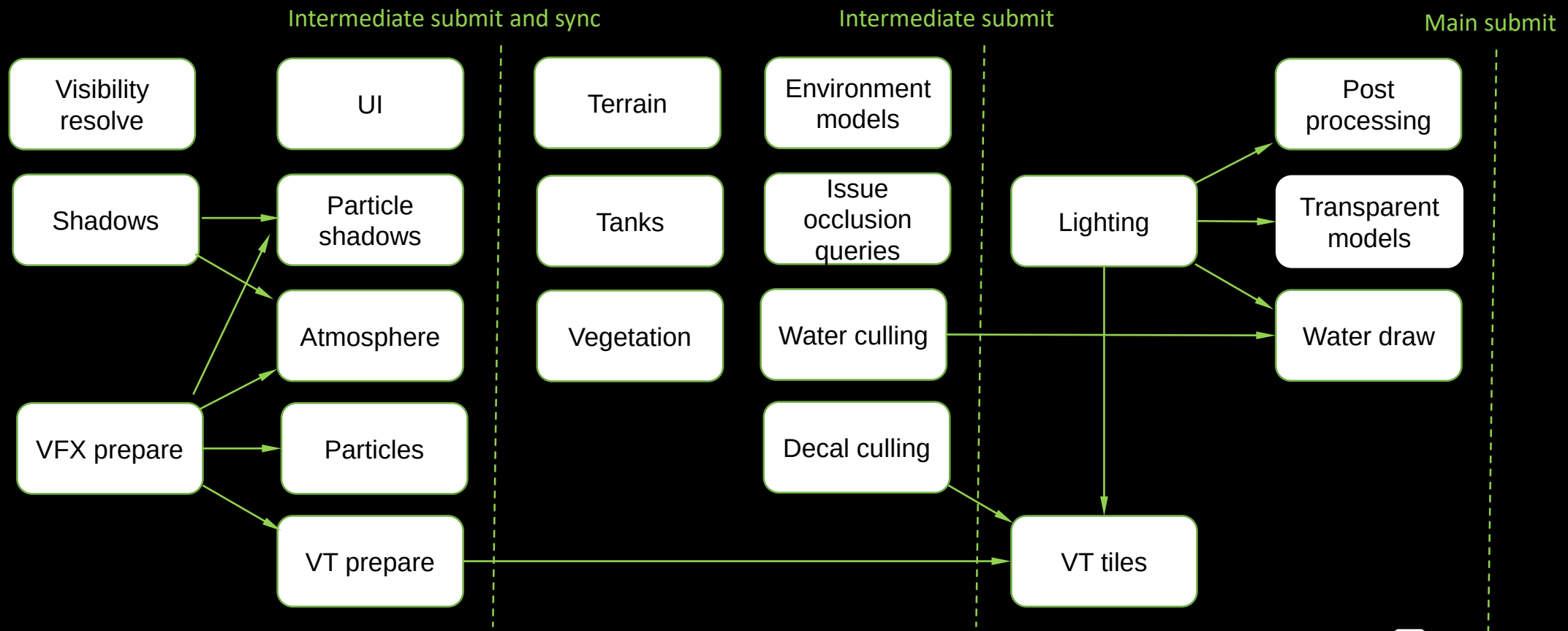




MULTI-THREADING IN WOT 1.0

Concurrent rendering

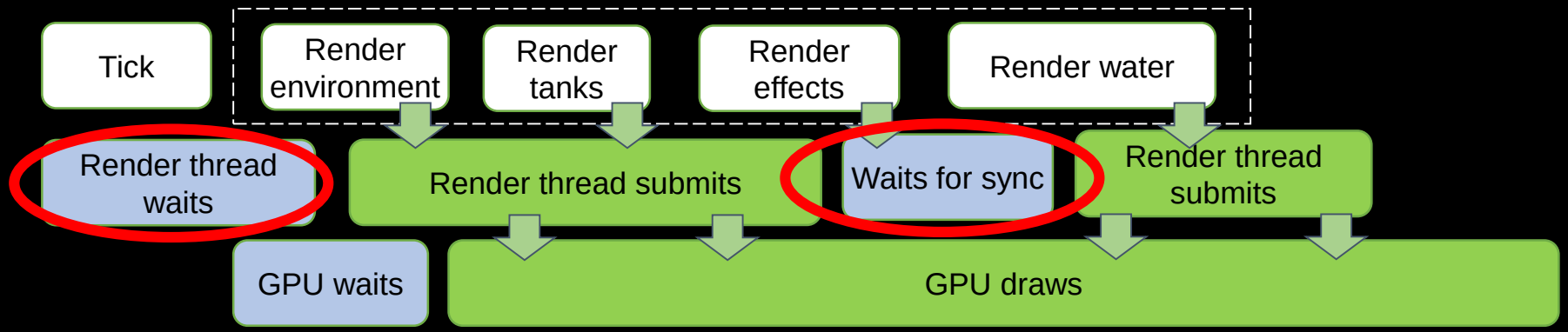
High-level frame render graph



Concurrent rendering

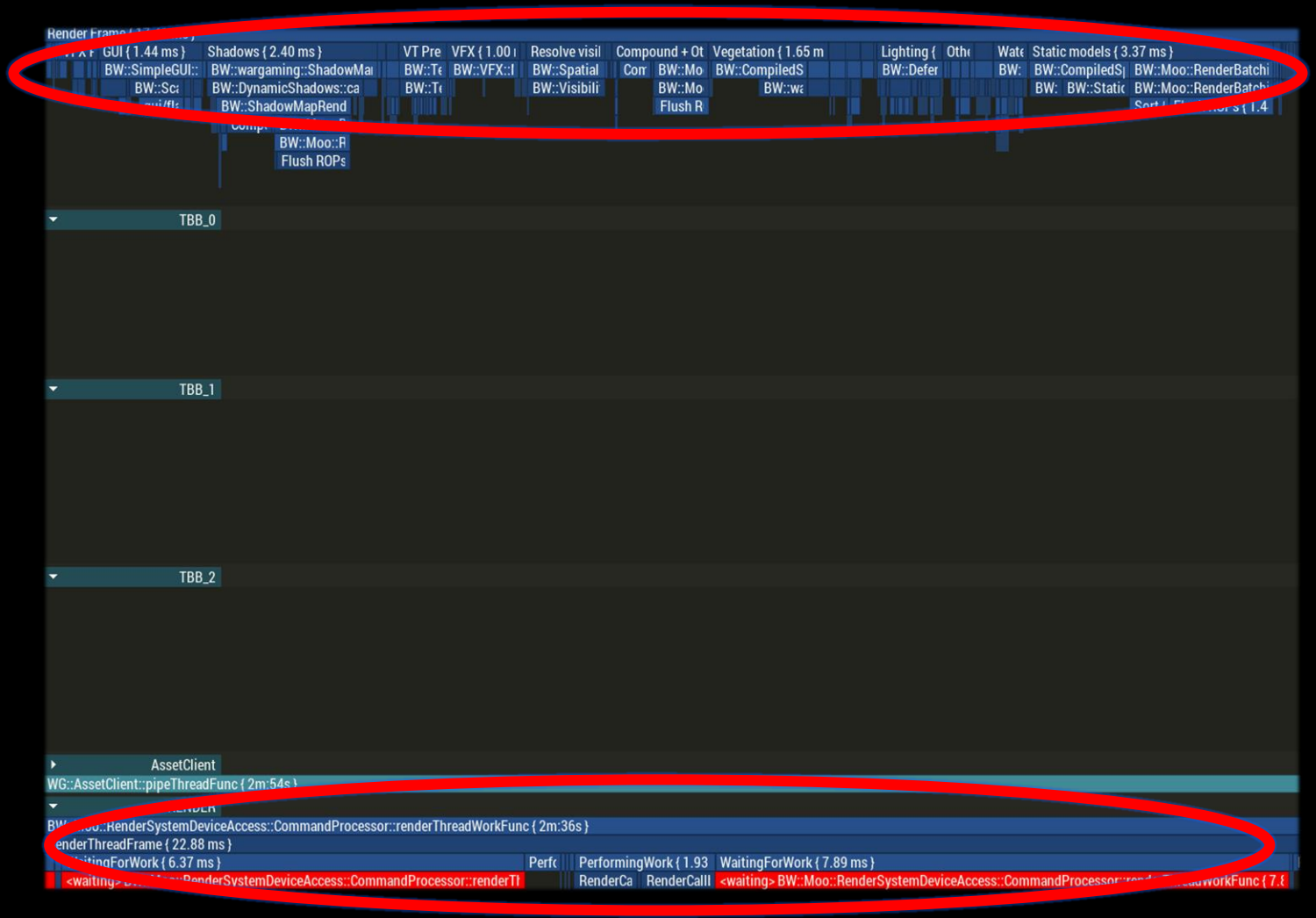
Submitting GPU workload

- Main per-frame contexts flush
  - Every path comes here
  - Uploads all gathered contexts to GPU submission thread
- Intermediate flush
  - Synchronization point for tasks that order-dependent on the GPU side
  - Prevents the GPU submission thread starvation



Concurrent rendering

Render frame ~17ms
Parallel execution off



Concurrent rendering

Render frame ~12ms

Parallel execution on

- Resolve visibility
- Static models
- Tanks
- Lighting
- Water
- Vegetation
- Post-processing

Concurrent rendering

Render frame ~8ms
Speedup ~2x
Parallel execution on

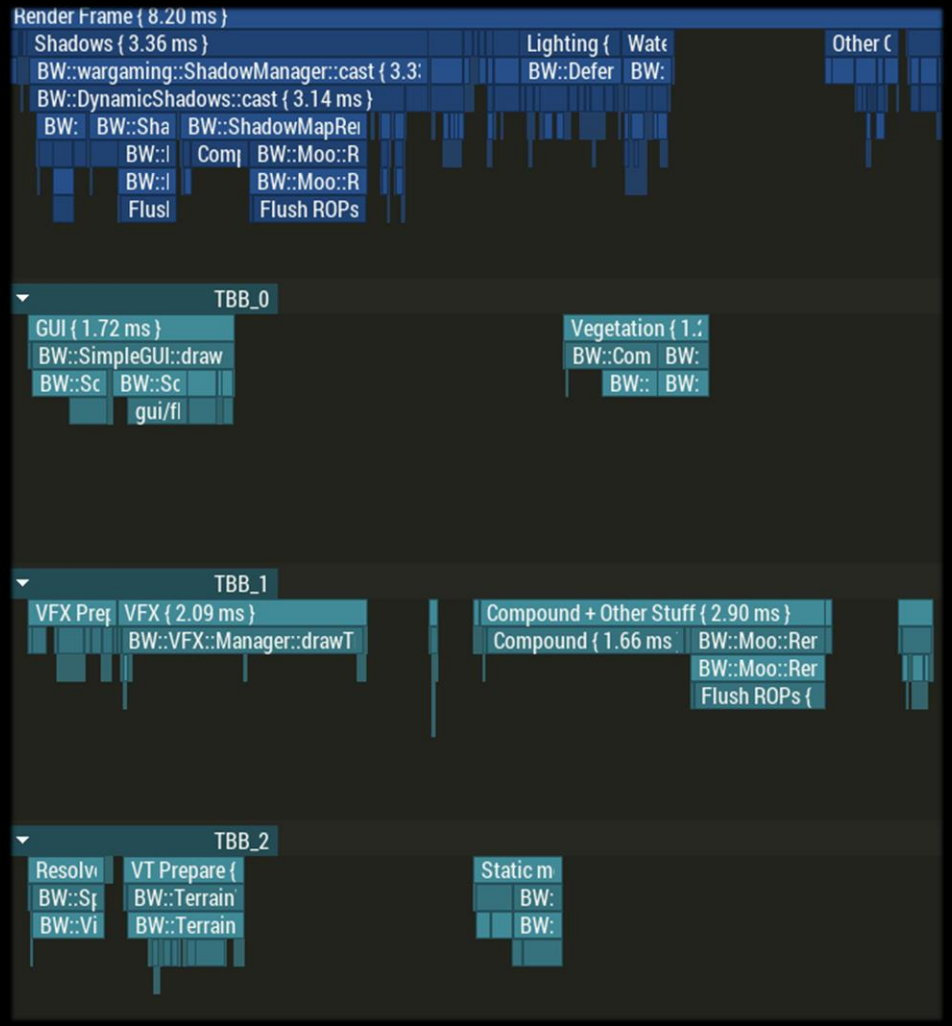
- Resolve visibility
- Static models
- Tanks
- Lighting
- Water
- Vegetation
- Post-processing
- Shadows

…

Concurrent rendering

Functional parallelism

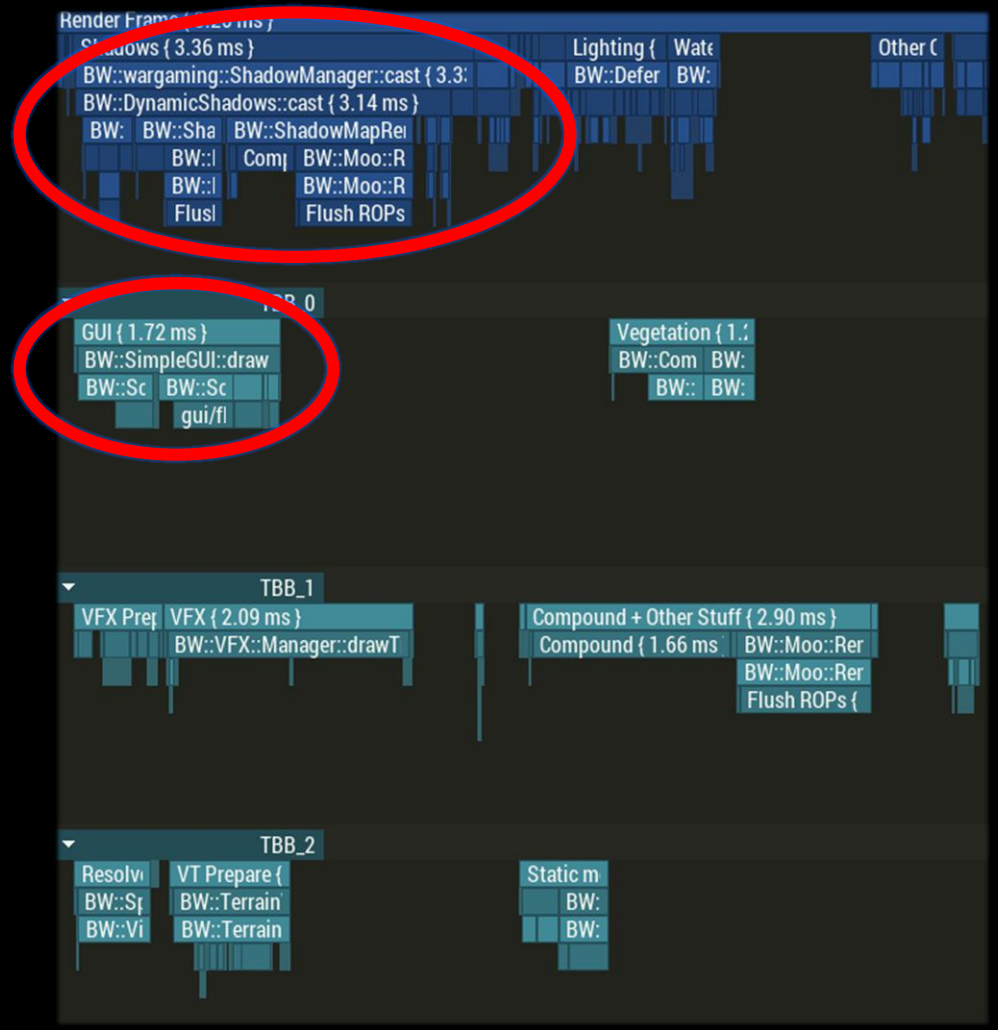
- Pros
  - Easy to implement
  - Easy to read and maintain
  - Easy to reason about



Concurrent rendering

Functional parallelism

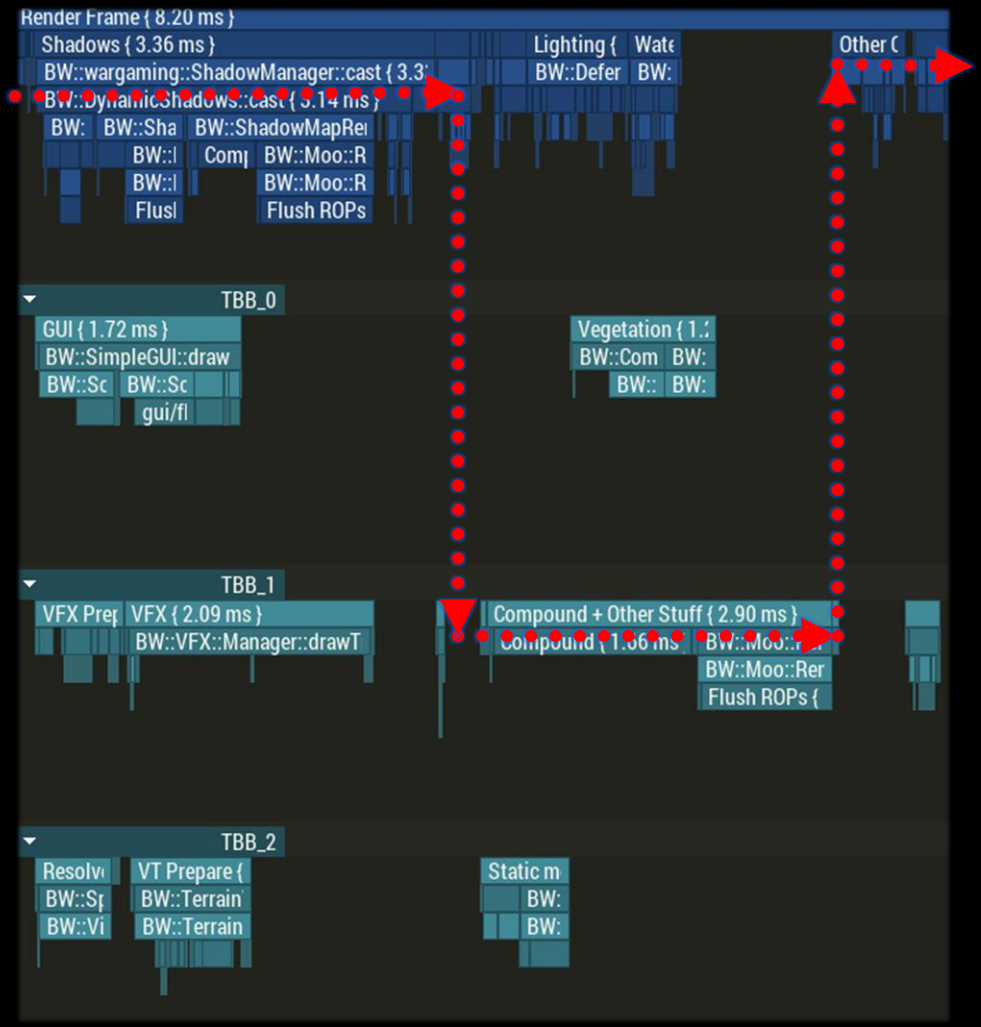
- Pros
  - Easy to implement
  - Easy to read and maintain
  - Easy to reason about
- Cons
  - Too high level
  - Some paths can't be shortened
  - Critical execution path



Concurrent rendering

Functional parallelism

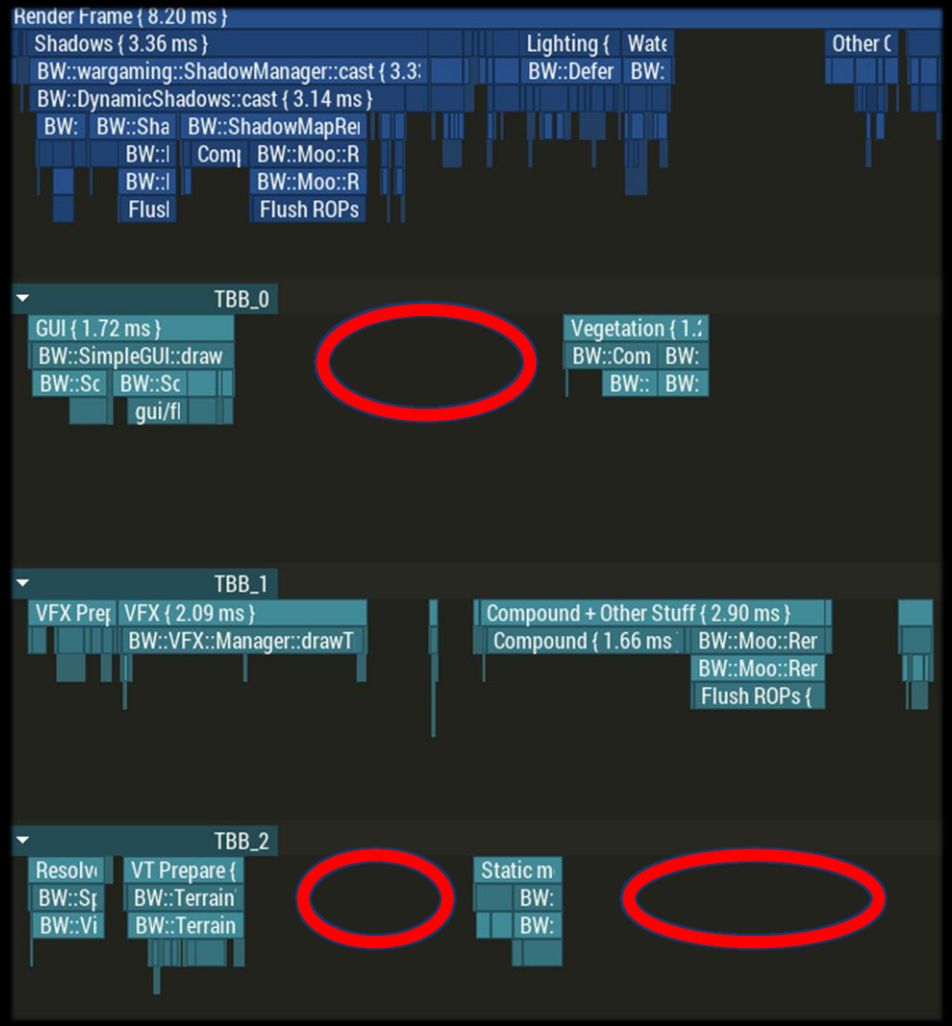
- Pros
  - Easy to implement
  - Easy to read and maintain
  - Easy to reason about
- Cons
  - Too high level
  - Some paths can’t be shortened
  - Critical execution path



Concurrent rendering

Functional parallelism

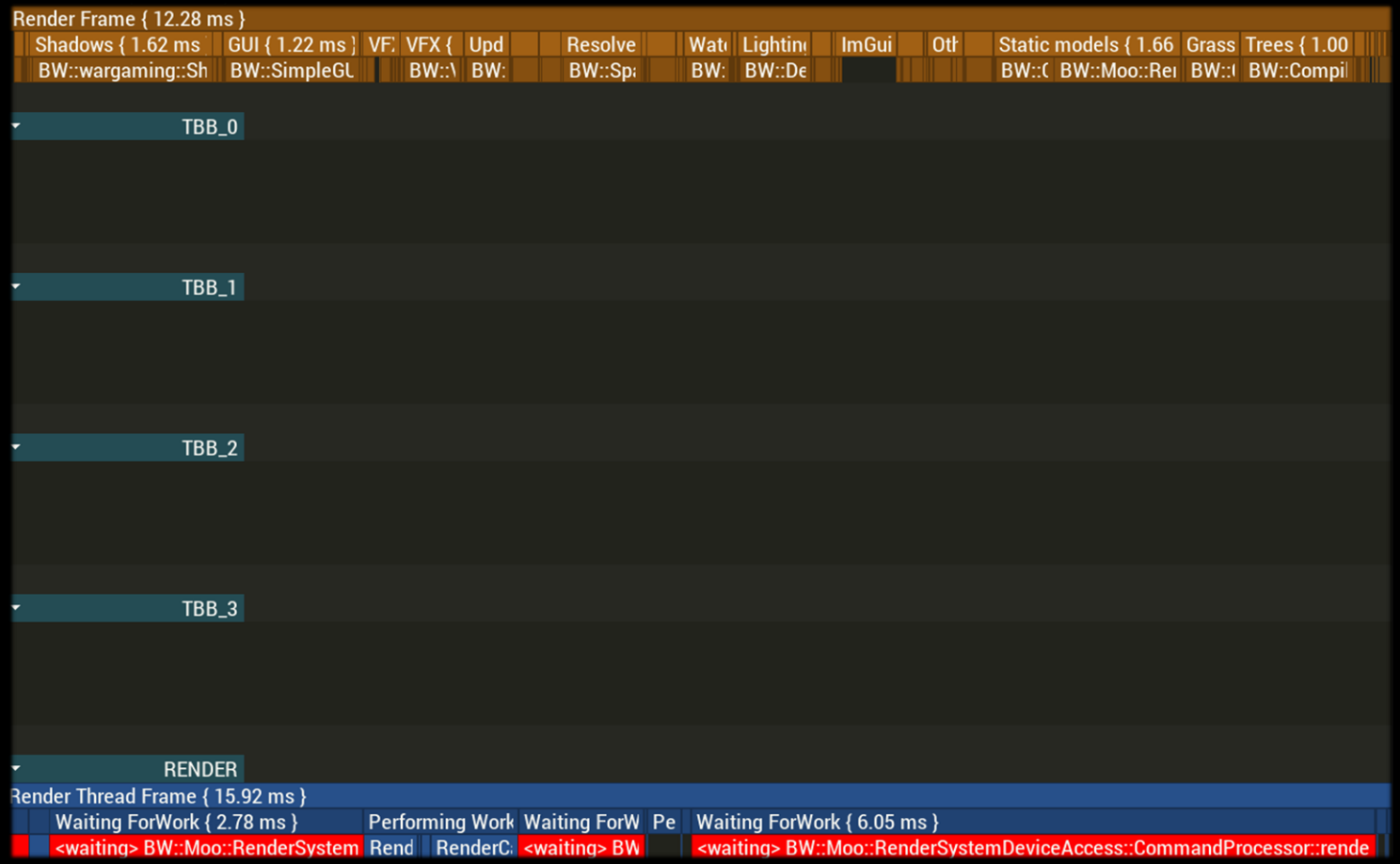
- Pros
  - Easy to implement
  - Easy to read and maintain
  - Easy to reason about
- Cons
  - Too high level
  - Some paths can't be shortened
  - Critical execution path



Data parallelism to rescue!

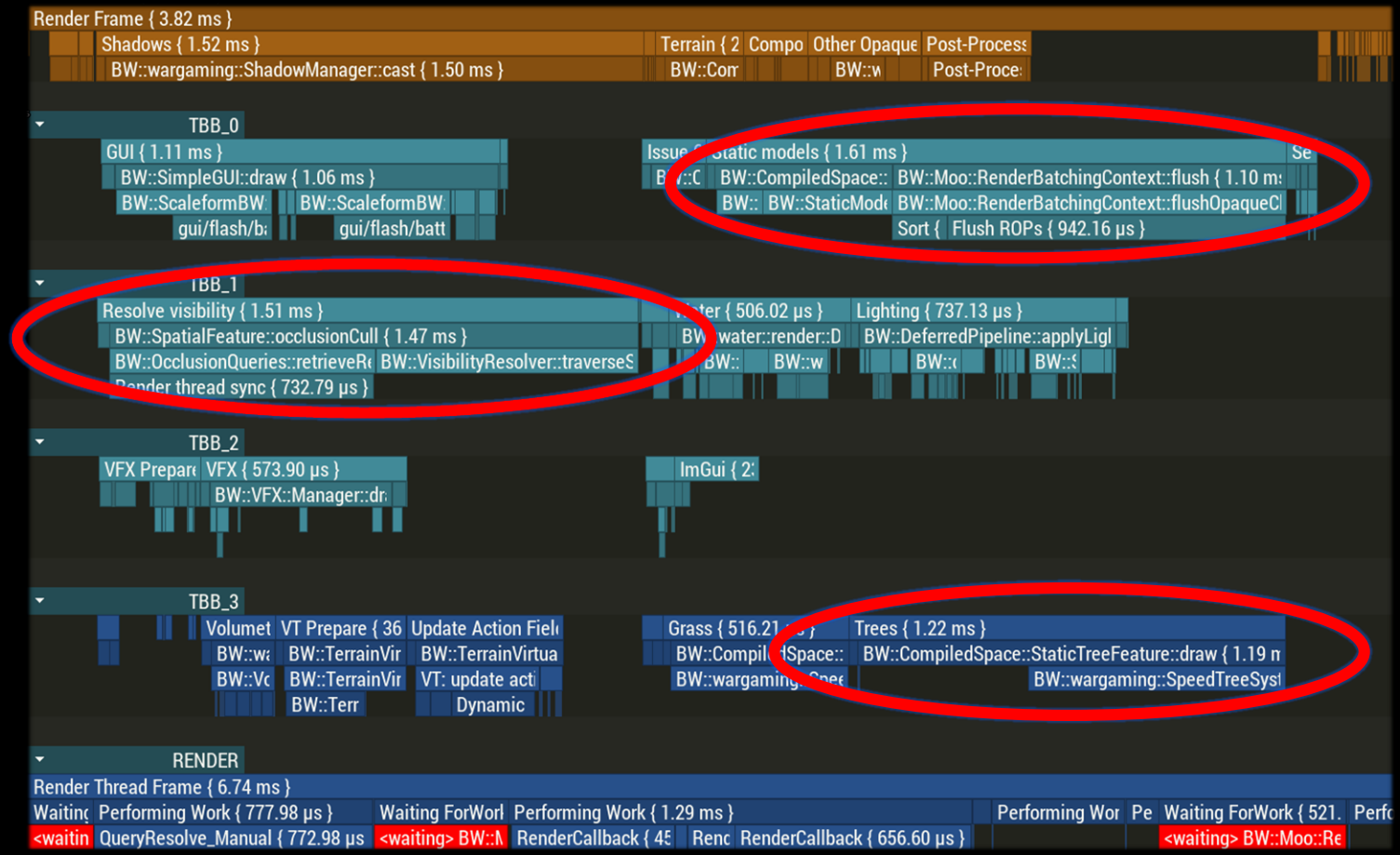
Concurrent rendering

Render frame ~12ms
Parallel execution off



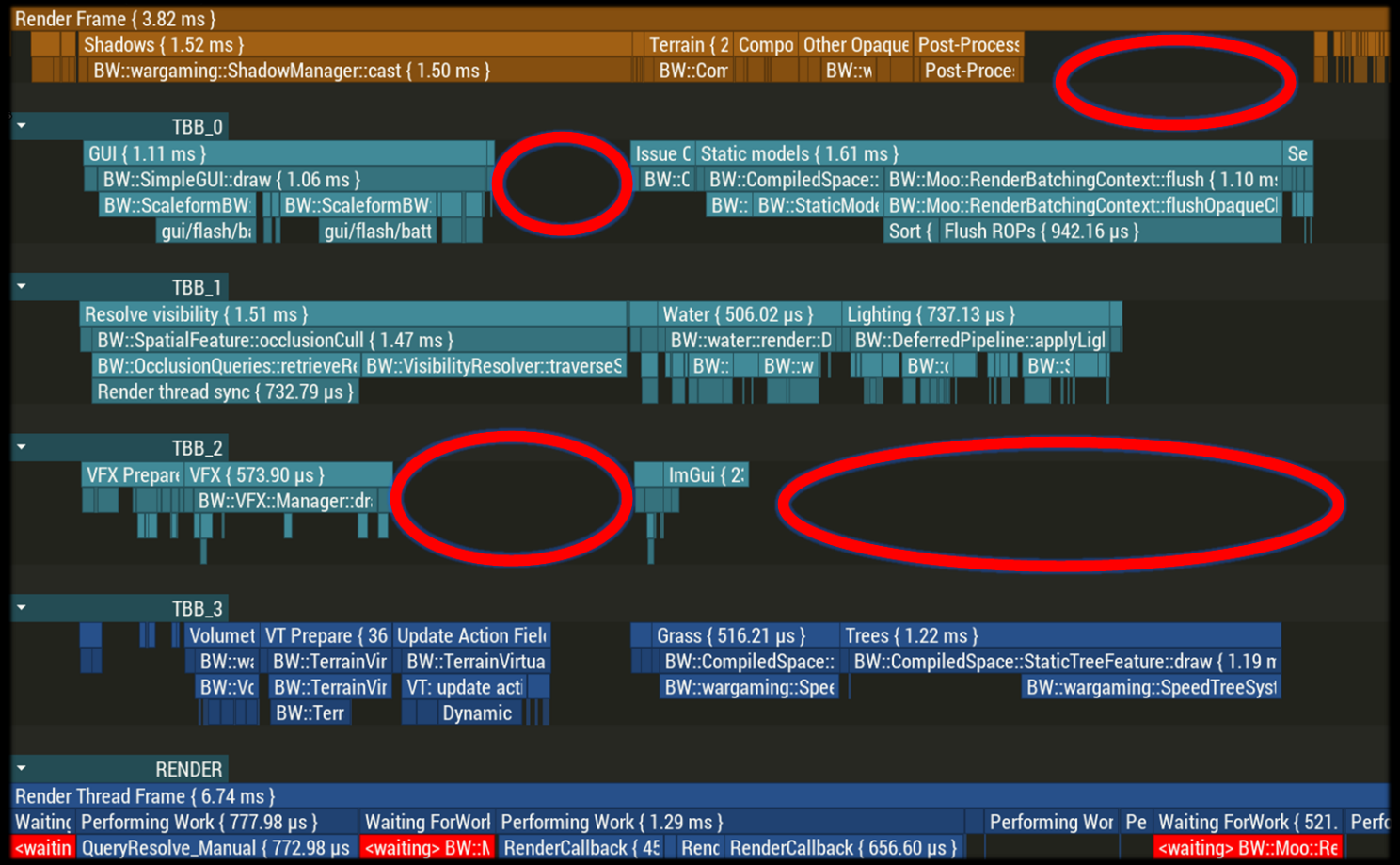
Concurrent rendering

Render frame ~4ms
Parallel execution on
Speedup ~3x



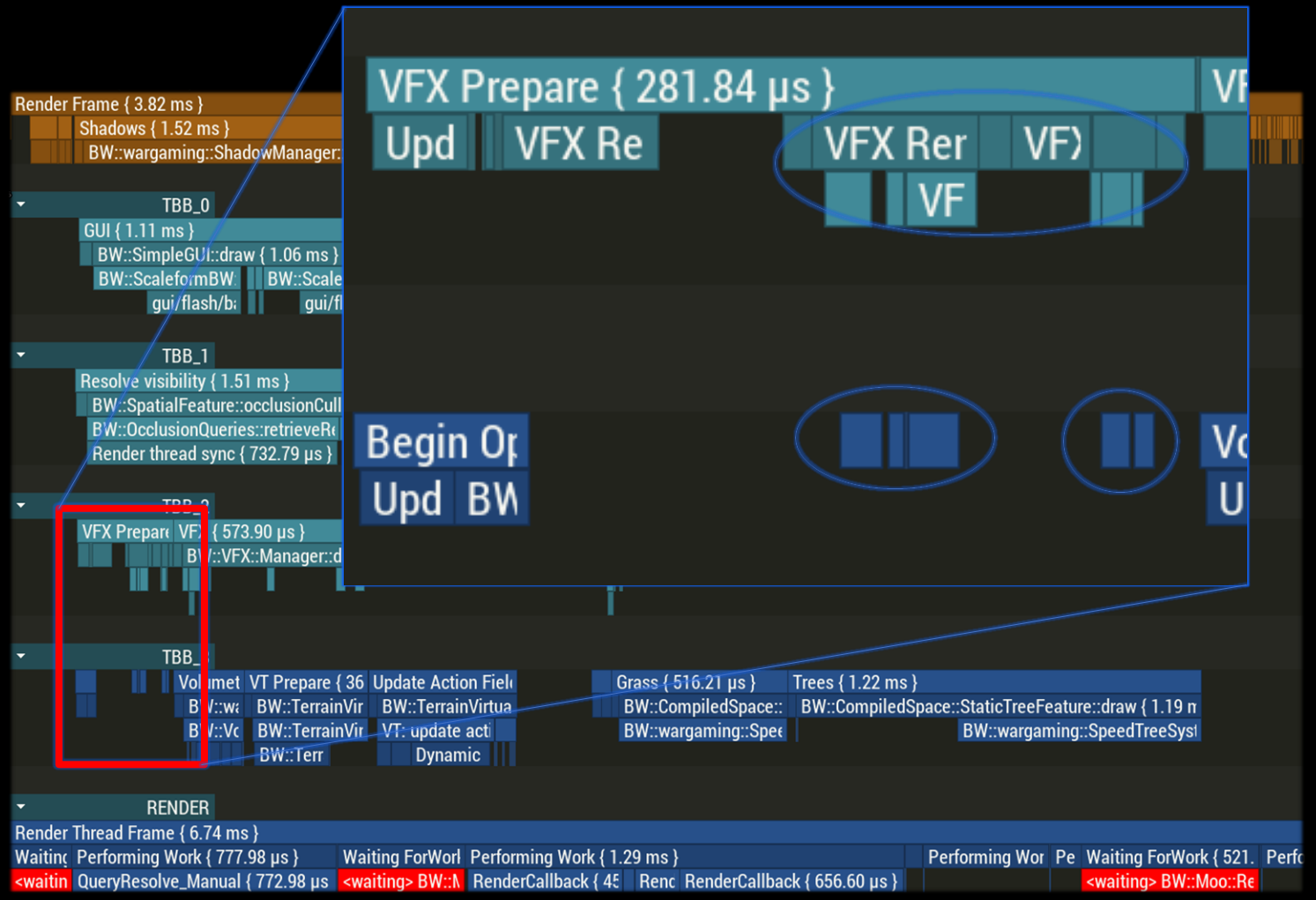
Concurrent rendering

Render frame ~4ms
Parallel execution on
Speedup ~3x



Concurrent rendering

Render frame ~4ms
Parallel execution on
Speedup ~3x
Data parallelism on




CPU
CORE 4
0%
CORE 3
0%
CORE 2
0%
CORE 1
0%
GPU
0%
CORE ENGINE

2014 2015 2016 2017 2018 2019

TANK TREADS SIMULATION

Tank Treads - Previous implementations

- Skinned mesh
  - Visually static
  - The tread moves by scrolling its texture

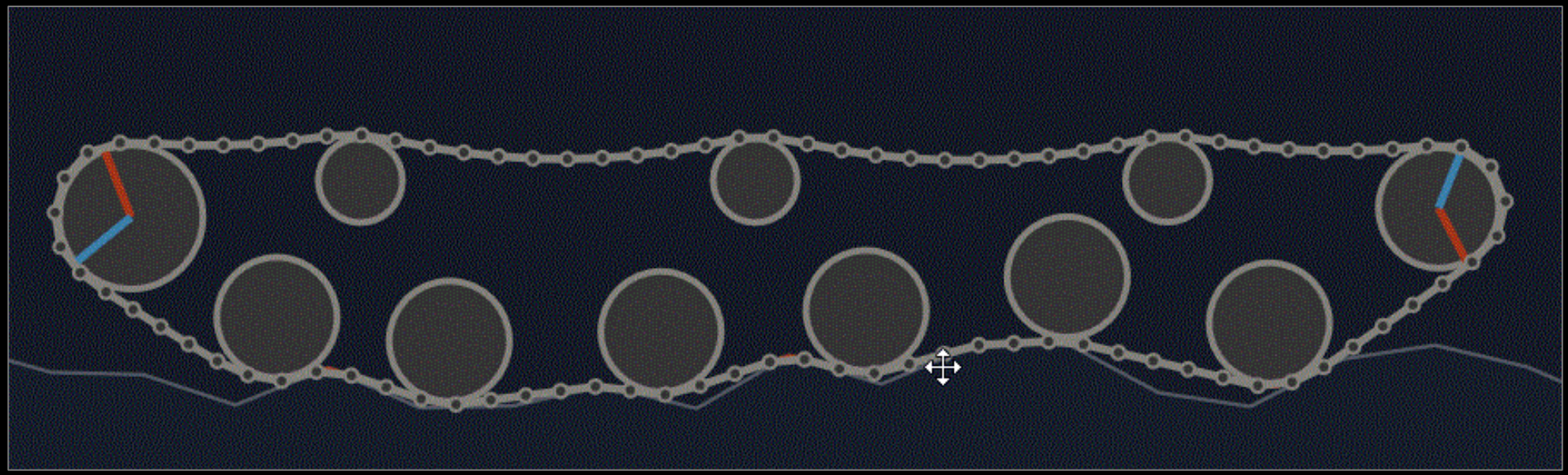
- Spline tread
  - The general shape of the tread is represented by a spline
  - Each segment is rendered as a separate model
  - Segments positions are determined by the spline
  - The spline shape is animated by moving its control points

Tank Treads - Things to improve

- Spline tracks visually superior to skinned meshes, but still have flaws:
  - No proper collisions with the environment
  - Too smooth shapes of curves
  - A very complex tuning process

Tank Treads - Designing the new treads

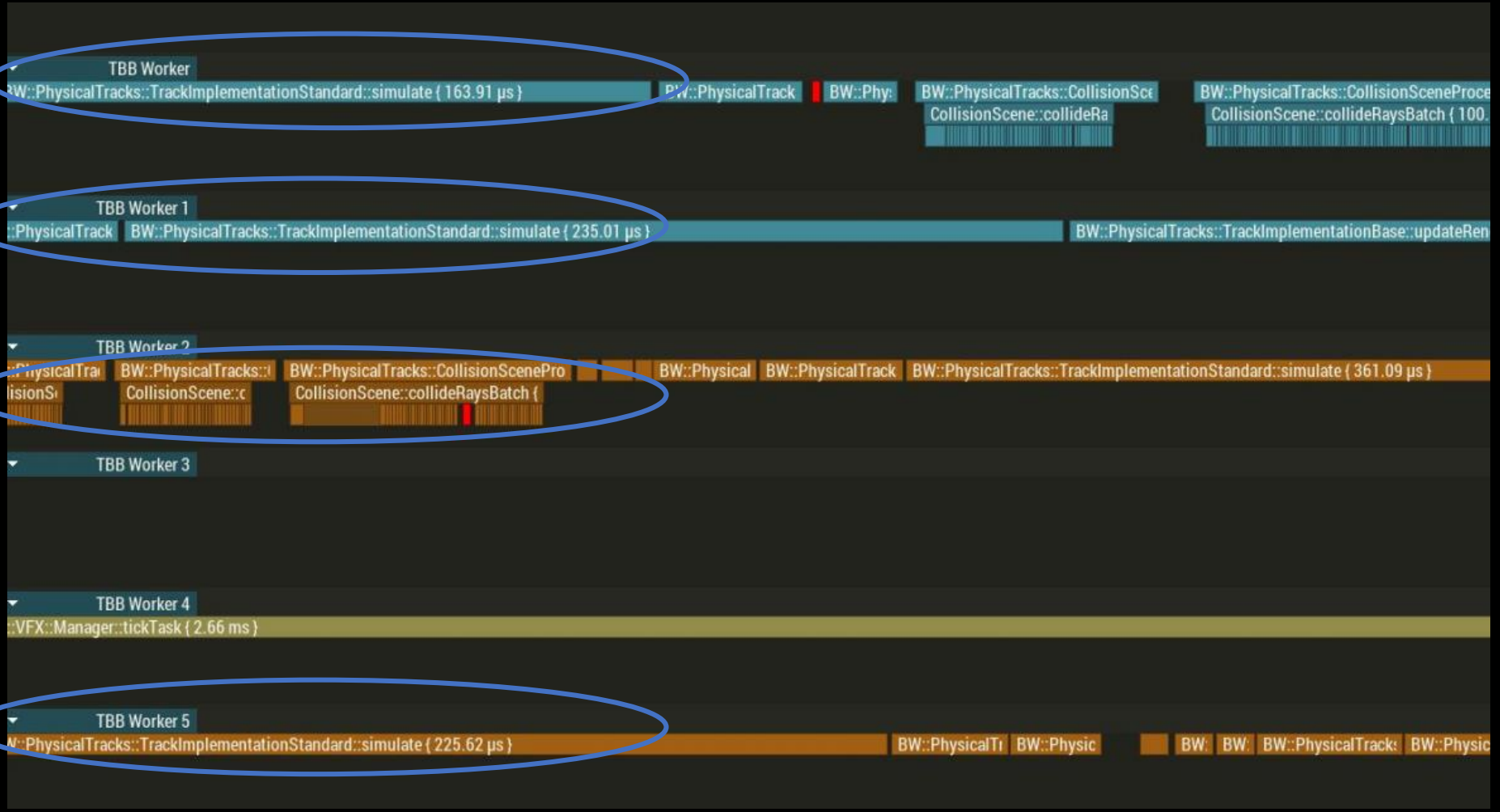
- Spring chain simulation
- Procedural animation
- Collisions with the environment



Tank Treads - General solution & collisions

- Spring chain controls tread shape
- Tread is divided into 4 parts: front, top, back and bottom
- Ray cast the area underneath the tank and from height field
- Collide each spring joint with it

Tank Treads - Performance



2014 2015 2016 2017 2018 2019

OPTIMIZING FOR SIMD

Optimizing for SIMD

Using latest SIMD instruction sets can significantly improve performance

- AVX2 – Optimizing Tank Threads Simulation
  - Intrisics based code
  - 13% performance improvements

Instruction Set	SSE (Previous versions)	AVX2 (Patch 1.6)
AVX2 workload per-frame time	1.00ms	0.87ms

Optimizing for SIMD – Intel SPDM Compiler

An open-source, LLVM-based language and compiler for Intel SIMD architectures:

- Generates high performance vector code for everything from Mobile Computing to HPC
  - SSE/AVX/Xeon Phi™
- Easy to use and integrate into existing code
- Transparently scale with additional resources
  - Moving to newer/bigger hardware? Recompile instead of rewrite!

C code for RGB2Grey

```
void rgb2grey(int N,
              float R[],
              float G[],
              float B[],
              float grey[])
{
  for (int i=0; i<N; i++) {
    grey[i] = 0.3f*R[i]
            + 0.59f*G[i]
            + 0.11f*B[i];
  }
}
```

ISPC code

```
export void rgb2grey(uniform int N,
                     uniform float R[],
                     uniform float G[],
                     uniform float B[],
                     uniform float grey[])
{
  foreach (i=0…N) {
    grey[i] = 0.3f*R[i]
            + 0.59f*G[i]
            + 0.11f*B[i];
  }
}
```

Optimizing for SIMD – Intel SPDM results

Tank Treads simulation using Intel SPMD compiler

- Simple porting from C++ code
- 2x+ performance improvements in Tank Tread Simulation

Instruction Set	SSE (Previous versions)	AVX2 (Patch 1.6)	ISPC AVX2 (WIP)
AVX2 workload per-frame time	1.00ms	0.87ms	0.42ms

2014 2015 2016 2017 2018 2019

HAVOK AND TBB

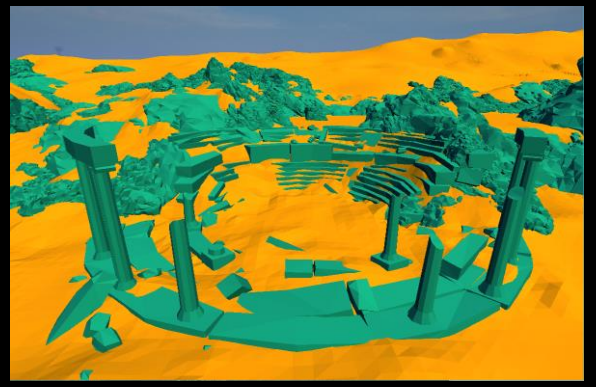
Havok Destructions

Havok® Destructions

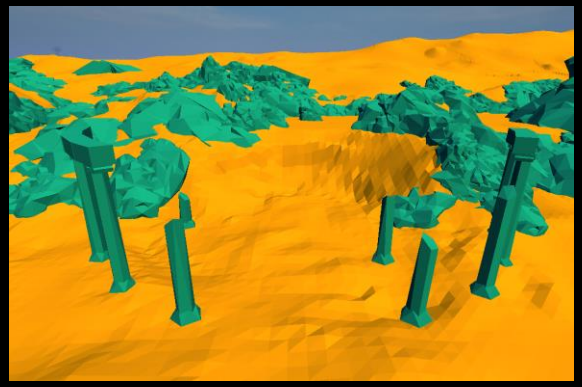
- Powered by Havok® Physics
- Collision and physics in one system
- Rich toolset
- Out of the box multithreading

Game scene

Collision scene



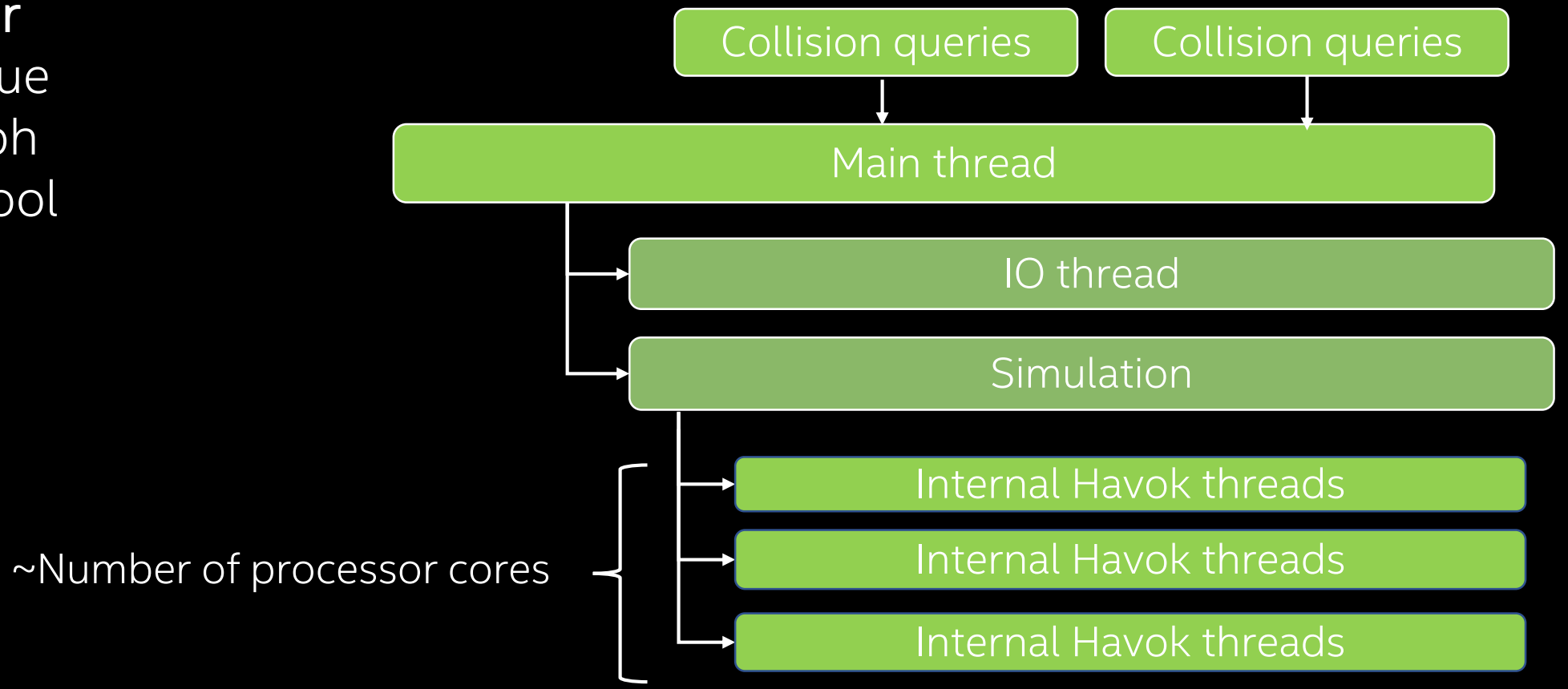
Destruction scene



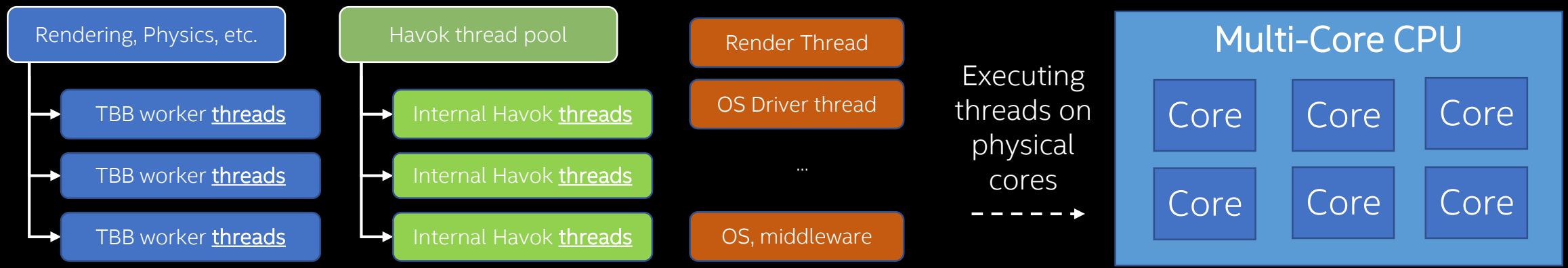
Havok Destructions

Task Manager

- Task Queue
- Task Graph
- Thread Pool

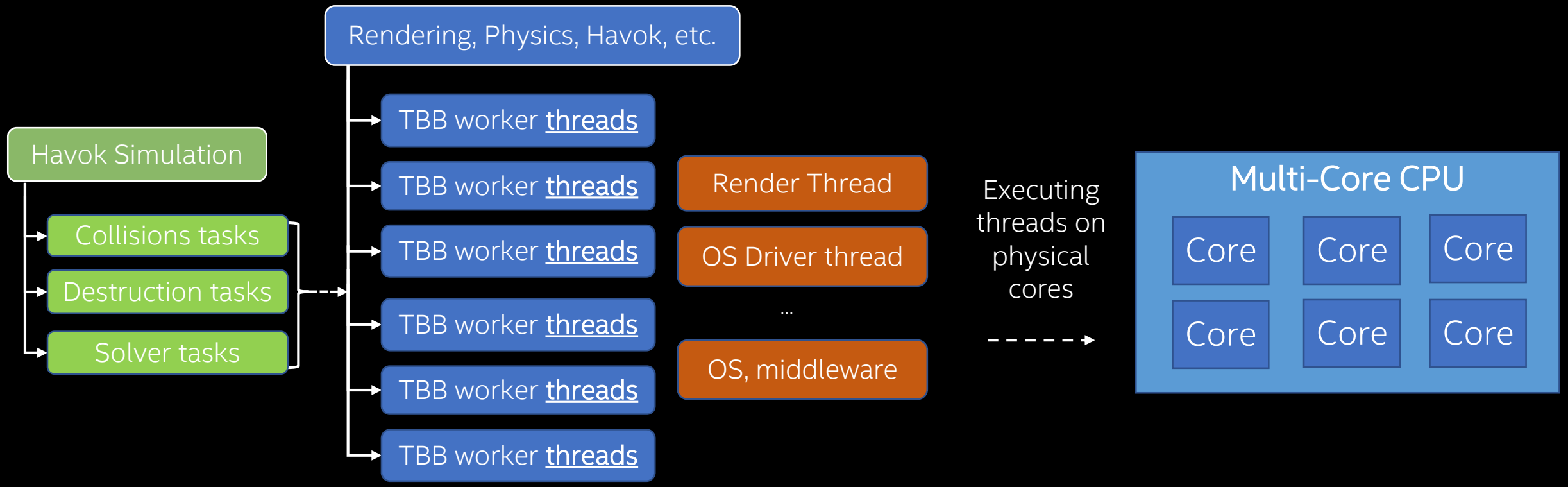


Havok Destructions - problem



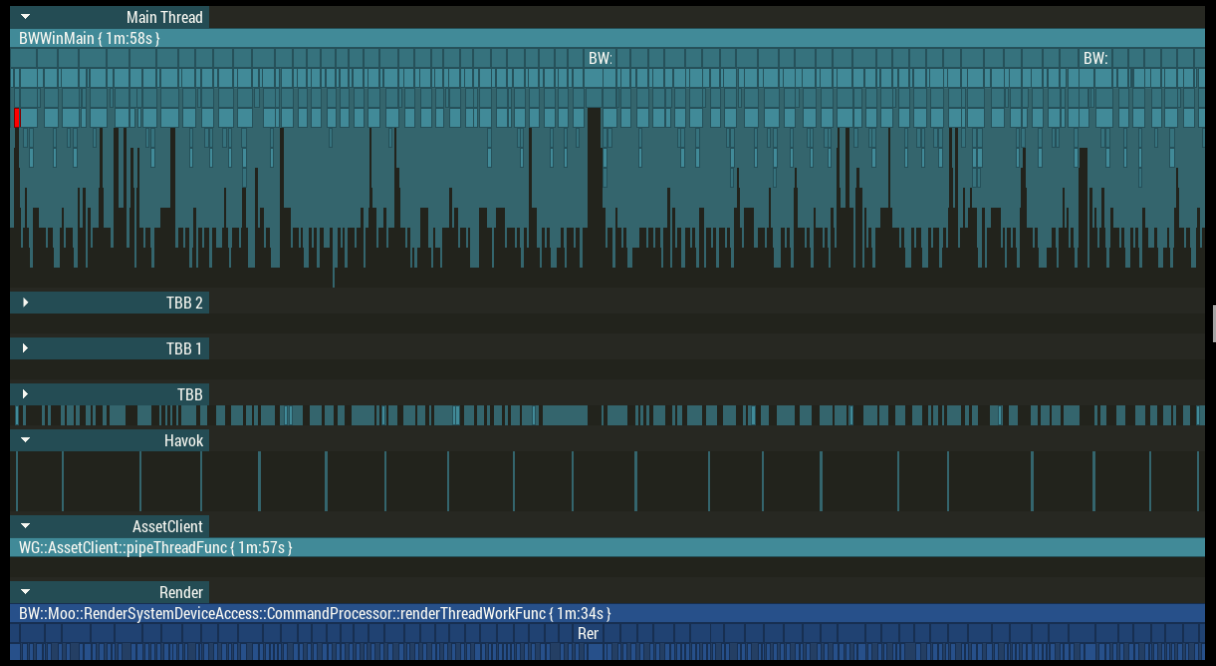
Multiple thread pools and various individual pools could create inefficient task execution which leads to **oversubscription**

Havok Destructions – solution with TBB

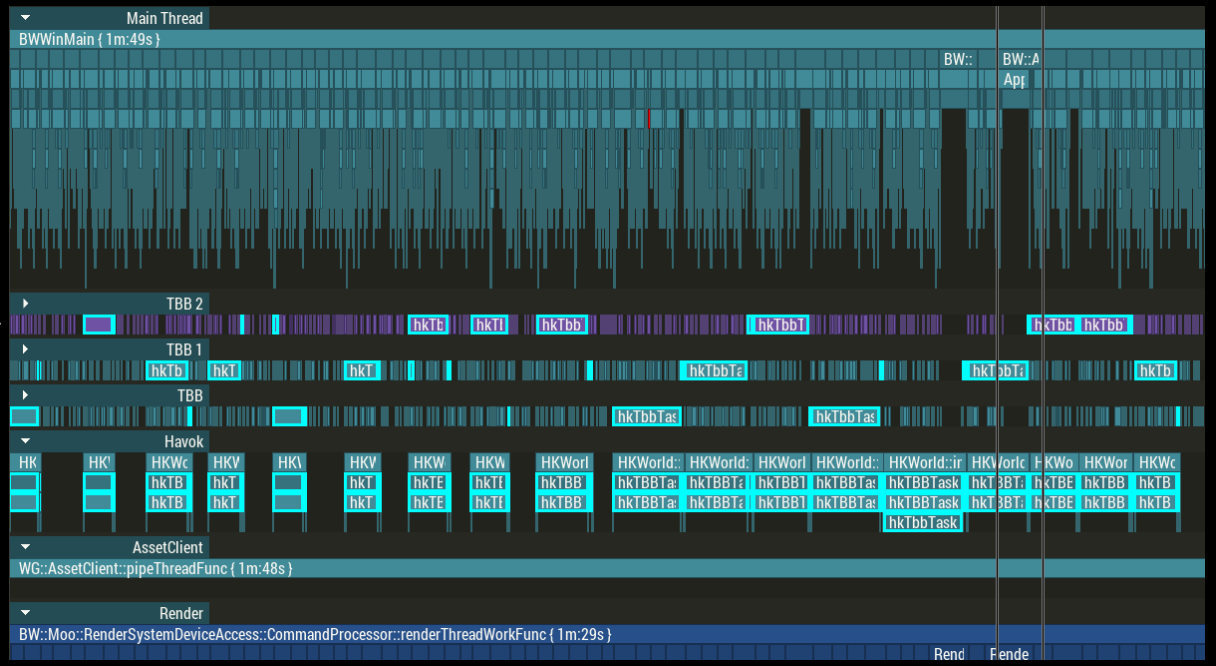


Porting Havok to TBB is the solution

Havok Destructions – solution with TBB



Havok vs TBB



Havok + TBB

2014 2015 2016 2017 2018 2019

RT SHADOWS AND ENCORE RT

Improving Shadow Quality

With all performance optimizations we now had “budget” to improve visual quality above and beyond – and the team started with shadows

Ray Traced Shadows

- Real-time ray traced physically correct soft shadows
- Does not require hardware RT cores
- Direct3D 11 support is minimal requirement
- Intel® Embree for BVH construction
- First game to use real-time RT shadows in D3D11


RT Shadows off


RT Shadows on

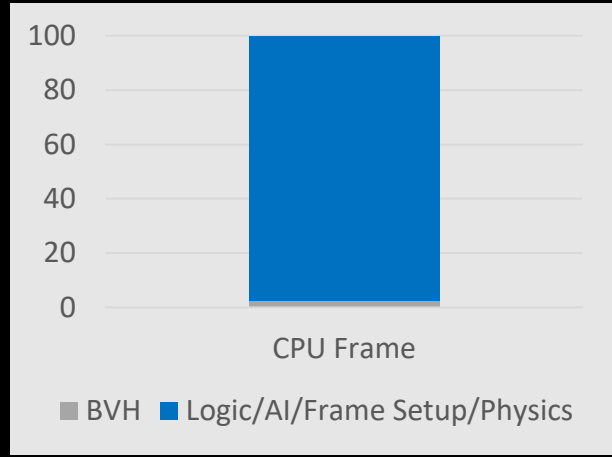
Ray Traced Shadows – Implementation

CPU side

Two level acceleration structure

- BLAS BVH
  - For all tank meshes
  - Constructed once during mesh loading and uploaded to GPU
  - Hard skinned parts in meshes split into multiple static BVHs
  - Soft skinned parts skipped

- TLAS BVH
  - Multiple threads
  - Uses Intel® Embree and Intel® TBB
  - Rebuild every frame and uploaded to GPU

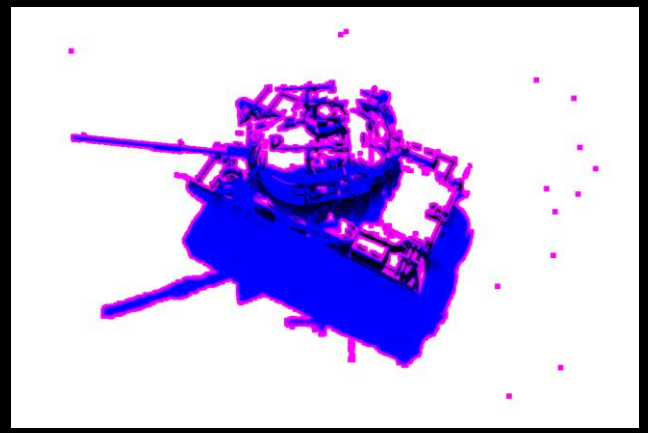


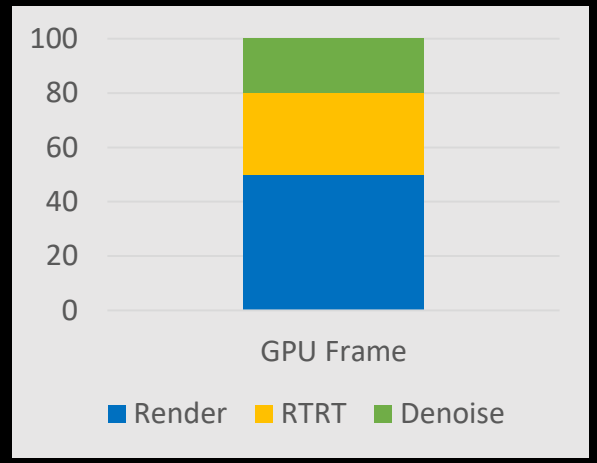
Ray Traced Shadows – Implementation

GPU side

Pixel Shader or Compute Shader

- Temporal ray jittering based on uniform cone distribution
- BVH traversal and ray-triangle intersections
- Temporal accumulation
- Denoiser (based on SVGF)
- Temporal anti-aliasing

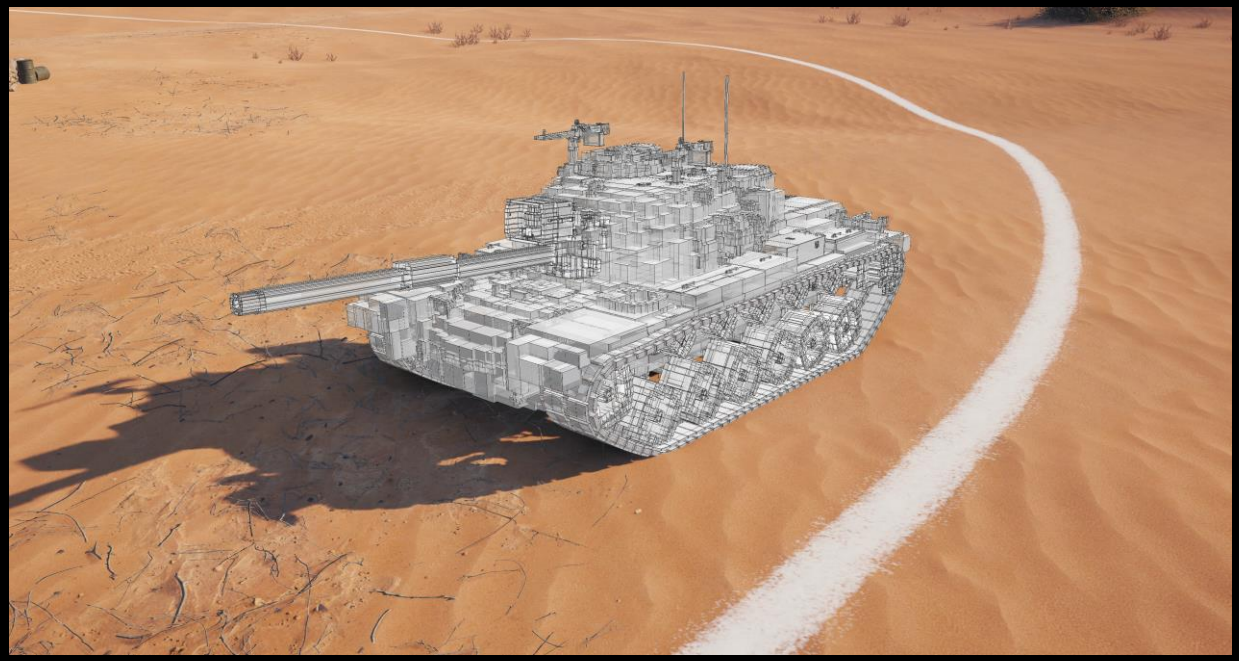




Ray Traced Shadows – BVH

TLAS BVH

BLAS BVH



CPU BVH performance

- **2.5%** of CPU frame time
- TBB threading, SSE 4.2 (**5.5x** faster than original WoT in-house BVH builder)
- Up to ~5mb of GPU data updated every frame
- Up to ~72mb of static GPU data

Ray Traced Shadows – Optimizations

- RT shadows are cast only by tanks
- No support for alpha tested geometry
- BLAS LODs
- 1 ray per pixel
- Don’t trace ray for a pixel if
  - NdotL <= 0
  - If pixel is already shadowed by shadow map
  - Distance to camera is more than 300m

Ray Traced Shadows – Performance

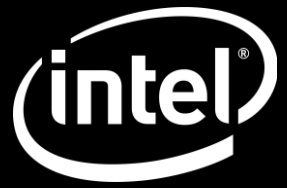
Hi-End PC (i7-4930K NV2080 Ti @ 1080p - Ultra quality)	RT passes (ms)	Total Frame Time (ms)	Total FPS
Low Quality RT – Hard Shadows	**0.89**	**4.78**	**209**
Medium Quality RT – Soft Shadows	**1.07**	**4.97**	**201**
Max Quality RT – Soft Shadows + Denoise + TAA	**1.41**	**5.31**	**188**

Laptop PC (i7-1065G7 Iris Plus @ 720p - Medium quality)	RT passes (ms)	Total Frame Time (ms)	Total FPS
Low Quality RT – Hard Shadows	**5.66**	**18.51**	**54**
Medium Quality RT – Soft Shadows	**10.40**	**23.25**	**43**
Max Quality RT – Soft Shadows + Denoise + TAA	**13.40**	**26.31**	**38**

World of Tanks Encore RT

The new version of WoT Encore was first to include new shadows

- Updated version of World of Tanks Encore Tech demo
- Released on October 15th 2019


intel