POWERING SPATIAL AUDIO ON GPUS THROUGH HARDWARE, SOFTWARE, AND TOOLS

CARL WAKELAND, ADVANCED MICRO DEVICES, INC.

LAKULISH ANTANI, VALVE CORPORATION

AGENDA

Introduction

Steam Audio and TrueAudio Next + Radeon Rays

AMD Resource Reservation Design Features

Multicore optimizations of convolution in TrueAudio Next

Stability demo

Conclusion

INTRODUCTION

STEAM® AUDIO

TrueAudio Next

AMD RADEON Rays

WHY AUDIO ON THE GPU

- HRTFs and occlusion are only the beginning of a realistic sound experience
- Sound reflections are needed to give the user the sense of “being in the environment”
  - Subtle and unconscious, yet tied deeply into emotional cues and engagement level
- Accurate positional reflections change continually with the environment as the user turns their head or moves in the scene
  - Can’t be done with static/pre-rendered reverbs, even when varied in the scene

WHY AUDIO ON THE GPU

- Simulating and rendering reflections with high-order Ambisonics, as done in **Steam Audio**, results in sound reflections that vary physically for improved immersion and presence
- But there is a computational cost
  - Ray Tracing – simulating audio reflection paths
  - Convolution – use ray tracing results to filter sound sources
- What happens if you are too slow?
  - Ray Tracing – updates to the sound filters lag as you move in the scene
  - Convolution: audio popping/dropouts

WHY AUDIO ON THE GPU

- Conventional coping strategies
  - Trace ~100 – 1000 realtime rays at best
  - 1 – 2 time-varying convolutions, if any
- The GPU is great at ray tracing and convolution
  - Use Radeon Rays for ray tracing
  - Use TrueAudio Next for convolution
  - Additional OpenCL kernels to connect the two
- With GPU (in real-time)
  - 16k rays per source, 4+ bounces
  - 512 time-varying convolutions
    - = 32 3rd-order Ambisonics sources
- But you need to do it the right way…

THE CONVENTIONAL GPU WORK MODEL

- The compute shader portion of a GPU uses identical computing blocks called workgroup processors or compute units
- Jobs consisting of kernels and data are submitted from an application to the GPU through a compute queue
- The conventional working model for GPU compute
  - Pushes all work through a single compute queue, targeting the entire array of compute units with each job
  - Runs jobs from different applications in series, but as fairly as possible*
- This design is very performant for massively parallel applications, and provides excellent throughput for multiple compute applications if they do not have absolute periodic latency requirements
- *Note: VR display refresh gets specific treatment

HOW NOT TO RUN AUDIO ON THE GPU

- What can go wrong when using the conventional model for audio?
- Graphics and compute performance on a game is tuned to run well on the conventional model
- Throwing audio into this mix can add performance uncertainty
  - If a ray tracing kernel takes too long, graphics frame rate may vary unpredictably
  - If a graphics kernel takes too long, convolution can miss a deadline, causing audio glitches
- What do we want?
  - Predictable execution times
  - Preventing some kernels from varying performance of other kernels

HOW TO SUCCESSFULLY RUN AUDIO ON THE GPU IN A GRAPHICS-INTENSIVE GAME

Use AMD Resource Reservation:

- Designed to allow one or two configurable, small portions of the GPU compute array to be partitioned away from the rest of the compute units and mapped one-to-one to specific, dedicated queues **called Real-time Queues (RTQs)**
- This is under application control (**opt-in**)

HOW TO SUCCESSFULLY RUN AUDIO ON THE GPU IN A GRAPHICS-INTENSIVE GAME

- **AMD Resource Reservation provides a number of important design features**:
  - RTQs are allocated and released from within the application and are released when the application closes for any reason
  - Kernels running on an RTQ can't use CUs that belong to other queues, kernels running on other queues can't use reserved CUs
    - Kernels can't adversely affect schedule of workloads running on other queues
  - RTQs are limited to up to ~25% of the total CUs in a GPU
  - Each partition's workload becomes self-limiting. If an RTQ's workload is oversubscribed, only its own performance is impacted
  - Memory and bus bandwidth are still shared between all queues, but interaction effects are less significant
  - Profiling by AMD for a stress use case shows that any reduction in frame rate of an intensive gaming benchmark is less than or equal to the percentage of total CUs reserved*

*See claim details in Endnotes.

AUDIO ACCELERATION WITH TRUEAUDIO NEXT (TAN) AND RADEON RAYS IN STEAM AUDIO

- Steam Audio includes optional support for Radeon Rays and TrueAudio Next in real-time
- On supported GPUs:
  - RTQ 1 is used to accelerate audio convolution with AMD TrueAudio Next (TAN)
  - RTQ 2 is used to accelerate simulation of audio reflections, consisting of multiple stages:
    - Ray tracing using Radeon Rays
    - Using ray tracing results to calculate impulse response (IR)
    - Preparing IRs for convolution using Fast Hartley Transform (FHT)
- Can also use CPU-based ray tracing with TAN on GPU
  - In this case, RTQ 2 is still used for FHT
- Example: on Radeon™ RX Vega 64, use 4 CUs for TAN convolution (RTQ 1), 8 CUs for reflection simulation (RTQ 2), remaining 52 CUs are allocated to graphics and non-RTQ compute

STEAM AUDIO AND TRUEAUDIO NEXT + RADEON RAYS

Lakulish Antani

Valve Corporation

STEAM® AUDIO

TrueAudio Next

AMD RADEON Rays

STEAM® AUDIO AMD

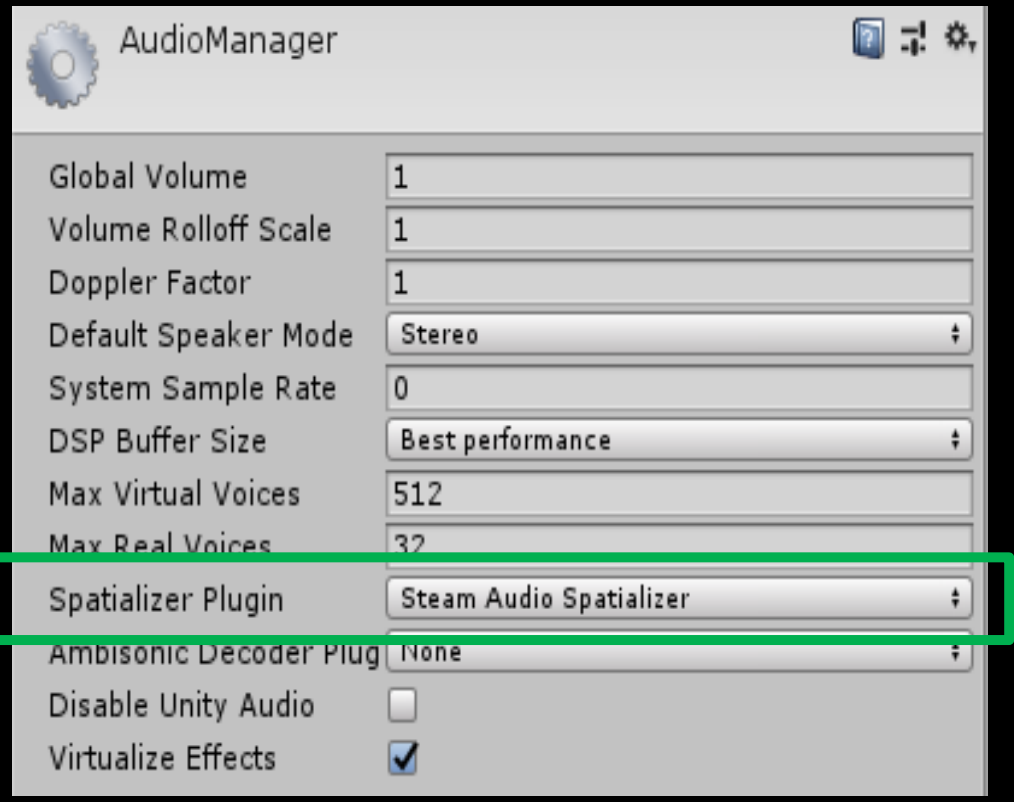
QUICK TUTORIAL

This tutorial is based on the Steam Audio Unity plugin and Unity's native audio engine

For information on other integrations, see the Steam Audio docs

QUICK TUTORIAL : STEP 1

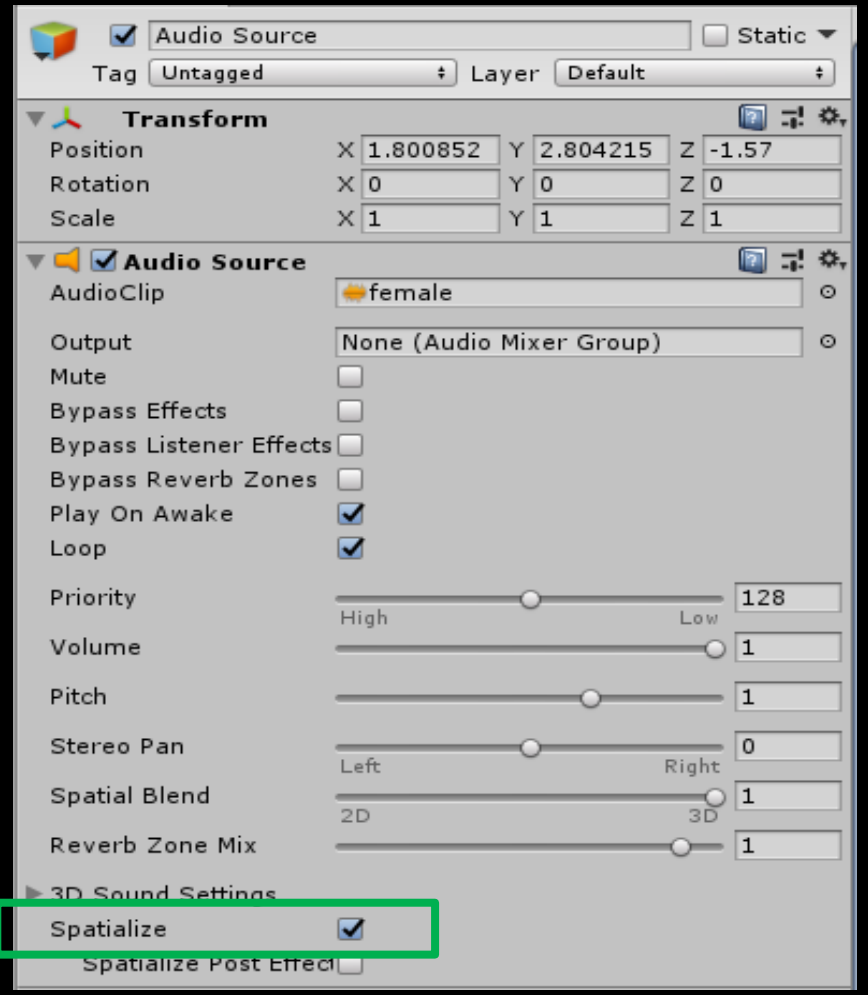
- Enable Steam Audio spatialization
  - Click Edit > Project Settings > Audio
  - Set Spatializer Plugin to Steam Audio Spatializer



STEAM AUDIO AMD

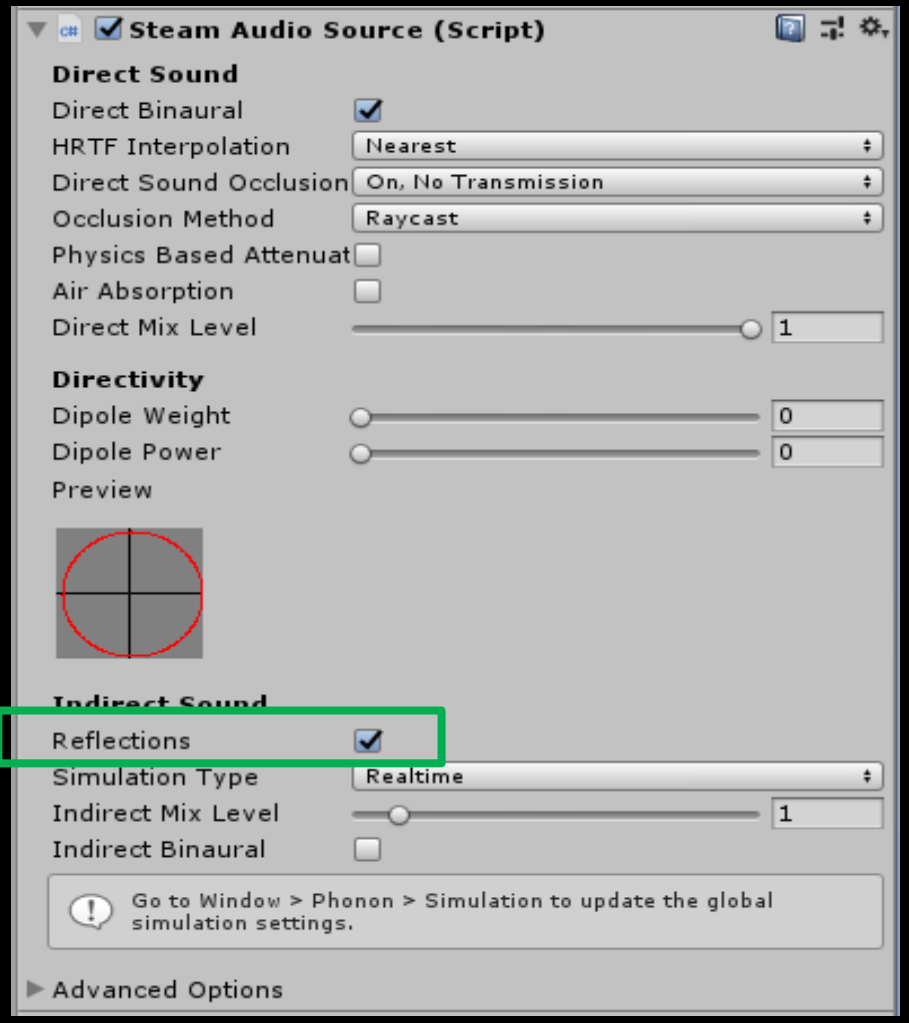
QUICK TUTORIAL : STEP 2

- Set up Audio Sources to be spatialized
  - Select Audio Sources, check Spatialize



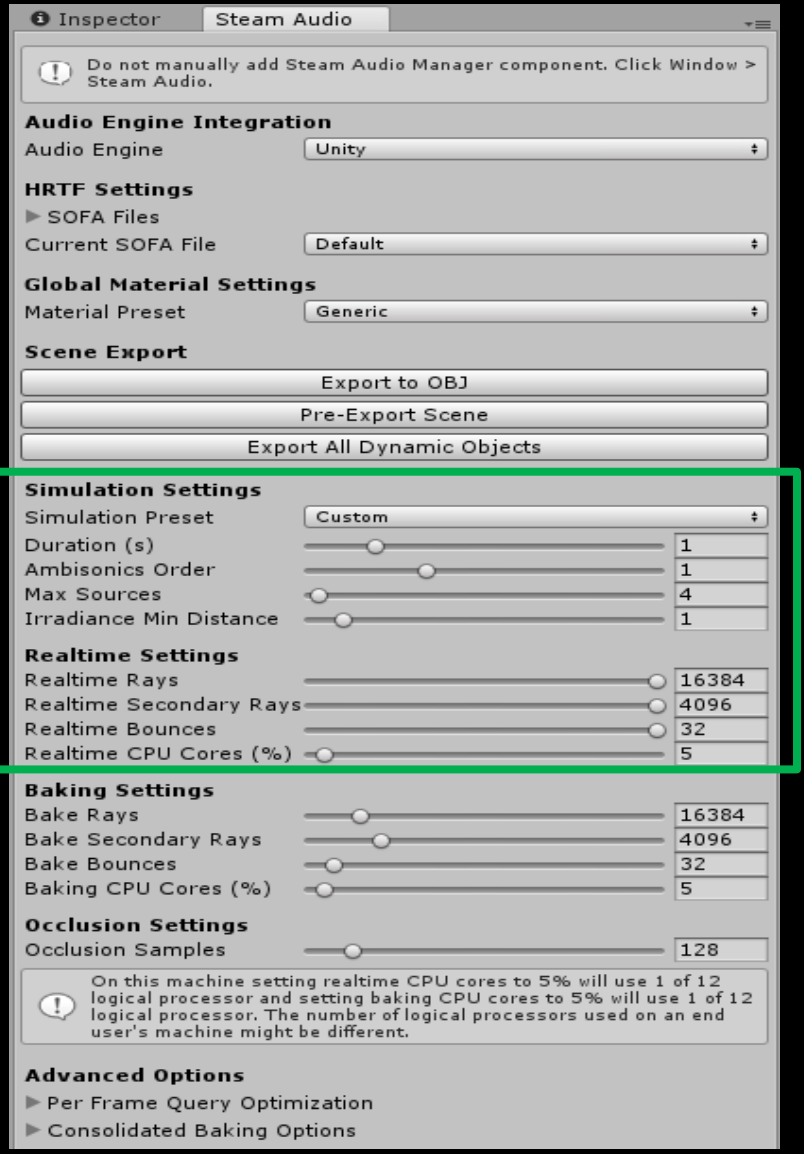
QUICK TUTORIAL : STEP 2

- Set up Audio Sources to be spatialized
  - Select Audio Sources, check Spatialize
- Enable indirect sound
  - Add a Steam Audio Source component to the Audio Source
  - Check Reflections



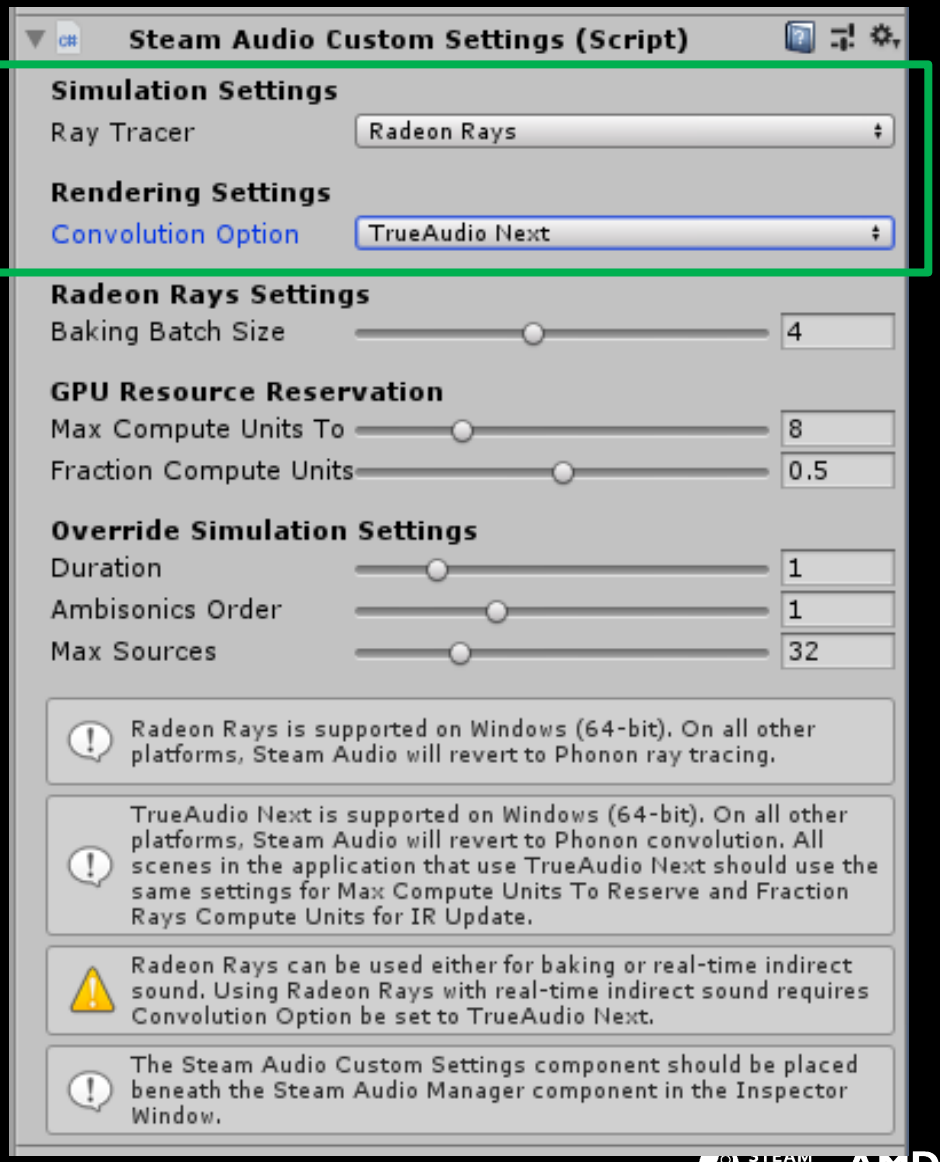
QUICK TUTORIAL : STEP 3

- Adjust simulation settings
  - Click Window > Steam Audio
  - Adjust Realtime Rays, Duration, Ambisonics Order, etc.



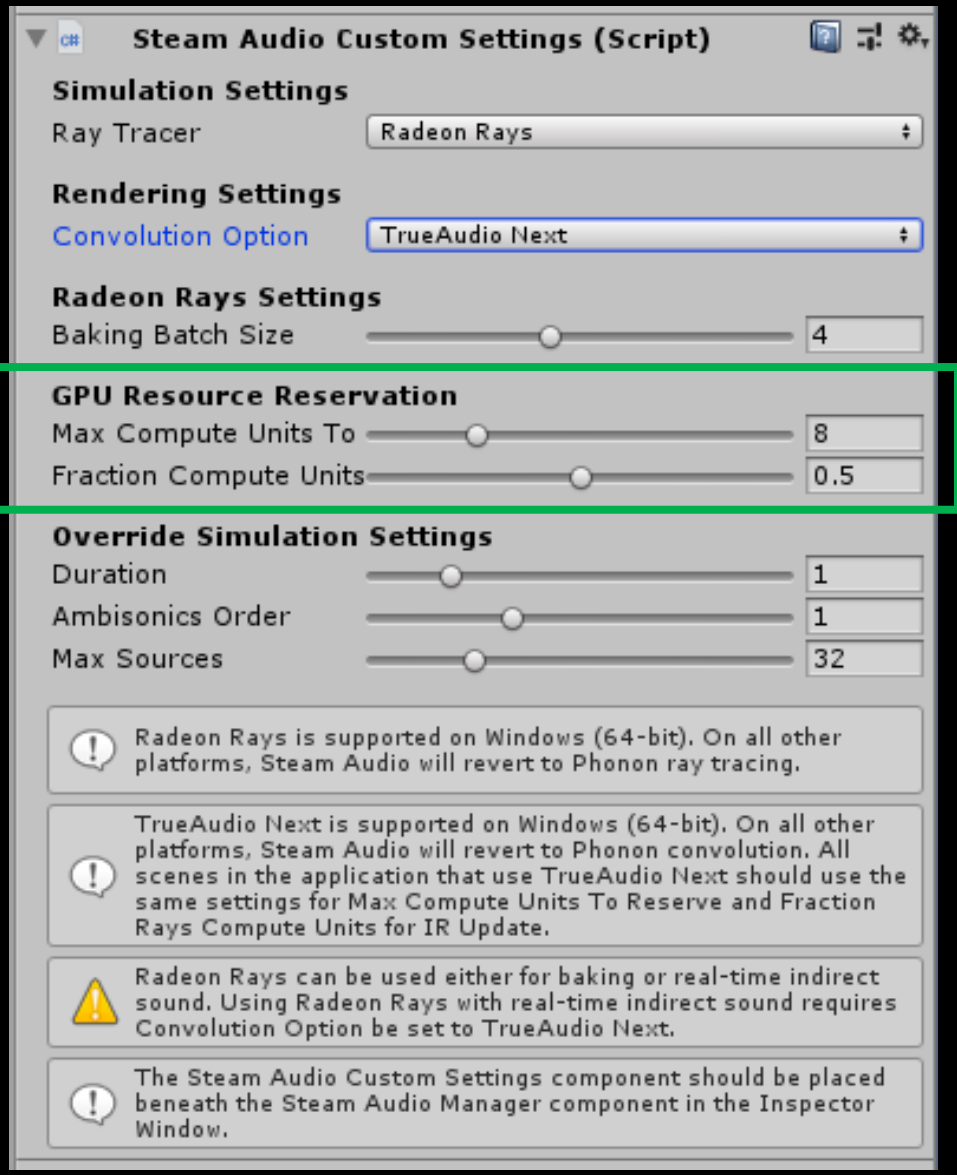
QUICK TUTORIAL: STEP 4

- Enable TrueAudio Next and Radeon Rays
  - Select Steam Audio Manager Settings
  - Add a Steam Audio Custom Settings component
  - Set Ray Tracer to Radeon Rays
  - Set Convolution Option to TrueAudio Next



CU RESERVATION

- Configure GPU Resource Reservation settings in the Steam Audio Custom Settings component
- Max Compute Units: total number of CUs to use for audio
- Fraction Compute Units for IR Update: the percentage of reserved CUs to use for IR update
  - Ray tracing, IR reconstruction, FHT



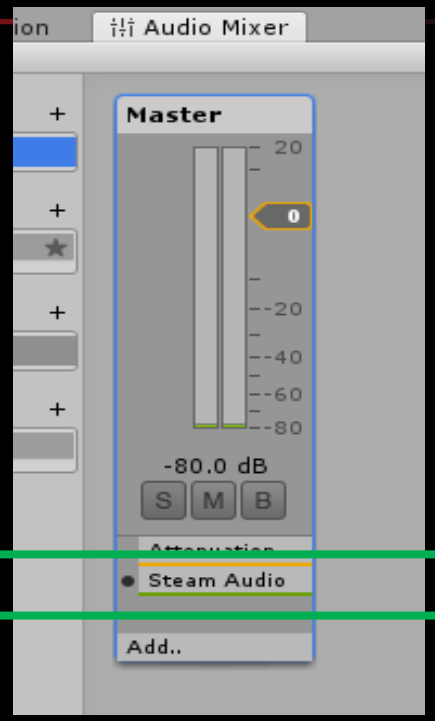
CU RESERVATION

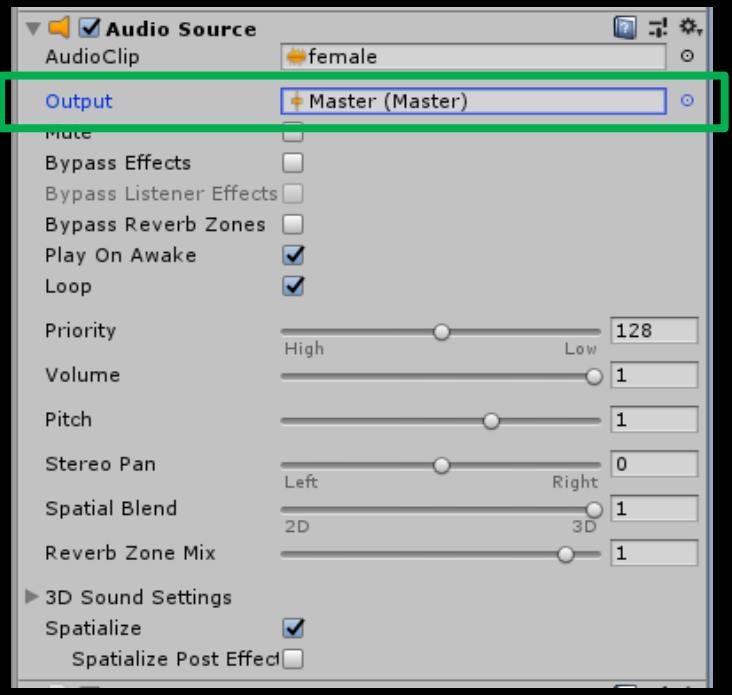
- Choosing total number of reserved CUs
  - Driven by overall GPU workload
- Choosing the fraction of reserved CUs used for IR update
  - Higher IR duration, Ambisonics order: spend more CUs on convolution
  - Many sources, complex scenes: spend more CUs on IR update
  - Test!
- Note: If using TAN with CPU ray tracing, spend 50% of reserved CUs on IR update

MIXER RETURN EFFECT

- Unity Mixer Effect: apply DSP processing to a submix from multiple audio sources

- When using TAN, always create a Steam Audio Mixer Return Effect:
  - Create at least one Mixer Group
  - Add Steam Audio Mixer Return effect
  - Set the output of the Audio Source to this Mixer Group





STEAM AUDIO AMD

MIXER RETURN EFFECT

- With Mixer Return Effect: audio sent to the GPU for processing in a single batch
- Without: audio sent to the GPU once per source, incurring overhead

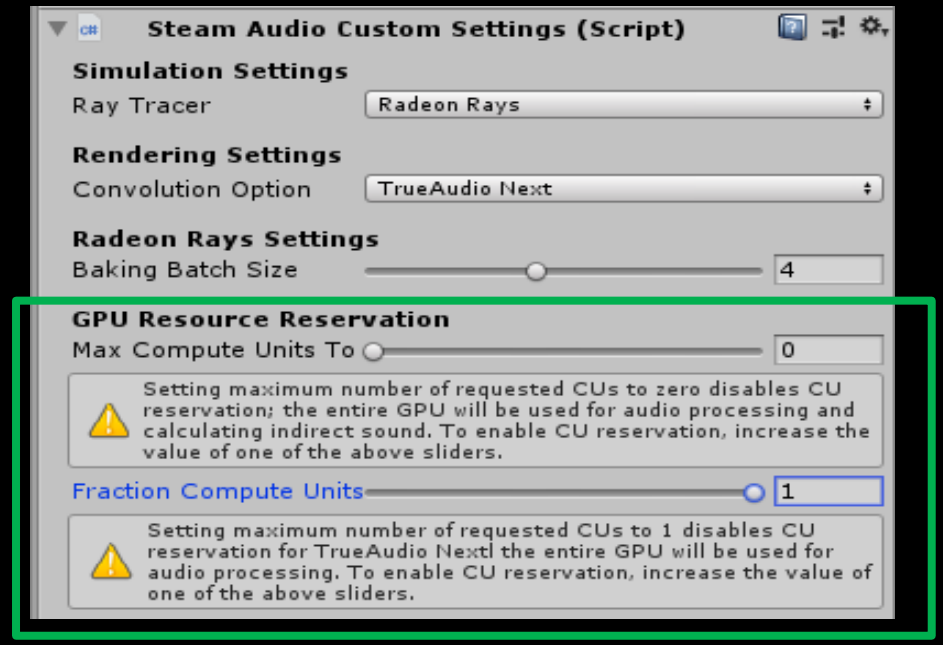
- Audio is taken out of Unity's pipeline at the Audio Source, reintroduced at the Mixer Return Effect
  - Can't insert per-source DSP between spatializer and mixer

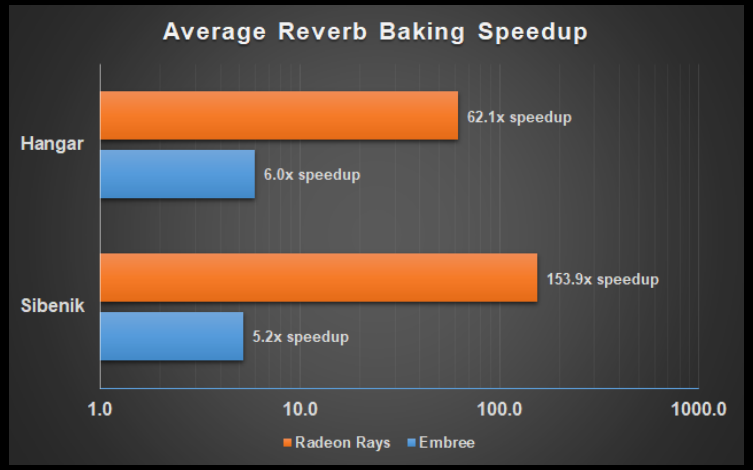
SETTING UP A GOOD CPU FALLBACK SCENARIO

- If the user's hardware doesn't support GPU functionality, Steam Audio will automatically fall back to CPU
- If falling back to CPU for ray tracing:
  - Reduce the number of rays, bounces (linear speedup)
  - Reduce the number of sources that use propagation (linear speedup)
  - If possible, increase the number of CPU threads used for propagation (close to linear speedup)
  - If supported, use Embree
- If falling back to CPU for convolution:
  - Reduce the Ambisonics order to 0 or 1 (quadratic speedup)
  - Reduce the IR duration (linear speedup)
- This behavior is customizable in Steam Audio integration code, which is open source

RADEON RAYS AND BAKING

- Baking: precompute sound propagation from static sources
  - Like global illumination
  - Geometry must also be mostly static
- Radeon Rays can be used to accelerate baking as well
- In this case, disable CU reservation, use the entire GPU for simulation
- Example speedups on Radeon™ RX Vega 64: 10-30x faster than single-core Embree





https://steamcommunity.com/games/596420/announcements/detail/1681419156989664451

STEAM AUDIO AMD

AMD RESOURCE RESERVATION DESIGN FEATURES

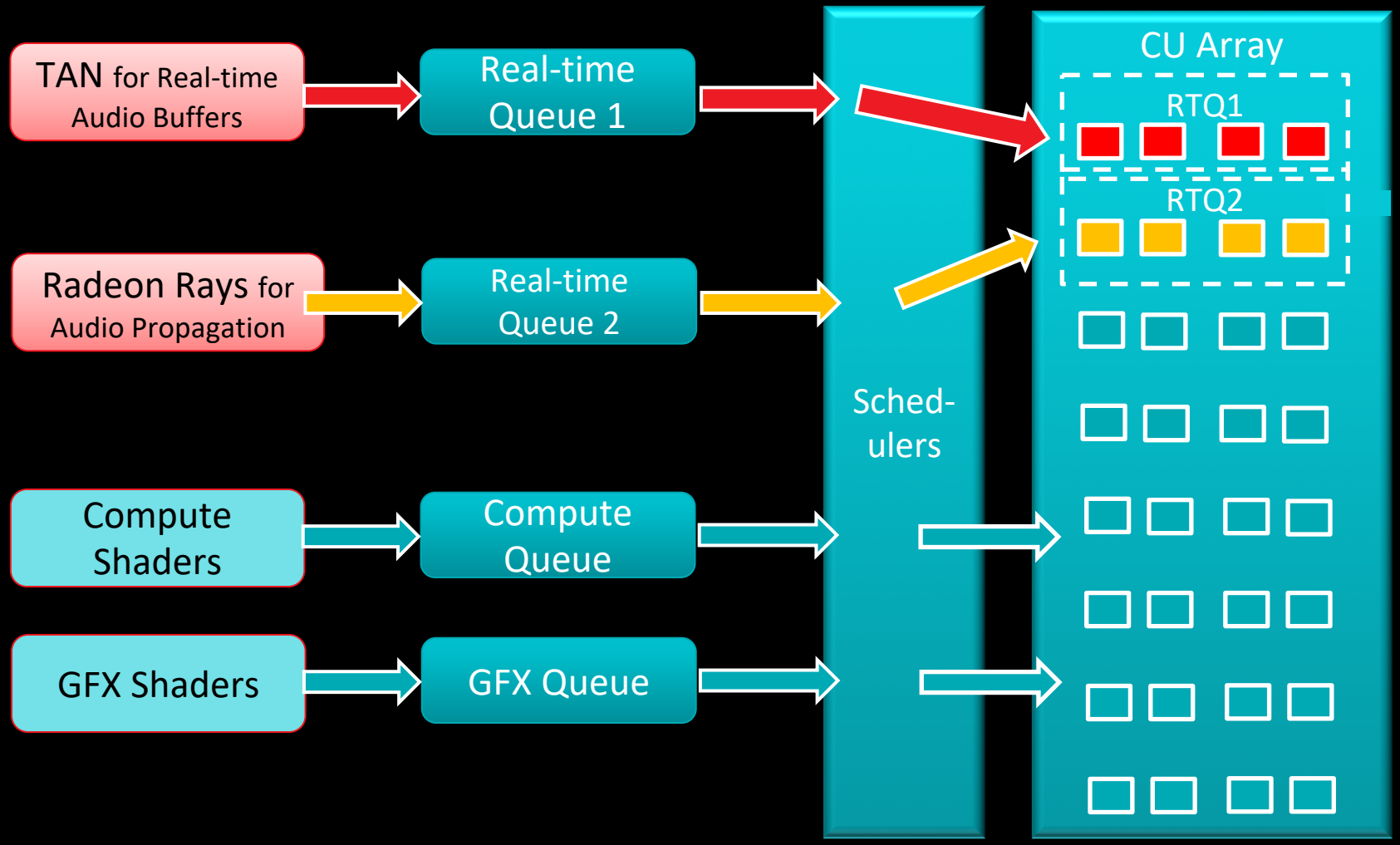
STEAM® AUDIO

TrueAudio Next

AMD RADEON Rays

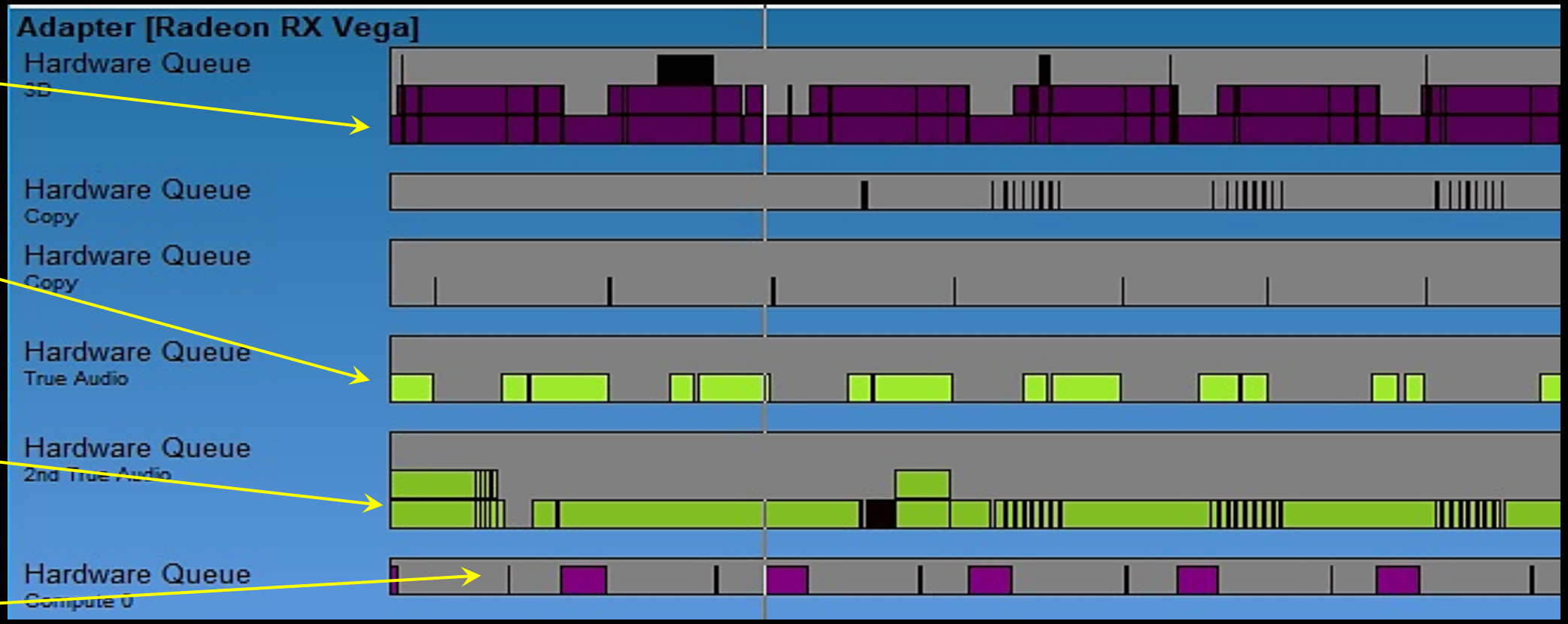
AMD RESOURCE RESERVATION: CONSISTENT, SCALABLE REAL-TIME PERFORMANCE

- A pool of up to ~25% of the available CUs on a GPU may be allocated to 2 RTQs. CUs can be allocated in blocks of 4
- Available CUs may be distributed to TAN and Radeon Rays under direction of parameters provided from the game to the Steam Audio API
- TAN can be used with Radeon Rays, or with a CPU-based ray tracer (choice is left to the game developer)
- OK to use acceleration in some scenes, CPU-only in other scenes



STEAM AUDIO AMD

AMD RESOURCE RESERVATION PROVIDES CONCURRENCY AND INDEPENDENCE OF AUDIO AND GRAPHICS + COMPUTE



This Windows Performance Recorder capture trace (as seen in GPUView) shows four queues running independently on one GPU, in a Direct X® 12 scenario: TAN, Radeon Rays, 3D graphics, and a gaming compute shader process

STEAM AUDIO AMD

RESOURCE RESERVATION PARAMETERS AND OPTIONS

- CUs are allocated from the available pool of reservable CUs. A GPUUtilities query is made from Steam Audio to determine:
  - Total number of reservable CUs (currently 8, 12, or 16)
  - Minimum reservation granularity (currently units of 4)
  - Whether Dual RTQ is supported (current AMD Radeon™ Software Adrenalin 2019 Edition drivers support it)
  - GPU's relative TAN audio convolution performance (AMD Radeon™ RX 480 = 1.0). *Note this is not indicative of the GPU's graphics performance
- The CUs reserved are allocated between the two RTQs by Steam Audio
  - Recommend 4 CUs for TAN convolution, and 4 to 12 for the second RTQ
  - The second RTQ runs the full propagation stack if Radeon Rays real-time is used, otherwise it runs the fast-Hartley transform on impulse responses received from the CPU ray-tracer

MULTICORE OPTIMIZATIONS OF AUDIO CONVOLUTION IN TRUEAUDIO NEXT

STEAM® AUDIO

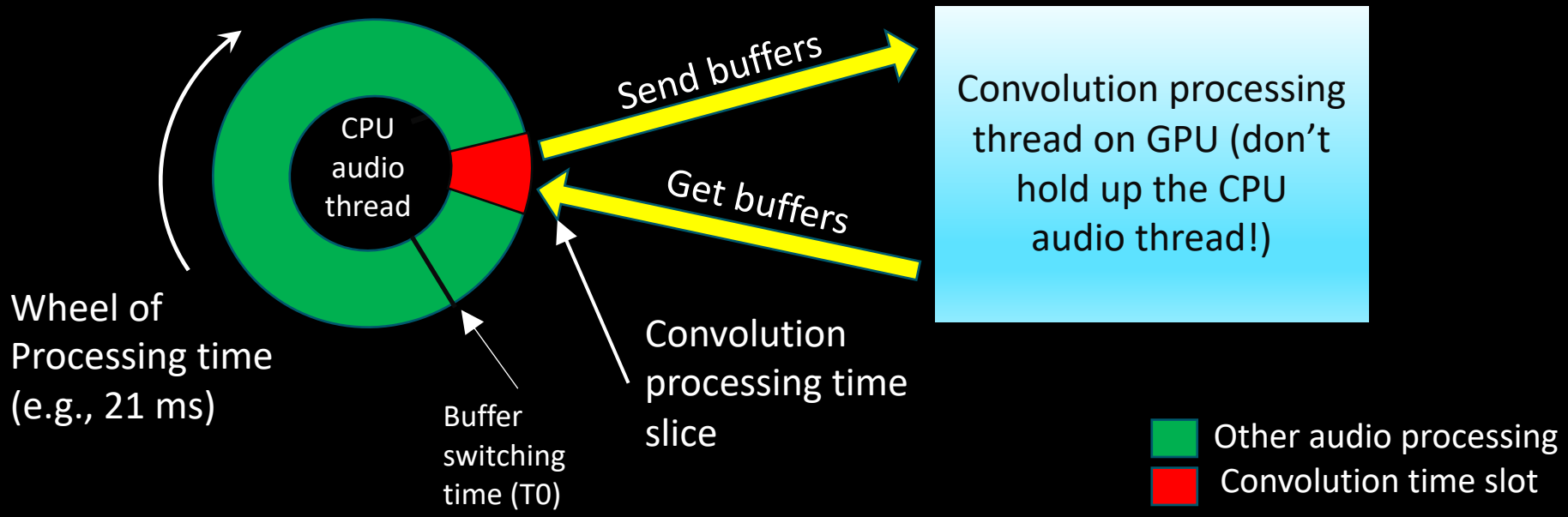
TrueAudio Next

AMD RADEON Rays

STEAM® AUDIO AMD

MULTICORE OPTIMIZATION OF AUDIO CONVOLUTION IN TRUEAUDIO NEXT

- Resource Reservation is only one part of the performance story of TrueAudio Next convolution. There is also a bit of math involved
- An effective multi-core convolution implementation is essential in order to minimize the latency of the convolution process that is directly visible to the main audio buffer loop in the game engine



HIGH-LEVEL TAKEAWAY: THE IMPORTANCE OF BATCHING FOR THE GPU

- Even though we are partitioning the GPU's compute units, the audio workloads we send to the GPU still need a "GPU-friendly" implementation to get high throughput and utilization
- The way we achieve this is to batch the GPU calculations in each frame to the maximum extent possible
- OpenCL allows us to parallelize kernels in up to three dimensions. We can apply these dimensions anywhere that parallel calculations are possible:
  - Simultaneous sources
  - Simultaneous channels in an Ambisonics source
  - Multiple parts of a spectrum processed in parallel
  - …etc.

CONVOLUTION AND THE CONVOLUTION THEOREM (QUICK REVIEW)

- Convolution is a generalized form of a filter, one that applies the impulse response of an environment to a signal to model the effect of the environment upon the signal
- The convolution theorem is the basis of fast convolution. For audio signals, the convolution theorem states that the convolution of two signals is equivalent to point-wise multiplication of the Fourier transforms of those signals followed by the inverse transform:

$$x \copyright y = IT(FT(x) * FT(y))$$

…where © is a convolution, * is an element-wise multiplication in the transformed space, FT is a forward Fourier transform, and IT is the inverse Fourier transform

Why this matters: *Point-wise multiplication of Fourier transforms is much faster than brute-force convolution* (Brute force: $\mathbf{y}[n] = \sum_{k=0}^{L-1} \mathbf{x}[k]\mathbf{h}[n-k]$)

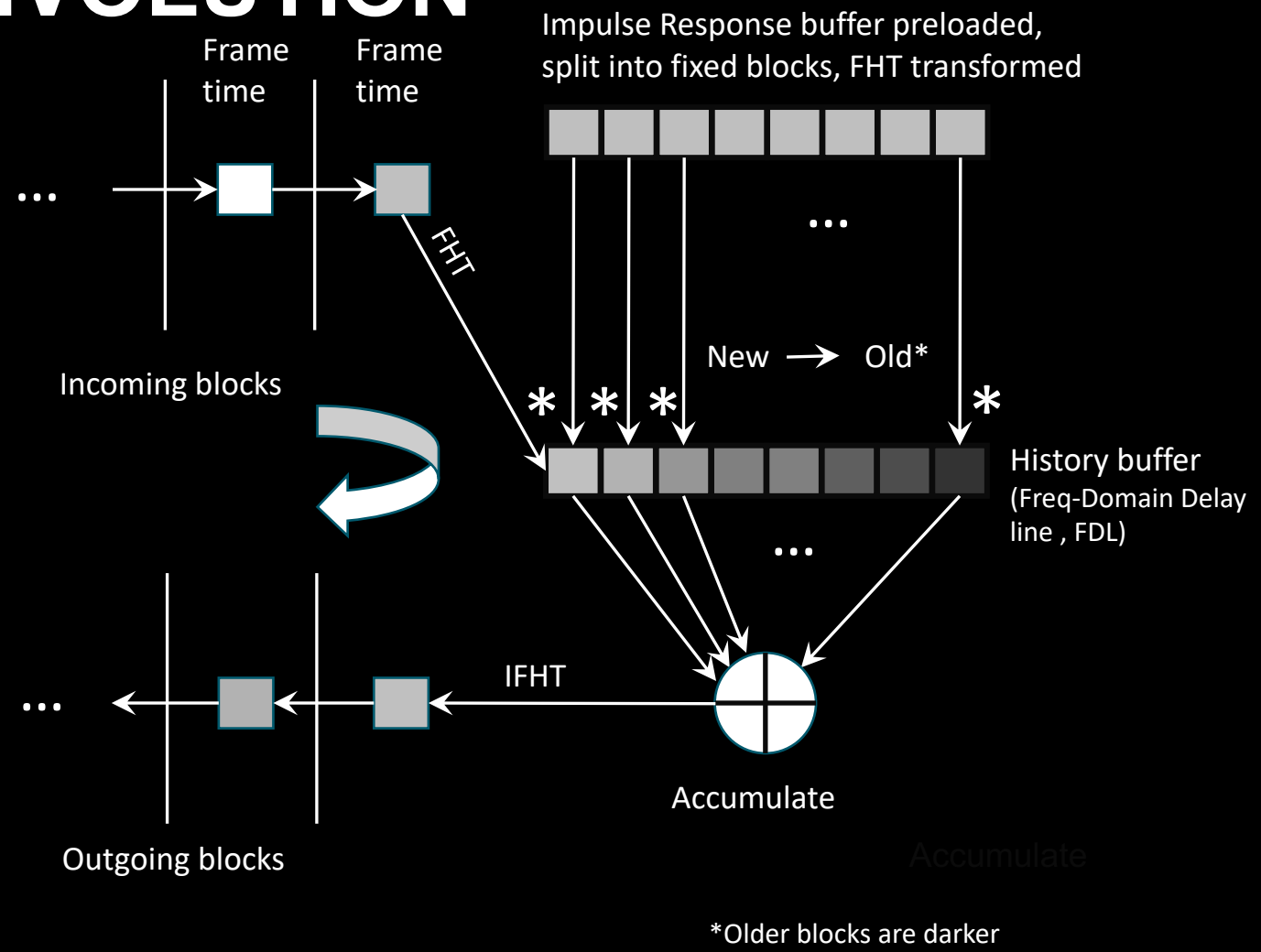
Despite this, fast convolution is still an expensive computation at large scales. Even more so when the IR kernels are being updated periodically

FIRST OPTIMIZATION: USE FAST-HARTLEY TRANSFORM INSTEAD OF FOURIER

- For audio signals, which only have real values (no imaginary part), the Fast Hartley Transform (FHT) is useful because it requires only half of the memory of the Fast Fourier Transform (FFT), with only a very small additional calculation overhead
- Saving memory is important so as to maximize the memory available to graphics and other compute processes
- The FHT also obeys the convolution theorem

SECOND LEVEL OF OPTIMIZATION: UNIFORM PARTITIONED CONVOLUTION

- Partitioned Convolution splits the IR buffer into fixed blocks (partitions) in the frequency domain to let them run faster
- Partitions have a uniform size equal to the audio frame size (typically 1024 samples). Zero-pad excess
- History buffer stores transformed samples. Visible latency is same as buffer size
- In each buffer time slot:
  1. Pointwise multiply-accumulate all transformed history buffer and IR partitions
  2. Sum to form the output
  3. Inverse transform back to the time domain, advance history buffer

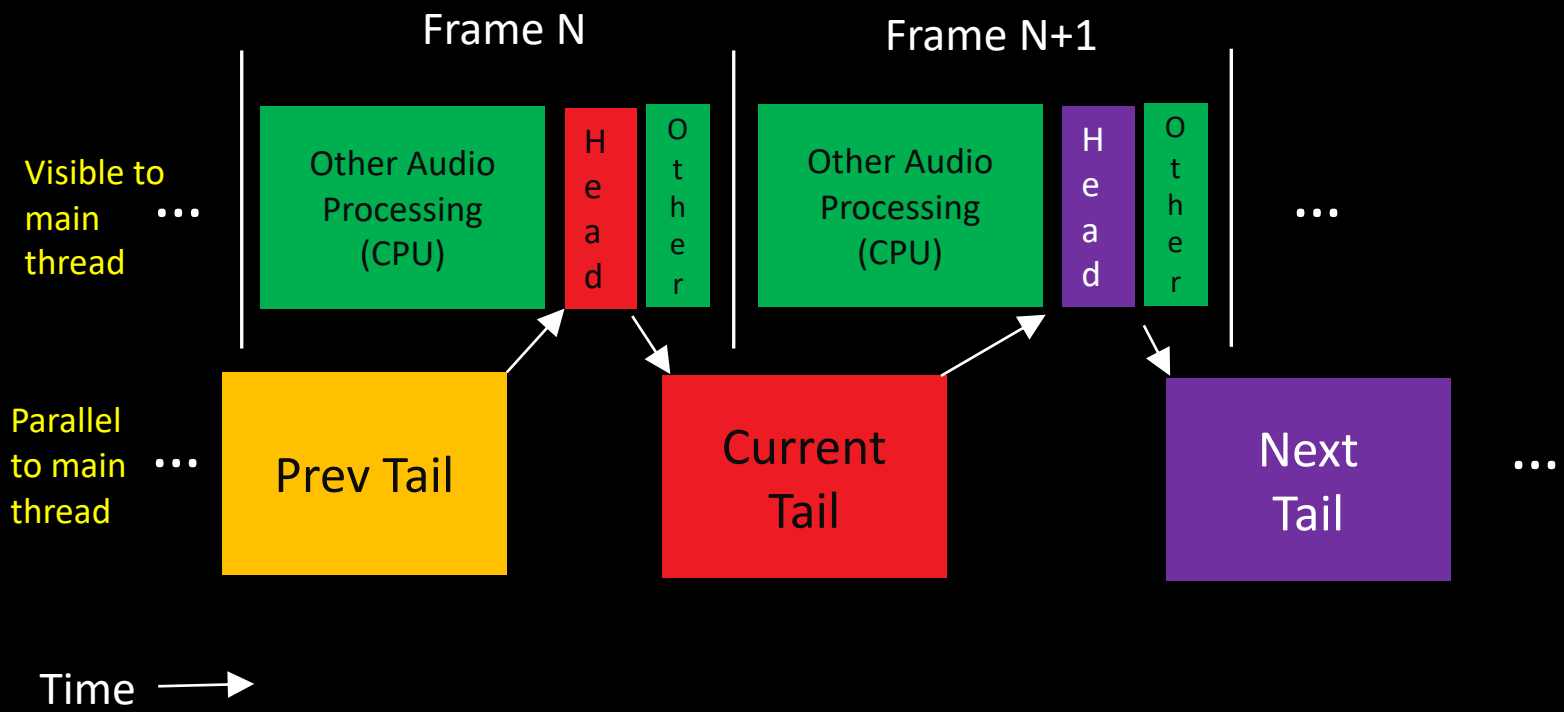


THIRD LEVEL: HEAD-TAIL CONVOLUTION PROCESSING

The **head-tail method** removes most of the work from the critical CPU thread and runs it in parallel.

Head and Tail blocks run on GPU. Other processing can run on CPU.

Process only the short "head" portion in critical path, and process the longer "tail" in- between audio buffer process calls.



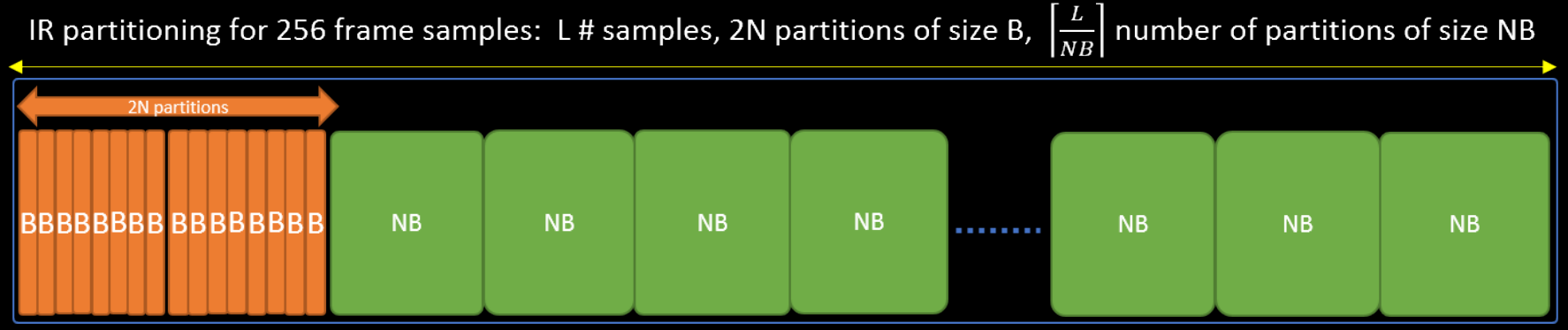
*Head performs point-wise multiply and then adds pre-calculated history from tail

FOURTH LEVEL: NON-UNIFORM HEAD-TAIL CONVOLUTION PROCESSING

With non-uniform head-tail, we get two key benefits:

1) Reduce the length of the "head" process that is visible to the main thread

2) Optimize the tail computation. Factor of "N" fewer MACs required – only process part of the history per tail



Head calculation (one "B" block).

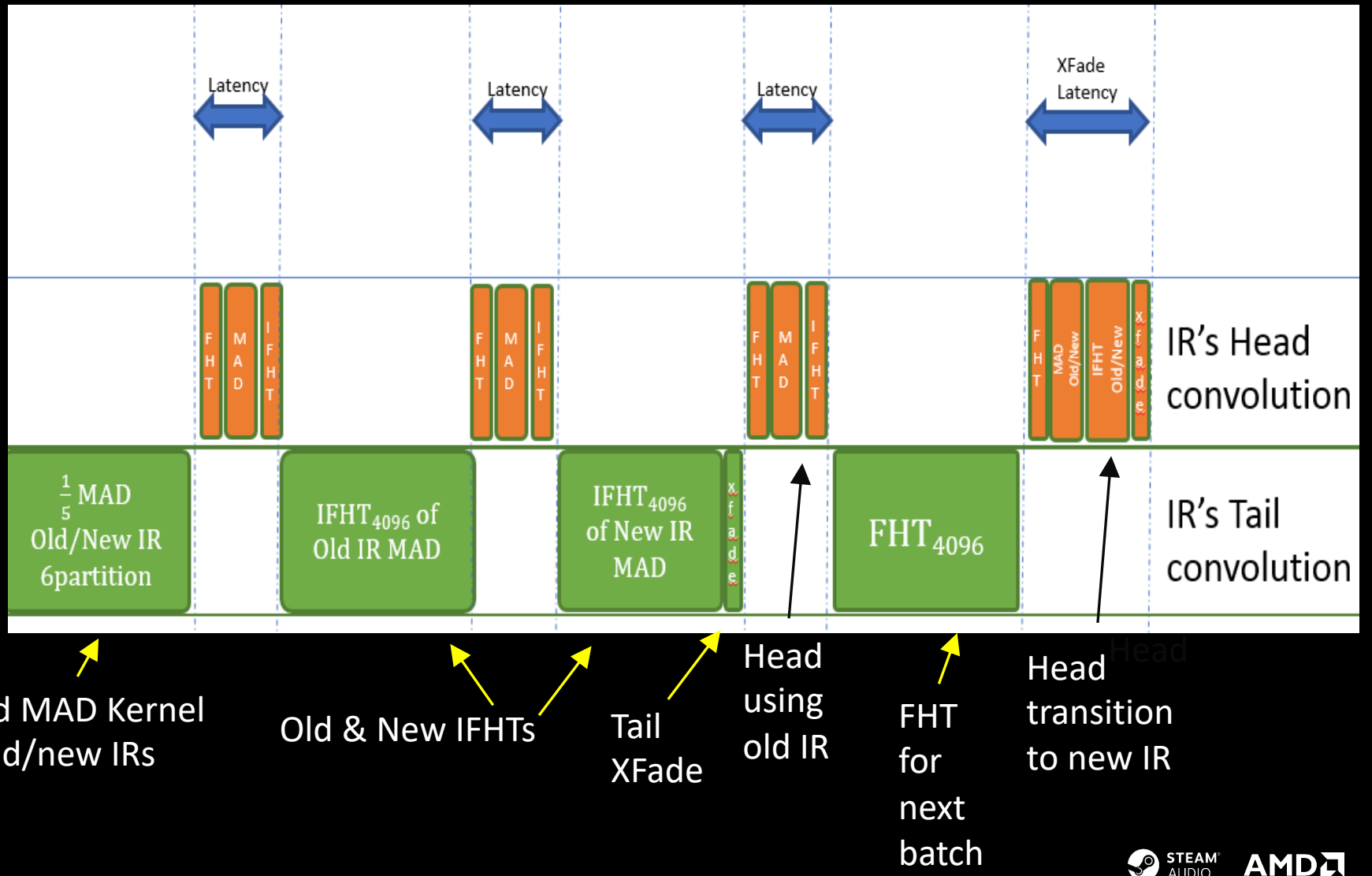
- Same as for uniform head-tail, but we can use a shorter frame size without affecting the tail calculation

Tail calculation (as before, on parallel GPU process):

- Tail's multiply-accumulate computation is roughly ***N times less*** compared to uniform head-tail, as a result, can support much longer IRs

TRANSITIONING TO A NEW IMPULSE RESPONSE

- To update to a new IR kernel:
  1) Compute the audio buffer output for both the old and the new IR
  2) Crossfade to the new filter
- With head-tail, the CPU-visible IR update overhead is very small



STABILITY DEMO: STEAM AUDIO WITH 3DMARK® TIME SPY TOGETHER ON THE SAME GPU

STEAM® AUDIO

TrueAudio Next

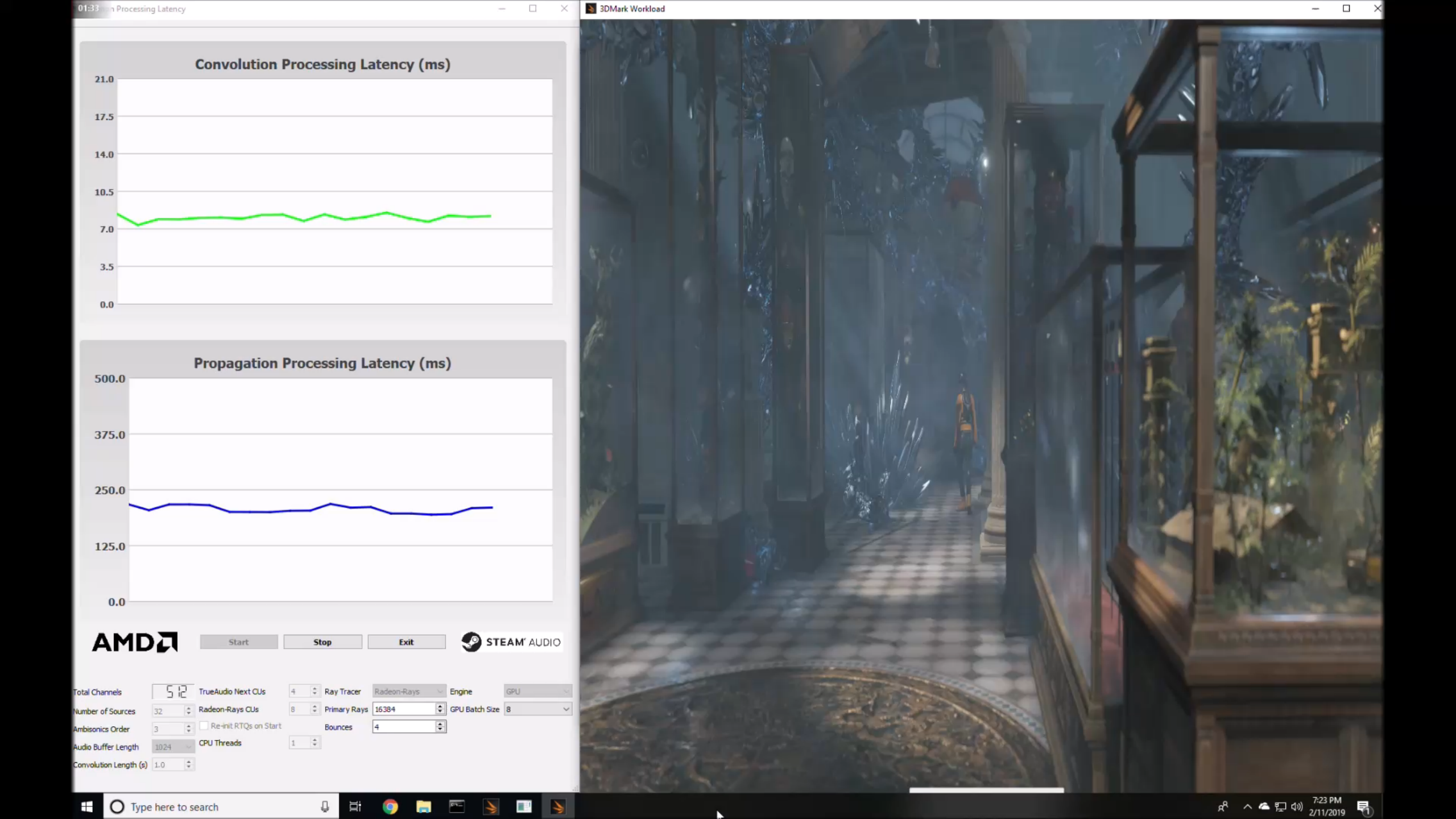
AMD RADEON Rays

STEAM® AUDIO AMD

WHAT THE VIDEO SHOWS

- Captured using Radeon™ ReLive Application
- Left side of screen: Steam Audio workload with Dual RTQ acceleration
  - Uses a total of 12 reserved CUs in 2 RTQs
  - Top graph: Worst-case convolution latency with settings:
    - 512 1-second time-varying convolution channels, rendered as 32 3rd-order Ambisonics
    - 4 CUs running TrueAudio Next
    - Buffer length is 20.8 ms, SR 48 kHz
    - Convolution to take less than ½ of the available time
  - Bottom Graph: Propagation update time using:
    - 8 CUs for Radeon Rays + additional OpenCL code to calculate & transform impulse responses
    - 16k Rays, 4 bounces, Sibenik Cathedral scene simulation
- Right side: 3DMark® Time Spy Stress running in window, other settings default*
  - Uses the remaining balance of 52 CUs
- System: Radeon™ RX Vega (64 CUs, 8G VRAM), default Radeon™ Software Adrenalin driver settings, AMD Ryzen™ 7 1800X with 32G DRAM, Windows 10 64b

*See Endnotes for test details.


Convolution Processing Latency (ms)
21.0
17.5
14.0
10.5
7.0
3.5
0.0
Propagation Processing Latency (ms)
500.0
375.0
250.0
125.0
0.0
AMD
Start
Stop
Exit
STEAM AUDIO
Total Channels
TrueAudio Next CUs
Ray Tracer
Radeon-Rays
Engine
GPU
Number of Sources
Radeon-Rays CUs
Primary Rays
16384
GPU Batch Size
Ambisonics Order
Re-init RTQs on Start
Bounces
Audio Buffer Length
CPU Threads
Convolution Length (s)
3DMark Workload
Type here to search
7:23 PM
2/11/2019

BUT WHAT ABOUT MY FRAME RATE?

- Performed test with 3DMark® Time Spy trials measured on Radeon™ RX Vega 64 / Ryzen™ 1800X system (average of 3 trials each scenario). Used default full-screen settings
- Test Procedure:
  1) Measure baseline score of system three times, average the results
  2) Repeat three measurements with Steam Audio performance workload (as shown in video) running simultaneously, average the results
  3) Compare % reduction in average score with % of CUs used for audio
- Results:
  - % of CUs used = 12/64 = 18.75%
  - % Reduction of average Time Spy score = 12.01%*
  - 12.01% < 18.75%

*See claim details in endnotes

STEAM AUDIO AND TRUEAUDIO NEXT USED IN INTERNATIONALLY-ACCLAIMED VR EXPERIENCE

"We implemented Steam Audio with TrueAudio Next in our VR experience ‘Made This Way: Redefining Masculinity’ and it truly changed the way people interacted with the testimonials of our subjects in the project. As creators this was incredibly important for us as we were telling the stories of marginalized people and therefore it was critical for their voice to be heard. Without a doubt Steam Audio with TrueAudio Next gave us this power to unlock the empathy of our project and connect the viewer to the story in a very deep and significant way."

-- **Elli Raynai**, co-director and co-producer of “Made this Way: Redefining Masculinity” which has been nominated for a 2019 Canadian Screen Award for Best Immersive Experience – Non-Fiction, and was an Official Selection at the 75th Venice International Film Festival.

MEDIA LAB OFC

75 MOSTRA INTERNAZIONALE D'ARTE CINEMATOGRAFICA LA BIENNALE DI VENEZIA 2018 Official Selection

CANADIAN SCREEN AWARDS 2019 PRIX ÉCRANS CANADIENS

NOMINEE CANADIAN SCREEN AWARDS 2019

CONCLUSION

STEAM® AUDIO

TrueAudio Next

AMD RADEON Rays

STEAM® AUDIO AMD

WITH STEAM AUDIO AND AMD RADEON, GREAT GAME AUDIO ON THE GPU IS REAL … AND REAL-TIME

- Supported on:

AMD Radeon™ RX 470
AMD Radeon™ RX 570
AMD Radeon™ RX 590
AMD Radeon™ R9 Fury X
AMD Radeon™ RX Pro Duo (Fiji)
AMD Radeon™ RXVega 56
AMD Radeon™ VII
AMD Radeon™ Pro WX 8200

AMD Radeon™ RX 480
AMD Radeon™ RX 580
AMD Radeon™ R9 Fury
AMD Radeon™ RX Pro Duo (Polaris)
AMD Radeon™ Vega Frontier Edition
AMD Radeon™ RX Vega 64
AMD Radeon™ Pro WX 7100
AMD Radeon™ Pro WX 9100

- Use current AMD Radeon™ Adrenalin drivers: https://www.amd.com/en/support
- Thanks to:
  - AMD: Gabor Sines, Geoffrey Park, Seyedreza Najafi, Peter Cao
  - Valve: Anish Chandak, Clinton Freeman

- Additional links:
  https://valvesoftware.github.io/steam-audio/
  https://gpuopen.com/gaming-product/true-audio-next/
  https://gpuopen.com/gaming-product/Radeon-Rays/

ENDNOTES

Using AMD Resource Reservation of 12 CUs, accelerating a test workload of Steam Audio with TrueAudio Next and Radeon-Rays running 512 channels of convolution and 16k rays/4 bounces for 32 3rd-order Ambisonics sources, the reduction of the Time Spy score from the system's baseline score when running the concurrent audio workload was consistently less than the % of total CUs reserved (12/64). using Radeon™ Software Adrenalin 2019 Edition on the Radeon™ RX Vega graphics card with Radeon™ Software Adrenalin Edition 18.12.3.0 at 1920x1080 (1080p). Testing conducted at AMD Santa Clara as of January 25th, 2019 on the 8GB Radeon™ RX 580, on a test system comprising of Ryzen 7 1800X CPU (3.6 GHz), 32GB DDR4-2128 MHz system memory, and Windows 10 x64. PC manufacturers may vary configurations, yielding different results. With default settings on Time Spy at 1920x1080, Radeon™ RX 580 scored with Radeon Software Adrenalin Edition 18.12.3, whereas the Radeon™ RX 580 scored 76.7 FPS with Radeon Software Adrenaiin 2019 Edition 19.1.1. Comparing software versions, Radeon Software Adrenalin 2019 Edition 19.1.1 has 4% faster performance in a popular game. Performance may vary based on use of latest drivers.

Using AMD Resource Reservation of 12 CUs, accelerating a test workload of Steam Audio with TrueAudio Next and Radeon-Rays running 512 channels of convolution and 16k rays/4 bounces for 32 3rd-order Ambisonics sources, Time Spy Stress achieved a consistent passing score, using Radeon™ Software Adrenalin 2019 Edition on the Radeon™ RX Vega graphics card than with Radeon™ Software Adrenalin Edition 18.12.3.0 at 1920x1080 (1080p). Testing conducted at AMD Santa Clara as of January 25th, 2019 on the 8GB Radeon™ RX 580, on a test system comprising of Ryzen 7 1800X CPU (3.6 GHz), 32GB DDR4-2128 MHz system memory, and Windows 10 x64. PC manufacturers may vary configurations, yielding different results. With default settings on Time Spy at 1920x1080, Radeon™ RX 580 scored with Radeon Software Adrenalin Edition 18.12.3, whereas the Radeon™ RX 580 scored 76.7 FPS with Radeon Software Adrenalin 2019 Edition 19.1.1. Comparing software versions, Radeon Software Adrenalin 2019 Edition 19.1.1 has 4% faster performance in a popular game. Performance may vary based on use of latest drivers.

Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.