



RADEON RYZEN

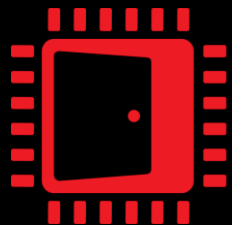


AMD

LUMINOUS
PRODUCTIONS®

BREAKING DOWN THE WORLD OF ATHIA: THE TECHNOLOGIES OF FORSPOKEN

AMD – LUMINOUS PRODUCTIONS



AMD
GPUOpen

GDC

AGENDA

- Forspoken presentation
- Partnership with AMD
- AMD technologies in the World of Athia
- Microsoft DirectStorage implementation, optimization & challenges
- Wrap up

FORSPOKEN

© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

FORSPOKEN

- PlayStation5 / PC (Microsoft Store, Steam, Epic Game Store)
- Releasing on October 11th, 2022, by SQUARE ENIX
- PC version
 - Many AMD graphics feature
 - Microsoft DirectStorage feature

FORSPOKEN™



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

PARTNERSHIP WITH AMD

- Started collaboration in July 2021
- AMD support
 - Ensure correctness of implementation
 - Support for code modification
 - Bug reports
 - Regular performance checks
- Benchmark to track performance evolution
- AMD DevTech serves as the interface to the whole AMD team

OVERVIEW OF AMD TECHNOLOGIES

- Luminous showed interest in our technologies for their game
- Forspoken is one of the games that use many AMD technologies in parallel
- Used on both PC and PlayStation5!

AMD
FidelityFX
Downsampler

AMD
FidelityFX
Ambient Occlusion

AMD
FidelityFX
Denoiser

AMD
FidelityFX
Screen Space Reflections

AMD
FidelityFX
Variable Shading

AMD
FidelityFX
Super Resolution

DISCLAIMER

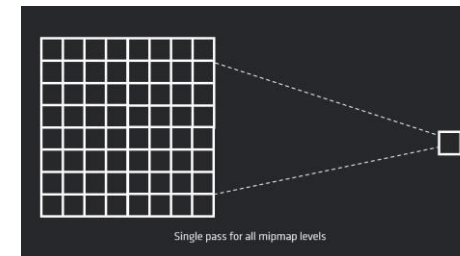
- What is presented here is a work in progress at the time of the recording

SINGLE PASS DOWNSAMPLER

- Very easy to integrate: a single dispatch to downsample to all mips
- Support for many downsample types (min, max, average, etc.)
- Completely customizable
- Performance improvements compared to multiple draw calls/dispatches
- Downsample used at multiple places in the frame:
 - depth buffer for screen space reflections
 - color buffer for water refraction
 - shadow texture for raytraced shadow mask
 - ...

AMD FidelityFX Downsampler

```
▼ Depth Mipmap
  ▼ 08 Depth min/max downsample
    ▼ SPD::ApplyDepthDownsampler0
      5348 ResourceBarrier()
      5349 ResourceBarrier()
      5354 ResourceBarrier()
      5359 Dispatch(60, 34, 1)
```



COMBINED ADAPTIVE COMPUTE AMBIENT OCCLUSION

- CACAO is a very customizable SSAO solution
- Source code access → working together on a modified version of this algorithm that operates at a higher resolution to support Luminous Productions desired output (increased sharpness)
- Currently executed in async compute

AMD
FidelityFX
Ambient Occlusion

▼ 13 CACAO Filter

6986 ResourceBarrier()

6987 CmdBarrierPreSyncClear()

6988 ClearUnorderedAccessViewUint()

6989 CmdBarrierPostSyncClear()

▼ 13.1 Prepare downsampled depths, normals and mips

▼ 13.1.1 Prepare depths and mips

6990 Dispatch(480, 270, 1)

▼ 13.1.2 Prepare normals from input normals

6991 Dispatch(480, 270, 1)

▼ 13.3 SSAO generation

6992 ResourceBarrier()

6993 Dispatch(480, 270, 1)

6994 Dispatch(480, 270, 1)

6995 Dispatch(480, 270, 1)

6996 Dispatch(480, 270, 1)

▼ 13.4 Deinterleaved blur

6997 ResourceBarrier()

6998 Dispatch(62, 47, 1)

6999 Dispatch(62, 47, 1)

7000 Dispatch(62, 47, 1)

7001 Dispatch(62, 47, 1)

▼ 13.5 Create output

7002 ResourceBarrier()

7003 Dispatch(480, 270, 1)

Previous AO solution



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

RAYTRACED AMBIENT OCCLUSION & DENOISER

- RTAO is used in conjunction with CACAO to improve AO
- RTAO is done at lower resolution
 - Create a mask (distance based) where raytracing will be done
 - Trace rays
 - Denoise
- You can use AMD shadow denoiser for AO!
- Currently takes roughly 2.3ms @ 4K output resolution on a Radeon™ 6900XT

AMD
FidelityFX
Denoiser

```
▼ DrawMaskTexture(Shared)
    6975 ResourceBarrier()
    6976 Dispatch(480, 270, 1)
    6977 ResourceBarrier()
    6985 CmdBarrierBlitSync()
▼ 13 Raytrace AO
    ▼ Raytrace AO inline(StaticPacked)
        7005 ResourceBarrier()
        7006 Dispatch(480, 540, 1)
    ▼ DrawRaytrace Denoiser AMD FFX v1.1 Shadow
        ▼ PrepareShadowMask
            7007 ResourceBarrier()
            7008 Dispatch(120, 135, 1)
        ▼ TileClassification
            7009 ResourceBarrier()
            7010 Dispatch(480, 270, 1)
        ▼ FilterSoftShadows
            ▼ Pass0
                7011 ResourceBarrier()
                7012 Dispatch(480, 270, 1)
            ▼ Pass1
                7013 ResourceBarrier()
                7014 Dispatch(480, 270, 1)
            ▼ Pass2
                7015 ResourceBarrier()
                7016 Dispatch(480, 270, 1)
        ▼ BlendTexture
            7017 ResourceBarrier()
            7018 Dispatch(480, 270, 1)
```


RTAO mask



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

Denoised RTAO



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

CACAO only



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

STOCHASTIC SCREEN SPACE REFLECTIONS & DENOISER

- Classify pixels where ray marching is necessary
- Hierarchical depth buffer traversal (generated with SPD)
- Advanced denoising
- Partnership: customized effect for Luminous Engine
 - More efficient intersection algorithm
 - More temporally stable denoiser
 - Improved classification
- Heavily customized for Luminous Productions needs:
 - New environment map fetching
 - Incorrect occluder detection
 - ...

```
▼ 20 Stochastic screen-space reflection
5328 ResourceBarrier()
5329 CmdBarrierPreSyncClear()
5330 ClearUnorderedAccessViewUint()
5331 CmdBarrierPostSyncClear()
5332 CmdBarrierPreSyncClear()
5333 ClearUnorderedAccessViewUint()
5334 CmdBarrierPostSyncClear()
5335 CmdBarrierPreSyncClear()
5336 ClearUnorderedAccessViewUint()
5337 CmdBarrierPostSyncClear()
▼ 20.1.2.1 CS Downsample
  ▼ SPD::ApplyDepthDownsamplerWithInlineCopy0
    5338 ResourceBarrier()
    5339 ResourceBarrier()
    5340 Dispatch(60, 34, 1)
  ▼ 20.1.2.2 CS FFX DNSR ClassifyTiles
    5341 Dispatch(480, 270, 1)
  ▼ 20.1.2.3 CS FFX DNSR PrepareBlueNoise
    5342 Dispatch(16, 16, 1)
  ▼ 20.1.2.4 CS FFX SSSR PrepareIndirectArgs
    5343 ResourceBarrier()
    5344 Dispatch(1, 1, 1)
  ▼ 20.1.2.5 CS FFX SSSR Intersection
    5345 ResourceBarrier()
    5346 ExecuteIndirect()
  ▼ 20.1.2.6 CS FFX DNSR Reproject
    5347 ResourceBarrier()
    5348 Dispatch(480, 270, 1)
  ▼ 20.1.2.7 CS FFX DNSR Prefilter
    5349 ResourceBarrier()
    5350 Dispatch(480, 270, 1)
  ▼ 20.1.2.8 CS FFX DNSR Resolve Temporal
    5351 ResourceBarrier()
    5352 Dispatch(480, 270, 1)
  ▼ 20.1.2.9 Create Output
    5353 ResourceBarrier()
    5354 Dispatch(480, 270, 1)
```

AMD
FidelityFX
Screen Space Reflections

AMD
FidelityFX
Denoiser

STOCHASTIC SCREEN SPACE REFLECTIONS & DENOISER

- In SSSR sample:
 - Prefetch the cubemap in areas where SSR won't run (e.g., above threshold roughness)
 - Get SSR value and blend it with the environment map (cubemap) based on the confidence
 - Denoise all

$$Color = Denoise(lerp(cubemap, SSR, confidence))$$

- Limitations:
 - Only **one** environment map
 - Environment map is a **cubemap**

STOCHASTIC SCREEN SPACE REFLECTIONS & DENOISER

- In Luminous engine:
 - Multiple environment maps blended with the SSR value with a weight (1 pass per environment map)
 - Contains a *brdf* factor that **we don't want to denoise**

$$Color = \sum_{i=1}^n brdf_i \times lerp(environment_i, SSR, weight)$$

- How can we insert SSSR in their pipeline without modifying it too much?

STOCHASTIC SCREEN SPACE REFLECTIONS & DENOISER

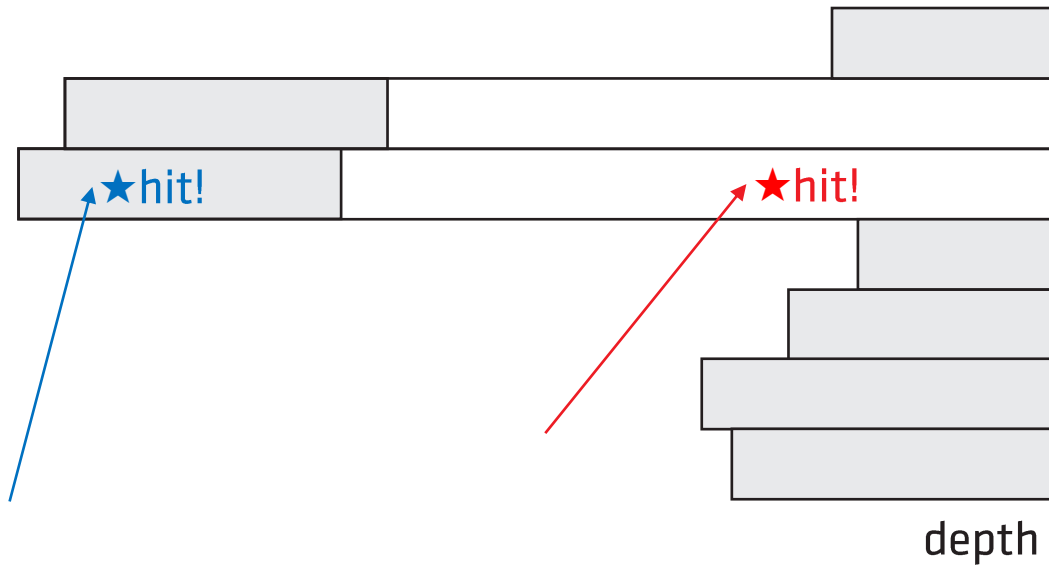
- Idea: denoise the confidence and use the current pipeline → create a custom version of SSSR
- Final version:
 - Stay close to Luminous Engine pipeline
 - Create a custom denoiser that also denoises the confidence

$$Color = \sum_{i=1}^n brdf_i \times lerp(environment_i, Denoise(SSR), Denoise(confidence))$$

- First modification of SSSR
 - Do not prefetch the cubemap where rays aren't needed → store a confidence of 0
- New version of the denoiser
 - Operates on color (float3) and confidence (float) → float4

STOCHASTIC SCREEN SPACE REFLECTIONS

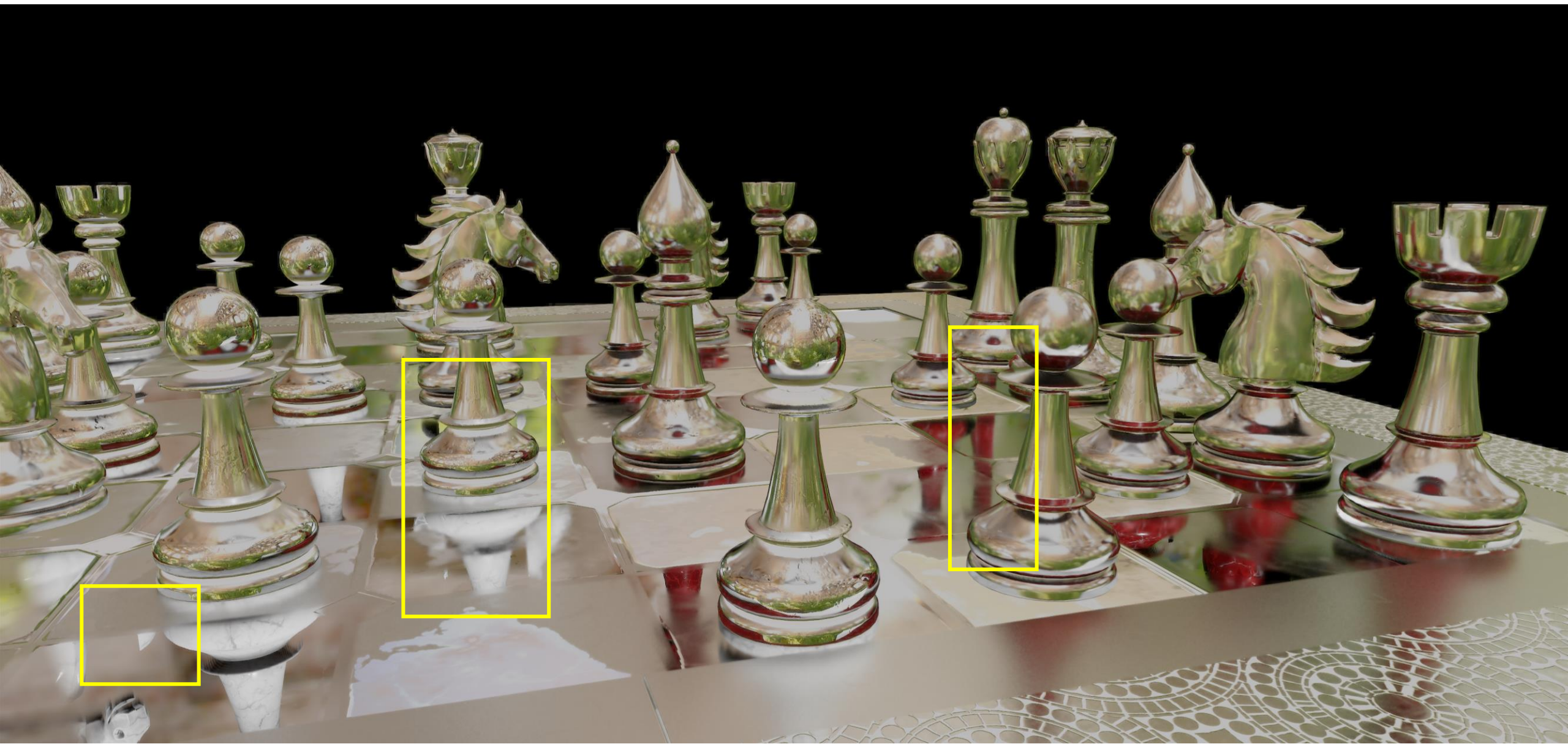
- SSSR stops when hitting an occluder



Default SSSR

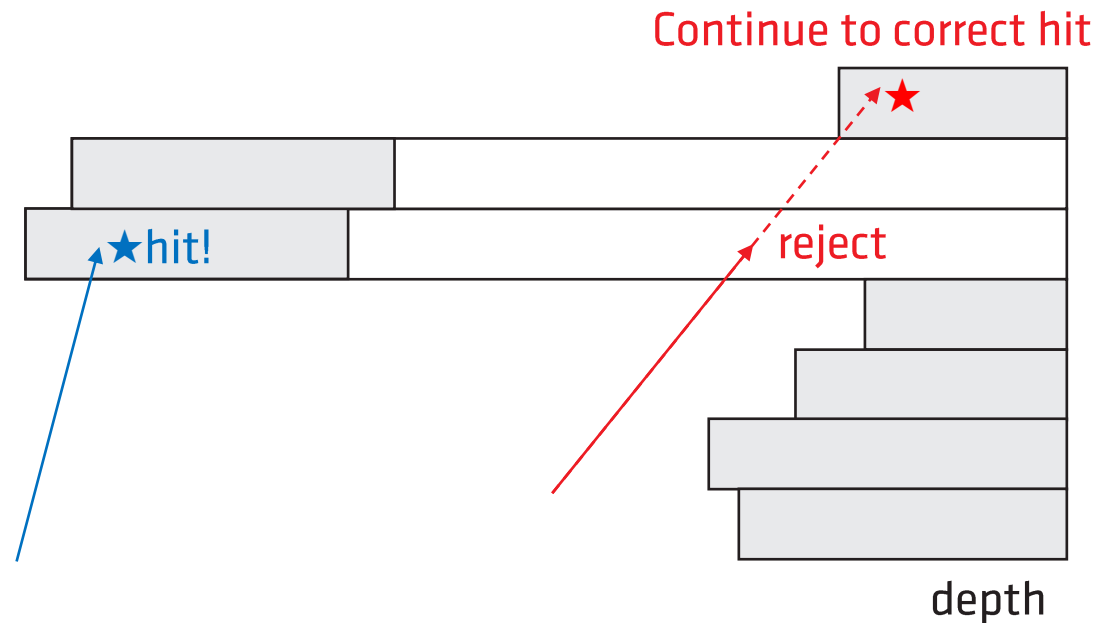


Default SSSR

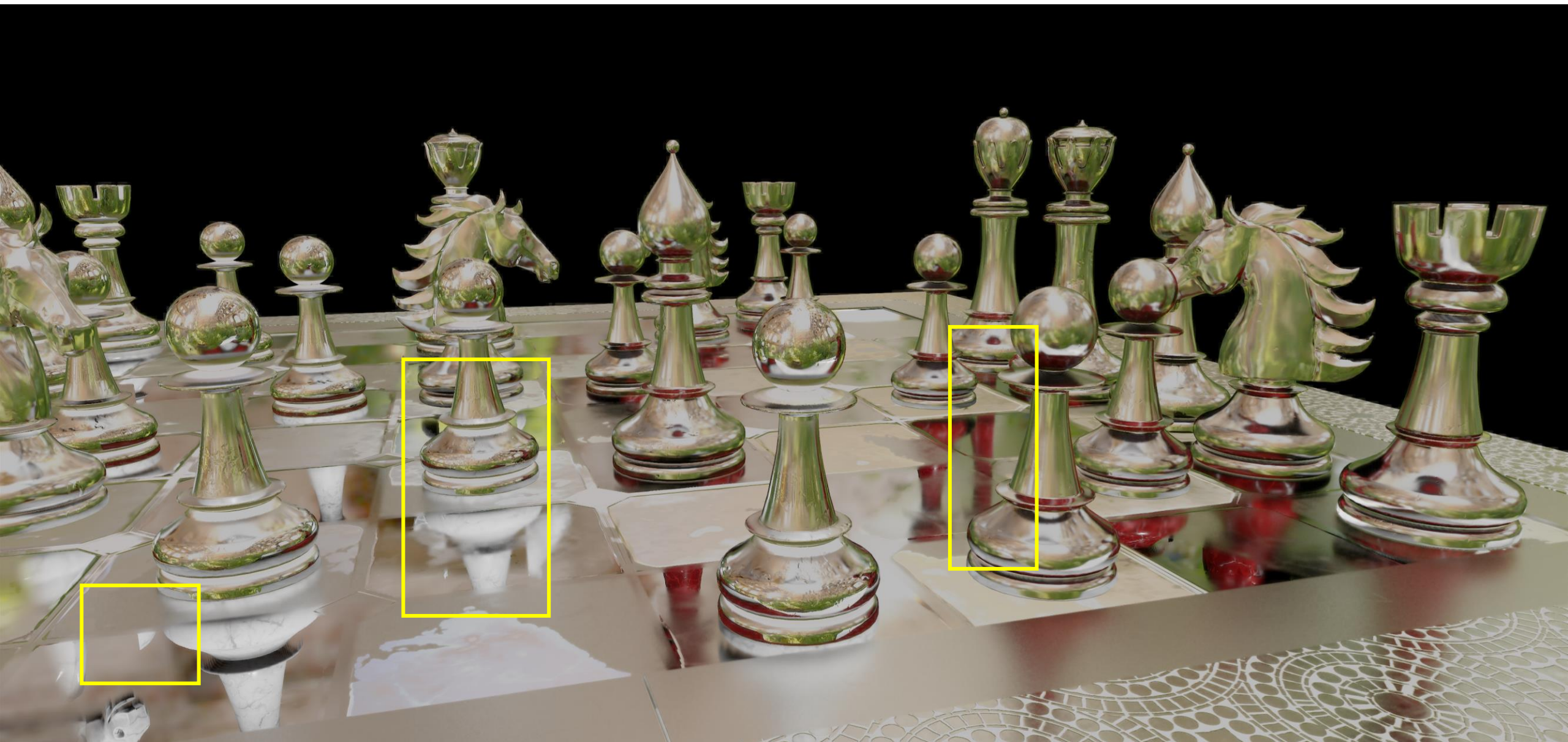


STOCHASTIC SCREEN SPACE REFLECTIONS

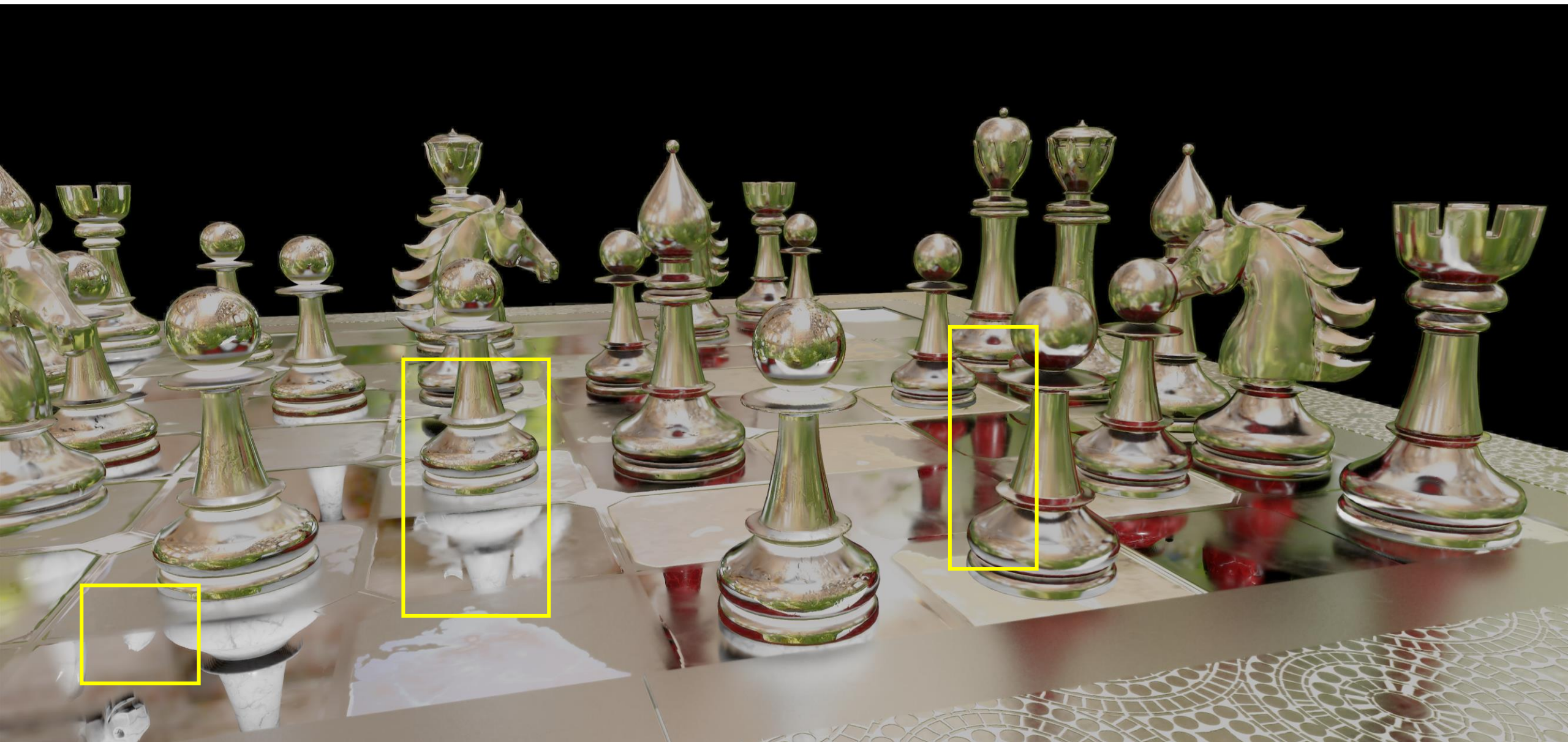
- When hitting an occluder, we can decide to accept the hit or pursue the ray marching (based on the distance between the ray and the occluder)
 - Entirely driven by parameters



Default SSSR



Modified SSSR



Luminous Engine's previous SSR



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

SSSR without occluder rejection



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

SSSR without occluder rejection



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

SSSR with occluder rejection



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

VARIABLE SHADING

- Takes advantage of the Variable Rate Shading feature (tier 2) present on AMD RDNA™ 2 architecture
 - Reduces the load on the pixel shader
- Uses the luminance variation within a tile from the reprojected previous frame to create the shading rate texture
- Used for lighting which is blended to the render target
 - Textures are at full resolution
→ no impact to the output quality

AMD
FidelityFX
Variable Shading

Variable Shading Off



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

Shading rate texture



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

Shading rate texture overlayed



Variable Shading On



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

VARIABLE SHADING

Off



On



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

HYBRID SHADOWS

- Only use raytracing where it matters (e.g., in the penumbra regions)
- Algorithm:
 - Create a mask to decide where it is allowed to trace a ray
 - Classification finds where a ray should be traced and where shadow map is enough
 - Trace rays
 - Denoise the result
 - Blend everything at the end
- Currently takes roughly 3.3ms @ 4K output resolution on a Radeon™ 6900XT (before optimization) on the following scene

```
✓ 14 Raytrace Directional light
  > DrawMaskTexture(Unique)
    7597 ResourceBarrier()
    7598 CmdBarrierPreSyncClear()
    7599 ClearUnorderedAccessViewUint()
    7600 CmdBarrierPostSyncClear()
  ✓ RaytraceHybridShadowsAMDv1.0 ClassifyByShadowDirMask
    7601 ResourceBarrier()
    7602 Dispatch(480, 540, 1)
  ✓ Raytrace Shadow inline(StaticPacked)
    7603 ResourceBarrier()
    7604 ResourceBarrier()
    7605 ExecuteIndirect()
  ✓ DrawRaytrace Denoiser AMD FFX v1.1 Shadow
    ✓ PrepareShadowMask
      7606 ResourceBarrier()
      7607 Dispatch(120, 135, 1)
    ✓ TileClassification
      7608 ResourceBarrier()
      7609 Dispatch(480, 270, 1)
    ✓ FilterSoftShadows
      ✓ Pass0
        7610 ResourceBarrier()
        7611 Dispatch(480, 270, 1)
      ✓ Pass1
        7612 ResourceBarrier()
        7613 Dispatch(480, 270, 1)
      ✓ Pass2
        7614 ResourceBarrier()
        7615 Dispatch(480, 270, 1)
  ✓ BlendTexture
    7616 ResourceBarrier()
    7617 Dispatch(480, 270, 1)
```


RT shadows Off



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

RT shadows On



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

RT shadows On



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

HYBRID SHADOWS

RT shadows Off



RT shadows On



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

RT shadows On



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

FIDELITYFX SUPER RESOLUTION 1.0

- Spatial image upscaling technique
- Easy to integrate in the Luminous Engine
 - Decoupled rendering/output resolution
 - MIP LOD Bias on the textures
 - 2 dispatches
- Gives more headroom for costly operations like raytracing
- Current gains in Forspoken:
 - 21% faster in ultra quality mode (x1.3 scale)
 - 26% faster in quality mode (x1.5 scale)

```
▼ MultiScreenWriteBack
  ▼ AMDSuperResFilter::Upscale
    7727 ResourceBarrier()
    7728 Dispatch(240, 135, 1)
  ▼ AMDRCasFilter::Apply
    7729 ResourceBarrier()
    7730 Dispatch(480, 270, 1)
```

AMD
FidelityFX
Super Resolution

FIDELITYFX SUPER RESOLUTION 2.0

- Brand new cutting-edge temporal image upscaling technique
 - Luminous engine supports:
 - Temporal Anti-Aliasing
 - Motion vectors buffer
 - Decoupled rendering and output resolution
- Fast integration (less than a week)
- Gives more headroom for costly operations like raytracing

AMD 
FidelityFX
Super Resolution 2.0

AMD FIDELITYFX

- Effects & documentation are available on GPUOpen
- The source code for all the effects is available under MIT licence!
- Don't hesitate to look into it and try them
- Contact your AMD DevTech engineer if you have any question



MICROSOFT DIRECTSTORAGE

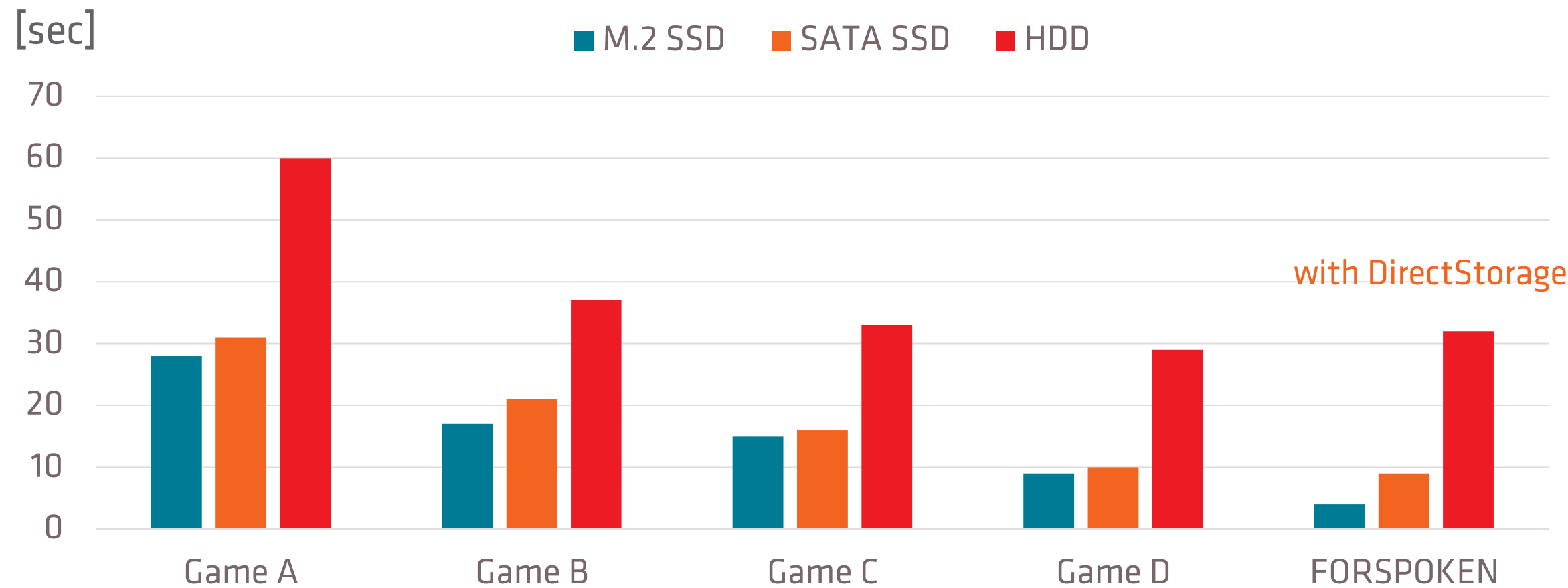
- A high-speed I/O API compatible with Windows 10/11
- Specifically optimized for Windows 11
- Effectively manages small files as well
- Incredibly fast NVMe M.2 SSD (> 5000MB/sec)
 - Many titles do not fully utilize its capability
 - CPU bottleneck issues
 - Most games have loading times of 10+ seconds
- Our game title, “FORSPOKEN,” supports DirectStorage
 - Loading time is only one second!



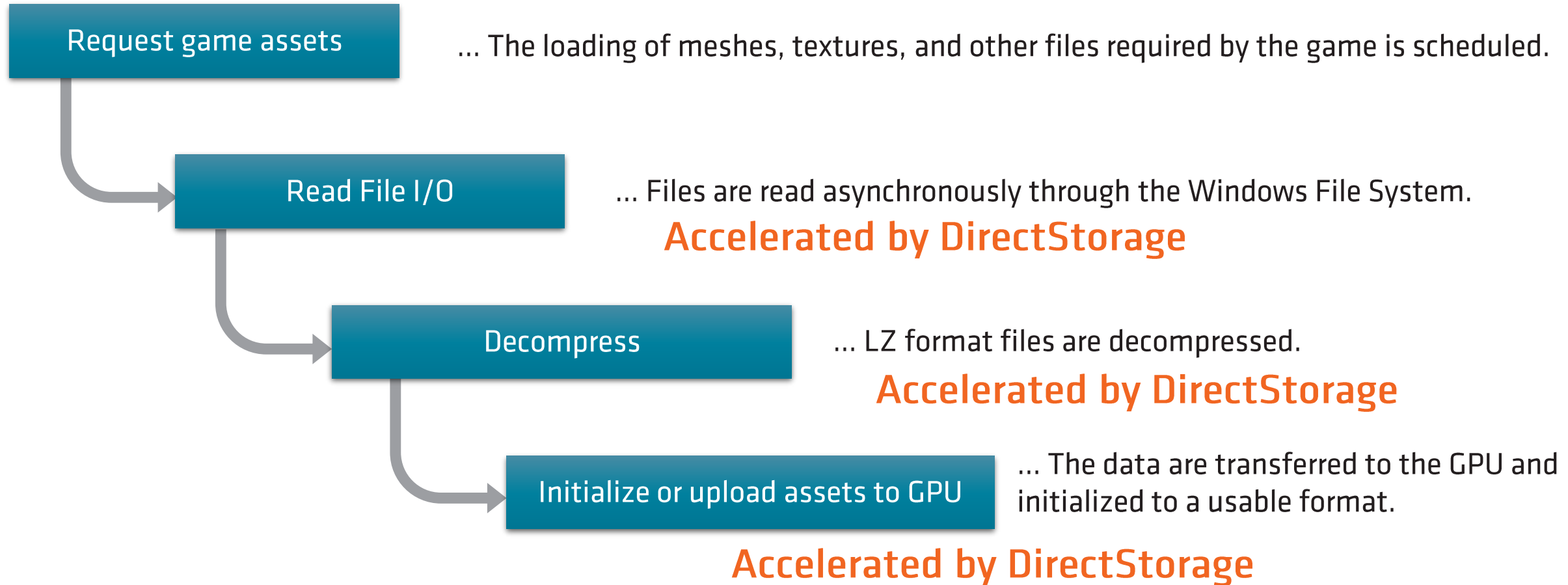
© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

PC GAME LOADING TIME

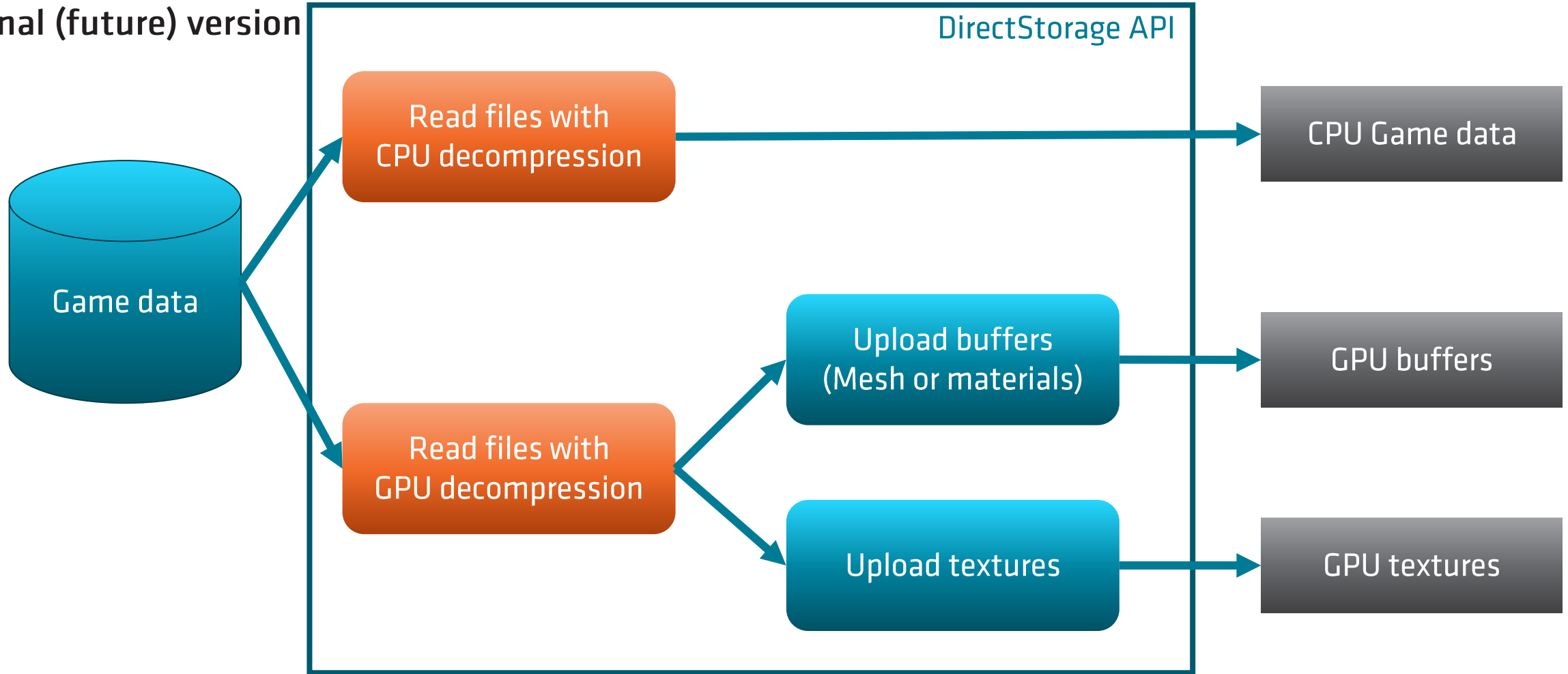


ASSET LOADING FLOW



DIRECTSTORAGE ARCHITECTURE

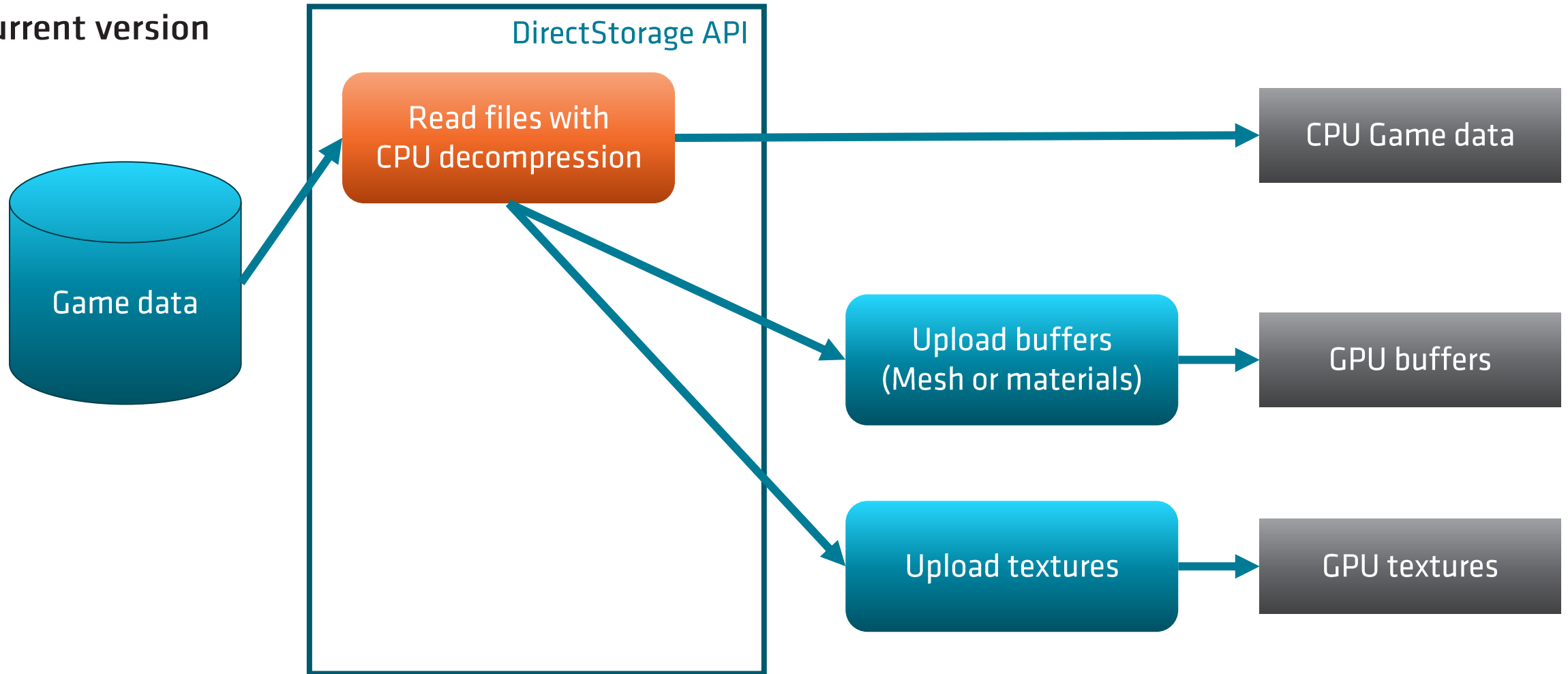
Final (future) version



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

DIRECTSTORAGE ARCHITECTURE

Current version



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

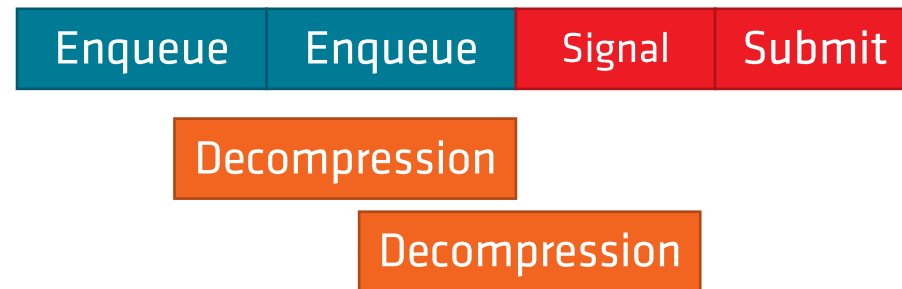
ARCHITECTURAL COMPARISON

- Win32 Asynchronous I/O
 - Each file read necessitates synchronization
 - ReadFile() with an OVERLAPPED option

<https://docs.microsoft.com/en-us/windows/win32/fileio/synchronous-and-asynchronous-i-o>



- DirectStorage
 - Multiple queues can be created and executed in parallel
 - Decompression support available
 - ID3D12Fence can be used for synchronization
 - Multiple read requests can be synchronized at once
 - Optimized towards asynchronous, streaming data transfers of file chunks from NVMe



DIRECTSTORAGE API PERFORMANCE

- Event scenes from the game



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

Scene 1

Amount of data to load: 20,000 files / 4.5GB



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

Scene 2

Amount of data to load: 25,000 files / 5.5GB

© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

DIRECTSTORAGE API PERFORMANCE



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

DIRECTSTORAGE API PERFORMANCE

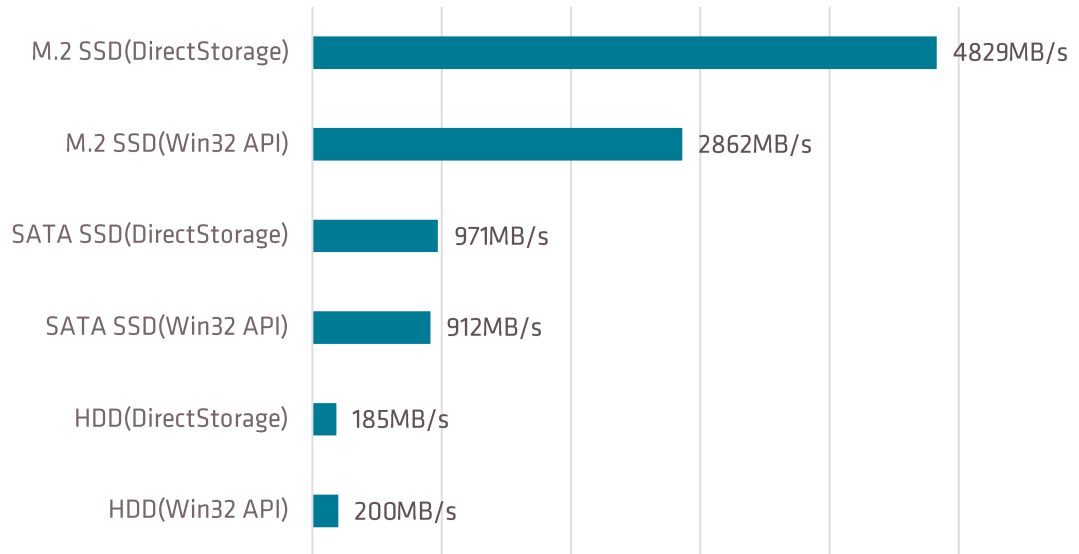


© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

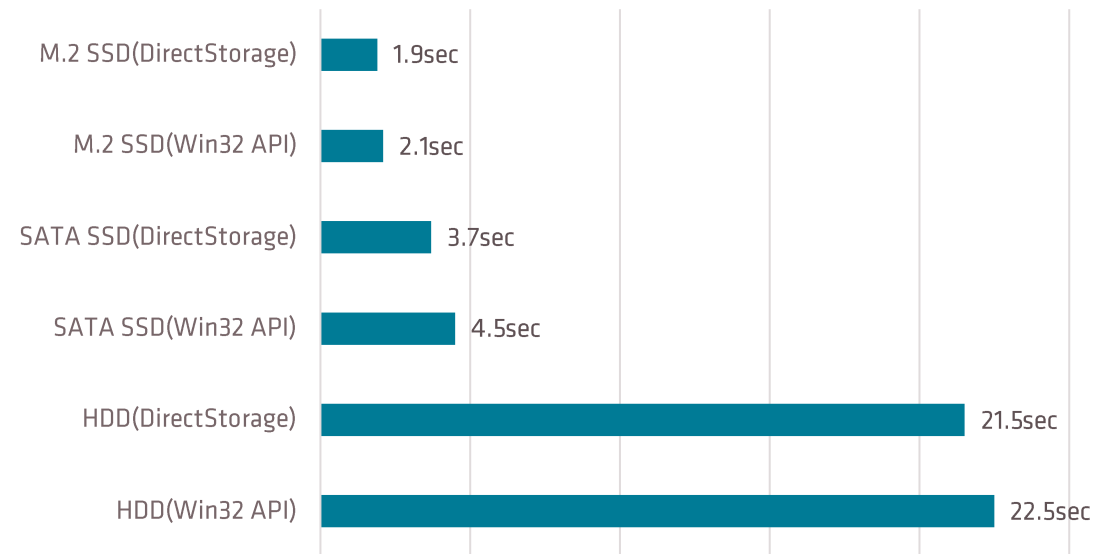
DIRECTSTORAGE PERFORMANCE

- M.2 SSD vs SATA SSD vs HDD (AMD Ryzen™ 9 5900X / RAM 64GB / Radeon™ 6800XT)

File IO speed (with decompression)



Actual loading time of the game scene



DirectStorage **4829MB/s** > win32 **2862MB/s**

DirectStorage **1.9sec** > win32 **2.1sec**

DIRECTSTORAGE X GAMING EXPERIENCE

- Playable game scenes



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

Scene 3

Amount of data to load: 18,000 files / 3.5GB



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

Scene 4

Amount of data to load: 20,000 files / 4.2GB

© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

DIRECTSTORAGE X GAMING EXPERIENCE



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

DIRECTSTORAGE X GAMING EXPERIENCE



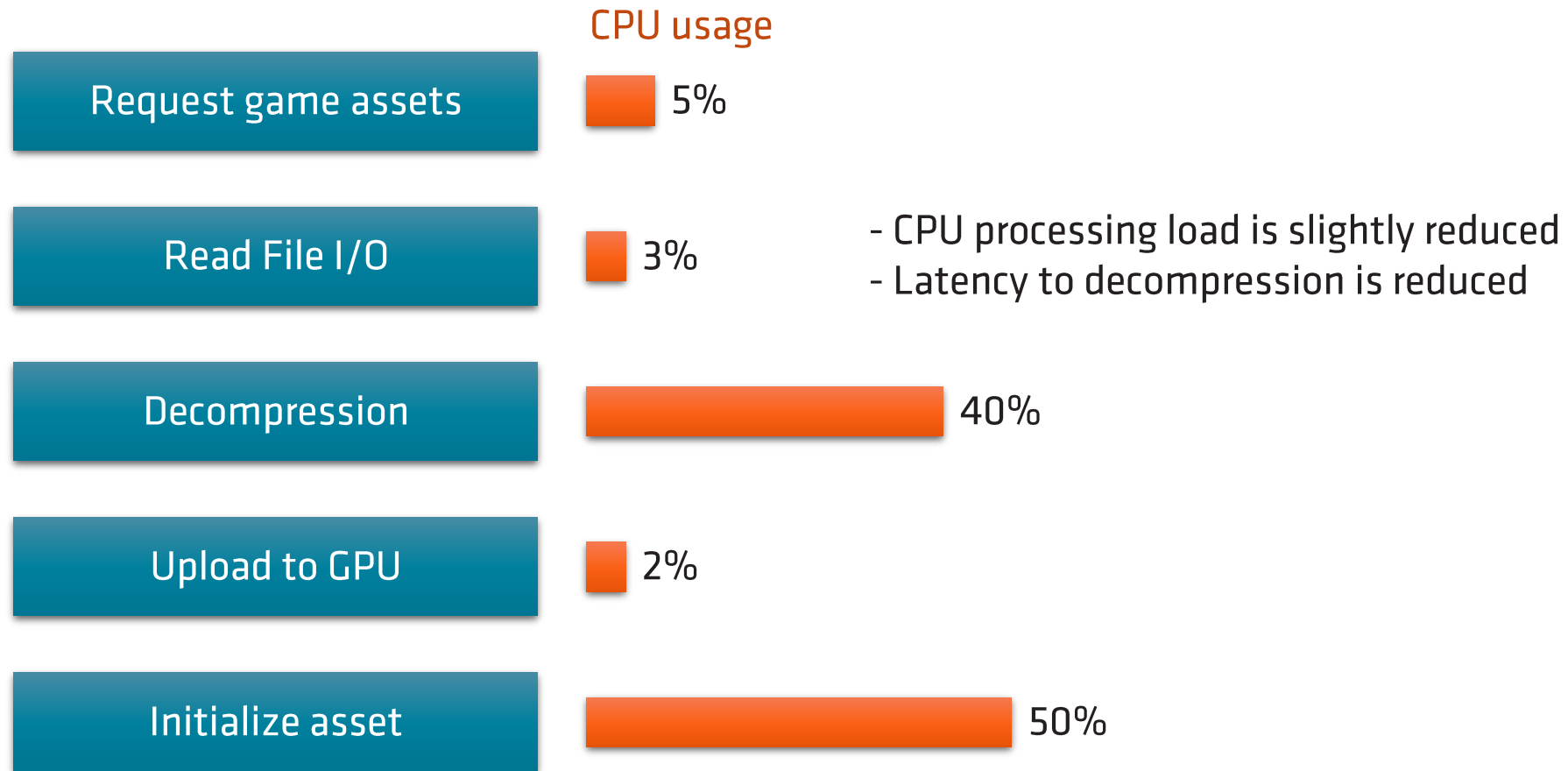
© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

DIRECTSTORAGE ANALYTICS BY PIX



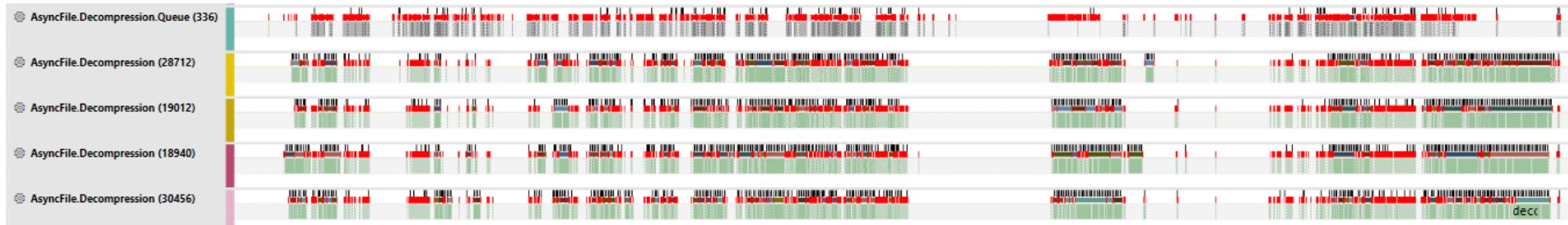
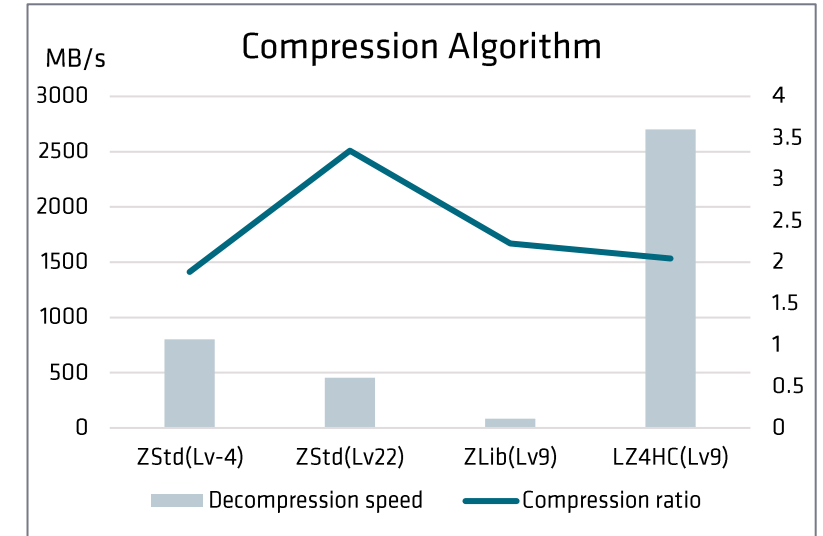
© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

DIRECTSTORAGE ANALYTICS



DIRECT STORAGE DECOMPRESSION

- Compression
 - Files are pre-compressed using the LZ4HC algorithm
 - LZ4 is an extremely fast decompression algorithm
 - The current compression ratio is approx. 2.0
- Implemented using IDStorageCustomDecompressionQueue
 - Enables the implementation of your own decompression processing
- Adjust the number of threads in decompression
 - Too many threads will result in blocking other threads



DIRECTSTORAGE MULTI-QUEUE

- Direct storage supports multiple queues
 - More efficient to prioritize files:
 - Level scenes are critical
 - Texture and mesh streaming data are of lower priority
- Four IDStorageQueue were used
 - 2%-5% performance improvement



GAME CPU BOTTLENECKS VS DIRECTSTORAGE I/O

- Challenges to address in order to attain one-second loading time:
 - CPU time for smaller assets must be minimized
 - mimalloc is used for multithreading heap allocation
- File I/O often has to wait for the initialization process to complete
 - No problem as long as file I/O takes longer than five seconds
 - Attaining one-second loading time requires further optimization
 - GPU decompression support is expected to bring some improvement



DIRECTSTORAGE - NEXT ACTIONS

- Our goal of a one-second loading time was attained in some scenes
 - The performance of the M.2 SSD is also sufficient
- While DirectStorage's performance is quite high, it is yet to be fully utilized
 - Many non-I/O CPU bottleneck issues still exist
- GPU Decompression support in the future



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

THANK YOU!



© 2022 Luminous Productions Co., Ltd. All Rights Reserved.

ACKNOWLEDGEMENTS

- The Luminous Productions team
 - Takeshi Aramaki
 - Teppei Ono
 - Kosei Itou
 - Yasunobu Itou
 - Shao Ti Lee
 - ...
- The AMD Game Engineering team
 - Jason Lacroix
 - Denisse Bock
 - Anton Schreiner
 - Jay Fraser
 - David Ziman
 - ...

DISCLAIMER

The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18

© 2022 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD Arrow logo, EPYC, Radeon, RDNA, Ryzen and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective owners.

