

# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

Ramin Zaghi  
Senior Applications Engineer  
Partner Enablement Group, ARM

24 Sep 2015

# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

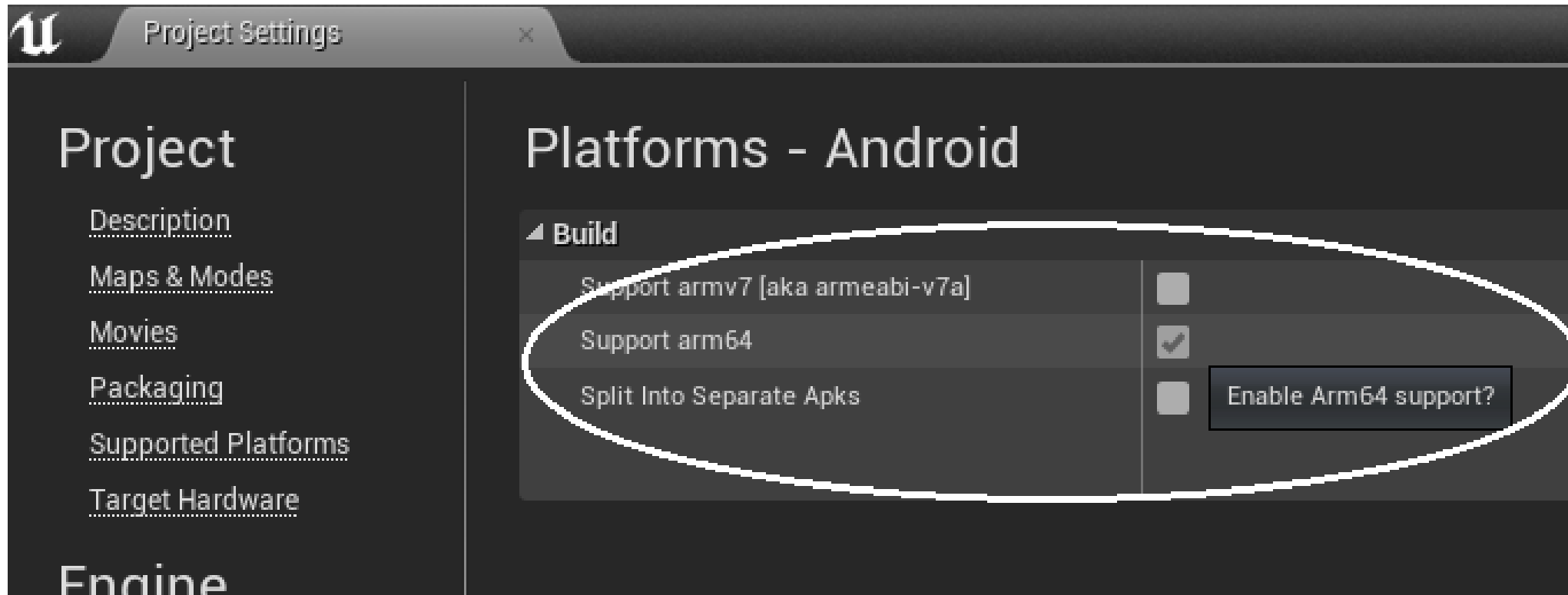
- Unreal Engine 4 source was made available to the public at GDC 2014 and is now free
- We had been working with *Epic*<sup>®</sup> for several months on and off to contribute
- Several patches created by the *Ecosystem* team in ARM<sup>®</sup> *Media-processing Group*
- I will show you three of them which we hope to see in the upstream codebase soon

# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

- The patches that we created for Unreal Engine 4 are to add support for
  1. ARM 64-bit Architecture (AArch64) targets
  2. Adaptive Scalable Texture Compression (ASTC) format
  3. Pixel Local Storage (PLS) useful for certain types of post processing effects

# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

- ARM 64-bit Architecture (AArch64) target support

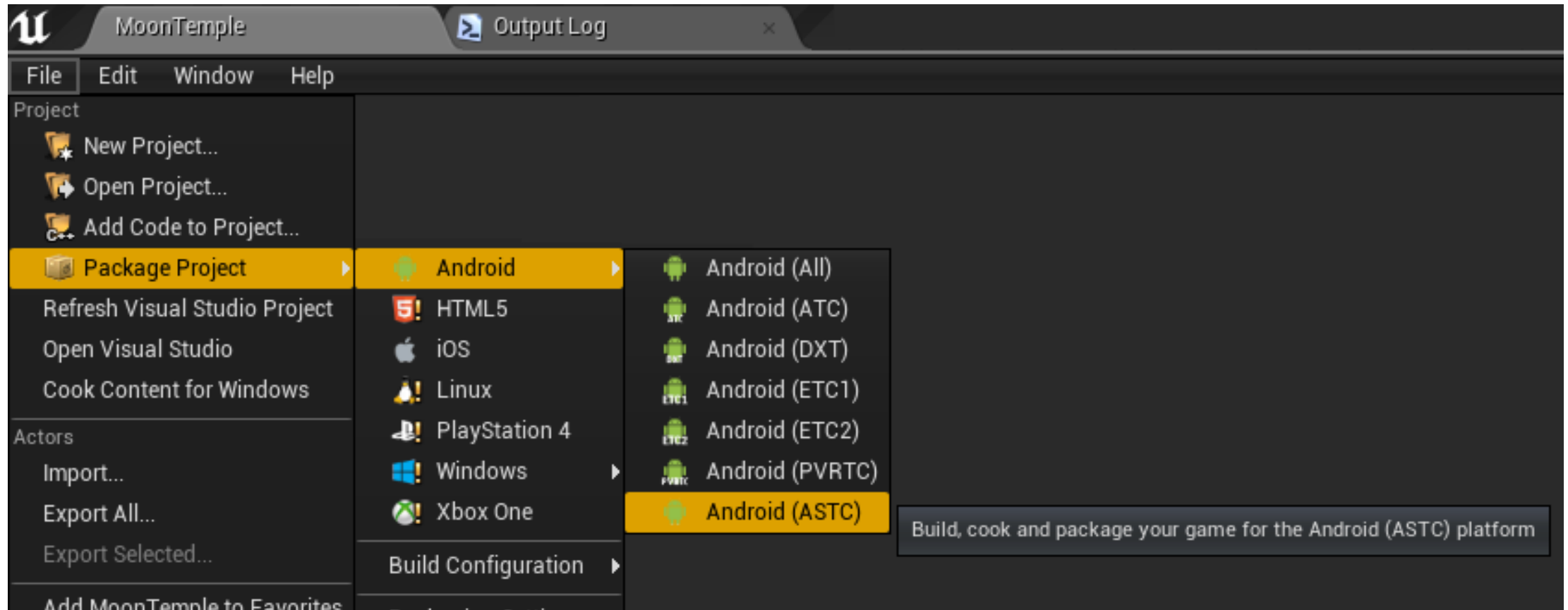


# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

- Most of the 64-bit porting involved making sure existing code built fine for AArch64
- Well written code meant no changes to data types or anything of that nature
- NDK version *r10c* (from October 2014) was the first to support 64-bit
- It also included GCC 4.9 which is currently the latest major release of the toolchain
- See [https://developer.android.com/ndk/downloads/revision\\_history.html](https://developer.android.com/ndk/downloads/revision_history.html)
- Results: 8% uplift in the SunTemple FPS, just from compiling for 64-bit

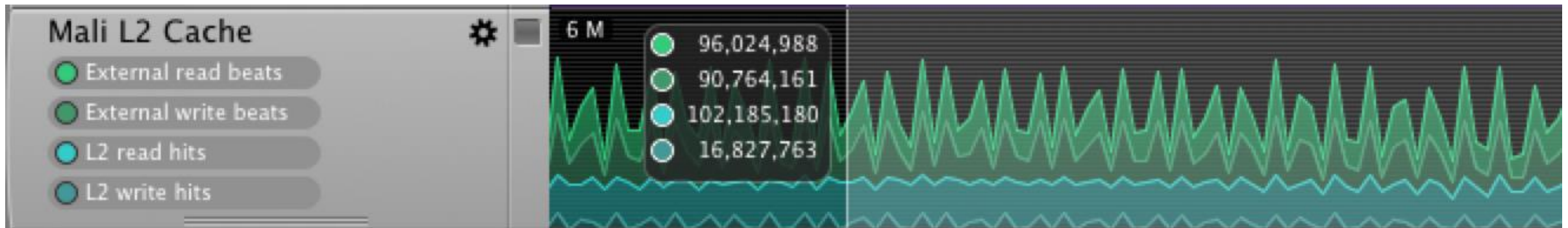
# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

- Adaptive Scalable Texture Compression (ASTC) format



# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

- Streamline tool, part of ARM Development Studio 5 (DS-5)
  - to learn more <https://ds.arm.com>
- Capture CPU and GPU parameters during runtime for analysis
- ASTC requires less memory, so bandwidth use should drop
  - We should see that reflected in L2 cache external R+W beats
  - Example image from Streamline



# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

- Result of Streamline L2 counters:
- ETC2 over 30s: **1.29 GB/s**
- ASTC over same 30s: **0.98 GB/s**
- **24.4%** less bandwidth used per frame
- ...and the *.obb* file using ASTC is 12% smaller than using ETC2 (179MB versus 203MB)



# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

- Pixel Local Storage (PLS) used for some post processing effects



Before

# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

- Pixel Local Storage (PLS) used for some post processing effects



After

# DEMO

# Porting Unreal<sup>®</sup> Engine 4 to ARMv8

- To learn more please watch the following video from GDC 2015

*“Unreal Engine 4: Mobile Graphics on ARM CPU and GPU Architecture”*

<http://malideveloper.arm.com/events/game-developers-conference-gdc-2015/>

- Note the slides for the video are also available through the same link

# Questions?

# Thank You

*The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Any other marks featured may be trademarks of their respective owners*