

Опыт реализации текстурного компрессора на гетерогенных высокопроизводительных системах

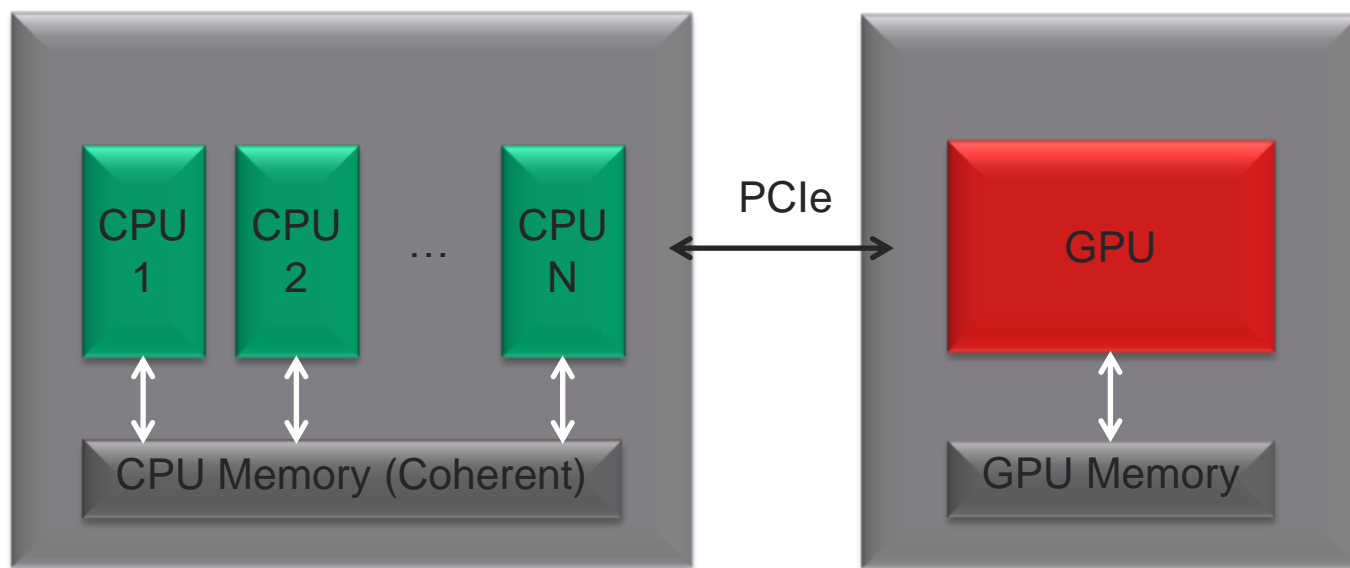
**ИЛЬЯ ПЕРМИНОВ
ТИМУР ПАЛТАШЕВ**



HETEROGENEOUS COMPUTING

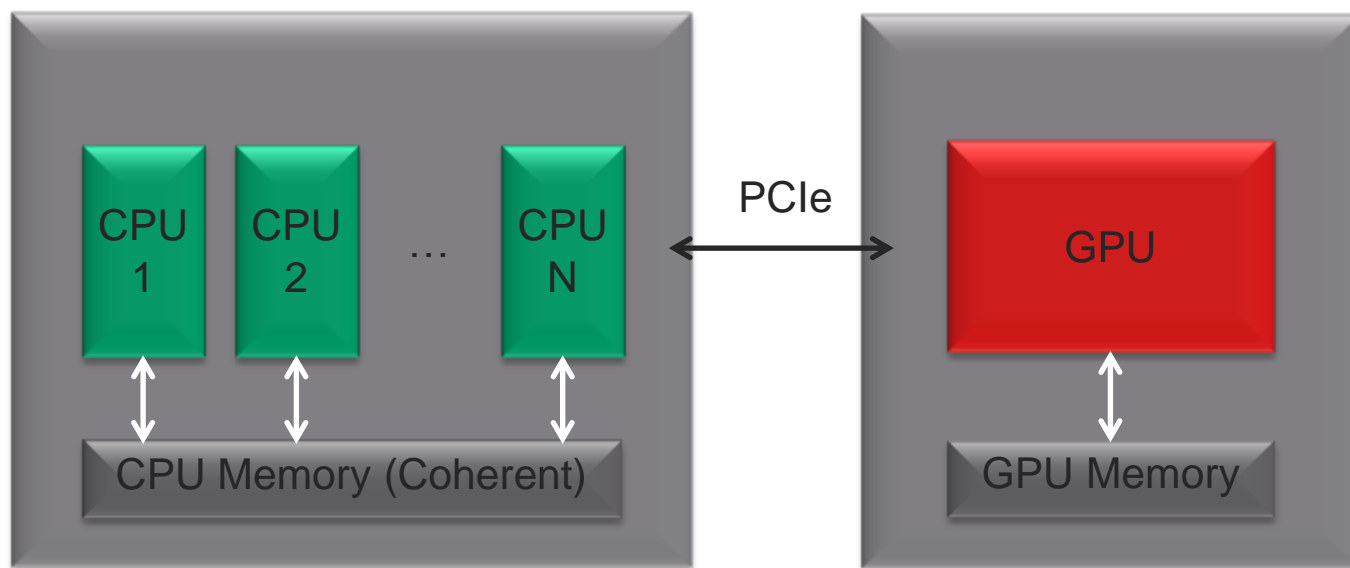


DISCRETE CPU & GPU



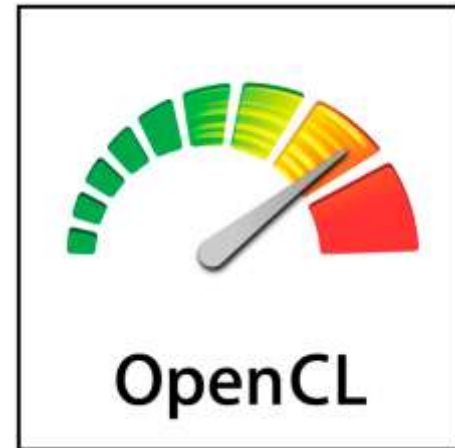
HSA = Heterogeneous System Architecture

DISCRETE CPU & GPU



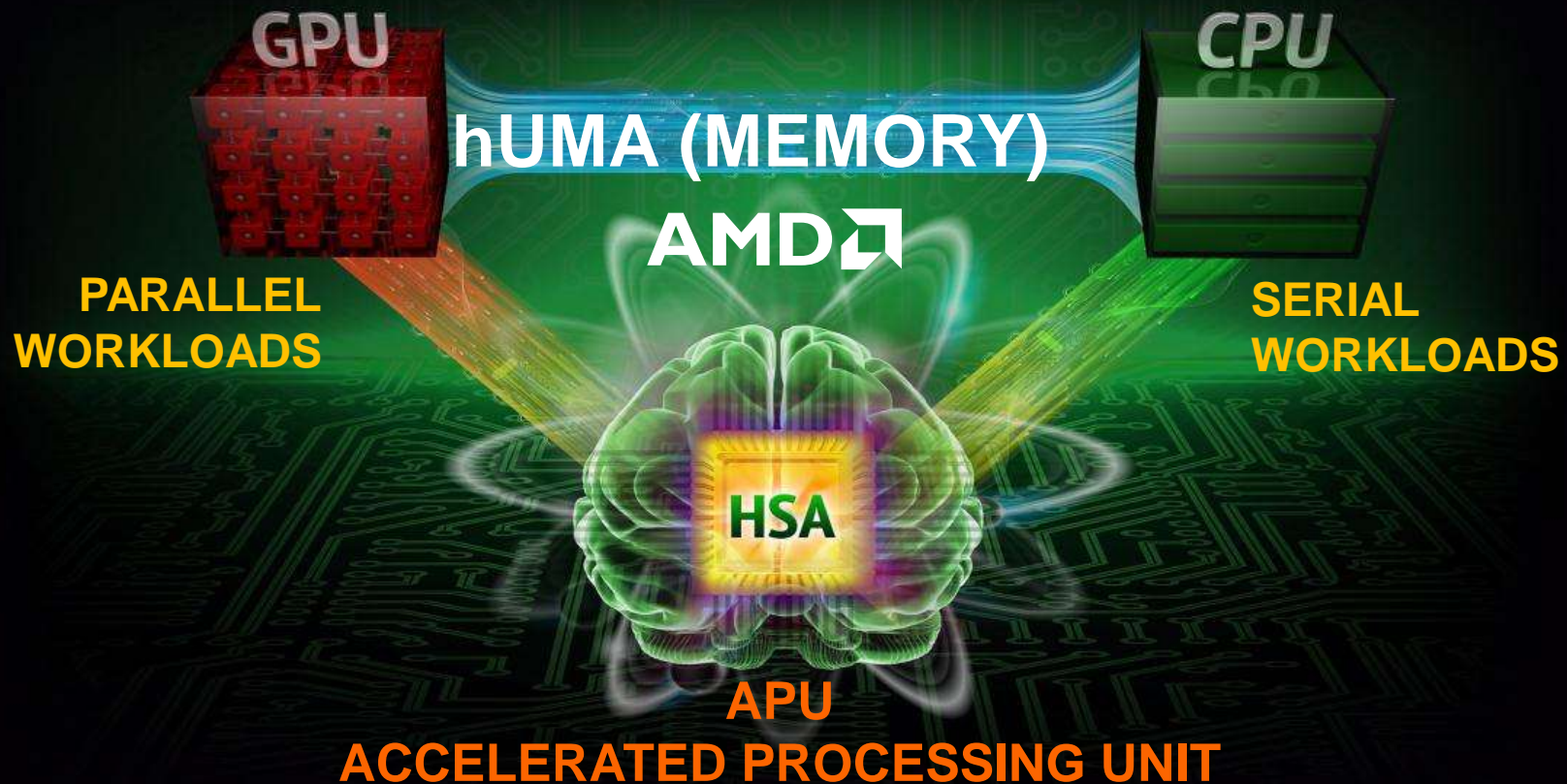
HSA = GPGPU ?

DISCRETE CPU & GPU

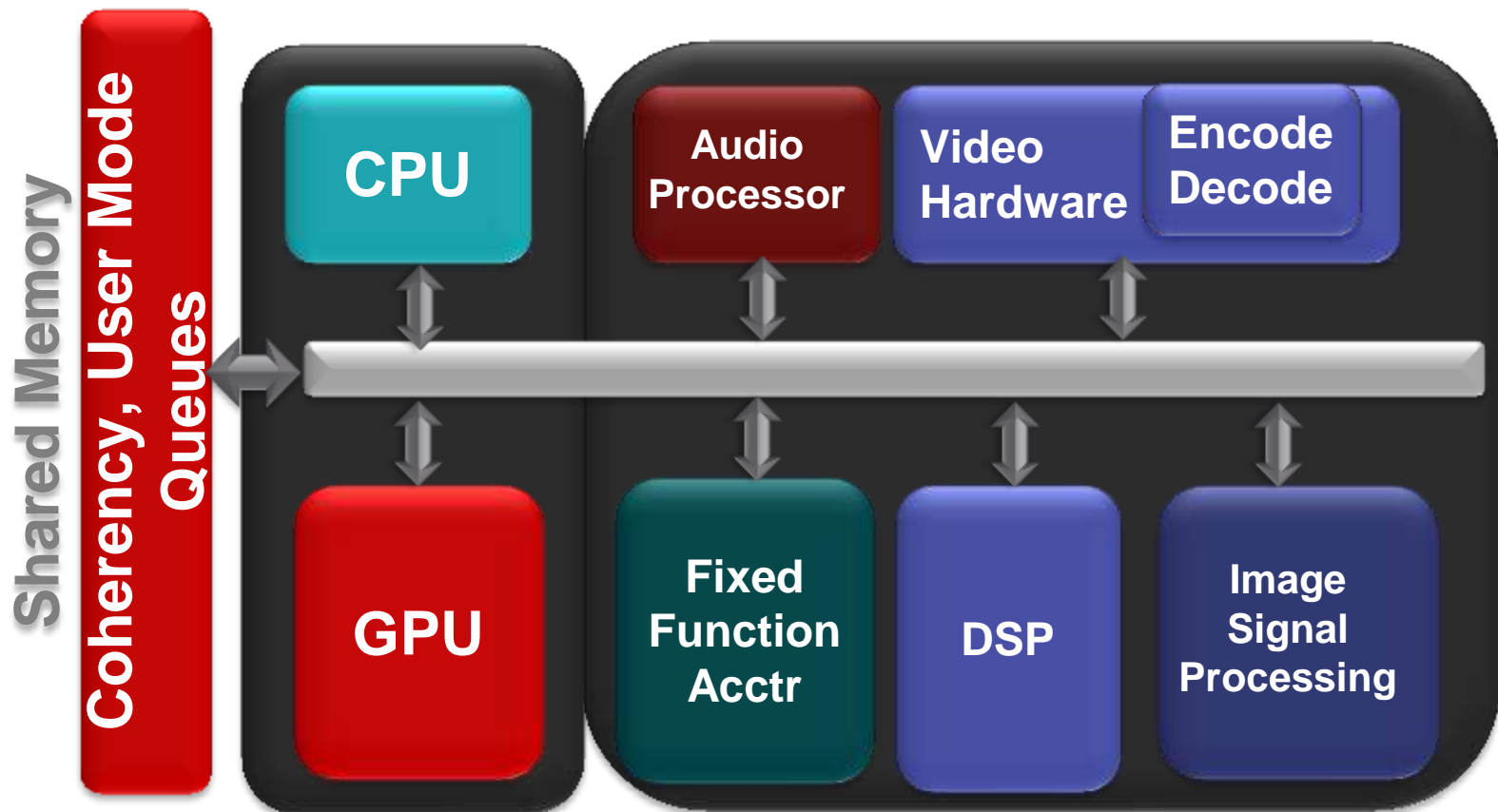


HSA = GPGPU ?

FUSED CPU & GPU



HIGH LEVEL ARCHITECTURE



HSA FOUNDATION

Founders



Promoters



Supporters



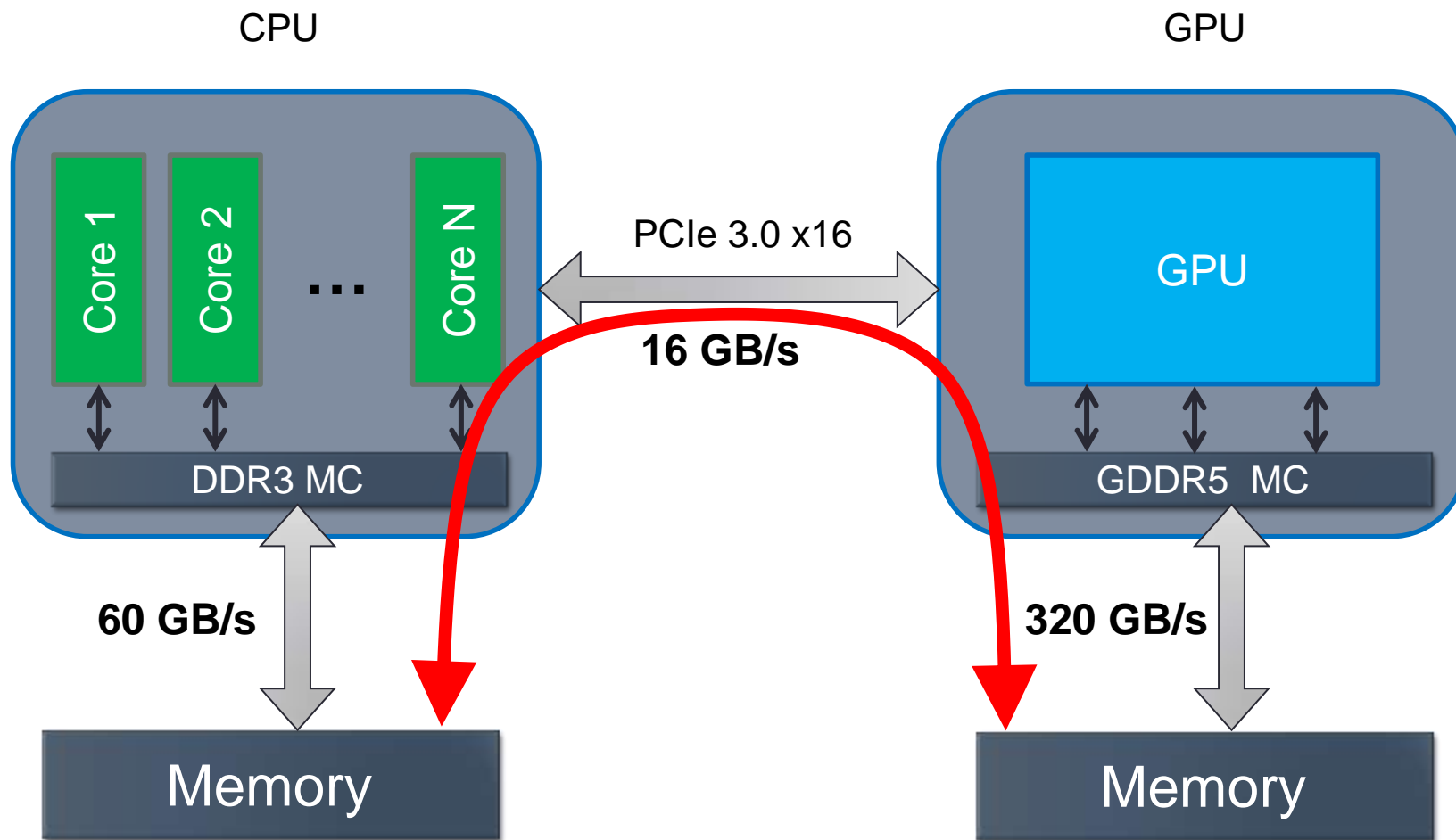
Contributors

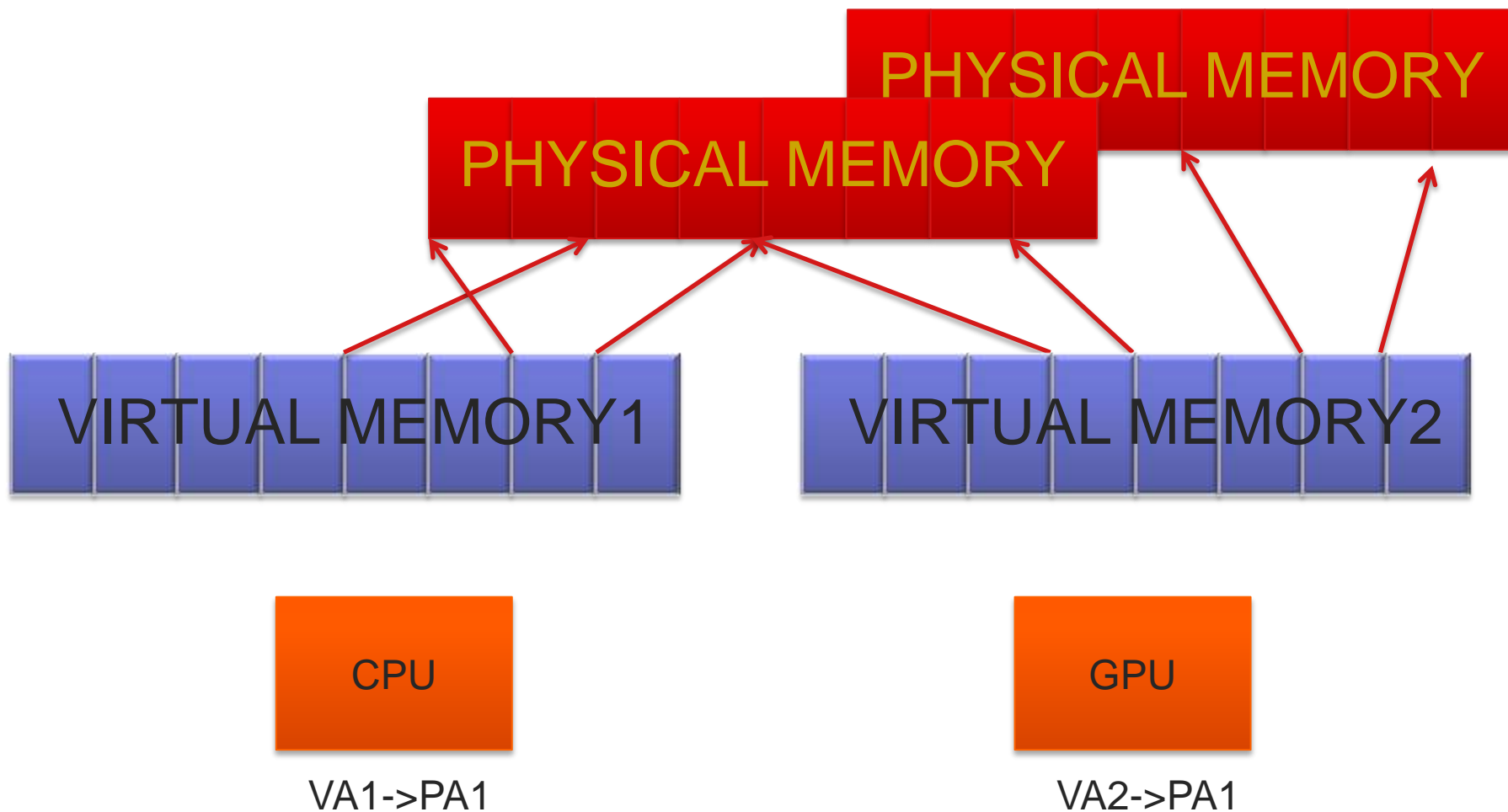


Academic



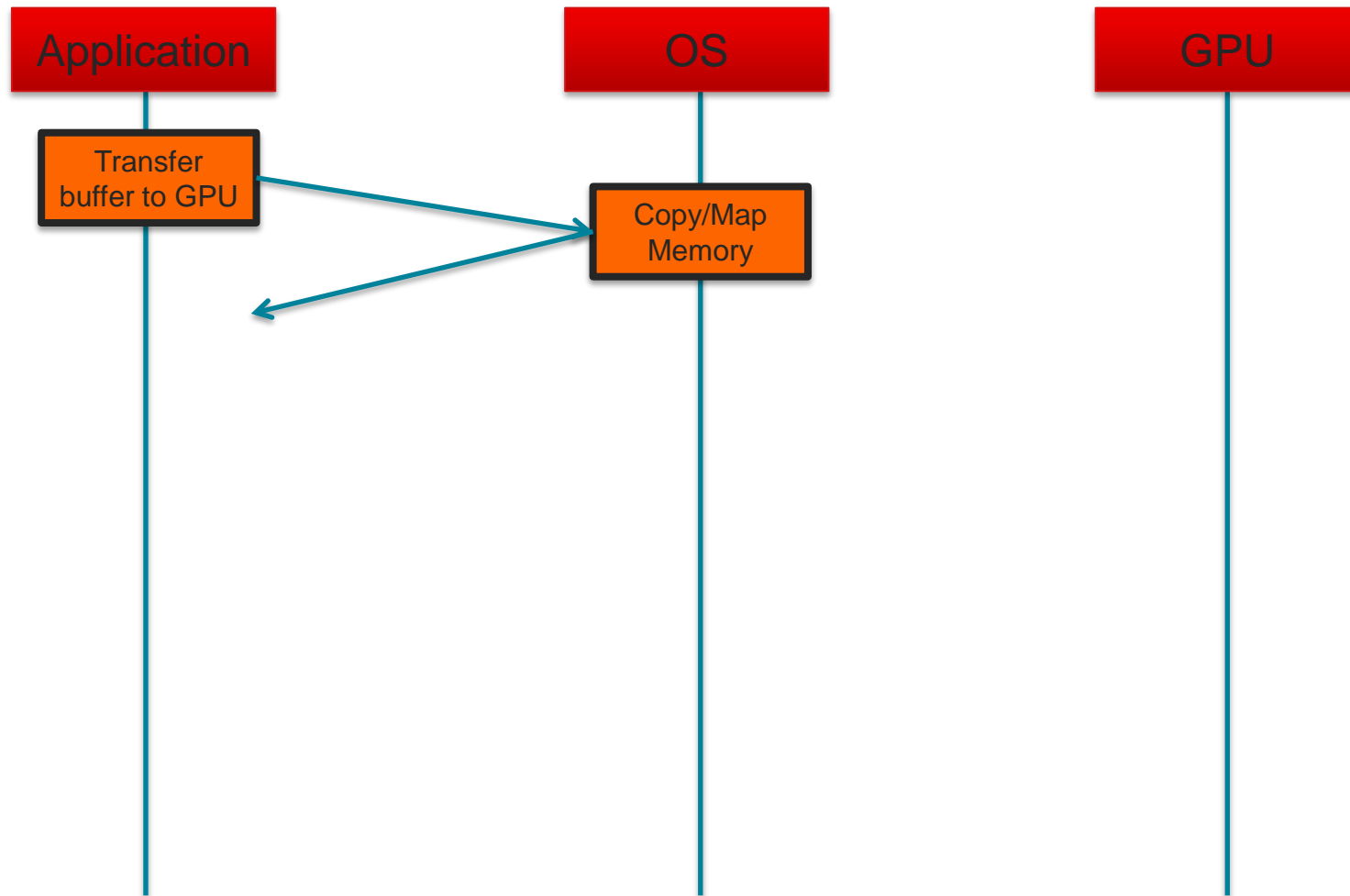
BANDWIDTH BOTTLENECK





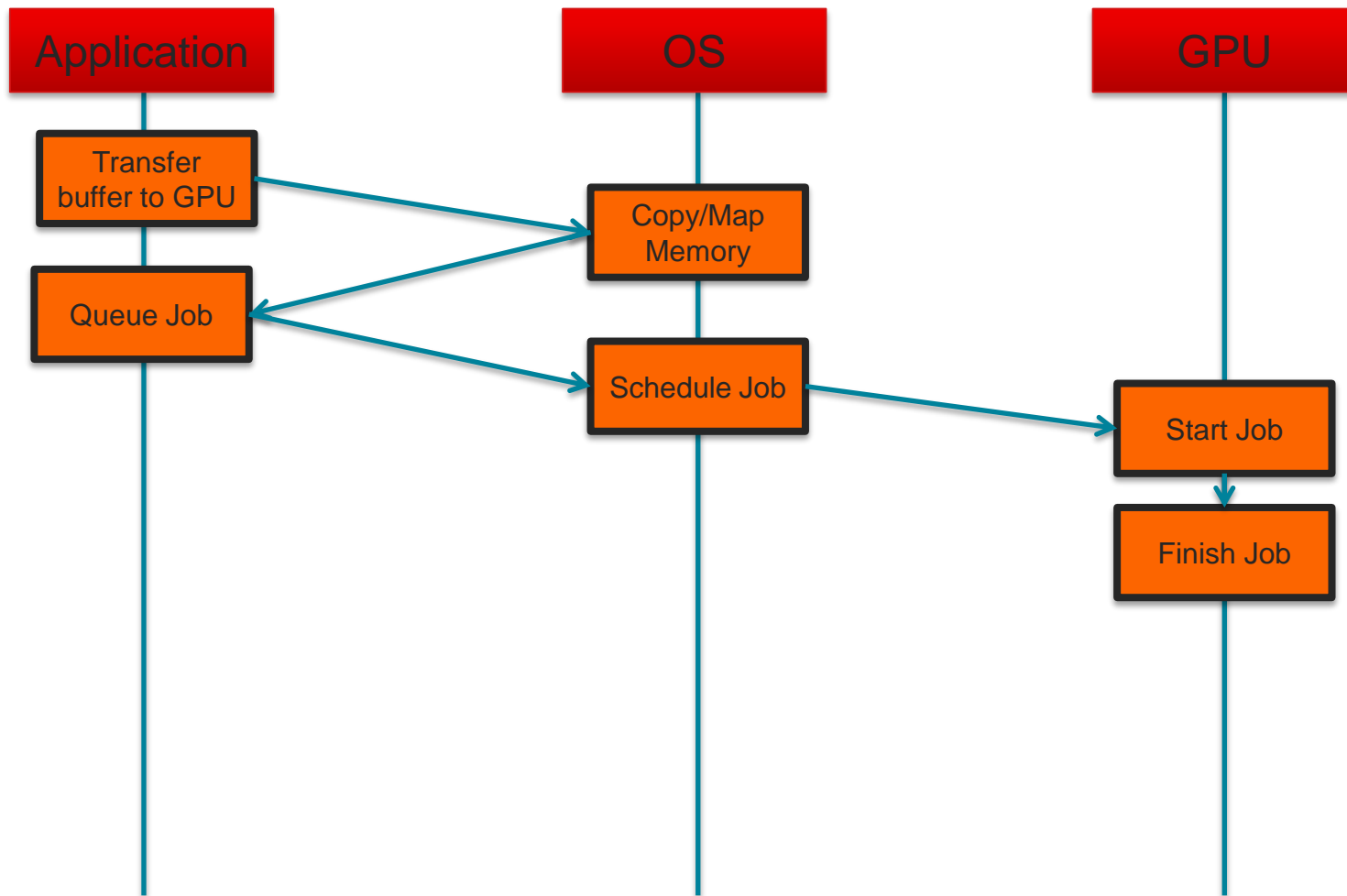
COMPUTE CHALLENGES

DISPATCH LATENCY



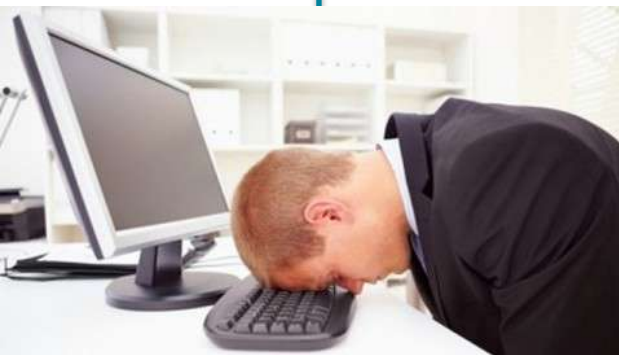
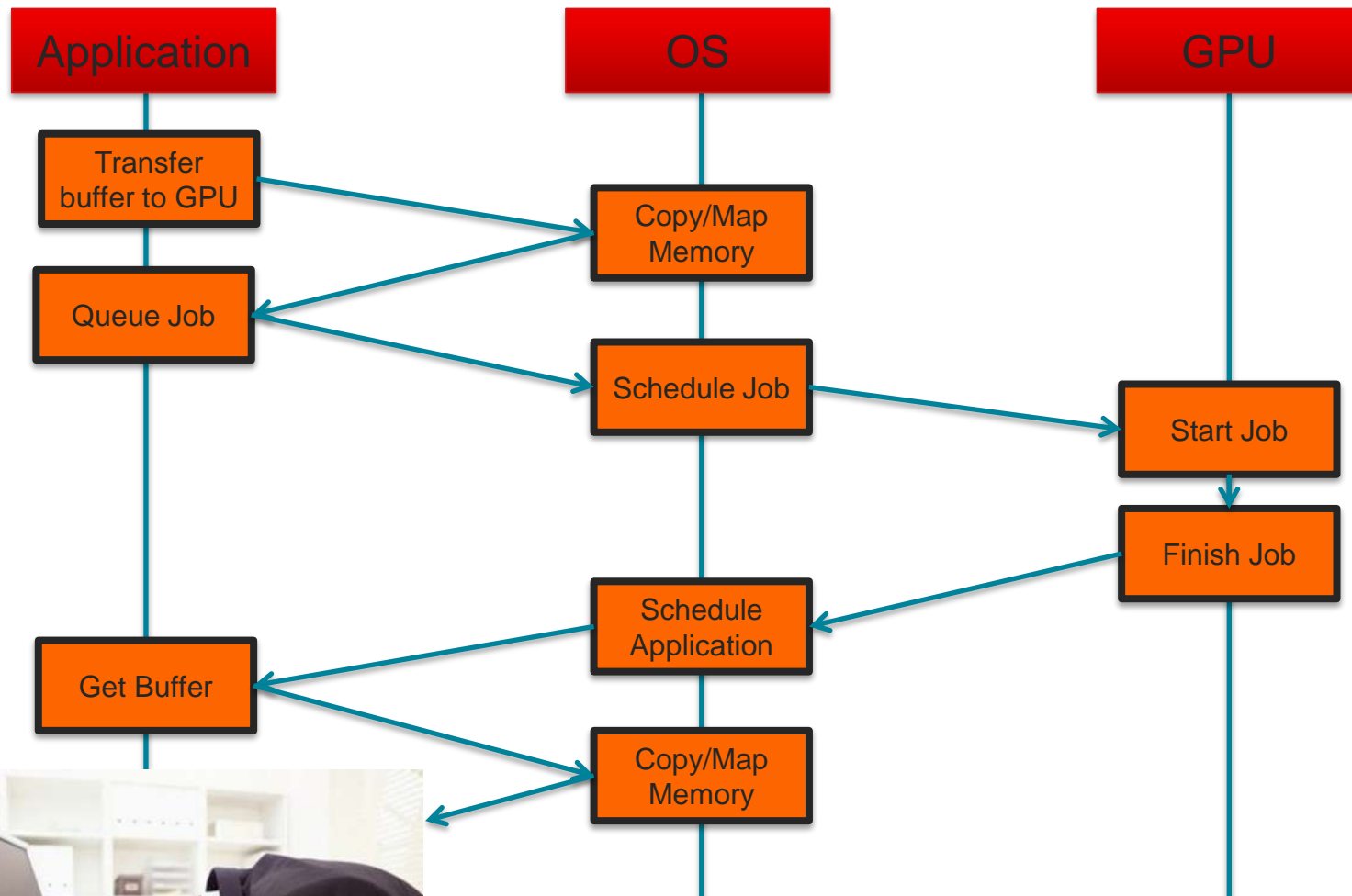
COMPUTE CHALLENGES

DISPATCH LATENCY



COMPUTE CHALLENGES

DISPATCH LATENCY





HSA FEATURES



HETEROGENEOUS UNIFIED MEMORY ARCHITECTURE

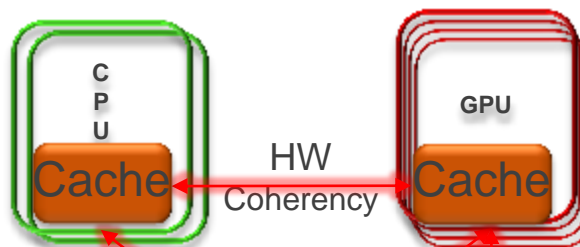
AKA HUMA



SHARED MEMORY

Coherent Memory:

Ensures CPU and GPU caches both see an up-to-date view of data



Pageable memory:

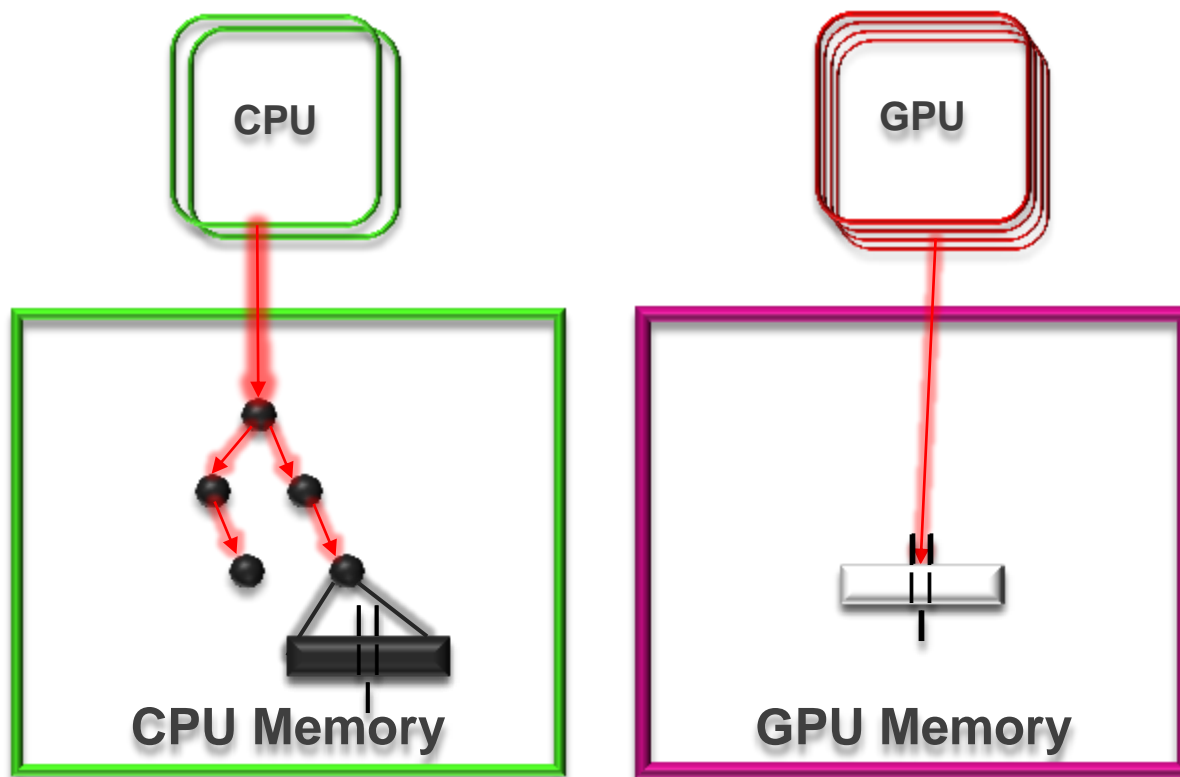
The GPU can seamlessly access virtual memory addresses that are not (yet) present in physical memory



Entire memory space:
Both CPU and GPU can access and allocate any location in the system's virtual memory space

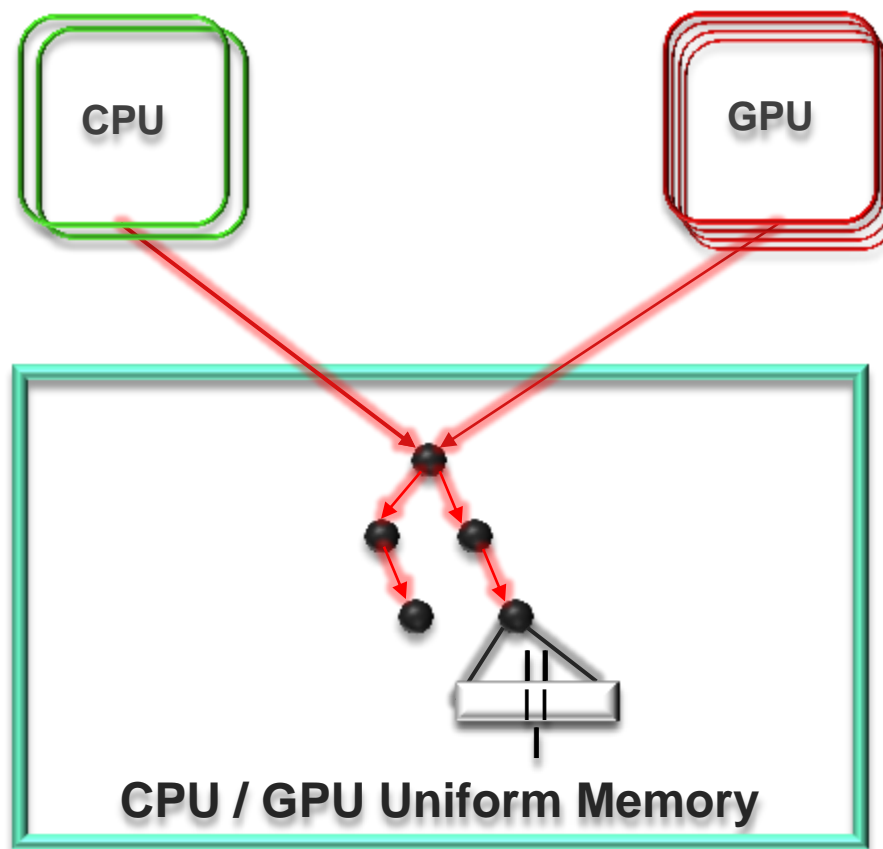
SHARED MEMORY

- CPU explicitly copies data to GPU memory
- GPU completes computation
- CPU explicitly copies result back to CPU memory

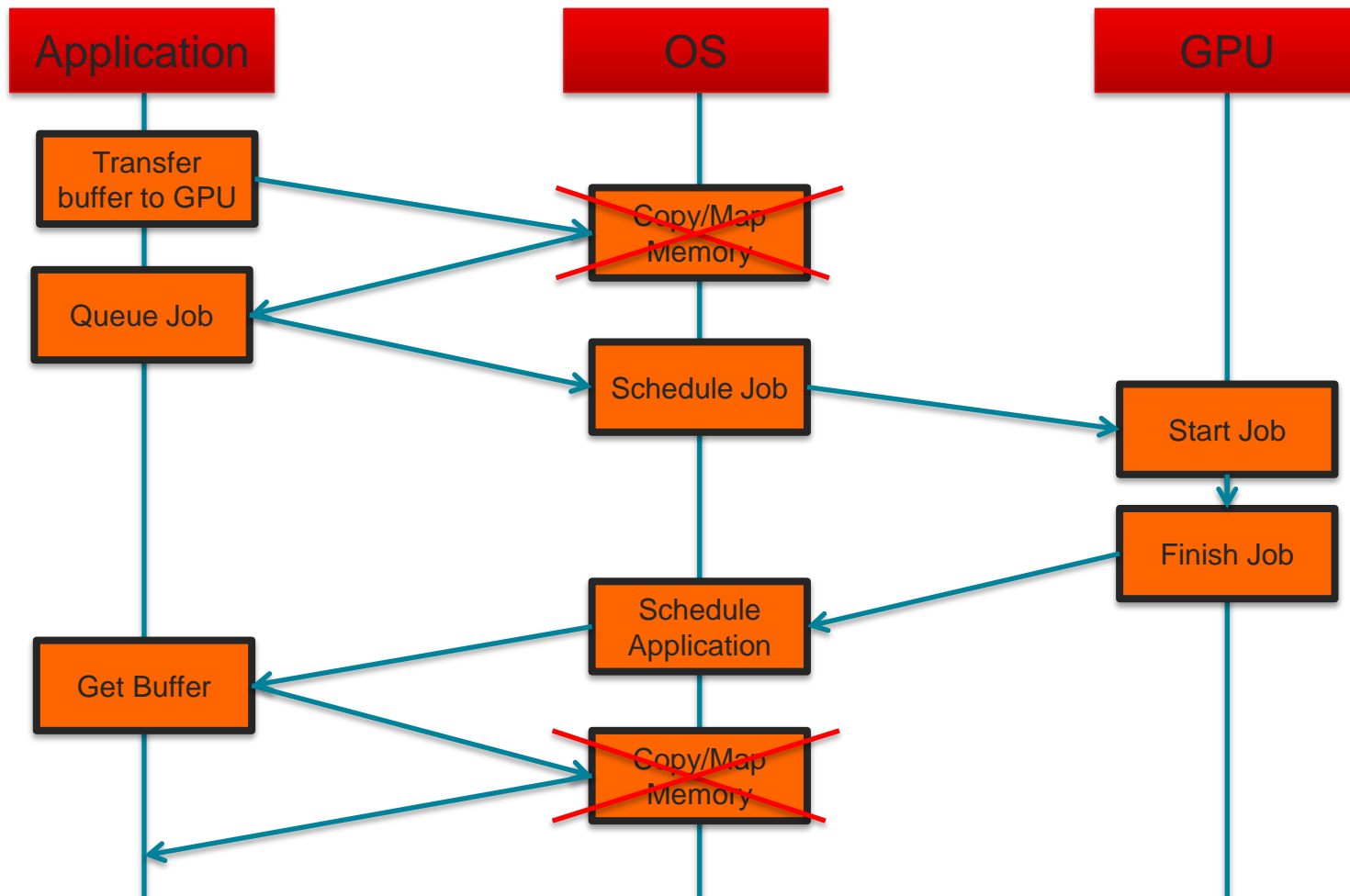


SHARED MEMORY

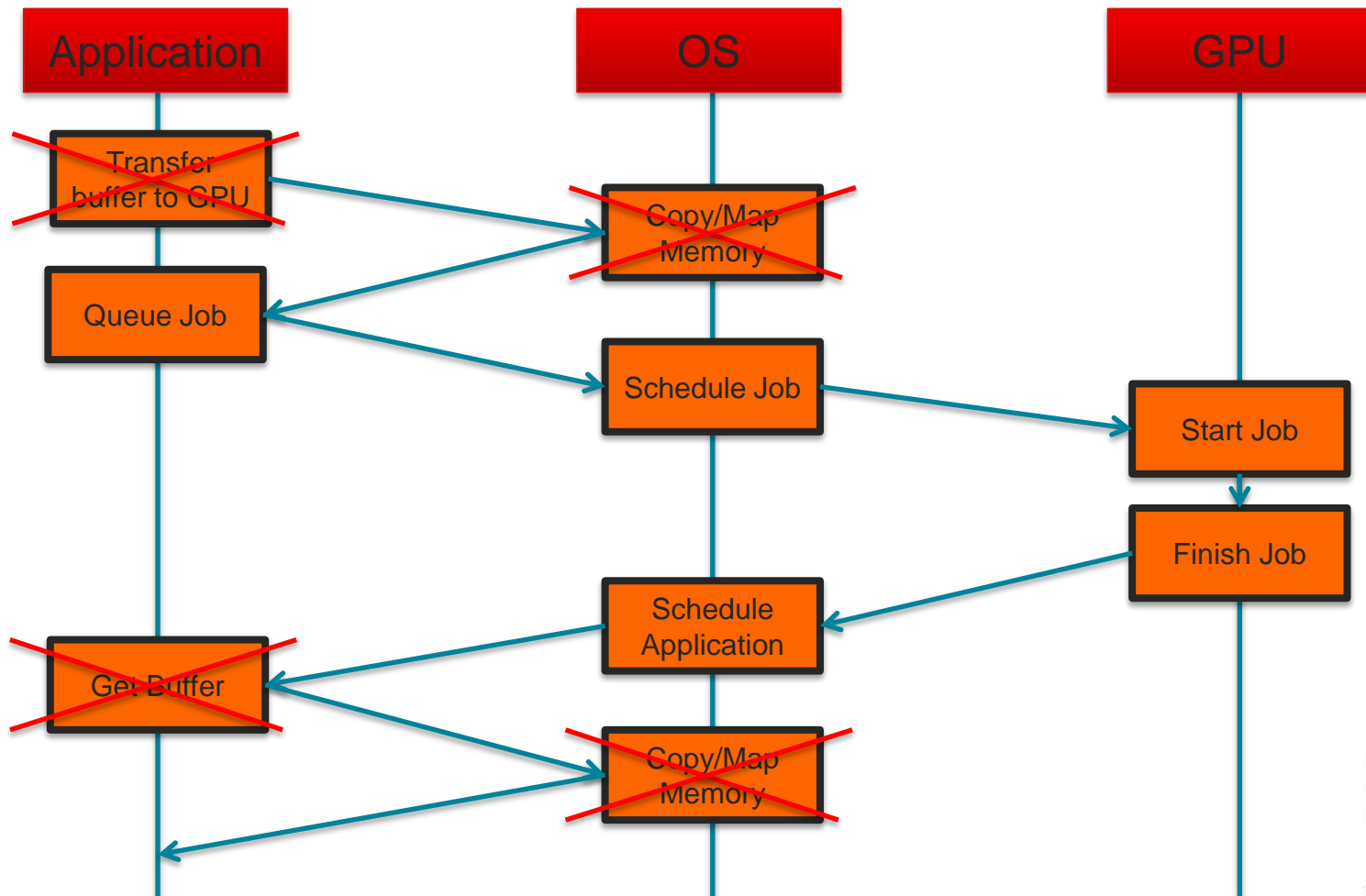
- CPU simply passes a pointer to GPU
- GPU complete computation
- CPU can read the result directly – no copying needed!



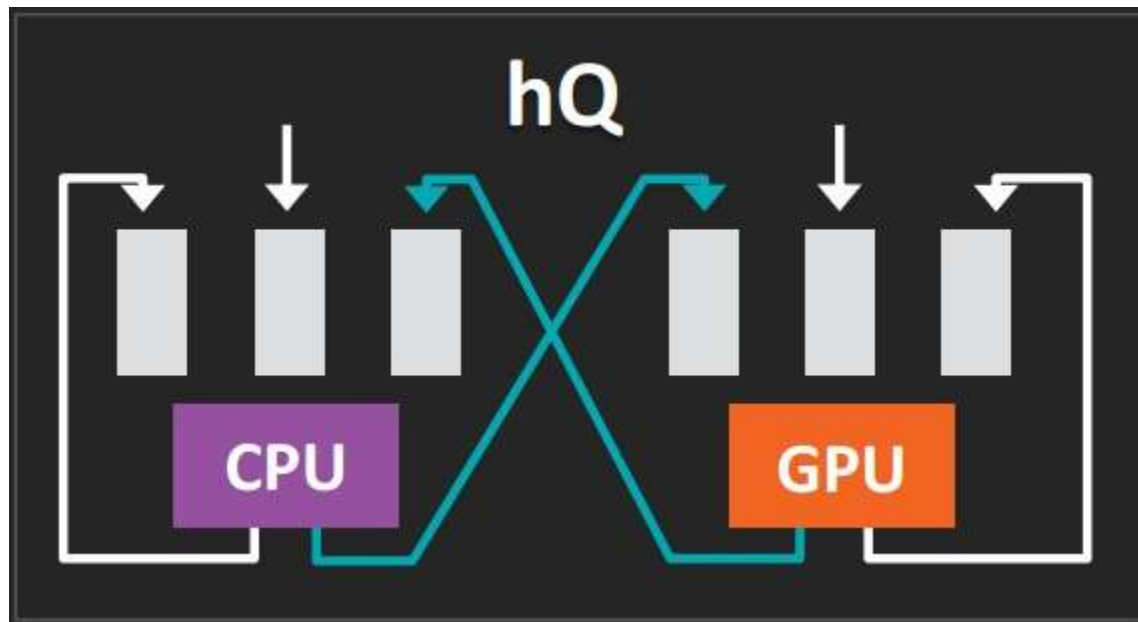
SHARED MEMORY



SHARED MEMORY

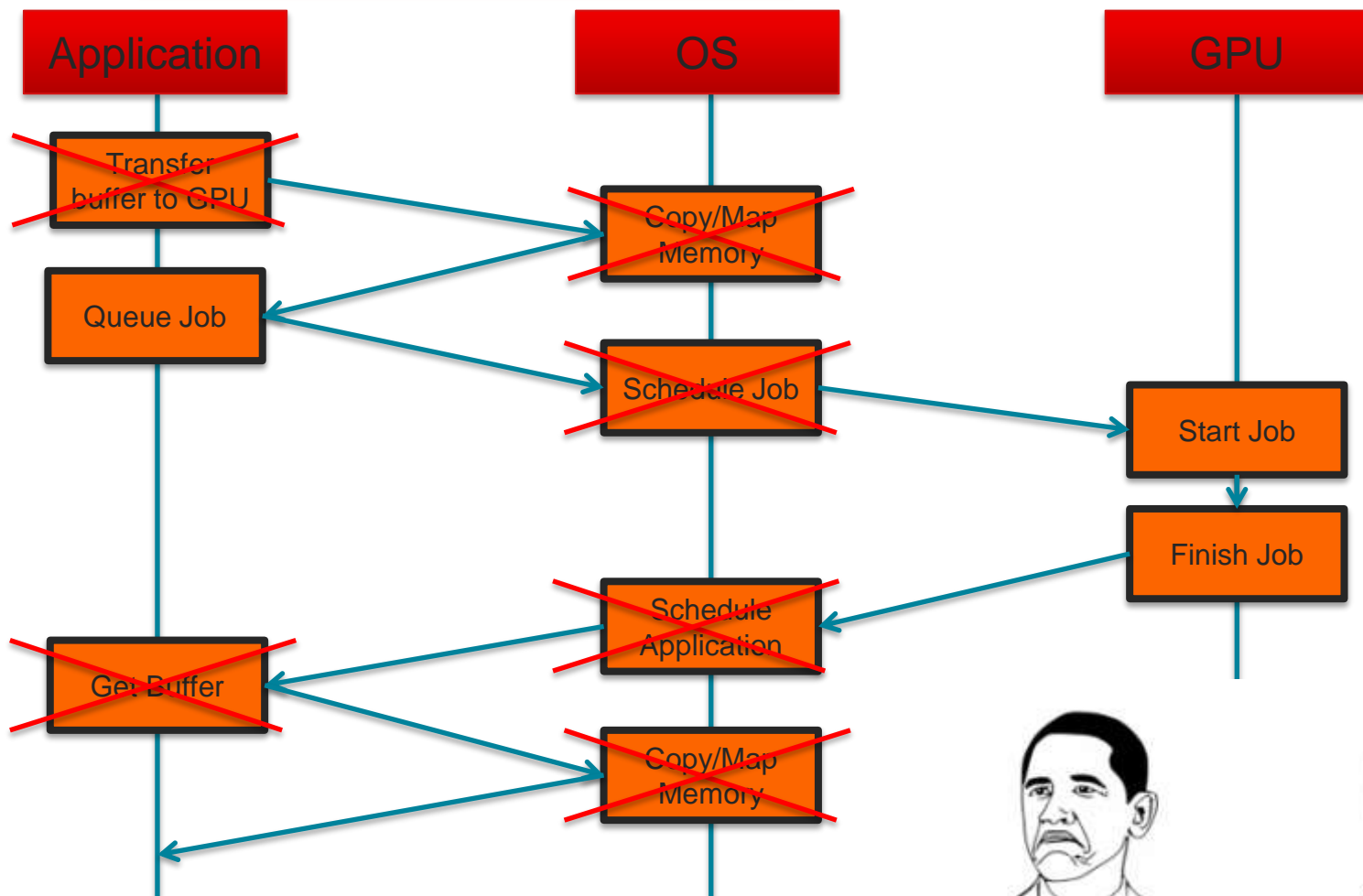


NOT BAD



- ▶ Heterogeneous queuing (hQ) defines how processors interact equally
- ▶ GPU and CPU have equal flexibility to create/dispatch work

HETEROGENEOUS QUEUING

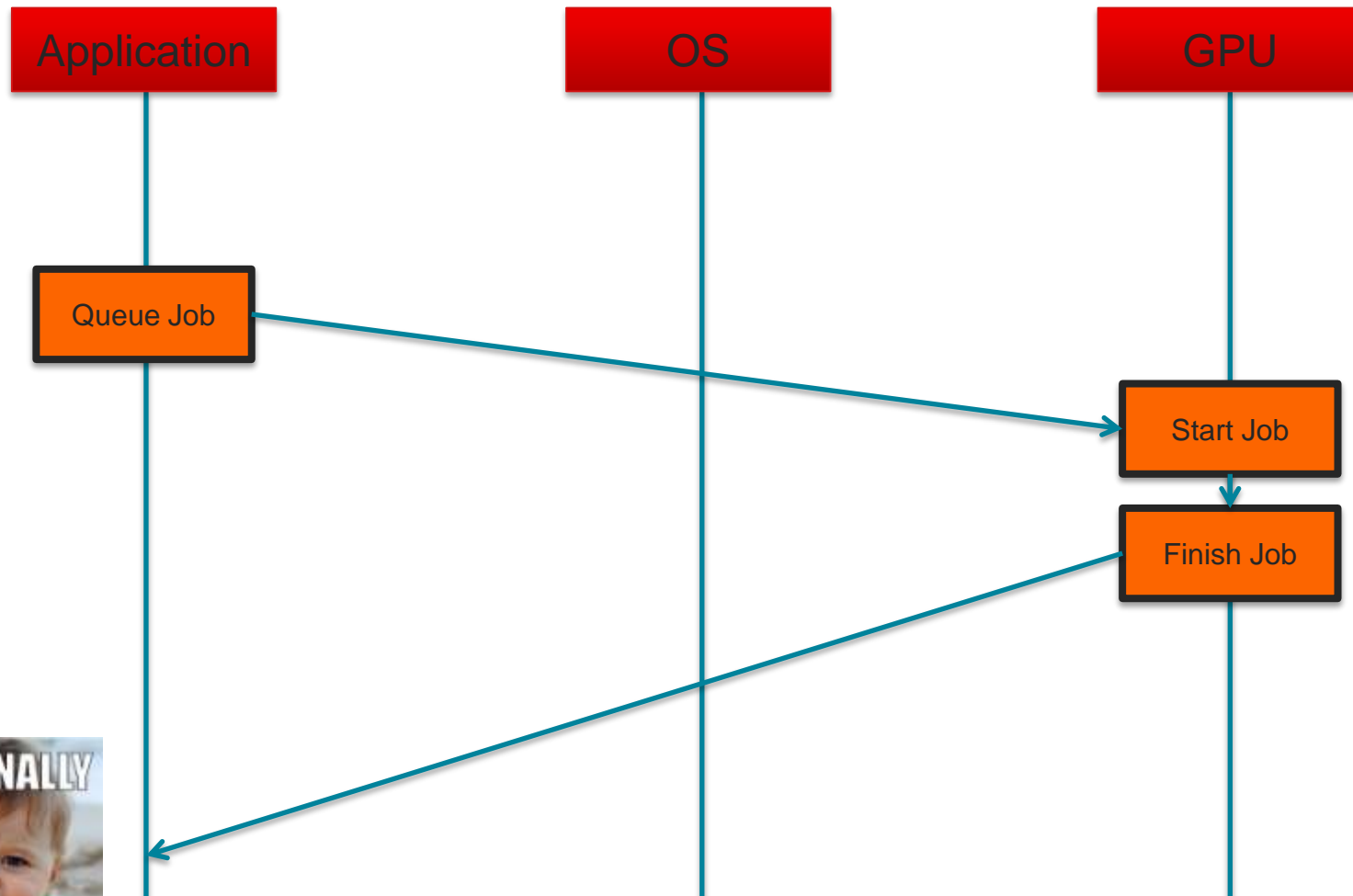


NOT BAD



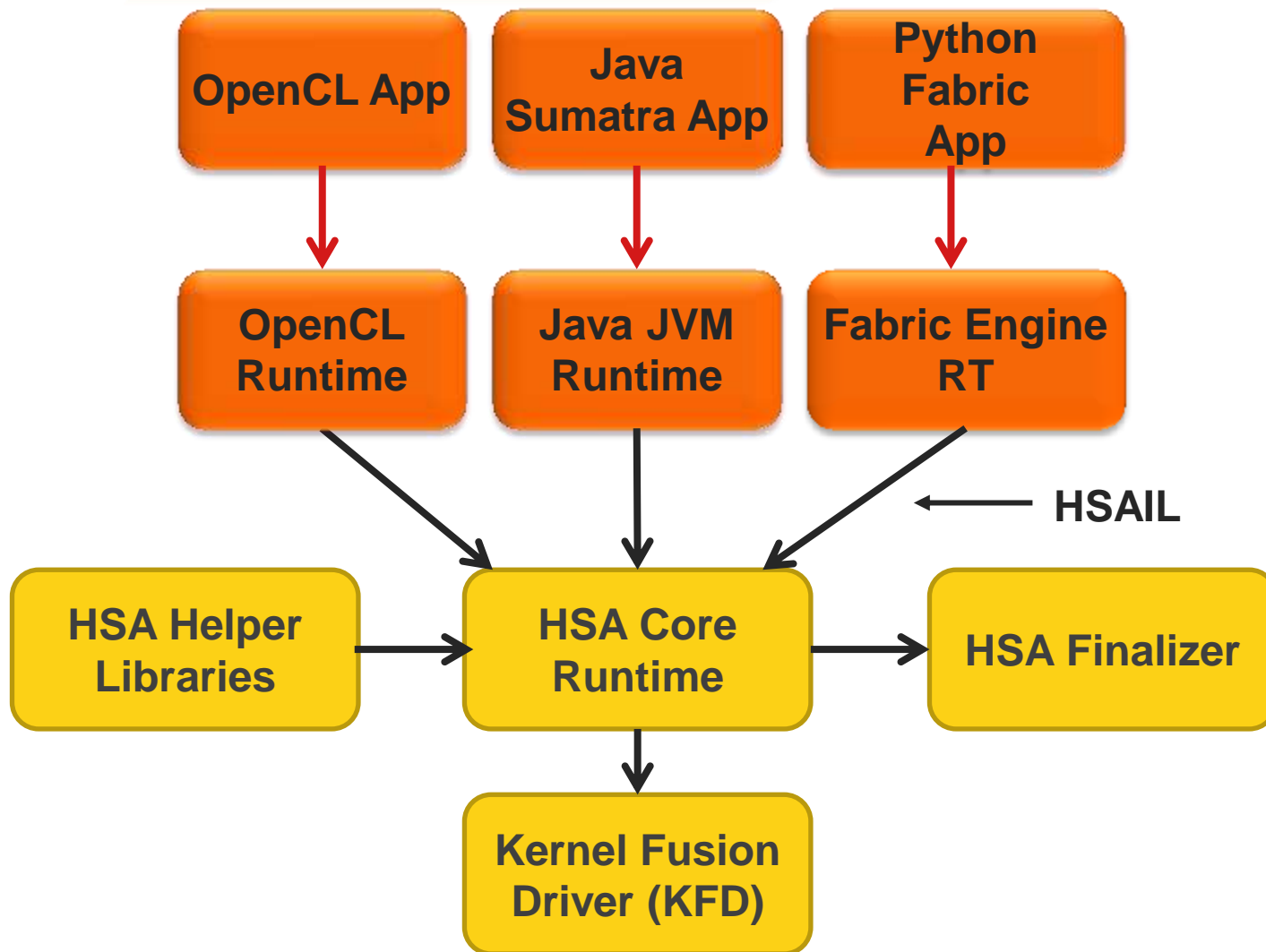
NOT BAD

LOW LATENCY DISPATCH



LANGUAGE SUPPORT

WITH HSAIL



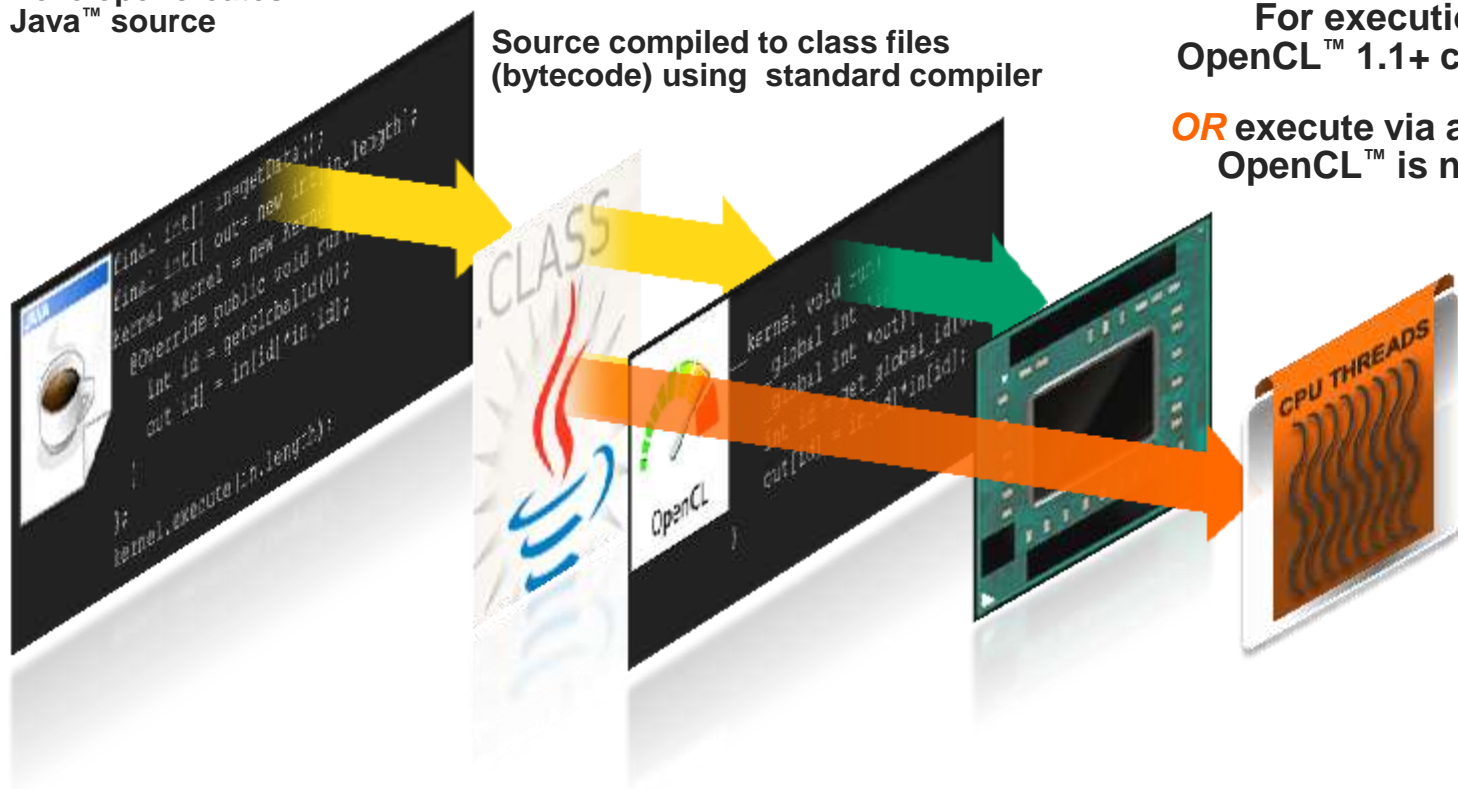
Aparapi = Runtime capable of converting Java™ bytecode to OpenCL™

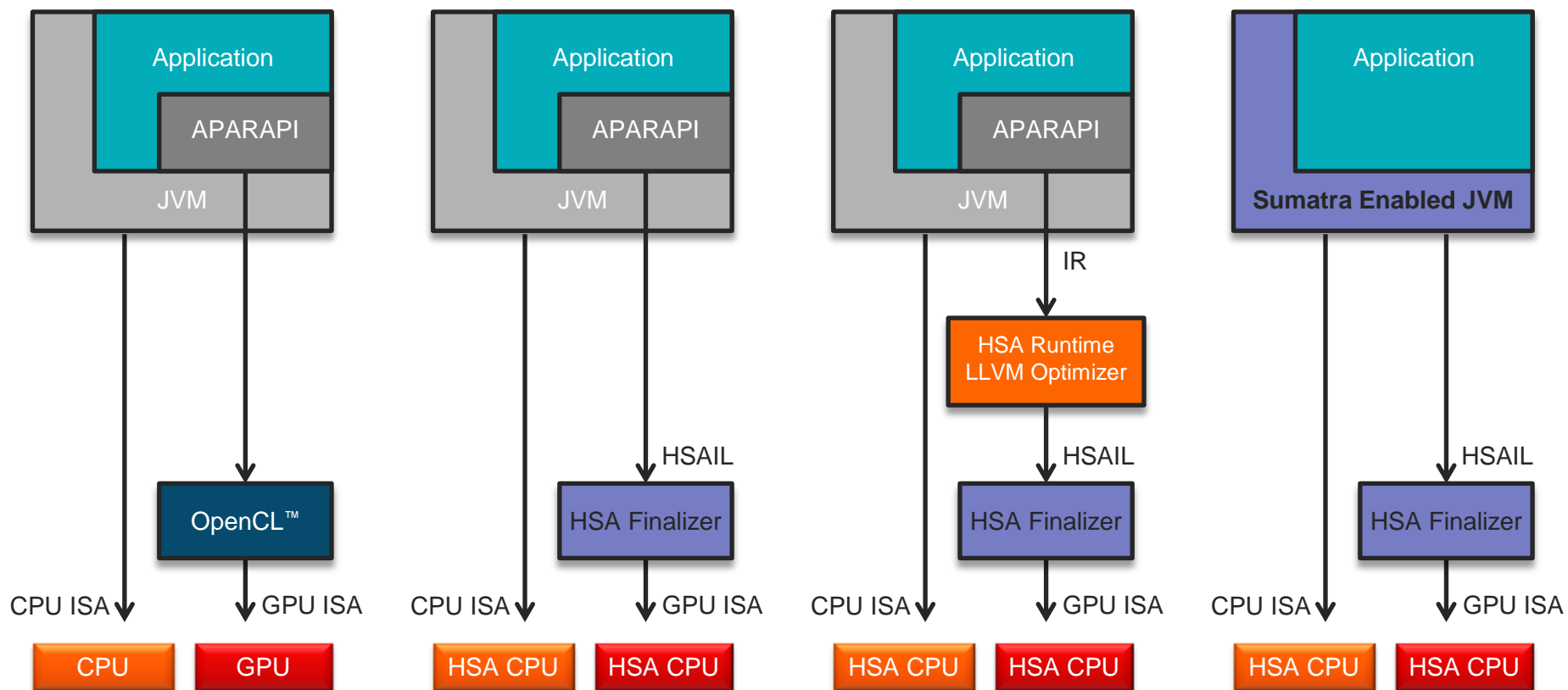
Developer creates
Java™ source

Source compiled to class files
(bytecode) using standard compiler

For execution on any
OpenCL™ 1.1+ capable device

OR execute via a thread pool if
OpenCL™ is not available



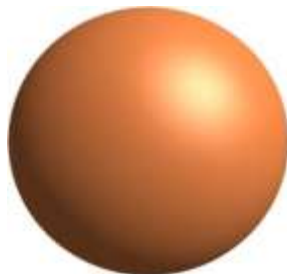
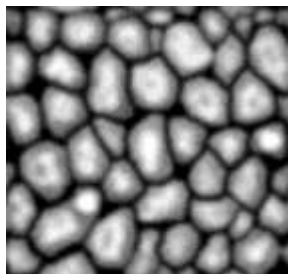
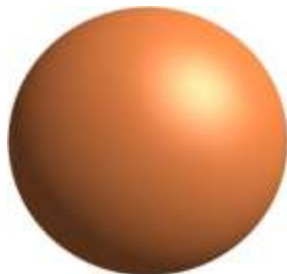
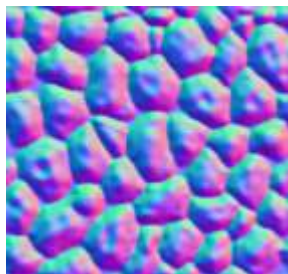
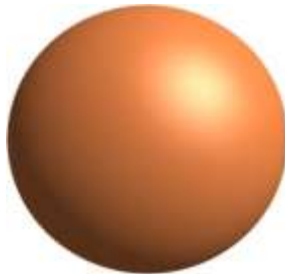




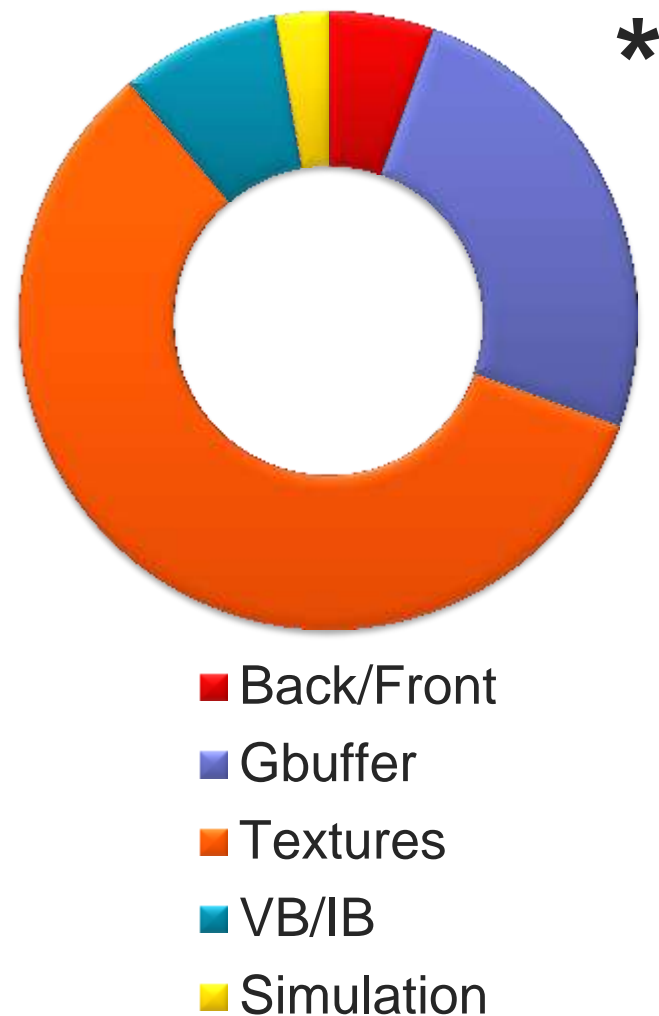
TEXTURE COMPRESSION



TEXTURE TYPES

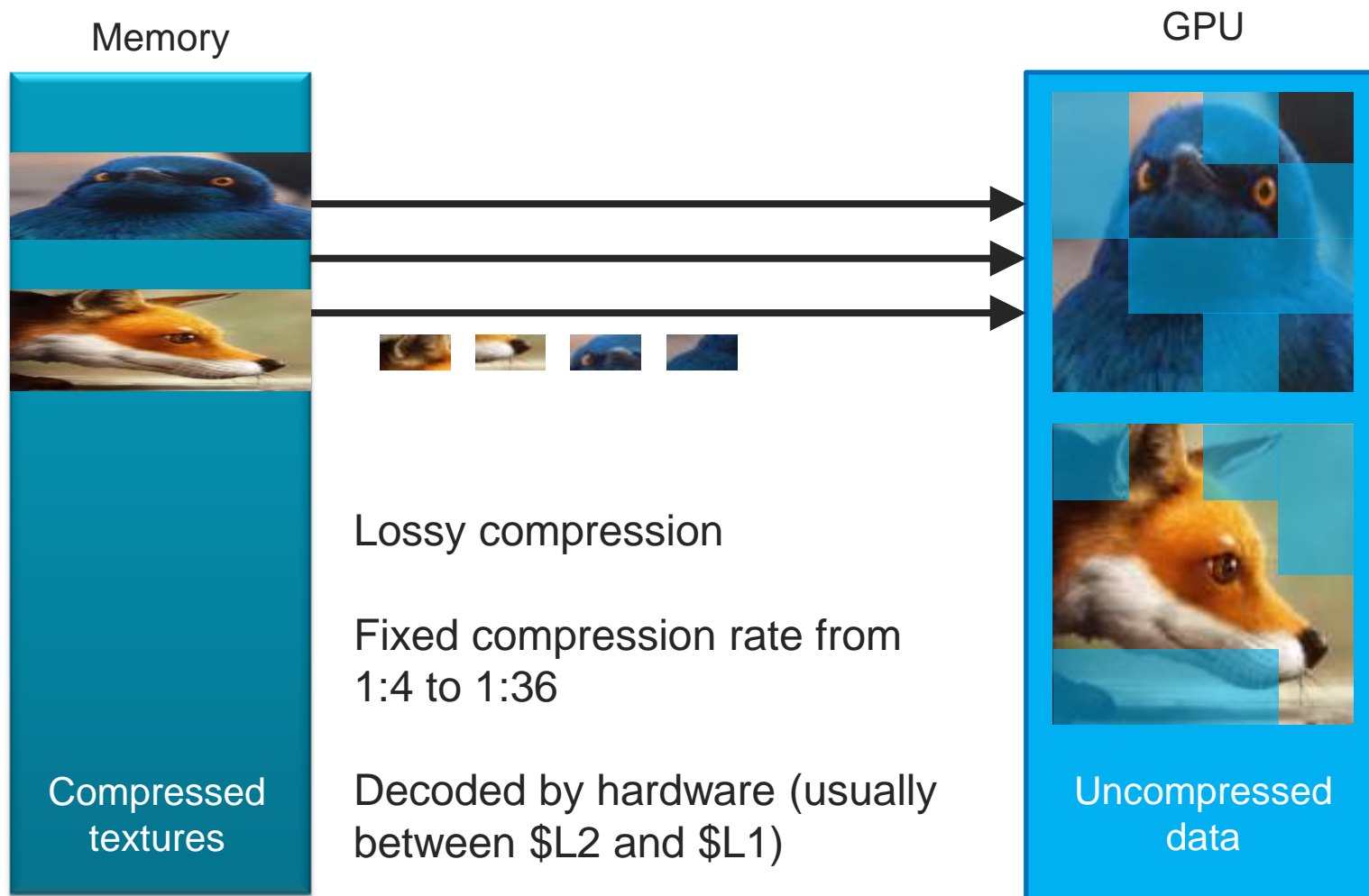


- ▶ Modern video games uses huge amount of memory for geometry and texture data
- The largest class of memory usage is **textures** (>60%)



* nVidia Eliminating Texture Waste: Borderless Ptex
// GDC2013

TEXTURE COMPRESSION



TEXTURE COMPRESSION





TEXTURE COMPRESSION

DXT1/BC1 (AKA S3TC)



Source block
512 bit

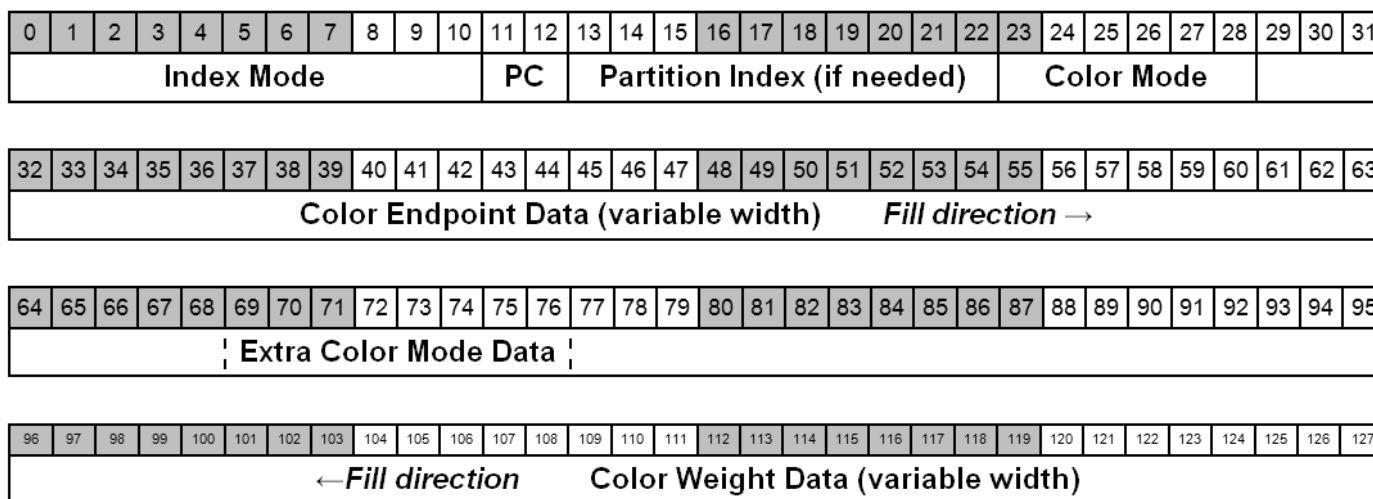
Color_0					}
Color_1					
10	00	10	01	}	
00	00	11	01		
00	10	11	01		
00	11	01	01		

Two color
endpoints
16 + 16 bit

Index table
4x4x2 bit

Compressed block
64 bit

- ▶ Adaptive Scalable Texture Compression
- ▶ Fixed block size of 128 bits; footprint determines bit rate
- ▶ BISE allows flexible allocation of bits between different kinds of information
- ▶ Supports:
 - ▶ LDR and HDR
 - ▶ 2D and 3D textures
 - ▶ Up to 4 color endpoint pairs
- ▶ Outperforms any other texture compression formats
- ▶ **Adopted by Khronos**



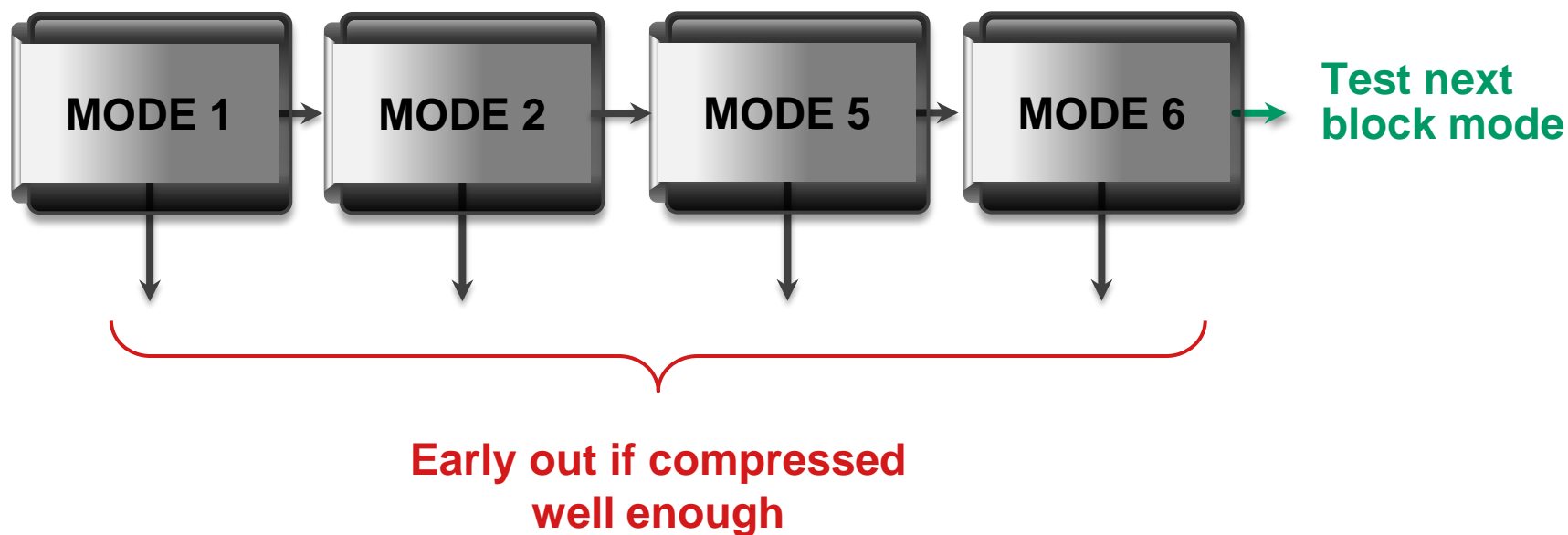


ACCELERATING ASTC COMPRESSION WITH HSA

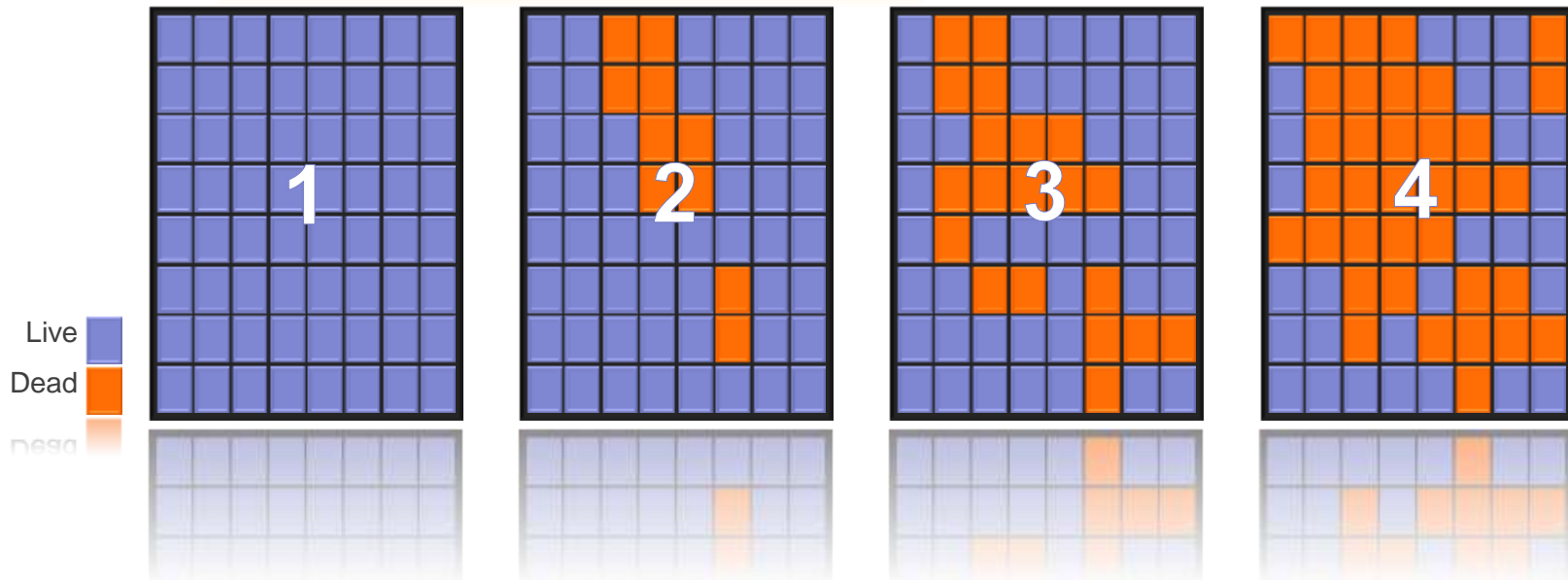


ASTC COMPRESSION

Testing 16 block modes, early out between each

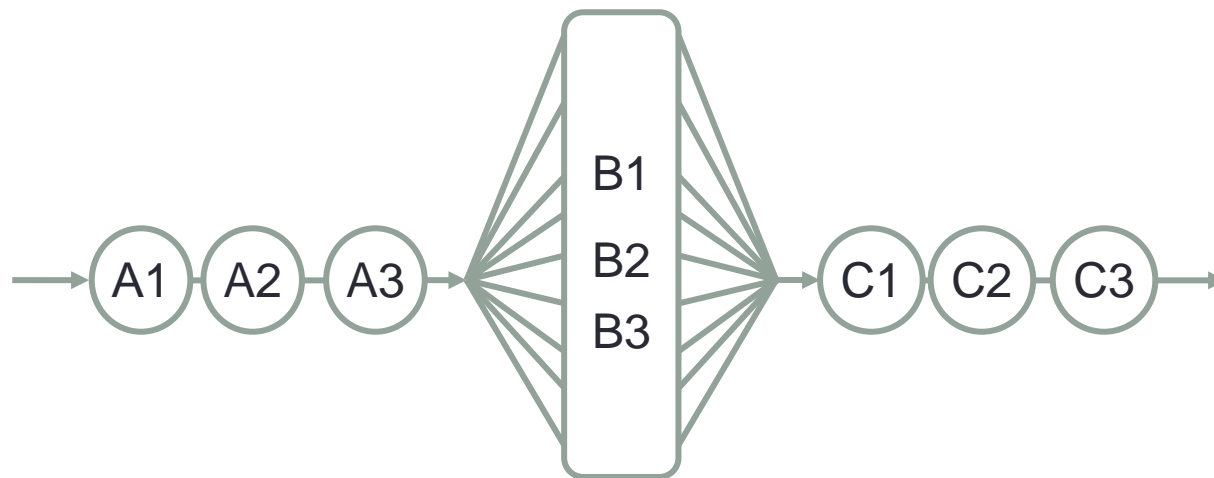
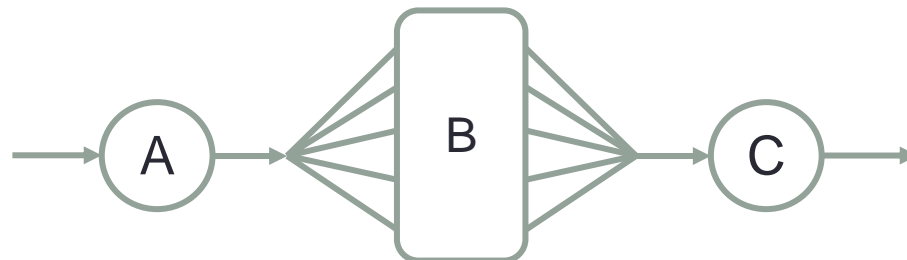


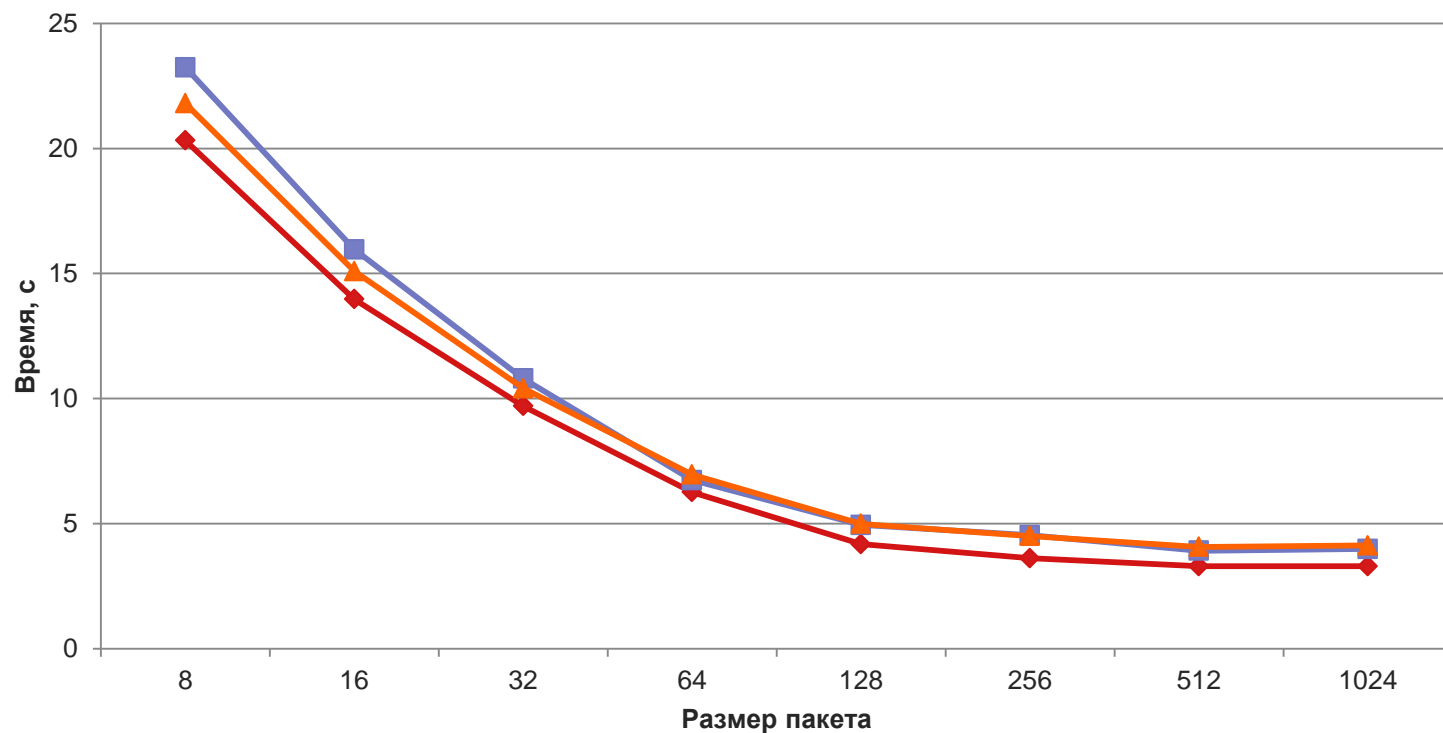
SIMD DIVERGENCE



- Early out algorithms exhibit divergence between work items
 - Some work items exit early
 - Their neighbors continue
 - SIMD packing suffers as a result

BLOCK BATCHING





- ▶ We need at least 128 blocks in a batch to feed SIMD core
- ▶ Big batches consumes a lot of memory (~900 MB for 512 blocks)
- ▶ Copying all this data will ruin performance

RESULTS SO FAR

Quality Settings	Compression time		Speedup
	Original codec	HSA accelerated codec	
Medium	12.2 sec	3.4 sec	3.59x
Thorough	47.1 sec	10.6 sec	4.44x
Exhaustive	109.3 sec	21.3 sec	5.13x

- ▶ AMD A10-7850K – 4 CPU cores @3.7GHz, 8 GPU cores @720Mhz
- ▶ Up to 5x speedup
- ▶ No dynamic load balancing, so there is a room for increasing performance even further



QUESTIONS AND
ANSWERS

