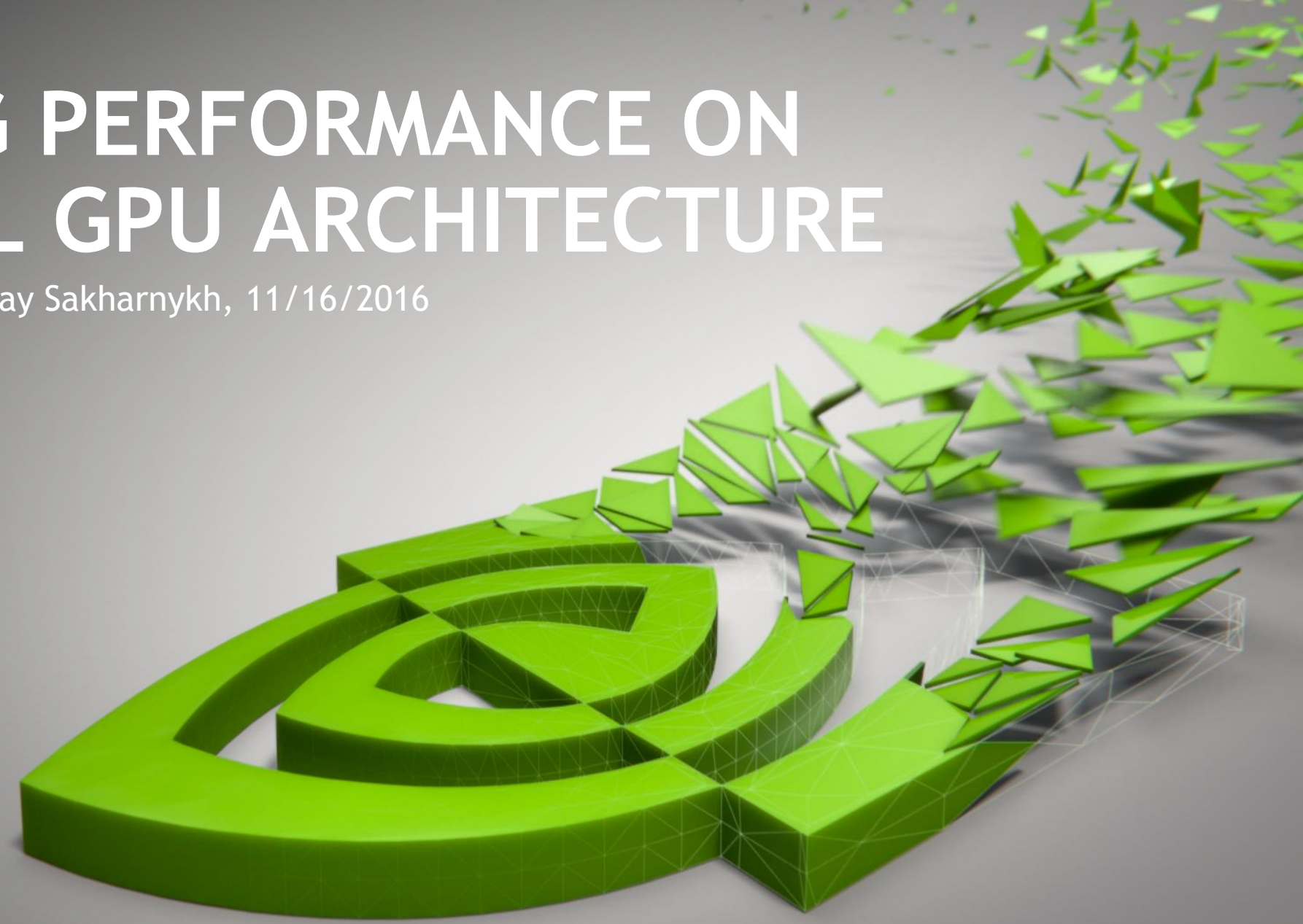
HPGMG PERFORMANCE ON PASCAL GPU ARCHITECTURE

Peng Wang, Nikolay Sakharnykh, 11/16/2016

NVIDIA



NEW DEVELOPMENTS IN 2016

https://bitbucket.org/nsakharnykh/hpgmg-cuda

Updated to include **4th order** implementation on GPU

Optimized setup phase by porting remaining routines to GPU

Updated levels allocation to **improve Unified Memory performance**

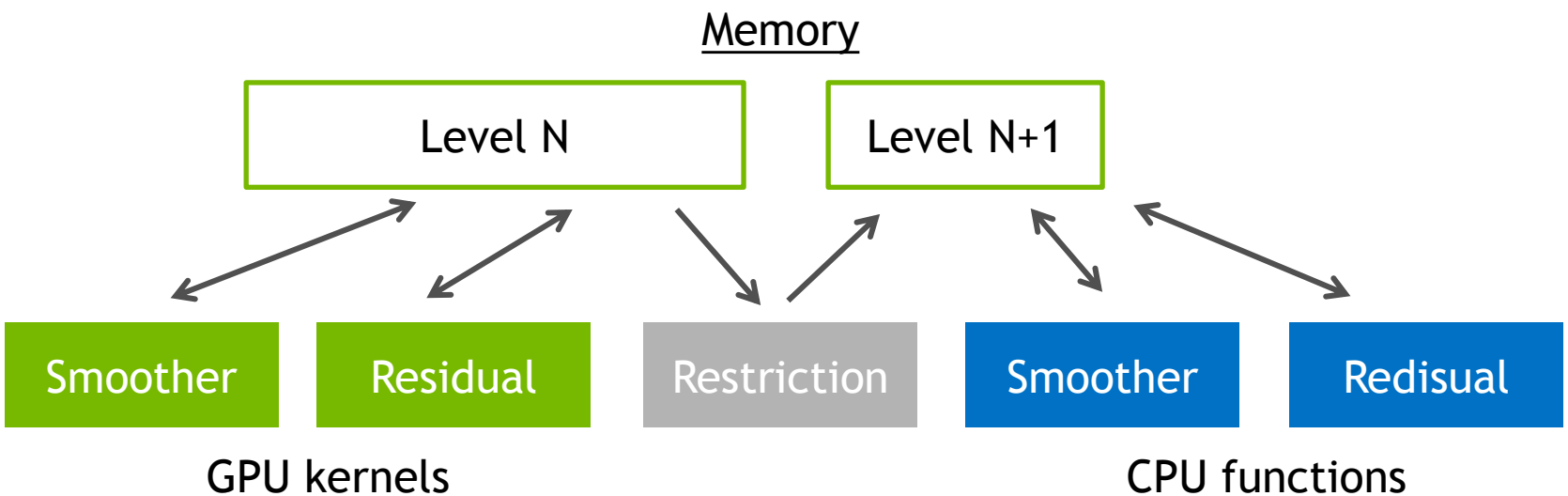
Better multi-GPU scaling using CUDA-aware MPI with GPUDirect P2P

GPU memory oversubscription study (Parallel Forall blog post is pending)

GPUDirect Async implementation (https://github.com/e-ago/hpgmg-cuda-async)

HYBRID IMPLEMENTATION

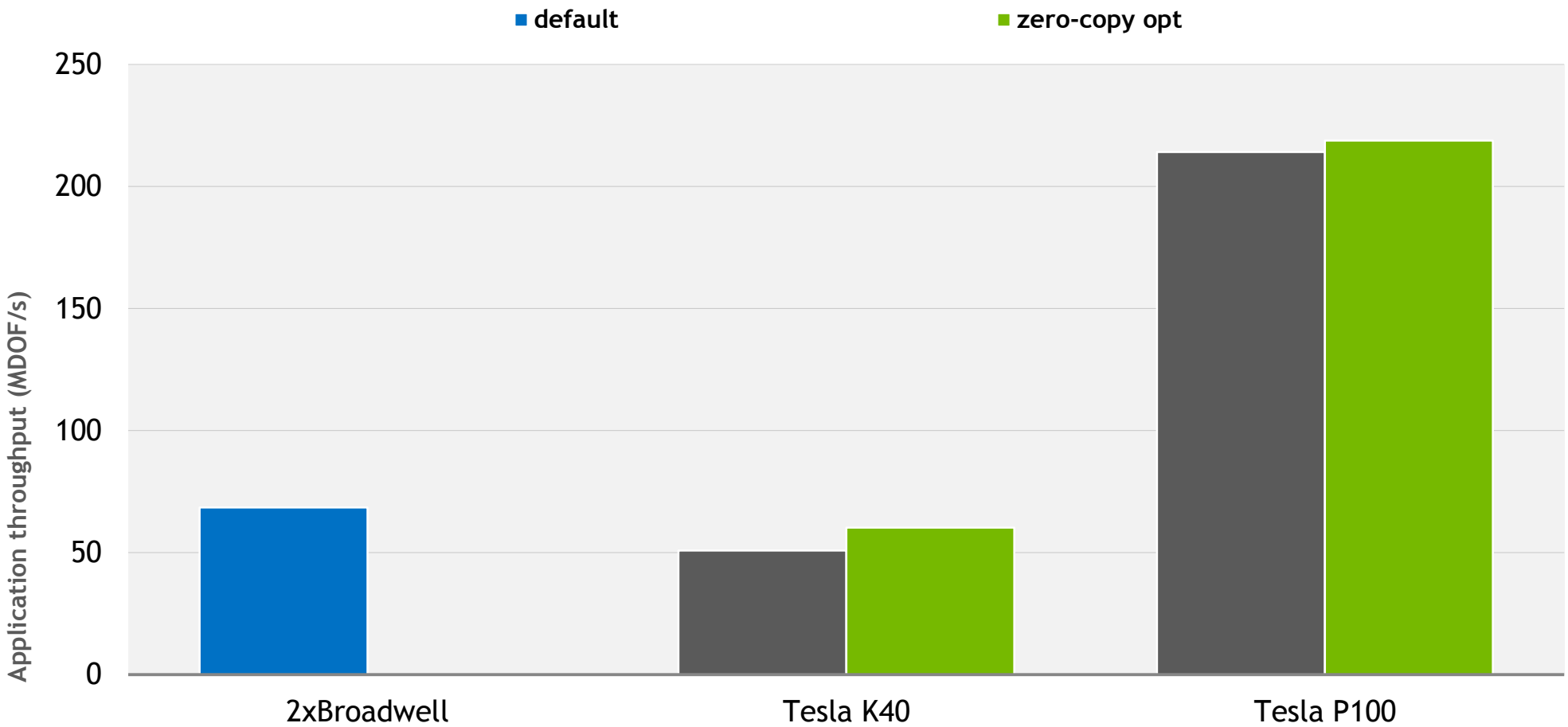
Data sharing between CPU and GPU



Level N+1 (small) is shared between CPU and GPU

To avoid frequent migrations allocate N+1 in **zero-copy memory**

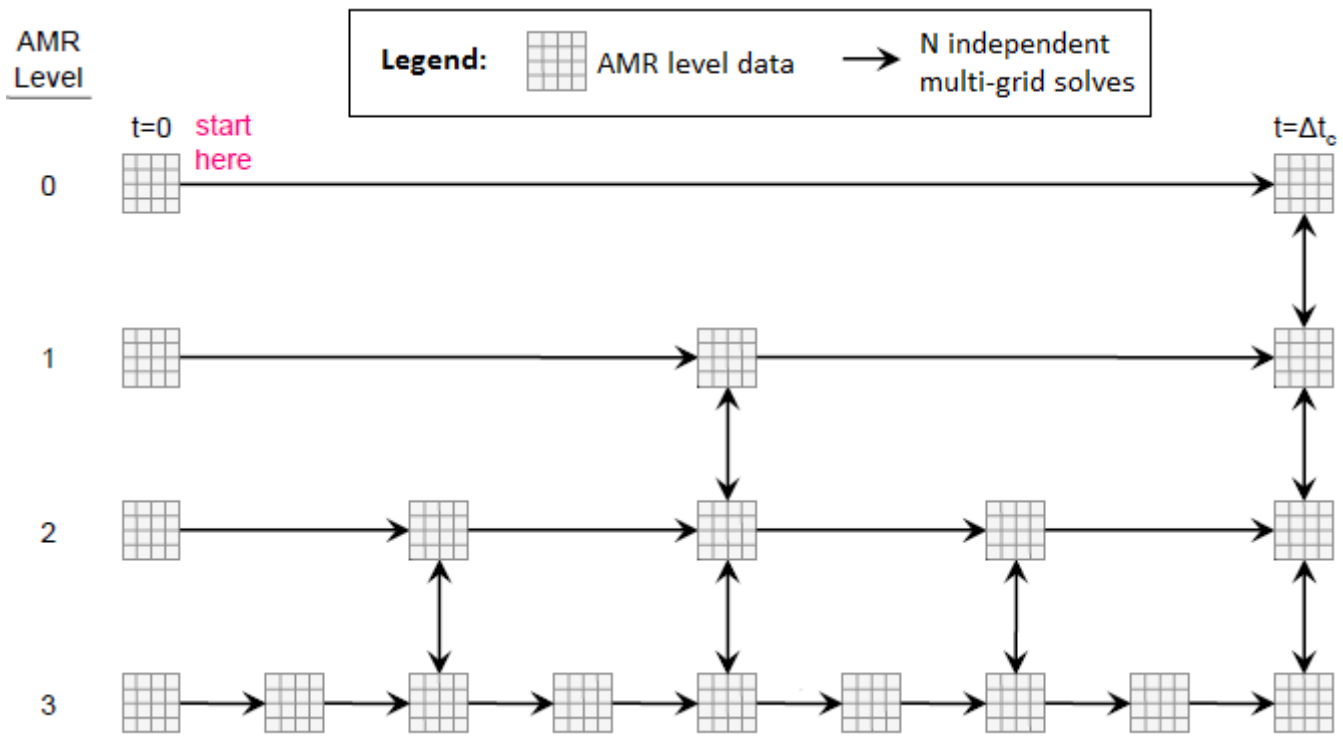
PERFORMANCE ON TESLA P100



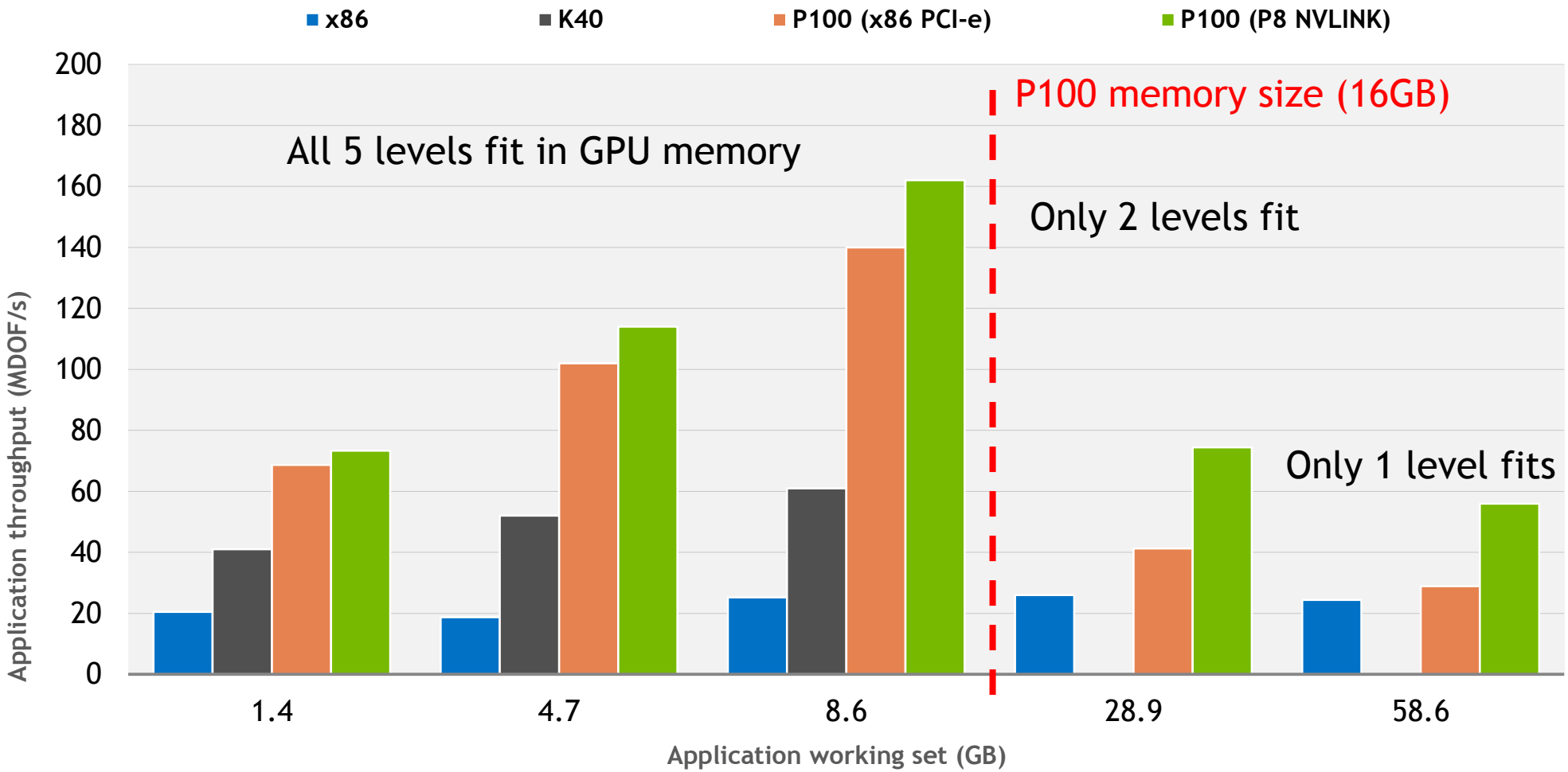
CPU: 2x Intel E5-2697 v4 @ 2.3GHz 3.6GHz Turbo (Broadwell-EP) HT off (36 cores)

HPGMG AMR PROXY

Data locality and reuse of AMR levels



OVERSUBSCRIPTION RESULTS

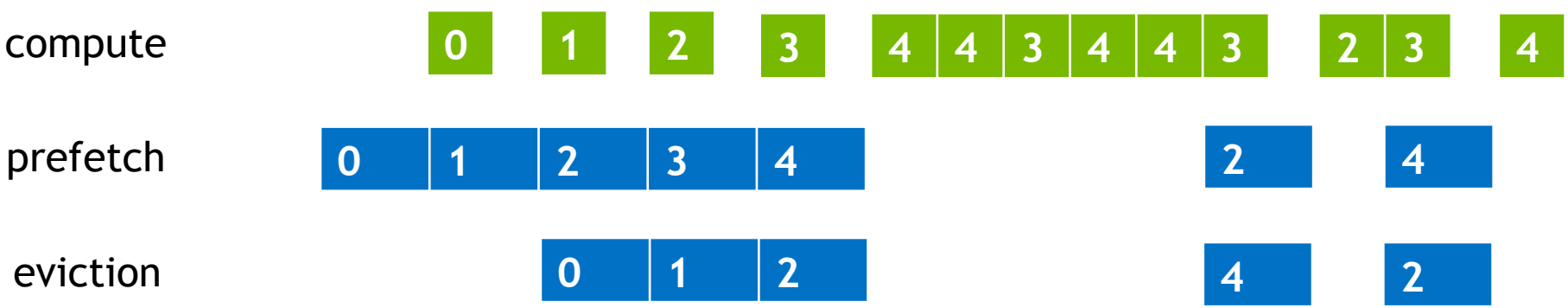


x86 CPU: Intel E5-2630 v3, 2 sockets of 10 cores each with HT on (40 threads)

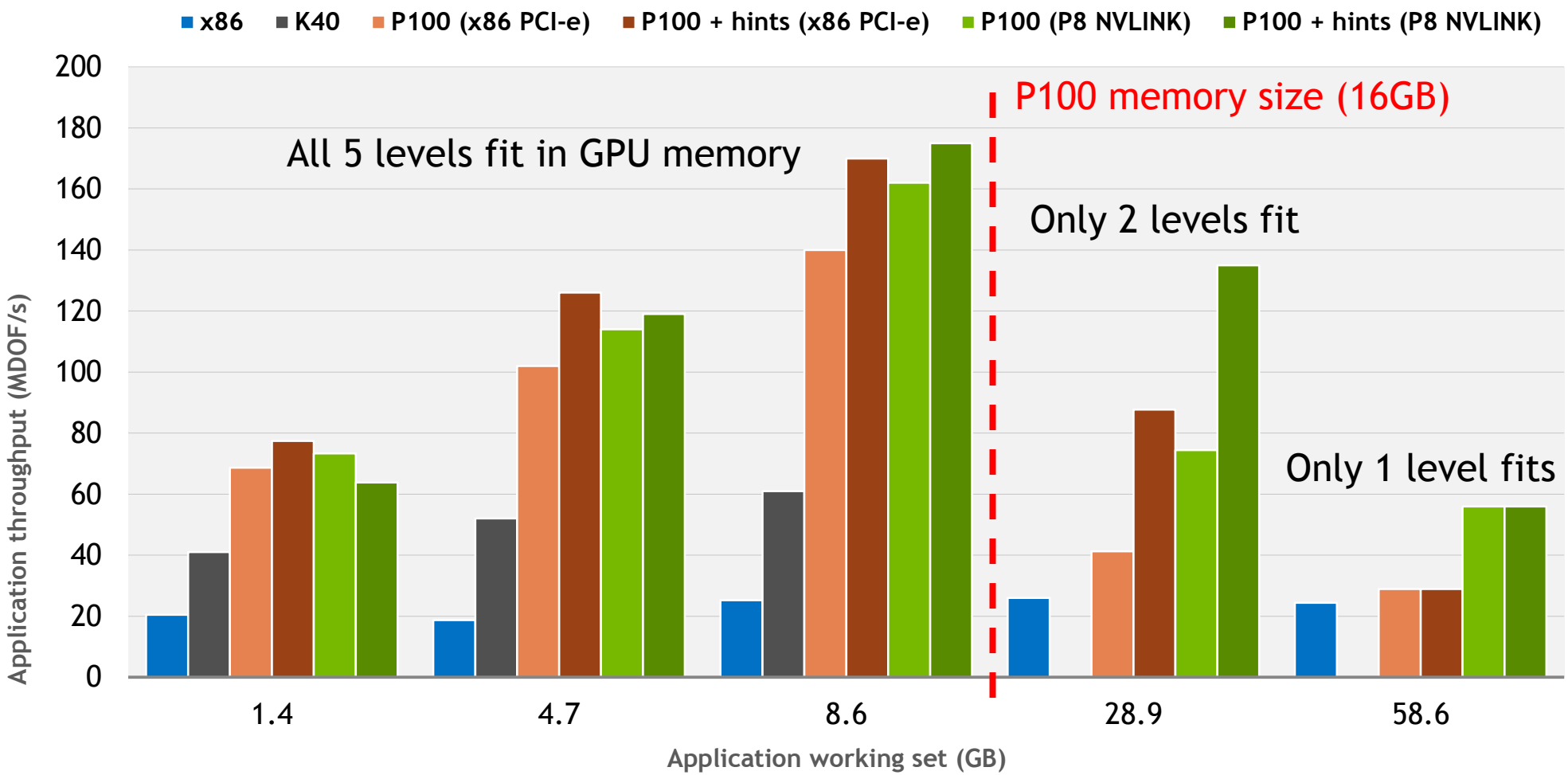
DATA PREFETCHING

Prefetch next level while performing computations on current level

Use cudaMemPrefetchAsync with non-blocking stream to overlap with default stream



RESULTS WITH USER HINTS

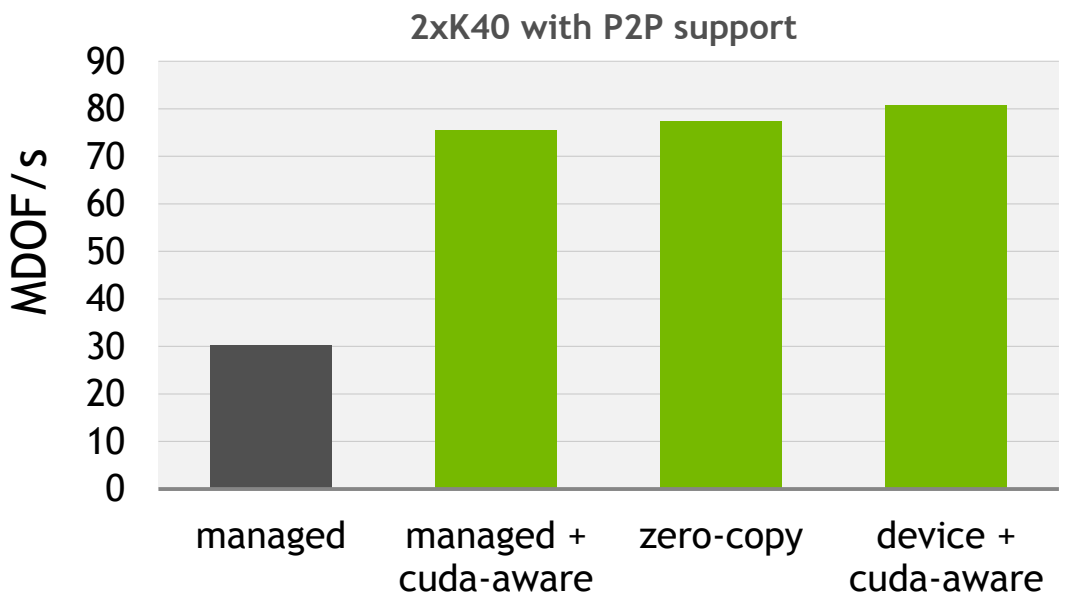


x86 CPU: Intel E5-2630 v3, 2 sockets of 10 cores each with HT on (40 threads)

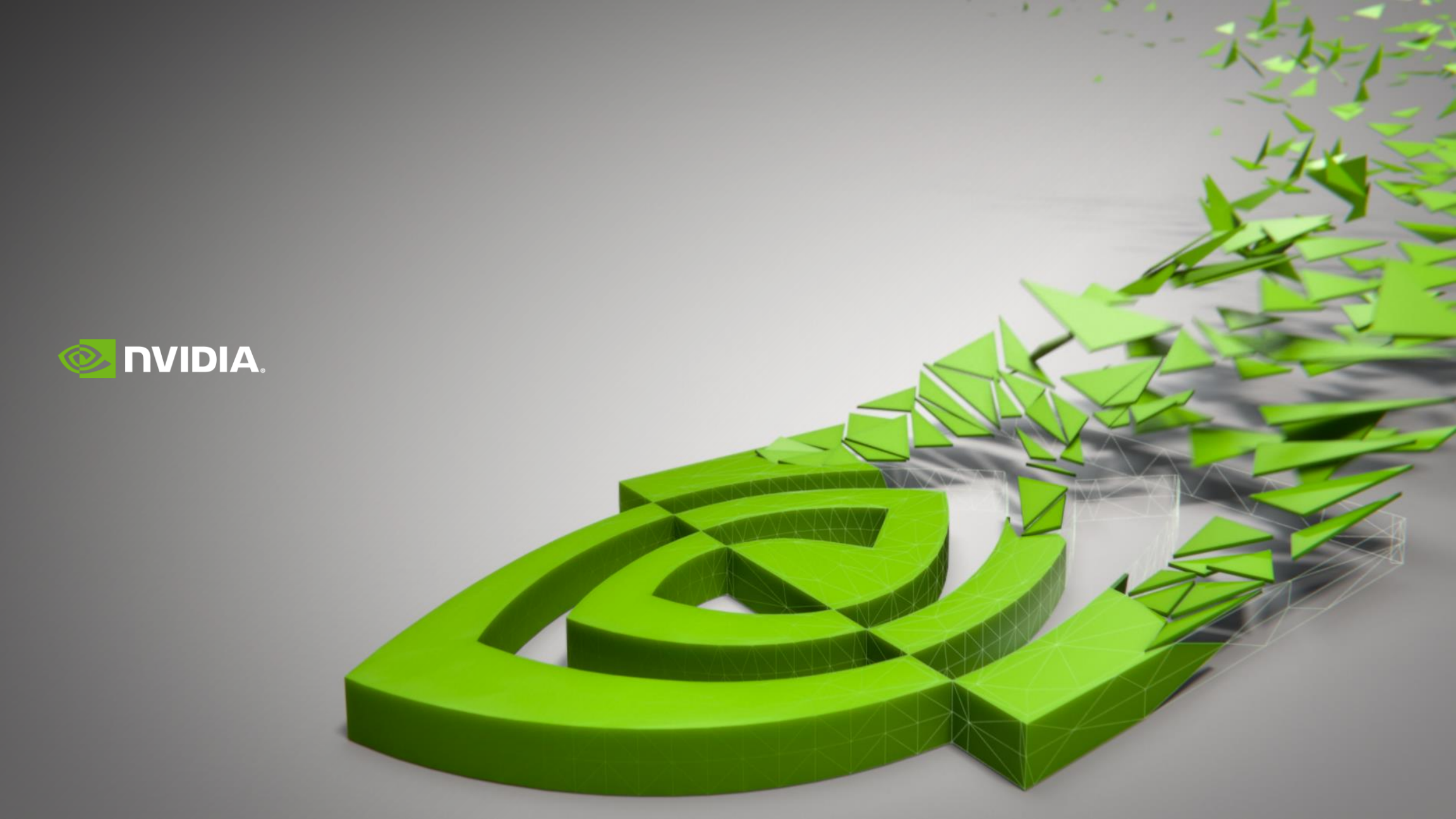
MULTI-GPU PERFORMANCE

MPI buffers can be allocated with cudaMalloc, cudaMallocHost, cudaMallocManaged

CUDA-aware MPI can stage managed buffers through system or device memory



Preliminary results on 8xP100:
5.2x scaling using host buffers
6.8x scaling with NVLINK


NVIDIA