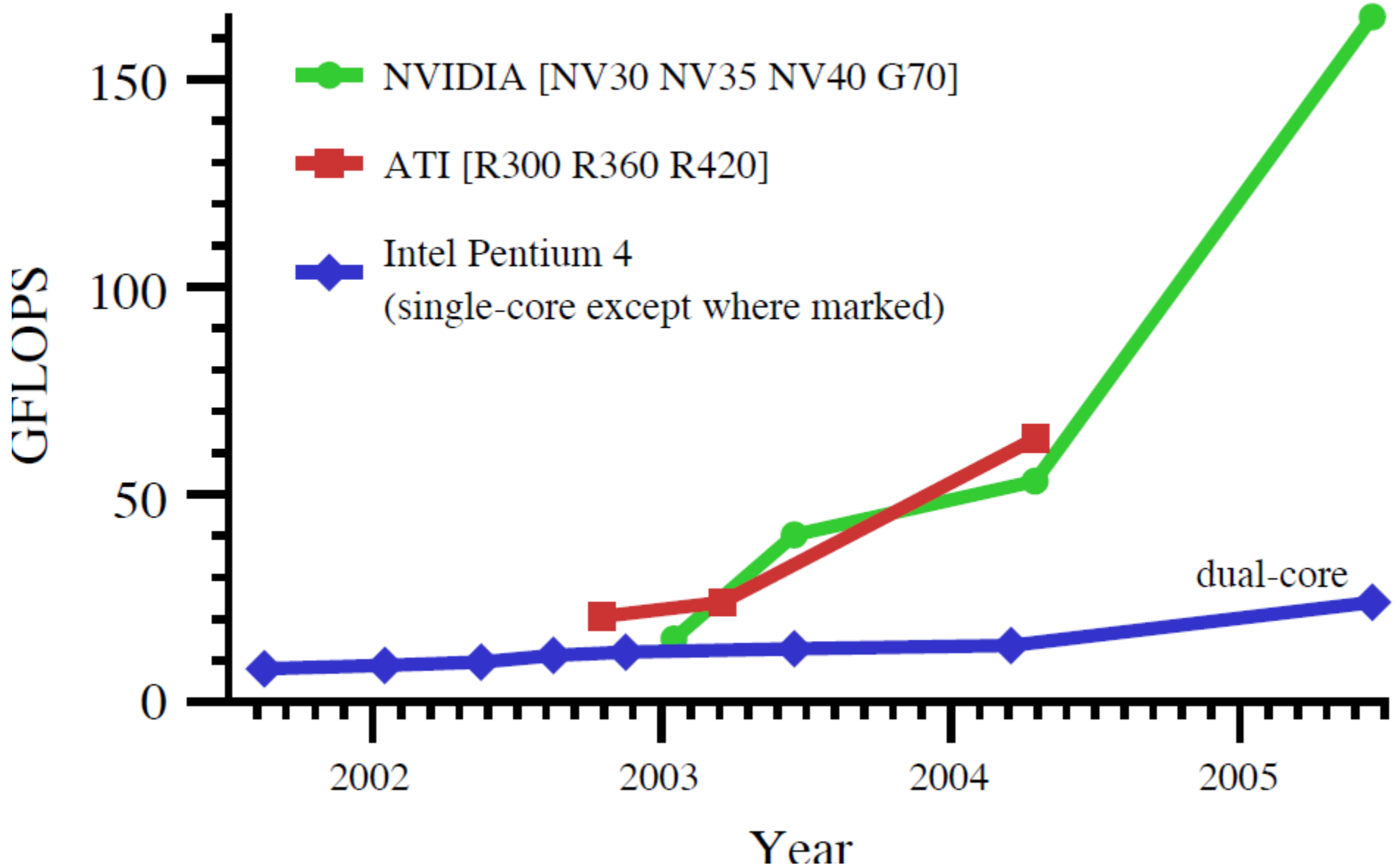
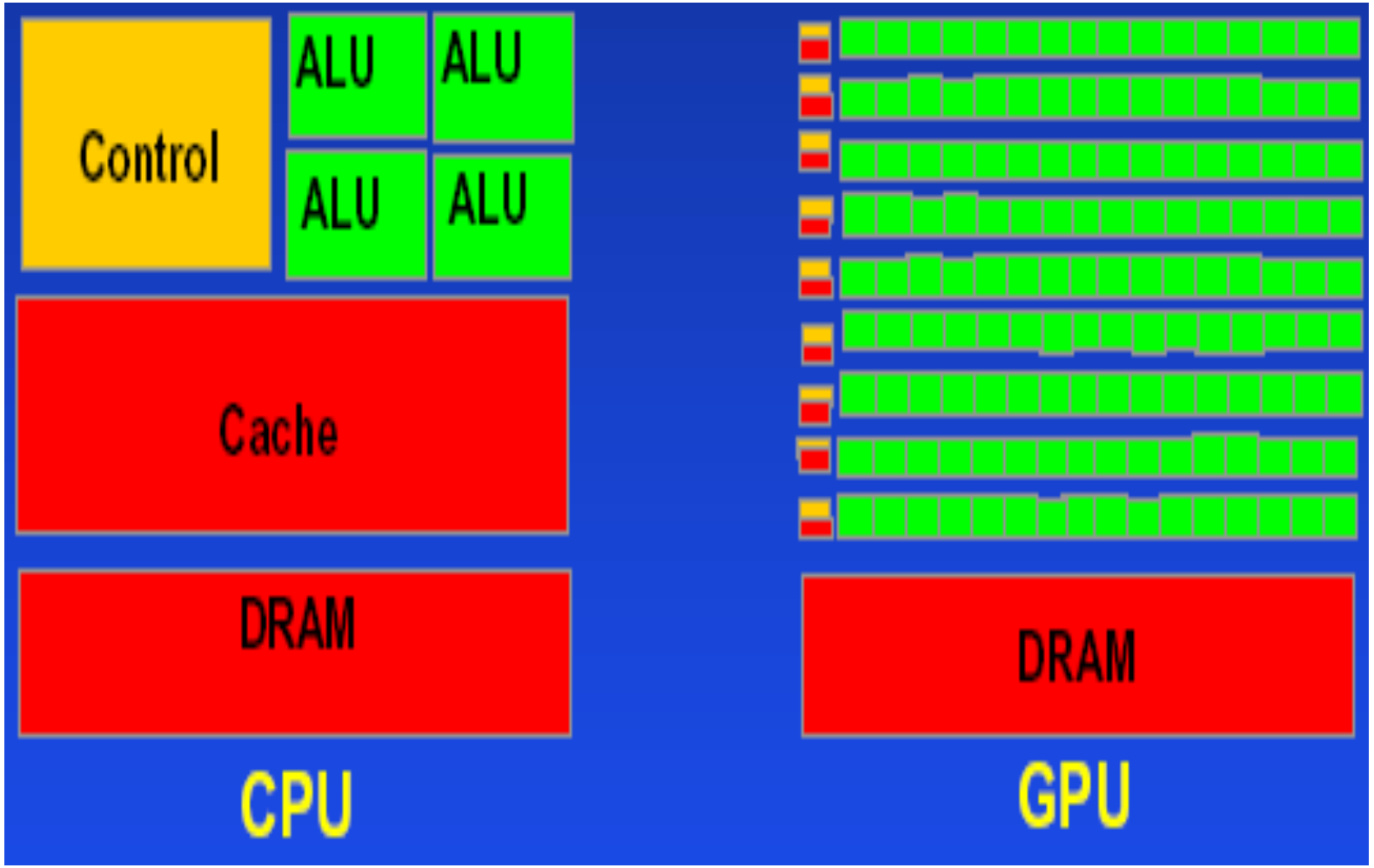
GPU Evolution


150
100
50
0
GFLOPS
NVIDIA [NV30 NV35 NV40 G70]
ATI [R300 R360 R420]
Intel Pentium 4
(single-core except where marked)
dual-core
2002
2003
2004
2005
Year

- Computing industry shifted to parallel computing
- Parallel computing no longer relegated to exotic super computers or mainframes
- Mobiles and music players begun to incorporate parallel computing capabilities to provide better functionalities
- Challenge for S/W developers to cope with variety of platforms and to provide novel and rich experience for an increasingly sophisticated base users
- CPU
  - Free lunch is over
  - 1000s of cores work in tandem for the supercomputers
  - Leading CPU manufacturers announced the arrival of 12 and 16-cores confirming parallel computing has arrived for good

- GPU History
  - Early1990s: graphically driven OS by microsoft helped create a market for new type of processor
  - User purchased 2D accelerator for PC
  - Silicon Graphics used 3D graphics in defense and scientific and technical visualization and stunning cinematic effects
  - 1992-silicon graphics released OpenGL library and the technology found its way into consumer applications. rapidly escalating the demand for 3D
  - First release of person games as Doom, Duke Nukem3D etc and release of affordable H/W
  - NVIDIA's GeForce 256 pushed the capabilities of consumer graphics by transform and lighting operation to be done on GPU..hence graphics pipeline
  - 2001..NVIDIA Geforce 3 series released…first to support DirectX 8.0, support vertex and fragment program. hence the programmer has control


Control
ALU
ALU
ALU
ALU
Cache
DRAM
CPU
DRAM
GPU

- Early GPU Computing

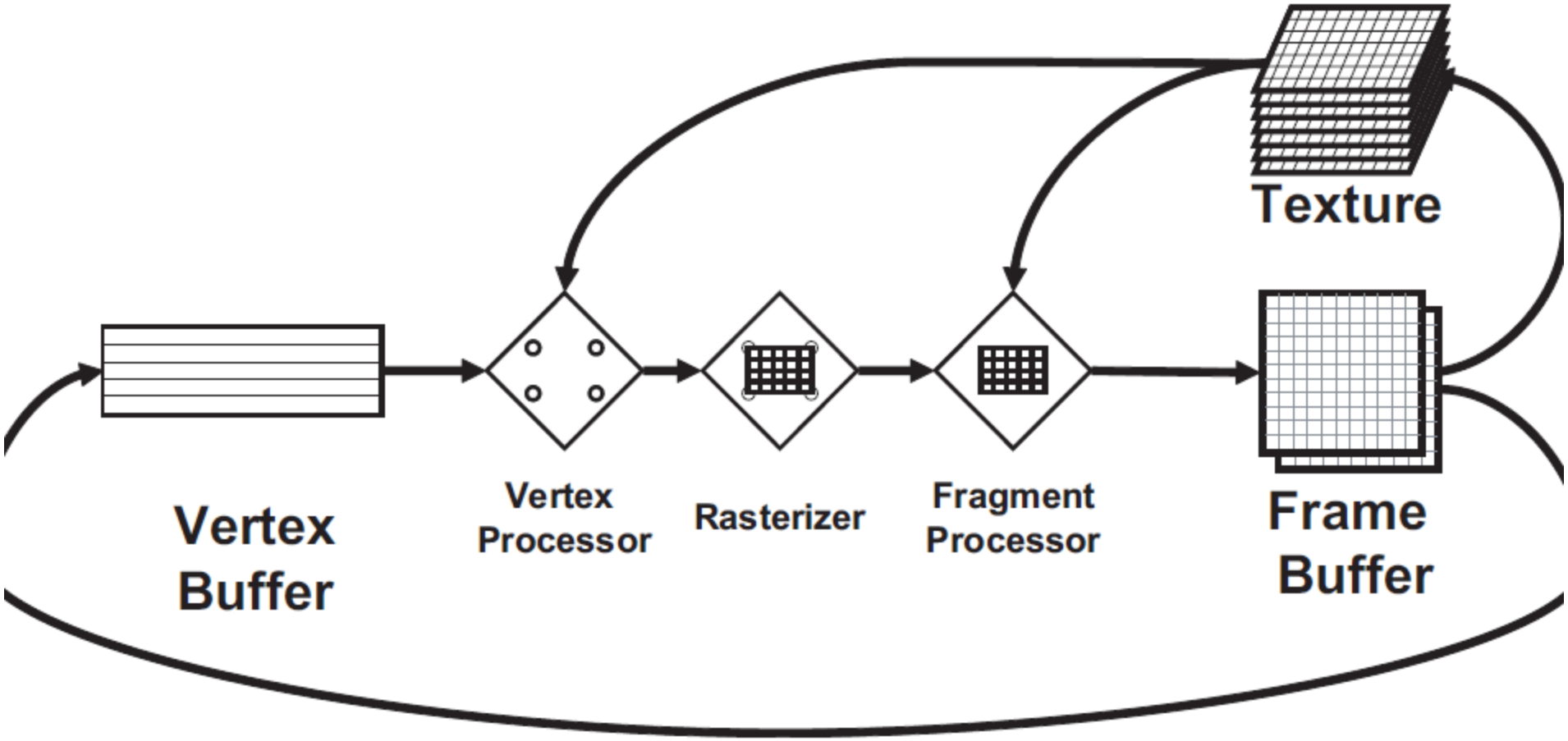
  - Programming GPUs allowed researchers to look into arbitrary computing. but still communication is through Graphics API only
  - 2000—GPUs started color computation by programmable arithmetic units known as pixel shaders
  - They use position etc to compute
  - And these input colors can be any data also
  - Results are handed over as final pixel color…hence GPU is being tricked to non rendering computations also
  - Initial throughput gave bright future. but programming model was a big obstacle (learning graphics is must)
  - No clue of floating point operations, system gets hung for errors
- After 5yrs of GeForce3 sereis, in 2006, NVIDIA unveiled the directX10 GPU..GeForce 8800GTX..first GPU to built with CUDA

Why GPGPU

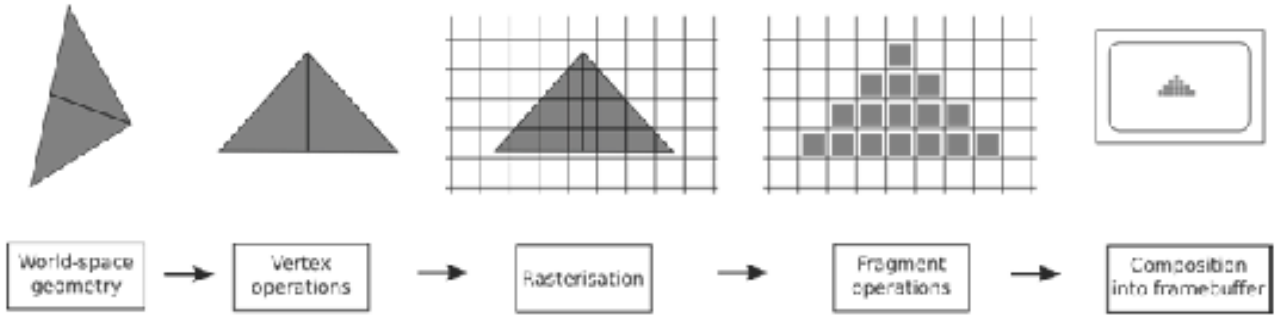
- Commodity computer graphics chips are today's most powerful computational hardware for the dollar
- Researchers and developers interested in harnessing the power of GPUs
- Recent years have seen explosion in interest in such research efforts
- Powerful and inexpensive
  - Computational capabilities: 1.7xpixels/sec to 2.3x vertices/sec, 1.4x for CPU
  - CPUs optimized for sequential code
- Flexible and programmable
  - Once Fixed function-8 bit per channel color values
  - Modern: fully programmable, IEEE floating point operations, programmability of vertex and pixel pipelines

GPGPU

- Limitations and difficulties
  - Pointer chasing not possible, dominated by mem comm and difficult to parallelize
  - Not suitable for crypto applications as do not support int based bit shift and logical operations
  - Intrinsic computer graphics hardware challenges and difficulties
  - Despite challenges. potential benefits are too large to ignore
- Overview of GPU hardware
  - Today’s commodity hardware structures its operation on Graphics pipeline


Texture
Vertex Buffer
Vertex Processor
Rasterizer
Fragment Processor
Frame Buffer

The graphics pipeline

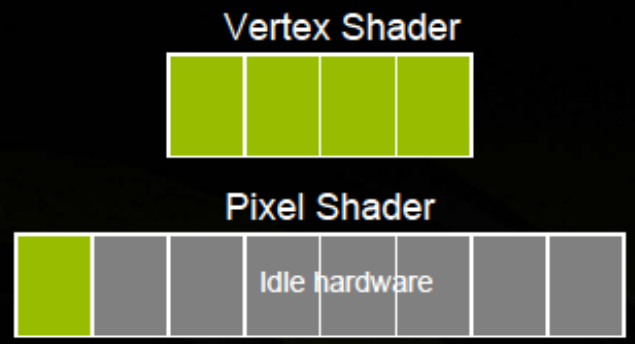


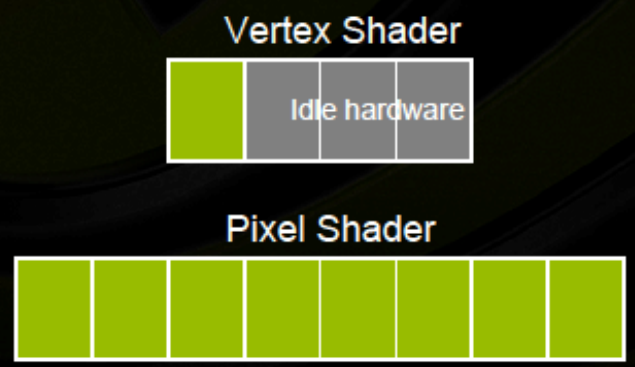
- Common abstraction of graphics workloads and hardware since 1992
- Exploits parallelism on all scales
- Maximizes throughput over latency

- CUDA

  - No partition of computing resources into vertex and fragment
  - Unified shader pipeline allowing every ALU on chip to be marshaled by a program intending to perform general purpose computation
  - Hence the compatibility with IEEE formats
  - Arbitrary access to Memory and shared memory
  - Using CUDA
    - Whatever features added still access is by OpenGL
    - Came out with standard C+ some keywords of CUDA architecture=CUDA C
    - First language to interact with GPU for GPGPU
    - Specialized hardware driver to exploit GPU as GPGPU

Why unify?



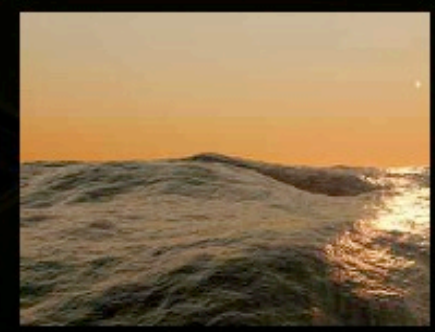


Heavy Geometry
Workload Perf = 4

Heavy Pixel
Workload Perf = 8

Why unify?

Heavy Geometry
Workload Perf = 11



Heavy Pixel
Workload Perf = 11