“Batch, Batch, Batch:” What Does It Really Mean?

Matthias Wloka

What Is a Batch?

- **Every DrawIndexedPrimitive() is a batch**
  - **Submits n number of triangles to GPU**
  - **Same render state applies to all tris in batch**
  - **SetState calls prior to Draw are part of batch**
- **Assuming efficient use of API**
  - **No Draw*PrimitiveUP()**
  - **DrawPrimitive() permissible if warranted**
  - **No unnecessary state changes**
- **Changing state means at least two batches**

Why Are Small Batches Bad?

- **Games would rather draw 1M objects/batches of 10 tris each**
  - **versus 10 objects/batches of 1M tris each**
- **Lots of guesses**
  - **Changing state inefficient on GPUs (WRONG)**
  - **GPU triangle start-up costs (WRONG)**
  - **OS kernel transitions (WRONG)**
- **Future GPUs will make it better!? Really?**

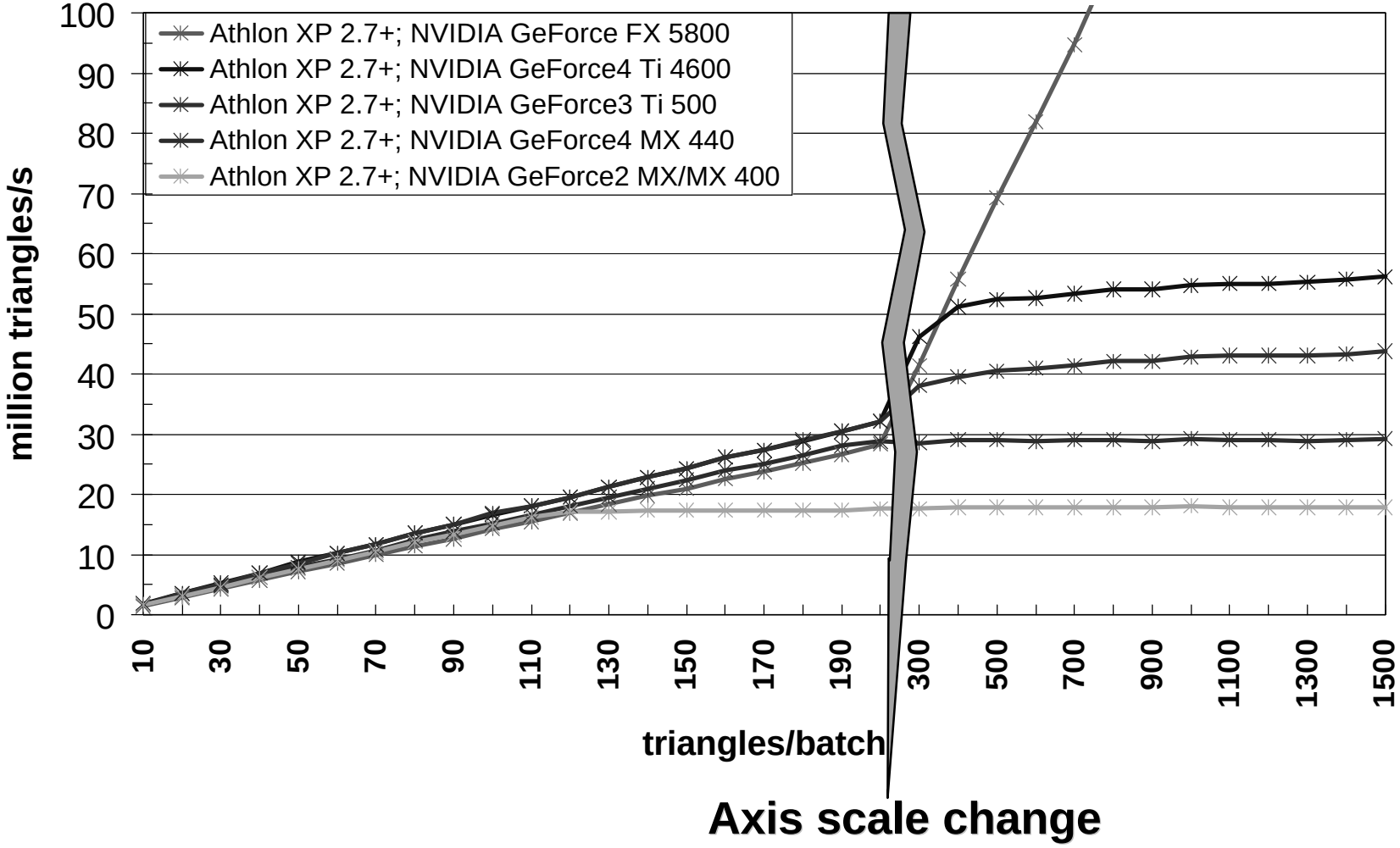
Let's Write Code!
Testing Small Batch Performance

- Test app does…
  - Degenerate triangles (no fill cost)
  - 100% PostTnL cache vertices (no xform cost)
  - Static data (minimal AGP overhead)
  - ~100k tris/frame, i.e., floor(100k/x) draws
  - Toggles state between draw calls:
    (VBs, w/v/p matrix, tex-stage and alpha states)

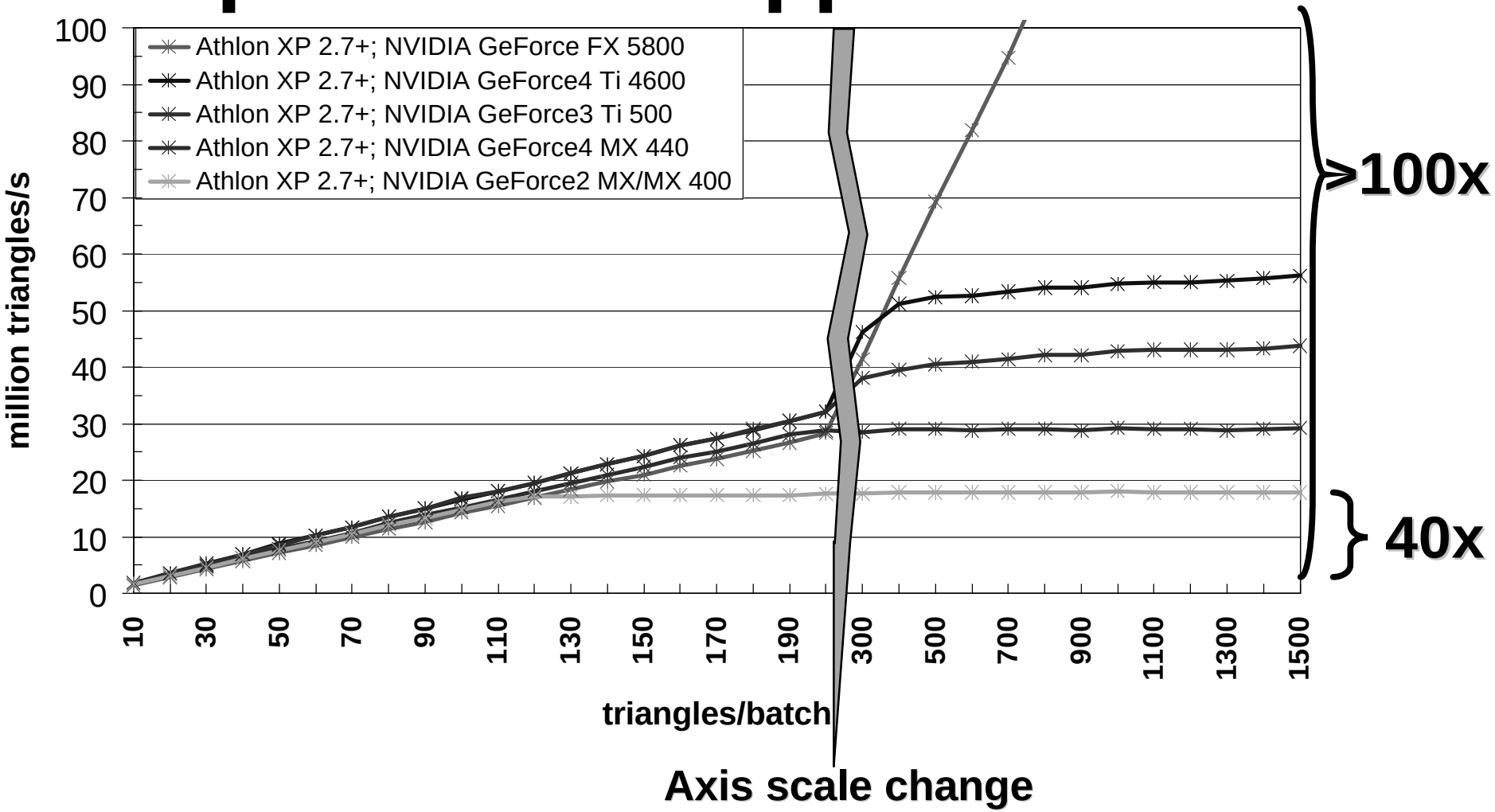
- Timed across 1000 frames

- Theoretical maximum triangle rates!

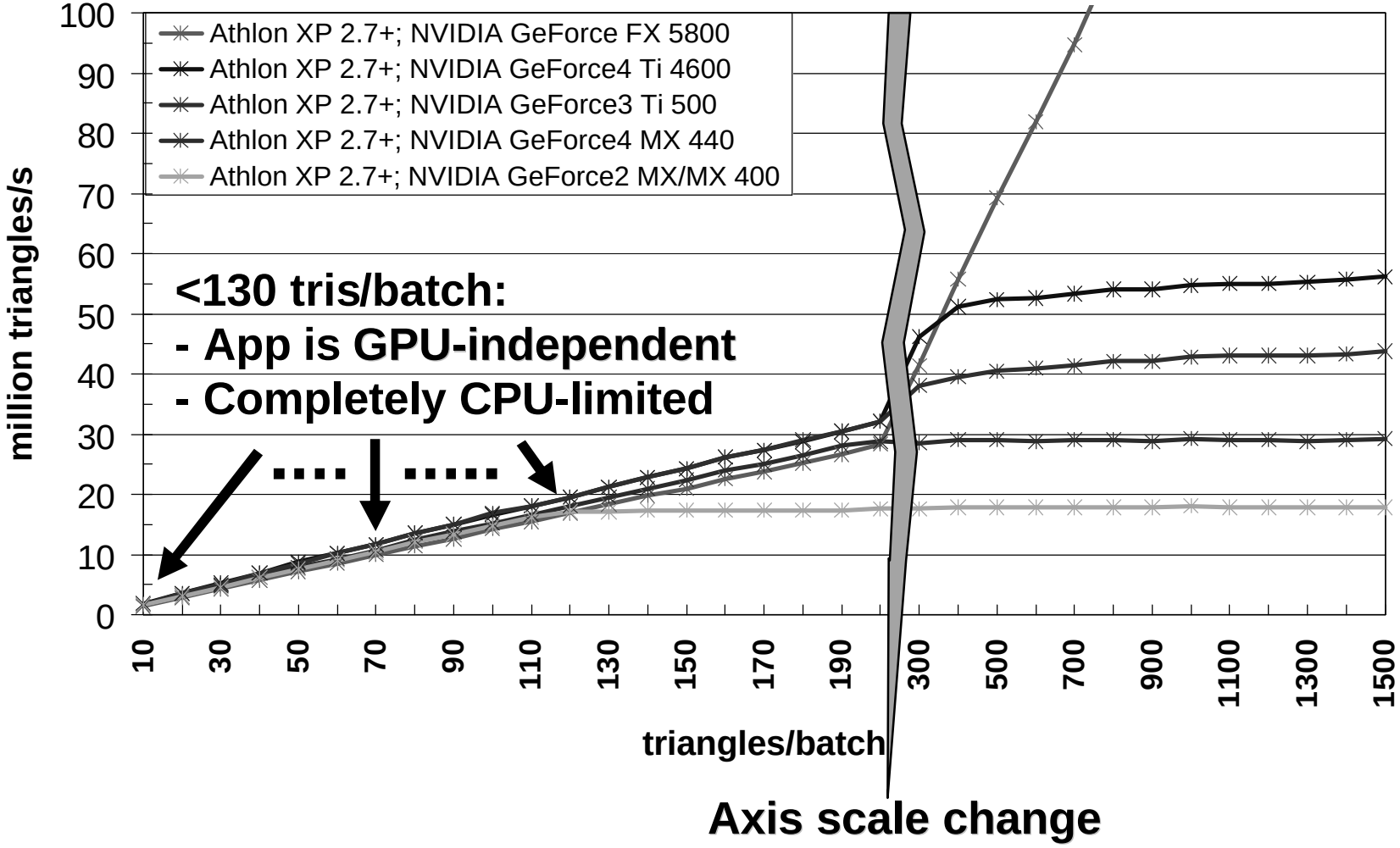
Measured Batch-Size Performance



Optimization Opportunities



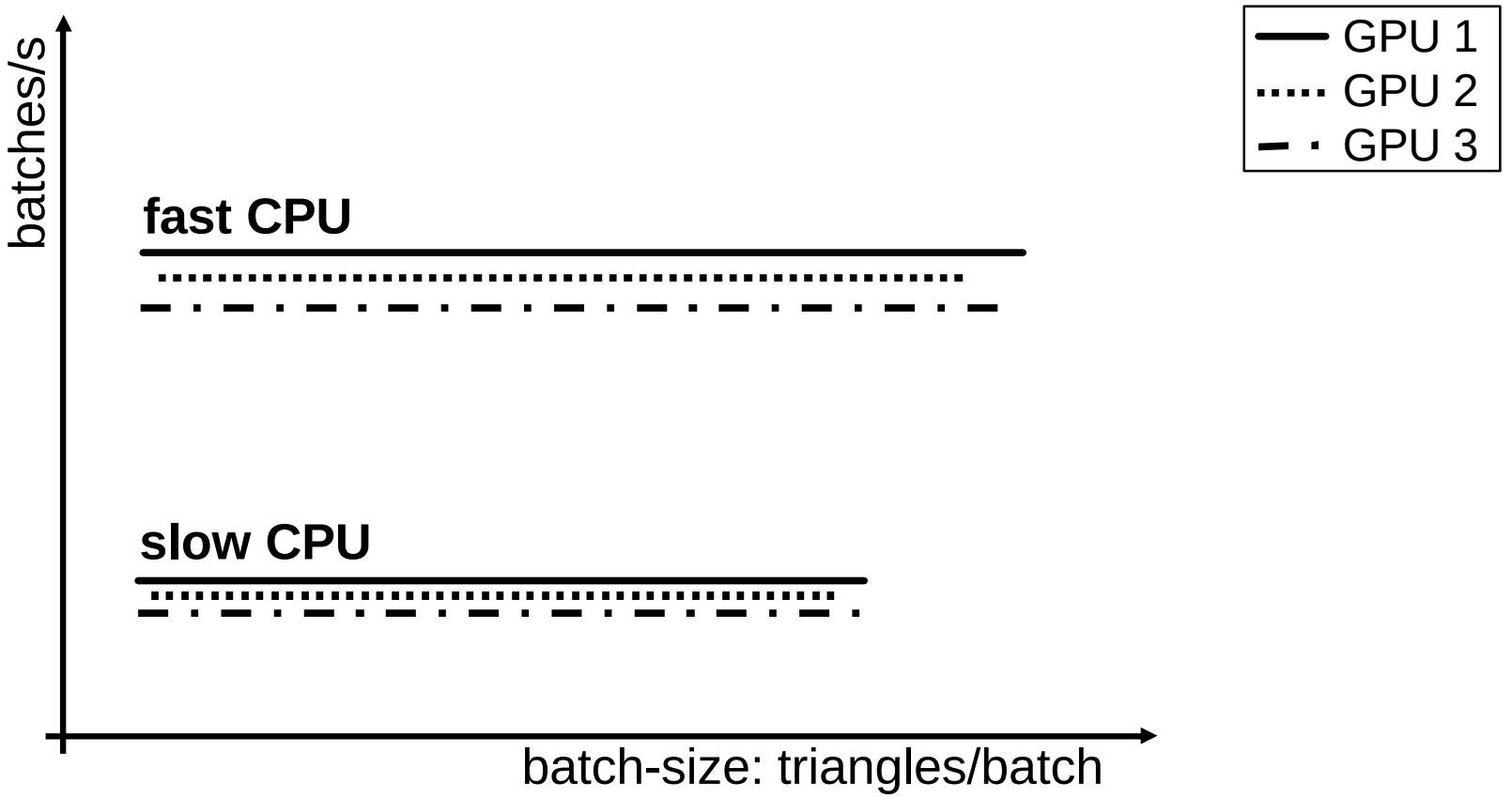
Measured Batch-Size Performance



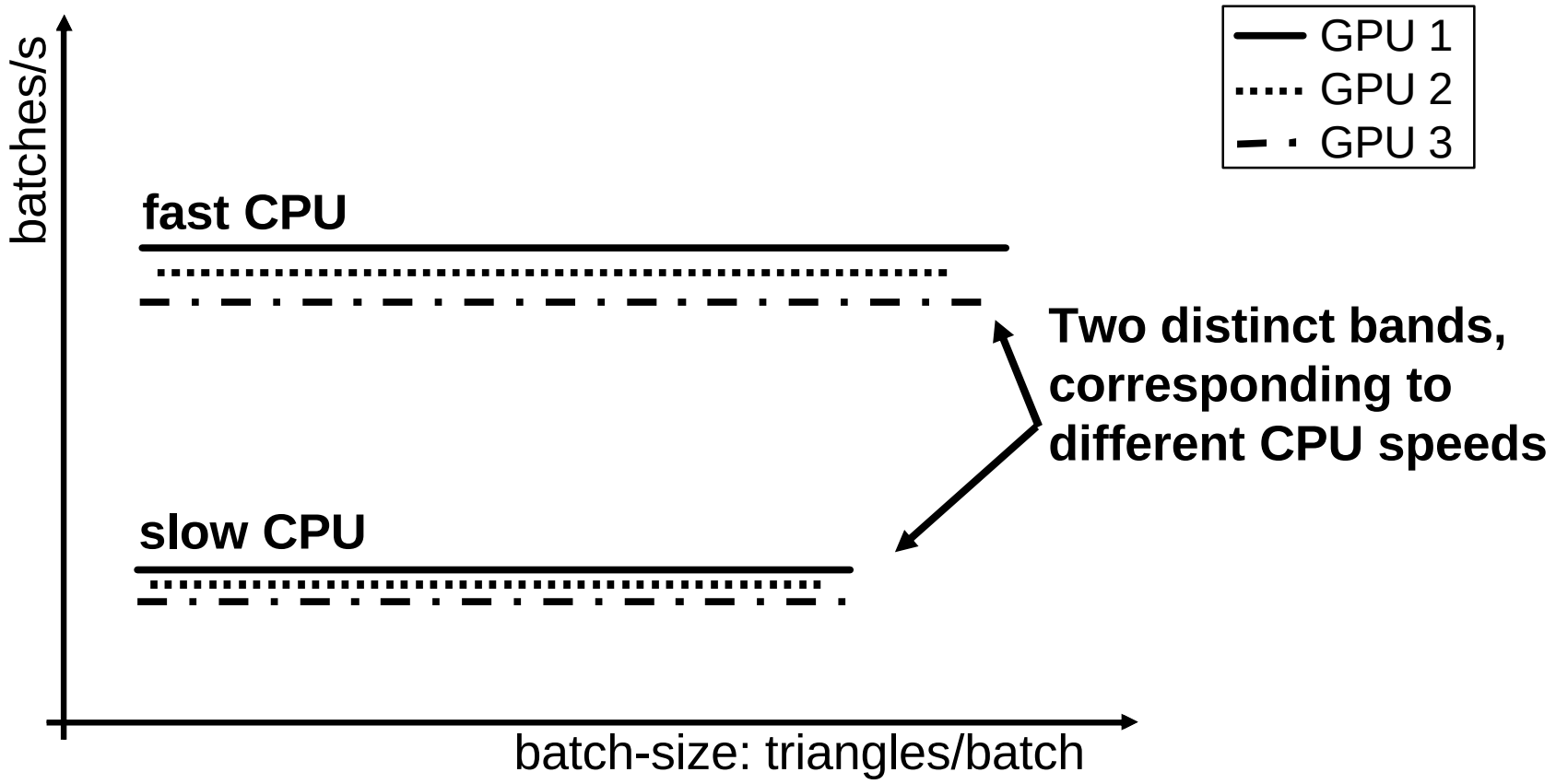
CPU-Limited?

- **Then performance results only depend on**
  - **How fast the CPU is**
    - **Not GPU**
  - **How much data the CPU processes**
    - **Not how many triangles per batch!**
- **CPU processes draw calls (and SetStates), i.e., batches**
- **Let's graph batches/s!**

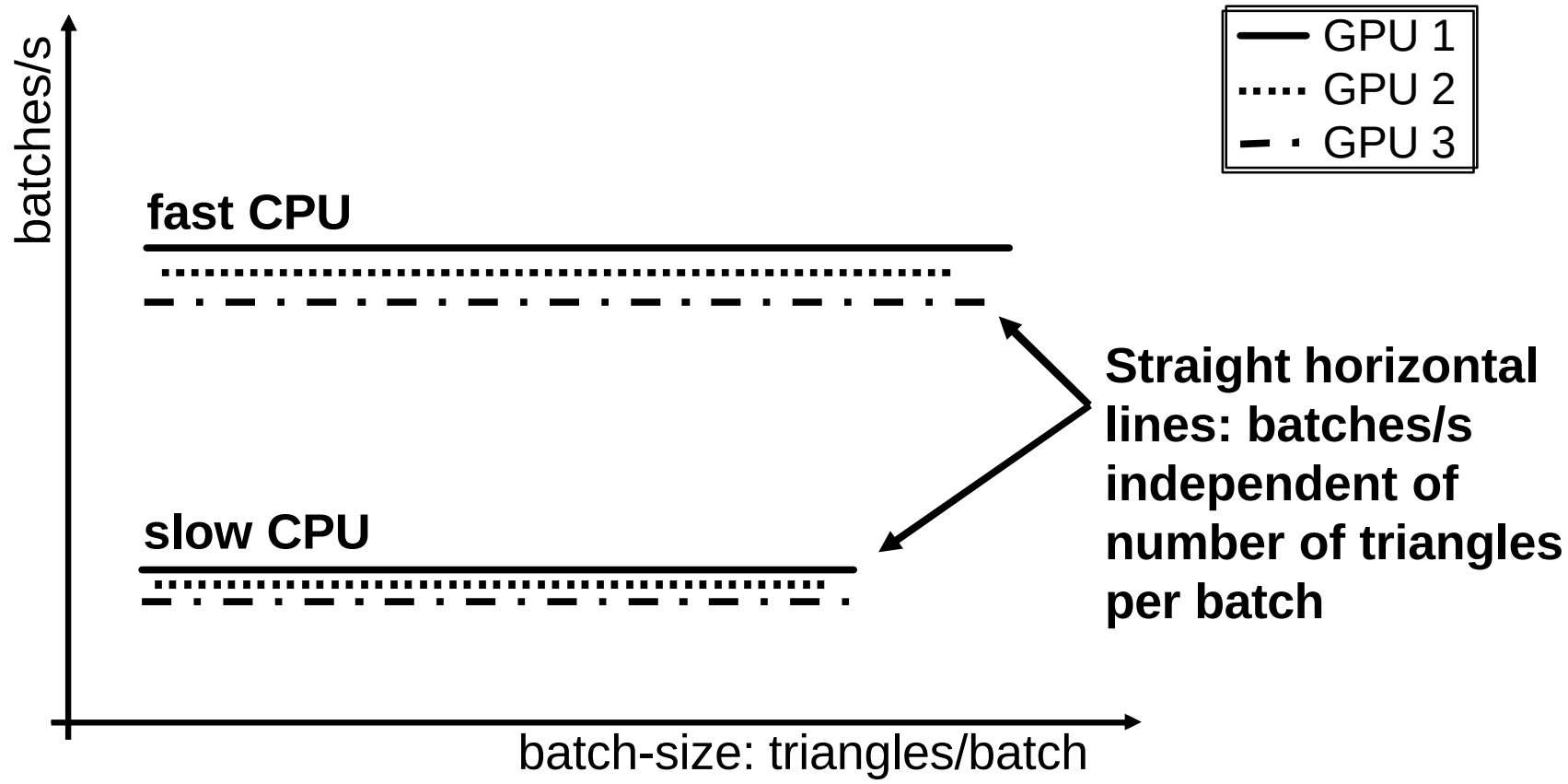
What To Expect If CPU Limited



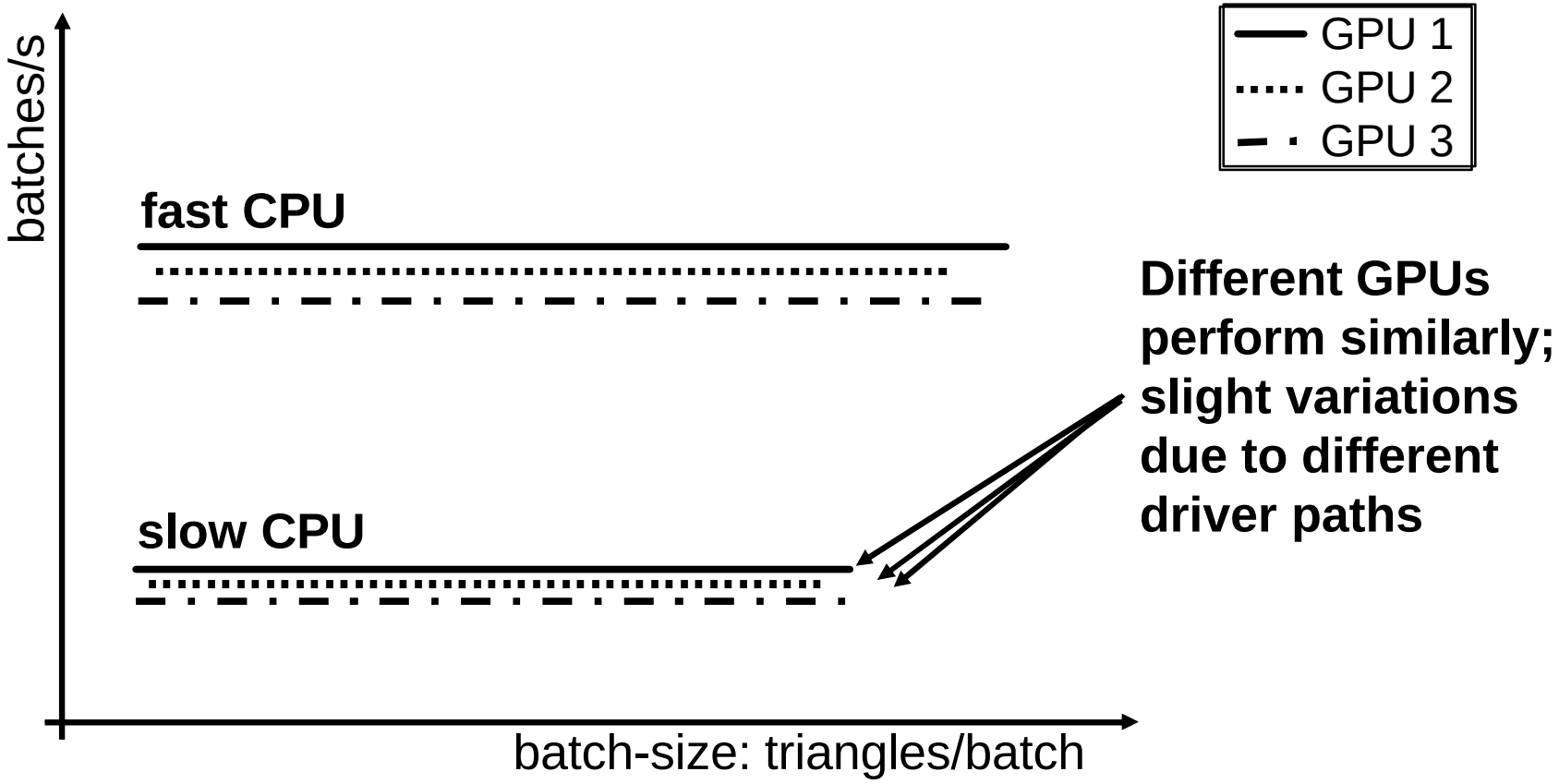
Effects of Different CPU Speeds



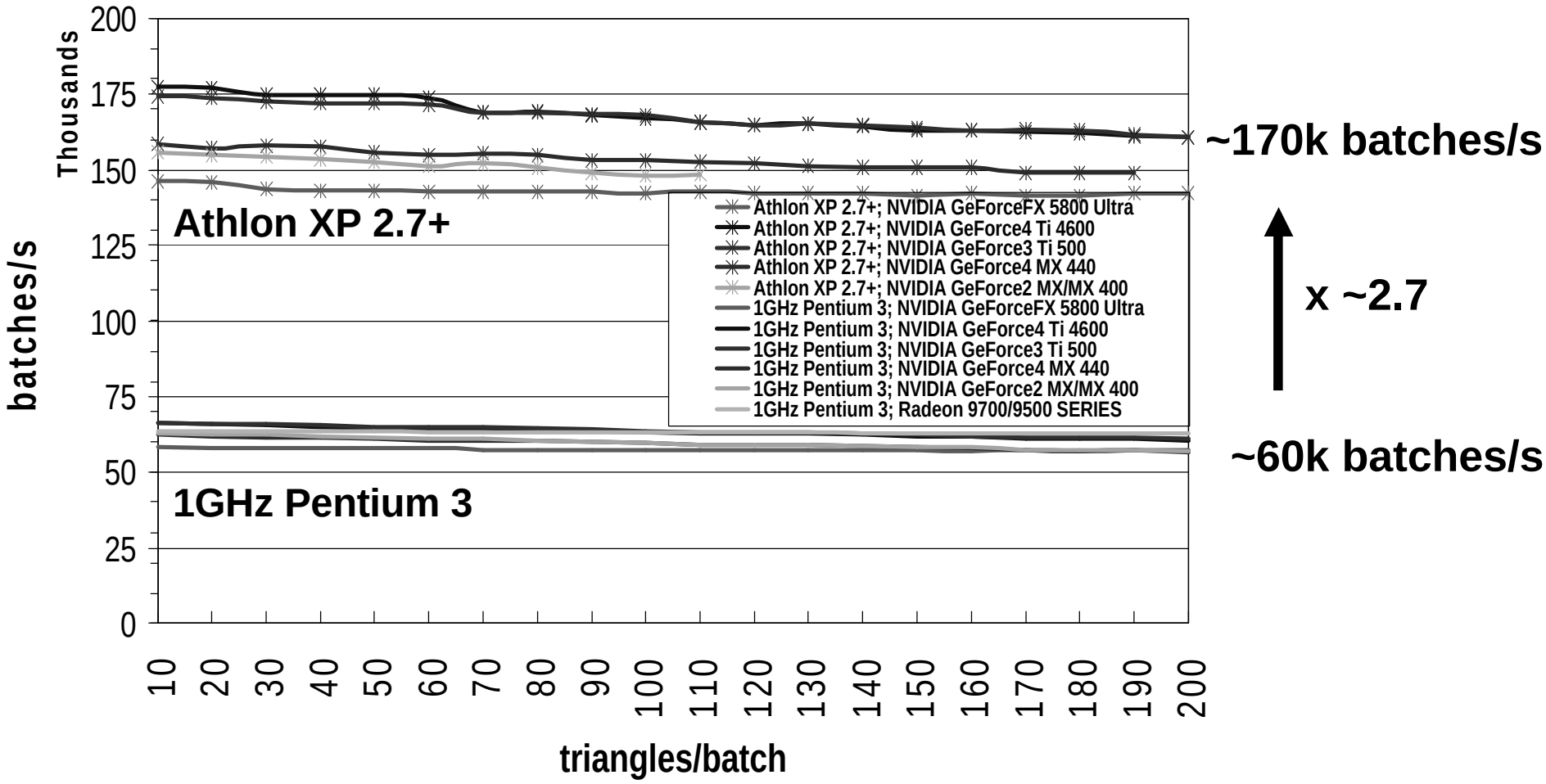
Effects of Number of Tris/Batch



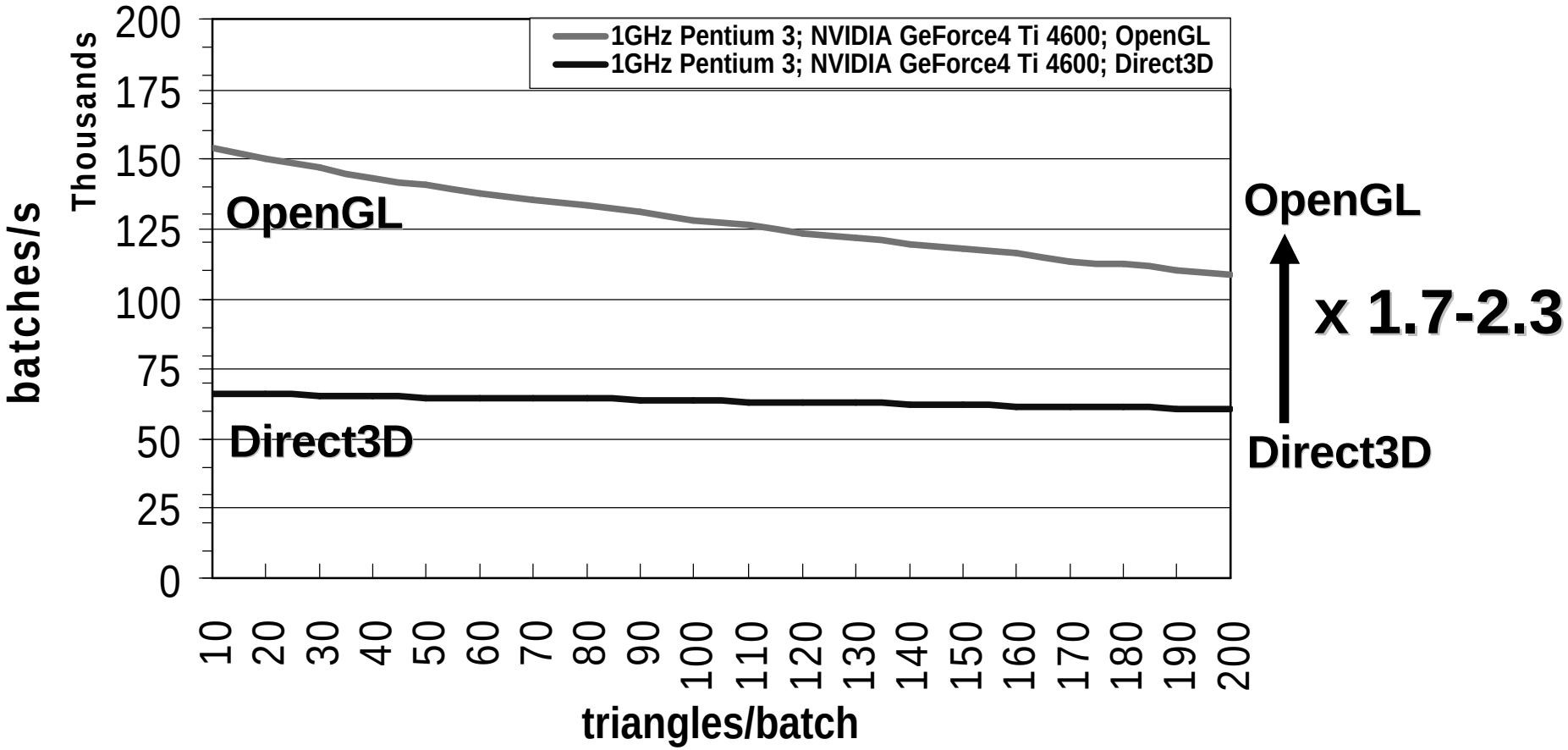
Effects of Different GPUs



Measured Batches Per Second



Side Note: OpenGL Performance



CPU Limited?

- **Yes, at < 130 tris/batch (avg) you are**
  - **completely,**
  - **utterly,**
  - **totally,**
  - **100%**
  - **CPU limited!**
- **CPU is busy doing nothing, but submitting batches!**

How ‘Real’ Is Test App?

- **Test app only does SetState, Draw, repeat;**
  - **Stays in CPU cache**
  - **No frustum culling, no nothing**
  - **So pretty much best case**
- **Test app changes arbitrary set of states**
  - **Types of state changes?**
  - **And how many states change?**
  - **Maybe real apps do fewer/better state changes?**

Real World Performance

- 353 batches/frame @ 16% 1.4GHz CPU: 26fps
- 326 batches/frame @ 18% 1.4GHz CPU: 25fps
- 467 batches/frame @ 20% 1.4GHz CPU: 25fps
- 450 batches/frame @ 21% 1.4GHz CPU: 25fps
- 700 batches/frame @ 100% (!) 1.5GHz CPU: 50fps
- 1000 batches/frame @ 100% (!) 1.5GHz CPU: 40fps
- 414 batches/frame @ 20% (?) 2.2GHz CPU: 27fps
- 263 batches/frame @ 20% (?) 3.0GHz CPU: 18fps
- 718 batches/frame @ 20% (?) 3.0GHz CPU: 21fps

Normalized Real World Performance

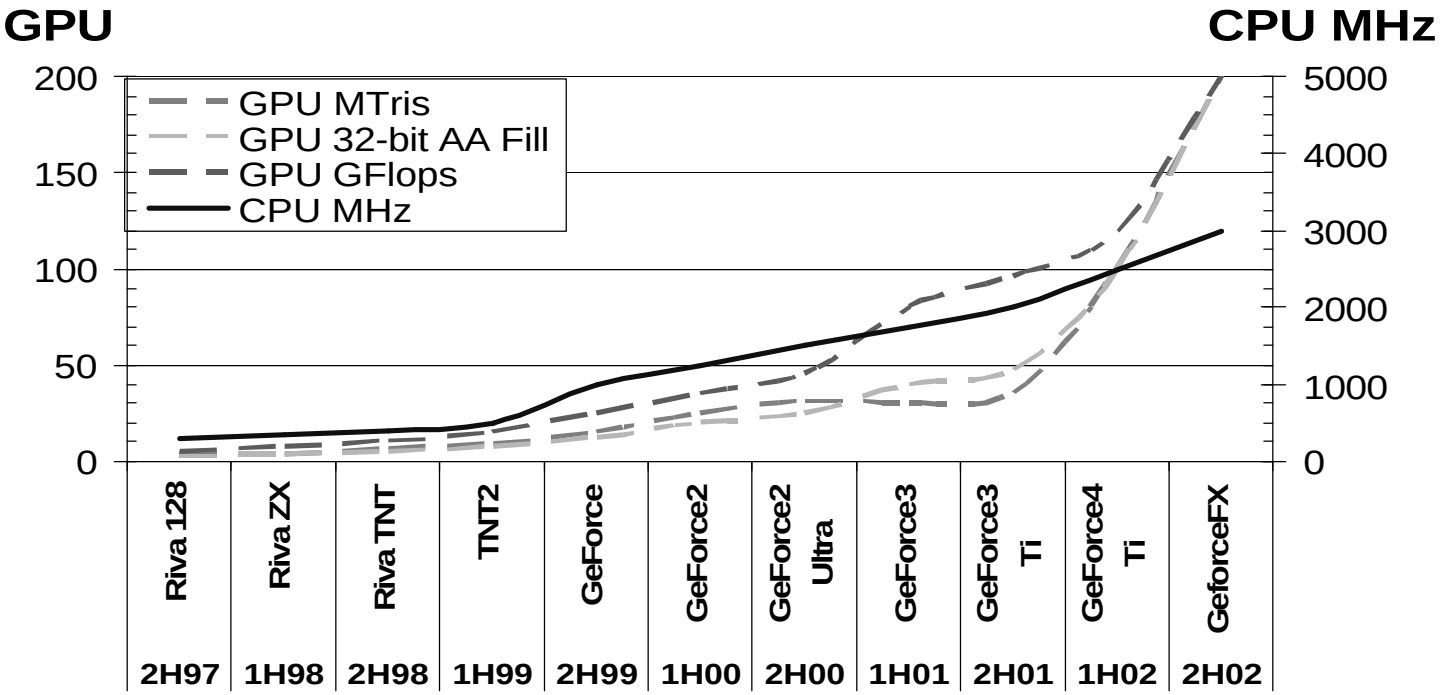
- ~41k batches/s @ 100% of 1GHz CPU
- ~32k batches/s @ 100% of 1GHz CPU
- ~42k batches/s @ 100% of 1GHz CPU
- ~38k batches/s @ 100% of 1GHz CPU
- ~25k batches/s @ 100% of 1GHz CPU
- ~25k batches/s @ 100% of 1GHz CPU
- ~25k batches/s @ 100% of 1GHz CPU
- ~ 8k batches/s @ 100% of 1GHz CPU
- ~25k batches/s @ 100% of 1GHz CPU

10k – 40k batches/s (100% 1GHz CPU)

Small Batches Feasible In Future?

- VTune (1GHz Pentium 3 w/ 2 tri/batch):
  - 78% driver; 14% D3D; 6% Other32; rest noise
- Driver doing little per Draw/SetState, but
  - Little times very large multiplier is still large
- Nvidia is optimizing drivers, but…
- Submitting X batches: O(X) work for CPU
  - CPU (game, runtime, driver) processes batch
  - Can reduce constants but not order O()

GPUs Getting Faster More Quickly Than CPUs



Avg. 18month CPU Speedup: 2.2

Avg. 18month GPU Speedup: 3.0-3.7

GPUs Continue To Outpace CPUs

- CPU processes batches, thus
  - Number of batches/frame MUST scale with:
    - Driver/Runtime optimizations
    - CPU speed increases
- GPU processes triangles (per batch), thus
  - Number of triangles/batch scales with:
    - GPU speed increases
- GPUs getting faster more quickly than CPUs
  - Batch sizes CAN increase

So, How Many Tris Per Batch?

- **500? 1000? It does not matter!**
  - **Impossible to fit everything into large batches**
  - **A few 2 tris/batch do NOT kill performance!**
  - **N tris/batch: N increases every 6 months**
- **I am a donut! Ask not how many tris/batch, but rather how many batches/frame!**
- **You get X batches per frame, depending on:**
  - **Target CPU spec**
  - **Desired frame-rate**
  - **How much % CPU available for submitting batches**

You get X batches per frame,

X mainly depends on CPU spec

What is X?

- **25k batches/s @ 100% 1 GHz CPU**
  - **Target: 30fps; 2GHz CPU; 20% (0.2) Draw/SetState:**
  - **X = 333 batches/frame**

- **Formula: 25k * GHz * Percentage/Framerate**
  - **GHz = target spec CPU frequency**
  - **Percentage = value 0..1 corresponding to CPU percentage available for Draw/SetState calls**
  - **Framerate = target frame rate in fps**

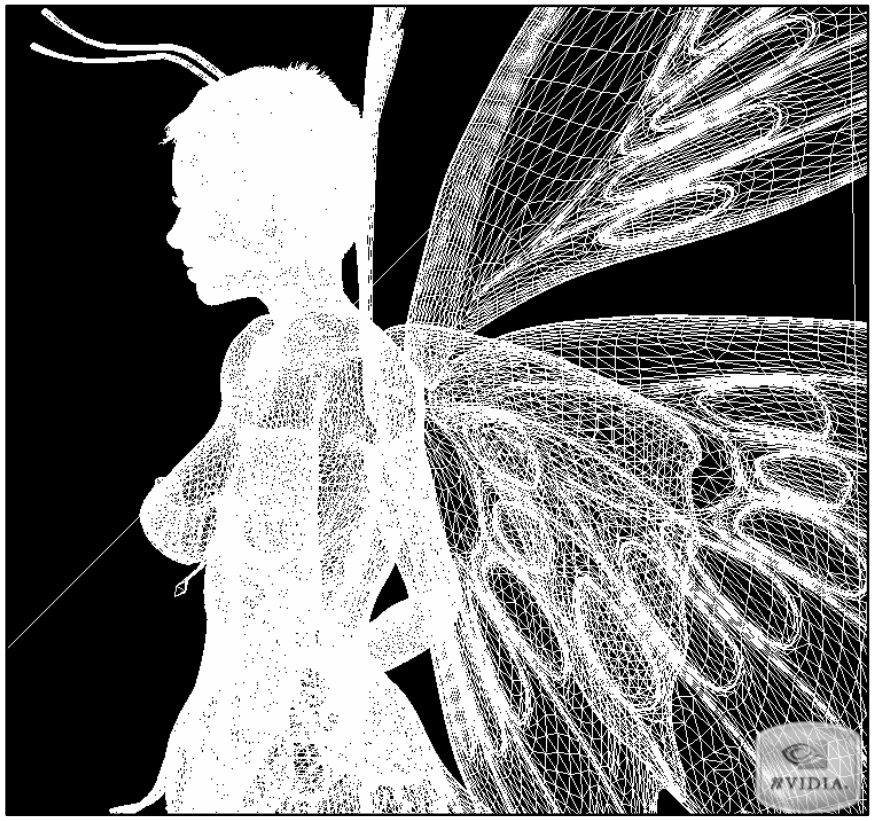
Please Hang Over Your Bed

25k batches/s @ 100%
1GHz CPU

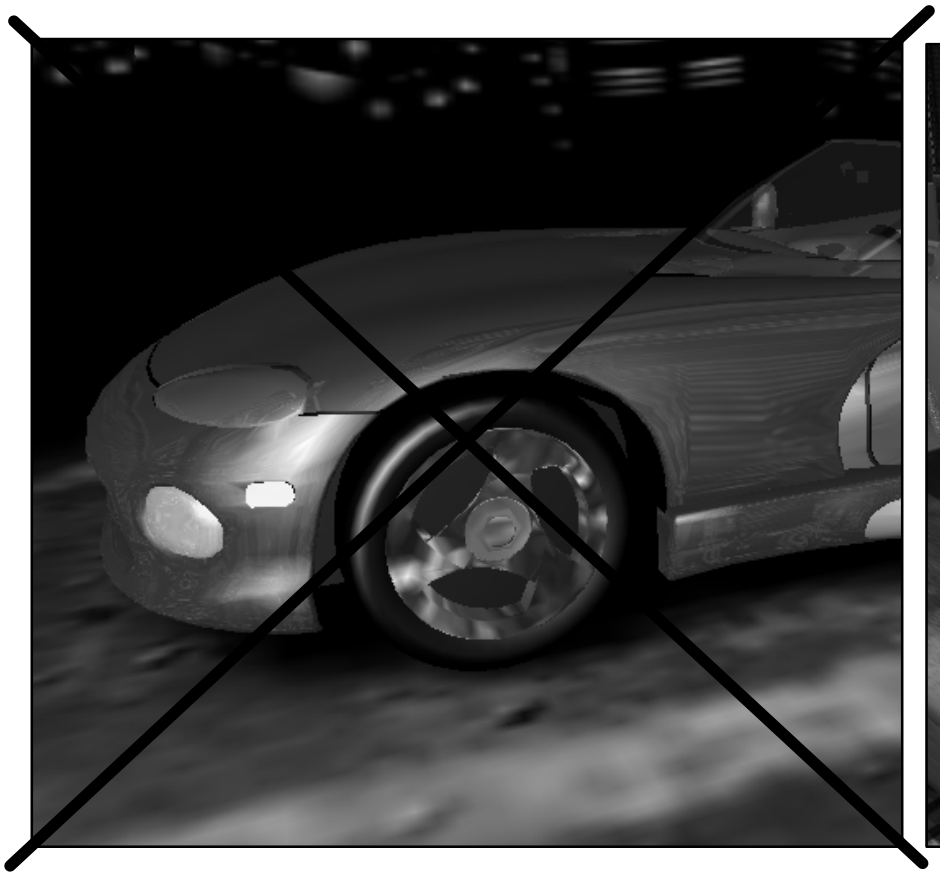
How Many Triangles Per Batch?

- **Up to you!**
  - **Anything between 1 to 10,000+ tris possible**
- **If small number, either**
  - **Triangles are large or extremely expensive**
  - **Only GPU vertex engines are idle**
- **Or**
  - **Game is CPU bound, but don't care because you budgeted your CPU ahead of time, right?**
  - **GPU idle (available for upping visual quality)**

GPU Idle? Add Triangles For Free!



GPU Idle?
Complicate Pixel Shaders For Free!



300 Batches Per Frame Sucks

- (Ab)use GPU to pack multiple batches together
- Critical NOW!
  - For increasing number of objects in game world
- Will only become more critical in the future

Batch Breaker: Texture Change

- **Use all of Geforce FX's 16 textures**
  - **Fit 8 distinct dual-textured batches into 1 single batch**
- **Pack multiple textures into 1 surface**
  - **Works as long as no wrap/repeat**
  - **Requires tool support**
  - **Potentially wastes texture space**
  - **Potential problems w/ multi-sampling**

Batch Breaker: Transform Change

- **Pre-transform static geometry**
  - **Once in a while**
  - **Video memory overhead: model replication**
- **1-Bone matrix palette skinning**
  - **Encode world matrix as 2 float4s**
    - **axis/angle**
    - **translate/uniform scale**
  - **Video memory overhead: model replication**
- **Data-dependent vertex branching**
  - **Render variable # of bones/lights in one batch**

Batch Breaker: Material Change

- **Compute multiple materials in pixel-shaders**
  - **Choose / Interpolate based on**
    - **Per-vertex attribute**
    - **Texture-map**
- **More performance optimization tips and tricks:**

Friday 3:00pm

“Graphics Pipeline Performance”
C. Cebenoyan and M. Wloka

But Only High-End GPUs Have That Feature!?

- Yes, but high-end GPUs most likely CPU-bound
- High-End GPUs most suited to deal with:
  - Longer vertex-shaders
  - Longer pixel-shaders
  - More texture accesses
  - Bigger video memory requirements
- To improve batching

But These Things Slow GPU Down!?

- **Remember: CPU-limited**
  - **GPU is mostly idle**
- **Making GPU work, so CPU does NOT**
- **Overall effect: faster game**

25k batches/s @ 100%
1GHz CPU

Acknowledgements

- **Many thanks to**

  Gary McTaggart, Valve

  Jay Patel, Blizzard
  Tom Gambill, NCSoft
  Scott Brown, NetDevil
  Guillermo Garcia-Sampedro, PopTop

Questions, Comments, Feedback?

- **Matthias Wloka: mwloka@nvidia.com**
- **http://developer.nvidia.com**

Can You Afford to Loose These Speed-Ups?

- 2 tris/batch
  - Max. of ~0.1 MTriangles/s for 1GHz Pentium 3
    - Factor 1500x away from max. throughput
  - Max. of ~0.4 MTriangles/s for Athlon XP 2.7+
    - Factor 375x away from max. throughput