Panel: Future Visualization Platform

William R. Mark
University of Texas at Austin

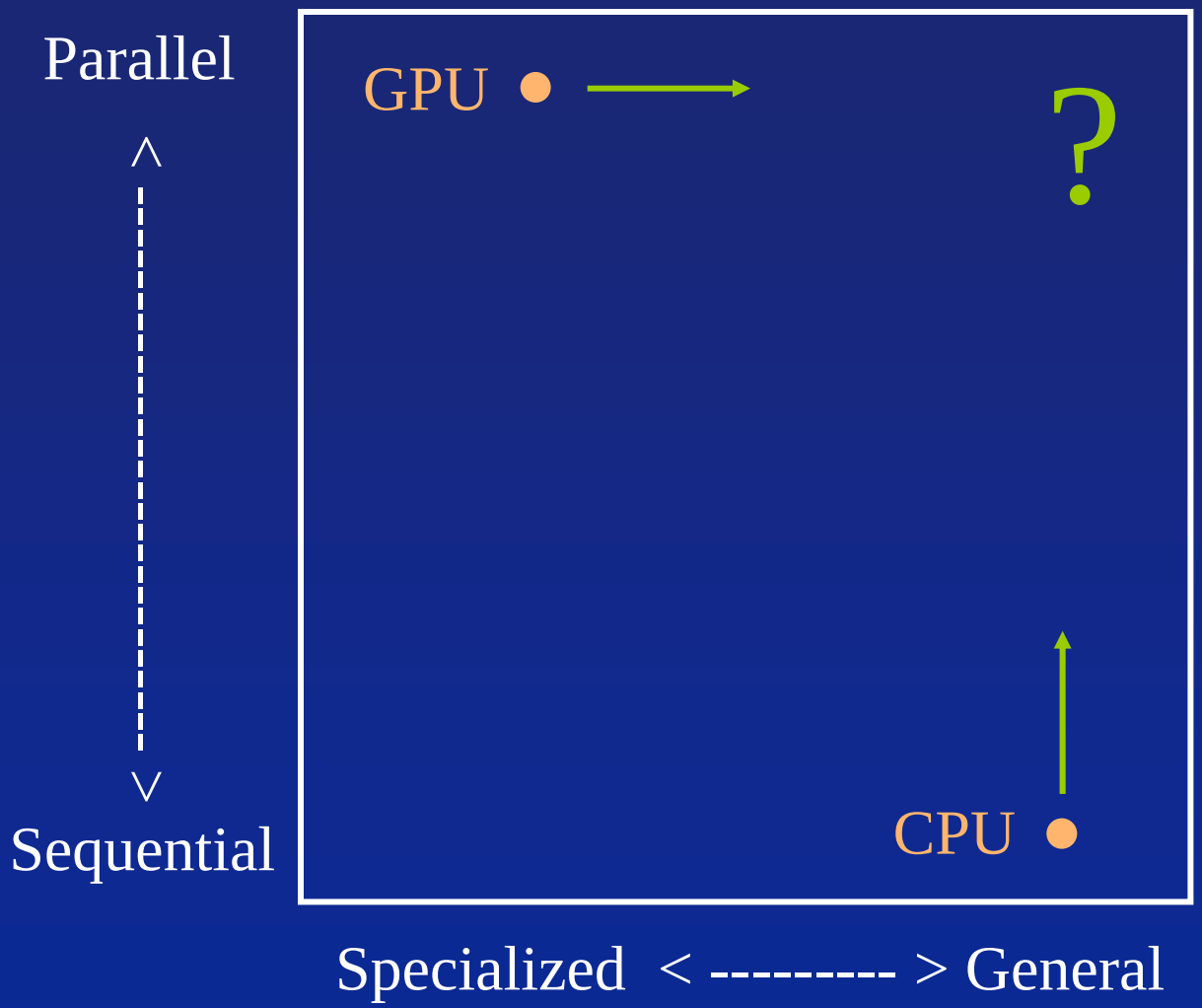
What do we want from our Vis “plumbing”?

- Easy to use and program
- High performance
- Reliable
- Permit focus on the task, not the plumbing

Despite recent progress, we’re not there yet…

Lots of papers about contorting GPU’s to do something that should be trivial, but isn’t.

The current hardware plumbing



Need at least two parallel programming models

- Stream model
  - Naturally exposes parallelism and communication
  - Easy to use, when problem maps well
- Communicating sequential processes (e.g. pthreads)
  - Explicitly exposes spatial dimension of HW parallelism
  - Efficiently supports data-dependent communication patterns
  - Useful for creating/modifying large irregular data structures
  - Harder to use – e.g. race conditions
  - Hard to get performance portability

The return of “software rendering” for vis

- Flexibility in choosing vis/rendering algorithms:
  - Choose your visibility algorithm… raycasting? OK!
  - Choose your scattering model
  - Choose your compression algorithms
  - Etc.
- Combine simulation with rendering
  - Render directly from the simulation data structures

Reality check

- Progress will be gradual
- Hardware and vis/rendering algorithms must co-evolve
- Now is the time to think about where we want to end up
  - What: HW, programming models, algorithms
  - Why: So we don't get stuck with a badly sub-optimal approach
  - Must co-design HW and algorithms

2-year predictions

- CPU's: multi-core trend accelerates
  - Multicore used by games and HPC
- GPU's: More powerful streaming model
  - Scatter, gather, conditional streams, reductions, etc.
  - Start to see more success stories for GPGPU
  - But limits of stream model become apparent
- "Dark Horses" attract increasing attention
  - CELL and others

6-year predictions

- One processing chip for PC's
  - Who makes it?
- Heterogeneous architecture for this chip:
  - Classical CPU
  - Parallel fine-grained shared memory (pthreads)
  - Parallel stream processor (Brook)
- Supports ray-casting visibility
- This architecture emerges in console space first
- This architecture meets many HPC needs