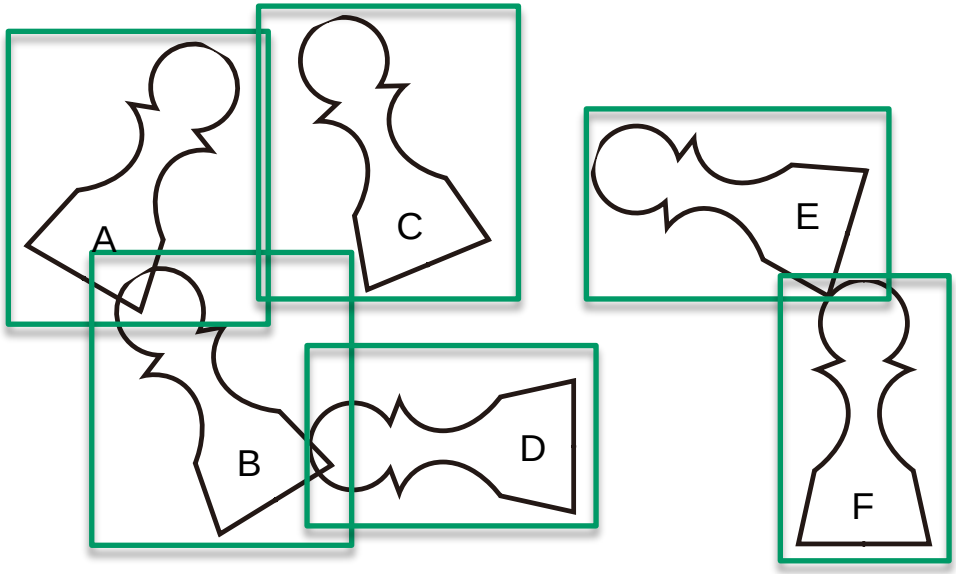
Accelerating Rigid Body Simulation on Today's GPUs

Takahiro Harada
takahiro.harada@amd.com

RIGID BODY SIMULATION PIPELINE

- Broad phase collision detection
  - Quick check using bounding volumes
- Narrow phase collision detection
  - Detailed check using geometry
- Solve



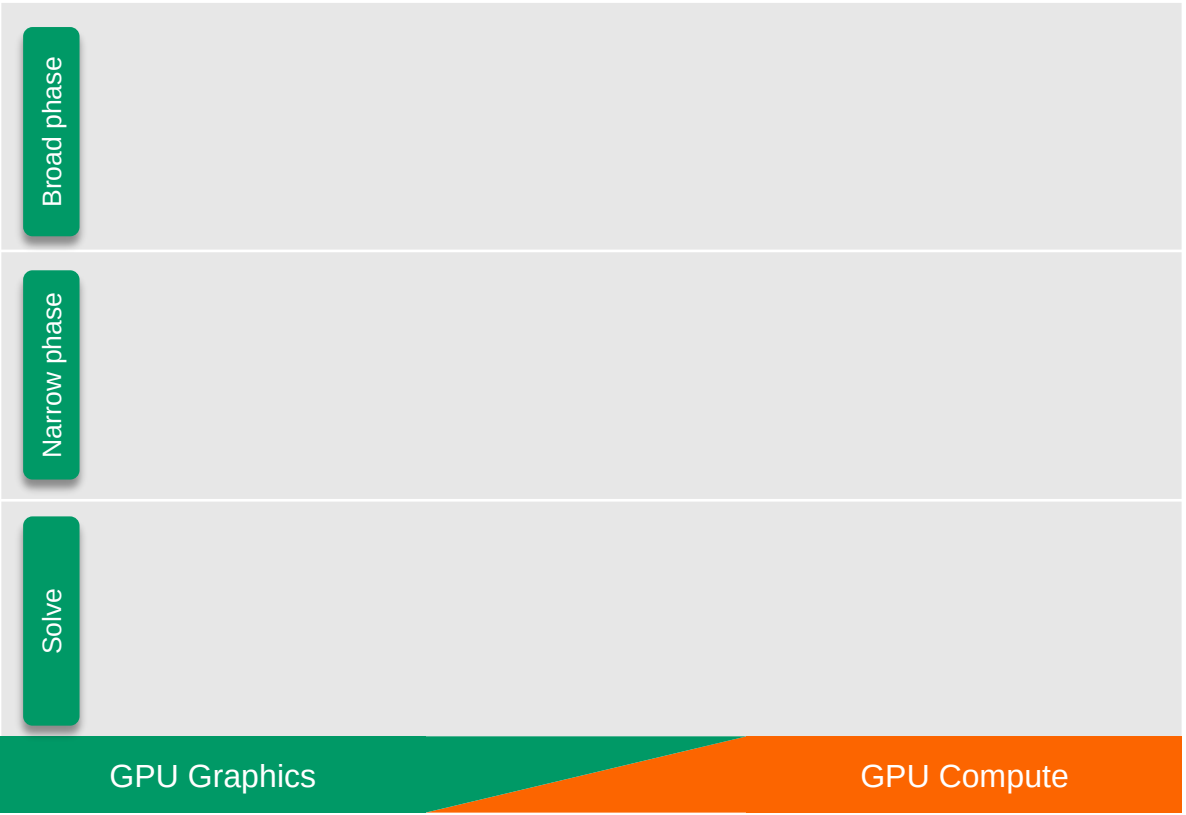
WHY USE GPU?

- Large scale simulation requires a lot of bodies
  - Destruction
- Increase the cost of a simulation

- GPU has high-
  - Peak performance
  - Memory bandwidth

- GPU is different from CPU

GPU RIGID BODY TECHNIQUES



GPU RIGID BODY TECHNIQUES

Broad phase

Narrow phase

Cube map
Sathe(2006)

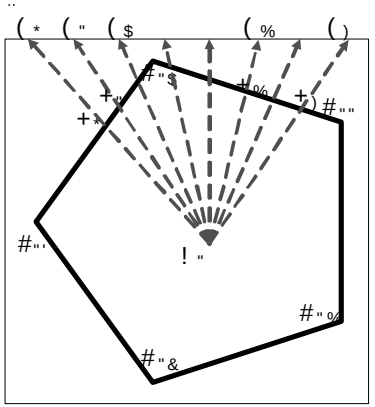
Cube map

Solve

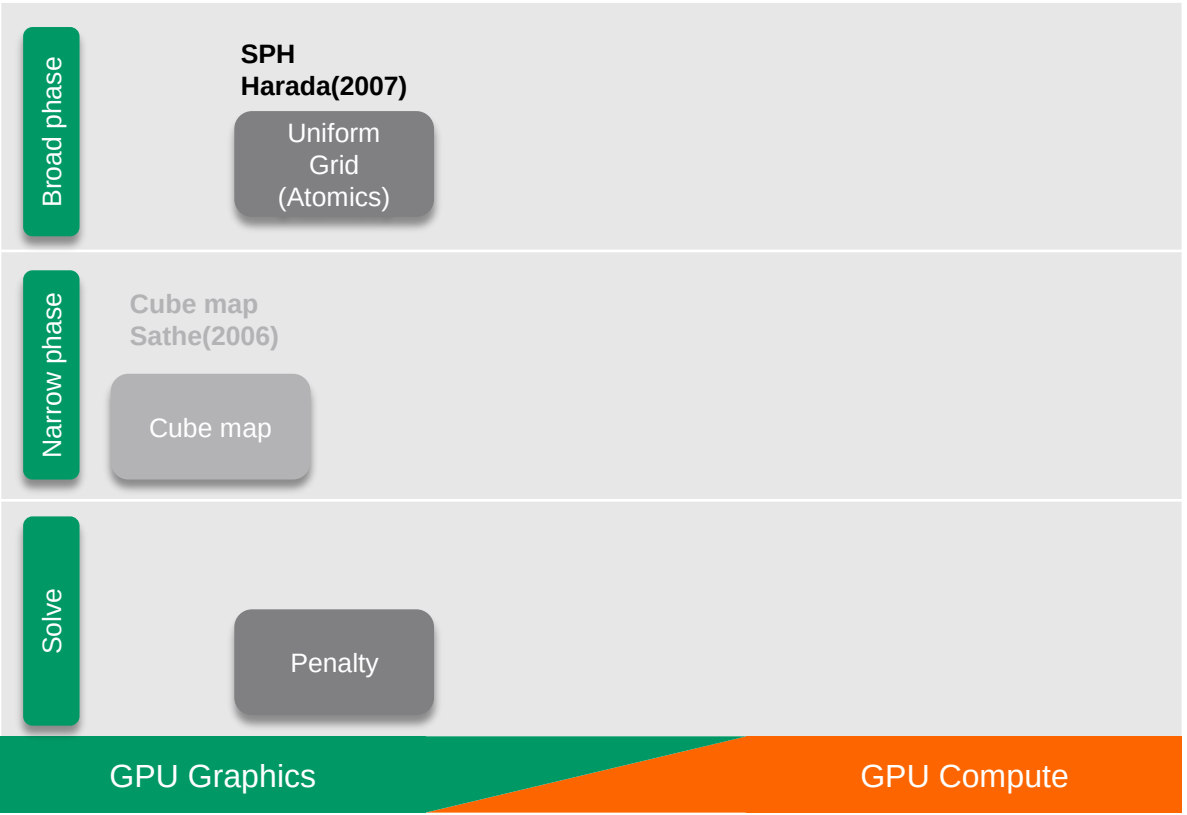
GPU Graphics

GPU Compute

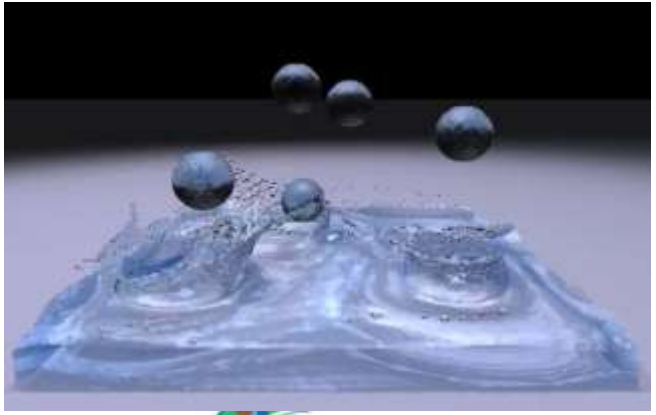
- Rahul Sathe, Rigid Body Collision Detection on the GPU, Sig Poster(2006)



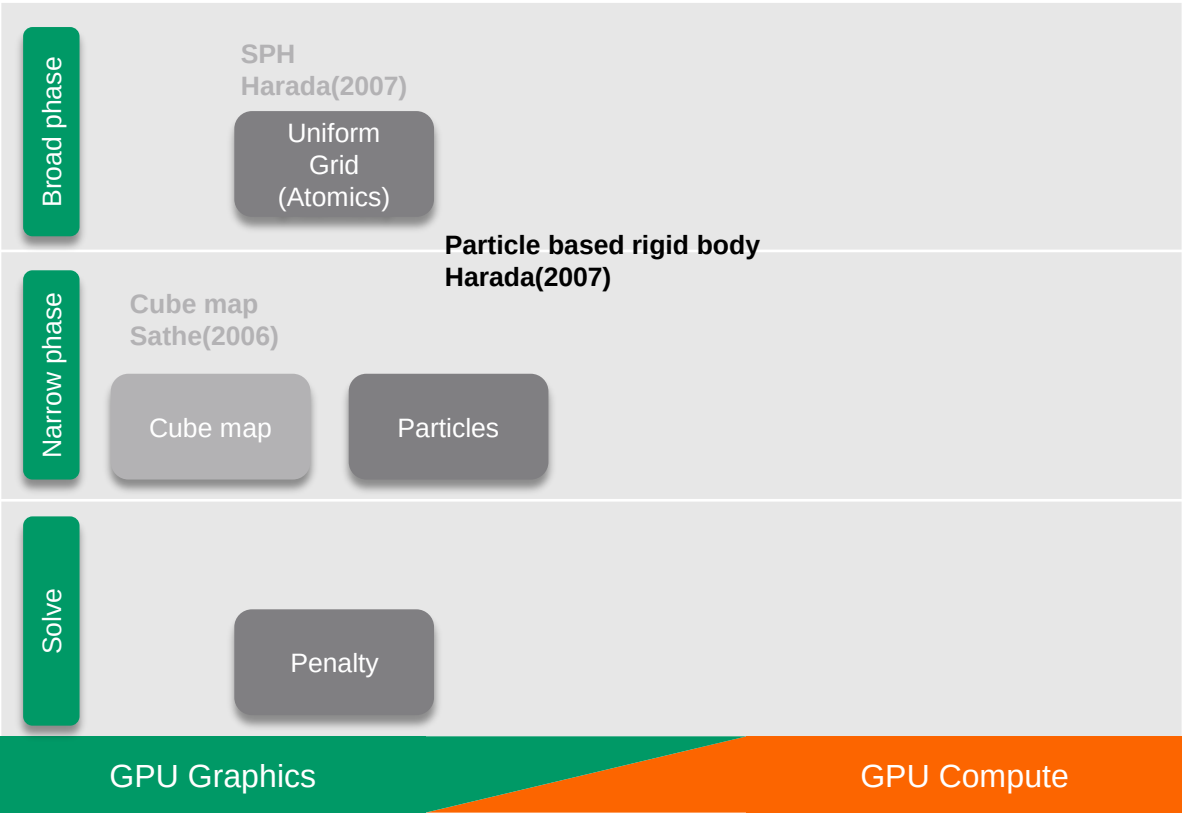
GPU RIGID BODY TECHNIQUES



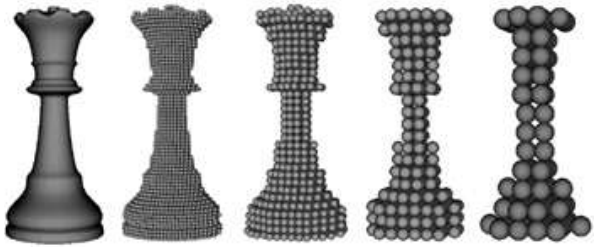
- Takahiro Harada et al. , Smoothed Particle Hydrodynamics on GPUs, Proc. of CGI(2007)

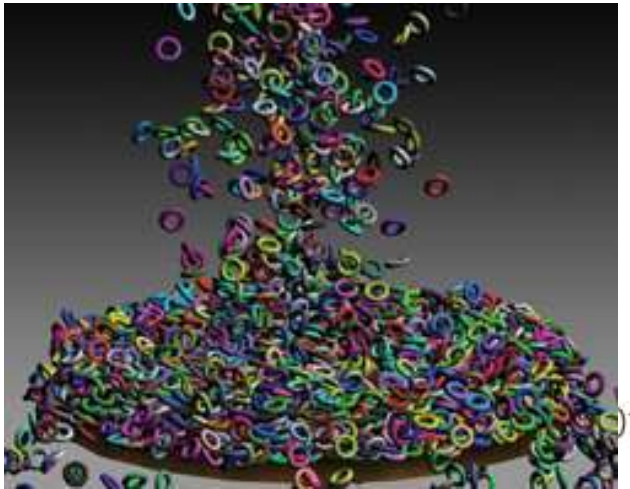


GPU RIGID BODY TECHNIQUES

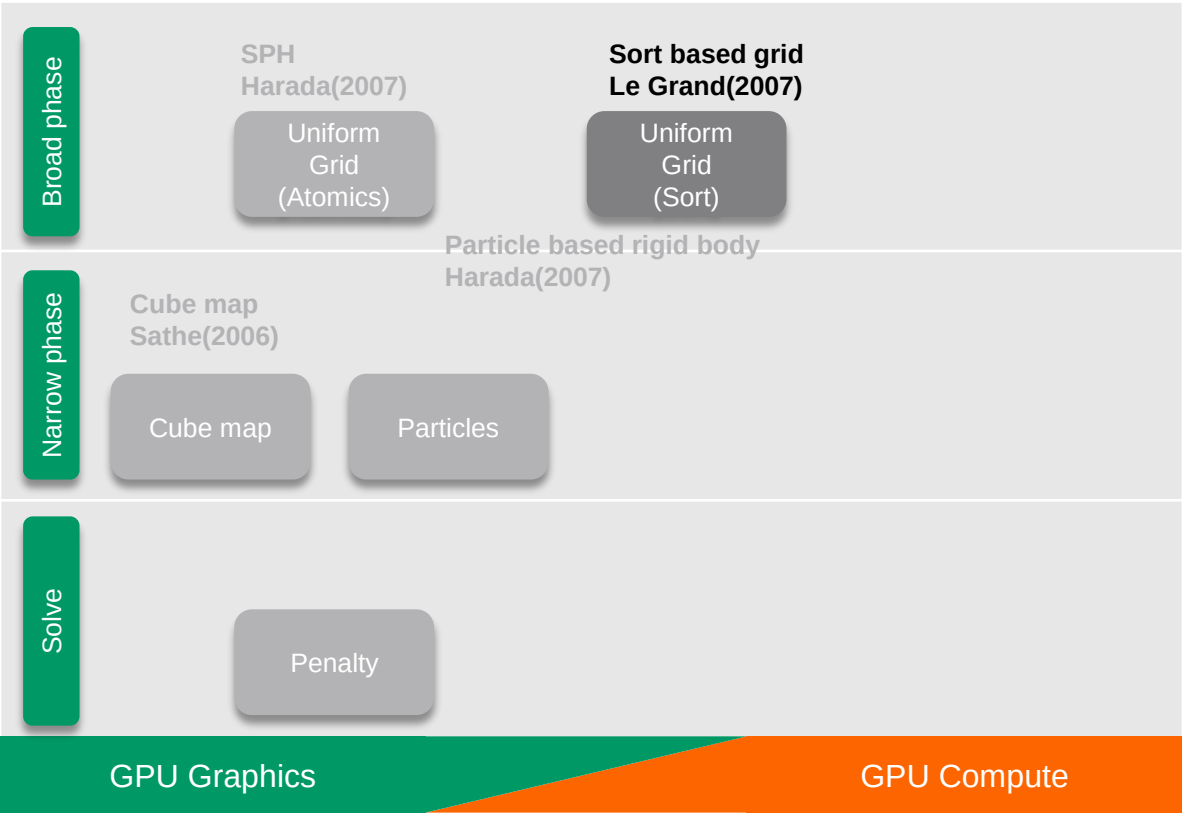


- Takahiro Harada, Real-time Rigid Body Simulation on GPUs, GPU Gems 3 (2007)

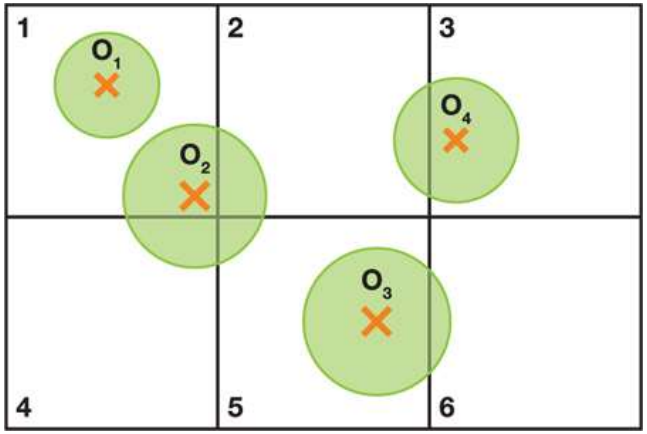




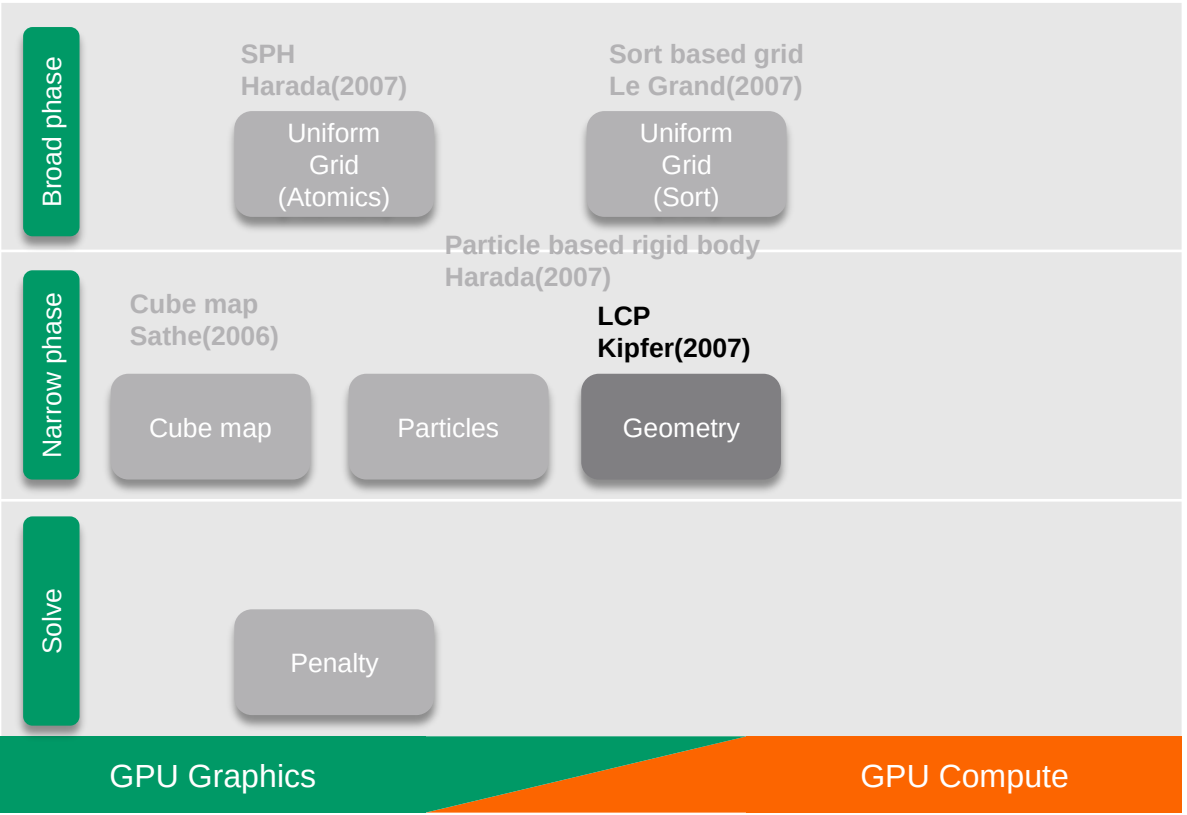
GPU RIGID BODY TECHNIQUES



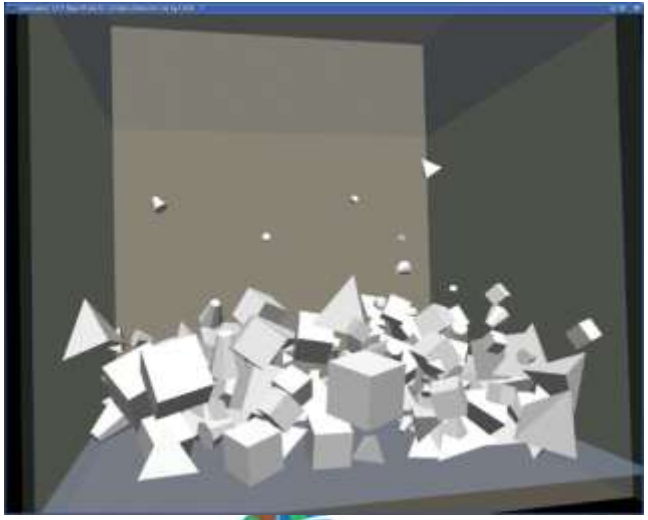
- Scott Le Grand, Broad-phase Collision Detection with CUDA, GPU Gems3(2007)



GPU RIGID BODY TECHNIQUES

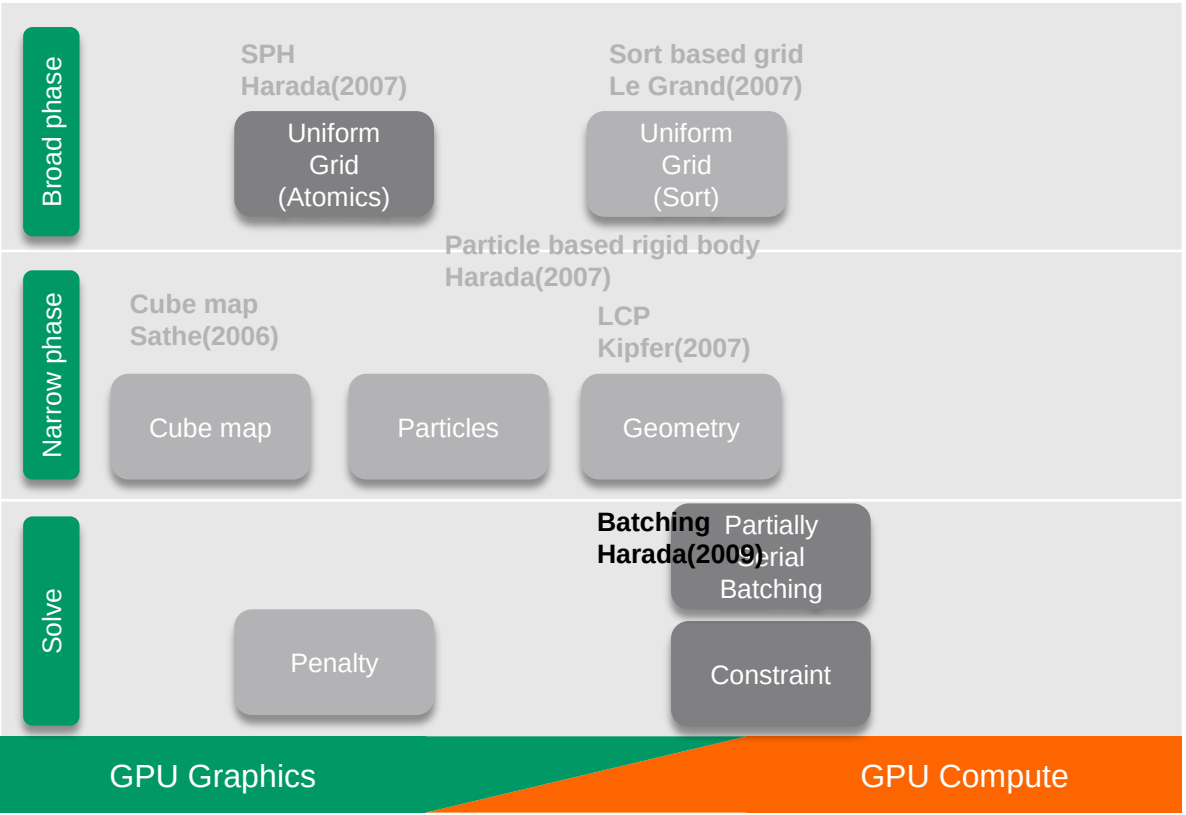


- Peter Kipfer, LCP Algorithms for Collision Detection using CUDA, GPU Gems 3(2007)

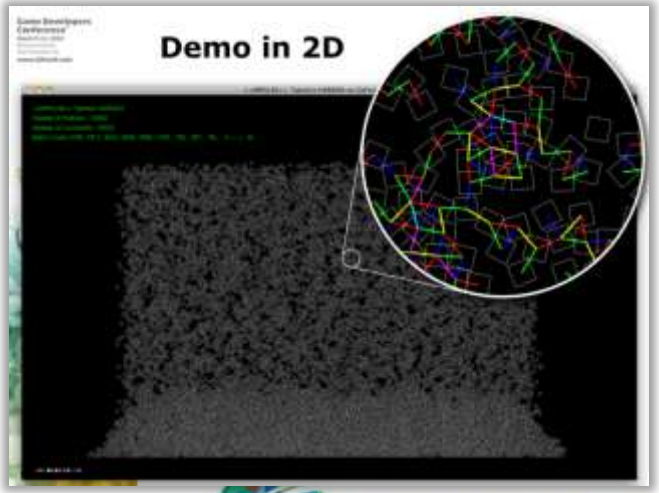


SIGGRAPH2011

GPU RIGID BODY TECHNIQUES

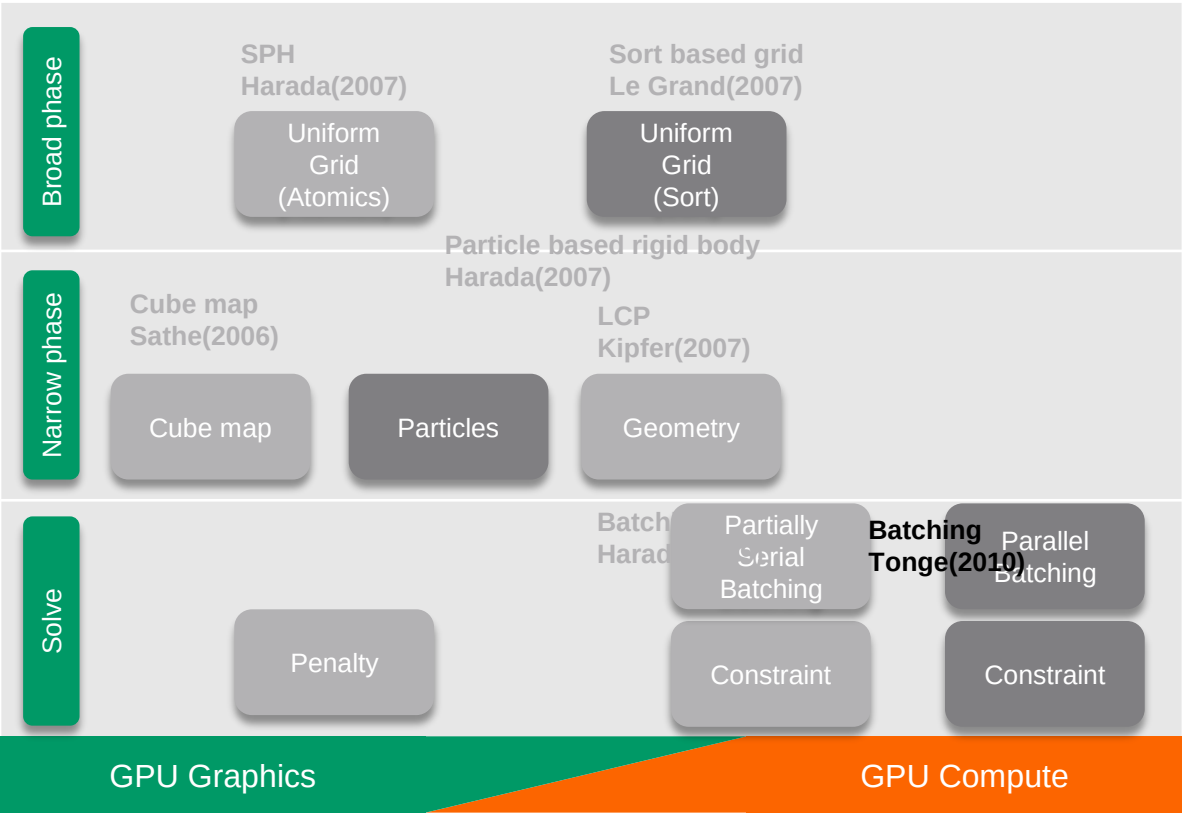


- Takahiro Harada, Parallelizing the Physics Pipeline, GDC(2009)



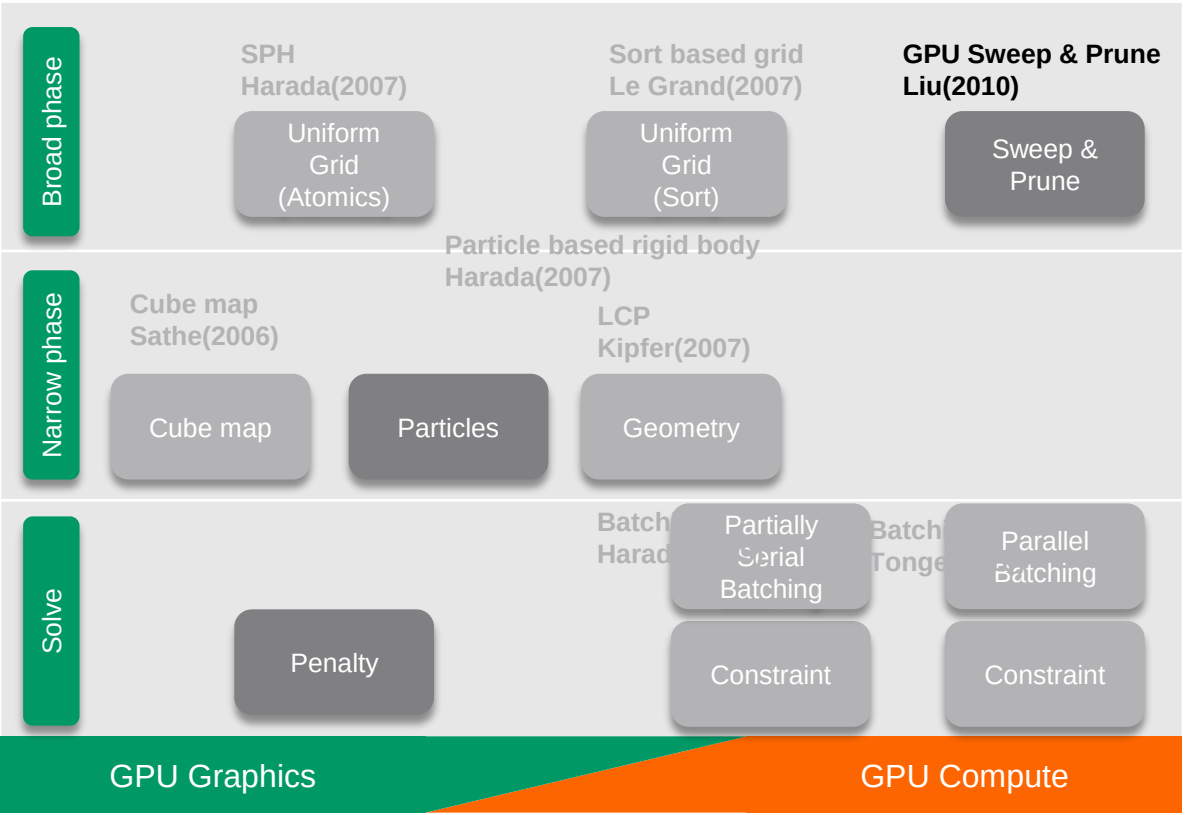
SIGGRAPH2011

GPU RIGID BODY TECHNIQUES

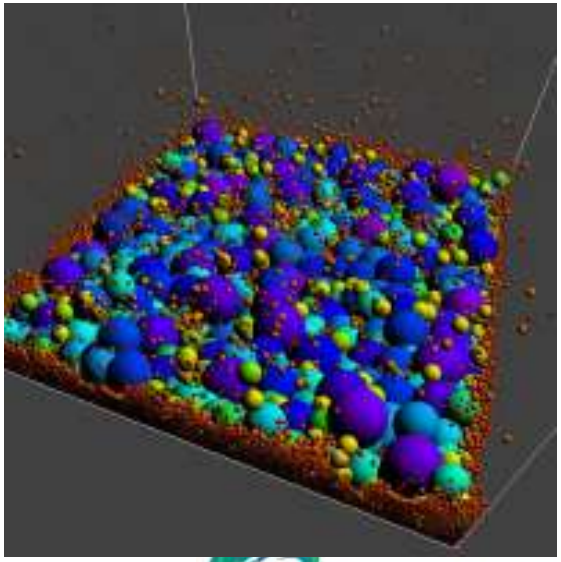


- Richard Tonge, PhysX GPU Rigid Bodies in Batman, Game Programming Gems 8(2010)

GPU RIGID BODY TECHNIQUES



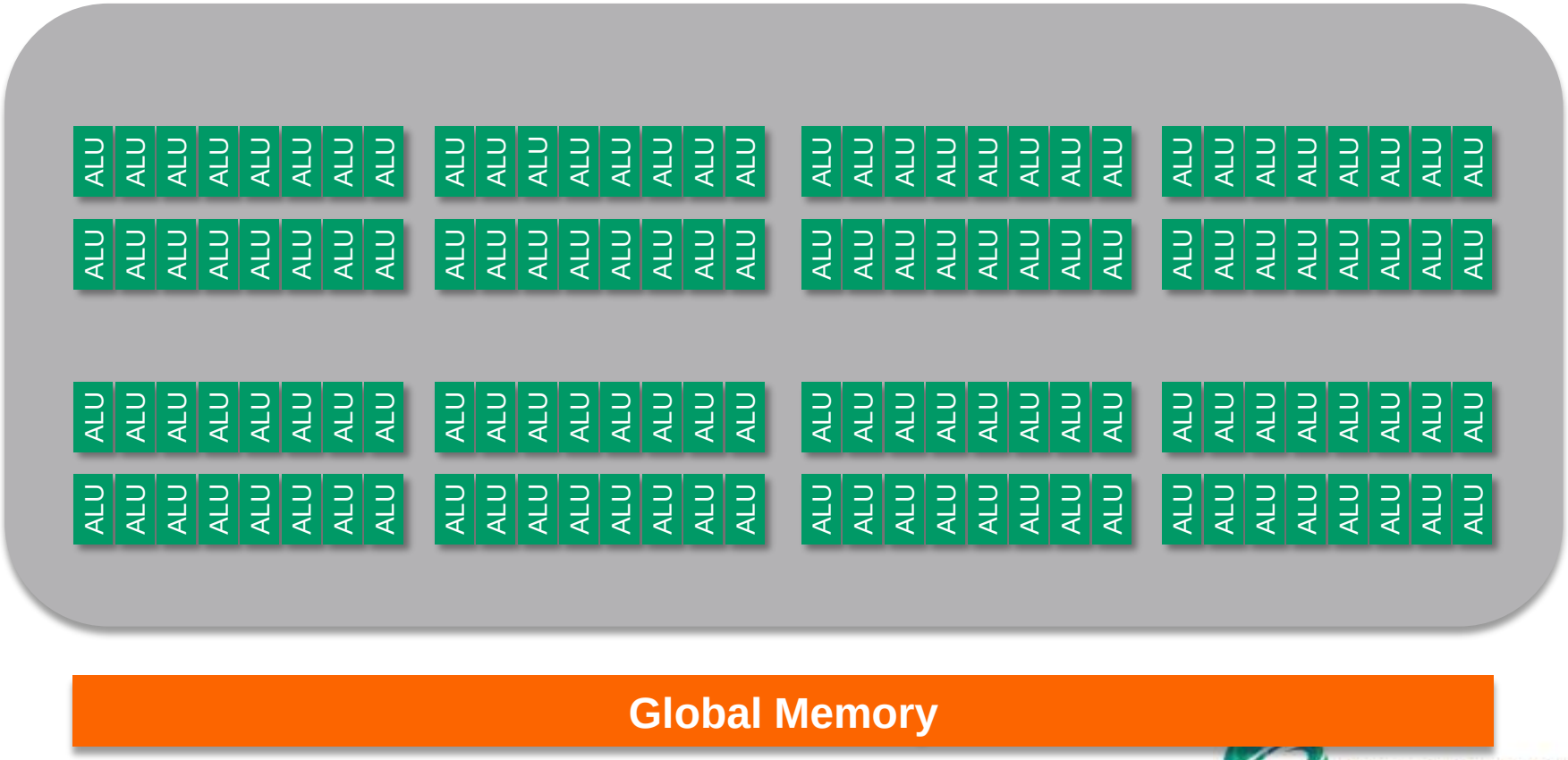
- Liu et al., Real-time Collision Culling of a Million Bodies on Graphics Processing Units, Siggraph Asia(2010)



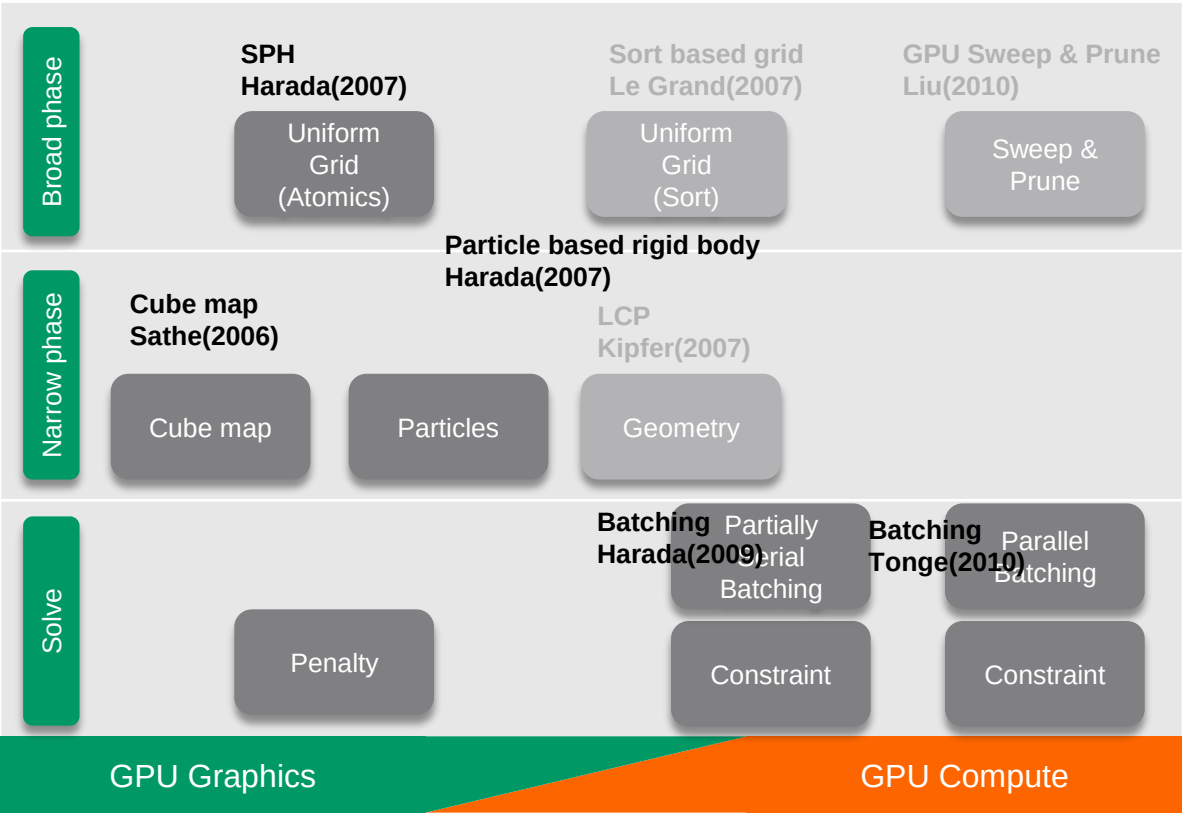
SIGGRAPH2011

Architecture & Algorithm

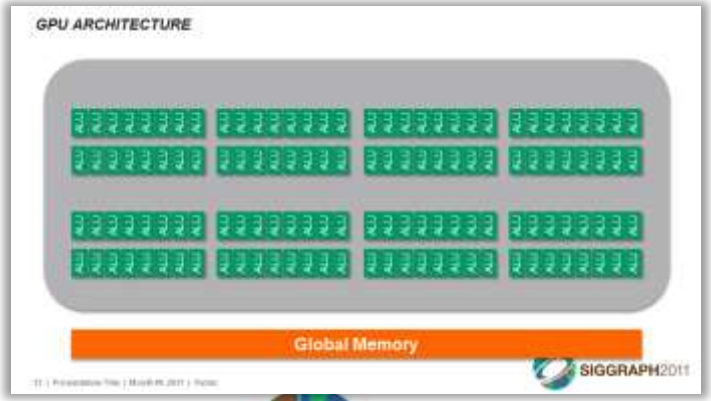
GPU ARCHITECTURE



GPU RIGID BODY TECHNIQUES

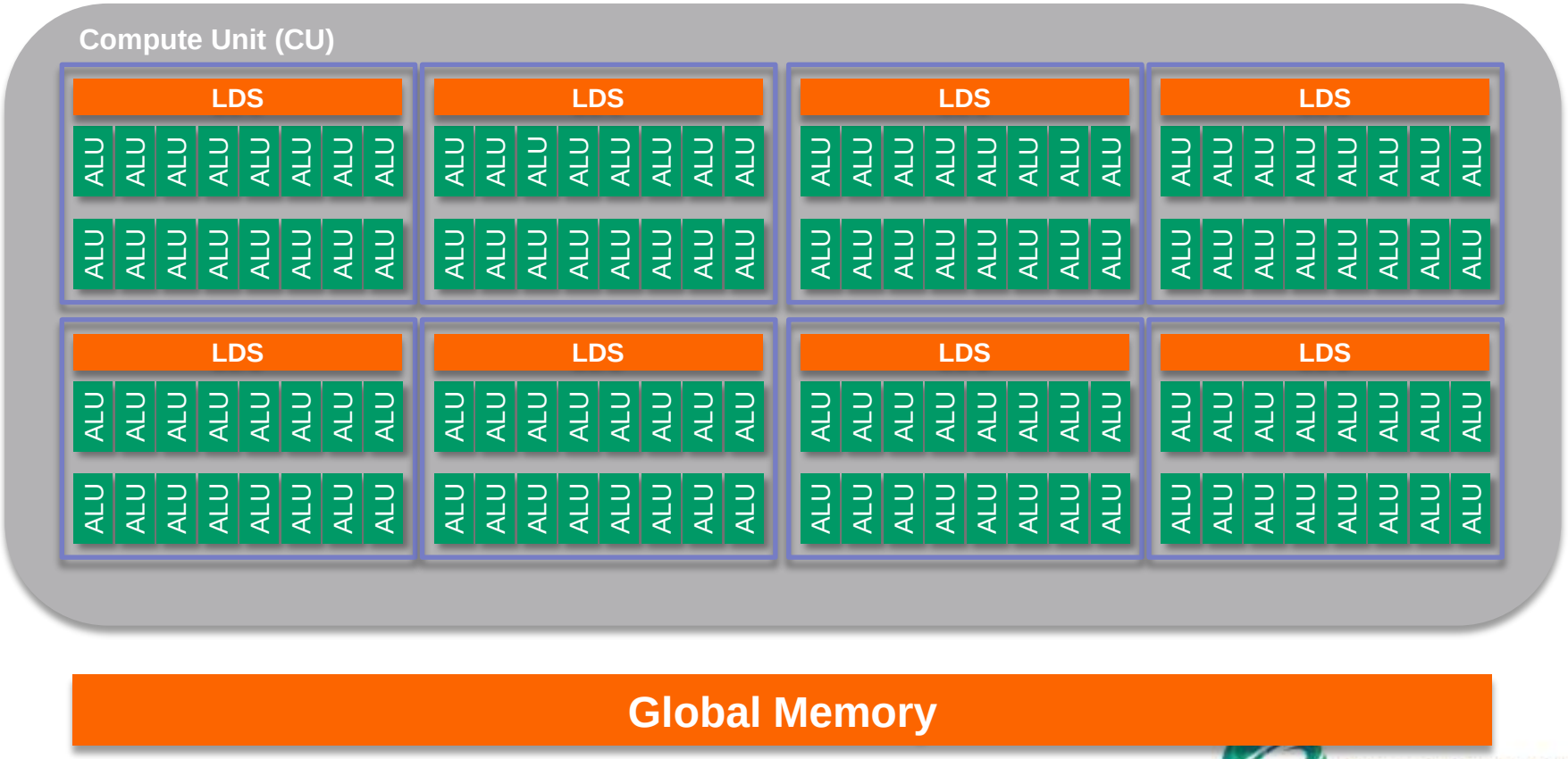


- Used GPU as a processor with many ALUs

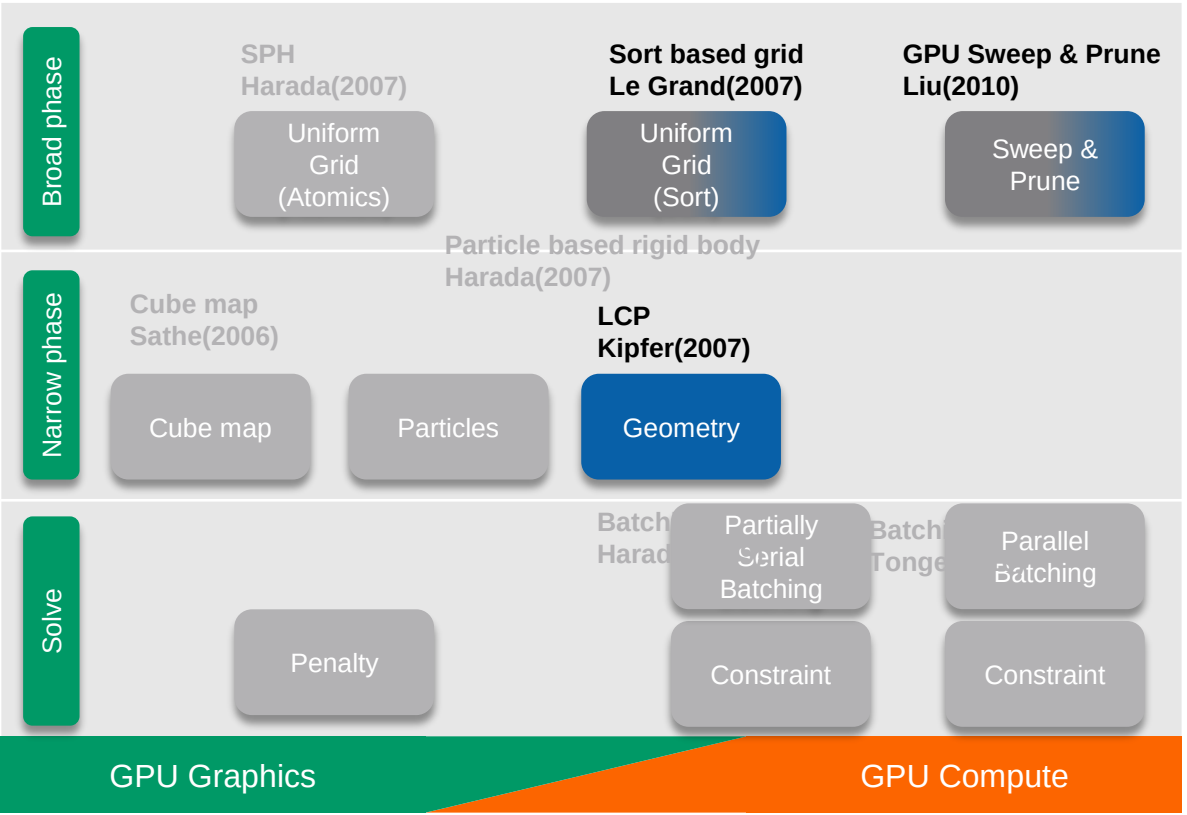


SIGGRAPH2011

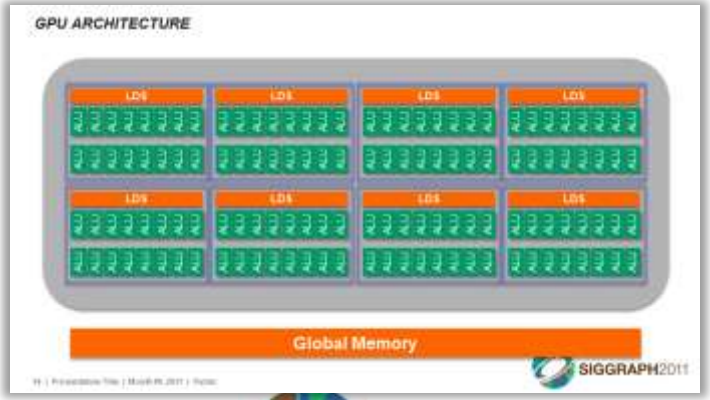
GPU ARCHITECTURE



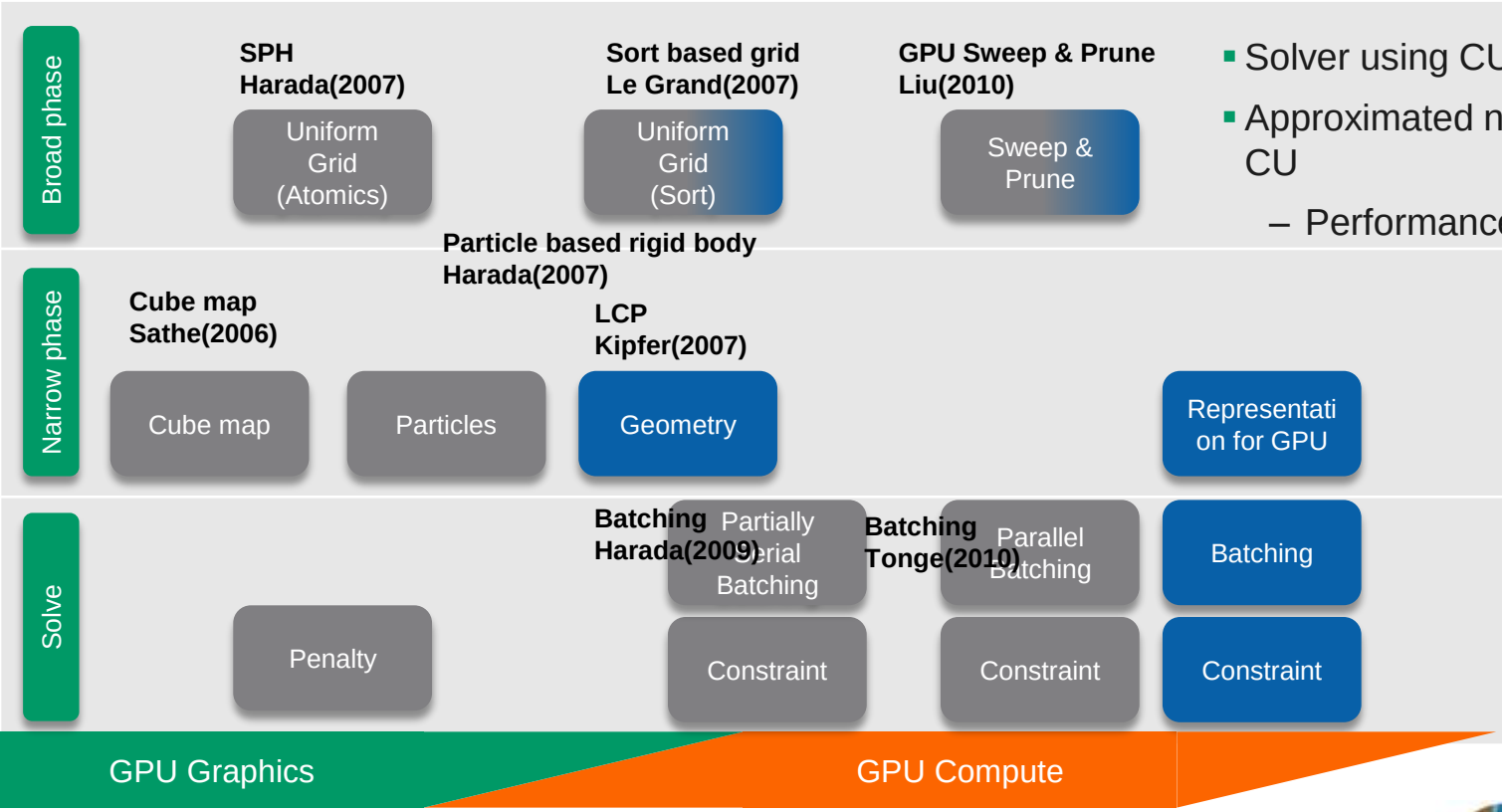
GPU RIGID BODY TECHNIQUES



- LCP
  - A CU processes a pair
  - Synchronization
  - LDS
- Broad phase collision
  - Radix sort



GPU RIGID BODY TECHNIQUES

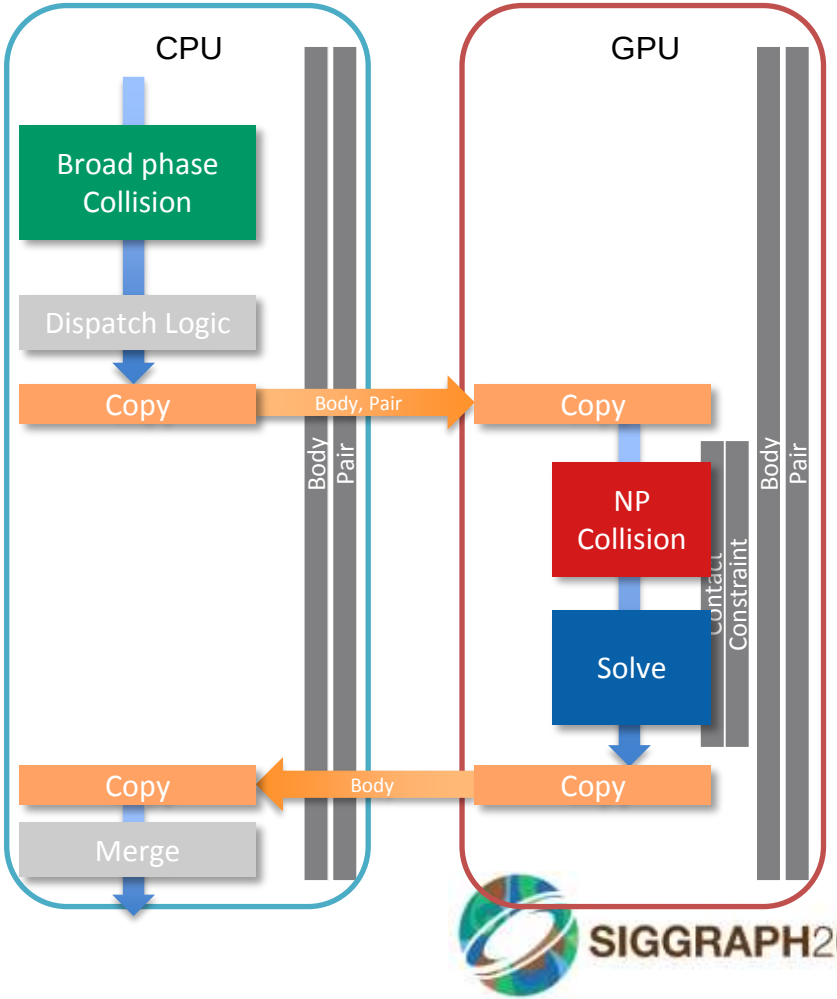


- Solver using CU
- Approximated narrowphase using CU
  - Performance

Compute Unit Aware Rigid Body Simulation

PIPELINE

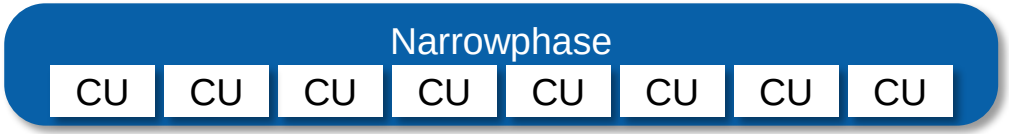
- Copy body and pair buffer
- GPU allocates big buffers
  - Contact
  - Constraints
- Narrow phase and solve is done on the GPU
- Don't have to read back big buffers



NARROWPHASE – CU LEVEL PARALLELIZATION

- Collision of a pair is independent
- Use a SIMD lane for a pair
  - Not enough resource for a SIMD lane
- Use a CU for a pair

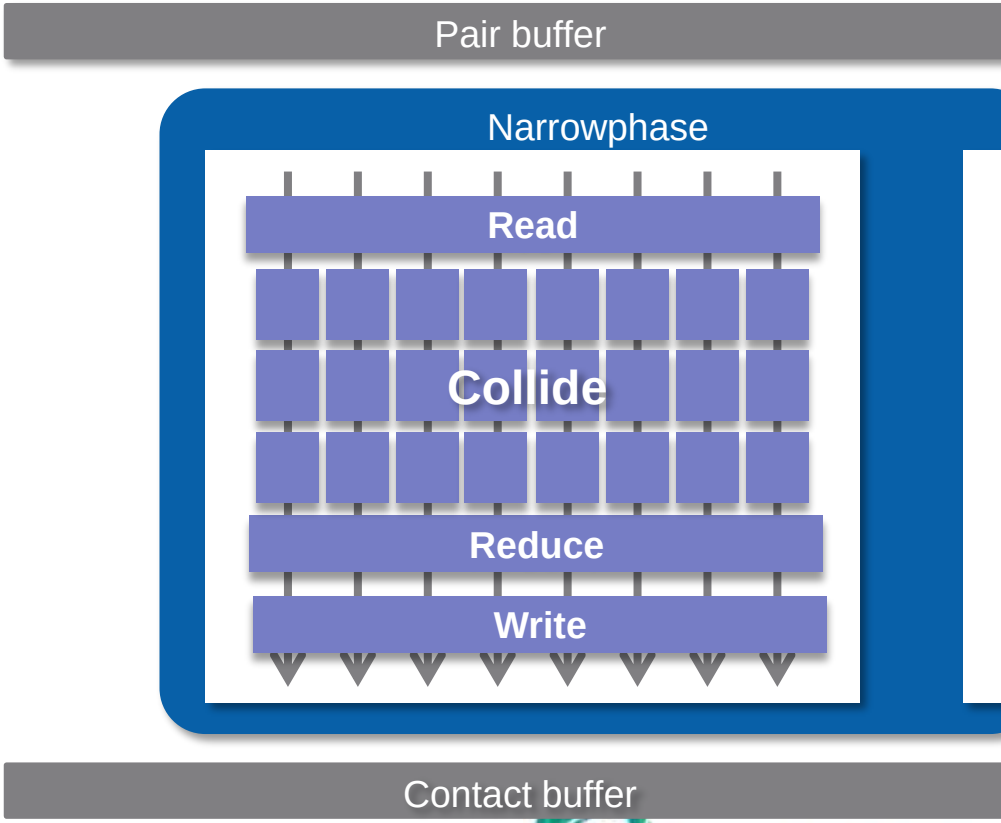
- A CU processes and “append” colliding pairs
  - HW accelerated append operation on AMD



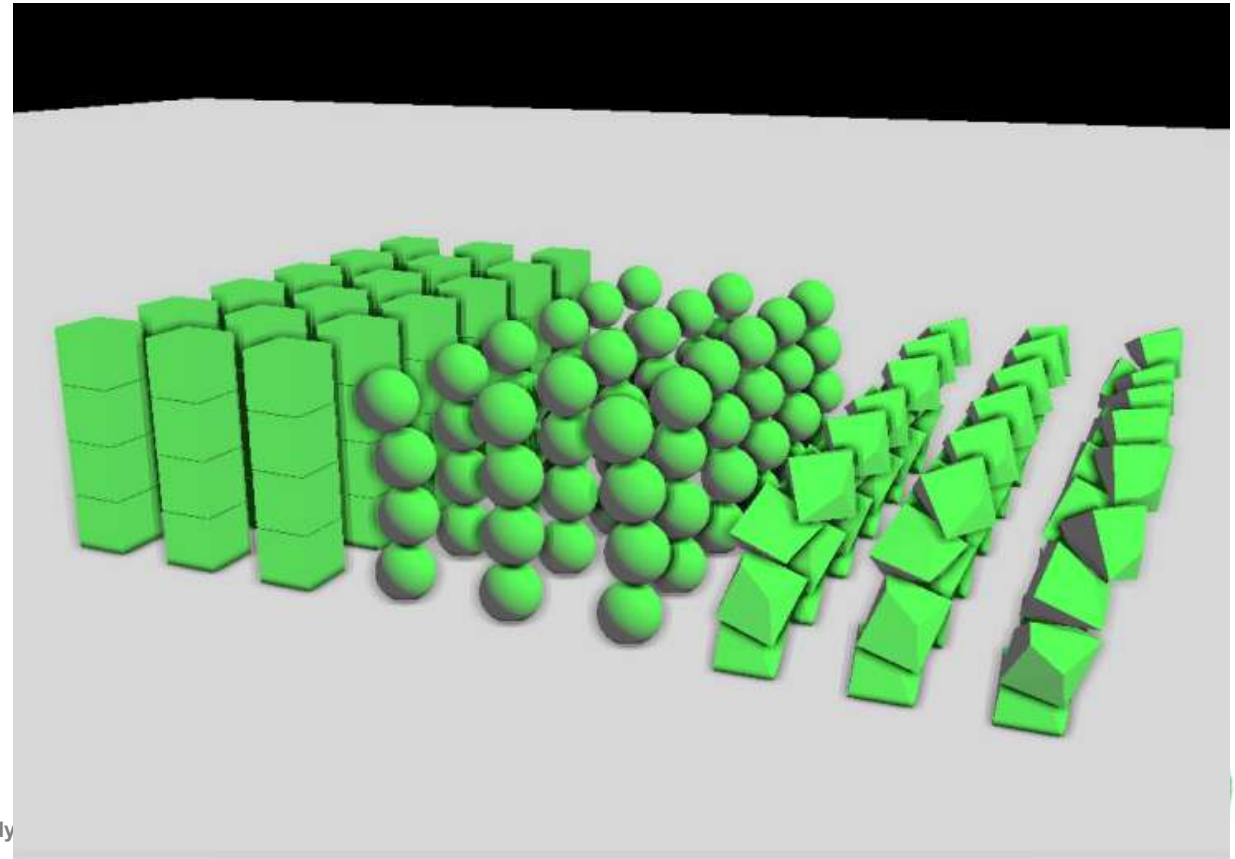
- Load balancing
  - CU can fetch a pair from queue
  - Explicit pair split
- CU level parallelization

NARROWPHASE – SIMD LANE PARALLELIZATION

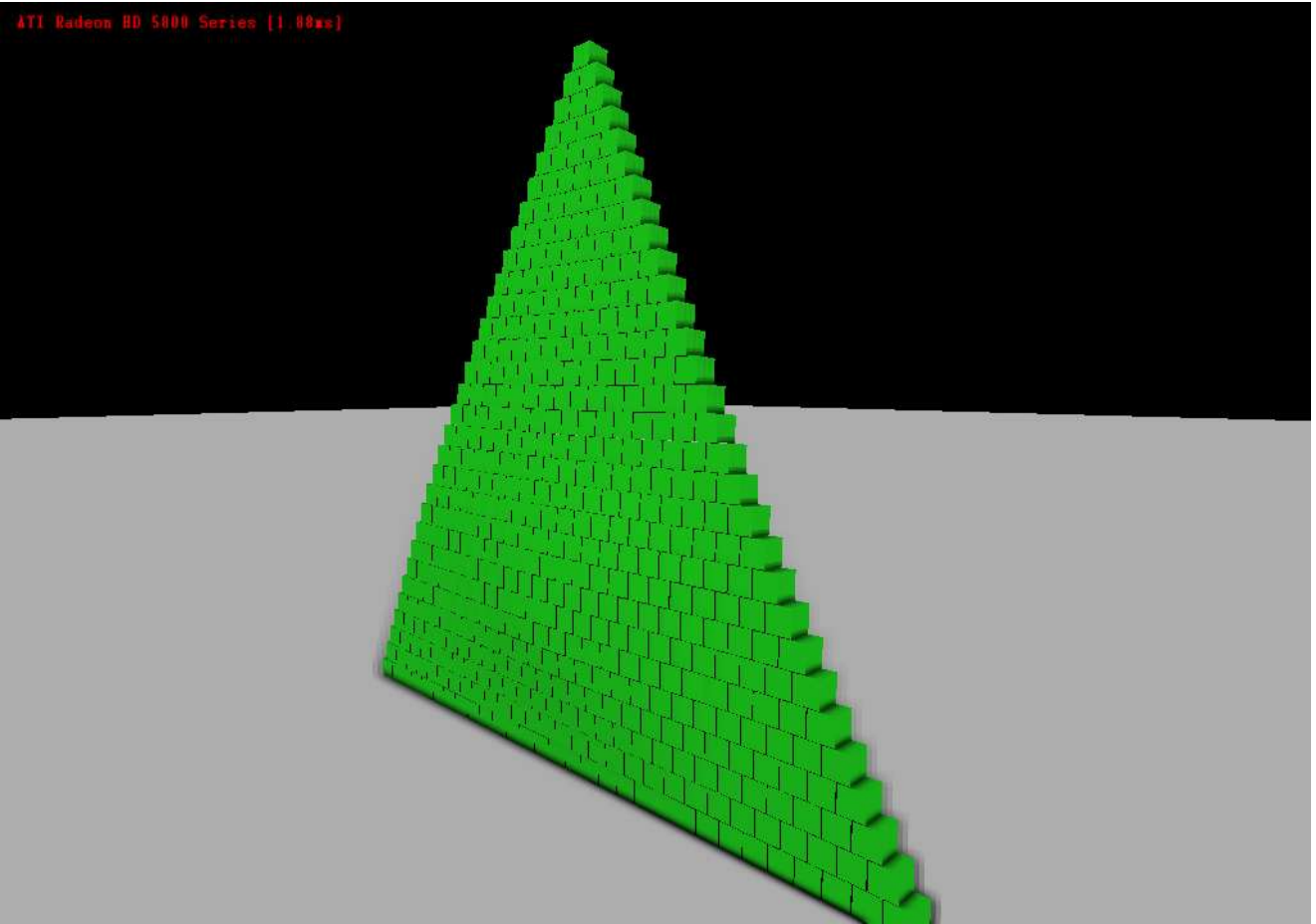
- A collision has to be split into parallel works
- Parallel collision detection
- Support arbitrary convex shapes
- Too many contact points
  - Increase cost of solver
- Redundancy
  - Use sync and LDS for vector wide operation
  - Eliminate redundant contacts

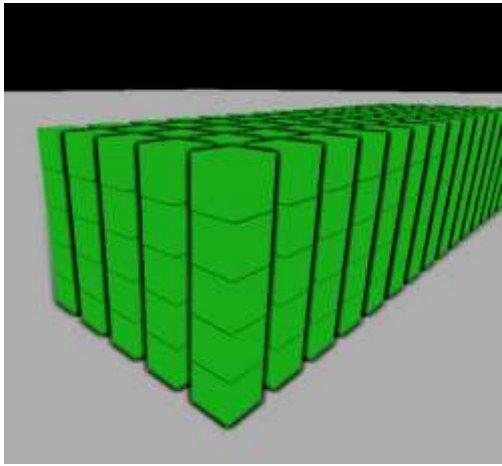


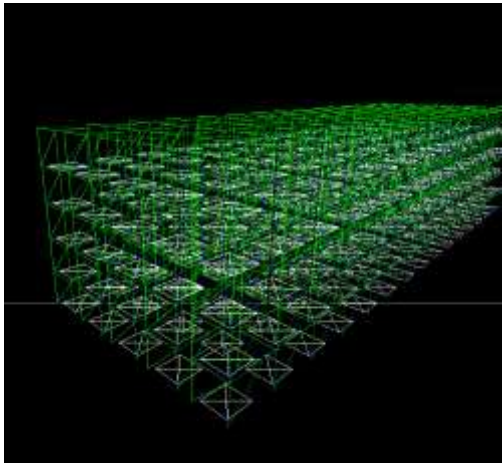
SHAPE REPRESENTATION



SHAPE REPRESENTATION

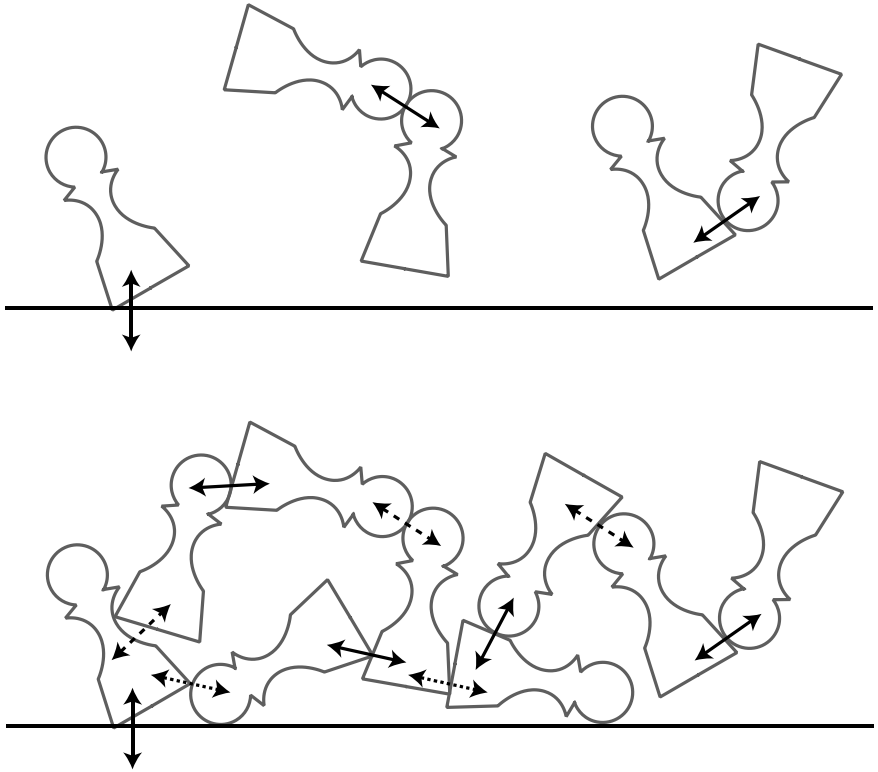






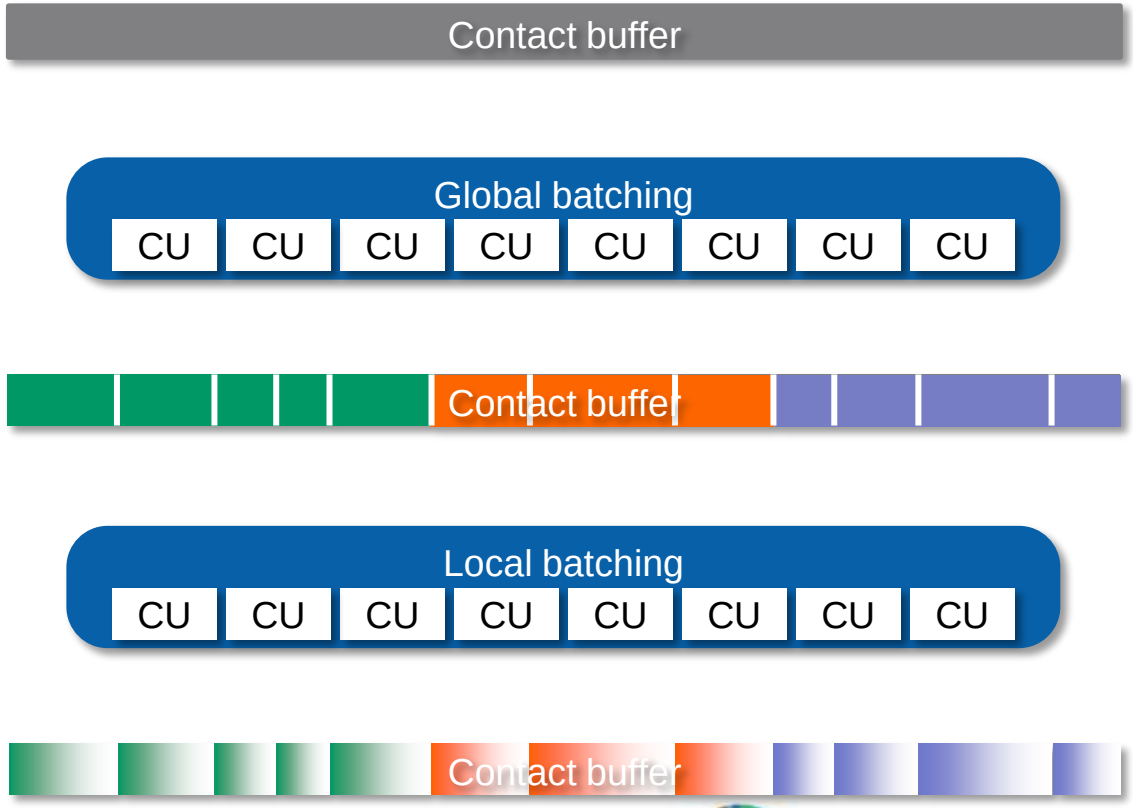
SOLVER

- Constraint based solver uses a Gauss Seidel method
  - Velocity is input and output
  - <-> Penalty method
- Sequential solve
- Parallel solve is possible only if
  - Bodies are not shared among constraints
- When a body is colliding to several bodies, parallel solve cannot be used
- Solution
  - Split constraints into batches
  - Constraints in a batch doesn't share bodies



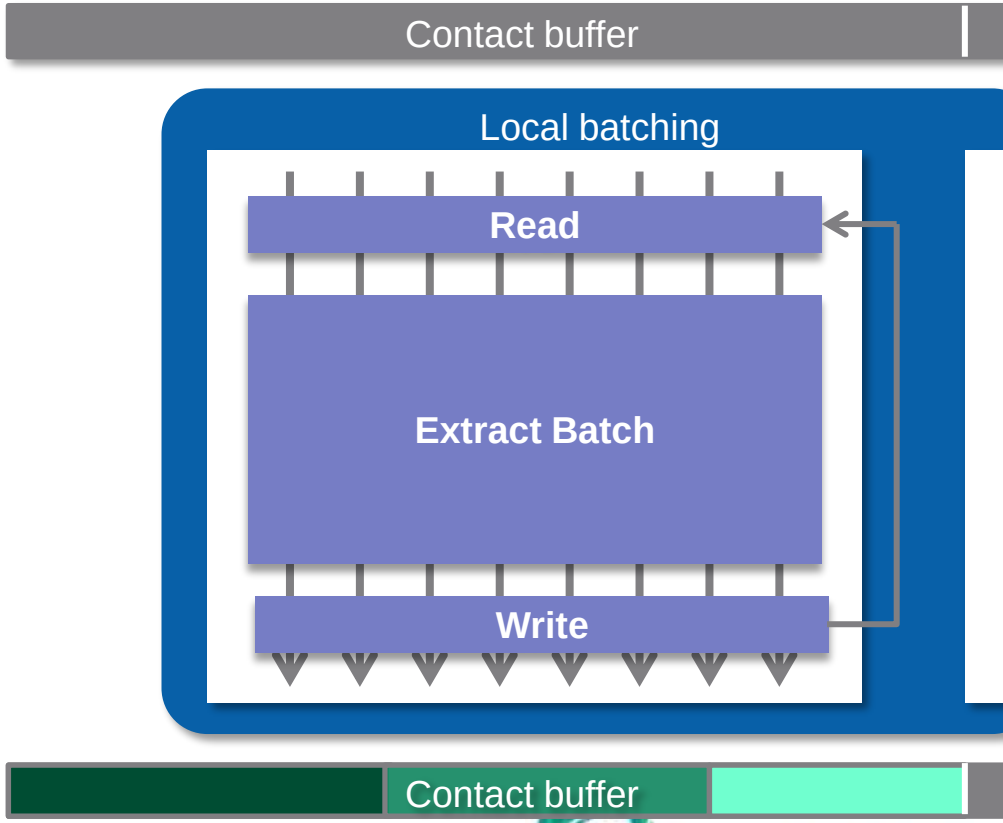
BATCHING

- 2 level batching
  - Global batching
    - Split into independent sections
  - Local batching
    - Batch in each section



ITERATIVE LOCAL BATCHING

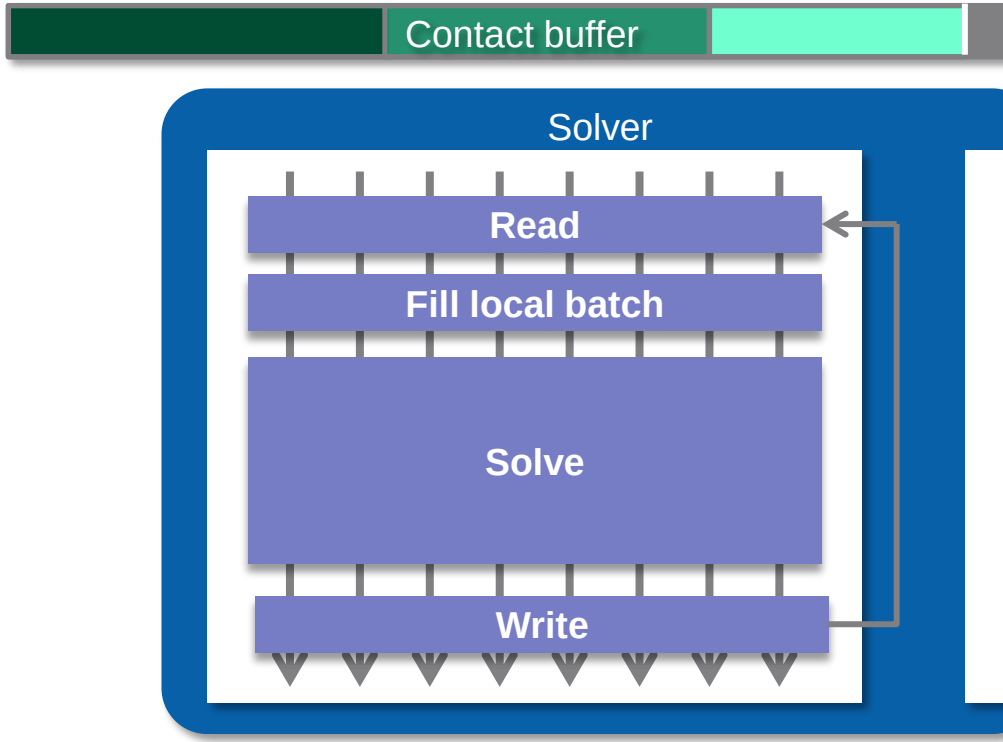
- A CU processes a section of contact buffer
  - Local batching <-> Global batching
  - Extract independent pairs in a set
- Stream processing pair buffer
  - Read to local, reorder
- Pairs are localized
  - X Parallel batching
  - X Serial batching
  - O Iterative batching



SOLVER

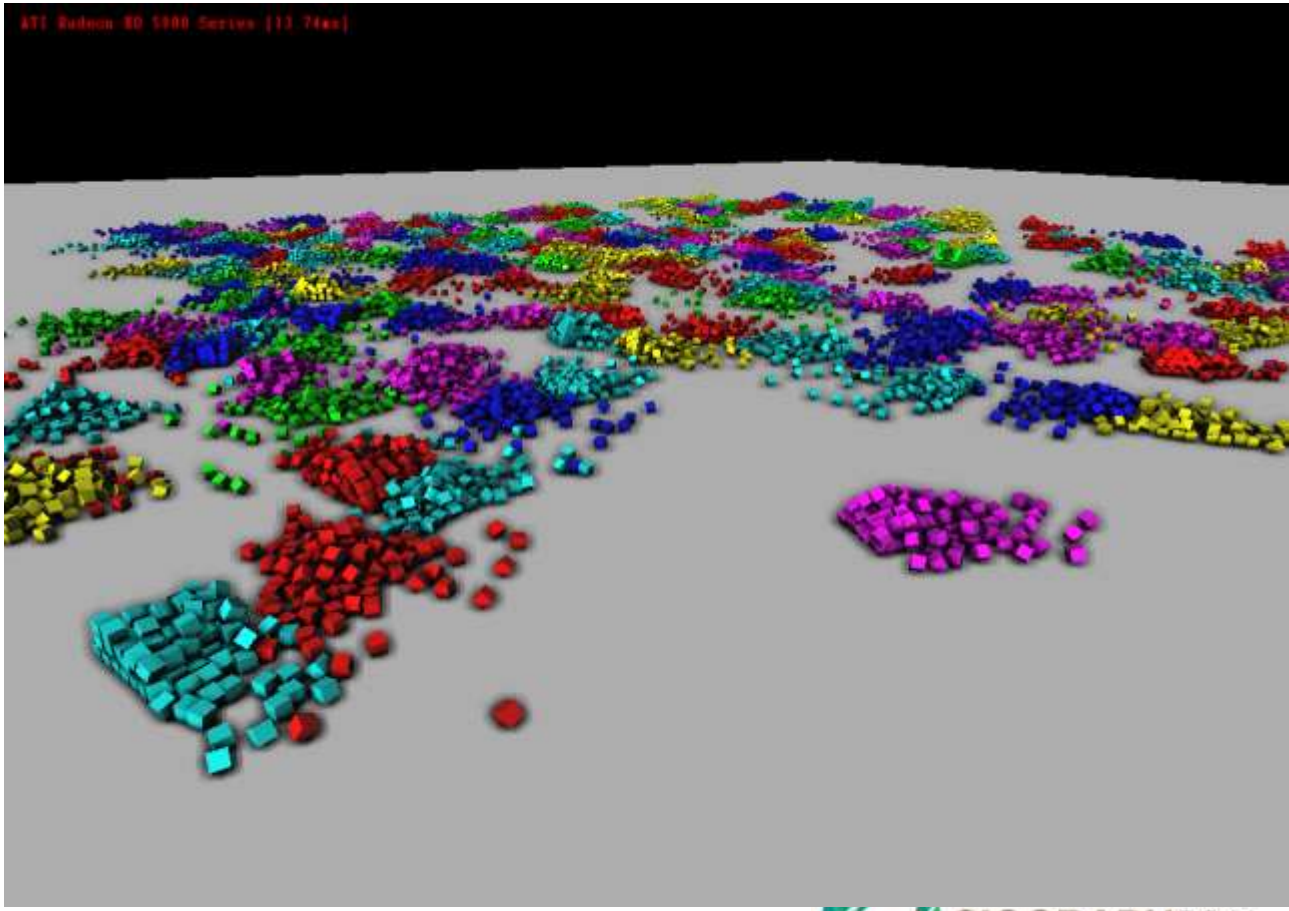
- A CU processes a section of contact buffer
- GPU dispatches works by itself
  - Read constraints
  - Fill batch buffer
  - Solve
  - If the batch is done, go to the next batch

- Previous work
  - CPU had to dispatch a kernel per batch
- Our method
  - CPU dispatches some
  - GPU dispatches works by itself



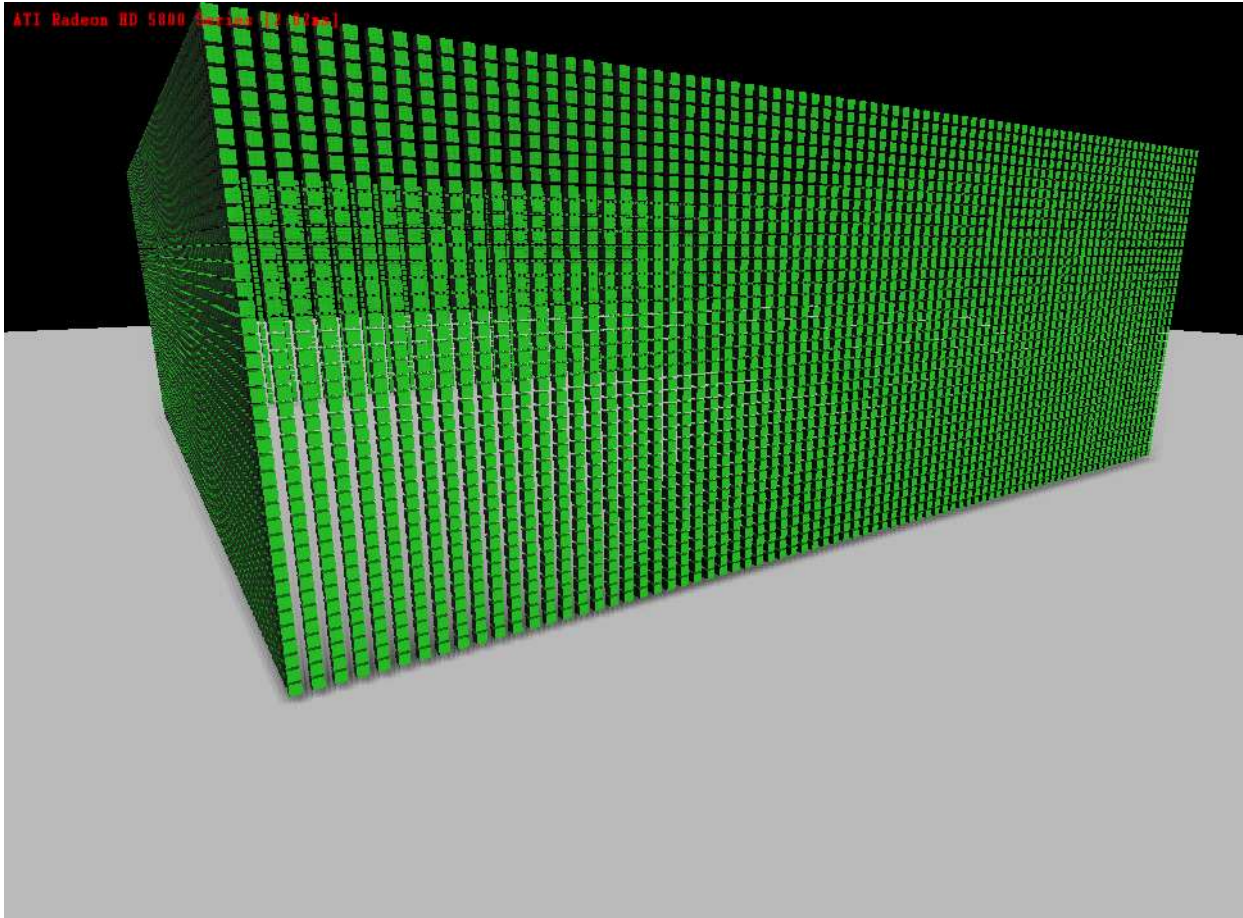
DEMO

- OpenCL
- Radeon HD5870
- 20K objects
- About 30fps
- GPU collide and solve ≈ 30ms (Inc. data transfer)



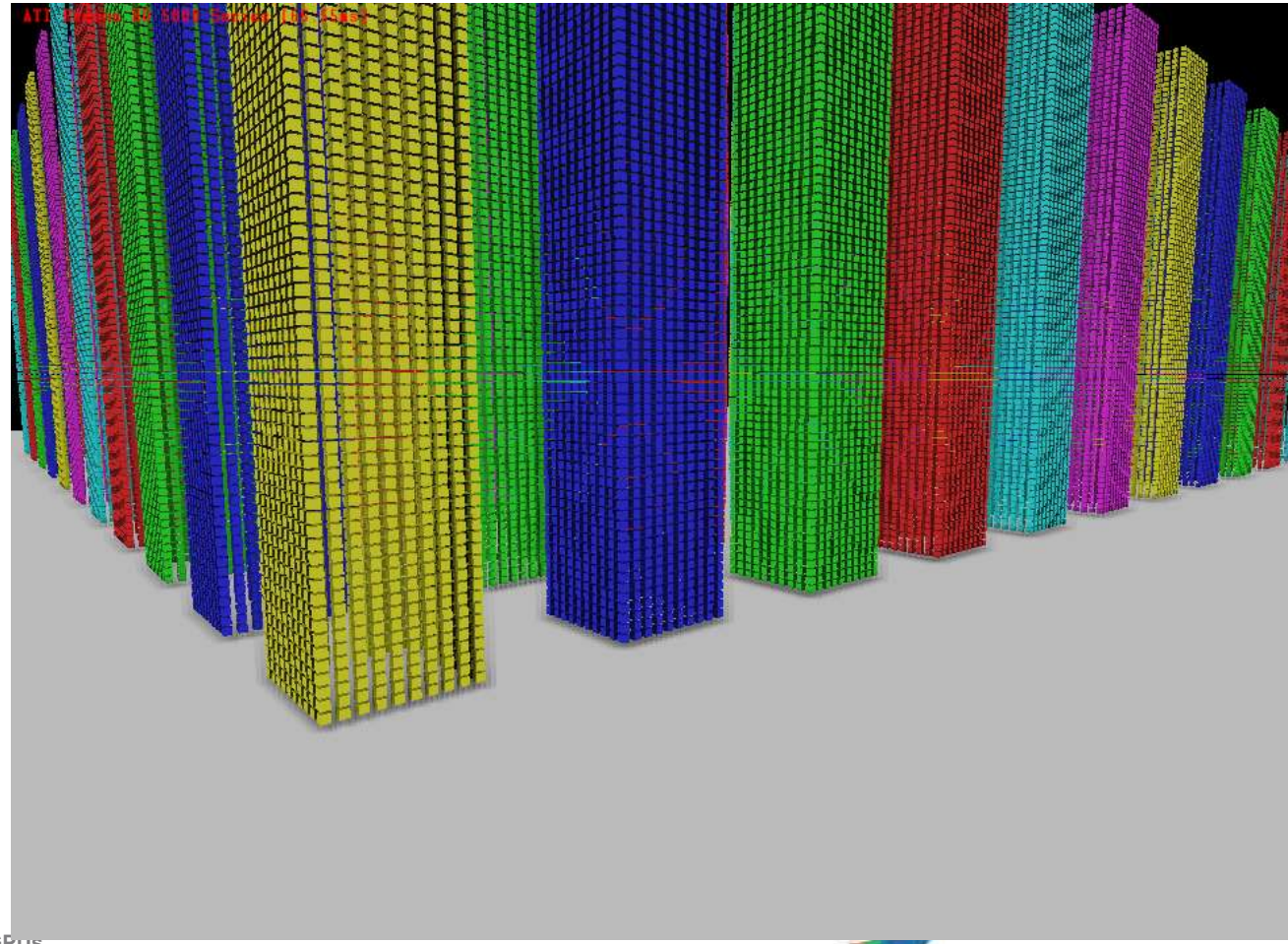
DEMO

- OpenCL
- Radeon HD5870
- 12K objects
- About 30fps
- GPU collide and solve ≈ 30ms
  (Inc. data transfer)



OFFLINE DEMO

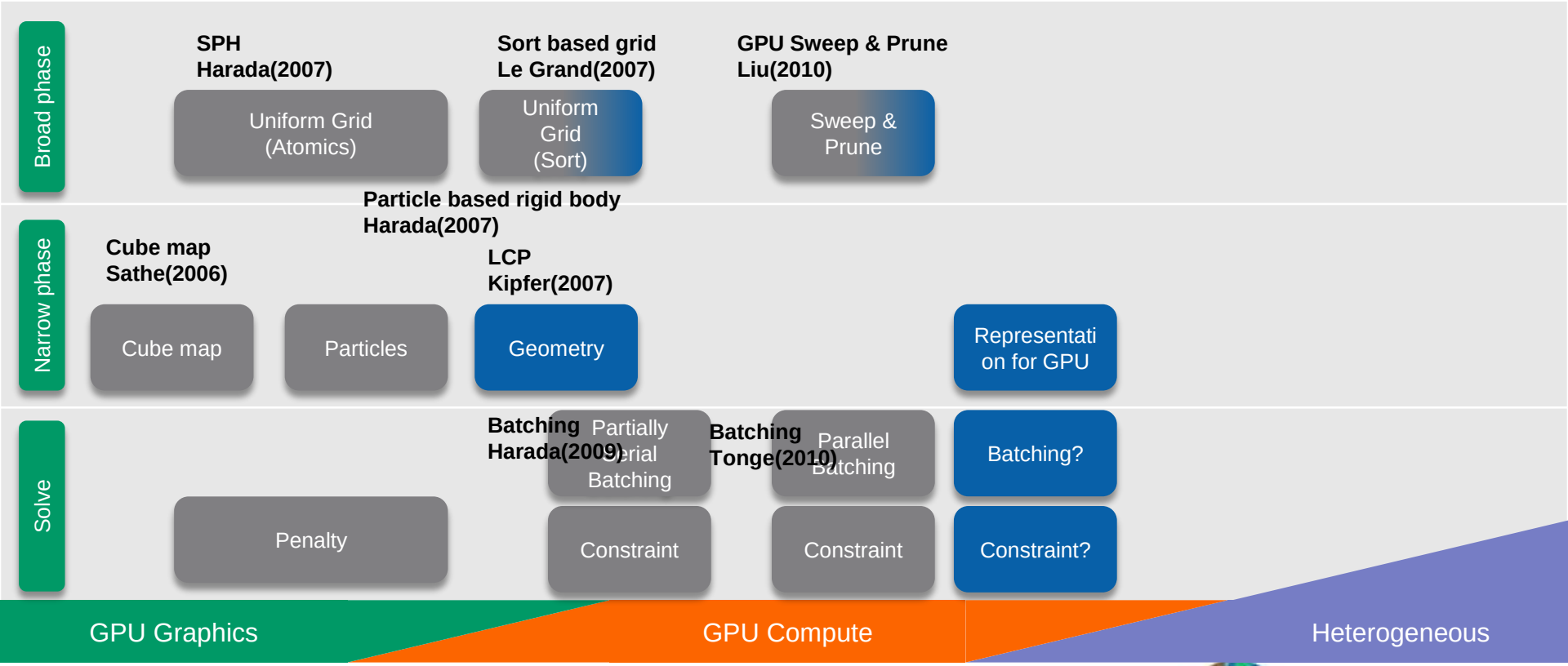
- 400K objects
- GPU collide and solve < 1s



LIMITATIONS

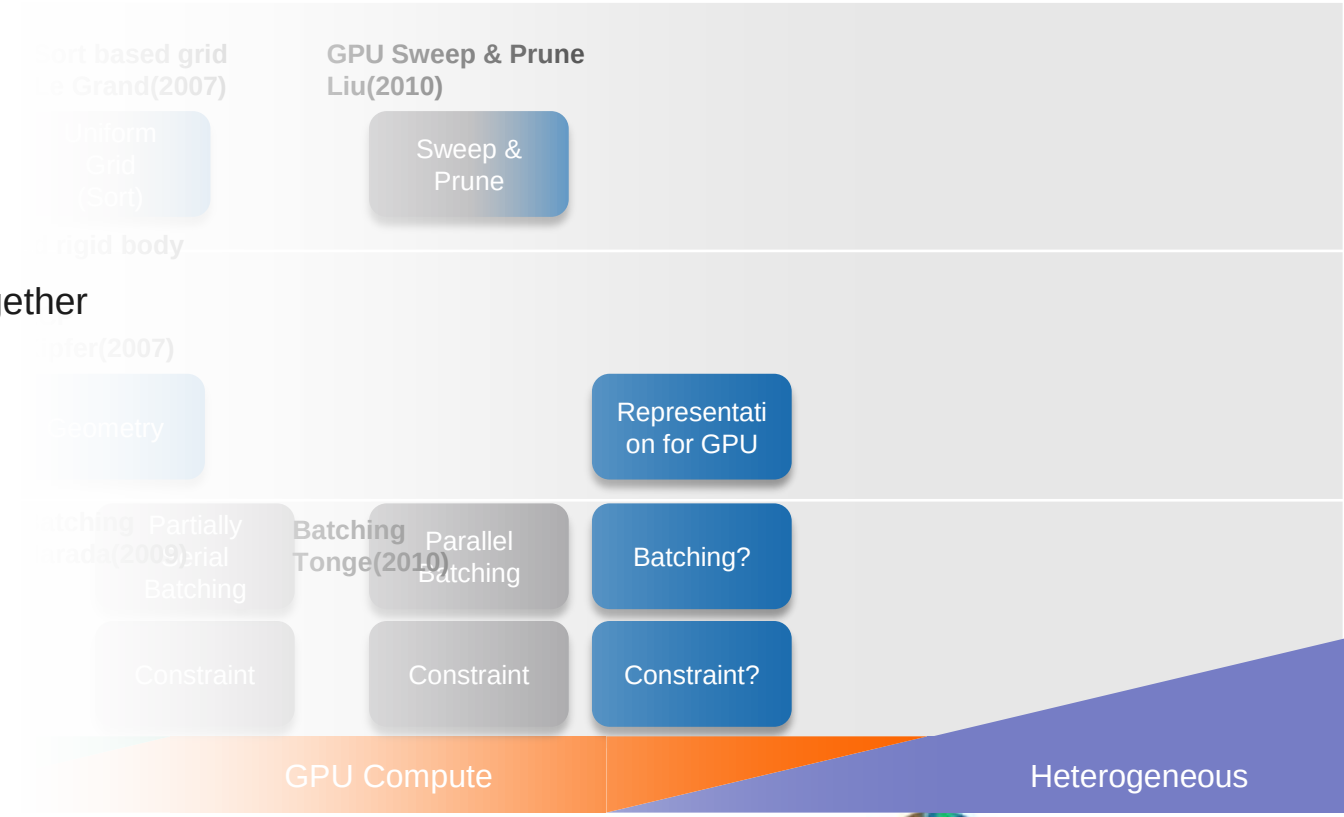
- GPU prefers uniform work granularity
- Some works are better to use CPUs

HETEROGENEOUS ERA

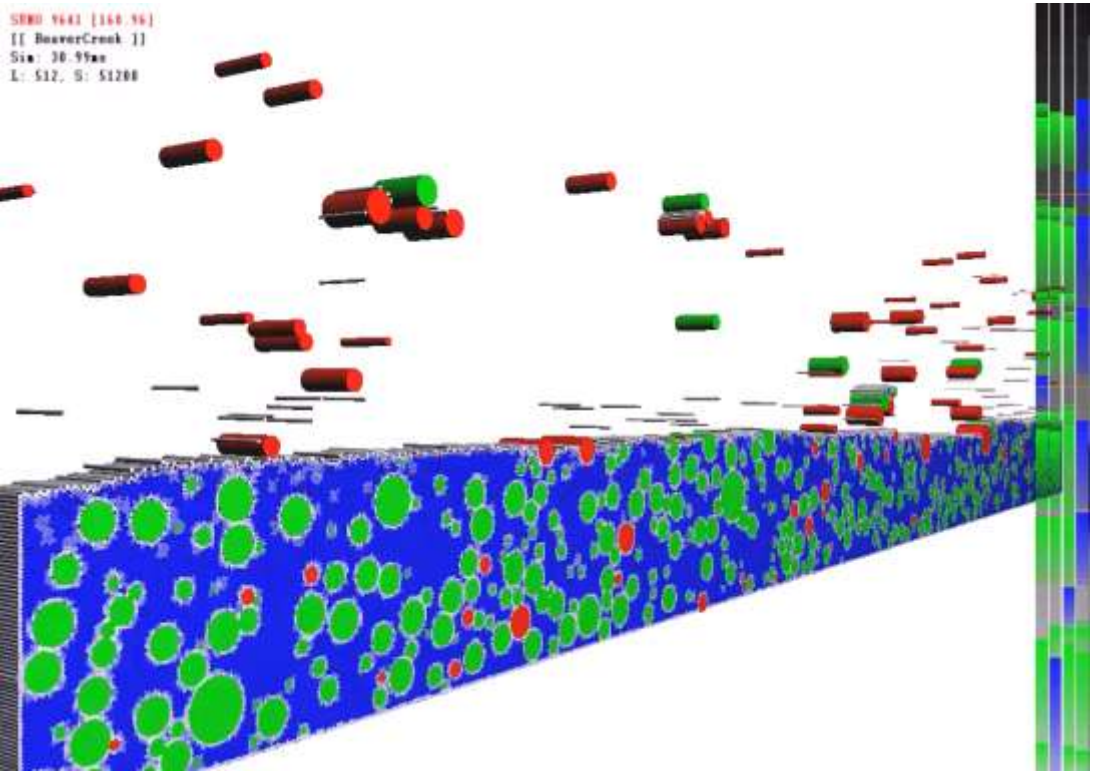


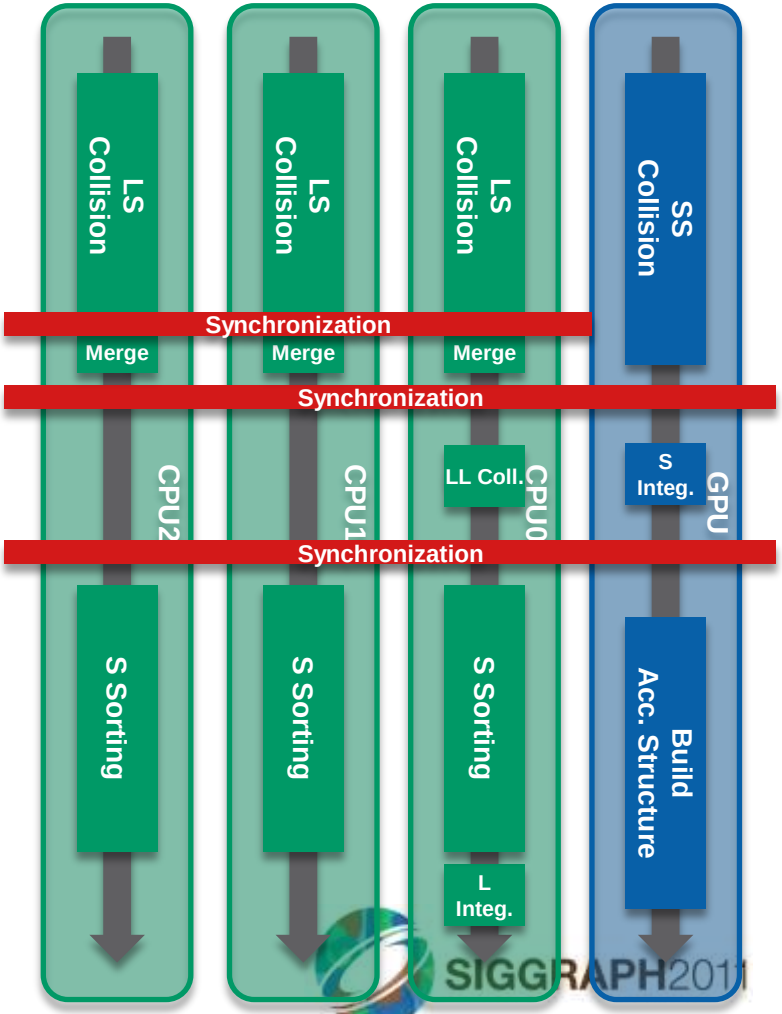
HETEROGENEOUS ERA

- AMD Fusion
  - Llano, Zacate
- Advantages
  - Shared memory
  - CPU & GPU can work together



HETEROGENEOUS PARTICLE BASED SIMULATION





Takahiro Harada, Physics Simulation on Fusion Architecture, AMD Fusion Developer Summit(2011)

REFERENCES

- Peter Kipfer, LCP Algorithms for Collision Detection using CUDA, GPU Gems 3(2007)
- Takahiro Harada et al. , Smoothed Particle Hydrodynamics on GPUs, Proc. of CGI(2007)
- Takahiro Harada, Real-time Rigid Body Simulation on GPUs, GPU Gems 3 (2007)
- Liu et al., Real-time Collision Culling of a Million Bodies on Graphics Processing Units, Siggraph Asia(2010)
- Takahiro Harada, Parallelizing the Physics Pipeline, GDC(2009)
- Richard Tonge, PhysX GPU Rigid Bodies in Batman, Game Programming Gems 8(2010)
- Rahul Sathe, Rigid Body Collision Detection on the GPU, Sig Poster(2006)
- Scott Le Grand, Broad-phase Collision Detection with CUDA, GPU Gems3(2007)
- Takahiro Harada, Physics Simulation on Fusion Architecture, AMD Fusion Developer Summint(2011)