

Geant4 On GPU



SEPTEMBER 10, 2012

DHRUVA T.B.
MENTOR –: JOHN APOSTOLAKIS

Project Goals



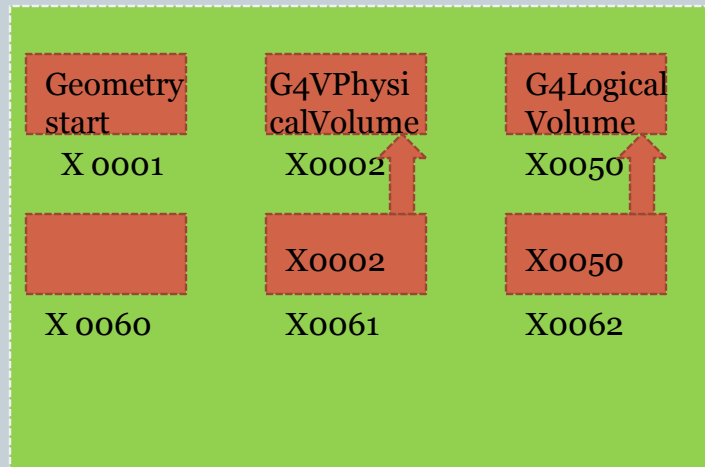
- Optimised Navigation for HEP geometries on GPU
 - Challenges :
 - ✦ Hierarchy of Volumes
 - ✦ Variable number of sub-volumes at each level (Zero to thousands)
 - ✦ Volumes made of many shapes : boxes, cylinders, spheres, ...
 - Existing Code - Otto Seiskari (2010):
 - ✦ Support four solids : Orb, Box, Tubs, Cons
 - ✦ Macros spanning CUDA and OpenCL
 - ✦ Navigation: Separate functions for few (Normal) or many (Voxel) sub-volumes – or Combined.

My Work : First Steps

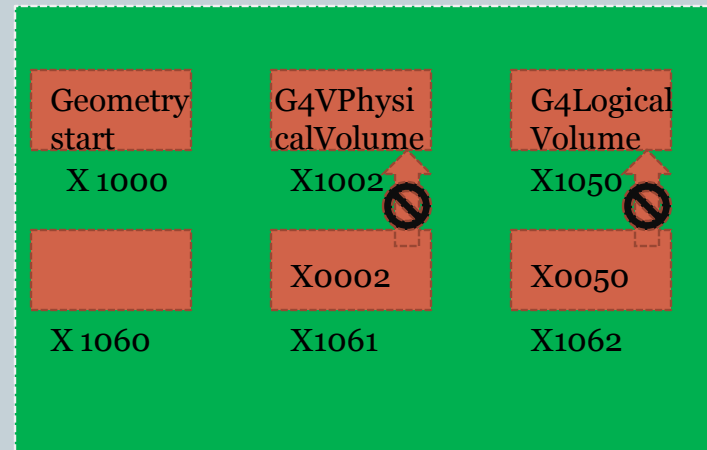


- Porting of Otto's code to run with OpenCL 1.1
 - New method of relocation of data structures

CPU



GPU



My Work



- **Automated Checks**
- **Existing Tests**
 - Toy1 : 2 volumes; A sphere in a box
 - Toy2 : 1000s of volumes; Spheres and Cones placed in a grid
 - SimpleCMS : 1643 solids; Simplification of CMS detector using supported solids
- **New Navigation algorithm**
 - Revamped algorithm
 - New way of Computing Steps
 - Taking GPU execution model into account

My Work : New Navigation



- **New Algorithm For Navigation**
 - Uses fast cache-like memory called Shared Memory
 - Optimized for GPU
 - With a 'Global Mode' for Debugging

Voxel Navigation :

Thread 1

- Box 1
- Cons
- Box 2

Thread 2

- Cons
- Box 3
- Box 4

New Navigation :

Boxes -4 ; Cons - 2

Thread 1 – Box 1

Thread 2 – Box 2

...



Thread 1 – Box 3

Thread 2 – Box 4

...

GPU Programming : Challenges



- **Memory Management**

- Global, shared and private memory equivalent to RAM, cache and registers
- Global and shared memory user managed
- Error on accessing global pointer from shared mem array
- Thousand of threads accessing memory at a time. Potential pitfall.

- **Barriers**

- To sync all threads
- Code will crash if all threads do not reach the statement.
- For instance, inside loops and conditionals

- **Debugging Errors and Crashes**

- Errors may not be very descriptive
- Code may crash without errors
- Use different approaches
- Examples -: Return statements; Load buffers and print from CPU

Results



- **Improvements to relocation method**
 - Macros to ensure 64 bit GPU compatibility
 - Tested on 64 bit Nvidia GPU
- **Checks to ensure correctness**
 - Relocation -: Have the geometry pointers updated correctly?
 - Distance -: Does the new track position correspond to the step taken ?
- **New Navigation algorithm implemented**
 - Runs on AMD GPU (32 bit)
 - Shared Memory Optimization not 'fully' implemented

Next Steps



- **Compare new algorithm with existing code**
 - Need to implement all arrays in shared memory before profiling
 - Testing on CUDA and OpenCL 64 bit GPUs
- **Documentation**
 - List of errors faced and ways to debug
 - Use of the macros
- **Support more (all?) of Geant4 geometry definition**
- **Test one complete example on GPU**



Thank You
Questions?



Backup Slides

Previous Work: G4VPhysicalVolume



```
typedef struct G4VPhysicalVolume
{
    G4RotationMatrix frot;
    G4ThreeVector ftrans;
    GEOMETRYLOC G4LogicalVolume *flogical;
    // The logicalvolume
    // representing the
    // physical and tracking attributes of
    // the volume
    GEOMETRYLOC G4LogicalVolume *flmother;
    // The current mother logical volume
}
G4VPhysicalVolume;
```