Multi-GPU Sort-Last Volume Visualization

Stéphane Marchesin, Catherine Mongenet, Jean-Michel Dischler

Louis Pasteur University, Strasbourg

30/04/08
Introduction

- Large scale visualization
  - Uses clusters
- Multi-GPU systems are a cheaper alternative
  - Different characteristics
  - Different optimization opportunities
  - Few work
  - Specific hardware
Related Works

- Sort-last vs Sort-first (Molnar et al.)
- Multipipe SDK (Bhaniramka et al.)
- Optimize compositing (Ma et al., Stompel et al.)
Approach

- Conduct a study
  - Compare 2 architectures
    - Specificities
    - Performance implications
  - Sort-last pipeline approaches
    - Different pipelines
    - Different optimizations
  - Resulting performance
Pipeline Optimizations

- Rendering
  - Brick size
  - Slicing vs Raycasting
  - Vertex submission
Pipeline Optimizations

- **Readback**
  - Data (red lines)
  - Full readback (white)
  - Screen-space bounding boxes (blue)
  - Screen-space aligned bounding boxes (yellow)
  - Span-based readback (green)
Pipeline Optimizations

- Composition
  - CPU-based composition
    - Read all subpictures
    - Compose on the CPU
    - Send back the result to the main GPU
  - GPU-based composition
    - Makes sense on a multi-GPU system
    - Read all subpictures but one
    - Send all the subpictures to the main GPU
    - Compose on the GPU
Results

Comparison

- Multi-GPU, Pentium D, GeForce 7800GT
  - 4 GPUs
  - Maximal number of GPUs
- Cluster, Athlon X2, GeForce 7800GT
  - 4+1 nodes
- Same codebase
- 2 datasets
  - Xmas tree $512^3$
  - Geological $1024^3$
Results: Brick Size With Slicing
Results: Brick Size With Raycasting

![Graph showing the relationship between brick size and frames per second for different GPU and dataset configurations.]
Results: Vertex Submission

Graph showing frames per second for different vertex submission methods and dataset sizes.
Results: Readback
Results: Scalability, CPU Compositing

![Diagram showing frames per second vs. resolution for different GPU configurations.](image-url)
Results: Scalability, GPU Compositing
Results: Compositing Speed

![Graph showing compositing speed results with different configurations.](image-url)
Results: Scalability
Results: Breakup Of The Times
Conclusions

- Slicing is still faster
  - On both architectures
  - Raycasting has slightly better quality
- Brick size
  - Matters especially in the multi-GPU case
- Readback
  - Overhead of reading small areas too high
  - Bounding rectangles is fastest
Conclusions

- **Composition**
  - Faster on the GPU in high PPS situations
  - Faster on the CPU in low PPS situations
    - Spares GPU time in the most draw-intensive situations

- **Multi-GPU and cluster performance are equivalent**
  - For the same number of GPUs

- **Multi-GPU is a competitive solution**
Future Works

- Tighter coupling between GPUs
  - Potential for speedup from exchanging information
- Further multi-GPU scalability
  - Not possible in a single machine
  - Clusters of multi-GPU machines
  - New hierarchical algorithms
Questions ?