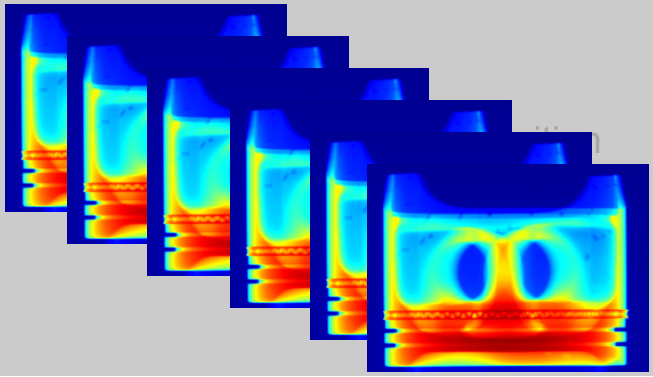
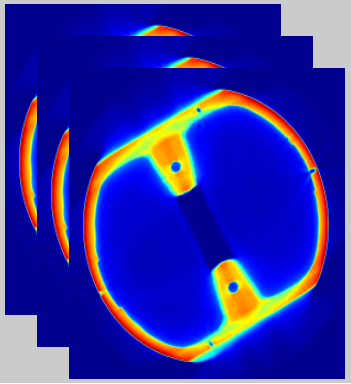
Exceptional service in the national interest

Sandia National Laboratories





An Irregular Approach to Large-Scale Computed Tomography on Multiple Graphics Processors Improves Voxel Processing Throughput

Edward S. Jimenez, Laurel J. Orr, and Kyle R. Thompson

Workshop on Irregular Applications: Architectures & Algorithms @ The International Conference for High Performance Computing, Networking, Storage, and Analysis (Supercomputing 2012)

November 11, 2012

U.S. DEPARTMENT OF ENERGY | National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Computed Tomography

- Computed Tomography (CT) is an indirect 3D imaging technique.
- Input: Set of X-ray images acquired about a center of rotation.
- Output: Three-dimensional approximation of internal and external structure
- Reconstruction: Convolution- Backprojection Algorithm (Feldkamp-Davis-Kress)
- Geometry and Configuration of CT System determines magnification
- Reconstruction algorithm is $O(n^4)$

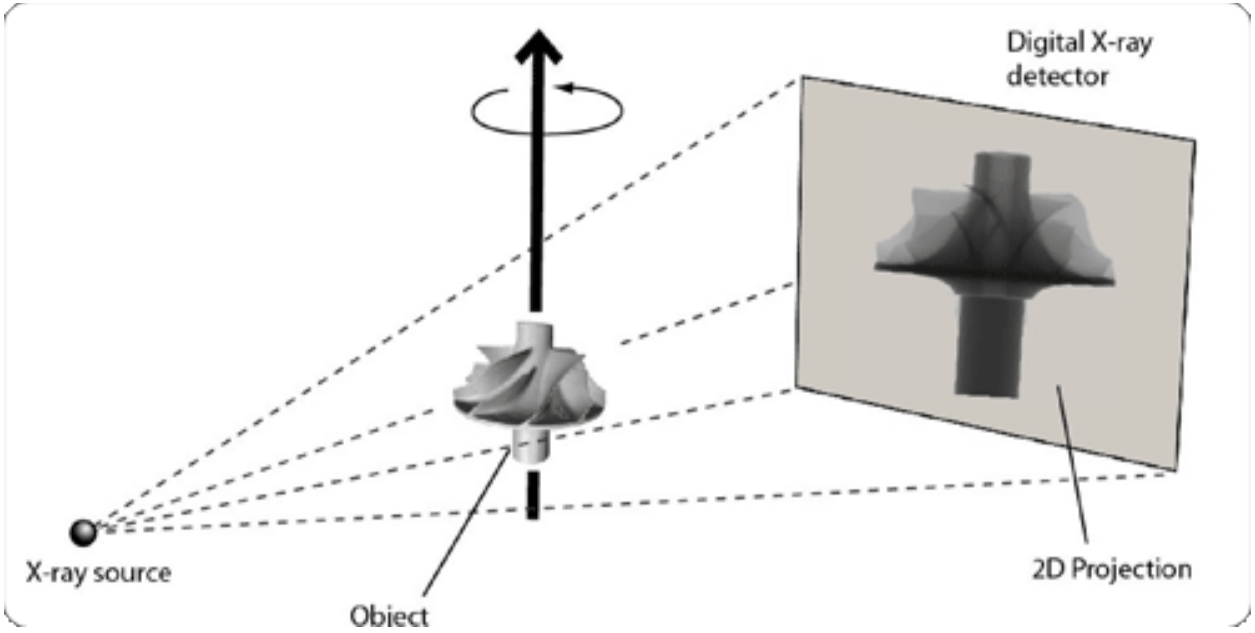


Image Source: http://www.xviewct.com/assets/images/how-ct-works.gif

GPU

Sandia National Laboratories

- Graphics Processing Units are coprocessors that handle image manipulation and now are being used for general purpose computing.
- Capable of Teraflops!
- This massive computational capability of GPUs can be harnessed for many applications.
  - Parallel computing environment
  - Fast dedicated memory
  - Fast Cache
- CT Reconstruction from projection images requires many arithmetic and trigonometric operations for every volumetric pixel (voxel).

CT on GPUs

Sandia National Laboratories

- "Porting" CT reconstruction on GPUs has shown major bottlenecks.
  - Usually not an issue with medical datasets.
  - Memory uploads/downloads to device (GPU).
  - What ratio of x-ray data to volume should be allocated?

- Traditional CPU-based code reconstructed one slice at a time
  - Predicable memory access even when multi-threaded.

- GPU-based reconstruction
  - Massively multithreaded environment creates scattered memory reads if large x-ray data is utilized per kernel launch.
  - Scattered Memory reads present for large volume storage too!
  - Suddenly reconstruction becomes an Irregular Problem!

Approach

Sandia National Laboratories

- Maximize resources by blocking x-ray data and sub-volumes.
- Counter Intuitive: *Maximize* x-ray data uploads to device!
  - Partition x-ray images and batch small x-ray image subsets
- Volume: Use most GPU memory for direct volume storage.
- Utilize GPU-Specific Hardware/Features
  - Massive parallelism
  - Texture memory/Texture Cache
  - Constant Memory
  - Data prefetch to pinned memory for fast upload
- Dynamic Task Partitioning

Implementation

Sandia National Laboratories

- CUDA Programming environment and C++
- Minimum requirements
  - Fermi-based Architecture
  - 1 GB Device memory
  - At least one x-ray sub image and one slice must fit simultaneously
- Allows for 1 – 8 GPUs per node
- Dynamic Partitioning determined by *slice-to-texture ratio* (*STR*)
- STR may not always be satisfied:
  - Resource maximization vs. Awkward task size
  - Reconstruction size – Too large or small?
  - Tail-end reconstruction

Dynamic GPU Tasking

Sandia National Laboratories

- For a given subvolume the amount of x-ray data necessary varies
  - Due to the geometry of the system.
  - Taken into account with STR to determine memory data allocation on device.
  - Typically, reconstruction along center slices require less data.

- Using OpenMP 2.0, a CPU thread controls one GPU in the system
  - Each GPU will usually be reconstructing sub-volumes of varying size
  - Load balancing difficult if subvolume is fixed for all GPUs

- No synchronization necessary for CPU threads while algorithm is executing.

- No synchronization necessary between GPU threads either.

- One atomic operation to update reconstruction progress and determine next subvolume to reconstruct.

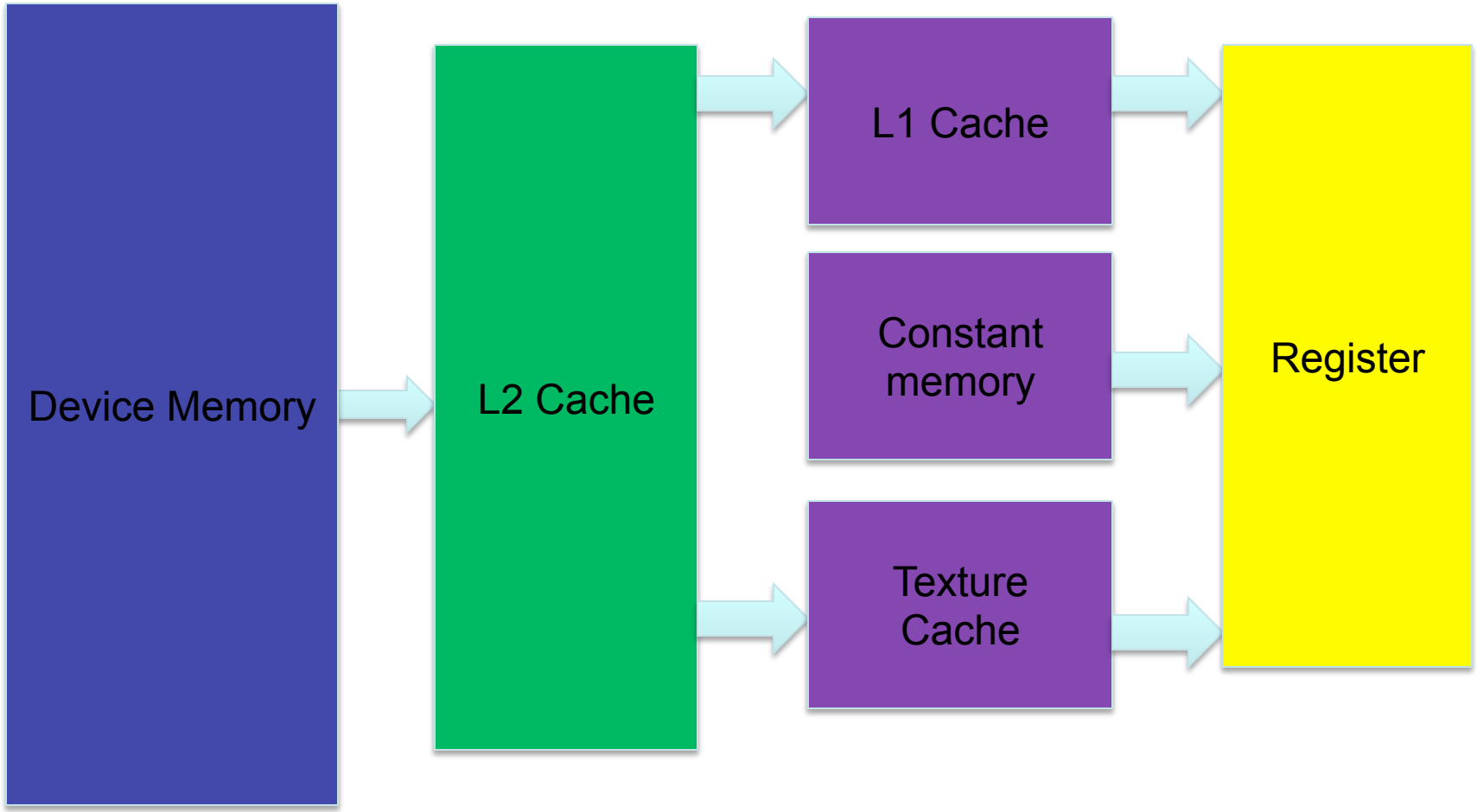
FDK Kernel Layout

Sandia National Laboratories

- **Input:** X-ray data, index, and size, subvolume data, index, and size, system geometry
- Get thread *ID* and voxel positions $p_1, \ldots, p_s$ based on *ID*
  - **if** Thread *ID* position within ROI **then**
    - **for** Every slice *j* in slice block **do**
      - Set register value to zero
      - **for** Every image *i* in image subset **do**
        - » Determine texture interpolation coordinate in image *i*
        - » Update register value with texture fetch and scaling
      - **end for**
      - Update voxel p_j in global memory with register value
    - **End for**
  - **End if**

GPU Cache Hierarchy

Sandia National Laboratories



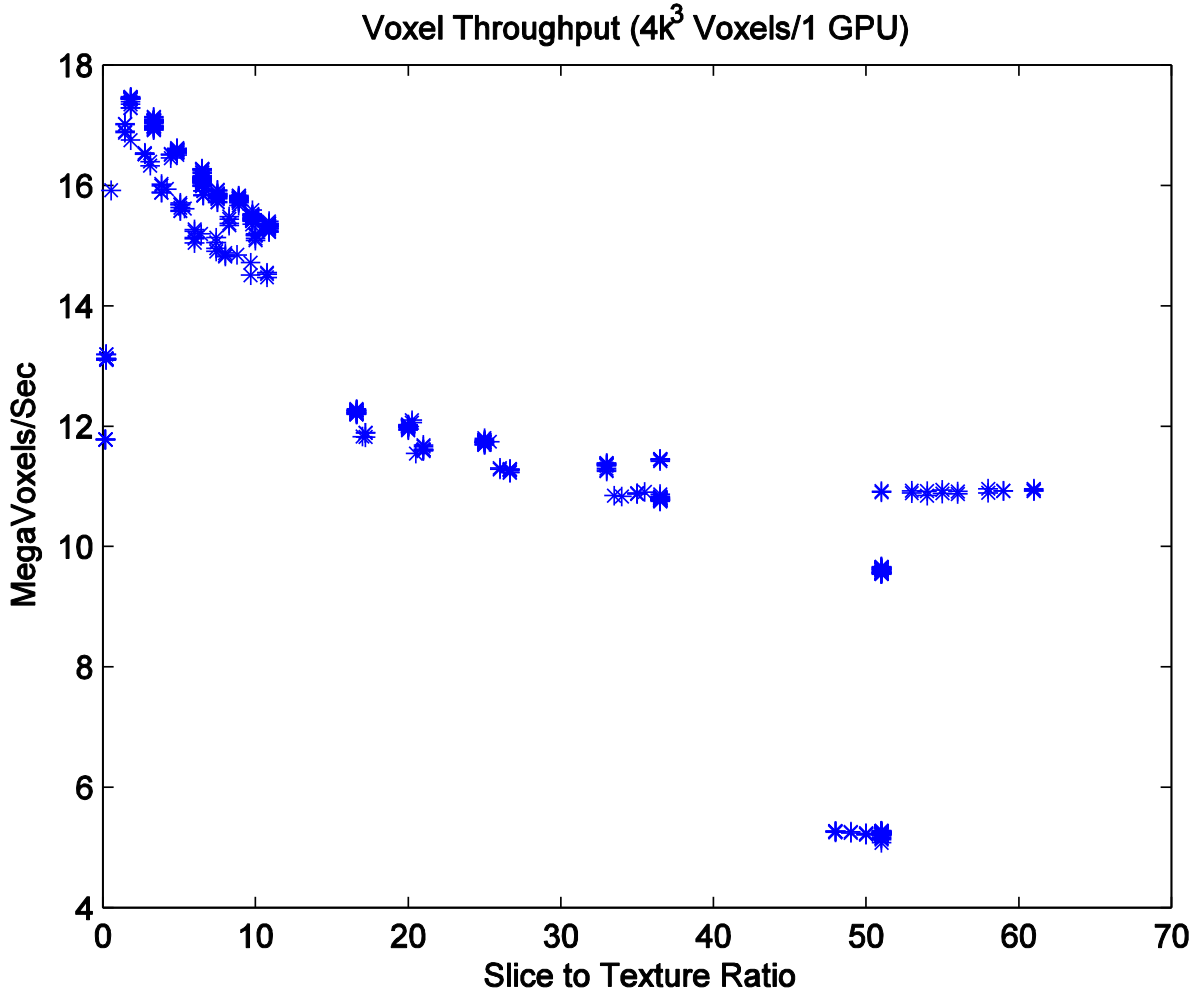
Evaluation

Sandia National Laboratories

- Supermicro workstation
  - Dual Hexacore Intel Xeon X5690 @ 3.46 GHz w/ hyper threading
  - 192 GB RAM
  - 4 PCI-E 2.0 x16 slots
- 2 NVidia S2090 Devices
  - 4 Tesla M2090 GPUs each (8 total)
  - Connected via 4 PCI-E host interface cards
- M2090
  - 6 GB GDDR5 memory apiece
  - 16 streaming multiprocessors (SM)
  - 768 KB L2 Cache (load, store, and texture operations)
  - 32 Compute cores per SM
  - 48 KB L1 Memory (explicitly set, shared memory not used)
  - 8 KB Constant Memory and Texture Cache
- Two datasets tested
  - 64 Gigavoxels
  - 1 Teravoxel

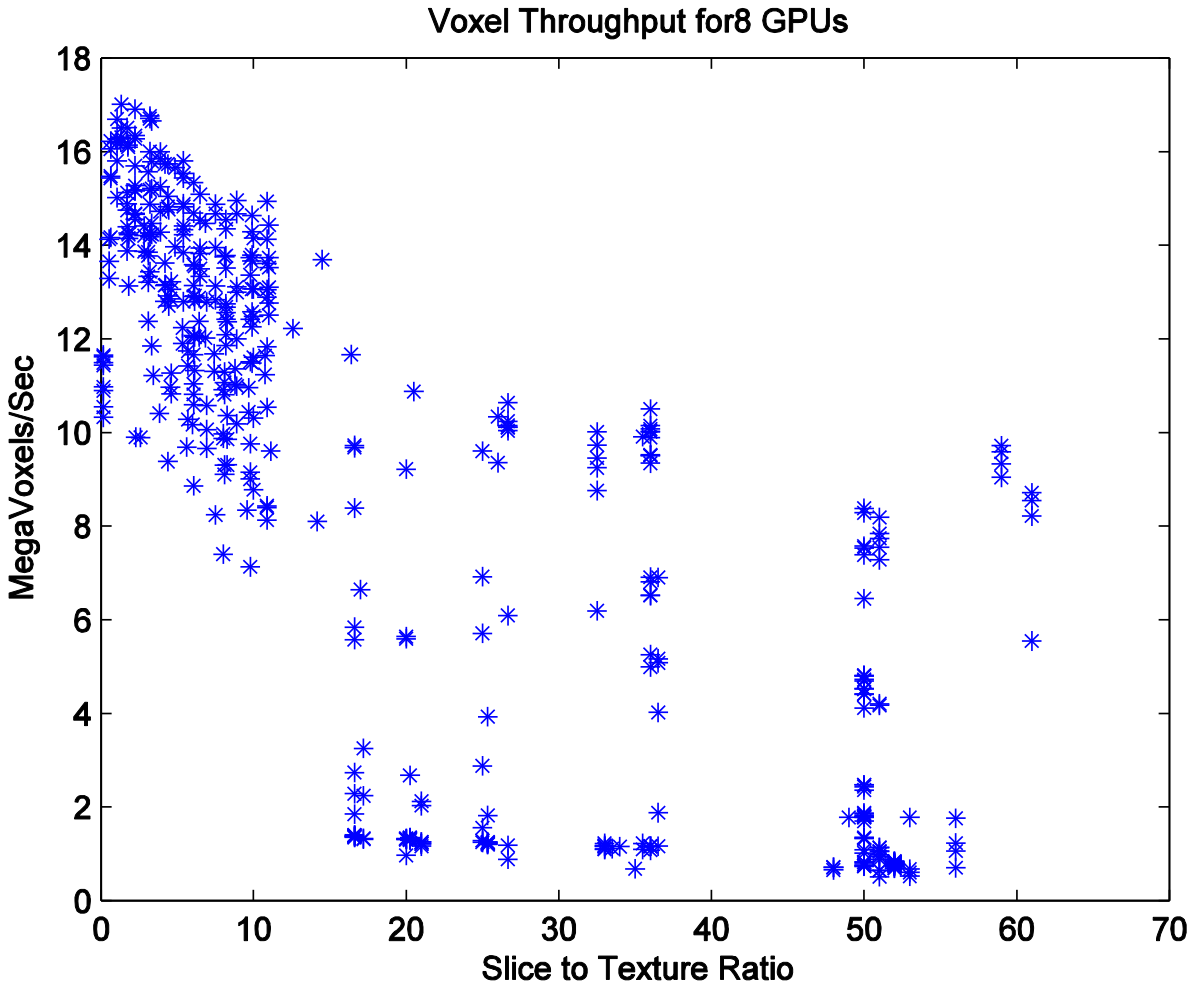
Results: Throughput 64 GV/ 1 GPU

Sandia National Laboratories



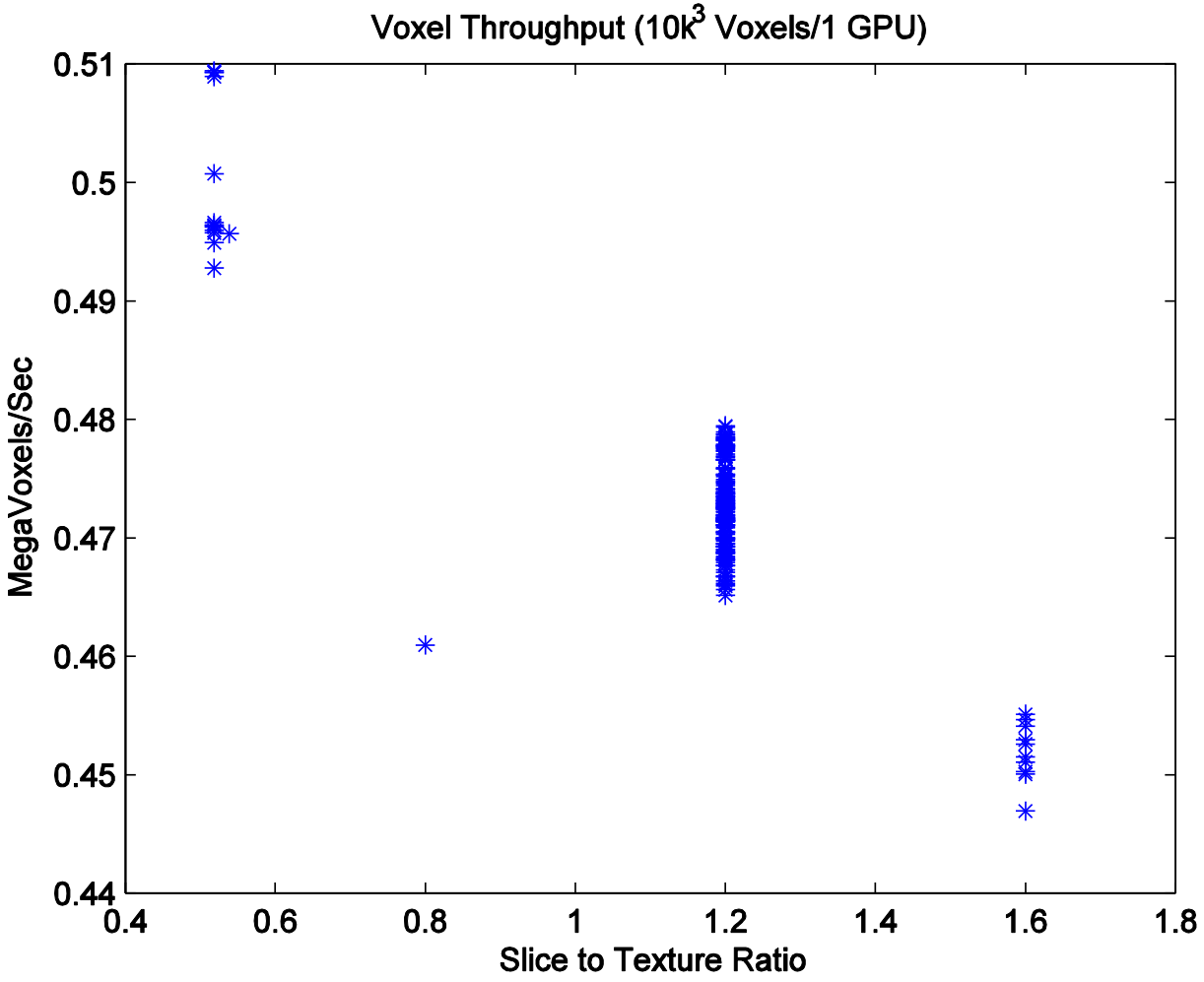
Results: Throughput 64 GV/ 8 GPUs

Sandia National Laboratories



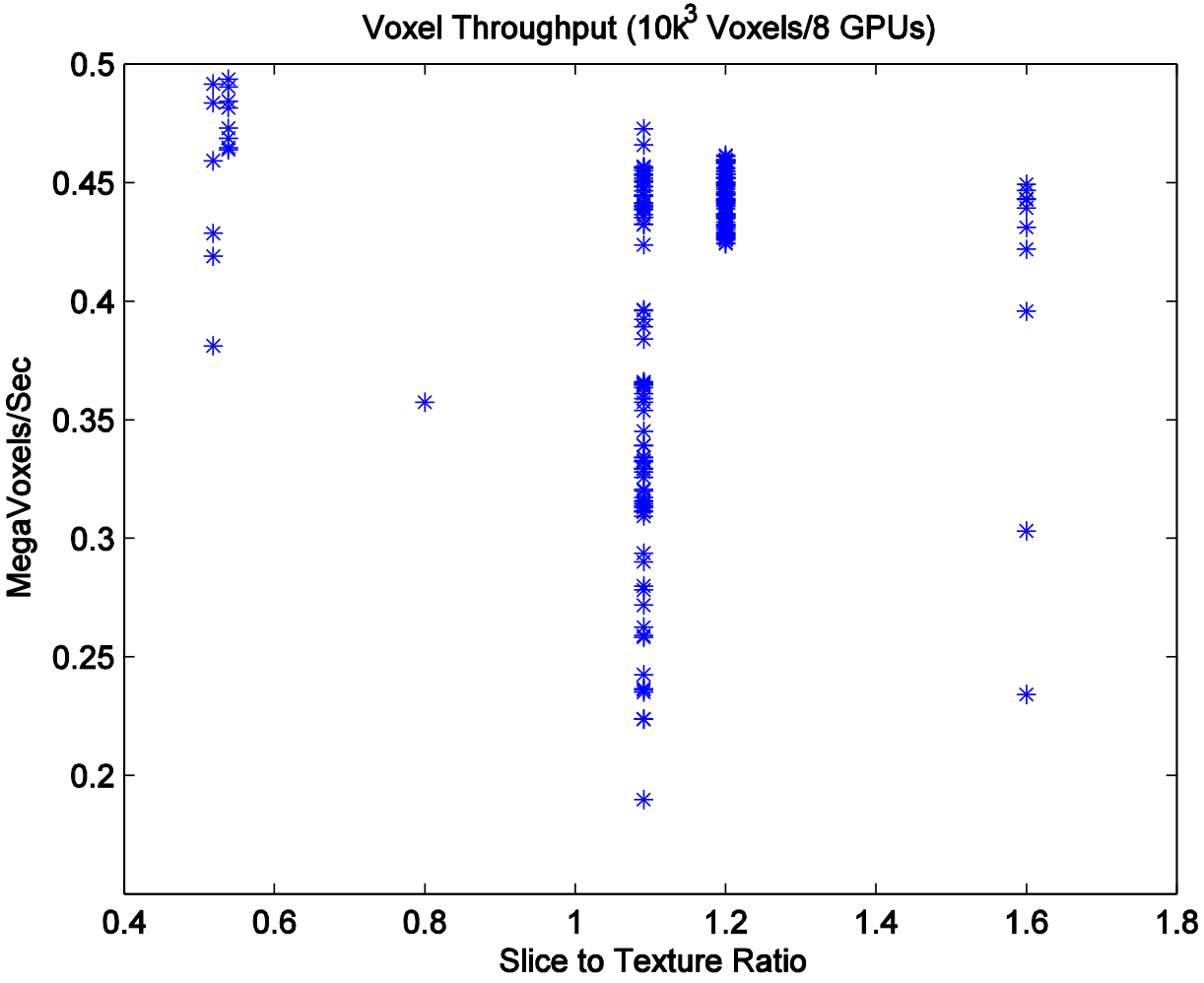
Results: Throughput 1 TV/ 1 GPU

Sandia National Laboratories



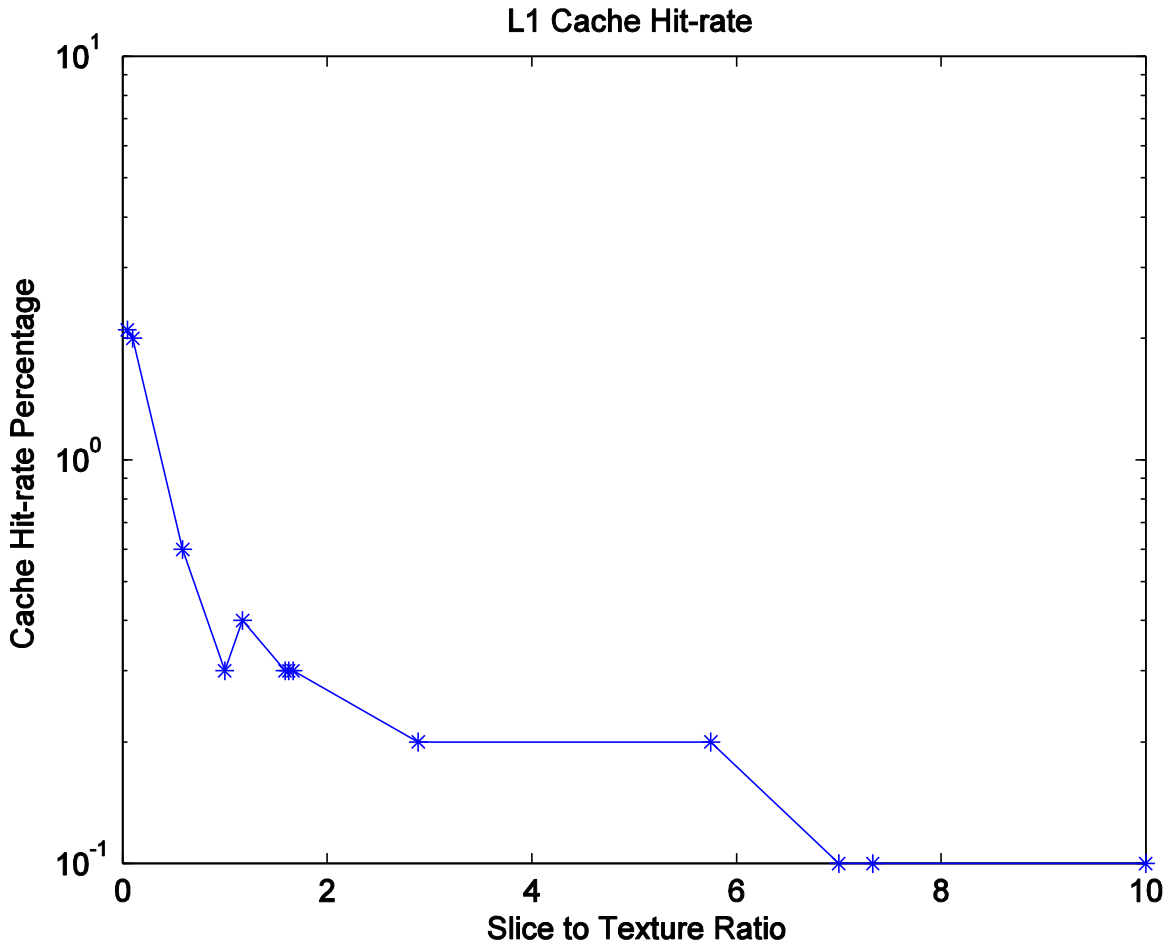
Results: Throughput 1 TV/ 8 GPUs

Sandia National Laboratories

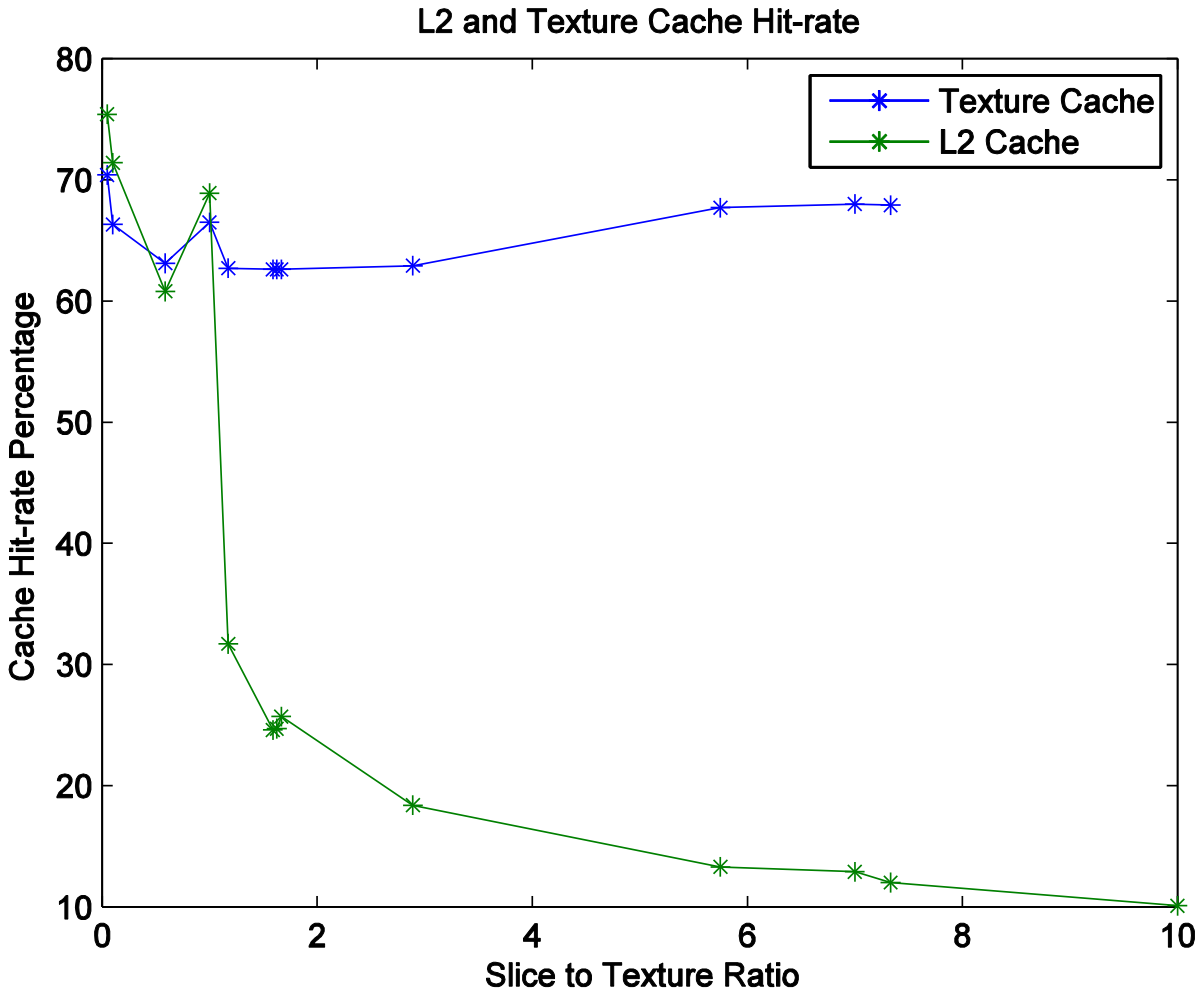


Results: L1 Cache Hit-rates

Sandia National Laboratories



Results: L2 and Texture Cache Hit-rates



Conclusion

Sandia National Laboratories

- Large-Scale CT Reconstruction algorithms clearly benefit from an Irregular approach
  - Massive parallelism has potential to destroy spatial locality
  - Counter Intuitive approach may create performance gains
  - Irregular approach improves voxel throughput by improving cache-hit rates
  - Small X-ray data batches and large subvolume tend to perform best.
- Are there other CPU-based algorithms that become irregular if implemented efficiently on a GPU?
- Thank you for your time!