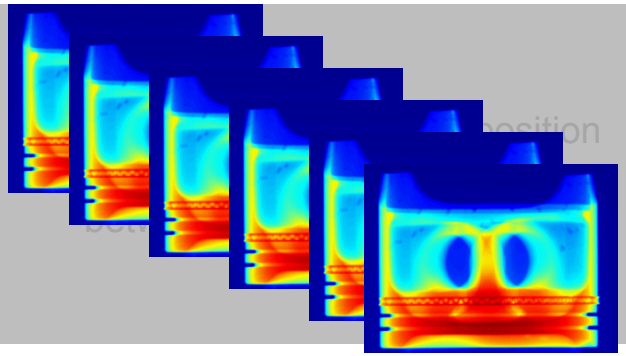
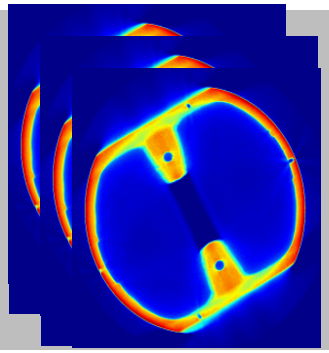
SAND2013-8959C

Exceptional service in the national interest

Sandia National Laboratories





A High-Performance and Energy-Efficient CT Reconstruction Algorithm for Multi-Terabyte Datasets

Edward S. Jimenez[(1)], Laurel J. Orr[(1)], and Kyle R. Thompson[(2)]

(1) Software Systems R&D

(2) Structural Dynamics and X-ray Non-Destructive Evaluation

U.S. DEPARTMENT OF ENERGY | NNSA

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. SAND NO. 2011-XXXXP

Introduction

Sandia National Laboratories

- Graphics processors have been used for many non-graphics applications with tremendous performance improvements.
- Medical-Scale Computed Tomography is one such example.
  - Reconstruction now can be done on the order of seconds.
- Industrial-Scale Computed Tomography
  - Medical-Scale algorithms frequently do not scale to large-scale data
  - Many industrial applications do not have a fixed acquisition geometry
  - Due to arbitrary acquisition configurations, Large-Scale CT reconstruction is an Irregular Problem

Why Graphics Processors?

Sandia National Laboratories

- Well known facts:
  - Massively Parallel Architecture
  - Fast Device Memory
- Lesser Known facts:
  - Unique Cache structure
  - User Configurable Cache
  - Instruction Ordering
  - Pinned-Memory
- A GPU can still require days to complete a large reconstruction

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

- We propose an optimized kernel that can support multiple GPUs working simultaneously on a single system.
- Must optimize computation to an irregular memory access pattern
- Reading and Writing Data to and from storage must be approached intelligently to minimize GPU downtime.
- The approach must be capable of reconstructing almost arbitrarily sized datasets.
- Must have reasonable energy requirements
  - Must not require a large cluster

GPU Kernel

Sandia National Laboratories

- Counter Intuitive: Maximize x-ray data uploads to device
  - Small pieces of x-ray data input forces cache hit-rate improvements
  - Ameliorate transfer times with pinned-memory
  - Effective memory bandwidth on device is improved
- More device memory available for volume storage
  - Less x-ray input data allows more image planes to be reconstructed simultaneously per device
- Texture Memory: Store x-ray input data as textures
  - Exploit hardware-based interpolation
  - Exploit texture cache, as fast as L1-cache

GPU Kernel Continued…

Sandia National Laboratories

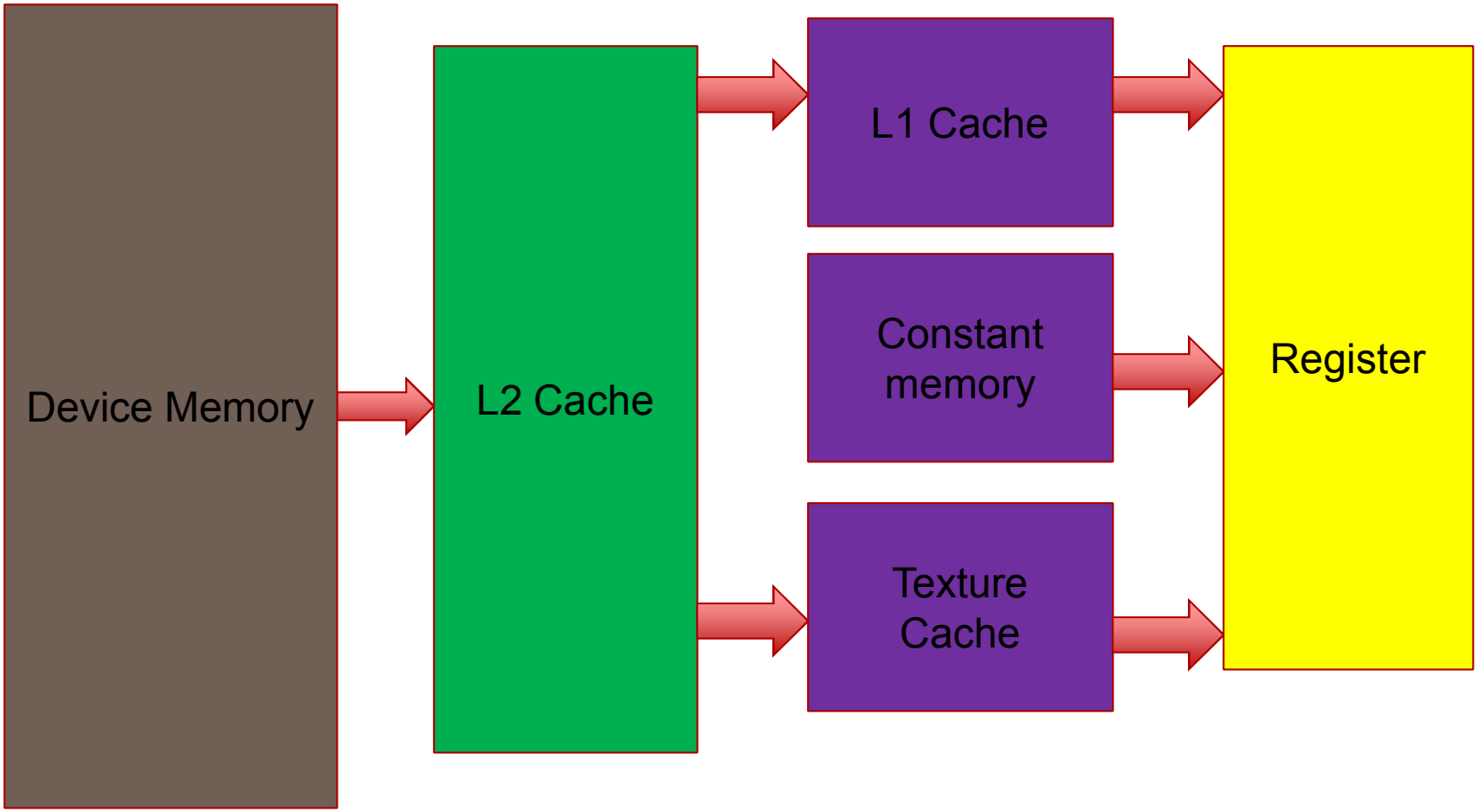
- Constant Memory: User configurable cache
  - Small amount of on-chip cache
  - Store static values such as geometry parameters
  - Like texture cache, reduces L1-cache pressure

- Block x-ray data and reconstructed sub volumes
  - While x-ray data block is applied, use a CPU thread to queue the next block of input data
  - Overlapping tasks improves efficiency and performance

- Dynamic Task Partitioning to determine blocking based on
  - System parameters
  - Reconstruction task size

GPU Cache Hierarchy

Sandia National Laboratories



Handling the I/O Bottleneck

Sandia National Laboratories

- The GPU kernel design described can accommodate up to 8 GPUs on a single system

- New Problem: Host storage cannot keep up!
  - Up to 48 GB worth of image planes created every few seconds to minutes
  - GPUs requiring input at an increased rate due to improvement
  - GPUs needlessly idling!

- Solution: A modularized Approach
  - MIMD-like approach
  - CPU threads performing various tasks simultaneously

I/O Bottleneck Continued

Sandia National Laboratories

- Serialized Approach
  - 1. Upload X-ray input data
  - 2. Run GPU kernel
  - 3. Download reconstructed sub volume
  - 4. Write to storage media (GPU Idle)
  - 5. Read next input subset (GPU Idle)

- Modularized Approach
  - 1. Upload X-ray input data
  - 2. Run GPU kernel while queuing next set of input data
  - 3. Download sub volume
  - 4. Write to storage and run next kernel

Evaluation

Sandia National Laboratories

- Supermicro workstation
  - Dual Octo-core Intel Xeon E-2687W @ 3.1 GHz w/ hyper threading
  - 512 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

Evaluation Continued

Sandia National Laboratories

- Two datasets tested
  - 64 Gigavoxels
    - 4000 image planes, 16 megapixels each
    - 1800 16-megapixel x-ray images
  - 1 Teravoxel (Synthetic dataset)
    - 10,000 image planes, 100 megapixels each
    - 10,000 100-megapixel x-ray images

- Measure
  - Performance Time
  - Energy Consumption
  - Scalability

- Performance compared to:
  - CPU-based implementation (Hybrid OpenMP/MPI-based Implementation)
  - “Naïve” GPU-based Implementation

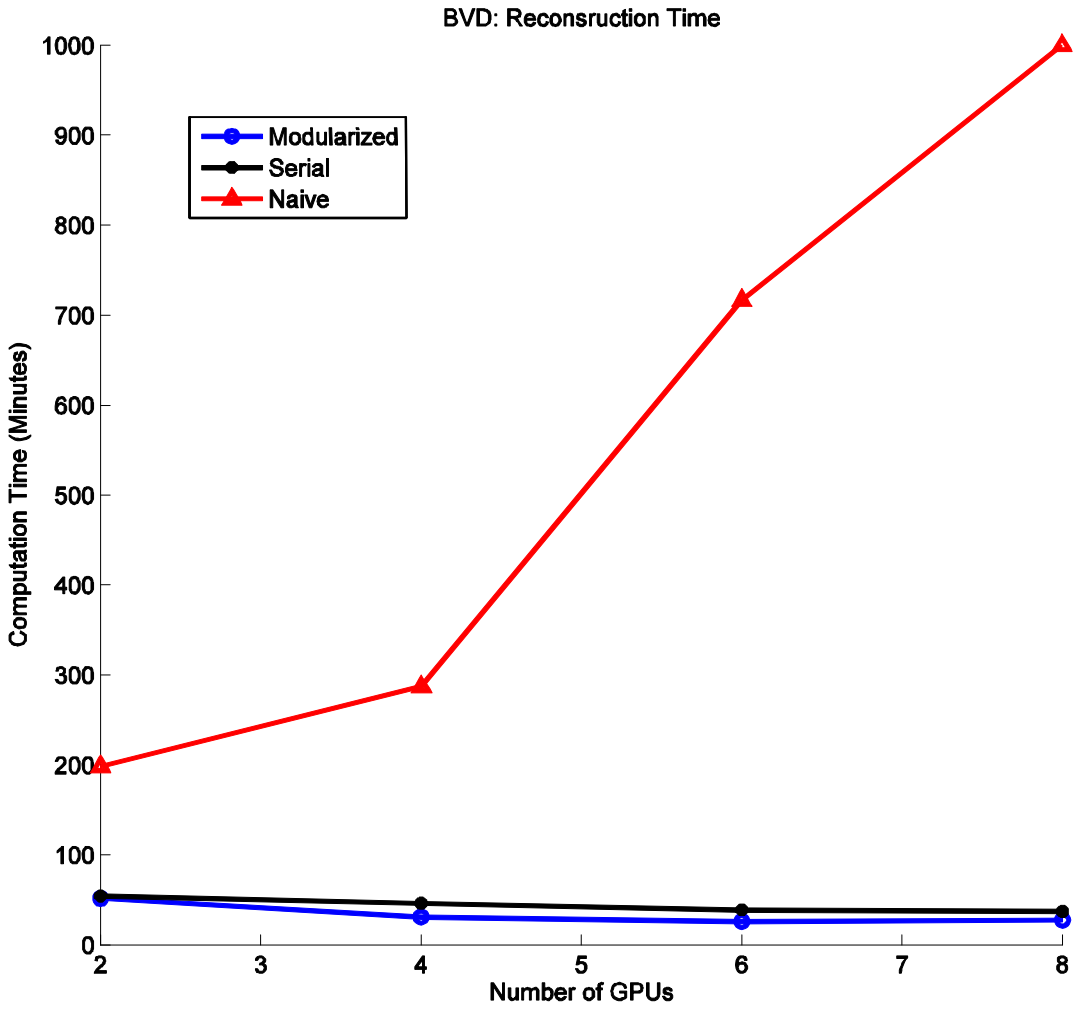
Naïve Approach

Sandia National Laboratories

- This approach is a GPU-based solution
  - Take a CPU-based implementation and “Port” to a kernel
- Approach
  - Only exploit massive parallelism
  - Reconstruct single image plane per kernel launch
  - No data processing overlap
  - No device optimization
  - No kernel optimization
  - No hardware interpolation

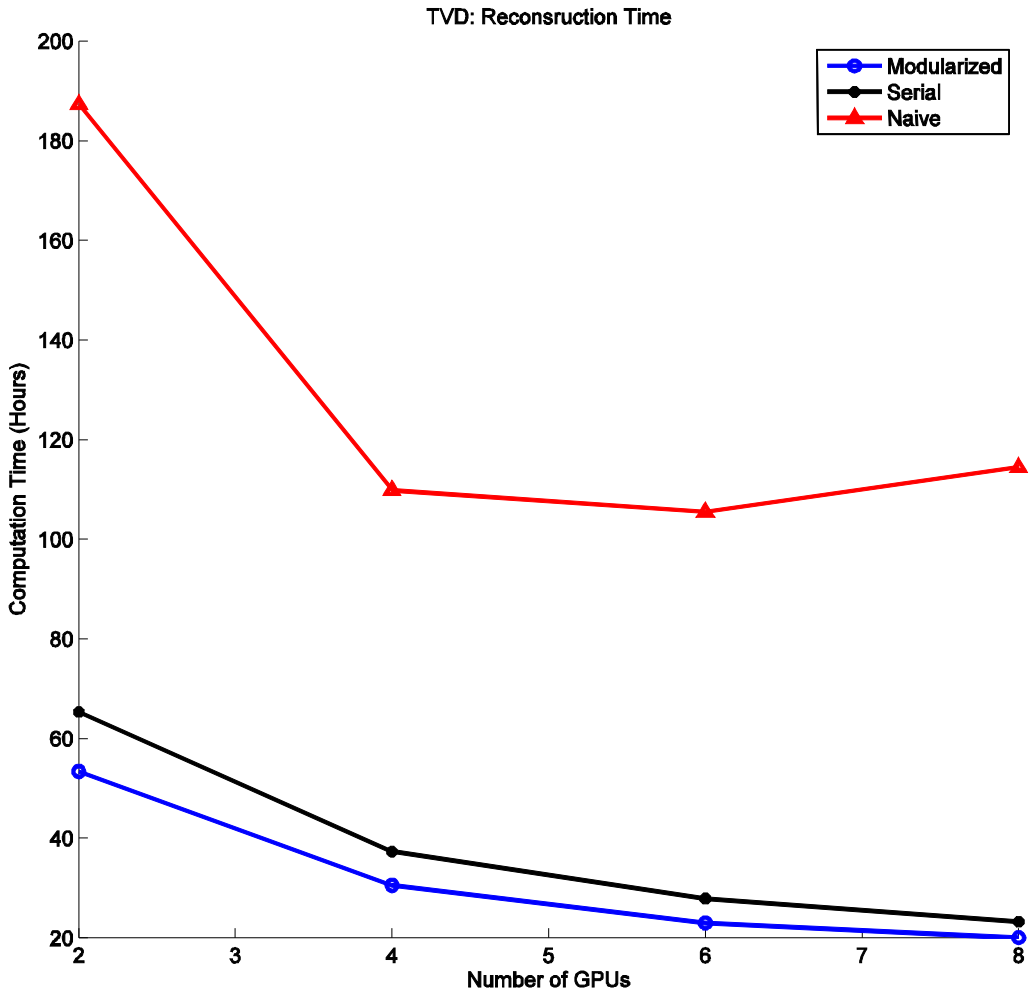
Computational Performance: 64 Gigavoxels

Sandia National Laboratories



Note: CPU Implementation required over 36 hours

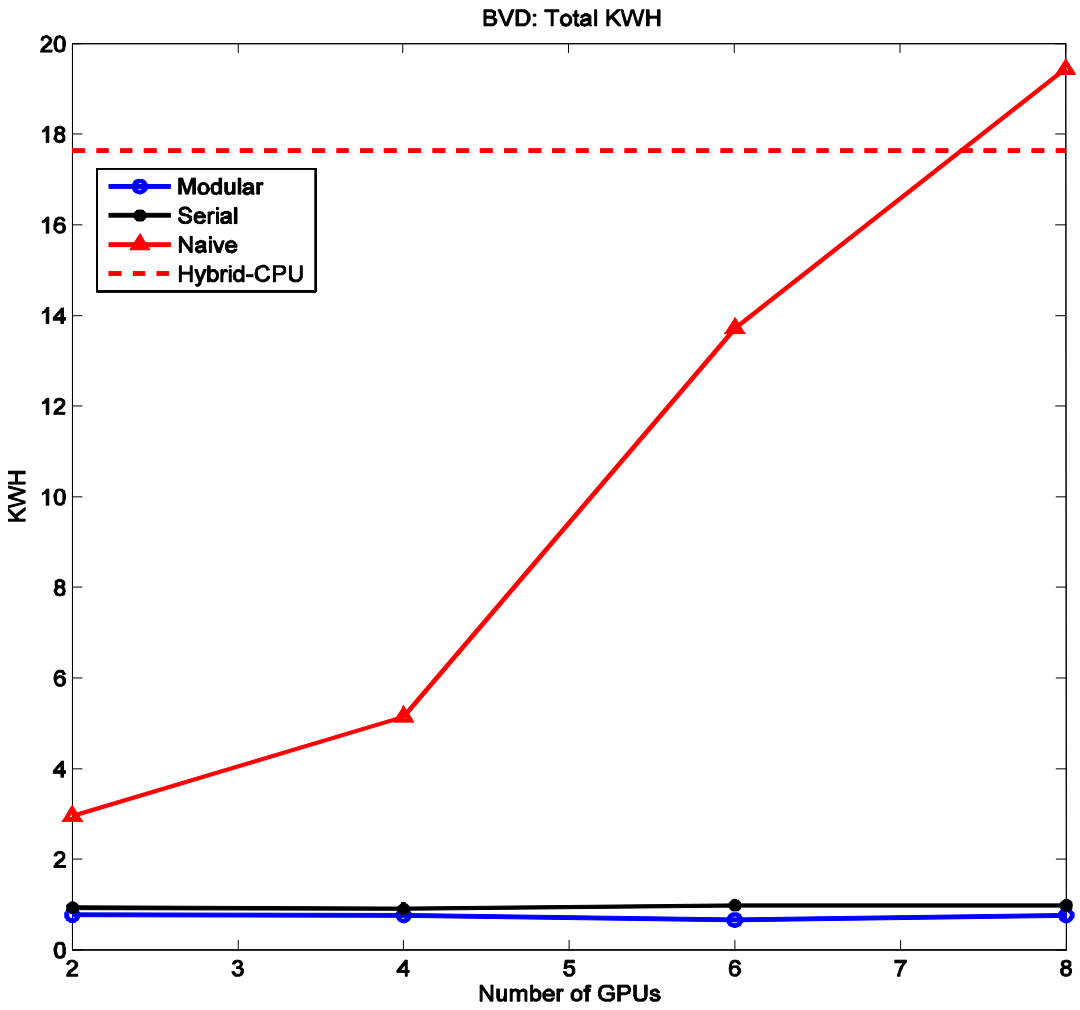
Computational Performance: Teravoxel



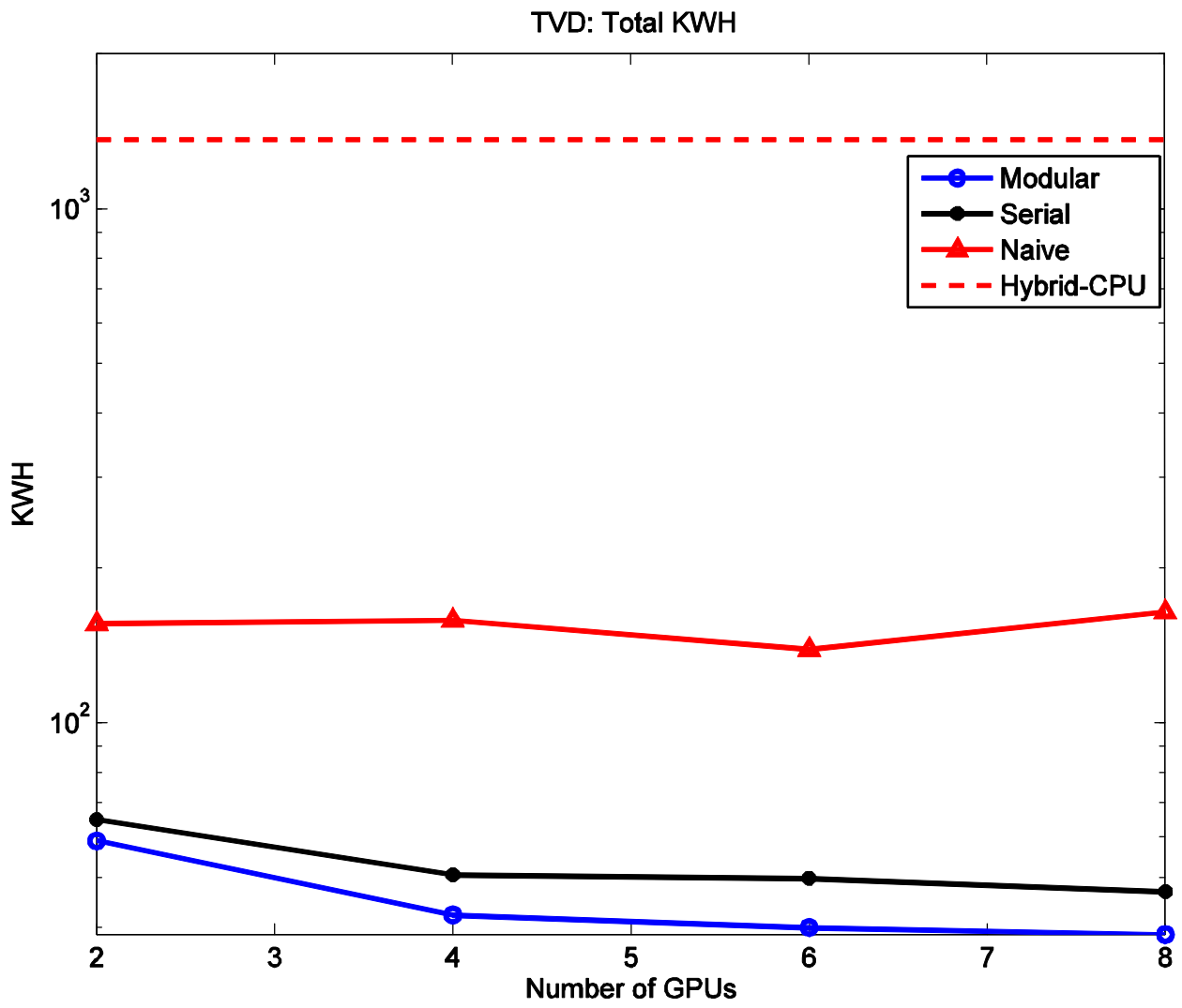
Note: CPU Implementation required over 2500 hours

Energy Performance- 64 Gigavoxels

Sandia National Laboratories

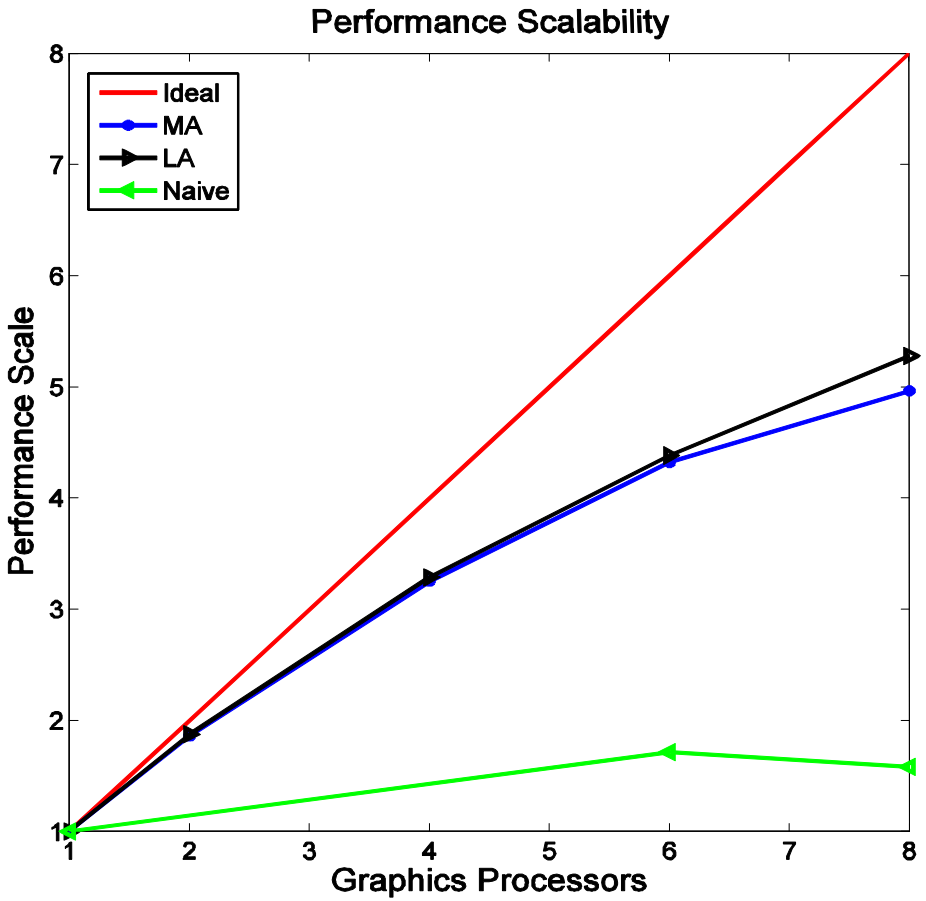


Energy Performance- Teravoxel



Scalability

Sandia National Laboratories



Conclusions

Sandia National Laboratories

- Industrial Scale Computed Tomography can clearly benefit from a multi-GPU based approach.
- Mindfulness of the Irregularity of Large-Scale Reconstruction can result in a flexible algorithm.
- Computational Performance is dramatically improved while improving energy efficiency.
- Does the same apply to iterative algorithms?