GPU-based Decompression for Medical Imaging Applications

Al Wegener, CTO
Samplify Systems

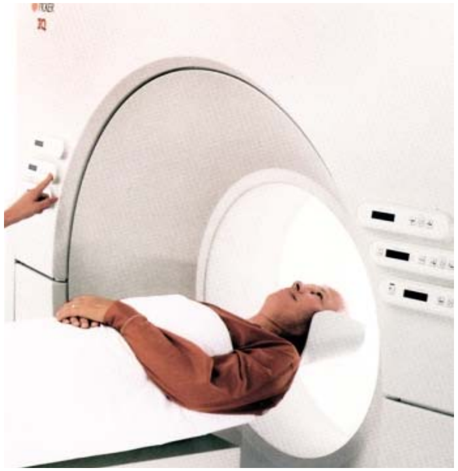
160 Saratoga Ave. Suite 150
Santa Clara, CA 95051
sales@samplify.com
(888) LESS-BITS
+1 (408) 249-1500

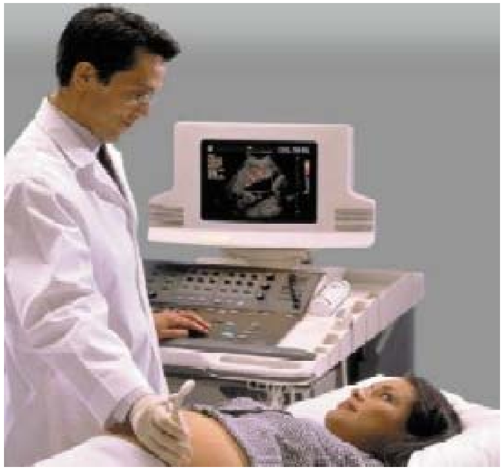
Outline

- Medical Imaging Bottlenecks
- Compression: Benefits for Medical Imaging
- Compression for Computed Tomography
- Decompression Alternatives: FPGA, CPU and GPU
- Speeding up GPU-based Decompression
- Reducing I/O Bottlenecks for CUDA Applications
- Conclusions

samplify SYSTEMS®

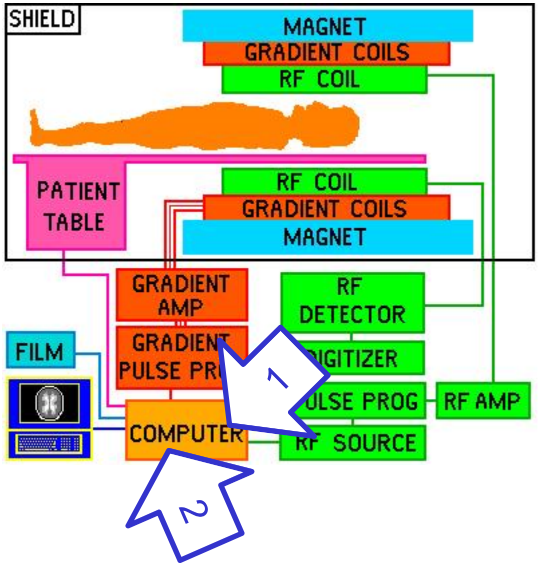
Motivation: Faster Imaging Systems





Parameter	CT	Ultrasound	MRI
Channel Count	64,000 → 320,000	128 → 2048	8 → 128
Sample Rate	3k → 15k samp/sec	40M → 65 M samp/sec	10M → 100M samp/sec
Bits per sample	18 - 22	10 - 14	14 - 16
Total bandwidth	***15 GB/sec***	***250 GB/sec***	***26 GB/sec***
Storage per scan	***900 GB***	***5 – 15 TB***	***2 – 3 TB***

Imaging Bottlenecks



CT Scanner:

1. Slip ring ($5k-$30k)
2. Disk drive array ($8k - $50k)

Ultrasound:

1. ADCs → beamformer ($1k - $5k)
2. Beams → Display ($1k - $2k)

MRI:

1. ADCs → console ($500 - $5k)
2. Image recon RAM ($200 - $2k)

Compression Benefits for Medical Imaging

Modality	Bottleneck	Compression Ratio	Before Prism™	After Prism™
CT	Slip ring	2:1	$32,000	$18,000
	RAID arrays	2:1	$40,000	$20,000
Ultrasound	ADC to beamformer	2.5:1	$3,000	$1,400
MR	Optical cables	4:1	$8,000	$2,000
	Blade RAM	4:1	4 GB x 8	1 GB x 8
GPU	Host RAM	2:1	10 GB/sec	20 GB/sec

Prism™ Lossless Results on CT Signals

Data Set	Description	Lempel-Ziv	Samplify Prism	Golomb-Rice	Huffman	Prism CT™
1	Low signal phantom (shoulders)	1.25	1.76	1.83	1.82	1.98
2	Low signal phantom (shoulders)	1.30	1.61	1.80	1.78	2.07
3	Low signal phantom (shoulders)	1.31	1.78	1.81	1.82	2.06
4	Low signal phantom (shoulders)	1.15	1.80	1.79	1.85	1.91
5	Low signal phantom (shoulders)	1.21	1.71	1.82	1.80	2.07
6	Low signal phantom (shoulders)	1.21	1.75	1.85	1.83	2.08
7	48 cm acrylic phantom (semi-circle)	1.19	1.72	1.83	1.83	2.08
8	48 cm acrylic phantom (semi-circle)	1.09	1.71	1.76	1.82	1.90
9	Air scan	1.10	1.63	1.75	1.77	2.75
10	Offset scan	1.23	1.72	1.80	1.80	2.06
11	35cm poly (65mm Offset in Y)	1.35	1.78	1.86	1.84	2.14
12	35cm poly (65mm Offset in Y)	1.08	1.48	1.53	1.64	1.90
13	Anatomical chest phantom	1.08	1.48	1.60	1.64	1.97
14	Anatomical head phantom	1.09	1.48	1.61	1.65	2.11
15	Catphan low contrast detectability	1.11	1.50	1.63	1.69	2.22
16	Catphan low contrast detectability	2.13	1.63	1.67	1.76	2.38
17	Erlangen abdomen phantom	1.31	1.86	1.65	1.80	1.88
18	Erlangen head phantom	1.21	1.71	1.54	1.60	1.73
19	Erlangen thorax phantom	1.31	1.88	1.57	1.78	1.89
20	Patient head scan 1	1.08	1.46	1.65	1.48	1.61
21	Patient head scan 2	1.09	1.46	1.65	1.46	1.59
22	Patient head scan 3	1.09	1.47	1.65	1.48	1.63
23	Patient head scan 4	1.10	1.49	1.67	1.52	1.66
24	Patient head scan 5	1.10	1.49	1.67	1.52	1.67
25	Patient head scan 6	1.10	1.49	1.67	1.52	1.68
26	Patient chest scan	1.07	1.45	1.63	1.48	1.69
	MEAN COMPRESSION RATIO	1.21	1.63	1.70	1.69	1.95
	MEAN BITS PER SAMPLE	13.27	9.83	9.39	9.46	8.20
	BEST COMPRESSION	2.13	1.88	1.86	1.85	2.75
	WORST COMPRESSION	1.07	1.45	1.53	1.46	1.59

Best algorithm: Prism CT

Source: A. Wegener, R. Senzig, et al., "Real-time compression of raw computed tomography data: technology, architecture, and benefits," *SPIE Medical Imaging Conference*, Vol. 7258-88 (Feb 2009).

Best mean: 1.95

Highest min: 1.59

Radiologist Scoring of Prism™ Lossy @ 3:1

JAN. 7. 2009 4:34PM LUCAS CENTER RADIOLOGY NO. 4649 P. 1

STANFORD
SCHOOL OF MEDICINE
Stanford University Medical Center

DEPARTMENT OF RADIOLOGY
Lucas MRS Imaging Center

- ☐ Maggie Bos (650) 725-4936
- ☐ Lin Davis (650) 721-6980
- ☐ Teresa Newton (650) 725-4933
- ☐ Michelle Christierson (650) 723-7863
- ☐ Mandeep Kaur (650) 498-6499
- ☐ Lanzie Rivera (650) 723-9629
- ☐ Donna Cronister (650) 723-8205
- ☐ Marlys LeSene (650) 723-9740
- ☐ Lakeesha Winston (650) 725-8432

From: BOB HERFKENS
To: AL WEGENER
Re: DATA
DATE: 1/7/09 Pages: 2
CC:

☐ URGENT ☐ FOR REVIEW ☐ FOR COMMENT ☐ PLEASE REPLY PLEASE RECYCLE

no discernable or un discernable difference
Bob

Courier/Shipping Address
1201 WELCH ROAD, PALO ALTO, CA 94304

JAN. 7. 2009 4:34PM LUCAS CENTER RADIOLOGY NO. 4649 P. 2

Samplify Image Comparison Study

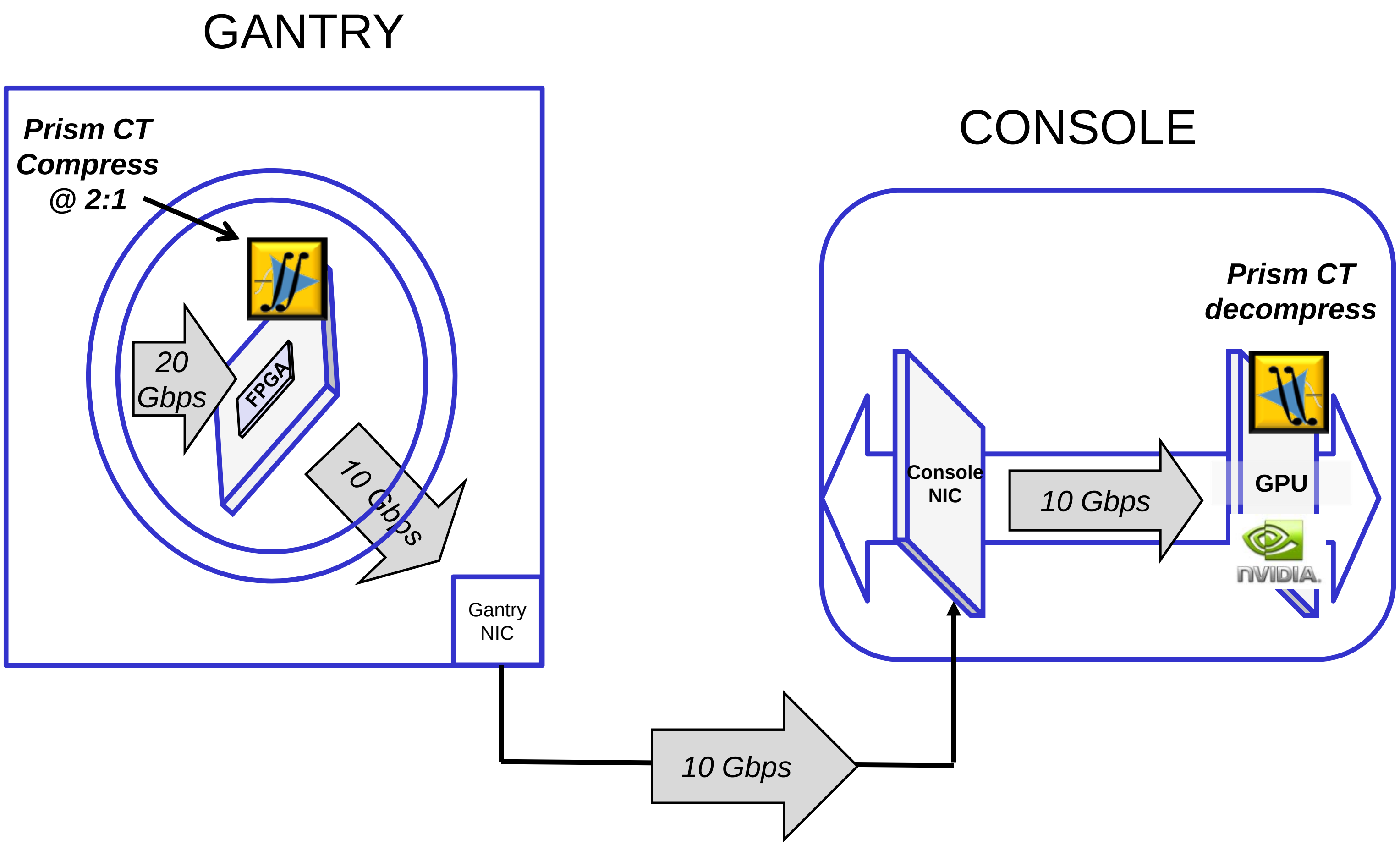
Date: 6 January 2008
Reviewer: Dr. Robert J. Herfkens, Stanford University

For image comparison scores of 2 or 3 (moderately or obviously different), please also answer the following Yes/No question: and this A/B question.

Considering the image regions where the two images differ, would your disease diagnosis (or lack thereof) have changed?

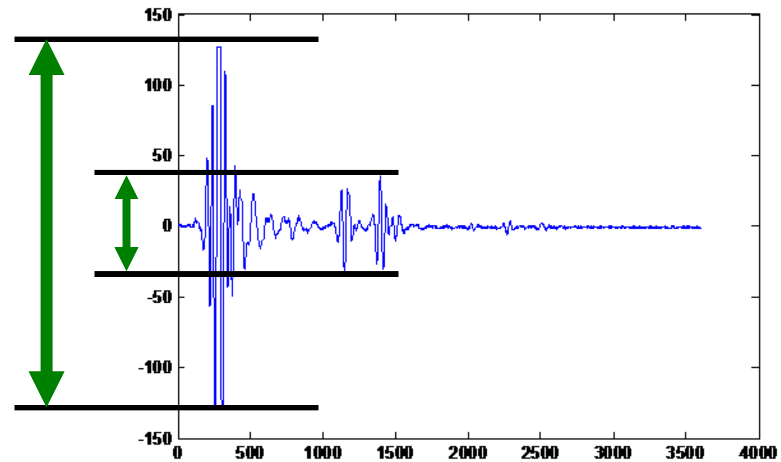
	ABOUT THE SAME	SLIGHTLY DIFFERENT	MODERATELY DIFFERENT	OBVIOUSLY DIFFERENT	Would the worse image change your diagnosis?	IF 2 OR 3, WHICH IMAG DO YOU LIKE BETTER?
1A vs. 1B:	0	1	2	3	YES NO	A B
2A vs. 2B:	0	1	2	3	YES NO	A B
3A vs. 3B:	0	1	2	3	YES NO	A B
4A vs. 4B:	0	1	2	3	YES NO	A B
5A vs. 5B:	0	1	2	3	YES NO	A B
6A vs. 6B:	0	1	2	3	YES NO	A B
7A vs. 7B:	0	1	2	3	YES NO	A B
8A vs. 8B:	0	1	2	3	YES NO	A B
9A vs. 9B:	0	1	2	3	YES NO	A B
10A vs. 10B:	0	1	2	3	YES NO	A B
11A vs. 11B:	0	1	2	3	YES NO	A B
12A vs. 12B:	0	1	2	3	YES NO	A B
13A vs. 13B:	0	1	2	3	YES NO	A B
14A vs. 14B:	0	1	2	3	YES NO	A B
15A vs. 15B:	0	1	2	3	YES NO	A B
16A vs. 16B:	0	1	2	3	YES NO	A B
17A vs. 17B:	0	1	2	3	YES NO	A B
18A vs. 18B:	0	1	2	3	YES NO	A B

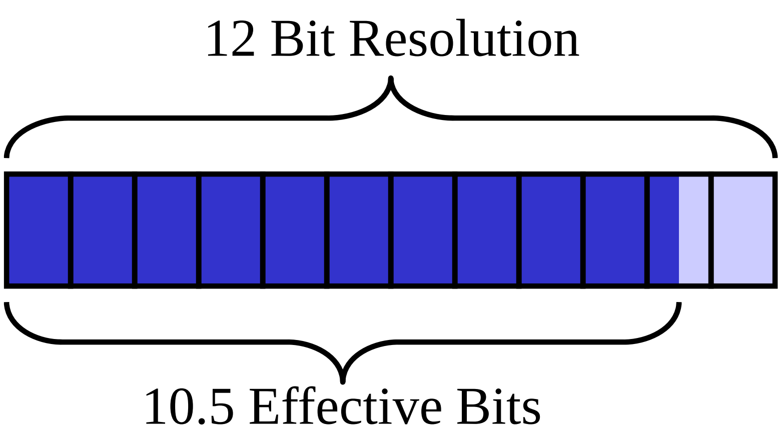
“Samplified” CT Scanner

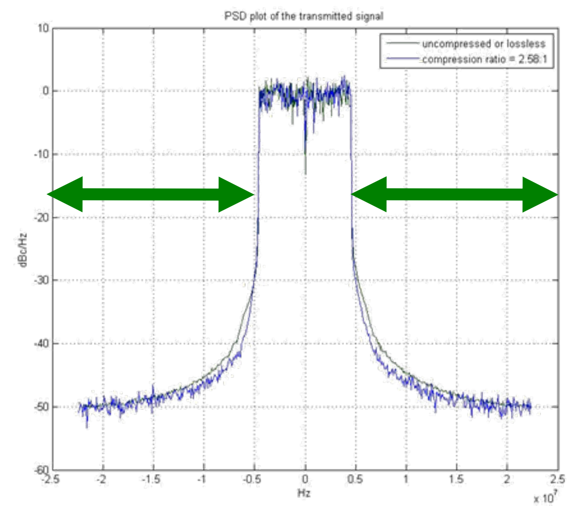


Samplify Prism™ Compression Overview

- Peak to Average Ratio
- Bandlimiting
- Effective Number of Bits







Decompression Alternatives

<u>Two approaches:</u>

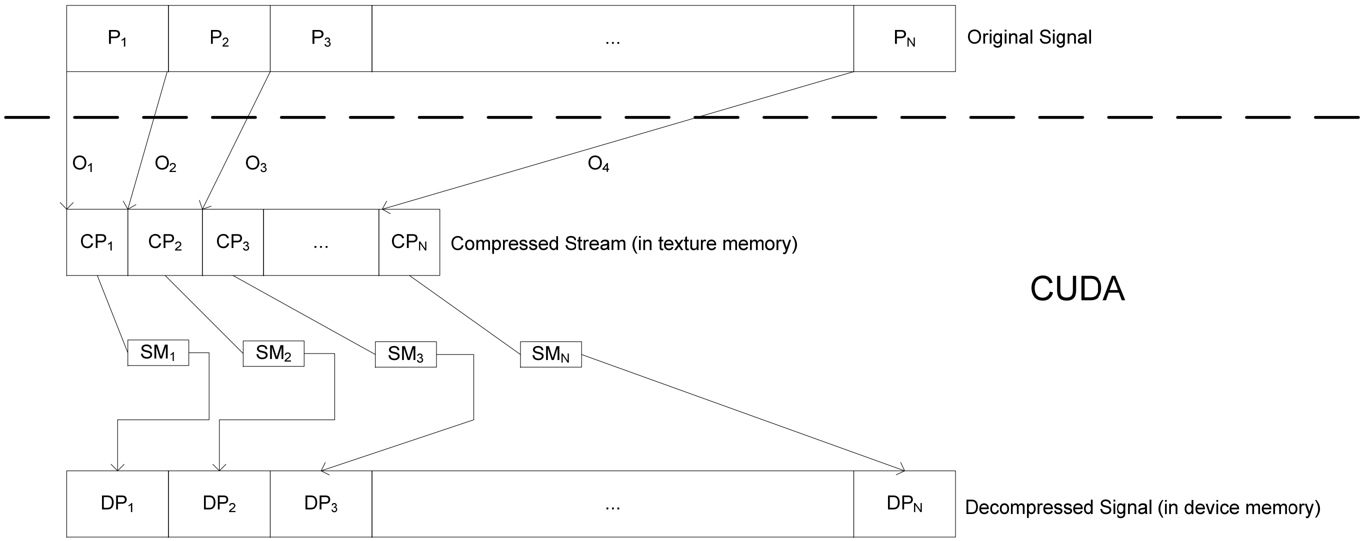
1. Decompress in software on the image recon platform
2. Dedicated HW decompress card feeds CPU/GPU via PCIe

Alternative HW	Pros	Cons
Intel CPU	Std. x86 architecture Std. Wintel/Linux OS	Max of 2 sockets Max of 16 cores Relatively expensive
Intel Larrabee	Std. x86 arch+GPU Open CL	Not yet available Legacy x86 instruction set
FPGA	Best bang-for-the-buck; flexible HW	Hard to program; few experienced developers
Nvidia GPU	**Cheap & available** **Most mature GPGPU** **Lots of cores on die**	**Divergent warps** **All I/O thru host RAM**

Baseline: Decompress on Intel CPU

- With only C compiler optimizations (no asm), Prism decompress runs at ~100 Msamp/sec per core, using Core2 Duo @ 2.3 GHz
- Linear speed-up using parallelized OpenMP code:
  - 2 cores: ~200 Msamp/sec
  - 4 cores: ~400 Msamp/sec
- Still to be exploited: MMX via compiler intrinsics
- Integer instructions dominate decompress time, so SSE instructions don't improve decompress speed

CUDA Decompress: First Effort



- Packetized bitstream
- Compressed packets in texture memory
- CUDA thread responsible for decompressing single packet of compressed data
- CUDA threads decode ***n*** VLC symbols into shared memory buffer, then copy back out to device memory
- ***n*** < length of decompressed packet due to lack of shared memory
- GTX260 PERFORMANCE:
  300 Msamp/sec

LEGEND
P_i = packet
CP_i = compressed packet
O_i = offset or index into compressed packet
SM_i = shared memory decompression cache
DP_i = decompressed packet

```
CUDA kernel pseudo-code0
do {
  parse VLC symbol
  "bit-whacking" ops to decode VLC symbol into decompressed samples
  copy n symbols out to device memory
} while (we haven't decompressed all samples in the packet)
```

- Key problem with vlc decode is that it is *inherently bit-serial* (must decode symbol i-1 before decoding symbol i) which inhibits parallelism
- Samplify decompress has advantage that it is **packetized** so we can parallelize decompress

CUDA Decompress: Second Effort

- Issues encountered during first effort:
  - Small amount of shared memory (16 kB) prohibits storing one entire decompressed packet (~200 16-bit samples) per thread
  - Decomp kernel full of conditional “control” type logic
  - Conditionals lead to divergent warps that reduce overall performance
- IDEA: Instead of having each kernel in a “while” loop:
  - Decode ***n*** VLC symbols, copy back out to shared memory, then decode the next ***n*** VLC symbols, and so on
  - CUDA kernel decodes one symbol at a time
- PERFORMANCE: 350 Msamp/sec (17% faster)
- We expected more than 17% improvement, although this idea seems in line with the “keep your CUDA kernels light-weight” idiom

CUDA Decompress: A Better Way

- Initial CUDA decompress kernel filled with conditionals
  - “while” loop has 16 case statement inside a switch block.
  - “High warp serialize rate” reported by profiler (divergent warps)
- Variable-length decode doesn’t fit single-program, multiple data (SPMD)
- IDEA: Paradigm shift and change in thought process:
  - Very difficult for C programmers!
  - Perform the worst-case thread … all the time (counter-intuitive)
  - Sometimes perform needless computations to remove conditionals
  - Replace traditional imperative code with data-driven LUT approach
  - Very simple example below
  - More complicated examples involve arrays of pointers

- ***BIG WIN: Linear code speed-up with increased # of cores !!***!

```
if (testval & 0xF) < 9
  n = 8;
else
  n = 4;
```

replaced with

```
int NTAB[] = {8,8,8,8,8,8,8,8,8,4,4,4,4,4,4,4};
n = NTAB[ testval&0xF ];
```


…simply the bits that matter®

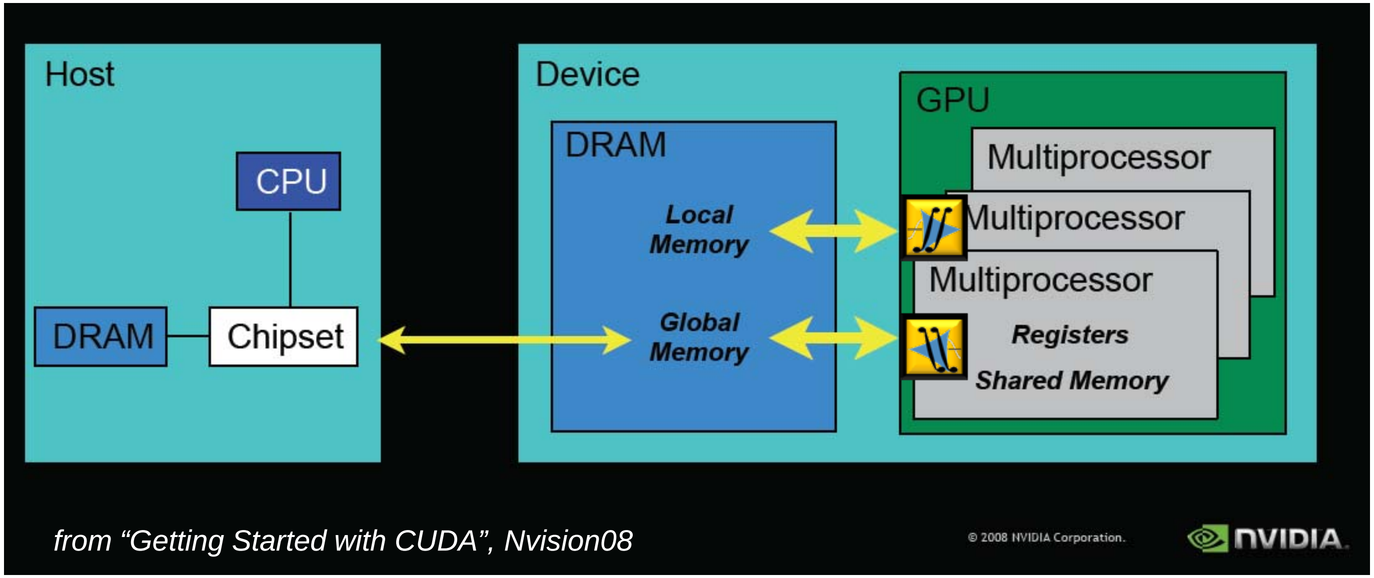
What's Next: More to Do…

- Decompress throughput profiled on GTX 260 (CUDA toolkit 2.2)
  - First attempt: 300 Msamp/sec (less than linear speed-up)
  - Latest attempt: 450 MS/s (close to linear speed-up)
  - Thruput *without* data xfer time – just kernel execution time
- Additional kernel tuning:
  - Fix apparent nvcc bug involving shared memory arrays of pointers, where "pointees" all point to shared memory scalar entities.
  - All LUTs reside in const memory → move to texture memory to reduce register pressure
  - No "caching" of decompressed samples into shared memory → move compressed stream into texture memory
  - Low-level optimizations: write 64 or 128 decompressed bits at a time when copying back out to device memory

Lessons Learned about CUDA Decompress

- Variable-length code (VLC) decode and decompress:
  - difficult to map efficiently to CUDA due to bit-serial nature.
- Packetized bit-streams ***can*** be parallelized
  - Have to work around limited amount of shared memory
- Conditionals cause divergent warps
- SPMD (single program multiple data) model dictates that each thread in a warp must executes the same instruction at the same time.
- Need to think “outside the box” to remove conditionals
- Key insight: data-driven LUT methodology was key to overcoming conditional code
- “high warp serialization rate” (profiler) of serialized CUDA threads is significantly reduced by the data-driven LUT methodology

Compress & Decompress for CUDA Apps



GPU precedent:

H.264, VC-1, MPEG2, and WMV9 decode acceleration in GTX200 series

Computer vision
Augmented reality
Supercomputing
Computational finance
Computational fluid dynamics
Advanced numeric computing
GPU computing in education
AstroGPU
MultiGPU programming
GPU Clusters

Digital content creation
Video and image processing
Advanced visualization
Computer aided engineering
Molecular dynamics
Reality simulation
3D Stereo
Scientific visualization
Visual simulation

C/C++ with CUDA Extensions
DirectCompute
OpenCL
OpenGL
DirectX 11
Medical imaging
Life sciences
Energy exploration
Film and broadcast
Automotive

Conclusions

- Medical imaging bottlenecks are increasing significantly
- Compression performed in FPGAs near the sensors
- Decompression in GPUs, along with image reconstruction
- Prism reduces I/O bottlenecks between sensors and GPUs
- Replacing conditionals with table look-ups results in a near-linear GPU speed-up of variable-length decoding

- Low-complexity hardware compress & decompress core could significantly reduce CPU-GPU I/O bottlenecks for many CUDA-enabled numerical applications

Special thanks to Shehrzad Qureshi for his significant contributions to these results.