GAIA

GPU Architectures for Intelligence Applications

Application Kernels on Graphics Processing Units

High Performance Embedded Computing
September 20 -22, 2005

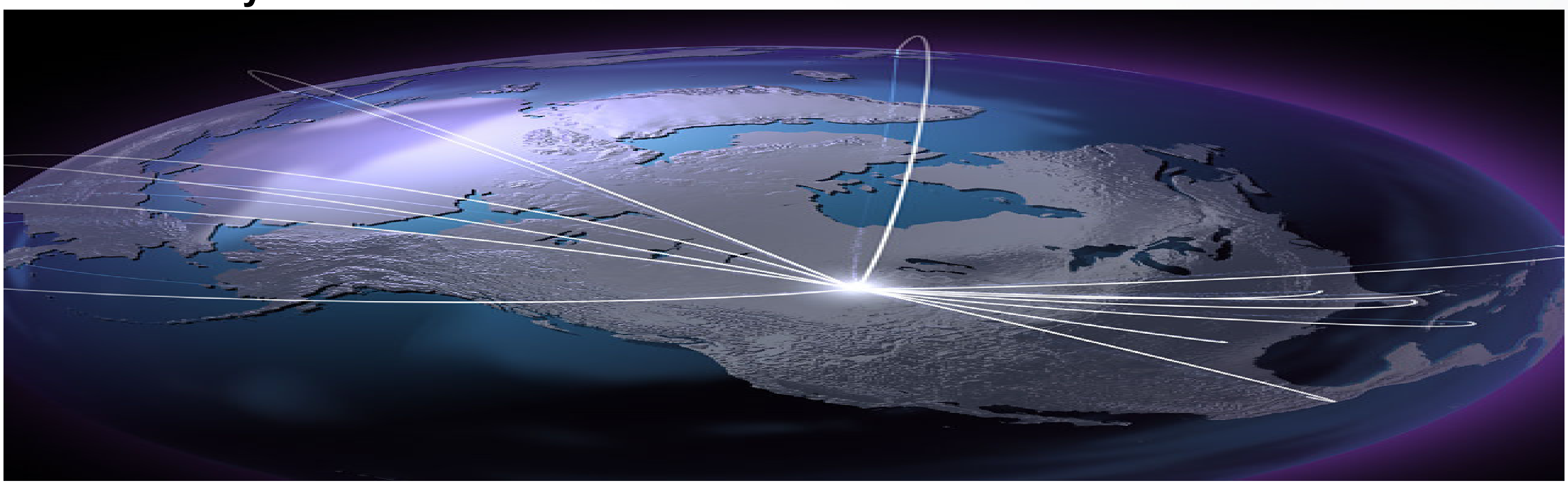
David Bremer, John Johnson, Holger Jones, Yang Liu,
Daniel May, Jeremy Meredith, Sheila Vaidya

Lawrence Livermore National Laboratory

In collaboration with Stanford Computer Graphics Laboratory

GAIA Highlights

- *200X speedup* for convolution
- *150X speedup* for georegistration
- *30X speedup* for hyperspectral imaging application
- *25X speedup* for Smith-Waterman string matching
- *24X speedup* for support vector machines
- *17X speedup* for Hidden Markov Models
- *17X speedup* for discrete cosine transform
- *16X* speedup for Feature Extraction in ISIP speech recognition
- *10X* speedup for correlation
- *4X speedup* for singular value decomposition
- "almost" double precision in software with only *1.4X* slowdown
- Distribution of production FFT library, libgpufft, in collaboration with Stanford University

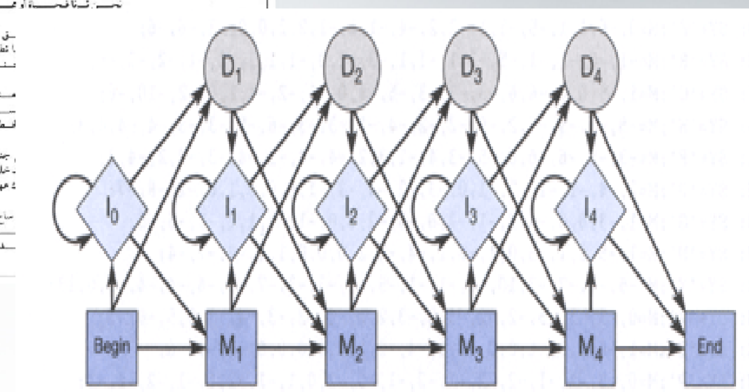


Exploring GPUs / COTS technologies for Knowledge Discovery applications

Application Focus

- Image Processing
  - Hyper & Multispectral
- Speech Recognition
- Text Analysis
- Statistical Learning
- Graph


السلام الوطني الجزائري
قسماً



Identify algorithm characteristics necessary for GPU exploitation

Develop data infrastructure, cache coherent streaming, download / readback, software abstractions

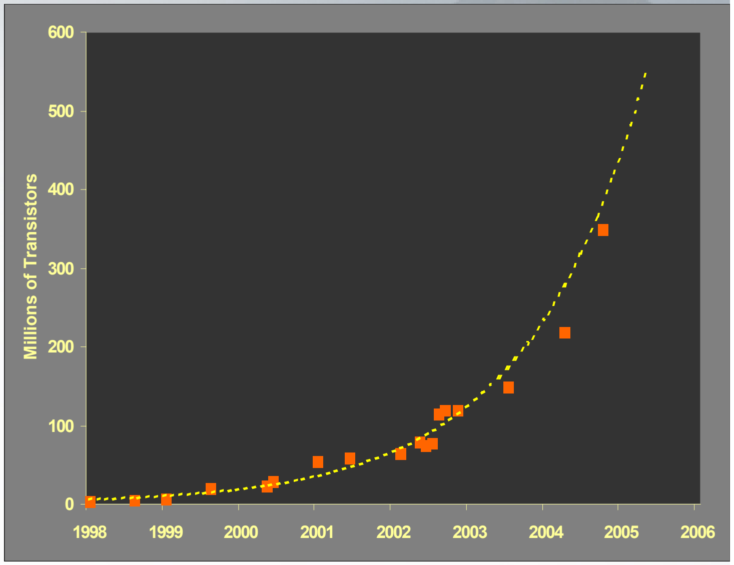
Delineate boundary conditions

Evaluate GPU vs. CPU & hybrid solutions

Investigate algorithm / application parameter space

Motivation

- Trends in the graphics marketplace
  - Move to programmable hardware
  - Performance increasing faster than for CPUs
  - Effects of mass markets
- Not expected to end anytime soon…
  - 2005: 160 GFLOPS
  - 2006: 400 GFLOPS
  - 2007: 1 TFLOP
  - Worlds fastest supercomputer: BG/L is 360TF



Upgrading from nVidia 6800 to nVidia 7800 GPU hardware dramatically improved performance increase over CPU

Speedup over current generation 2.4GHz P4		
	nv6800	**nv7800**
Convolution	96X	**194X**
SVM	18X	**24X**
ISIP Speech Recognition Feature Extraction	7X	**16X**

GAIA

GPU Architectures for Intelligence Applications

Application Highlights

Hyper-Spectral Imaging (HSI)

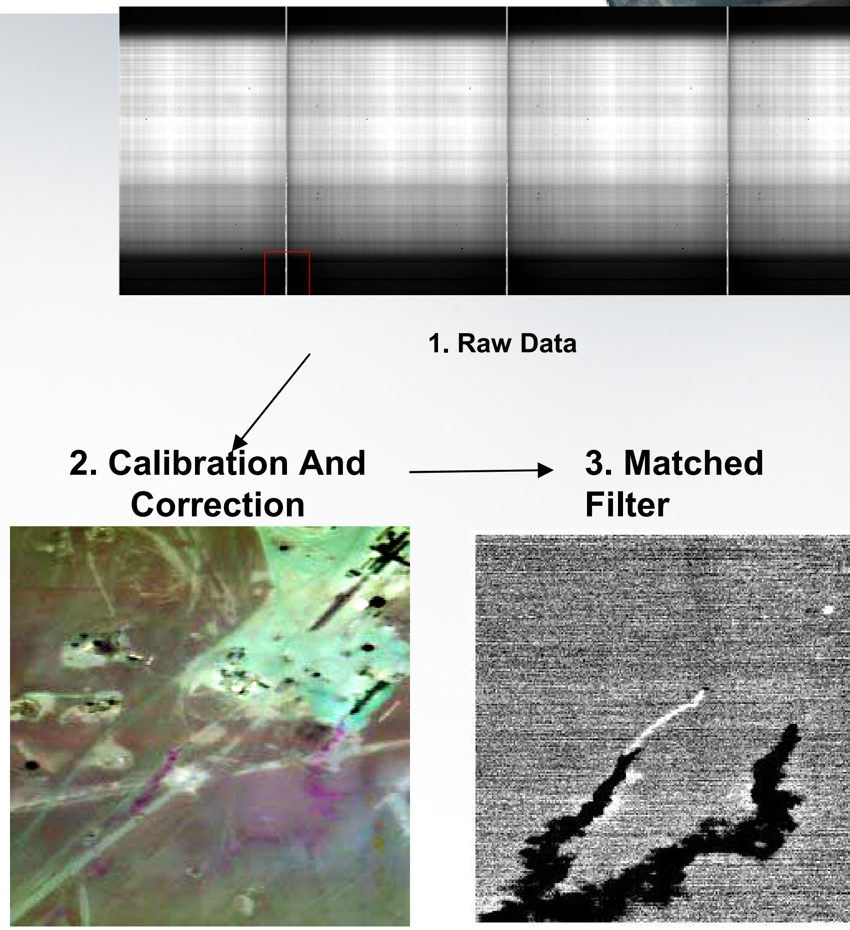
Motivation

- Conventional hyperspectral imaging (HSI) instruments only collect and store raw data.
- Processing and exploitation of the data is performed later using conventional computers.
- From collection to analysis takes hours or even days.

How can we achieve real-time capability?

Move the processing onboard

- Faster answers: *minutes rather than hours*
- Provide computational capability that can keep up with the HSI instrument in real time
- Results can be used in a single pass
  - e.g. for positioning of a second sensor
- Data reduction: *up to 0.012%*
  - *Dramatically lowers storage requirements*
  - *Enables low bandwidth real-time transmission of data*



Example RGB Composite Image

Matched Filter

HSI/GPU results

- Preliminary results demonstrate 15-30X speedup on current generation mobile GPUs

- Small form factor and relatively low power consumption allow mobile GPUs to be used as high performance embedded processors for real-time processing

- Potential for other embedded applications

nVidia NV40

- 256MB+ RAM,
- 128 32bit IEEE FP units @ 400Mhz
- 220M transistors, 16 pipelines

nVidia go6800

- Low power, small size laptop GPU -- same design as desktop model
- Potential for real time HSI computation

GAIA

GPU Architectures for Intelligence Applications

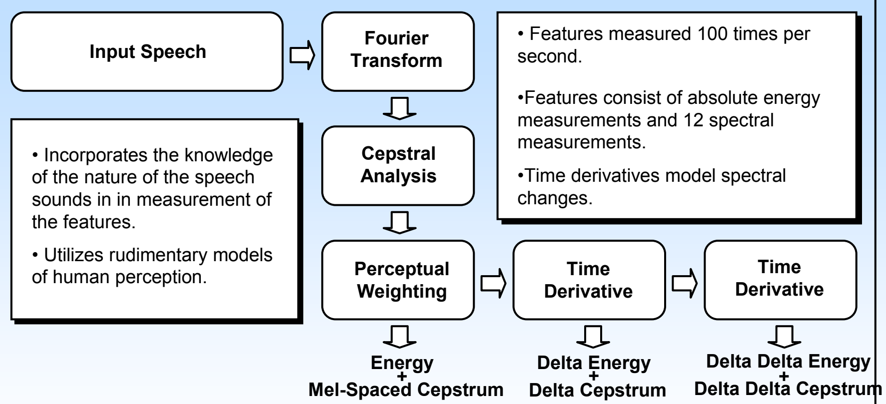
Application Highlights
Speech Recognition (ISIP)

In collaboration with Mississippi State University

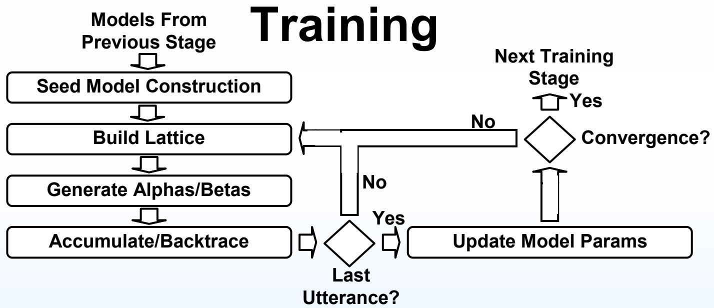
ISIP Overview

Feature Extraction

All signal frames are independent and their features can be extracted in parallel. These conditions are ideal for implementation on a GPU.



Training

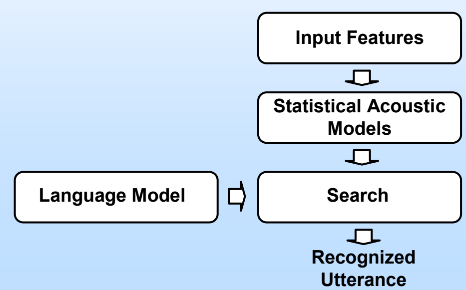


Training can be batched into multiple jobs, and the GPU can process these in parallel, generating accumulators for each batch which can be combined after all batches are complete.

Baum-Welch is used to re-estimate the acoustic models.

Decoding

The decoder uses a Viterbi beam search to hypothesize the transcription of an utterance. The GPU can assist state probability calculations and the state-level search.



A GPU Accelerated Speech Recognition System

Timing results for feature extraction from a subset of the TIDigits corpus.

Algorithm	**CPU (Opteron) Speed (sec)**	**GPU NV7800 Speed (sec)**
Download Texture Memory	N/A	0.40
Absolute Energy	0.13	0.04
Pre-emphasis/Hamming	0.15	0.04
FFT Magnitude	8.40	0.36
Mel-Scale/Cepstrum	6.00	0.44
DCT/Liftering	6.30	*
Derivatives	0.07	N/A
Entire Front-end	**21.15**	**1.28**
Speedup GPU/CPU		**16.5x**

GAIA

GPU Architectures for Intelligence Applications

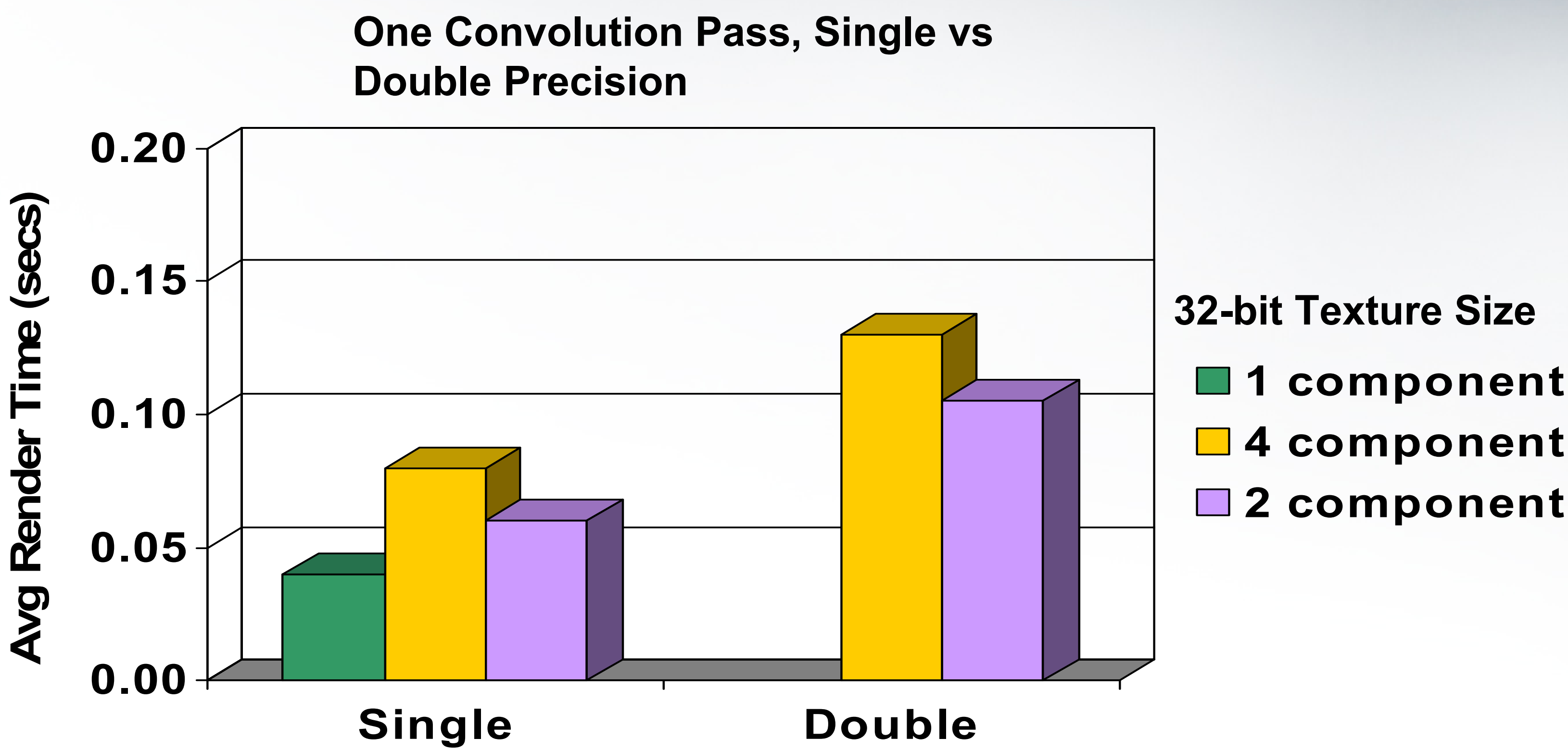
Other Highlights

Double Precision

- Port of David Bailey's *single-double* Fortran library* to NVidia's Cg language
- Can emulate double precision
- Use two single-precision floats
- High order float is estimate to the *double*
- Low order float is error of that estimate
- Resulting precision is almost *double*
- The exponent remains at *single* range

*http://crd.lbl.gov/~dhbailey/mpdist

Double Precision Results



1.3X slowdown for double precision

Smith-Waterman

- Key algorithm in BLAST
- Used to find approximate matches in long character sequences
  - e.g. proteins, DNA in bioinformatics
- May have direct fuzzy text search applications
- Implementation uses dynamic programming – similar to:
  - text processing algorithms like Levenshtein (edit) distance
  - graph algorithms like shortest paths
  - speech processing algorithms like Viterbi for HMMs

Smith-Waterman

- Local optimal sequence alignment algorithm
- Tolerate mismatches & gaps for optimal alignment
- Parameters
  - Matching cost matrix. (e.g. BLOSUM62)
  - Gap penalty model
    - Linear $g(n) = \alpha n$
    - Single-Affine $g(n) = \alpha + \beta n$
    - Double-Affine $g(n) = \alpha + [(n < t) \, ? \, \beta : \gamma]n$
- Dynamic programming
  - Find best matching score
  - Trace back computation to build alignment
  - Optimal subsequence, recursion

Smith-Waterman

```
SCORE: 1352  EVALUE: 0
Alignment Length: 467   IDENTITIES: 58.24% (272/467) SIMILARITIES: 63.81% (298/467)
GAPS: 2
QUERY START: 1  QUERY END: 367
HIT START: 1  HIT END: 464
    1 MVSILTVLLCLRLSLGQKAQALAGTLPKPSLWAEPGSVITWESPMTLWCQGTLDTQGYYLTKEGNPMTWY    70
      |  |||||.|| ||||  . . |  |||| |||||| |||| .|.|.||||||. ||| | |||| |. .
    1 MTPILTVLICLGLSLGPRTRVQAENLPKPILWAEPGPVITWHNPVTIWCQGTLEAQGYRLDKEGNSMSRH    70

   71 QQSPPEPRNKTNFFIPSMREHHAGRYHCHYLSPAGWSERSEPLELVVT----------------------   118
           |  ||    ||||   |||||||.| ||||||| |.|||||||
   71 ILKTLESENKVKLSIPSMMWEHAGRYHCYYQSPAGWSEPSDPLELVVTAYSRPTLSALPSPVVTSGVNVT   140

  118 ----------------------------------------------------------------------   118

  141 LRCASRLGLGRFTLIEEGDHRLSWTLNSHQHNHGKFQALFPMGPLTFSNRGTFRCYGYENNTPYVWSEPS   210

  118 --------GVSRKPSLLTLQGPVVAPGENLTLQCGSDVGYDKFTLYKEGGHDLVQGSGRQPQAGLSQANF   180
              |||||||||||||||| ||||||||||||||| ..||||||   | |  |||||||||||||
  211 DPLQLLVSGVSRKPSLLTLQGPVVTPGENLTLQCGSDVGYIRYTLYKEGADGLPQRPGRQPQAGLSQANF   280

  181 TLGPVRVSHGGQYRCYGAHNLSSEWSAPSDPLSILIAGQIRGRPSLSVQPGPTVASGENVTLLCQSREQL   250
      || ||  |.|||||||||||.||||||||||| |||||||  |||||||||||| ||| ||||||| . .
  281 TLSPVSRSYGGQYRCYGAHNVSSEWSAPSDPLDILIAGQISDRPSLSVQPGPTVTSGEKVTLLCQSWDPM   350

  251 DTFLLTKEGAAHHPLRLRSEHQAQQHQAEFPMSPVTSAHAGTYRCYSSRRFFPYLLSHPSDPLELVVSGA   320
       ||||||||||| |||||| . | ..|||||||||||||||||||| ||   ||||||||.|||||||||
  351 FTFLLTKEGAAHPPLRLRSMYGAHKYQAEFPMSPVTSAHAGTYRCYGSRSSNPYLLSHPSEPLELVVSGA   420

  321 AETLSPSQNKTDSKTASPASHPQDYTVENLIRVAVAGLVLVVLGILL   367
       |||.|.| |.||||   | | ||||||||||. ||||||. |||||
  421 TETLNPAQKKSDSKT---APHLQDYTVENLIRMGVAGLVLLFLGILL   464
```

Smith-Waterman

- Preliminary Results
  - Data:
    - Query Sequence
      4096 characters
    - Database:
      337,603 sequences
      110,088,276 total characters
  - Performance:
    - GPU (GeForce 7800) 41m 16s (2476s)
    - CPU (AMD Opteron 246 2GHz) 15h 18m 17s (55097s)
  - GPU ~22.3 times faster than CPU implementation

GAIA

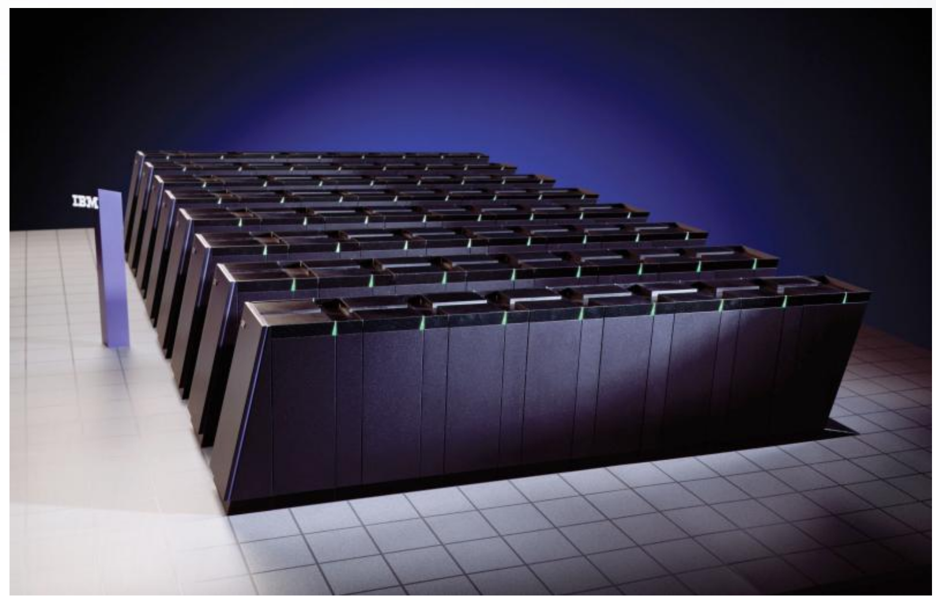
GPU Architectures for Intelligence Applications

Future Directions

Exploration of COTS CPU & GPU clusters and hybrids

- Text
  - Indexing
  - Clustering
  - Classification
- Speech
  - Training
  - Decoding
- Graph
  - Search
  - Connected components

LLNL’s 11.2-teraflops
Multiprogrammatic Capability Resource (MCR) supercomputer



LLNL’s 360-teraflops BlueGene/L