GPUs: Economic Attraction and Performance Challenges

Dan Reed
Dan_Reed@unc.edu

University of North Carolina at Chapel Hill
Duke University
North Carolina State University
Presentation Outline

- **Historical perspectives**
  - technology evolution
  - lessons from the past
  - HPC application attributes

- **PlayStation2 experiences**
  - architectural implications
  - application porting

- **Economics and government policy**
  - HPC studies and lessons
  - current status and futures

- **Thanks to**
  - Craig Steffen, Pedro DeRose, Celso Mendes
  - Rob Pennington, Perry Melange
  - NSF and DOE
The Siren Call: Peak Performance

The Sirens inhabited an island surrounded by dangerous rocks. They sang so enchantingly that all who heard were drawn near and shipwrecked. Jason and the Argonauts were saved from them by the music of Orpheus, whose songs were lovelier. Odysseus escaped them by having himself tied securely to a mast and by stopping the ears of his men.
The Siren Call: Peak Performance

- **1890-1945**
  - mechanical, relay
  - 7 year doubling
- **1945-1985**
  - tube, transistor, ...
  - 2.3 year doubling
- **1985-2004**
  - microprocessor, GPU, ...
  - roughly 1 year doubling
- **Every year**
  - equal to all previous history!
- **Storage, networks and graphics**
  - even faster, with qualifiers!
- **Delivered performance and software development**
  - dependent on algorithms and architecture match
  - a much more nuanced story …

Data source: Jim Gray

Renaissance Computing Institute
The Siren Call …

• We’ve seen *parts* of this movie before
  – systolic arrays, attached processors, …

• **Success requires optimizing for efficiency**
  – data movement, computation and software costs

• **Efficient exploitation, in two senses**
  – achieved application performance
    • holistic assessment, not just application kernels
  – high human productivity
    • extant software base, available tools
Floating Point Systems AP120B (1975)

- 6 MHz (167 ns), 38-bit floating point
- Multiple operations per 64-bit instruction
  - data movement and arithmetic
- Multiple independent function units
  - floating addition (2 stage) and multiply (3 stage)
  - peak 12 MFLOPS
- Parallel memories
  - two 32-word data pad (DX, DY)
    - 2 per cycle
  - 2560 word fixed table memory
    - 1 per cycle, 2 cycle delay
  - 64 KW data memory
    - ½ per cycle, 3 cycle delay
  - 512 word instruction memory
- Address indexing and counting (SPAD & ALU)

Source: David Culler
AP120B: Portable Is A Fluid Term

Introducing the AP-120B, the unique peripheral floating-point array transform processor.

You’ll look at the specifications and see the computational speed and accuracy of a mega-dollar mainframe.

Then you’ll look at the cost, size, weight and ease of programmability, and realize what a breakthrough the AP-120B really is.

The unique culmination of years of floating-point and array processing expertise, the AP-120B offers a completely new source of convenient dedicated computational power.

Source: David Culler
• 64 bit “wide word” instruction issue
• Libraries for AP120B use

Source: David Culler

Renaissance Computing Institute
Lessons Learned: Not Like the Others

• Which one doesn’t fit?
  – cheap, high capacity storage
  – high bandwidth networks
  – low cost, high productivity software development 😞
  – inexpensive processors

One of these things is not like the others,
One of these things just doesn't belong,
Can you tell which thing is not like the others
By the time I finish my song?
The Six Modern Computing Eras

- Big Iron (post WW II)
- Mainframe (‘60s/’70s)
- Workstations (‘70s/’80s)
- PCs (‘80s/’90s)
- Internet (‘90s)
- Implicit computing
  - embedded intelligence in everyday objects
    - cell phones, thermostats, watches, anti-lock brakes
    - microwave ovens, dishwashers, radios, pacemakers
  - broadband wireless networking
- What’s changed and what does it mean?
  - processors/person → infinity
    - $O(100M)$ PCs and $O(8B)$ embedded processors/year
  - software developers/users → zero
Scientific Computing Building Blocks

- **Processors**
  - x86, x86-64, Opteron, Itanium, PowerPC
  - GPUs

- **Memory systems**
  - the jellybean market
  - memory bandwidth 😞

- **Storage devices**
  - vibrant storage market
    - bandwidth remains an issue

- **Interconnects 😞**
  - Ethernet (10/100, GbE, 10GbE)
  - Infiniband
  - Myrinet, Quadrics, SCI, …
Cables, NICs and Switches

- **NCSA Platinum**
  - 8.3 km total (512 2-way nodes)
- **NCSA TeraGrid**
  - 32.1 km total
    - 8.3 km (phase one)
    - 23.8 km (phase two)
- **PCI-Express is not enough**
  - Infiniband 4x helps, but …
    - deeper integration is needed
Each strikes a different balance
- computation/communication coupling

Implications for execution efficiency

Applications for diverse needs
- computing is only one part of the story!

As Keith Cooper said
- large-scale science applications achieve 5-15% of peak
Large Scale Scientific Applications

• Developed over at least a decade
  – incremental changes
    • solvers, science modules, tools
  – evolving development teams
    • lossy knowledge transfer

• Programmed to LCD
  – lowest common denominator (LCD)
    • tools and “fads” come and go
    • MPI – the assembly language of parallel programming
  – multiple execution platforms
    • interoperable capabilities and software

• Increasingly multidisciplinary
  – science and module interaction
    • local and global component optimization
  – diverse needs and demands
    • large memory, high I/O,
    • real-time sensor streams, compute intensive, …
Biochemical Physical Questions

- Genomics
- Biochemical network modeling
- Cellular modeling
  - Intracellular trafficking and regulation
  - Motors to cilia
- Hydrodynamics
  - Cilia/cilia coupling
  - Cilia PCL/mucus coupling
  - PCL/mucus mixing
- Rheology
  - Molecules to bulk properties

Source: Ric Boucher, UNC
Software Complexity and Growth

- Large Scale Data Management
- Worldwide Collaboration (Grids)
- Feature Extraction and Simulation
- Detector and Computing Hardware

1971

2001

~7 Million Lines of Code (BaBar)

~100k LOC

~500 people (BaBar)

~10 people

Source: Richard Mount, SLAC

Renaissance Computing Institute
Observations on Software

• Business
  – capital is cheap
  – labor is expensive
  – costs are usually explicit
    • and had better be lower than revenues!

• Academia and government
  – capital is (seemingly) expensive
  – labor is (seemingly) cheap
    • student, faculty and staff time
  – costs are usually implicit
    • and often skew realistic assessment

• This is a critical issue for software
  – development, support and sustenance
  – total cost of ownership
    • NRE plus unit costs
For Scientists: It's All About the Science
Three Scientific Computing Sweet Spots

• Domain-specific desktop toolkits
  – invisible desktop acceleration
  – high-level scripting languages and tools
    • MATLAB™, Mathematica™, …

• Laboratory systems, typically clusters
  – 64-128 node sweet spot
  – some user software development
  – community and ISV software toolkits
    • BLAST, NWChem, ANSYS, LS-DYNA, Gaussian …

• Large-scale systems
  – size bounded above by $$$ and reliability
  – mostly “roll your own” software
  – large scale, often multidisciplinary codes
Presentation Outline

• Historical perspectives
  – technology evolution
  – lessons from the past
  – HPC application attributes

• PlayStation2 experiences
  – architectural implications
  – application porting

• Economics and government policy
  – HPC studies and lessons
  – current status and futures
Computing On Toys

- **Sony PlayStation2**
  - 6.2 GF peak (fast then, slow now)
  - 70M polygons/second
  - 10.5M transistors
  - superscalar RISC core
  - plus vector units, each:
    - 19 mul-adds & 1 divide
    - each 7 cycles

- **NCSA/Illinois CS project**
  - started three years ago
PlayStation2 Linux Kit

• **Why Linux?**
  – lots of scientific applications on Linux clusters
    • ready familiarity and access
  – educational science opportunity

• **Kit components**
  – Linux kit release 1.0 software
  – monitor cable adaptor
  – internal 40 GB disk
  – 10/100 Ethernet network adaptor
    • performance limiting effect
  – USB keyboard and USB mouse

• **Vector unit compiler not included**
  – generally, must be a Sony licensed game developer
  – we worked directly with Sony
NCSA/CS “EBay” PlayStation2 Cluster

- **Linux kit components**
  - Linux kit release 1.0 software
  - monitor cable adaptor
  - internal 40 GB disk
  - 10/100 Ethernet network adaptor
    - performance limiting effect
  - USB keyboard and USB mouse

- **PlayStation2 Linux kits**
  - first released in Asia
  - then in Europe
  - finally released in the U.S.
  - now discontinued

- **We got two systems**
  - acquired via EBay
  - shipped from Japan

- **Reading the manual …**
  - was “interesting”
NCSA PlayStation2 Cluster

- 70 unit NCSA cluster
  - 65 compute, 4 login and 1 development
  - 24-inch rack; five shelves at 13 units/shelf
- Linux software and vector unit use
  - over 0.5 TF peak but …
PlayStation2 Architecture

- **MIPS Core**
  - standard 32-bit processor
- **USB, Firewire and PCMCIA connectors**
  - PCMCIA for Ethernet
- **Small vector unit memories**
  - 4KB and 16KB for V0 and V1
• Each unit
  – 19 multiply-adds and 1 divide each 7 cycles
    • upper and lower instruction units
  – macro and micro modes
    • macro (MIPS co-processor) for V0
    • micro (downloadable code) for V0 and V1
  – Vector Interface Unit (VIF)
    • main memory transfers of data and code (micromode)
    • DMA activated

PS2 Vector Unit Architecture

Renaissance Computing Institute
Architectural Challenges

• Small vector unit memories
  – MIPS core must constantly feed data to the VUs
  – streaming/double buffering is critical to performance
  • overlapped data transfers and vector computations
  – need high compute/data transfer ratio

• Matrix multiply library
  – configurable sub-block data transfers
    • sub-block size chosen to maximize performance
      – ranges from 4x4 to 28x28 (maximum VU memory size)
  – source chain DMA transfers for non-contiguous data
    • chain of data block size and memory pointers
    • avoids data copies for scatter/gather
  – MIPS scratch pad region (SPR) used for assembly
Matrix Multiplication

- Consider $\hat{A} = \hat{B}x\hat{C}$
  - $B$ and $C$ are conformable and partitioned
  - blocks are of “optimal” size
  - row/column blocks transferred and accumulated

\[
\begin{bmatrix}
A_{\alpha,\alpha} & A_{\alpha,\beta} & A_{\alpha,\chi} & A_{\alpha,\delta} \\
A_{\beta,\alpha} & A_{\beta,\beta} & A_{\beta,\chi} & A_{\beta,\delta} \\
A_{\chi,\alpha} & A_{\chi,\beta} & A_{\chi,\chi} & A_{\chi,\delta} \\
A_{\delta,\alpha} & A_{\delta,\beta} & A_{\delta,\chi} & A_{\delta,\delta}
\end{bmatrix}
\begin{bmatrix}
B_{\alpha,\alpha} & B_{\alpha,\beta} & B_{\alpha,\chi} & B_{\alpha,\delta} \\
B_{\beta,\alpha} & B_{\beta,\beta} & B_{\beta,\chi} & B_{\beta,\delta} \\
B_{\chi,\alpha} & B_{\chi,\beta} & B_{\chi,\chi} & B_{\chi,\delta} \\
B_{\delta,\alpha} & B_{\delta,\beta} & B_{\delta,\chi} & B_{\delta,\delta}
\end{bmatrix}
\]

\[
C_{\alpha,\alpha} = \sum_{\alpha} A_{\alpha,j}B_{j,\alpha}
\]

- Achieved performance is $\sim1$ GF (PS2 V1 only)
  - $\sim40$ percent of peak
- Generalized to SGEMM $\hat{C} = \alpha\hat{A}\hat{B} + \beta\hat{C}$

Renaissance Computing Institute
Lattice QCD

• No “Grand Unified Theory”
  – quantum theory (electroweak and strong forces)
  – gravity integration and rationale for mass
    • search for the Higgs boson
  – dark matter and dark energy

• Quantum Chromodynamics (QCD)
  – why protons and neutrons live happily together in nuclei
  – strong interaction between quarks, mediated by gluons
    • expressed via Dirac operators of varying complexity

• Lattice QCD
  – numerical simulation of QCD via discretized space/time
    • quarks at lattice points, with gluons mediating along edges
      – SU(3) matrix operations dominate the calculation
    – yields complex, sparse matrices
      • solution via conjugate gradient techniques

• MILC (MIMD Lattice Computation)
  – one lattice QCD implementation
  – see www.physics.indiana.edu/~sg/milc.html

\[ D\psi(x) = \frac{1}{2a} \sum_\mu \left[ U(x)\psi(x + \hat{\mu}) - U^\dagger(x - \hat{\mu})\psi(x - \hat{\mu}) \right] \]
MILC Code Structure

**Conjugate gradient core**
- start gathers of data from positive direction
- multiply quark vectors by matrix operators
  - gluon field in negative directions
- start gathers from negative directions
- await gathers from positive directions
- multiply quark vectors by matrix operators
  - gluon field in positive directions
- await gathers from negative directions
- accumulate results
- check convergence
- repeat until converged

**Critical features**
- scatter/gather
- matrix-vector operations
- compute/communicate intensive ratio
  - varies with local lattice size
Presentation Outline

• Historical perspectives
  – technology evolution
  – lessons from the past
  – HPC application attributes

• PlayStation2 experiences
  – architectural implications
  – application porting

• Economics and government policy
  – HPC studies and lessons
  – current status and futures
Intelligent Software: An Analogy

• **50 MPH is a legal stricture with no ambiguity**
  – 51 MPH is a violation and you could be cited and fined
    • rarely are violators ticketed for such small violations
  – context determines actual behavior
    • city rush hour traffic rarely obeys speed limits
    • hazardous conditions change the effective speed limit

• **What really happens**
  – police use contextual discretion
    • “small” violations for “reasonable intervals” are tolerated
  – obeying the spirit of the law is usually the correct thing
    • perturbations about the limits are expected and accepted
  – if something happens, you want justice, not legalisms

• **Intelligent, adaptive software is similar**
  – application needs and available resources should determine behavior
Choose At Most Two

• High performance
  – exploitation of system specific features
    • cache footprint, latency/bandwidth ratios, …
  – militates against portable code

• Portability
  – targeting the lowest common denominator
    • standard hardware and software attributes
  – militates against high performance code

• Low development cost
  – cost shifting to hide human investment
    • people are the really expensive part
  – specialization to problem solution
  – militates against portable, high-performance code
How To Choose Wisely

• Performance
  – runtime adaptation
  – dynamic code generation

• Portability
  – automatic specialization

• Development cost
  – quantitative cost-benefit ratios

• The moral of the story
  – capture insights/experience
    • do what humans do well
  – automate the dull stuff
    • ATLAS, FFTW, …
MPI: It Hurts So Good

• **Message Passing Interface (MPI)**

• **Observations**
  – “assembly language” of parallel computing
  – lowest common denominator
    • portable across architectures and systems
  – upfront effort repaid by
    • system portability
    • explicit locality management
  – remember what Churchill said about democracy
    • it applies to MPI as well

• **Costs and implications**
  – human productivity
    • low-level programming model
  – software innovation
    • limited development of alternatives
HPF: I Feel Your Pain

• High-Performance Fortran (HPF)
  – data parallel model for distributed memory

• Lessons
  – irregular data structures
    • better support needed
  – data distributions
    • best not part of the language
  – compilation and tuning
    • major research challenges
    • inverse mappings for tuning

• Observations
  – HPF locality model is semi-implicit
  – we expected too much too soon, but long term matters
    • see Earth System Simulator
Some Other Issues

- Double precision floating point
  - critical to most scientific applications
- Standard software development environments
  - domain-specific packages/tools
    - MATLAB™ and Mathematica™
  - data parallel languages/tools
    - FORTRAN90, …
- ISV code porting/support
  - independent software vendors
    - ANSYS, Gaussian, CHARMM
  - LS-DYNA, NASTRAN, …
- Industrial HPC
  - desirable but “hard to use”
  - recall the cost of people
“Due to its impact on a wide range of federal agency missions ranging from national security and defense to basic science, high end computing—or supercomputing—capability is becoming increasingly critical. Through the course of 2003, agencies involved in developing or using high end computing will be engaged in planning activities to guide future investments in this area, coordinated through the NSTC. The activities will include the development of interagency R&D roadmap for high-end computing core technologies, a federal high-end computing capacity and accessibility improvement plan, and a discussion of issues (along with recommendations where applicable) relating to federal procurement of high-end computing systems. The knowledge gained for this process will be used to guide future investments in this area. Research and software to support high end computing will provide a foundation for future federal R&D by improving the effectiveness of core technologies on which next-generation high-end computing systems will rely.”
Many Workshops and Reports

- Computation as a Tool for Discovery in Physics, September 2001
- Blueprint for Future Science Middleware and Grid Research
  - www.nsf-middleware.org/MAGIC/default.htm
- NSF Cyberinfrastructure Report, January 2003
  - www.cise.nsf.gov/sci/reports/toc.cfm
- DOE Science Network Meeting, June 2003
  - gate.hep.anl.gov/may/ScienceNetworkingWorkshop/
- DOE Science Computing Conference, June 2003
  - www.doe-sci-comp.info
- DOE Science Case for Large Scale Simulation, June 2003
  - www.pnl.gov/scales/
- DOE ASCR Strategic Planning Workshop, July 2003
  - www.fp-mcs.anl.gov/ascr-july03spw
- Roadmap for the Revitalization of High End Computing, June 2003
  - www.hpcc.gov/hecrf-outreach
- House Science Committee Hearing, “Supercomputing: Is the U.S. on the Right Path?”
  - www.house.gov/science/hearings/full03/index.htm
- PITAC Computational Science, 2004-2005
  - stay tuned
HECRTF Interagency Perspectives*

- HEC is a declining fraction of the overall market
  - future systems may be less suitable to HEC needs
  - commercial market is diverging from science/government needs
- Future success will require coordinated effort
  - R&D and engineering of new architectures and systems
  - software research and development
    - systems and middleware
    - programming environments and applications
  - new domain science and algorithms
  - procurement of new COTS and custom systems
    - sustainable strategies
- Targeted funding of HEC systems may be required
  - including development of new systems

*My assessment; my apologies for any misrepresentations
HECRTF Reports

- See www.hpcc.gov/hecrtf-outreach
- President’s Information Technology Advisory Committee (PITAC)
  - computational science subcommittee
The Cambrian Explosion

- Most phyla appear
  - sponges, archaeocyathids, brachiopods
  - trilobites, primitive mollusks, echinoderms
- Indeed, most appeared quickly!
  - Tommotian and Atdbanian
  - as little as five million years
- Lessons for computing
  - it doesn’t take long when conditions are right
    - raw materials and environment
  - leave fossil records if you want to be remembered!