Accelerating the Data Center

Steven A. Guccione, Ph.D.

June 2, 2015

Bank of America

Bank of America Merrill Lynch U.S. Trust Bank of America Merrill Lynch

Introduction

Financial firms have long been users of large compute resources. While some of the compute infrastructure in banking and finance is used for traditional business activities, supercomputer style installations are increasingly used to perform tasks such as trading support and risk analysis. These computations consume large amounts of compute resources and have requirements not found in traditional supercomputing. While the application area is different from the scientific explorations of traditional supercomputing, the underlying computations are very similar. Both are large scale simulations involving the solving of complex systems of equations, often by numerical techniques such as Monte Carlo. As in other areas, the desire for improved performance and reduced cost has led to the use of compute accelerators or coprocessors. Today, these accelerators primarily take the form of discrete GPUs. It is expected, however, that a variety of forces will lead to increased use of accelerators of differing architectures for financial applications. The classification of accelerators into 'horizontal' and 'vertical' architectures is proposed and some future directions for accelerator architectures explored.

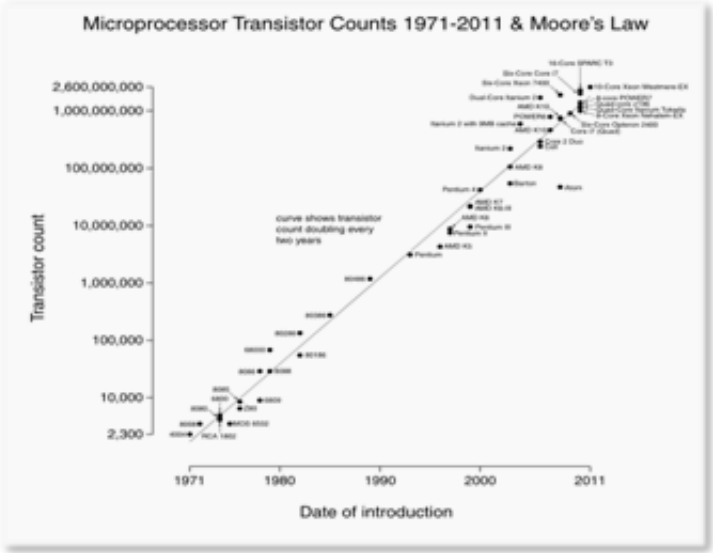
An Introduction to Compute Acceleration

The Baseline: Instruction Set Architectures

- Central Processing Units (CPU)
- Linear sequence of instructions
- ‘Von Neumann” architecture
- Modest parallelism (1 – 10x)
- Very flexible
- Almost half a century of success
  - Intel 4004 – 4 bit / 0.74 MHz (1971)
  - Intel Core series – 64 bit / 4.0 GHz (2015)
- Orders of magnitude performance increases
- Billions of units sold
- One of mankind’s greatest accomplishments
- Jack of all Trades
- Master of None

Silicon Technology

- CPU performance driven by improvements in silicon technology
- **Dennard Scaling**: transistors scale independent of power
- **Moore's Law**: transistor count doubles in 2 years
- Moore / Dennard made using accelerators difficult
  - Just wait a couple of years and get 2x from CPUs
  - Competes against multiple CPUs
  - Limited to architectural changes
  - Increased software complexity
  - Market necessarily narrow
  - Keeping costs low difficult
- Dennard scaling may be breaking down
- *Dark Silicon:* scaling limits of silicon
- May be good news for accelerators



Accelerators / Coprocessors

- Increased performance over CPUs
- Usually application area specific
- Special purpose coprocessing
  - FPUs in the 1980s
  - SIMD / MMX in the 1990s
  - Multicore in the 2000s
  - GPUs in the 2010s
- Challenges:
  - CPUs catch up quickly
  - Costs must be low
  - Market necessarily small
- CPU integration if successful

Accelerators: Early History (pre-1990s)

- Offloaded computation from CPU
- Often instruction-level acceleration
- Required custom hardware (board level or ASIC)
  - Expensive
  - Difficult to upgrade
  - Very limited functionality
- Not generally 'programmable'
- Vector / floating point coprocessors
  - Early systems board-level
  - Formalized with FPU instructions
  - Enhanced with MMX instructions
- Became tightly integrated with CPUs
- Eventually integrated into CPU

Accelerators: Early Programmable Solutions

- Systolic Arrays
  - Work by H. T. Kung and others at CMU (1980s)
  - Programmable pipelined architecture
  - Commercialized briefly by Intel (iWarp)
- Field Programmable Gate Arrays (FPGAs)
  - Originally used in 'glue logic' on circuit boards
  - Increasing density and lower cost made computing feasible
  - Reconfigurable / Adaptive Computing
- Graphics Processing Units (GPUs)
  - Silicon Graphics, ATI, Nvidia, S3, many others
  - Increasingly programmable
  - API interface (OpenGL)
  - High level functionality

Early FPGA Accelerators

- 10,000x speedups for selected algorithms
- CPU + FPGA silicon (2000s)
  - Xilinx Virtex II Pro (PowerPC)
  - Altera Excalibur (ARM)
- DARPA Adaptive Computing Systems (ACS) program
- A large number of hardware and software companies
  - FPGA boards for acceleration
  - Tools for software development
  - Application specific systems (image processing)
- Reconfigurable Computing Eras:
  - 1990s: Bit level (gene matching, cryptography)
  - 2000s: DSP (image processing, signals, etc)
  - 2010s: Floating point (traditional supercomputing)

FPGA Accelerators (cont.)

Adaptive Computing Systems

"The microchip that rewires itself"

June 1997

"Computers that modify their hardware circuits as they operate are opening a new era in computer design. Because they can filter data rapidly, they excel at pattern recognition, image processing and encryption"

Approved for Public Release, Distribution Unlimited

4

GPU Accelerators

- Accelerators for computer graphics
- Many GPU companies (Nvidia, ATI, Voodoo, S3, 3DLabs, Matrox, …)
- Started with high end workstations (Silicon Graphics, CAD)
- Migrated to low end (PC)
- Investigations into general purpose computation (GPGPU)
  - Brook / Stream Computing (Stanford)
  - OpenCL
- Evolved into more general-purpose multiprocessors
  - Floating point standardized
  - Integer and bit operations added
  - All features for HLLs like ‘C’ added
- Nvidia’s CUDA environment commercialized GPGPU
- Finding widespread use in supercomputing (Top 500)

Financial Computations

Computation in Finance

- Banking / Finance an early user of computers and networking
- $500B per year spent in Finance and Banking IT worldwide
- Many functions of compute technology in finance
  - Commercial Banking (ATMs, mobile banking)
  - Record keeping / archives
  - Others (web servers, etc)
- Compute intensive calculations
  - Trading support
  - Reporting
  - Risk analysis
- Core calculations all involve asset pricing

Asset Pricing

- Asset pricing is calculating value of a *financial instrument*
- Crucial to most aspects of finance
- Solving *Black-Scholes* and similar equations
  - Partial differential equation
  - Identical to Heat Equation in physics
  - ***V(S,t)*** is derivative price
  - ***S*** is the stock price
  - ***σ*** is standard deviation of stock's return
  - ***r*** is the risk-free interest rate

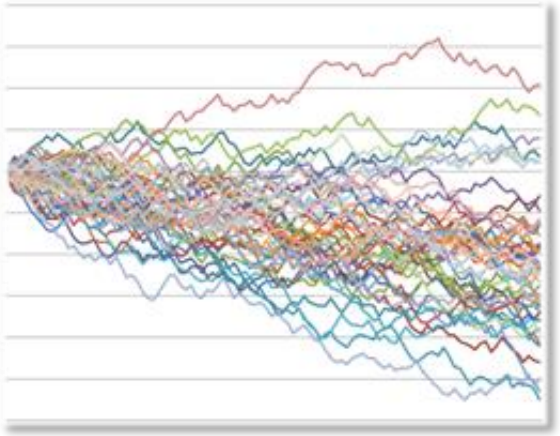
$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

Asset Pricing Computations

- Various types of options:

- *European*	- *Asian*	- *Binary*	- *Shout*	- *Basket*
- *American*	- *Russian*	- *Verde*	- *Double*	- *Rainbow*
- *Bermudian*	- *Loopback*	- *Canary*	- *Compound*	. . .

- Only the *European* option has a solvable PDE ('vanilla')
- All 'non-vanilla' options require simulation techniques
- Monte Carlo and similar techniques used
- Floating point intensive
- Compute bound
- Slow convergence
- Accuracy important (double precision)



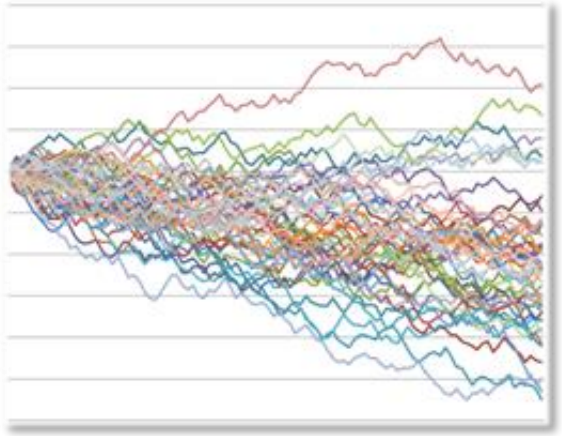
Asset Pricing Computations (cont.)

- Large and expanding list of algorithms and techniques
- All similar or greater complexity compared to Black-Scholes

- *Black-Scholes*
- *Brace-Gatarek-Musiela (BGM)*
- *Heath–Jarrow–Morton (HJM)*
- *Ho-Lee*
- *Black-Derman-Toy*
- *Hull-White*
- *Vasicek*
- *etc., etc., etc.*

- Added computation of *'Greeks'*
  - 1st and 2nd derivatives
  - Change and sensitivity of price

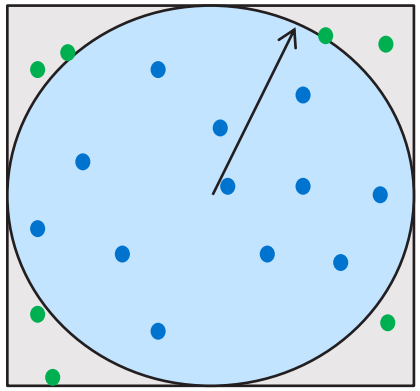
- *Delta*
- *Vega*
- *Theta*
- *Gamma*
- *Vanna*
- *Charm*
- *Vomma*
- *Veta*
- *Speed*
- *Zomma*
- *Color*
- *etc.*



Monte Carlo Method

- Typical type of calculation used in option pricing
- Used when there is no closed-form solution to equation
- Example: Calculating Pi (π)
  - Generate a series of random coordinates
  - Ratio of points in circle to points outside converges to Pi

Random Points	Pi Value	Error
1,000	3.176	-0.0344073
10,000	3.1424	-0.000807346
100,000	3.14604	-0.00444735
1,000,000	3.14133	0.000260654
10,000,000	3.14156	0.0000366536
100,000,000	3.14165	-0.0000617064

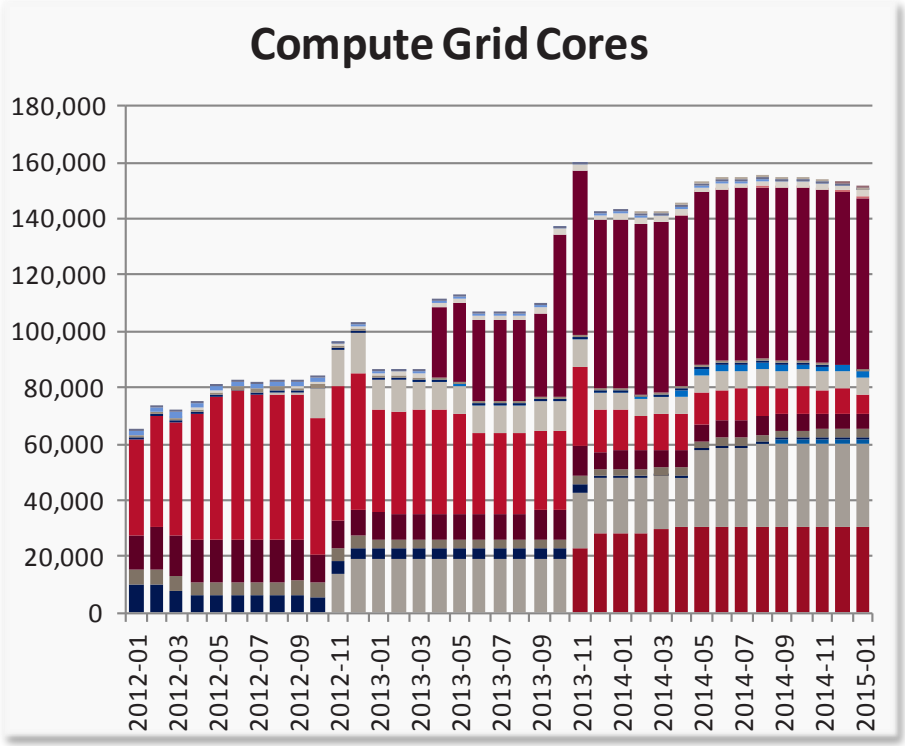


Pricing and Risk Computation

- Bank portfolios on the order of one trillion of dollars US
- Total value computed regularly
  - Regulatory requirements
  - Management
- Risk Analysis: re-compute under different scenarios
  - 100s to 1000s of scenarios
  - Required for regulatory reporting
- Some potential variables:
  - Interest rates
  - Foreign exchange rates
  - Counterparty default risk
  - Economic growth rates
  - Commodity prices
- Daunting level of compute

Grid Computing at Bank of America

- A shared compute resource ('Bare Metal Cloud')
  - Approx. 150,000 x86 CPU cores
  - Approx. 2,000 GPUs (~1.5 PFLOP)
  - Multiple data centers
- Software
  - Red Hat Enterprise Linux
  - Custom Middleware
  - Mostly Java, Python, C/C++
  - No VMs
- What we don't do (yet)
  - Web servers
  - Storage / database
  - Commercial banking / ATMs

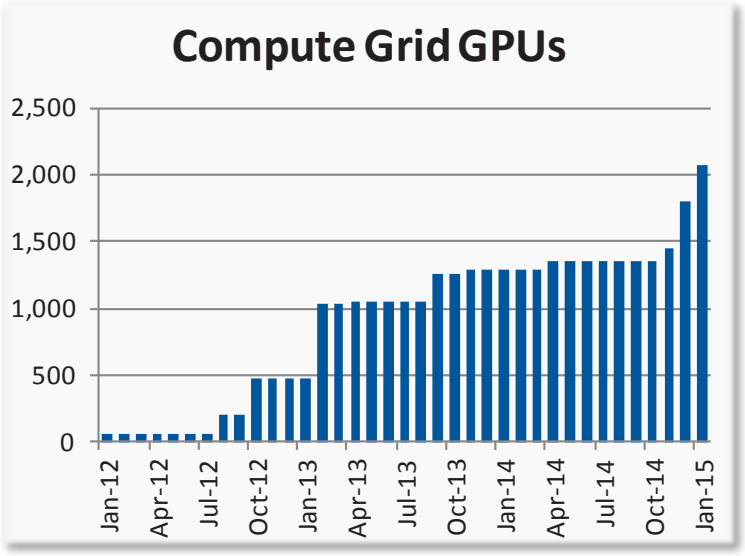


Grid Computing at Bank of America: Challenges

- More than a collections of servers (*Amazon EC2, Google*)
- High reliability
  - Jobs guarantees (SLAs)
  - Failure tolerance at all levels
  - Applications across multiple data centers
  - HW and SW resiliency
  - More than just high (average) uptime
- Security
- Accuracy
- Accelerators support
- Continuously upgraded
- Did I mention *security*?
- Half the price of competing offerings (2013)

Accelerating Pricing Computations

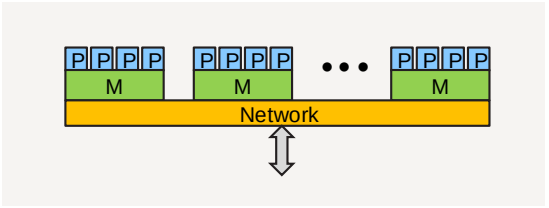
- Recent move to GPUs
  - Large amounts of computation
  - Modest amounts of I/O
  - GPUs reduce cost by 5x
- Higher efficiency permits:
  - New computations / models
  - Higher accuracy
- Most of FLOPs on grid now GPU
- Always exploring other alternatives
  - New CPU technology (vector instructions)
  - New architectures (Intel Phi)
  - Lower cost approaches (ARM, mobile CPU / GPUs)
  - Field Programmable Gate Arrays (FPGAs)
  - Others

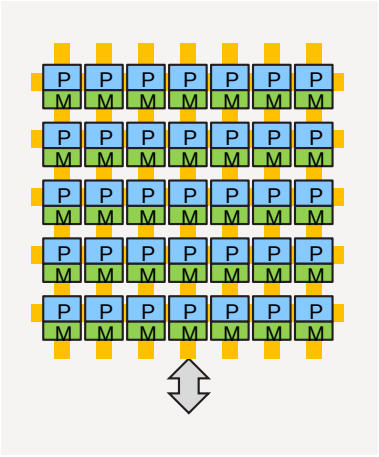


Accelerator Architectures

Modern Accelerator Families

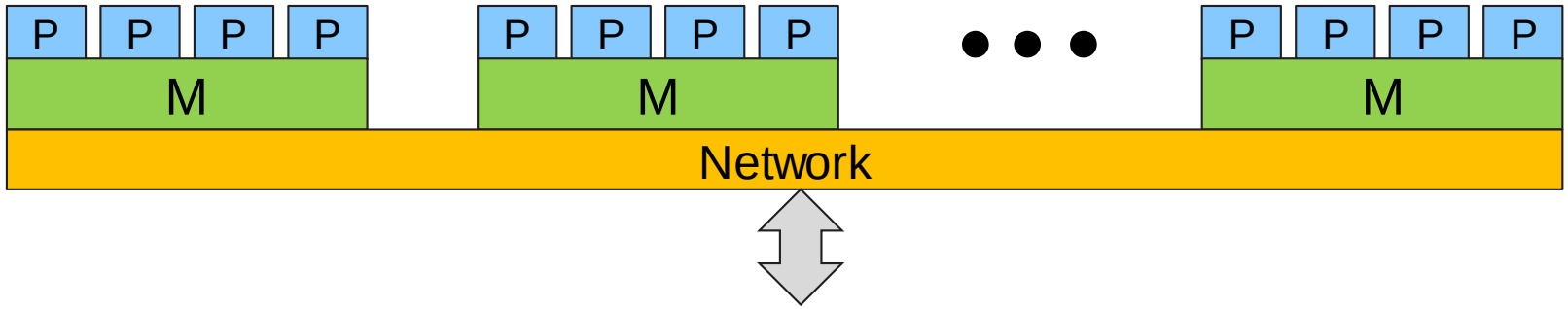
- *Horizontal* accelerators
  - ISA processors
  - Highly parallel / SIMD instructions
  - Shared memory
  - Memory interface to host
  - *Examples*: GPUs, Intel Xeon Phi
- *Vertical* accelerators
  - Programmable logic
  - Programmable interconnect
  - Direct / network interface to host
  - *Examples*: FPGAs, Array Processors





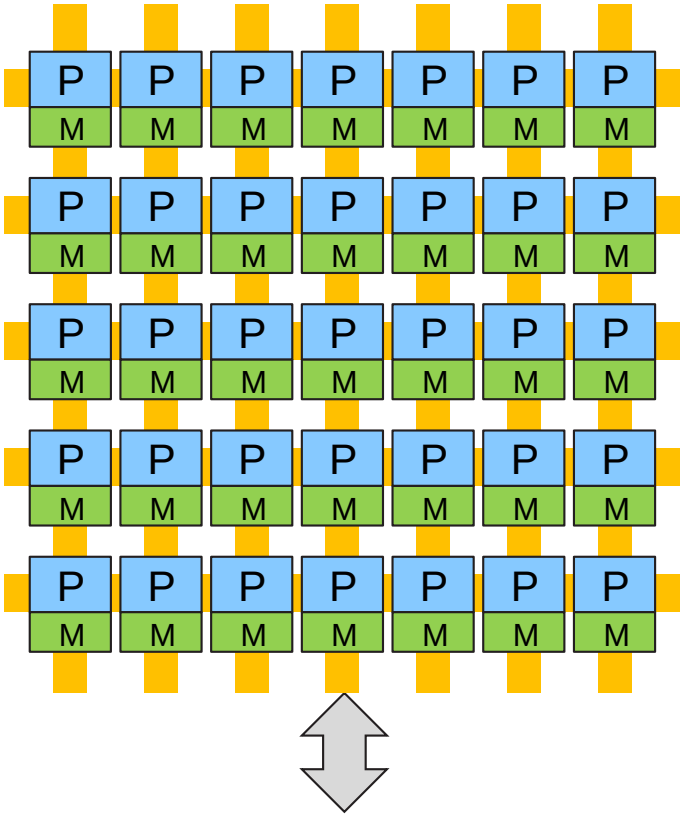
Horizontal Accelerator Architectures (GPUs)

- Shared memory interface
- High latency
  - Often uses very high numbers of threads to hide latency
  - Memories must be filled before the computation begins
- High throughput
- Favors Data Parallel algorithms
- Programmable in High level Languages (HLLs)
- Scheduling issues when Parallelism ≠ Processors

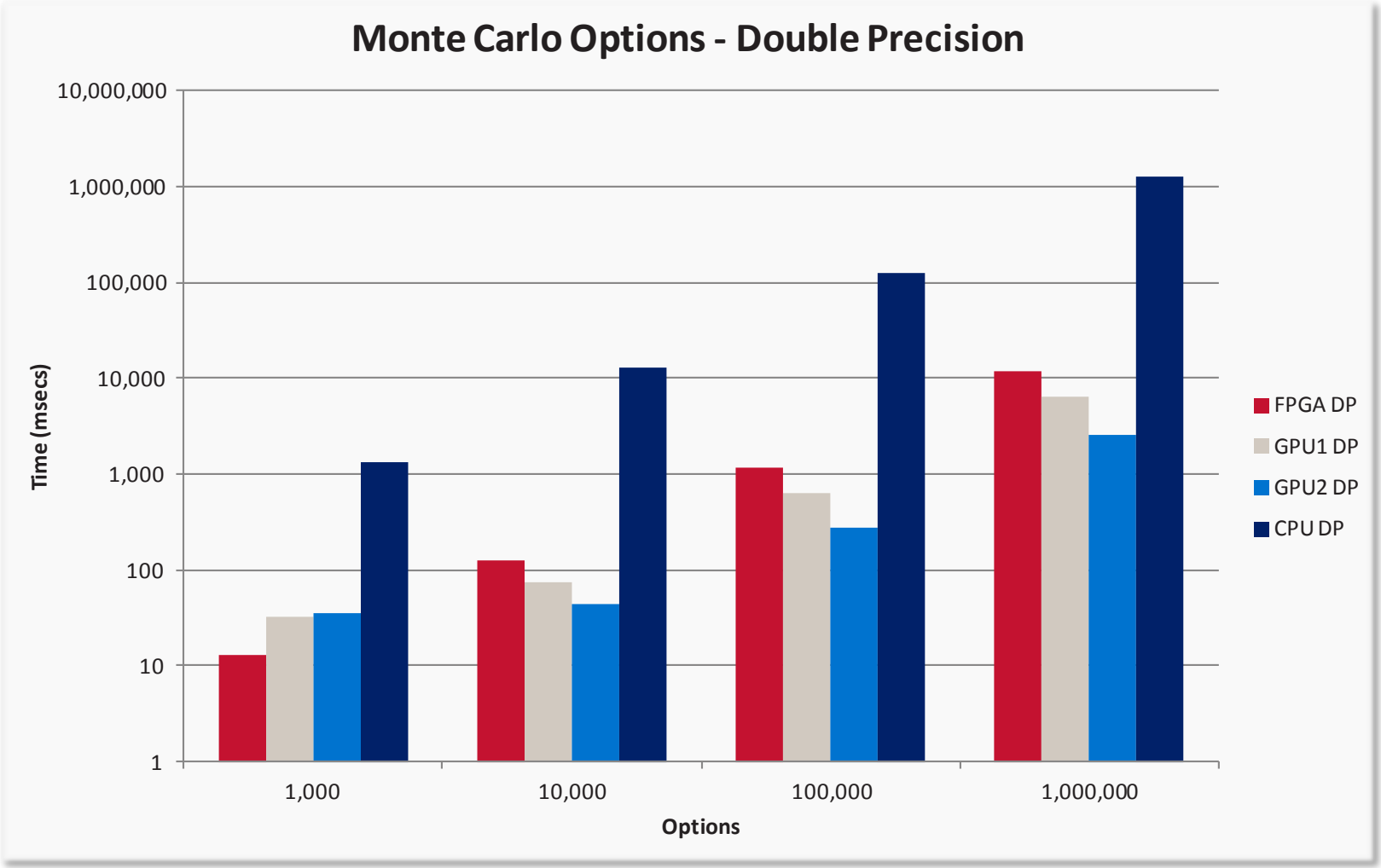


Vertical Accelerator Architectures (FPGAs)

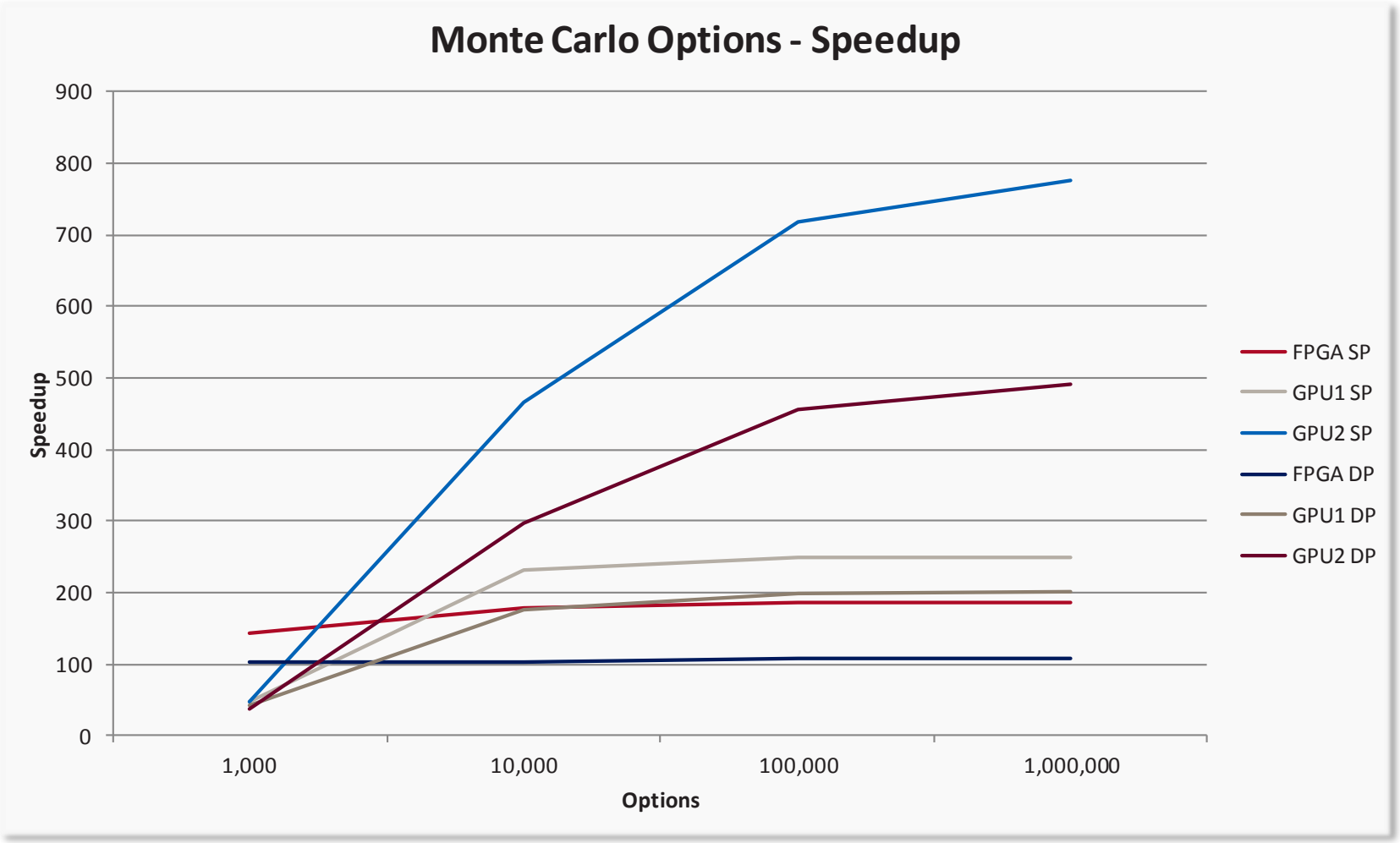
- Direct (network) interface
- Low latency
- High throughput
- Highly pipelined
- Programmable in
  - Hardware Description Languages
  - Some HLLs, OpenCL in 2015
- Configurable interconnect
- Reconfiguration possible
- Processors
  - May be very simple (LUTs)
  - Could be more complex (CPUs)
  - May have a mix of processors
- Very power efficient



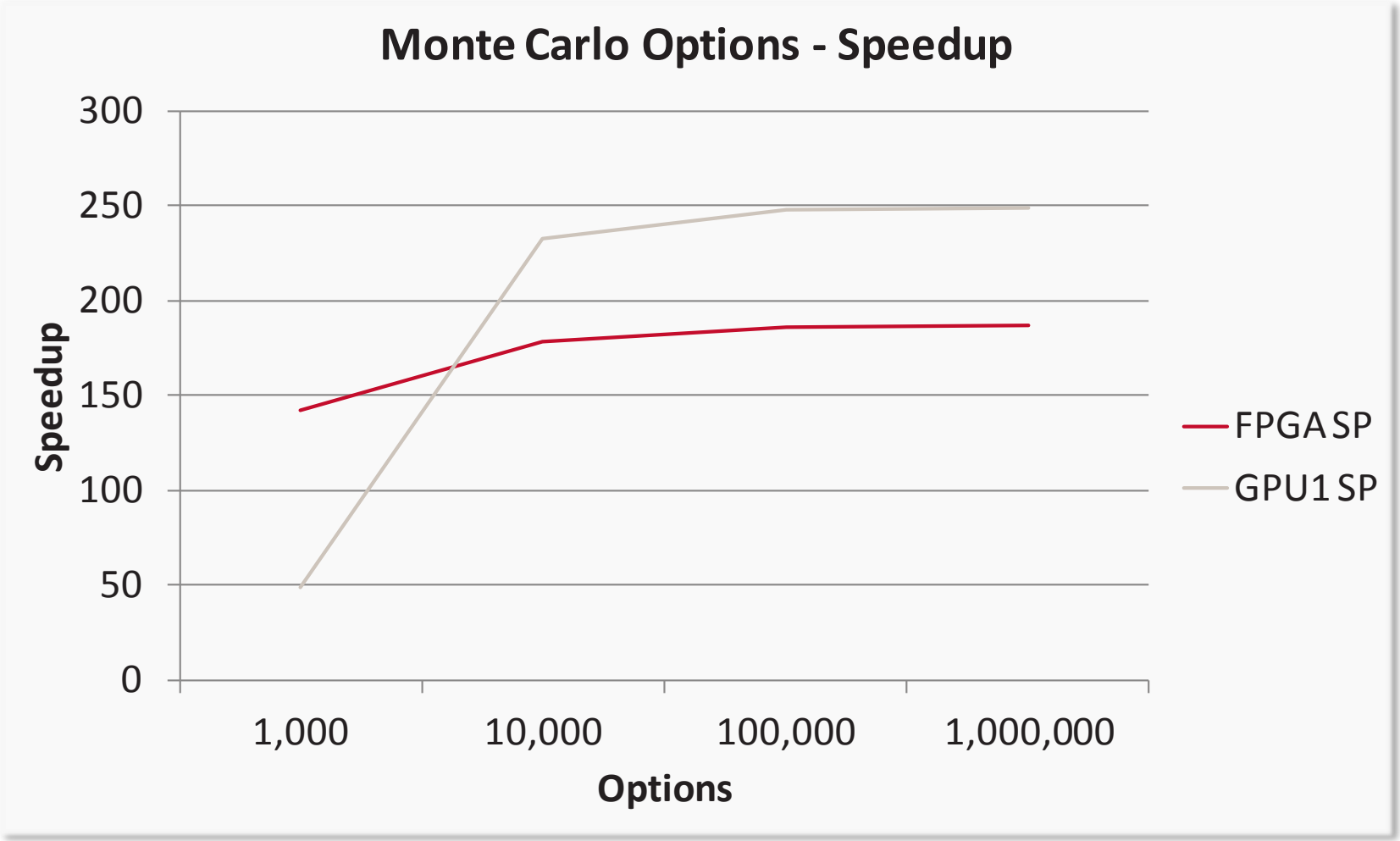
Comparing Accelerators



Comparing Accelerator Speedups



Comparing Accelerator Speedups (cont.)



The Problem with Accelerators

- Moore's Law / Dennard Scaling more forgiving in 2015
- Still competing with CPUs
  - Ongoing performance improvements
  - Ongoing process improvements (14 nm)
  - Very competitively priced
- Relies on other markets for high volume / low cost
  - GPUs are really made for graphics
  - FPGAs are really sold to networking companies
  - 'Product abuse' (A. Krumel)
  - HPC may not be able to support purpose built silicon
- Difficult to program (OpenCL is an improvement)
- Restricted to highly parallel computations
- (same as in the 1980s)

The Future of Accelerators

- Power is limiting Horizontal Architectures (GPUs)
- GPUs are being absorbed into CPUs
- FPGAs now sharing MCM with Intel CPU
  - We have seen CPU + FPGA before
  - It could be different this time (driven by CPU vendor)
- FPGAs increasing size and number of FPUs
  - Thousands of hard-wired Double Precision floating point units
  - Beginning to look a bit like a Horizontal Architecture
- Programmable interconnect power efficiency hard to ignore
- Future convergence of Horizontal and Vertical Architectures?
  - Highly programmable processors with local memory
  - Programmable Interconnect

References

• Bank of America: https://www.bankofamerica.com/
• Grid Computing: http://en.wikipedia.org/wiki/Grid_computing
• Black-Scholes Model: http://en.wikipedia.org/wiki/Black%E2%80%93Scholes_model
• Monte Carlo Method: http://en.wikipedia.org/wiki/Monte_Carlo_method
• Mathematical finance: http://en.wikipedia.org/wiki/Mathematical_finance
• Moore's law: http://en.wikipedia.org/wiki/Moore's_law
• Dennard scaling: http://en.wikipedia.org/wiki/Dennard_scaling
• Dark silicon: http://en.wikipedia.org/wiki/Dark_silicon
• Option Style: http://en.wikipedia.org/wiki/Option_style
• Singh, Satnam, "The Last Programmer Standing will be holding an FPGA", http://blogs.msdn.com/b/satnam_singh/archive/2011/08/01/the-last-programmer-standing-will-be-holding-an-fpga.aspx , August 1, 2011.
• Black, Fischer and Scholes, Myron. "The Pricing of Options and Corporate Liabilities". *Journal of Political Economy* **81** (3): 637–654.
• Greeks (finance), http://en.wikipedia.org/wiki/Greeks_(finance)
• IT in banks: What does it cost?, Deutsche Bank AG DB Research, https://www.dbresearch.com/PROD/DBR_INTERNET_EN-PROD/PROD0000000000299039/IT+in+banks%3A++What+does+it+cost%3F+High+IT+costs+call+for+an+eye+on+efficiency.pdf, December 20, 2012.
• Holding Companies with Assets Greater Than $10 Billion, National Information Center, http://www.ffiec.gov/nicpubweb/nicweb/top50form.aspx.
• Transistor Count and Moore's Law, http://commons.wikimedia.org/wiki/File:Transistor_Count_and_Moore%27s_Law_-_2011.svg, 13 May 2011.

References (cont.)

• H. Esmaeilzadeh et al., "Dark silicon and the end of multicore scaling", in 38th International Symposium on Computer Architecture (ISCA), pages 365 –376, 2011.
• Steven A. Guccione, List of FPGA-based Computing Machines, https://sites.google.com/site/stevenguccione/home/hw_list-html, August 21, 2000.
• Coprocessor, http://en.wikipedia.org/wiki/Coprocessor
• Graphics Processing Units, http://en.wikipedia.org/wiki/Graphics_processing_unit
• Xilinx enhances FPGAs with embedded PowerPCs, EE Times, http://www.eetimes.com/document.asp?doc_id=1200084, March 4, 2002.
• Altera Unveils Technical Details of ARM- and MIPS-Based Excalibur Offerings, EE Times, http://www.eetimes.com/document.asp?doc_id=1214924 , Oct. 25, 2000.
• Jose Munoz, Adaptive Computing Systems (DARPA), http://fas.org/spp/military/program/imint/acs_approved.pdf, May 1998.
• General-purpose computing on graphics processing units, http://en.wikipedia.org/wiki/General-purpose_computing_on_graphics_processing_units
• Top 500 Supercomputing Sites, http://www.top500.org/

Attributions

• "Swiss army knife open 20050612 (cropped)" by Jonas Bergsten - Photo taken by Jonas Bergsten using a Canon PowerShot G3.. Licensed under Public Domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Swiss_army_knife_open_20050612_(cropped).jpg#mediaviewer/File:Swiss_army_knife_open_20050612_(cropped).jpg

• "Cabinet Asile" by Robert.Harker - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Cabinet_Asile.jpg#mediaviewer/File:Cabinet_Asile.jpg

• Canada's Skyhawk skydiving team jumping over the Nation's capital. Picture by Jeff Hallam, available at http://www.sxc.hu/browse.phtml?f=view&id=300606, which states "There are no usage restrictions on this photo".

•Philippine-stock-market-board.jpg, 30 July 2008. http://www.flickr.com/photos/thewalkingirony/3051500551/, Katrina.Tuliao.

•"WinonaSavingsBankVault" by Jonathunder - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:WinonaSavingsBankVault.JPG#mediaviewer/File:WinonaSavingsBankVault.JPG

• Federal_reserve_bank_of_chicago.jpg, From the Chicago's Fed home page; should be in the public domain.

• Opened umbrella, http://commons.wikimedia.org/wiki/File:Umbrella_opened.svg Creative Commons License.

• Black Umbrella Clip Art, http://www.clipartbest.com/clipart-LiKa44nia (Free clip art)

• Ensign John Gay, U.S. Navy, FA-18 Hornet breaking sound barrier, http://commons.wikimedia.org/wiki/File:FA-18_Hornet_breaking_sound_barrier_(7_July_1999).jpg, 7 July 1999, This Image was released by the United States Navy with the ID 990707-N-6483G-001. This file is a work of a sailor or employee of the U.S. Navy, taken or made as part of that person's official duties. As a work of the U.S. federal government, the image is in the public domain.

• Picture of Intel 80386 CPU with Intel 387 Math Coprocessor , http://commons.wikimedia.org/wiki/File%3A80386with387.JPG, By Bumper12 (Own work) [Public domain], via Wikimedia Commons

Biography

Dr. Steven A. Guccione is Vice President, Senior Technology Manager at Bank of America Merrill Lynch in the Grid Engineering group. His interests are in high performance computing, in particular software and tools for parallel computing. Dr. Guccione received his Bachelor of Science degree in Electrical and Computer Engineering from Boston University, his Master of Science degree in Electrical Engineering from the University of Minnesota and his Ph.D. degree in Electrical Engineering from the University of Texas at Austin. His Ph.D. work was an early contribution in the field of reconfigurable computing, and one of the first to address software and system issues. Dr. Guccione has approximately 40 published papers and approximately 20 issued patents in this field. He has previously held engineering positions at Xilinx, Advanced Micro Devices, Motorola/MCC, Texas Instruments and several smaller companies.


Bank of America
Bank of America
Merrill Lynch
U.S. Trust
Bank of America
Merrill Lynch

Moore's Law Graph