ClearSpeed™

The Return of Acceleration Technology

John L. Gustafson, Ph.D.
CTO, HPC
ClearSpeed Technology, Inc.

Thesis

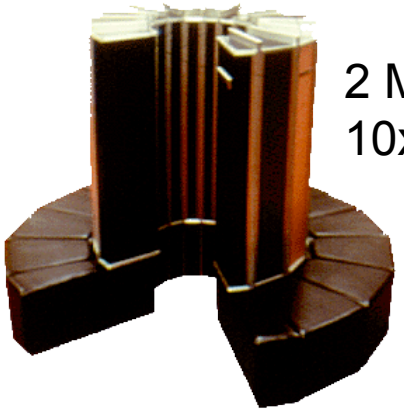
25 years ago, transistors were expensive; accelerators arose as a way to get more performance for certain applications.

Then transistors got very cheap, and CPUs subsumed the functions. Why are accelerators back?

Because supercomputing is no longer limited by cost. It is now limited by heat, space, and scalability. Accelerators once again are solving these problems.

The accelerator idea is as old as supercomputing itself

3 MB/s



2 MB
10x speedup

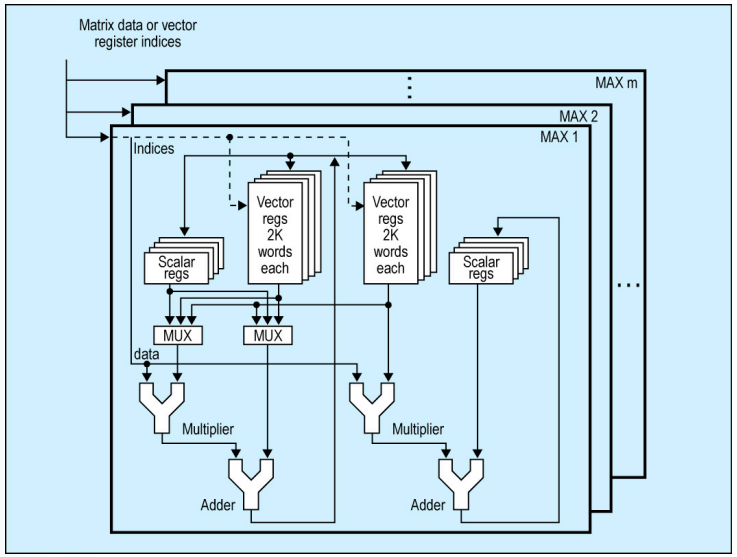
General-purpose computer Runs OS, compilers, disk, printers, user interface

Attached vector processor accelerates certain applications, but not all

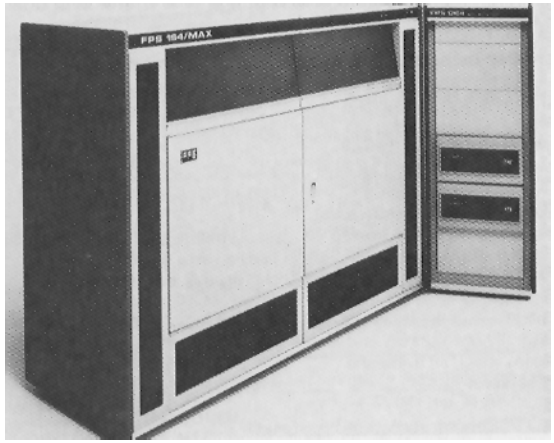
Even in 1977, HPC users faced issues of when it makes sense to use floating-point-intensive vector hardware.

"History doesn't repeat itself, but it does rhyme."
—*Mark Twain*

From 1984… My first visit to Bristol



- **30 double precision PEs under SIMD control; 16 KB of very high bandwidth memory per PE, but normal bandwidth to host**
- **Specific to matrix multiplication**
- **Targeted at chemistry, electromagnetics, and structural analysis**



FPS-164/MAX
0.3 GFLOPS, $500,000
(1/20 price of a 1984 Cray)

This accelerator persuaded Dongarra to create the Level 3 BLAS operations, and to make the LINPACK benchmark scalable… leading to the TOP500 benchmark a few years later.

The Beowulf approach was to lower the *cost*

Thomas Sterling's 1997 Recipe:

Consumer-grade electronics

+

Armies of "free" graduate students

=

TOP500® SUPERCOMPUTER SITES

Dirt-cheap supercomputing?

Sterling now believes that the only way forward is with* hybrid *architectures: Accelerators

HPC limits, obvious and subtle

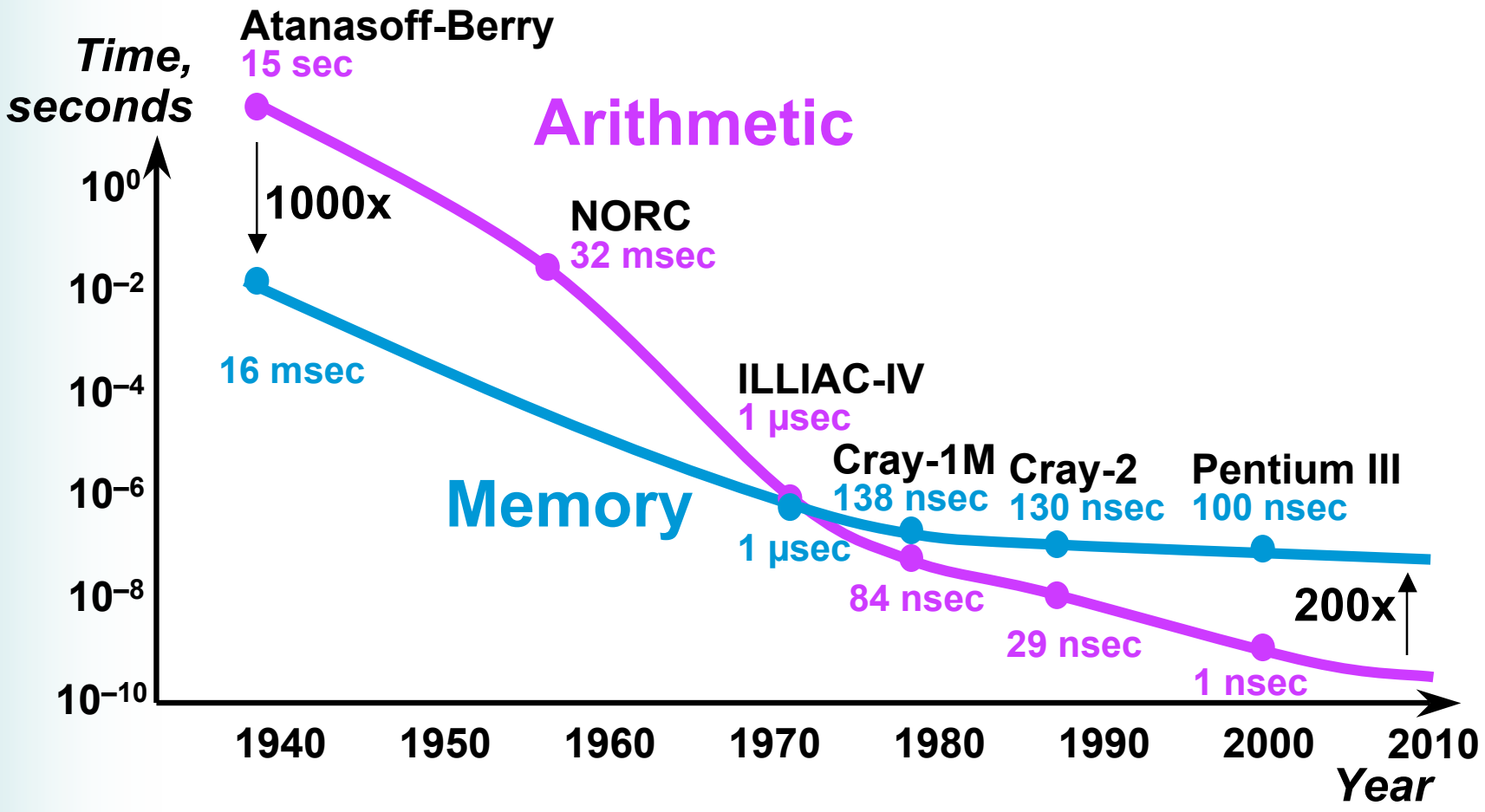
What we all know:

- **25–40 kilowatts per cabinet, 300–1500 watts/node)**
- **Each rack consumes about 12 sq. ft.**
- **Typical rack (over 1000 pounds) taxes floor loading limits**

More subtle… *too many nodes*:

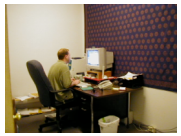
- **Switches hit scaling limits around 10,000 nodes**
- **MPI, OS jitter, reliability, app. parallelism hit the wall, too**

History of processor latency vs. memory latency



ClearSpeed™

If your workday were like processor balance



- Spend 4 hours commuting from home to work.
- Work for 2.4 minutes.
- Take 4 hours to commute home.

Watts are from wires, not switches!

Operation	Energy (130 nm, 1.2 V)
32-bit ALU operation	5 pJ
32-bit register read	10 pJ
Read 32 bits from 8K RAM	50 pJ
Move 32 bits across 10 mm chip	100 pJ
Move 32 bits off chip	1300 to 1900 pJ

Debates like this have raged for years…

System User

Make the communication faster!

Then make the wires shorter.

What about liquid cooling?

How about optical interconnect?

So use optoelectronic converters.

Why don't you make the whole system optical?

Because then we have to THINK.

System Designer

Can't. Near the speed of light now.

If we do, the computer will melt.

Dead-end solution, and expensive.

Yeah, but the logic is electrical.

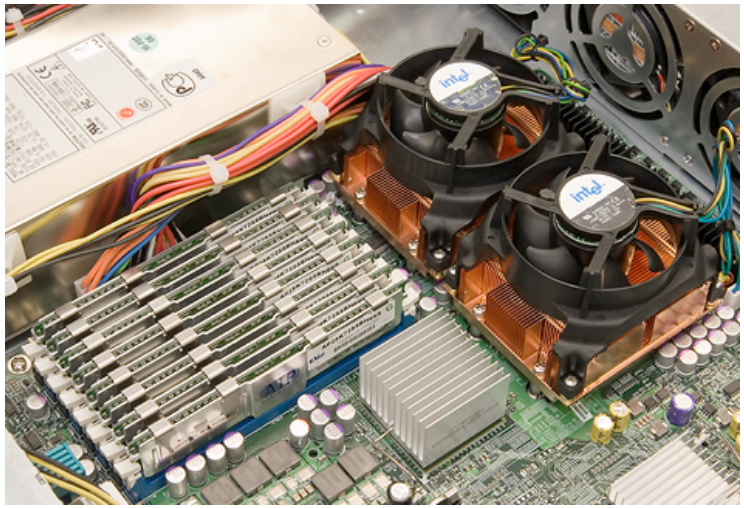
That adds latency everywhere.

Why don't you just learn how to use machines with distributed memory?

I see.

Simple view of power per processor chip

$$Power \propto capacitance/device \times devices/chip \times Voltage^2 \times clock\ rate$$



- **Energy density is like that of stove top heating coils**
- **At limits of forced air cooling to keep chip from overheating and failing**
- **Chip lifetimes are dropping from 20 years to… 3?**

Remembering basic physics…

A Watt is a Joule per second.

A nanojoule times a gigahertz is 1 watt.

Feeding a 100 Gflops/sec processor chip from *external* memory (6 32-bit moves per op) takes

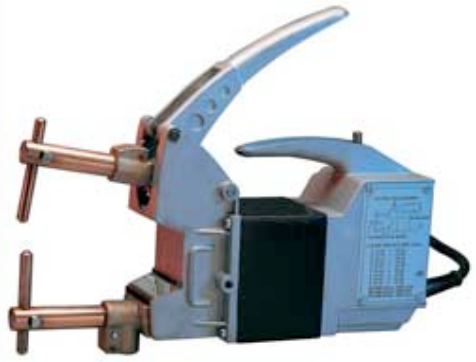
6 × 1.9 pJ × 100 Gflops/sec = 1140 Watts (*gulp!*)

Wait, it gets worse…

A Watt is also a Volt times an Amp.

New processor chips are about 150 Watts,

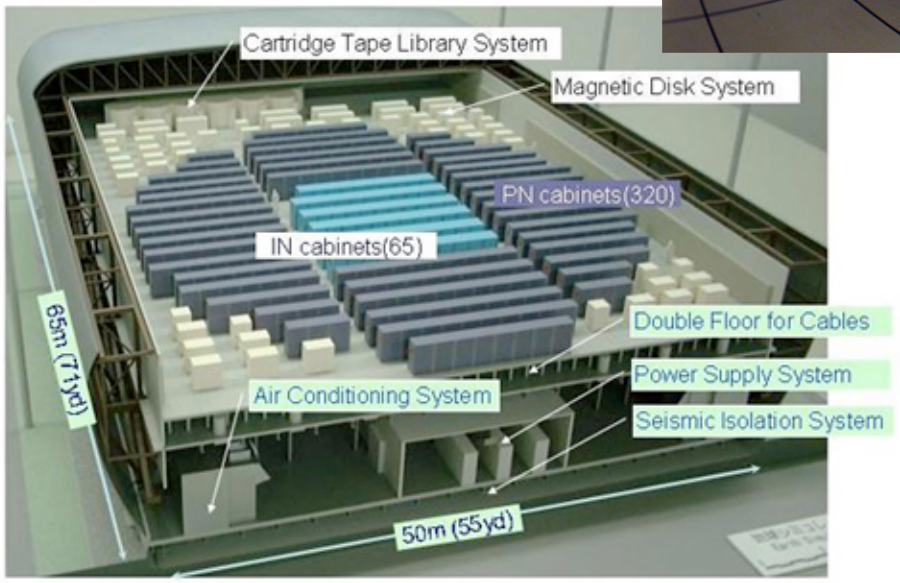
But at only *one Volt…* That's *150 Amps!*



**IN OTHER WORDS,
AN ARC WELDER.**

Example of volume-limited computing

The CPUs can perform 10 million operations in the time it takes a photon to traverse the Earth Simulator facility.



6 megawatts.

It doesn't just simulate global warming. It *causes* global warming.

The new limits to supercomputing

- **The current fastest general-purpose supercomputer, Ranger (U of Texas) only cost $30M. Other top supers have cost $200–500M!**
- **In 2008, you don't run out of money first; you run out of**
  - Electrical power
  - The ability to remove dissipated heat
  - Space
  - Reliability
  - Algorithm scalability to over 60,000 instruction streams
- **Dirty little secret of HPC: "Capability" systems are now operated as *Capacity* systems. Only a few hundred instruction streams used per job, tops.**

Three "shackles" from the 20th Century

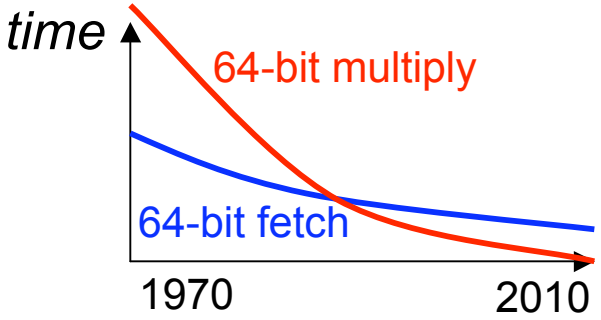
1. **Floating-point arithmetic is hard, especially 64-bit precision, so you must use the algorithm that does the fewest possible operations.**
2. **Memory is expensive, dominating system cost, so you must make sure your program and data fit in the fewest possible bytes.**
3. **Parallel programming is hard because you have to coordinate many events, so you must express your algorithm sequentially.**

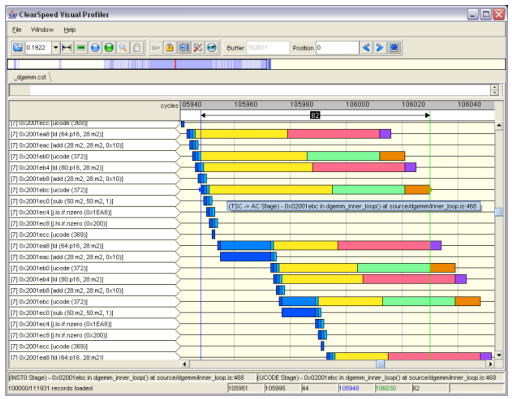

```
LOAD A(I)
ADD B(I)
STORE C(I)
INCREMENT I
IF I < N GO TO
```

The shackles still influence the way we use systems, but we must consciously move away from this mind set.

21st Century reality: Dirt-cheap transistors

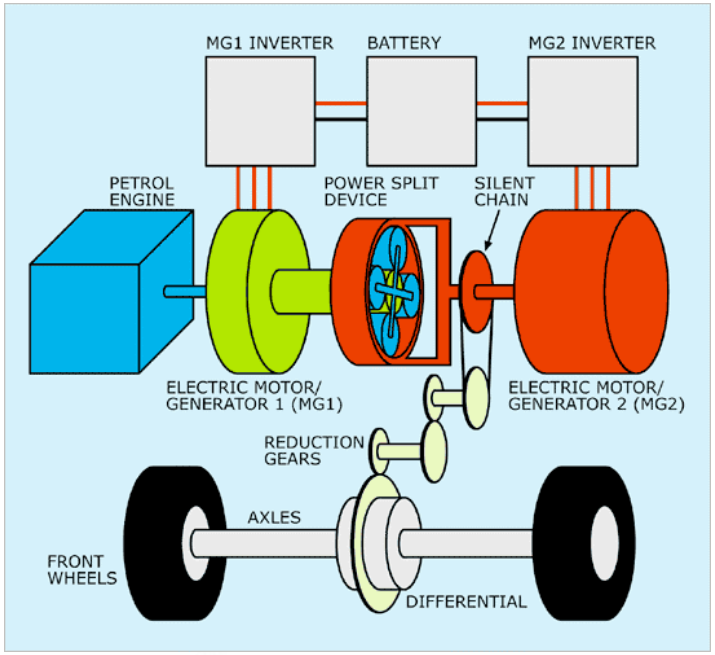
- **Floating-point arithmetic is a miniscule part of the execution time and cost, hidden by data fetches and stores.**
- **Memory is so inexpensive per byte that we think nothing of gigabytes sitting idle.**
- **A single thread can easily express data parallelism, and heterogeneous parallel threads can be coordinated if you have tools that provide *full visibility*.**





Idea: Hybrid tech increases performance density

- Hybrid cars use two engine types, each for a different kind of driving, to reduce energy use
- Why not use two processor types, each for a different kind of computing?
- Energy savings allows computing in a much smaller volume



GPUs paved the way for HPC accelerators

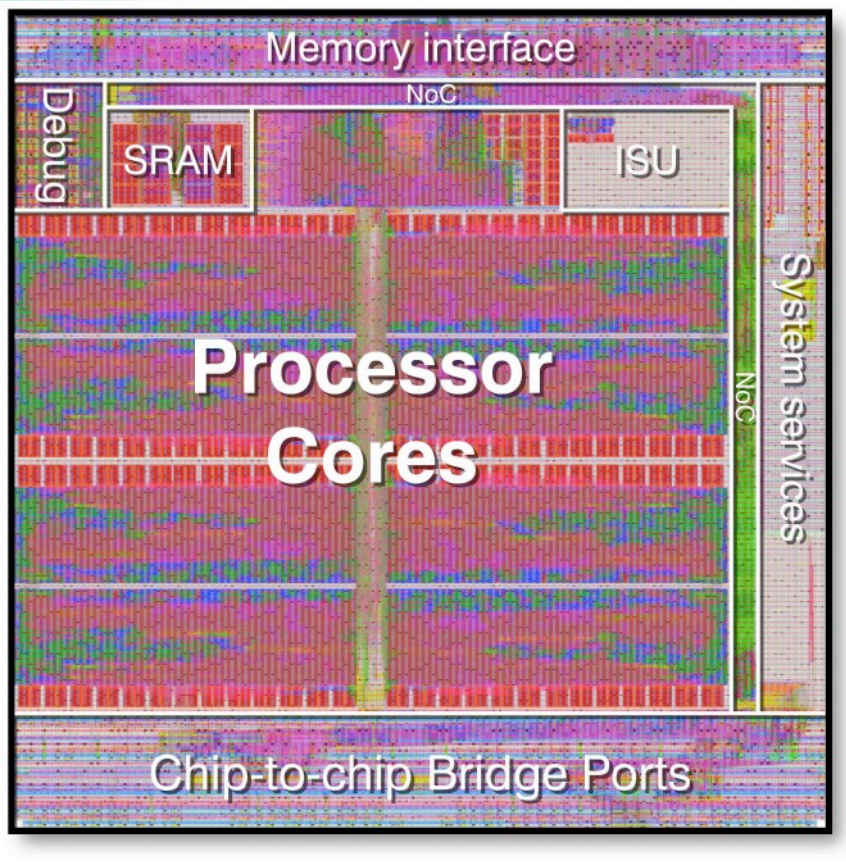
- The Nvidia and ATI model of use has led to *dynamically-linked libraries*.
- HPC could never have overcome that hurdle by itself with ISVs. (Widespread use of MPI helped, too.)
- Plug-and-play acceleration is now available for environments like MATLAB, Mathematica®… with more on the way.

HPC accelerators return: The ClearSpeed approach

- **Lower the clock to 0.25 GHz, but do far more per clock**
- **Architect for HPC: use a high ratio of 64-bit arithmetic functional units to control threads**
- **Don't use data caches, since they waste 80% of the bytes moved**
- **Embrace the need for very large performance ratio between local and global memory**
- **Create a very power-efficient bus (ClearConnect) that uses power only when communicating**

The combination of approaches relieves the previously-mentioned limitations to HPC.

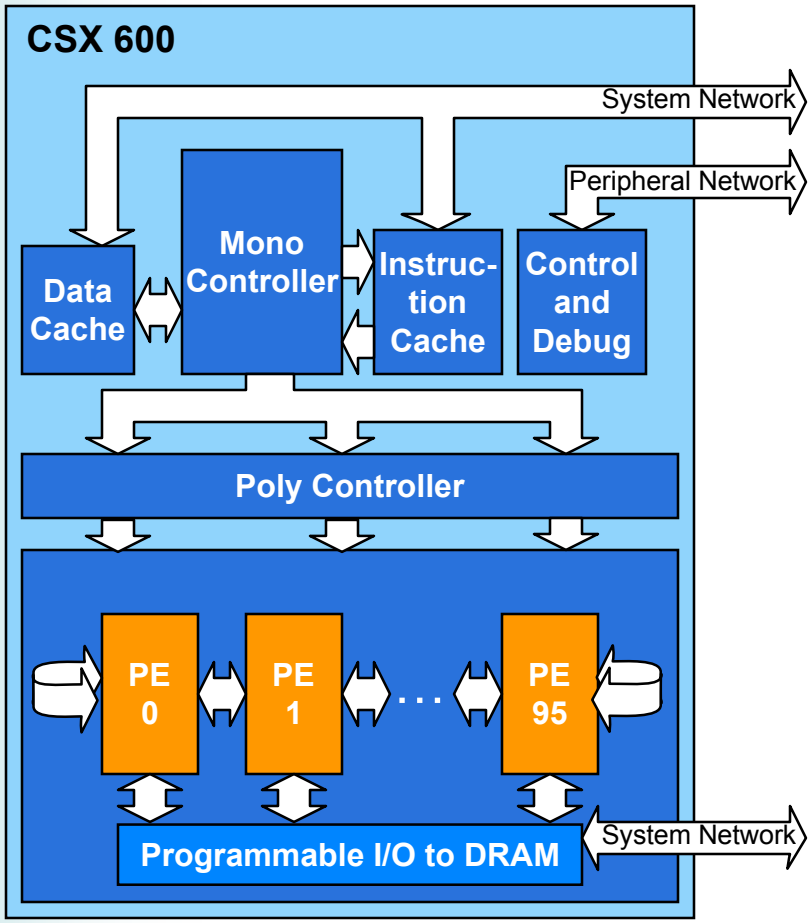
Example of accelerator designed specifically for HPC



ClearSpeed CSX600

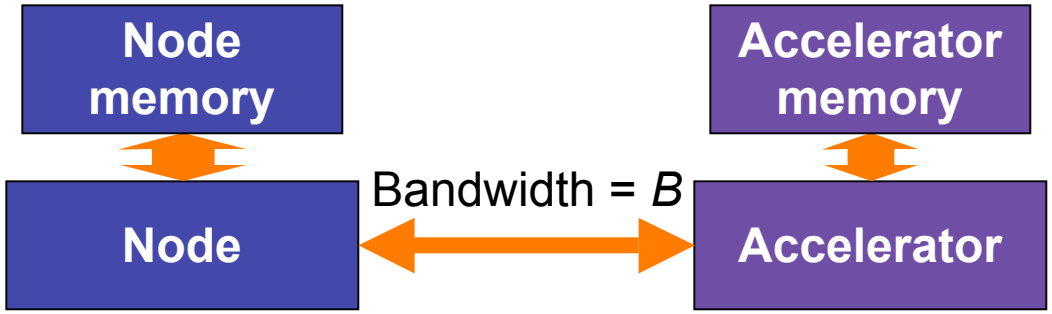
- **2005-era technology still beats the latest Intel and AMD chips by 10x for flops per watt!**
- **96 Processor Elements; 64-bit and 32-bit floating point… but only one control thread needed**
- **210-250 MHz… key to low power**
- **Intentionally "over-provisions" the floating-point hardware**
- **About 1 TB/sec internal bandwidth (high ratio to external bandwidth, OK for many kernels)**
- **128 million transistors**
- **Approximately 10 Watts**

CSX600 processor core

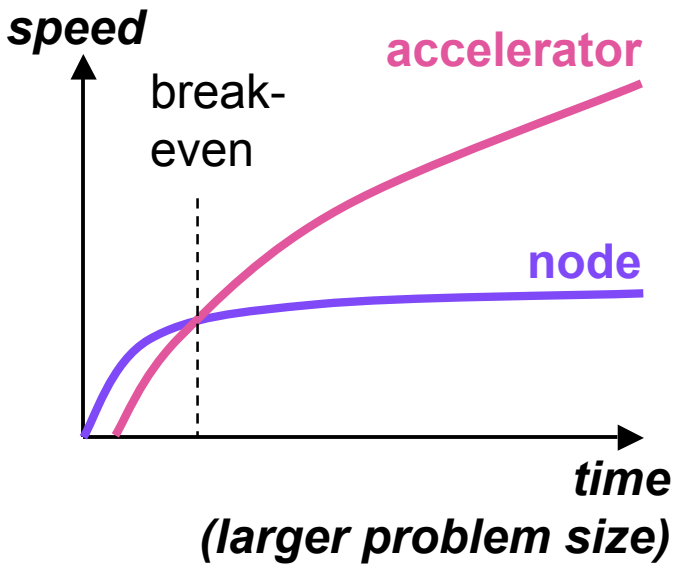


- Multi-Threaded Array Processing
  - Hardware multi-threading
  - Asynchronous, overlapped I/O
  - Run-time extensible instruction set
  - Bi-endian for compatibility
- Array of 96 Processor Elements (PEs)
  - Each is a Very Long Instruction Word (VLIW) core, not just an ALU

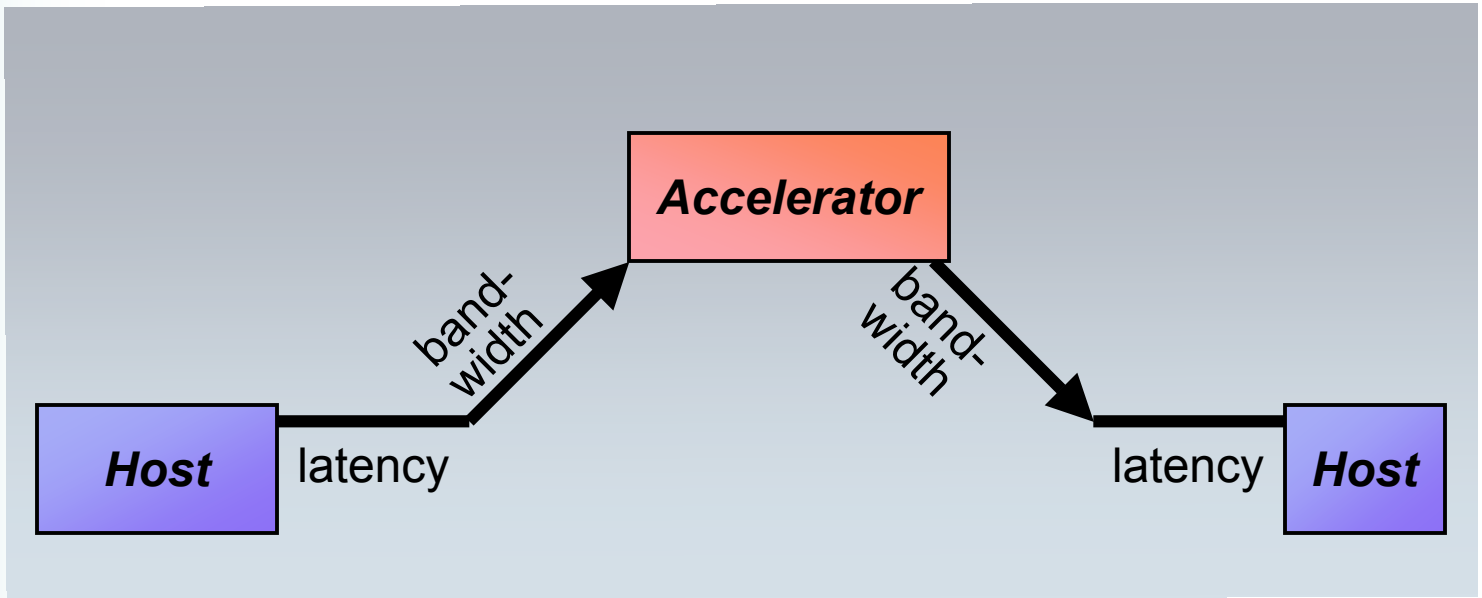
Is this trip necessary? Bandwidth issues



- **Acceleration software tests candidates for work on the board. If too small, it leaves them on the host.**
- **Performance claims *must* assume host-resident data. Beware of benchmarks that leave out the time to move the data to accelerator memory**

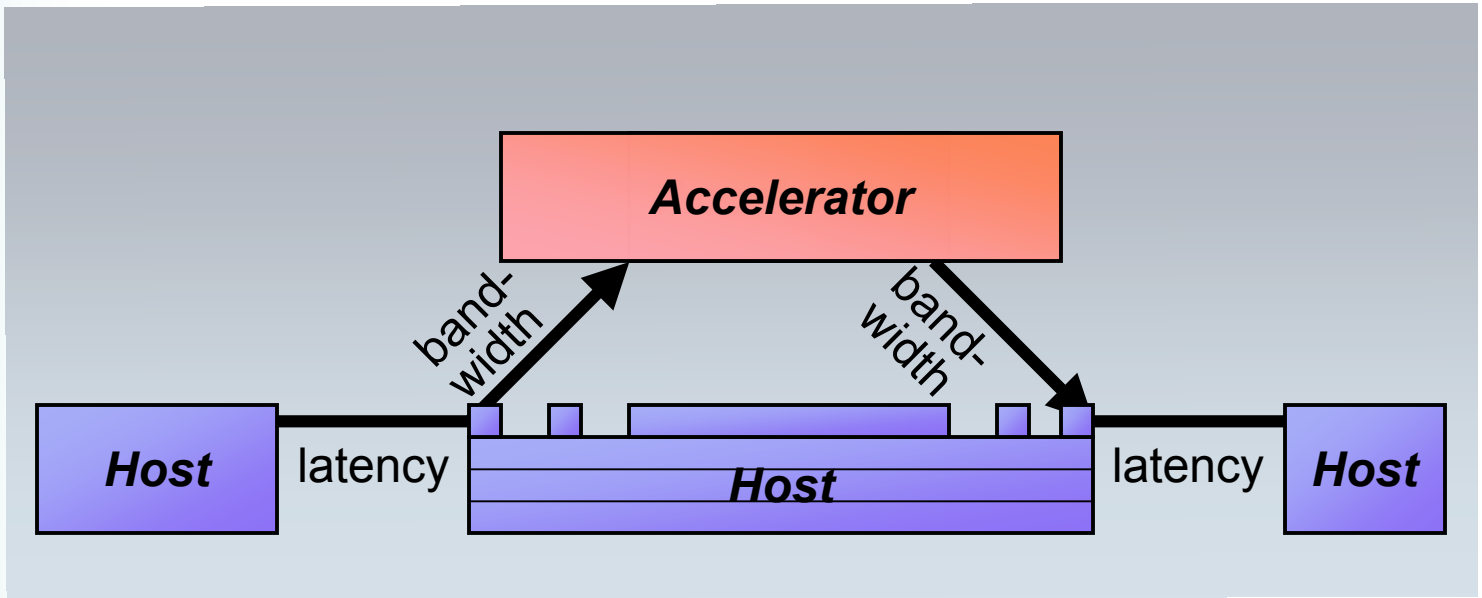


Simple offload model is out of date



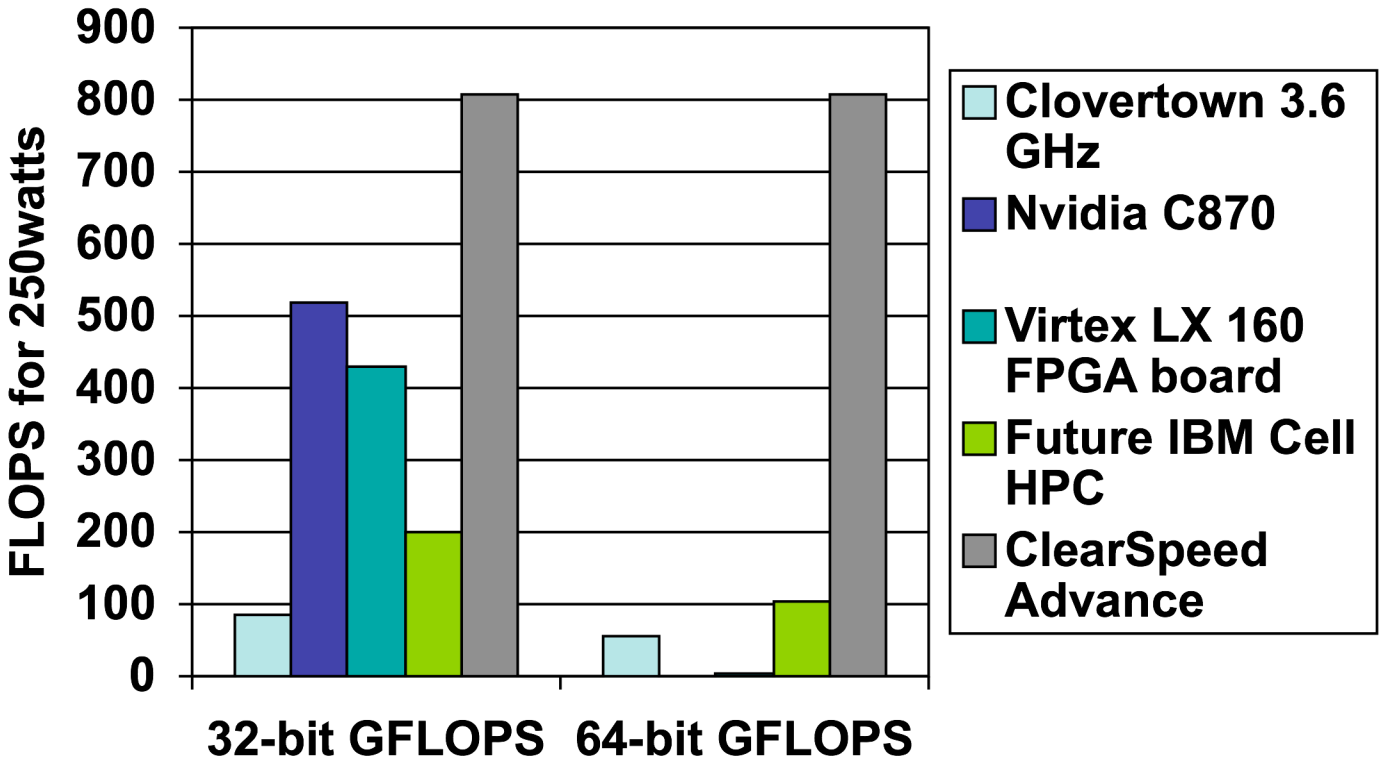
- **Accelerator must be quite fast for this approach to have benefit**
- **This "mental picture" may stem from early days of Intel 80x87, Motorola 6888x math coprocessors**

Card need not wait for *all* data before starting



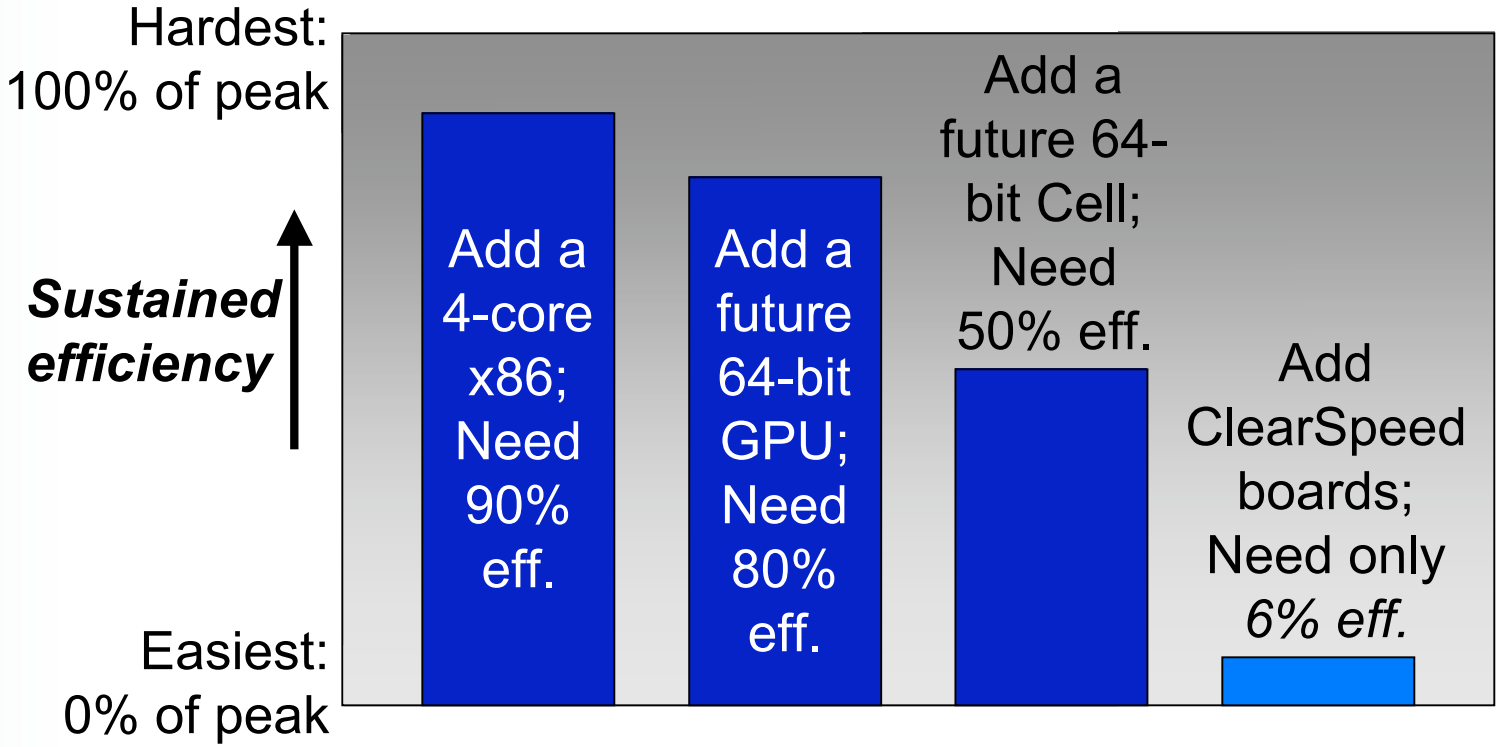
- **In practice, latency is *microseconds*; the accelerator task takes *seconds*. Latency gaps above would be microscopic if drawn to scale.**
- **The accelerator can be *slower* than the host, and still add performance!**

Ultra-low power approach delivers more GFLOPS



- ClearSpeed is highest performance for 32-bit & 64-bit GFLOPS within any power budget
- Lowest power ultimately allows smallest footprint

50 GFLOPS from 250 W is easier at *low efficiency*



ClearSpeed's power efficiency translates into ease-of-use by reducing optimization pressure on programmers

Accelerators have high-ratio memory hierarchies

Total: 1.0 GB

Total: 6.4 GB/s

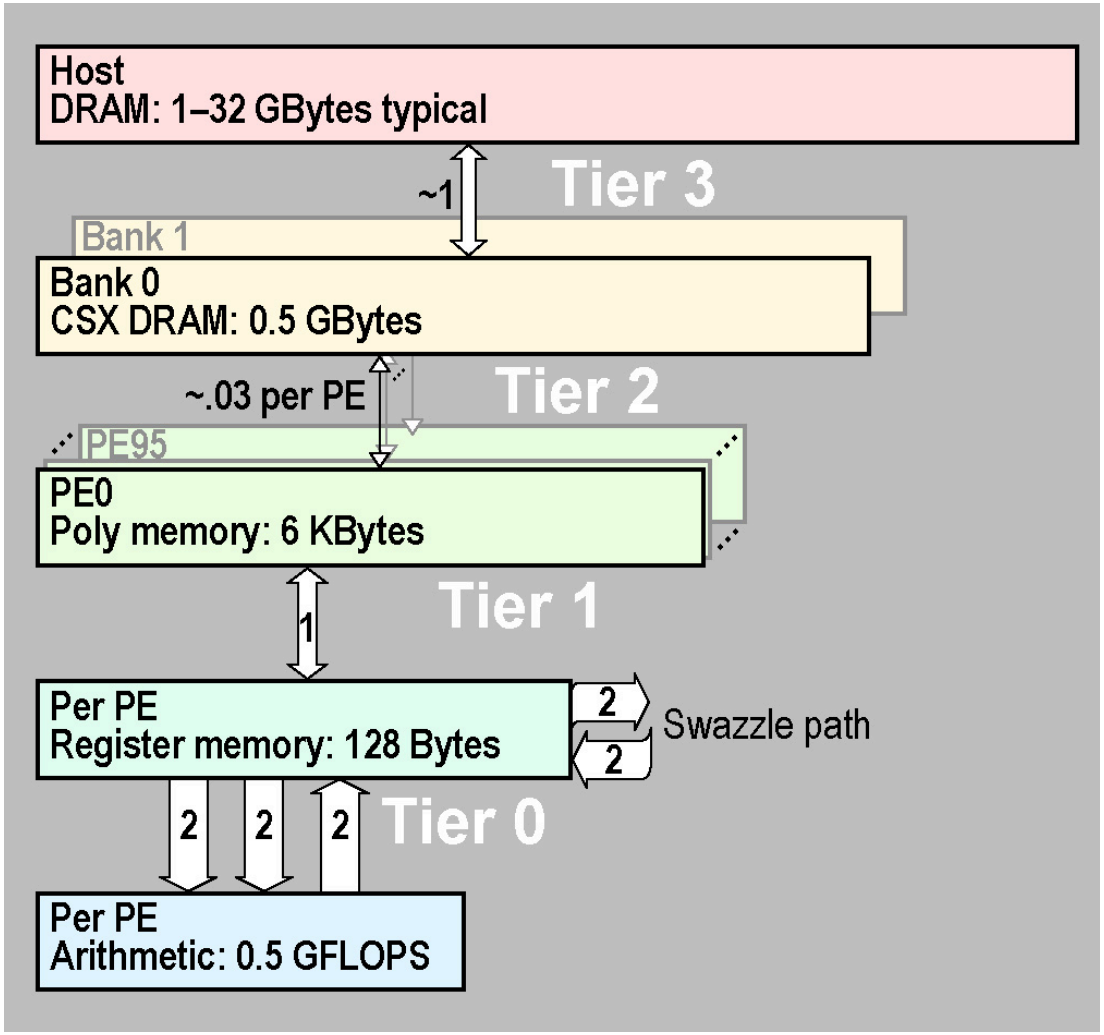
Total: 1.1 MB

Total: 192 GB/s

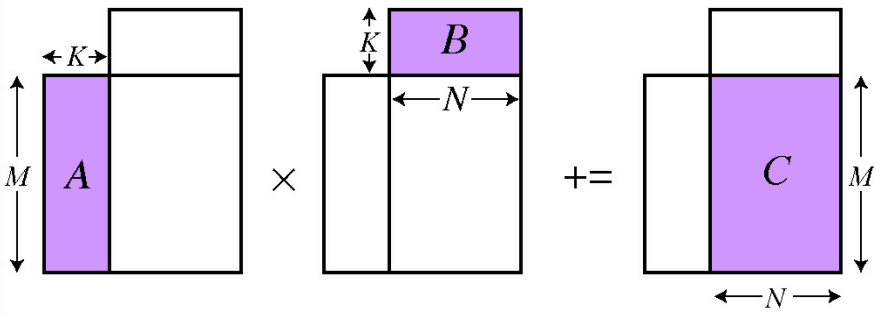
Total: 24 KB

Total: 2 TB/s

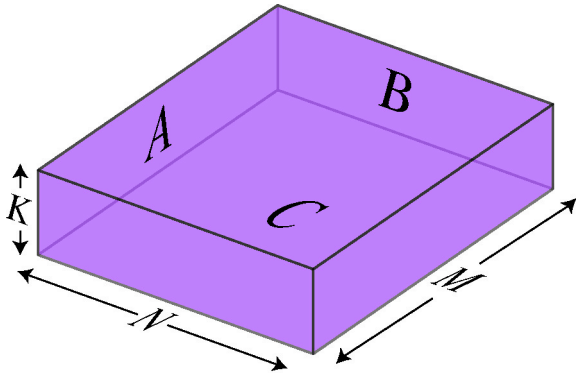
Total: 96 GFLOPS
(but only 25 watts)



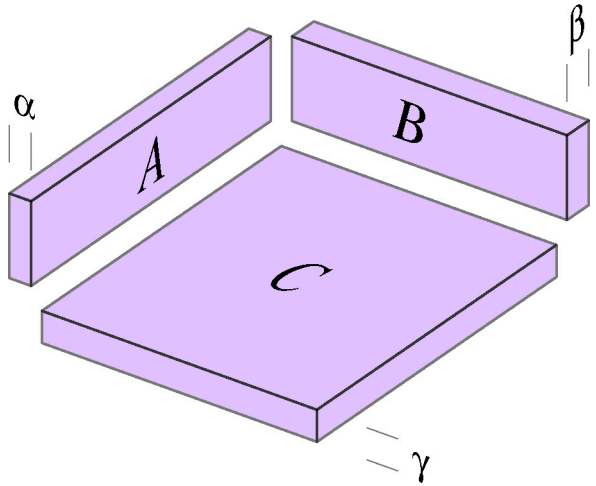
Visualization of algorithm overhead: DGEMM



Matrix multiply (DGEMM) is a perfect analog to a folded box.

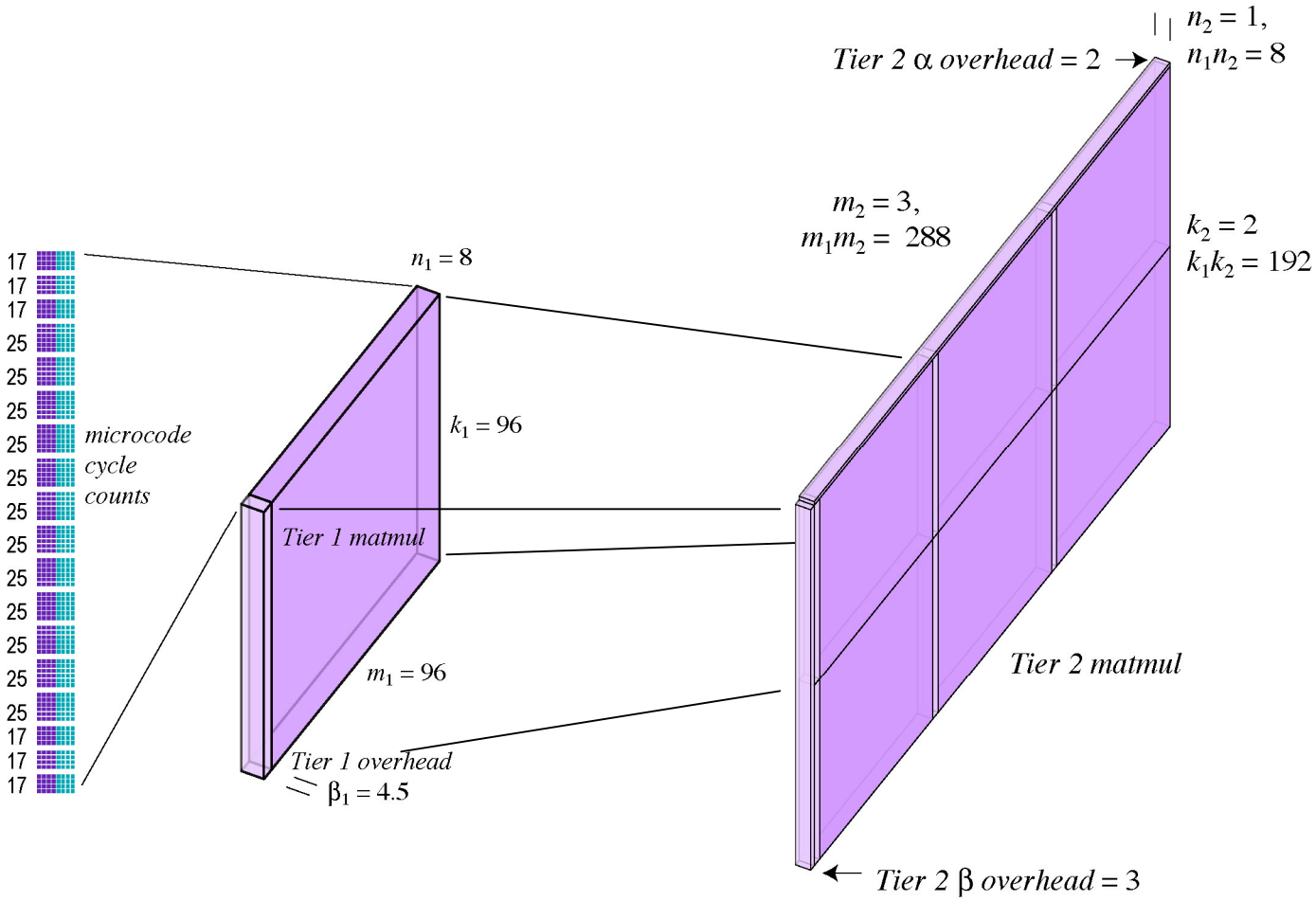


Volume is the number of multiply-adds.

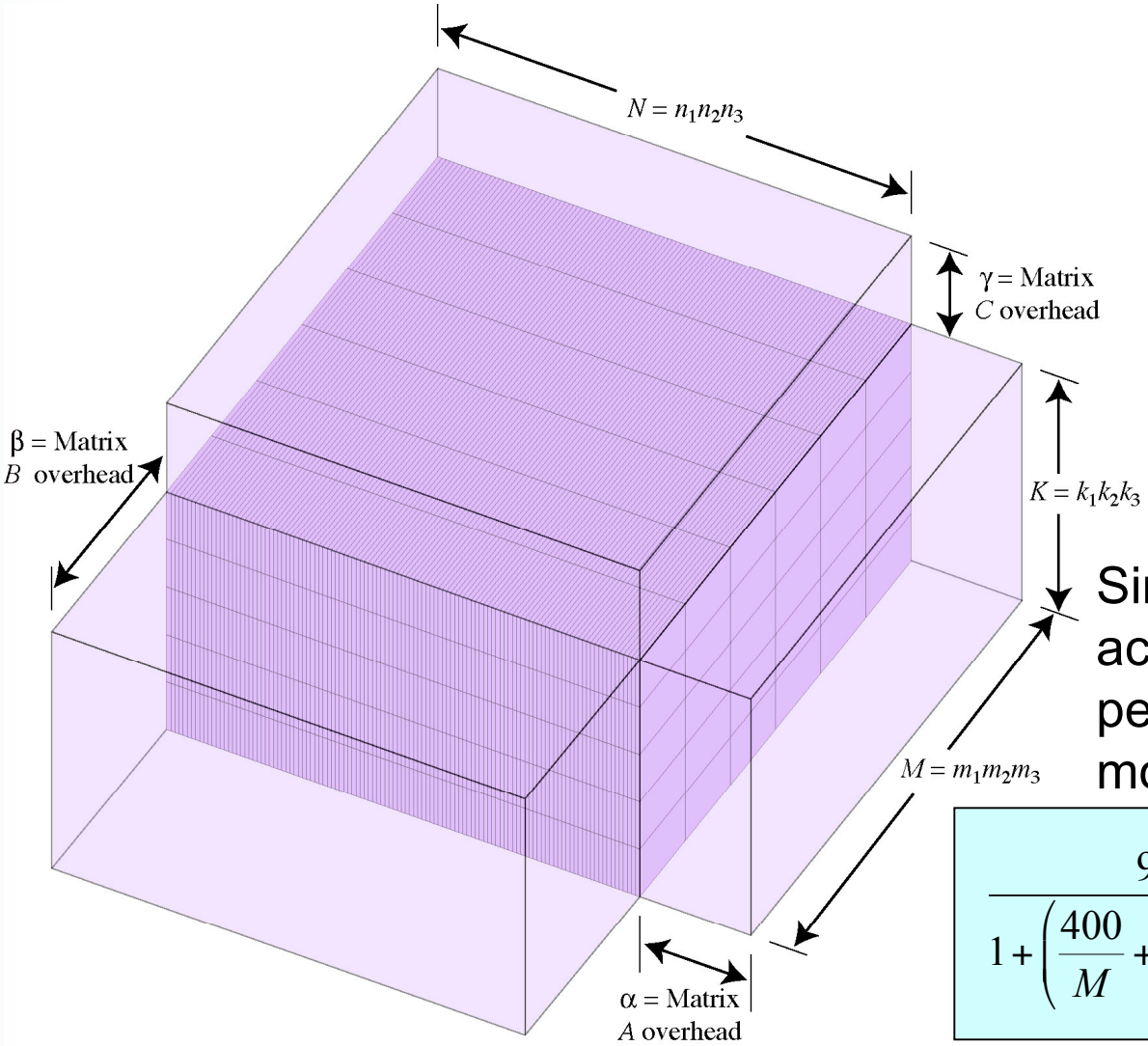


Surface “padding” shows overheads

Build up the bricks through the hierarchy



Result: high DGEMM speed from memory hierarchy



Simple but decent accelerator performance model:

$$\frac{90.2}{1+\left(\frac{400}{M}+\frac{200}{K}+\frac{200}{N}\right)} \text{ GFLOPS.}$$

Bulk is the Enemy of HPC

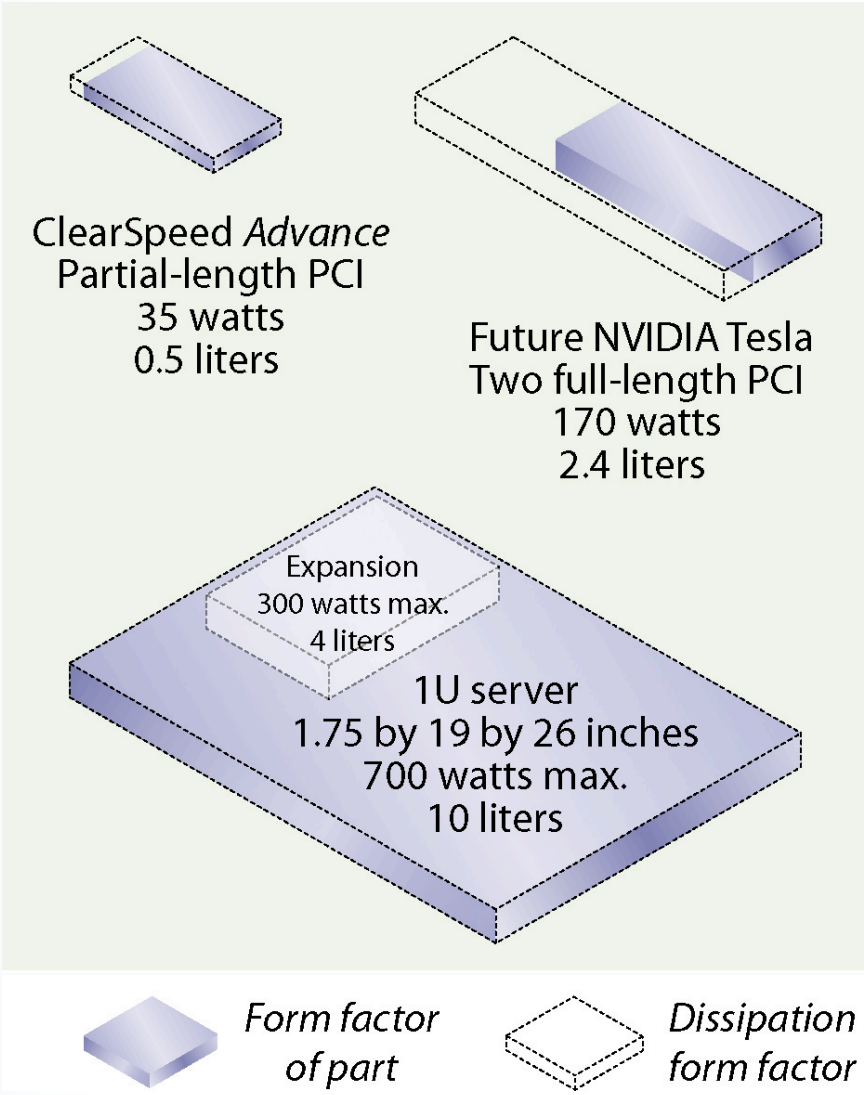
- **Increases latency**
- **Strains parallelism and the sharing of data**
- **Raises connectivity costs**
- **Can exceed distance limits, like InfiniBand**
- **Exceeds available floor space**

To reduce bulk, we must reduce *heat dissipation*. Performance per watt is the key to reducing the volume of computers.

Heat leads to *bulk*

- **Air cooling hits limits at about 70 watts/liter**
  - PCI standard of 25 watts, size is ***0.3 liters*** ✔
  - A 1U server might use 1000 watts, volume is ***14 liters*** ✔
  - A 42U standard rack might use 40 kilowatts, ***3000 liters*** ✔
- **Exceed 70 watts/liter, and temperatures rise above operational limits**

Dissipation volume can exceed actual volume



- **To find the *real* volume occupied by a component in liters, divide its wattage by 70**
- **What may seem like a dense, powerful solution might actually *dilute* the GFLOPS per liter because of heat generation.**

ClearSpeed™

What can we configure within a 250-watt budget?

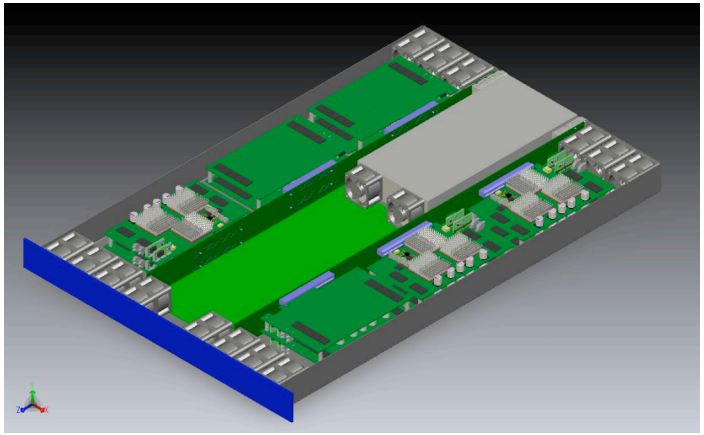
	Average Wattage	32-bit Peak GFLOPS	64-bit Peak GFLOPS
Intel Clovertown (3.6 GHz)	**250**	**86**	**57**
Nvidia C870	**170**	**518**	not supported
Future Nvidia 64-bit	unknown	unknown	1/8th SP performance
10 FPGA PCI cards virtex LX160 based	**250**	**430**	**4.2**
Cell BE	**210**	**230**	**15**
Future Cell HPC	**220**	**200**	**104**
10 ClearSpeed Advance™ Boards	**250**	**806**	**806**

Even ClearSpeed's 2005 product has not been matched by existing or announced alternatives.

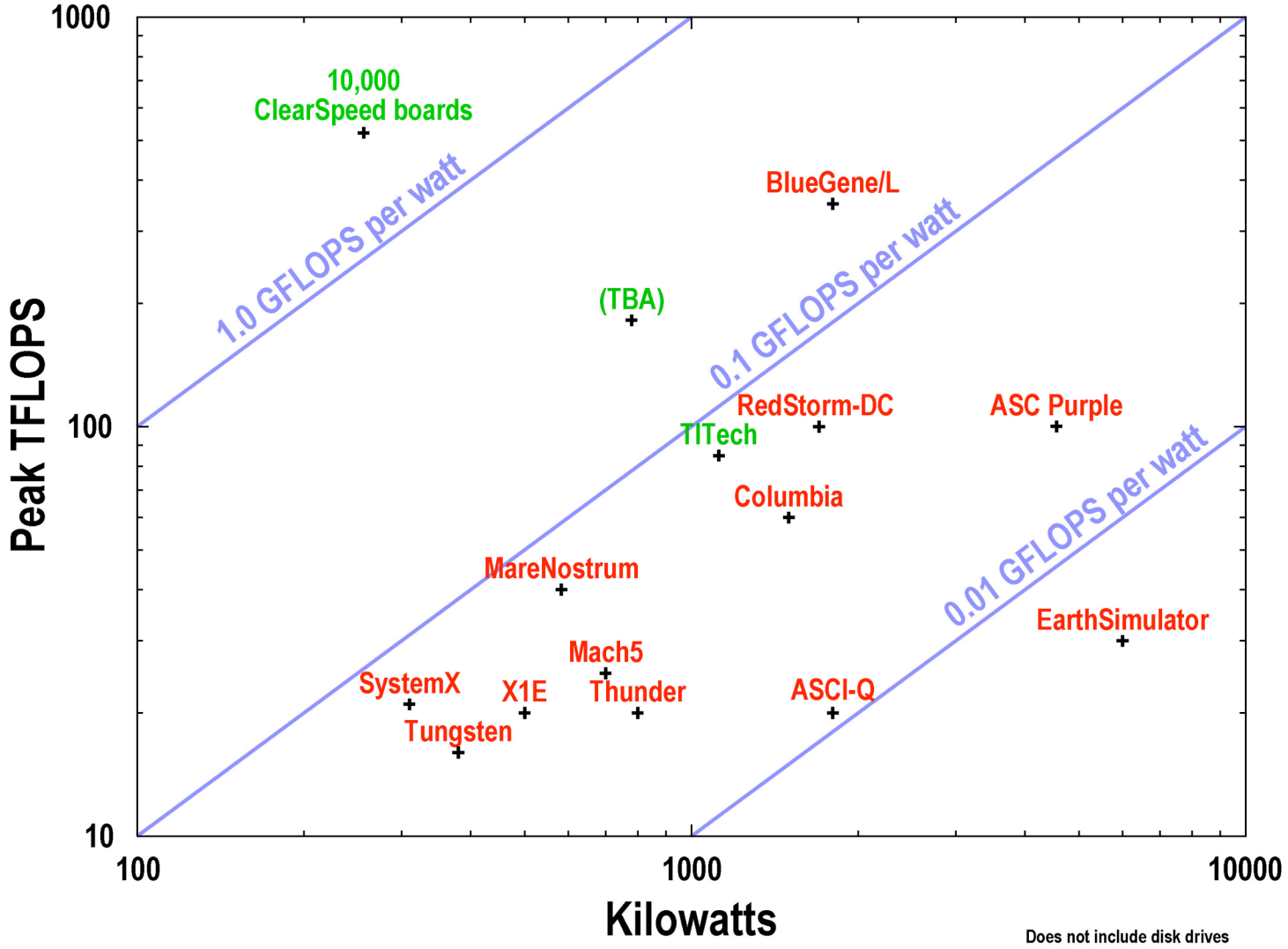
New Design Approach Delivers 1 TFLOP in 1u

- **1u standard server**
- **Intel 5365 3.0GHz**
  - 2-socket, quad core
  - 0.096 DP TFLOPS peak
  - Approx. 650 watts
  - Approx. 3.5 TFLOPS peak in a 25 kW rack

- **ClearSpeed Acceleration Server Concept**
  - 24 CSX600 hectacore processors
  - ~1 DP TFLOPS peak
  - Approx. 500 watts
  - Approx. 19 TFLOPS peak in a 25 kW rack
  - 18 standard servers & 18 acceleration servers

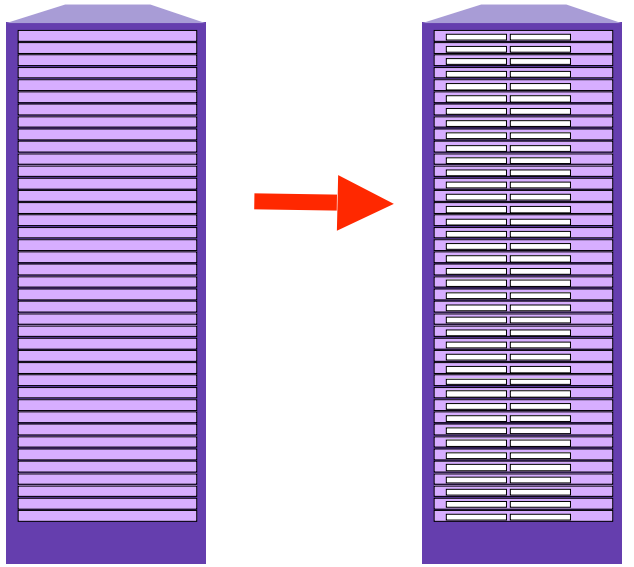


GFLOPS per watt for some capability systems



Hypothetical 1U-based cabinet

- 40 servers with 2.66 GHz x86 quad-core
- 0.64 TB DRAM
- 3.4 TFLOPS peak
- ~2.8 TFLOPS LINPACK (82% eff.)
- 24 kW
- 10 sq. ft.
- 800 pounds
- ~$400,000 with IB



- Add 80 ClearSpeed Advance cards
- 0.80 TB DRAM
- 11 TFLOPS peak
- ~7 TFLOPS LINPACK (64% eff.)
- 26 kW
- 10 sq. ft.
- 850 pounds
- < $1,000,000

ClearSpeed increases…

- **Power draw by 8%**
- **Floor space by 0%**
- **Weight by 6%**
- **Speed by 150%**

Single-cabinet TOP500 supercomputer

Uses for ClearSpeed *after* the TOP500 press release…

Dense matrix-matrix kernels: order N^3 ops on order N^2 data

Boundary element and Green's function methods

Gaussian, NAB, other chemistry codes use DGEMM intensively

***N*-body interactions: order N^2 ops on order N data**

Astrophysics, low-density CFD, molecular mechanics

Look to MD-GRAPE for examples

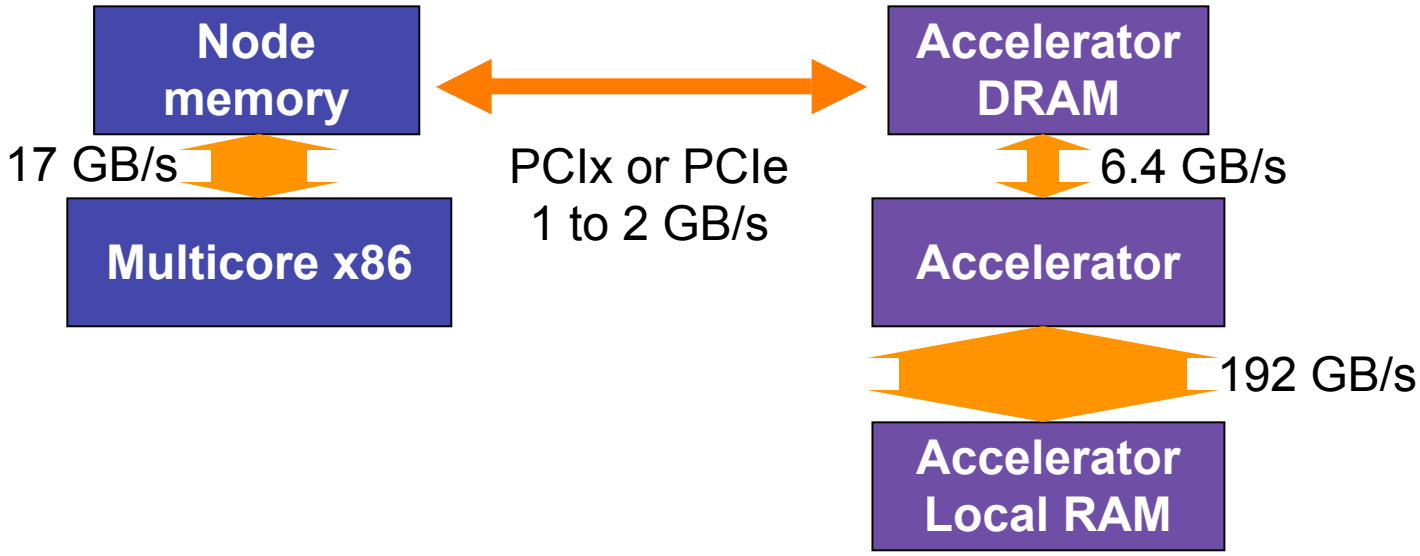
Some sparse matrix operations: order NB^2 ops on order NB data where B is the average matrix band size

Structural analysis, implicit PDEs generally

Time-space marching: order N^4 ops on order N^3 data

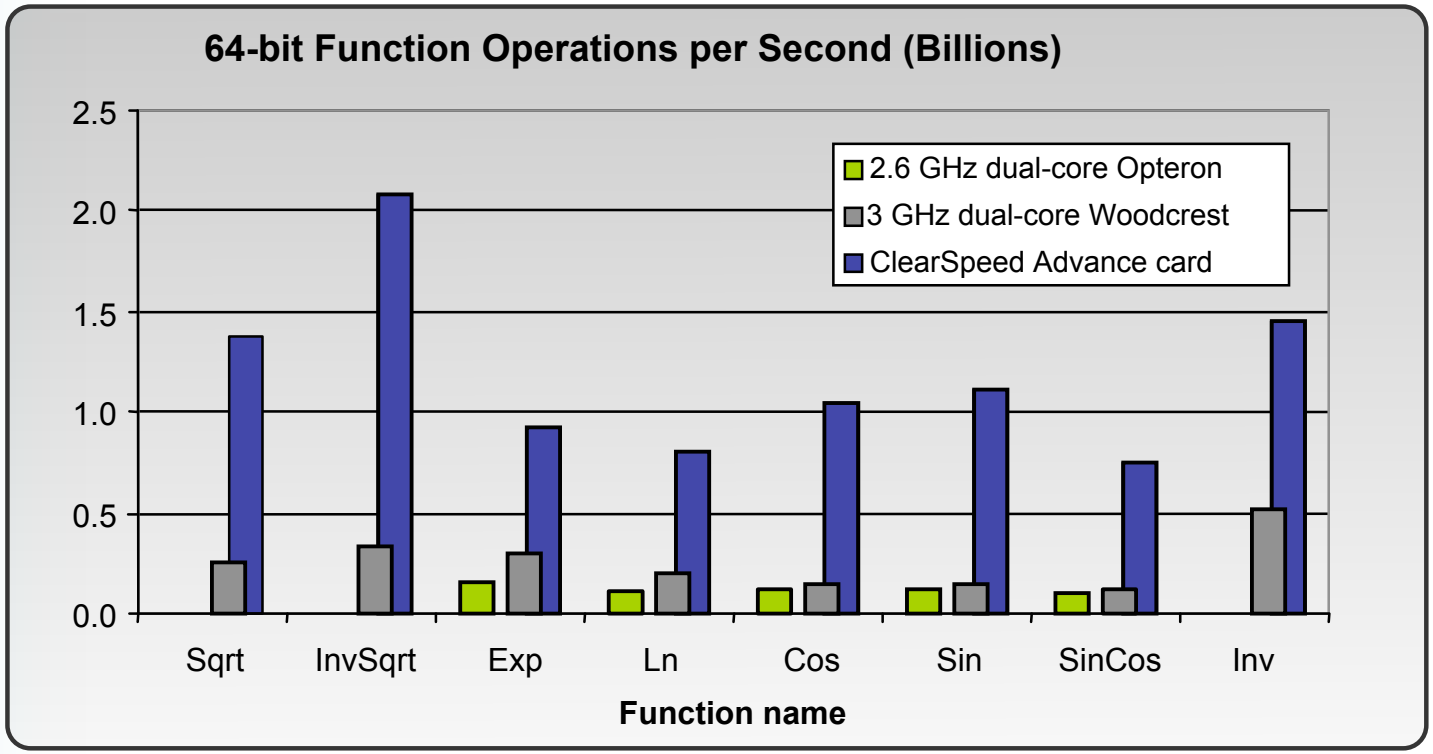
Explicit finite difference methods; data must reside on card

Memory bandwidth dominates performance model



- **Apps that can stage into local RAM (Tier 1) can go 10x faster than current high-end Intel, AMD hosts**
- **Apps that must reside in DRAM (Tier 2) will actually run *slower* by about 3x (for fully optimized host code)**
- **Fast Fourier Transforms can go either way!**

Math functions reside at Tier 1, hence fast

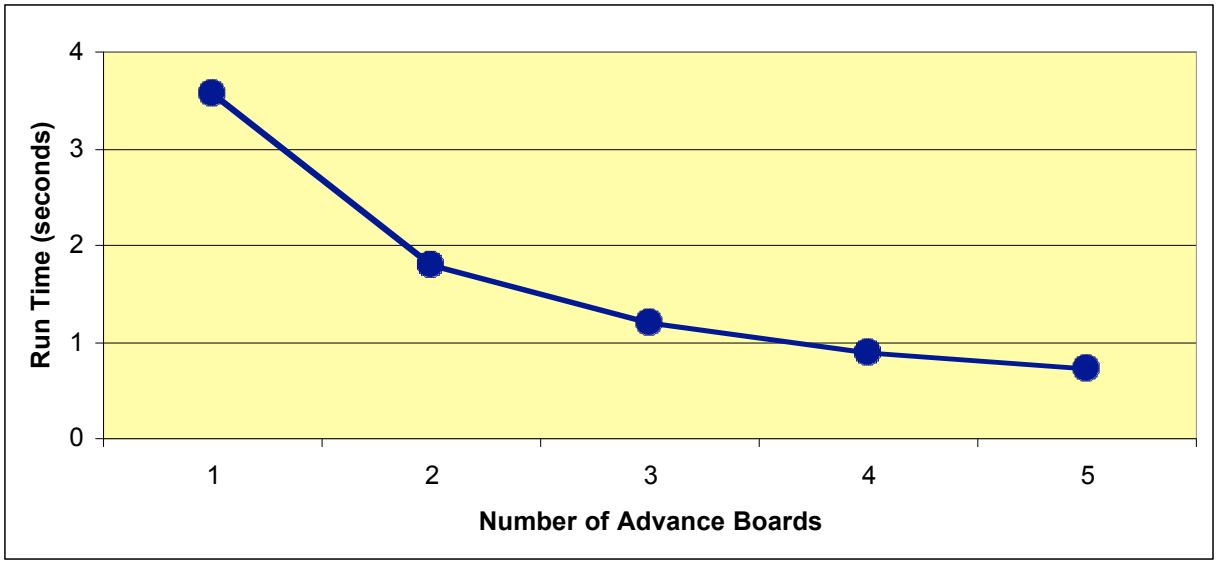


Typical speedup of ~8X over the fastest x86 processors, because math functions stay in the local memory on the card

Monte Carlo PDE methods exploit Tier 1 bandwidth

Real apps do work resembling "EP" of NAS Parallel Benchmarks. "Quants" solve PDEs this way for options pricing, Black-Scholes model (a form of the Heat Equation)

- **No acceleration: 200M samples, 79 seconds**
- **1 accelerator: 200M samples, 3.6 seconds**
- **5 accelerators: 200M samples, 0.7 seconds**

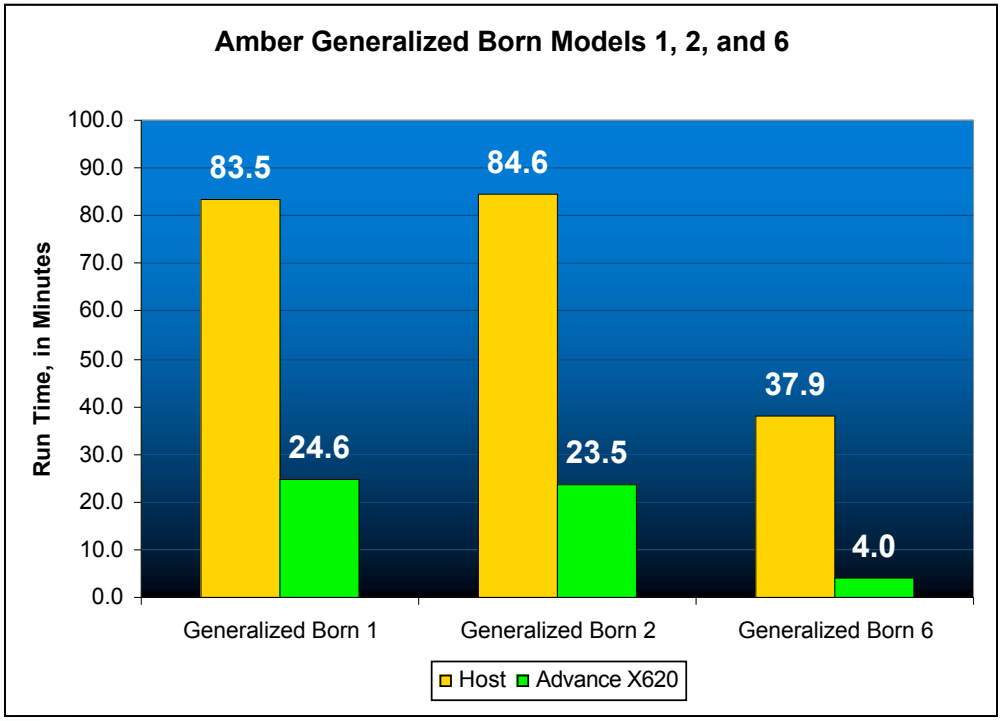


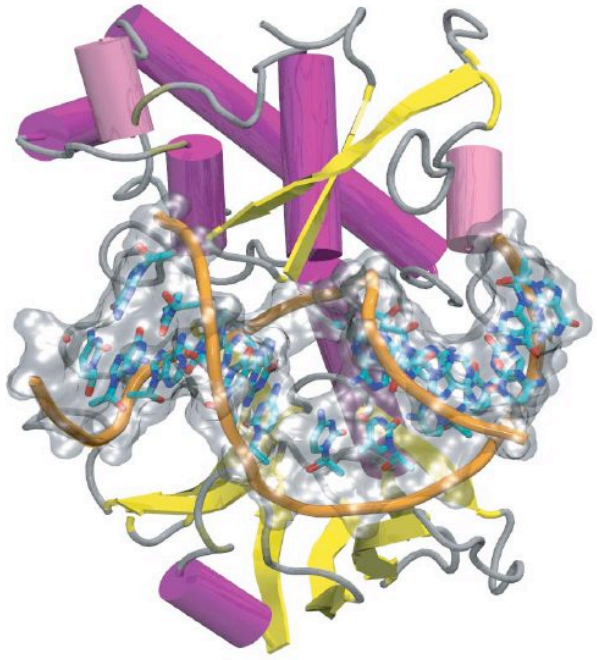
Accelerating Amber 9

- **Accelerates *70%* of the runs submitted to the Tokyo Tech system; real benefit for real applications.**
- **1% of Source Code; 95-97% of CPU compute time**
- **Order *N* data motion, order N^2 floating-point operations**
- **Supported Options**
  - Generalized Born (GB) Models: 1, 2, & 6
  - Constant pH
  - Analytical Linearized Poisson Boltzmann (ALPB)
  - Options that do not directly change the force calculation, including NMR restraints
- **Delivered via a ClearSpeed modified patch to Amber 9**
- **Downloadable from: http://amber.scripps.edu/**
  - Customer must have valid Amber 9 license

AMBER Molecular Modeling with ClearSpeed

AMBER module	Host	Advance X620	1-board Speedup
Gen. Born 1:	83.5 min.	24.6 min.,	3.4x speedup
Gen. Born 2 :	84.6 min.	23.5 min.,	3.6x speedup
Gen. Born 6 :	37.9 min.	4.0 min.,	9.4x speedup



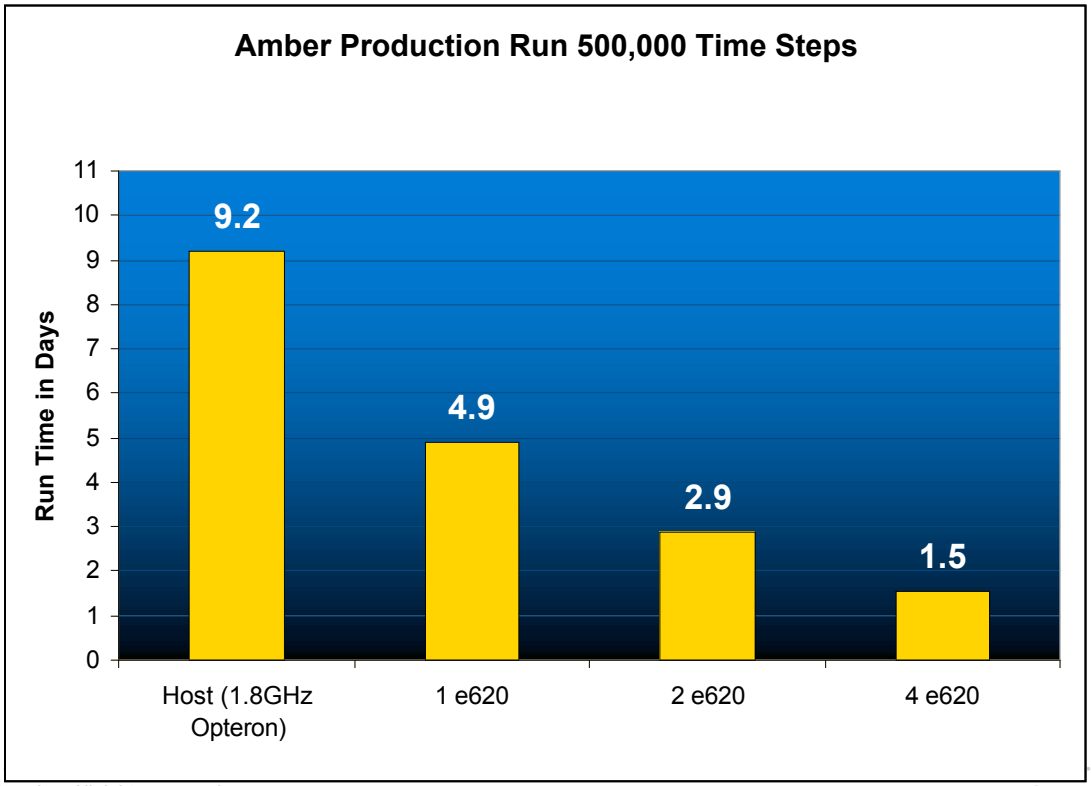


AMBER Run from Major Pharmaceutical Company

- **Generalized Born (GB) Model: 1**
- **Just under 3000 Atoms**
- **Three runs**
  - Heating Run (40,000 time steps): 18 hours, 2 minutes
  - Production Run (5000 time steps): 2 hours, 15 minutes
  - Full Production Run (500,000 time steps): approx. 9 days
- **Platforms**
  - Reference platform: Opteron 1.8 GHz
  - ClearSpeed with various host platforms and operating systems
  - Accelerated run times appear independent of host and OS
  - 1, 2 and 4 Advance board systems: Performance scales almost linearly…

AMBER Test Case from Major Pharmaceutical Company-cont.

System	Time in Days	Speedup
1.8 GHz Opteron host	9.2	1.00x
1 Advance e620	4.9	1.87x
2 Advance e620	2.9	3.20x
4 Advance e620	1.5	6.02x



Exploiting *ij* symmetry might increase performance another 1.5x to 1.9x!

NAB and AMBER 10 acceleration

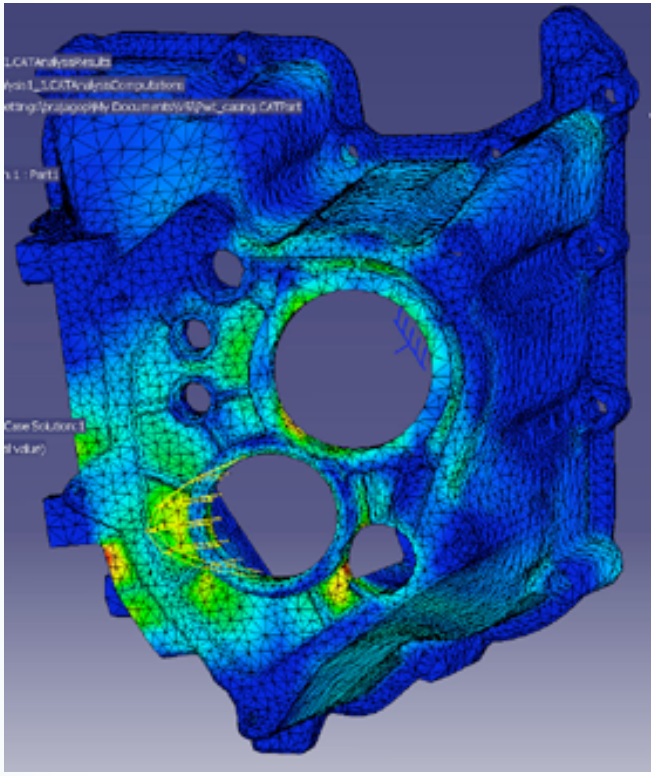
- Newton-Raphson refinement now possible; analytically-computed second derivatives
- 2.6x speedup obtained for this operation in three hours of effort (no source code changes)
- Enables accurate computation of entropy and Gibbs free energy for first time.
- Available now in NAB (Nucleic Acid Builder) code. Slated for addition to AMBER 10.

Plug-and-play *ab initio* chemistry acceleration?

- DGEMM content is 18% to 65% in GAUSSIAN test suite, but typical sizes only ~10 to 100.
- Sample GAUSSIAN tests to date are *too small* to accelerate with host-resident DGEMM calls; below N = 576 threshold.
- PARATEC, Qbox much better candidates. Plane wave models are over half DGEMM, huge dimensions.
- Molpro example shows the right approach: make DGEMM calls card-side, not host-side. Good speedups then obtained over x86 multicore.

The economics of CAE acceleration

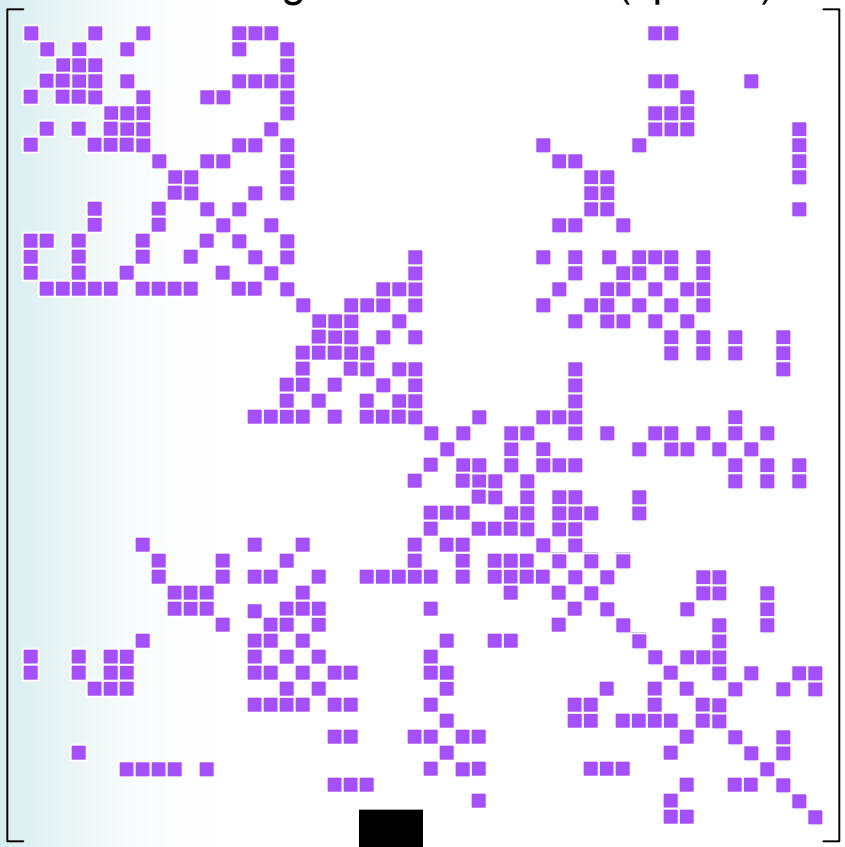
Structural Analysis



- **Each host costs $3,000.**
- **Software license costs ~$30,000 *per core*, which discourages use of multiple cores.**
- **MCAE engineer costs over $200,000/year.**
- **In California, anyway.**
- **Accelerator card would be cost-effective even with a 7% performance boost. Actual performance boost should be more like 260% for large problems.**

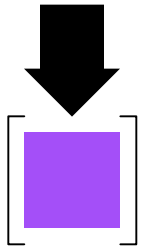
Accelerating Finite Element Method (FEM) solvers

10 million degrees of freedom (sparse)

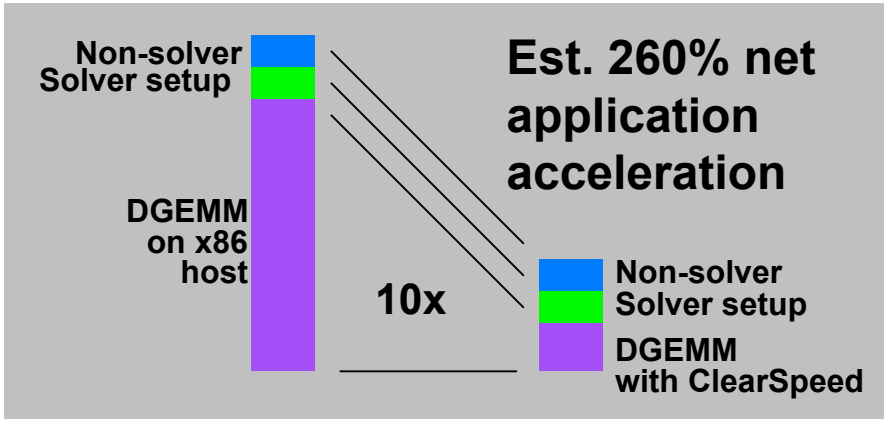


becomes…

50,000 *dense* equivalent

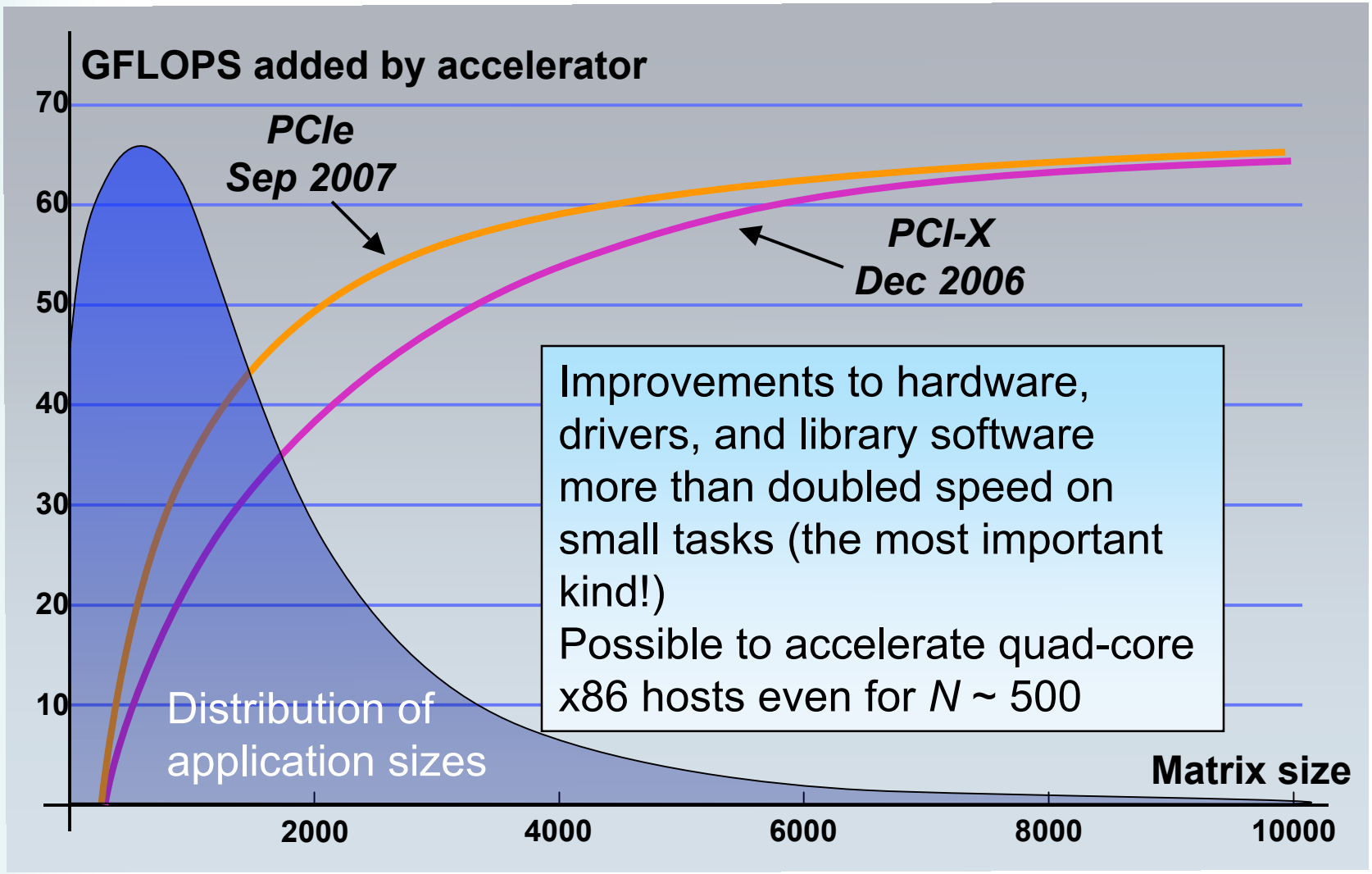


Accelerator can solve at over 50 GFLOPS



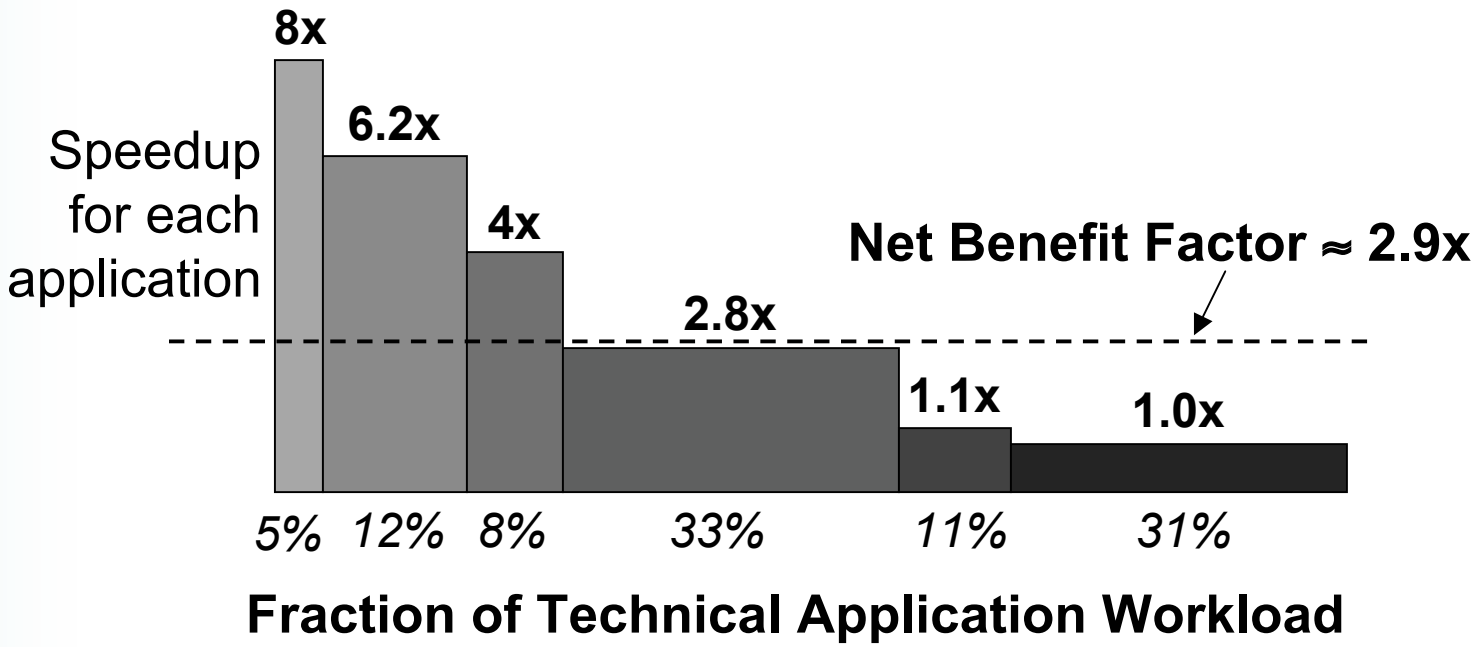
- **Potentially pure plug-and-play**
- **No added license fee**
- **Needs ClearSpeed's 64-bit precision and speed**
- **Enabled by recent DGEMM improvements; still needs symmetric A^TA variant**
- **Could enable some CFD acceleration (for codes based on finite elements, low Reynolds numbers`)**

Matmul speed changes broaden application space



Note: curve only samples integer multiples of vector size

Assume a *range* of accelerator applicability



- **While high-speedup applications motivate use of accelerators, they then have a broad benefit for other uses with only modest speedup.**
- **In many cases, the job mix will result in a net performance/price benefit when total cost of ownership is taken into account.**

Summary

- **Computing accelerators solve limitations of simply piling up x86 servers**
  - 10x more performance per watt
  - 10x more performance per unit volume
  - Fewer, more powerful cores reduce pressure to scale to huge numbers of threads of control
  - Lets us evolve gradually to better architectures instead of requiring a total rewrite of application codes
- **So accelerators and hybrid architectures are once again "the latest thing." And with good reason. ■**