

eHiTS Lightning: Docking performance accelerated 30-50 fold on the Cell/BE processor



Zsolt Zsoldos
SimBioSys Inc.

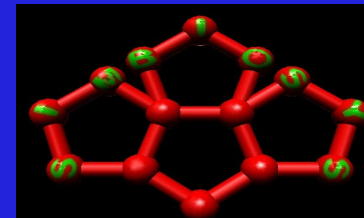


<http://www.simbiosys.ca/>

Booth #316

Contents:

- Overview of the eHiTS docking software and its accuracy results
- Hardware acceleration technology overview and comparisons
- Cell/BE programming challenges and tricks
- Speed-up results, performance achievements



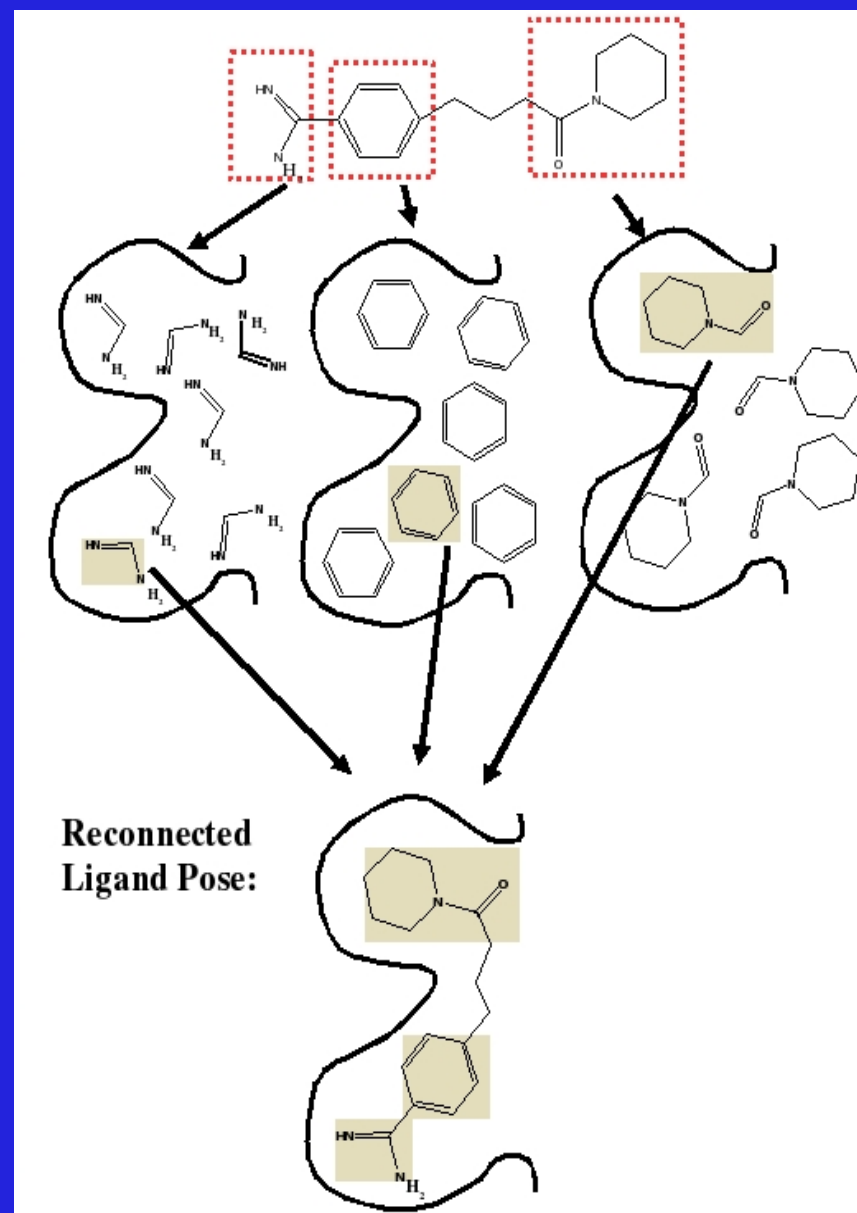
2. Overview of eHiTS

- Ligand is divided into rigid fragments, flexible chains
- All rigid fragments are docked **independently** (many poses)
- Pose matching (clique detection)
- Flexible chain fitting (continuous)
- Local energy minimization

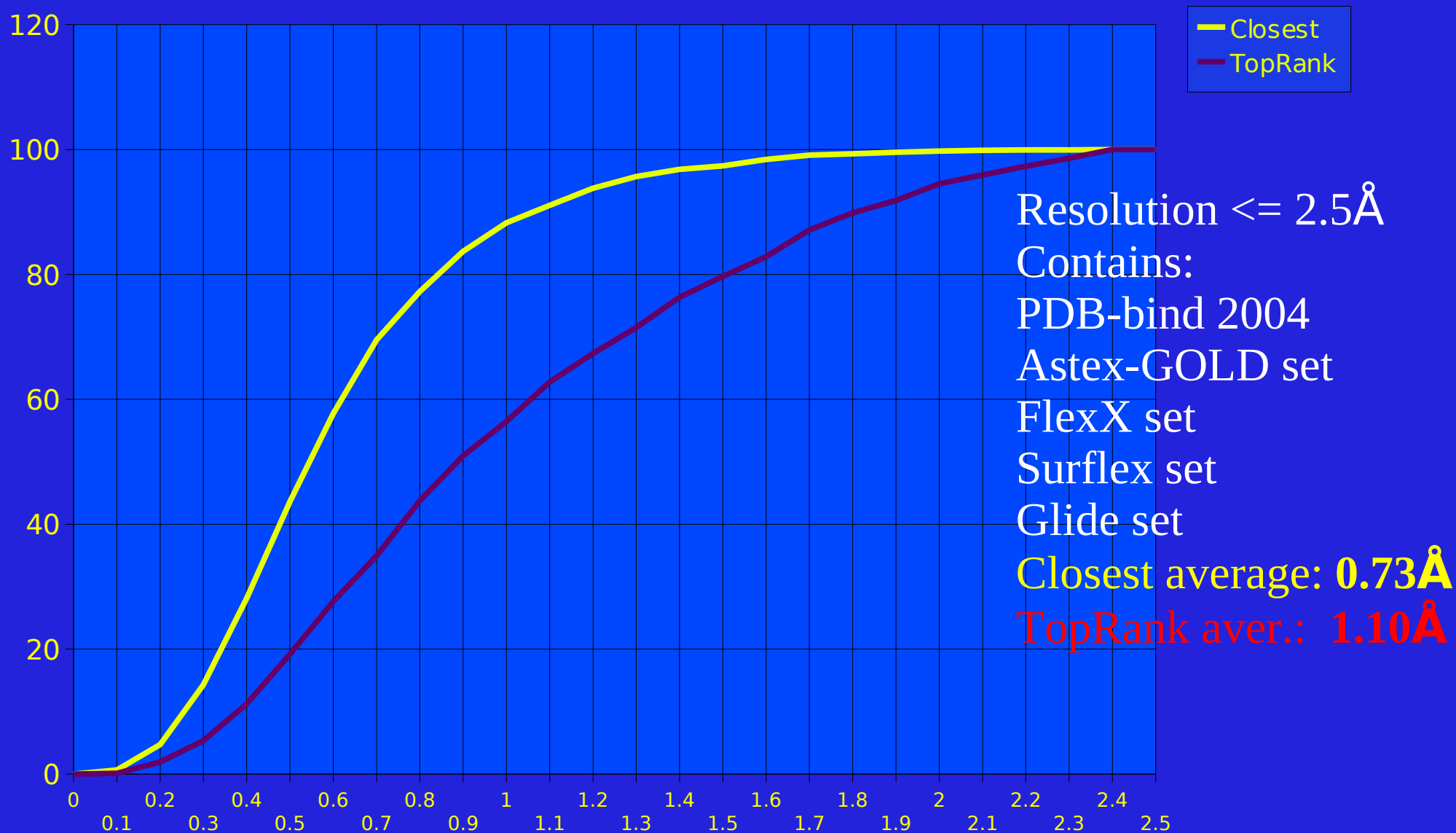
J.MGM (26) #1, July 2007, pp 198-212

doi: 10.1007/s10822-007-9164-5

Tuesday, 24th, 10:10 AM, Room: 254 A



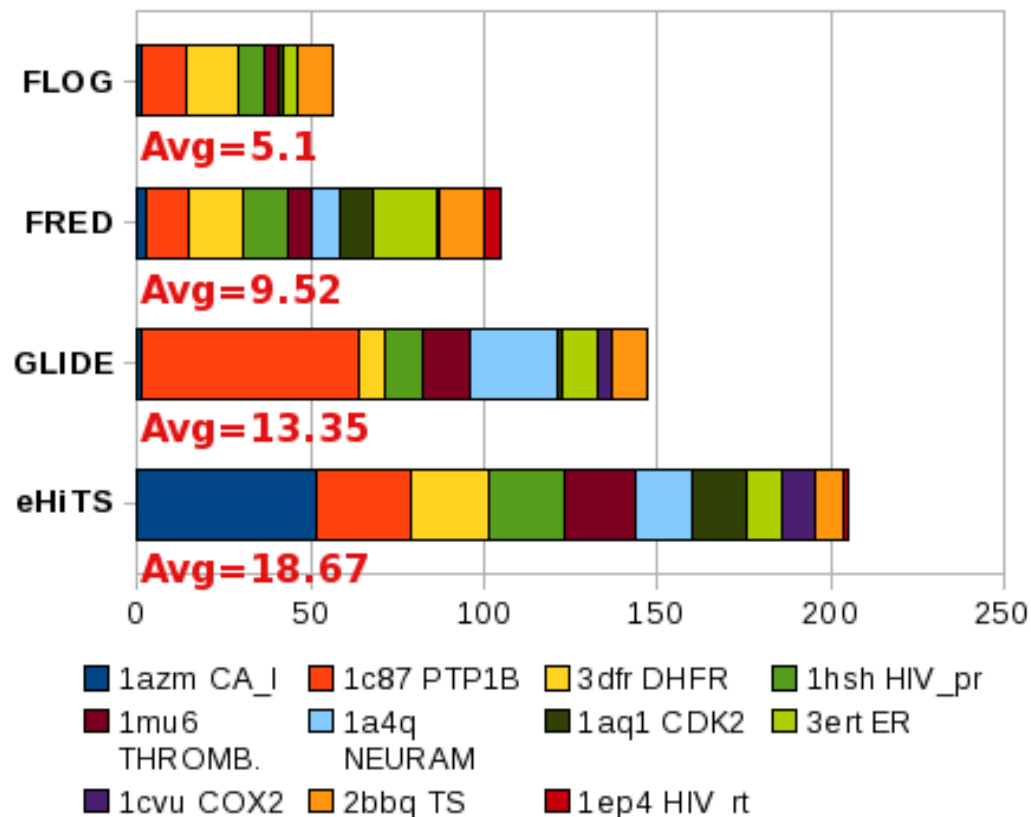
3. Cognate pose accuracy on 1568 PDB complex with drug-like ligands



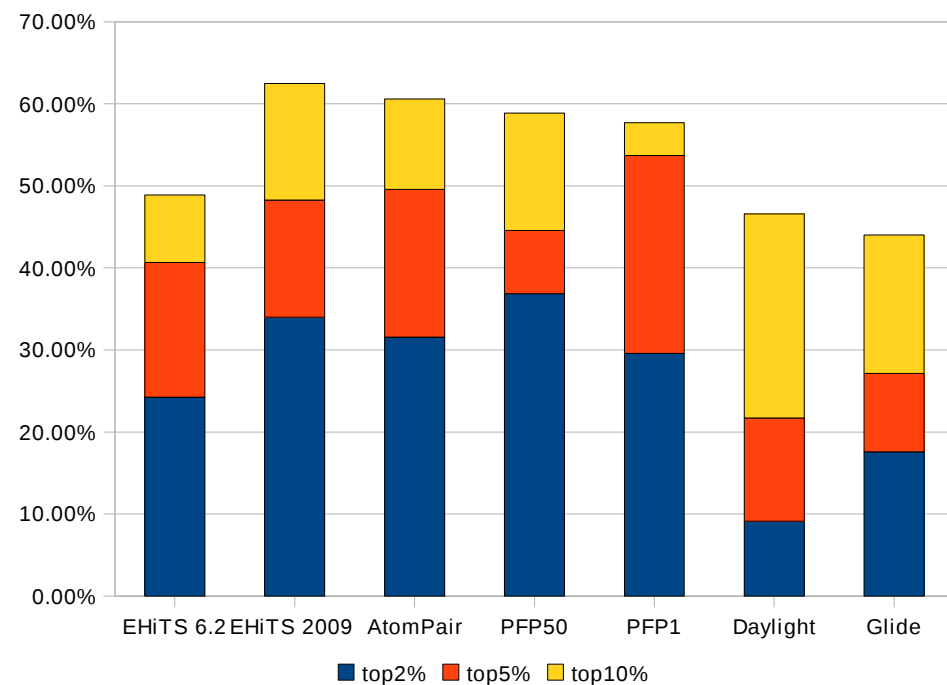
4. Enrichment results

Merck's comparison of 4 dockers on 11 test cases

Enrichment results with: Flog, Fred, Glide and eHiTS



eHiTS, Glide and I.Muegge's fingerprint methods on the BI dataset



J. Chem. Inf. Model. 2007; 47(4), pp 1504 - 19 DOI: 10.1021/ci700052x

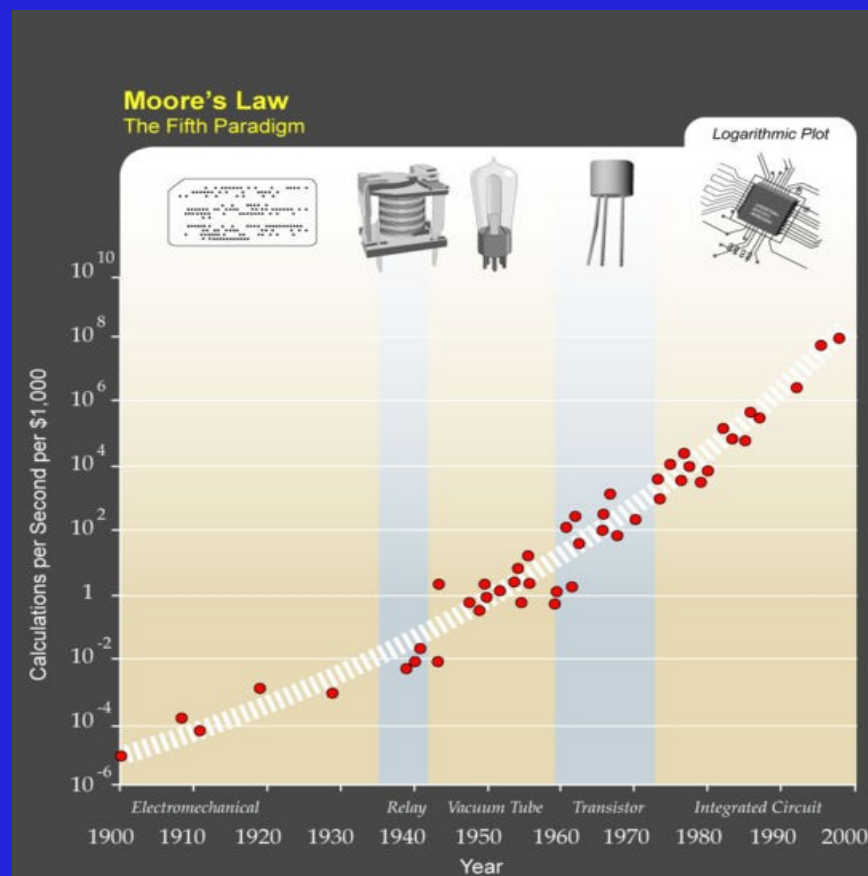
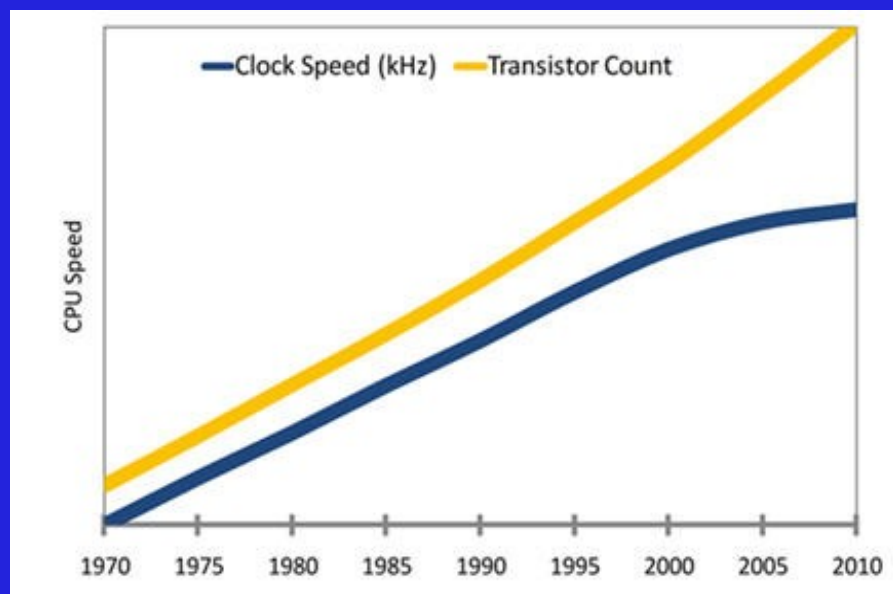
More results in scoring talk: Wednesday 10am Room 254A

5. The “Need for Speed”

- Problem complexity:
 - conformational search space has to be sampled at 0.5Å or finer for any scoring function
 - Translations(0.5Å) * Rotations(5°) * Dihedrals(5°) = $20^3 * 72^3 * 72^n \sim n=6 \text{ rot. bonds} \Rightarrow 2*10^{20}$ poses *per ligand*
- Candidate library size
 - ZINC ligand DB: 8.6 million compounds
 - Corporate DB typically in the range of 5-15 million ligands
- Competition with experimental HTS
- eHiTS: good accuracy and enrichment, but needs to be faster
- How to speed-up the exhaustive, accurate docking ?

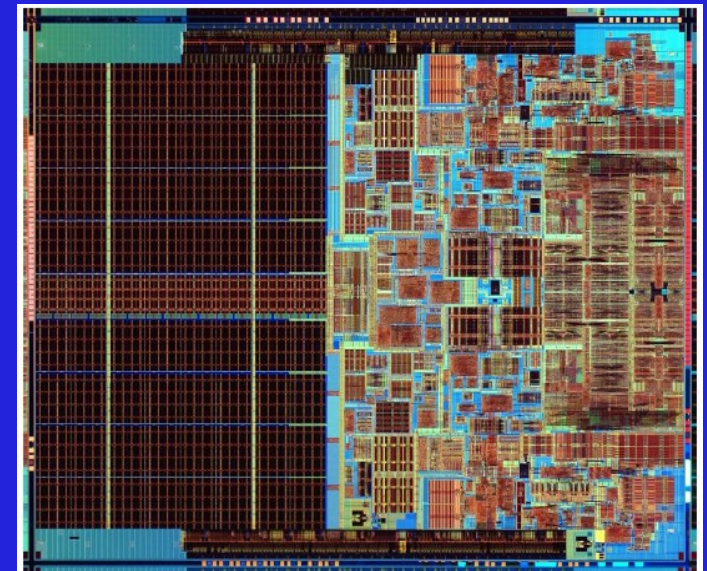
6. Hardware Architecture Evolution:

- **Moore's law:** number of transistors per area doubles every 18 months
- **Effect during 1970-2003:** CPU clock speed doubled in 18 months
- **In the last 5 years** clock speed stuck at ~3GHz (perf. walls)
- **Paradigm shift to parallel computing:** data or code
- **Data:** SIMD approach (GPU)
- **Code:** increase number of cores
- **Both:** Cell/B.E. architecture



7. Performance walls for speed-up

- **Power Wall:**
 - Quadratic increase in power consumption and heat generation
 - Diminishing returns
- **Frequency Wall:**
 - Processing pipes getting longer
 - Branch prediction hardware and cost of miss
- **Memory Wall:**
 - I/O speed falling behind
 - More and more on-die cache
 - Multi-level cache complexity
 - Increased latency
 - Stalling due to cache-miss



8. HW Acceleration Technologies

- **Field Programming Gate Arrays (FPGA)**

- hardware engineering,
- ideal for integer arithmetic



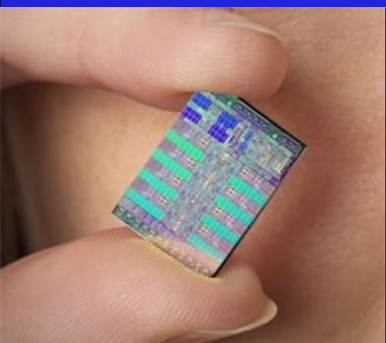
- **Graphics Processing Units (GPU)**

- ideal for massive independent parallelism (~graphics)
- multi-layer memory cache system
- single (or lower, e.g. 14Bit) precision floating point
- many-thread programming model



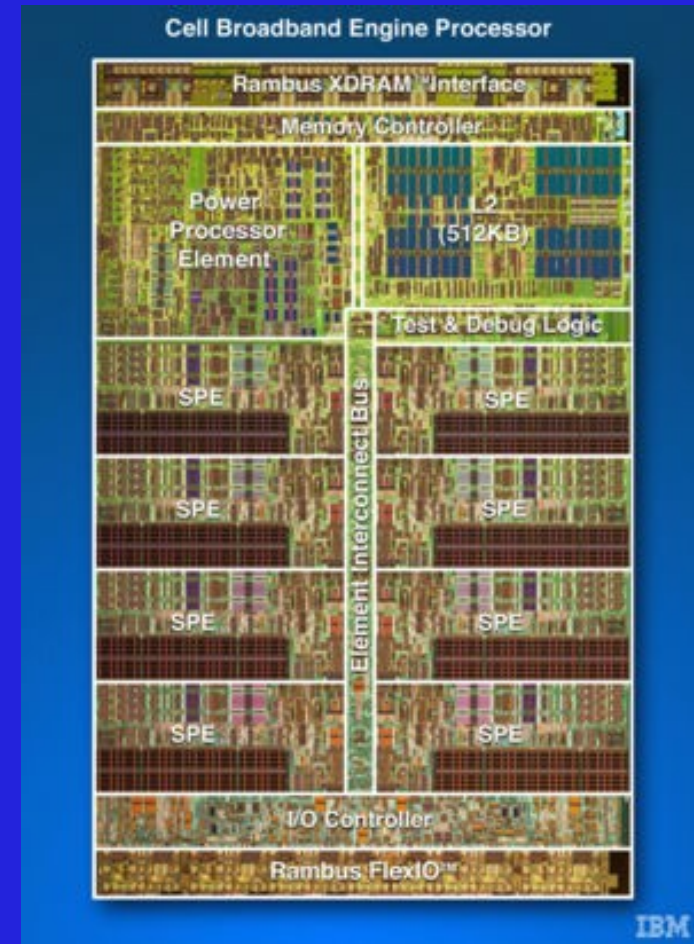
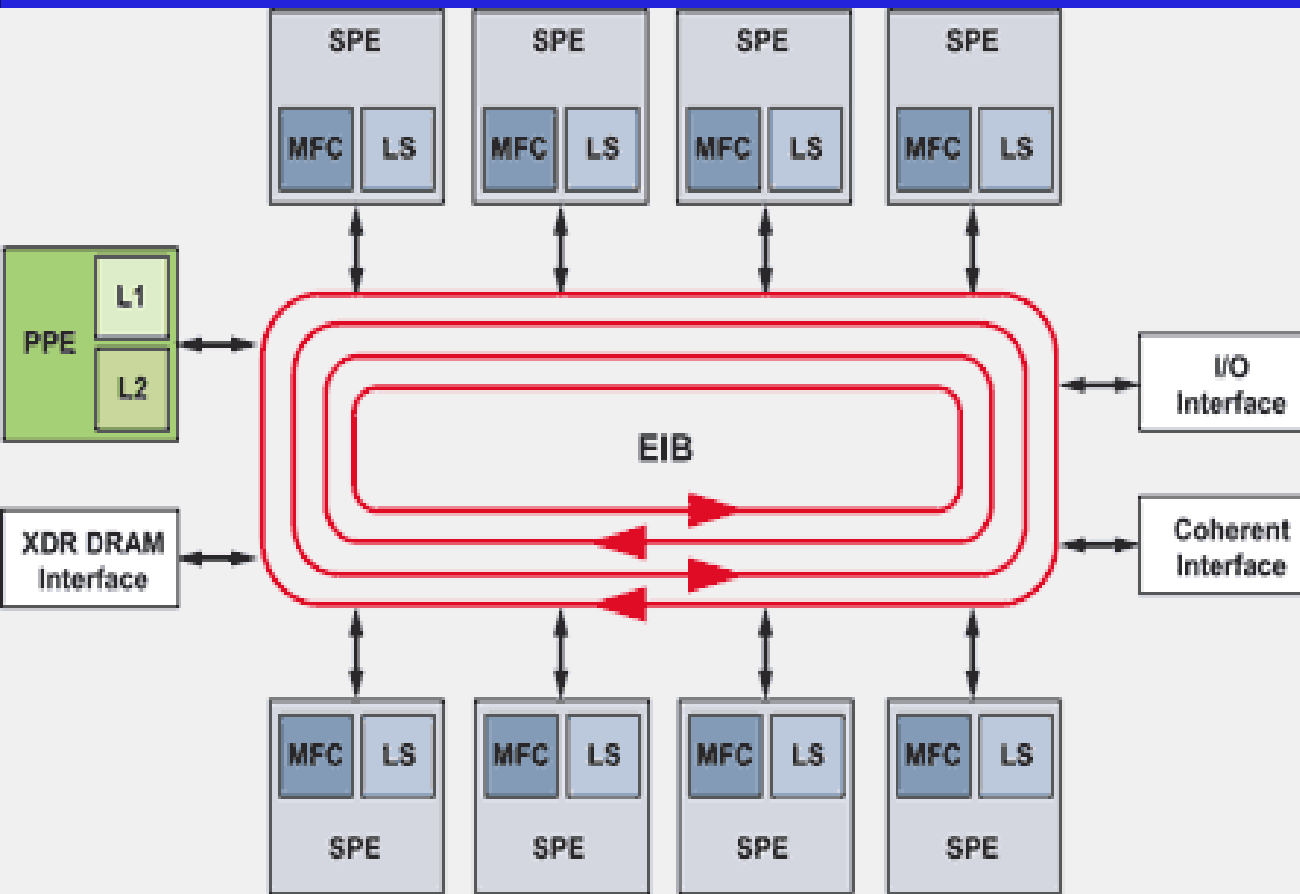
- **Cell BE from IBM – Sony – Toshiba**

- general purpose computing: next generation CPU
- designed for physics simulation to keep up with GPU
- explicit SIMD vector operations and DMA control



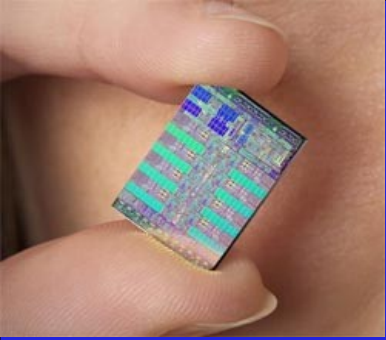
9. The Cell/B.E. architecture

- 8 SPU cores provide 8X speed-up factor
- Dual pipe 128bit SIMD => another 8X
- $8 * 8 * 3.2\text{GHz} = 204.8\text{GFlops}$
- Programmable MFC+DMA for each SPU
- 300 GB/s Element Interconnect Bus (EIB)



It was designed to accelerate physics simulation in a game console, now let's use it for scientific simulation

10. Cell versus FPGA



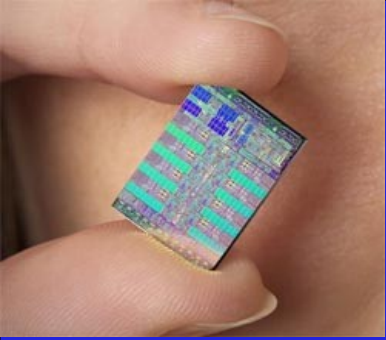
Cell

- efficient in floating point arithmetic (8x8)
- programmed using standard C/C++ code
- 3D transformations, force field calculations
- IBM's Cell SDK 3.0+ guarantees backward compatibility for new Cell processor generations

FPGA

- efficient for integer and bit arithmetic / logic
- hardware gate level design
- discrete math problems, graph algorithms, gene sequence alignment, fingerprinting
- Hardware advancements in newer generations often break compatibility with earlier "code"

11. Cell versus GPU



Cell

- General purpose CPU
- Programmable MFC DMA to local store via fast EIB
- Explicit SIMD vector operations on any data
- Standard profiling and debugging tools available
- Uniform hardware specs
- Double precision floating point calculations

GPU

- Designed for massively parallel independent tasks
- Multi-layer memory cache prone to latency problems
- Many-thread kernel code to operate on data arrays
- Tuning and debugging by trial and error (4x4 vs 2x8)
- Wide variety of hw. Specs
- Only single precision floating point operations

12. The chicken crowd test

- Cell was developed to allow the CPU running the simulations to keep up with the GPU rendering for 3D games in the PS3 console
- A RapidMind demo performs crowd simulation on a chicken farm, where each chicken interacts with all other birds around and the environment in a complex behavior model
- The simulation runs smoothly (30fps) with graphics for 3000 chickens. The Cell can handle 15000 chickens real-time, but graphics rendering drops to 10fps “sluggish” video dropping 2 out of 3 frames



Cell Programming challenges

- Each SPE has a 256KB local store for all code and data
- Code has to be split into execution units for SPE runs
- Explicit management of SPE resources (loading code)
- Explicit data management via MFC programming
- No branch prediction hardware on SPE: one branch statement could cost as much as 144 float operations
- SIMD operations require the data to be in consecutive memory location aligned to 16 byte boundaries
- DMA transfers are efficient in multiples of 128 bytes
- One DMA transfer block is limited to 16KB
- Code should be streamlined for dual-pipe instructions (loop unrolling to work with 4x4 data units)

High level HPC languages

- **RapidMind**
 - Originally designed for GPU acceleration
 - Supports Cell/B.E. and multi-core Intel/AMD CPUs
 - Dynamic runtime level code optimization
 - C language extension, vector and matrix operations
 - Performance in pilot test (Lennard Jones 6-12 vdw) was significantly slower than direct SPE coding
- **OpenCL - a hope for portable accelerator programming**
 - Apple initiated API accepted as open standard by various vendors including IBM, Nvidia, AMD, Intel
 - No implementations available yet - later this year
 - The BIG question: performance loss at abstraction ?

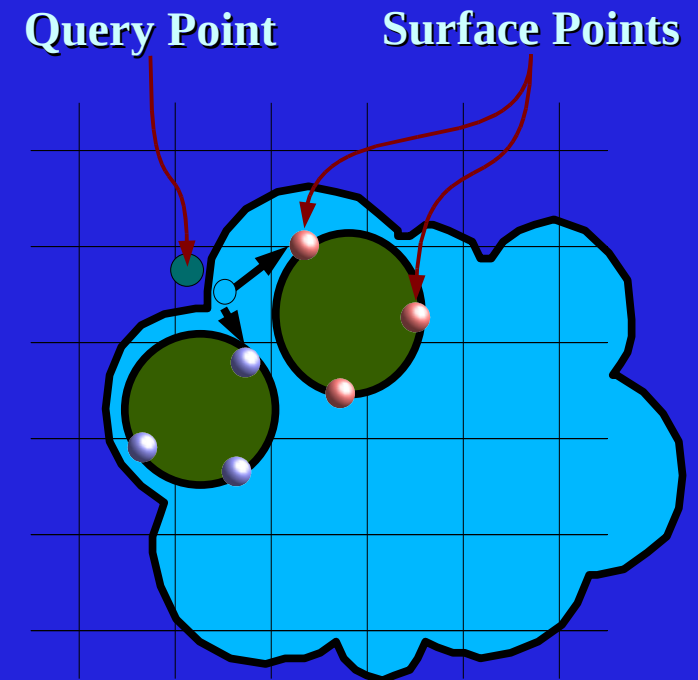
15. Programming tricks in porting eHiTS to the Cell B.E. to reach maximum performance

- **Traditional tricks that do not work well on the cell B.E.:**
 - **Precomputed steric clash grid to reduce pairwise computations**
 - **doesn't fit SPU**
 - **Lookup grid for finding surface points in query proximity**
 - **doesn't fit SPU**
 - **Many conditions to reduce computing load – branch penalties**
- **eHiTS Lightning is fundamentally different from the previous code:**
 - **new data structures to support SIMD operations**
 - **low level intrinsic coding (direct map to assembly)**
 - **branch elimination, loop unrolling**
 - **direct double-buffered DMA programming**
- **No grid approximation, longer cut-offs**
- **We get the benefits of IMPROVED accuracy and FASTER results**
 - **no compromise**

16. Proximity grid for SIMD use on the SPU

To find the set of all atoms close to a given point:

- For each grid cell, pre-store the array of (x,y,z) coordinates of surface points close to the center of the cell in the main memory. The coordinates are stored in vector floats for 4-point sets for ideal SIMD access.
- In the cell SPU local memory we store the address and the length of this array for each grid cell.
- For the given query point we fetch this array using double buffered DMA commands.



17. Specific SPE function Speed-up Results

Code component (task)	Speed-up factor	
	PS3	QS21
eHiTS Scoring (with rotamer opt.)	53x	125x
Rigid Fragment Docking	31x	76x
Pose Matching	7x	18x
Conformation Minimization	34x	45x
Final Optimization	12x	33x
Total (complete flexible docking)	5-56x	8-117x

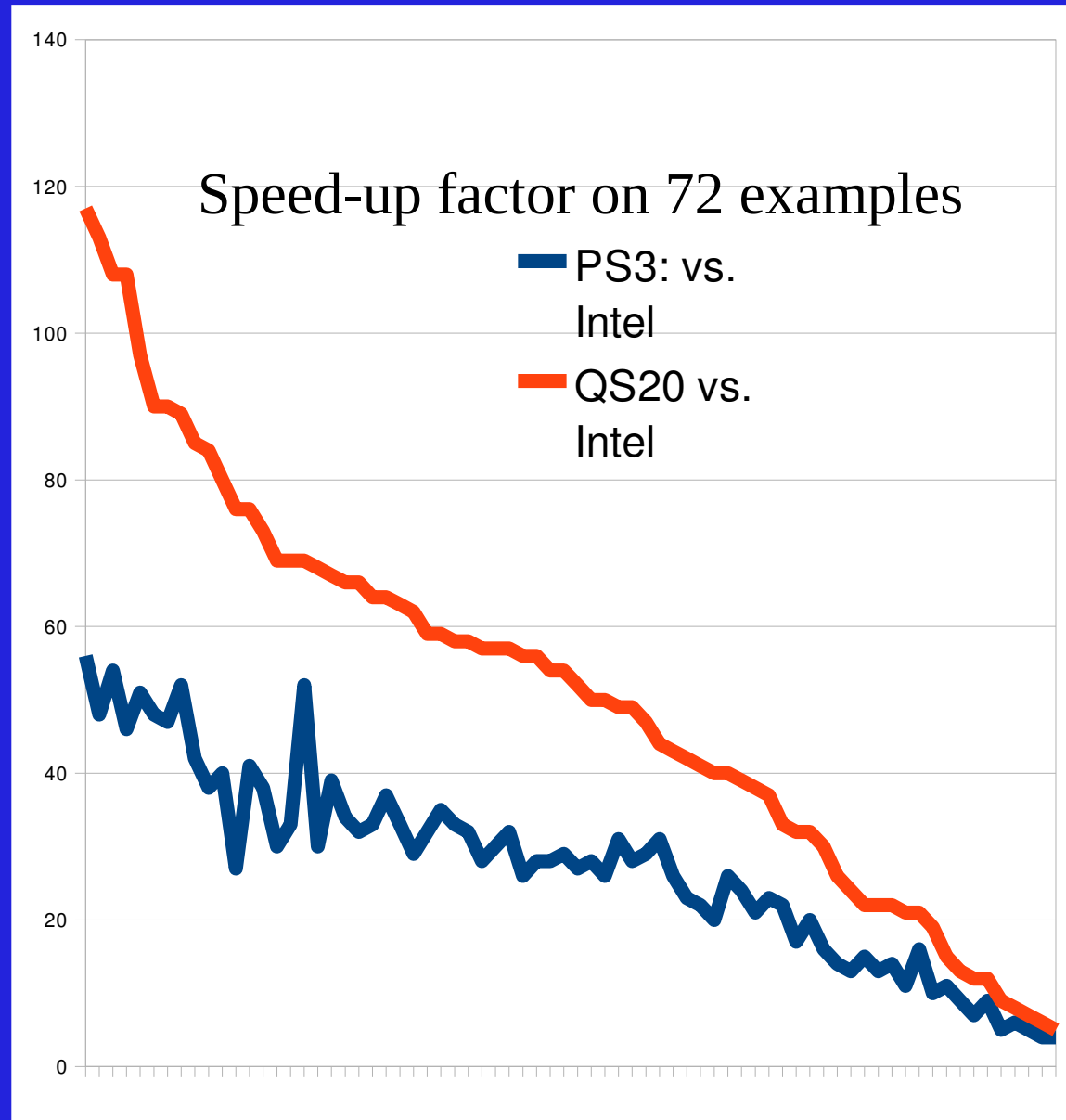


18. eHiTS Lightning

Speed-up Results: up to **117X**

run speed
time up

Intel avg.	221s	n/a
PS3 avg.	7s	32x
QS20 avg.	4s	62x
Intel max.	913s	n/a
PS3 max.	31s	56x
QS20 max.	16s	117x

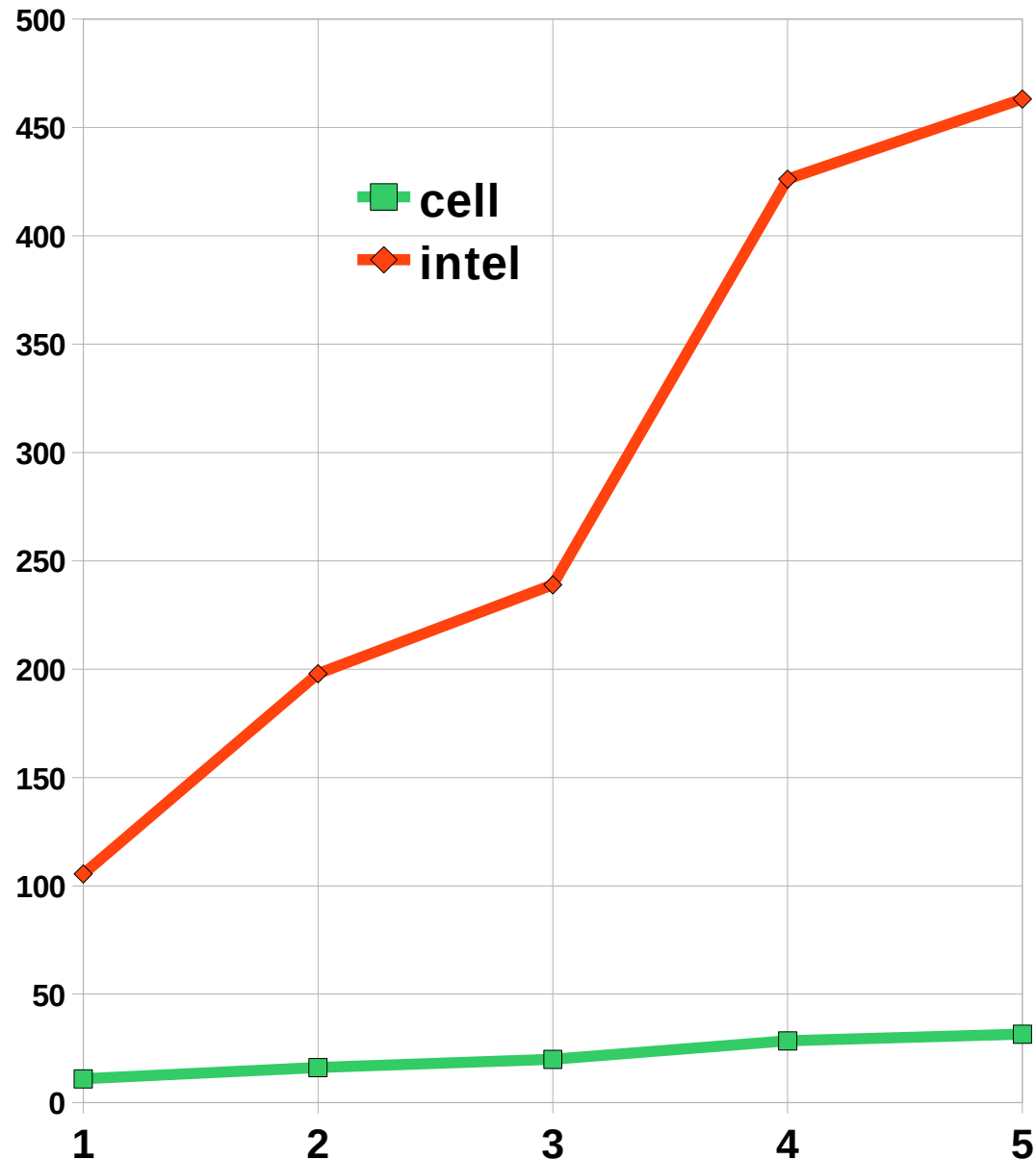


19. Speed-up dependence on accuracy parameter

Graph shows the run time in seconds for accuracy levels 1-5

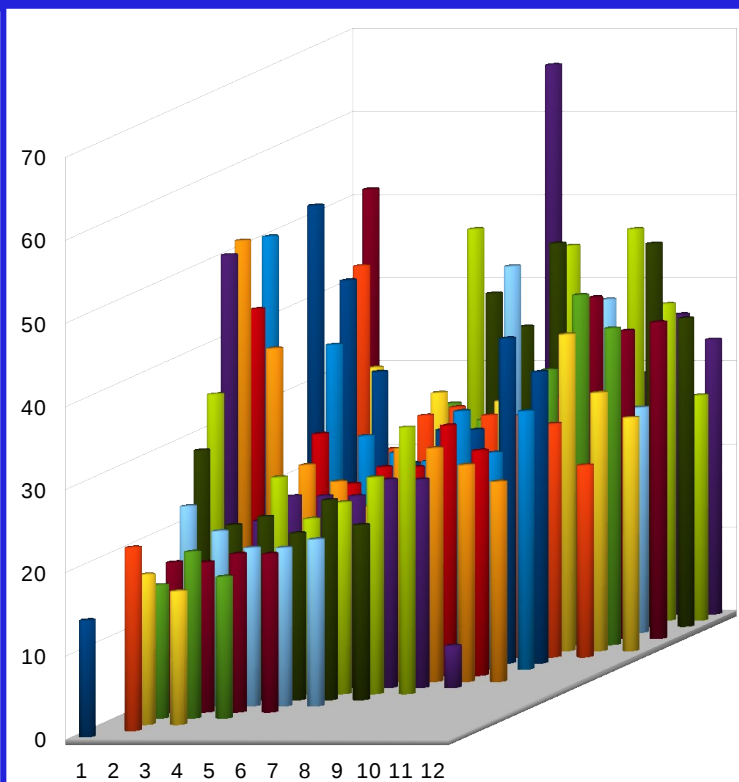
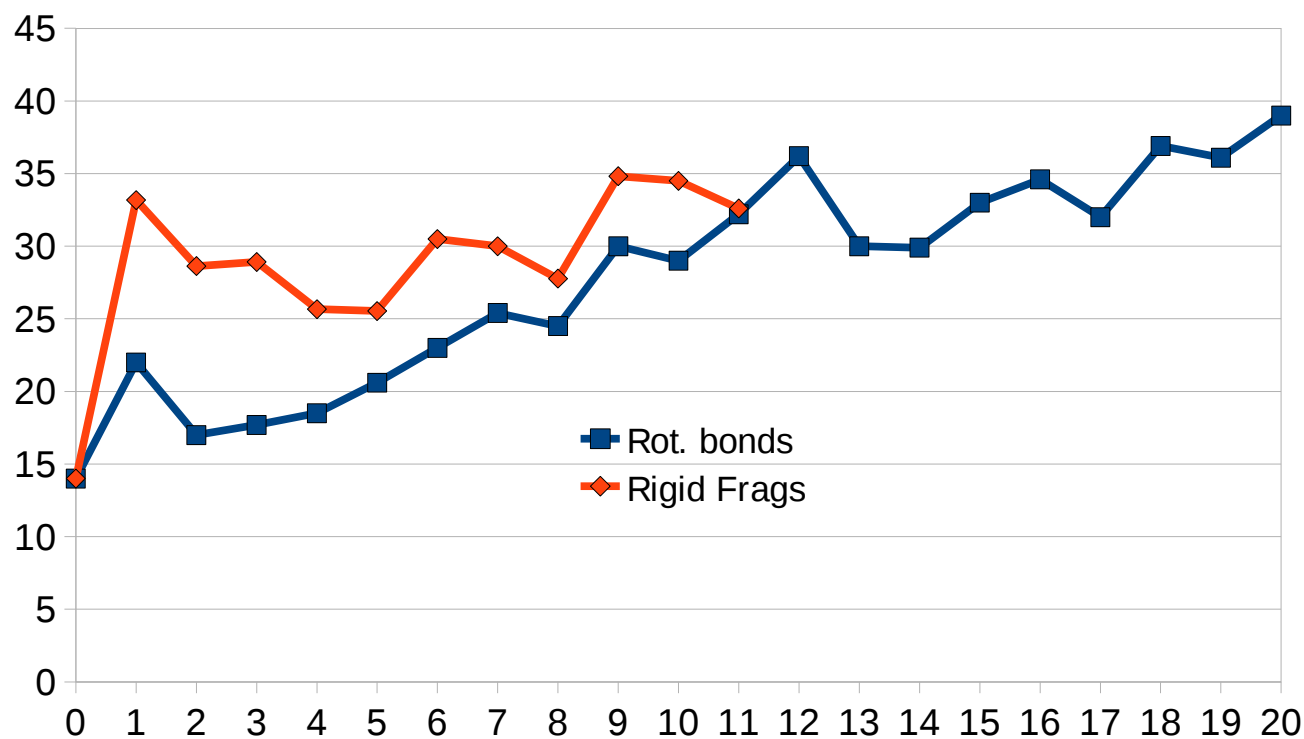
Values are averaged over test cases with various complexity

The Cell/BE speed-up increases with the accuracy



20. Speed-up dependence on case complexity

The speed-up factor between PS3-cell versus intel processor is plotted in standard accuracy 3 in the function of the number of rigid fragment and the number of rotatable bonds (complexity of docking)



21. HW operating cost comparison

3 year cost	400 CPU cluster	FPGA	GPU	Cell/B.E.
Hardware	\$200-\$400K	\$60K	\$30K	\$4-40K
Electricity	\$180-\$360K	\$6K	\$18K	\$3K
Total	\$380K-\$760K	\$66K	\$48K	\$7-43K

Hardware quantities are chosen to represent the same docking throughput

Cell hardware cost varies by platform:

Sony PS3 ~\$400, Mercury CAB ~\$8K, IBM QS21 ~\$10K

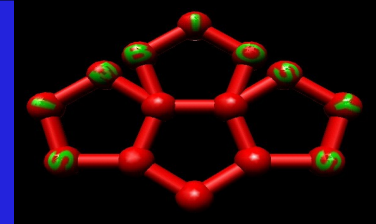


22. Top 500 and Green 500

- **Fastest** supercomputer in the world from Top500 list:
RoadRunner IBM QS22 cluster PowerXcell 8i - Cell/B.E.
<http://www.top500.org/lists/>
Los Alamos National Laboratory (physics simulations)
- Green 500 list measures highest MFLOPS/Watt
<http://www.green500.org/lists/>
- The **first 7 entries** on the Green 500 list are Cell based
- RoadRunner is #7 on the Green 500 list
- There are 11 BlueGene systems following the cells,
leaving the first Intel Xeon based system at rank 19
- There are no GPU or FPGA based systems on the list



23. Summary



- Exhaustive, highest accuracy - **don't miss a potential drug!**
- Very fast – **electronic High Throughput Screening** on the Cell B.E. “supercomputer in a chip”
- The Cell technology offers low cost, low power consumption hardware acceleration with very high speed-up performance
- Programming on the Cell is more challenging than Intel/AMD coding, but still far simpler than GPU or FPGA
- Other talks about eHiTS at this ACS meeting:
 - Method: Tuesday Mar.24th 10:10 am, room 254A
 - Scoring: Wednesday Mar.25th 10:00 am, room 254A
 - Application: Thursday Mar.26th 8:30 am, room 258
- Visit us at Booth #316 at the exhibition
Request a free evaluation online: <http://www.simbiosys.ca/>