

Improving PolyBench: Heroes Wanted

Sanjay Rajopadhye
Colorado State University

Easy fixes & low hanging fruit

- Bugs:
 - ADI
 - A^TAX (already fixed but not released)
 - ...
- No test inputs/outputs provided
- Parameters are static (`#define`) so compilers can optimize away
 - even “dynamic” parameters use `#define`
 - must be passed as arguments to functions

Low Hanging Fruit

- Some benchmarks are really microkernels
 - Vector add, 1D Jacobi stencil, ...
- Some are terribly naïve
 - in **memory** – **fdtd_apml** is single assignment
 - in **work and memory** – **dynprog** and **reg_detect** are $O(N^4)$ with $O(N^3)$ memory
 - can be done in $O(N^3)$ work with $O(N^2)$ memory.
 - What is the goal? Do we expect compilers to reduce space/work complexity? If so, mark them as “challenges”

Some Higher Fruit

- Not very readable – heavy use of C macros
 - Motivation – need different “kinds of benchmarks” – e.g., linearized vs truly multidimensional arrays
 - Cleaner way: generate with a script – don’t use C macro processor as a scripting language.
- Relatively few truly difficult codes

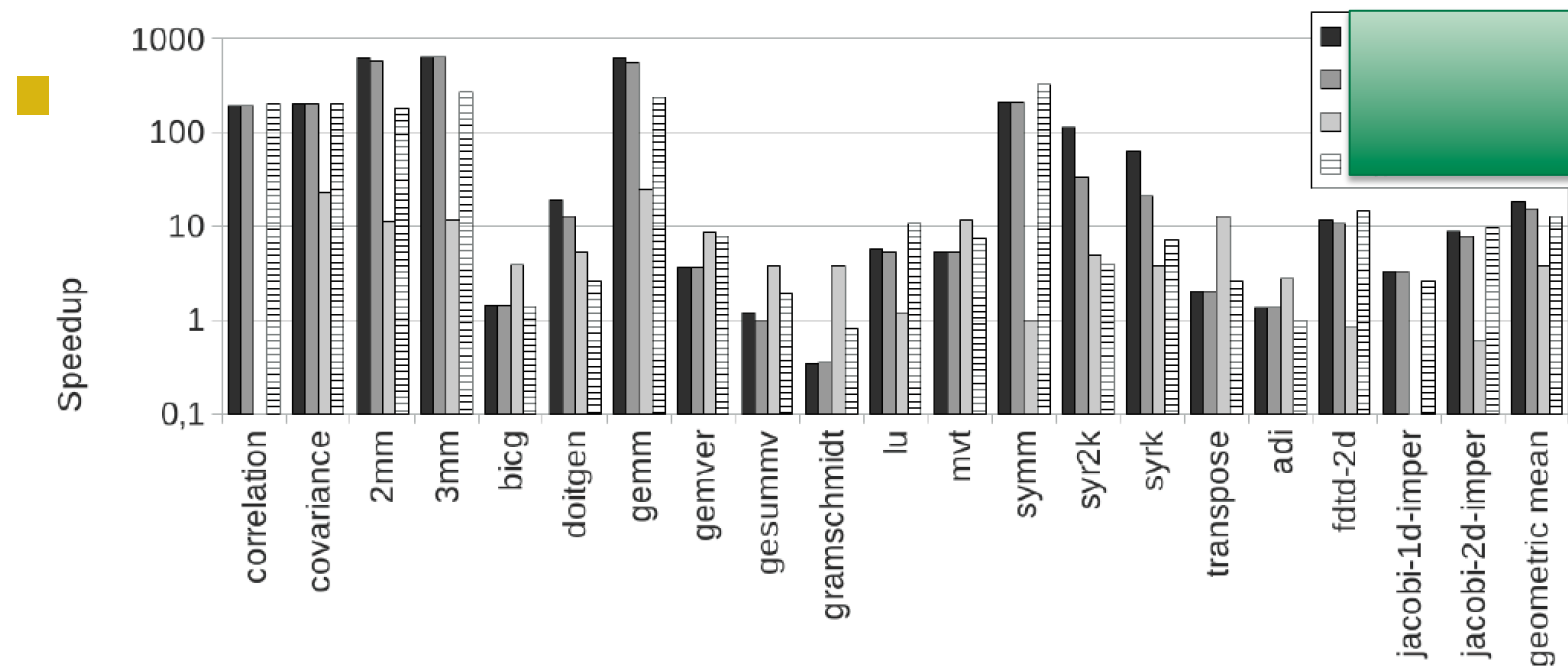
Higher Fruit (complete apps)

- HMMER (Hidden Markov Model – Viterbi)
 - hmmer.janelia.org
- Back-propagation NN training
- UAV tracking using POMDP
- Nupack (nucleic acid structure)
 - www.nupack.org
- Mfold/Unafold (RNA secondary structure)
 - mfold.rna.albany.edu
- Smith Waterman/BLAST/...
- FDTD apps
- ...

Serious Benchmarking

■ Q: What's wrong with this?

Overall Results: The Same One Produced By Others



Serious Benchmarking

- Q: What's wrong with this?

- Answer
 - No absolute performance data
 - No information about GFLOP/sec attained
 - No comparison with the best possible (by hand)

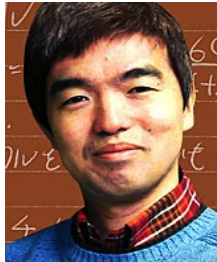
Towards a solution

- Need to know best achievable performance
 - on a given target
 - for a given set of size parameters
 - for all the benchmarks
- Separate compiler optimizations from “algorithmic innovations”
 - Dynamic programming
 - Floyd-Warshall
- Calls for heroes

Cédric's heroes

Outstanding Optimization Heroes

- John Carmack
Game developer, id Software
innovations in 3D Graphics, fast inverse square root



- Kazushige Goto
Engineer, Intel
hand-tuned programs for supercomputers
“Goto BLAS”

- Vasily Volkov
PhD student, Berkeley
3x faster FFT than nVidia's proprietary implementation



Students are our heroes

- Set up a competition
- Like DARPA's HPC Challenge benchmarks
- Reward successful students with
 - Fame
 - Travel to next IMPACT
 - Riches – ipad, laptop, ...