Hardware acceleration opportunities in bioinformatics and computational biology

Leonid Yavits, Roman Kaplan

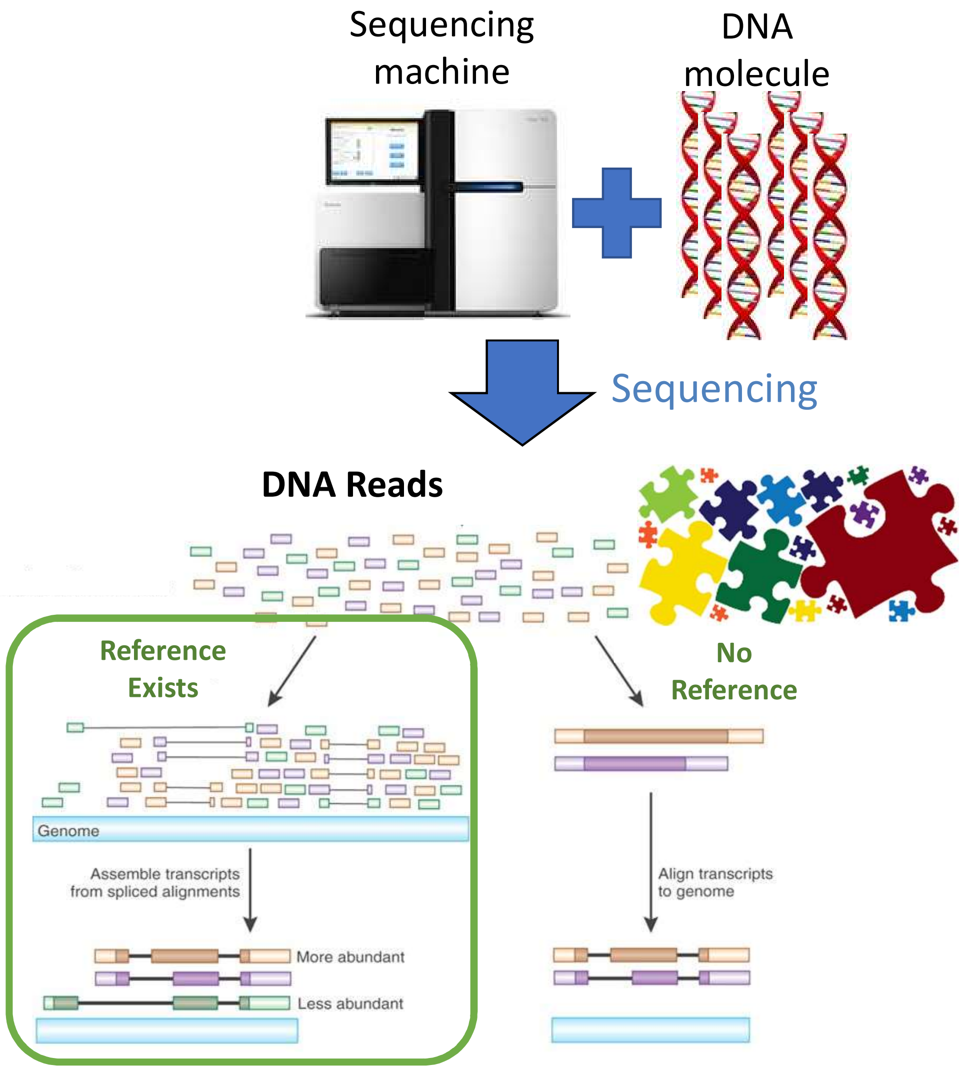
Accelerator Architecture for Computational Biology and Bioinformatics (AACBB-2019) @ HPCA-2019

Acceleration of Genome Assembly

Genome Assembly

- Genomics focuses on the structure, function, evolution, mapping, and editing of genomes.
- Genomics leads the revolution in
  - Precision medicine, personalized healthcare, on-site disease detection
  - The way we understand origins of life and evolution
- Genomics starts with genome assembly

→ **Almost prohibitively expensive, takes hundreds of hours on HPC**



Haas and Zody, Nature Biotechnology 28, 421–423 (2010)

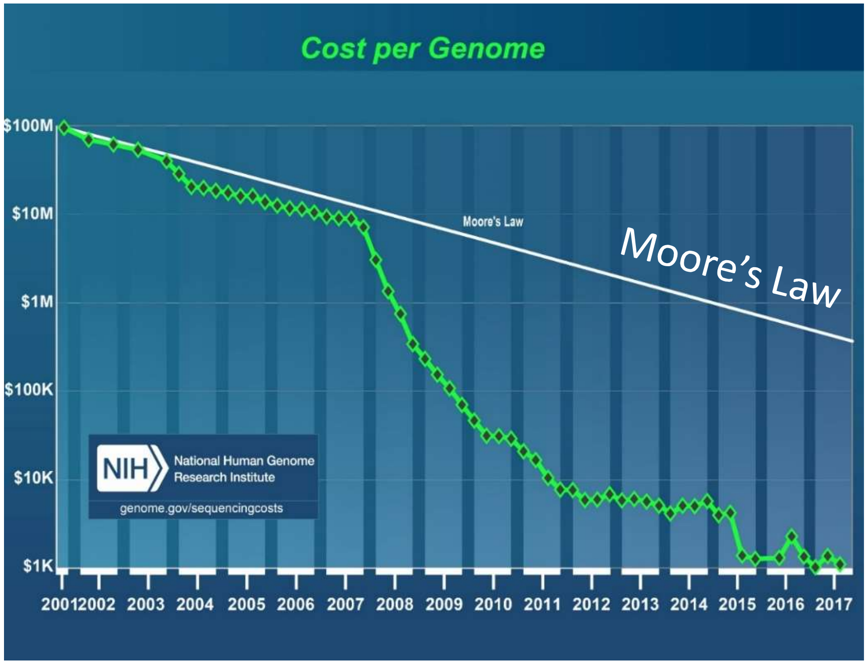
Challenges of 3rd generation DNA sequencing

- Very long reads are of varying lengths: almost up to 1M bp
  - Providing a great coverage
- High error rates: 15% to 20+%
- Poses a huge challenge but also a great opportunity for hardware acceleration of genome assembly

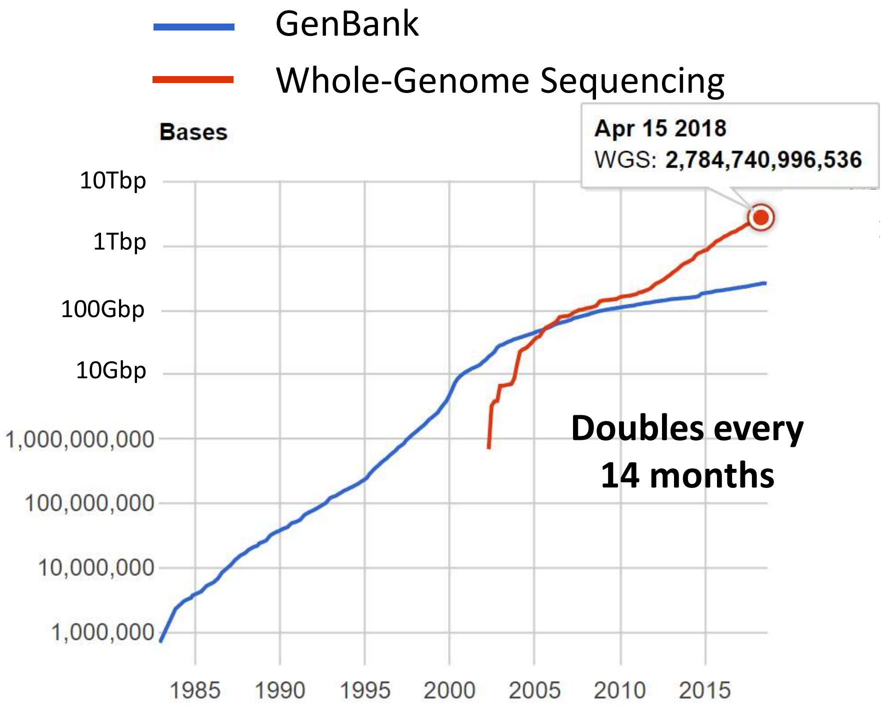
Sources: [right] Rhoads, Anthony, and Kin Fai Au. "PacBio sequencing and its applications." Genomics, proteomics & bioinformatics 13.5 (2015): 278-289.
[left] https://nanoporetech.com

Why Genome Assembly Requires Acceleration?

1. Reduced costs ➔ Exponentially growing database sizes
    1. Example: human genome=3Gbp. Sequencing requires ~30× coverage
- Even worse in other fields, like Metagenomics

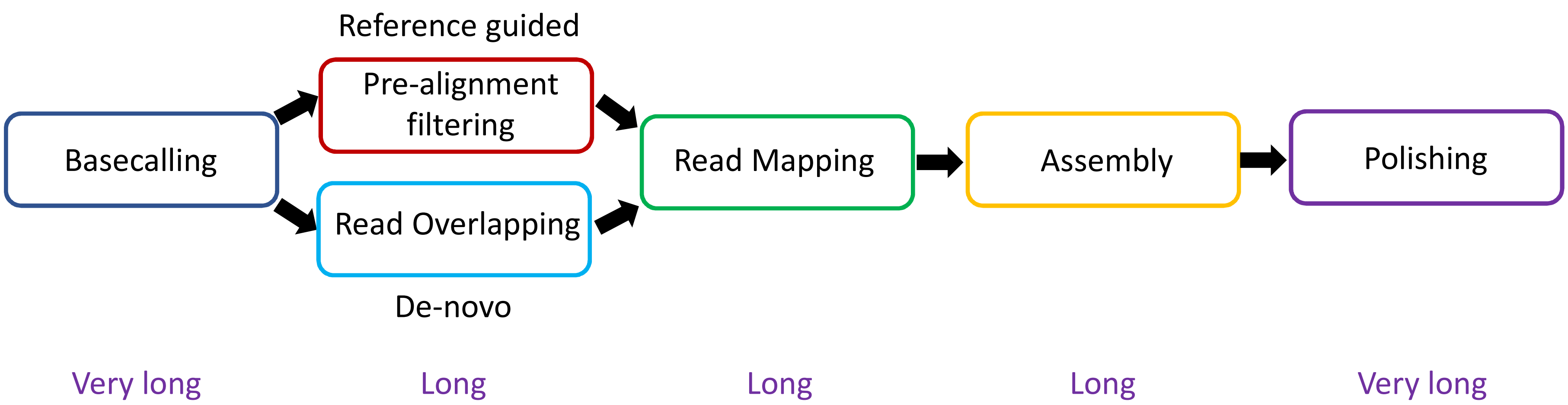


Source: https://www.ncbi.nlm.nih.gov/genbank/statistics/



Doubles every 14 months

Genome Assembly Pipeline*
(or why bioinformatics requires acceleration 2)



Reference guided

Pre-alignment filtering

Basecalling

Read Overlapping

De-novo

Read Mapping

Assembly

Polishing

Very long

Long

Long

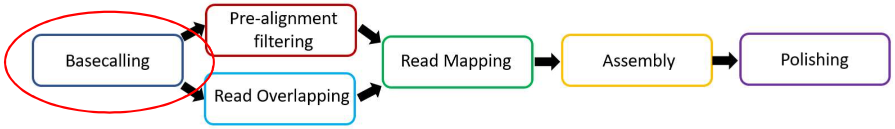
Long

Very long

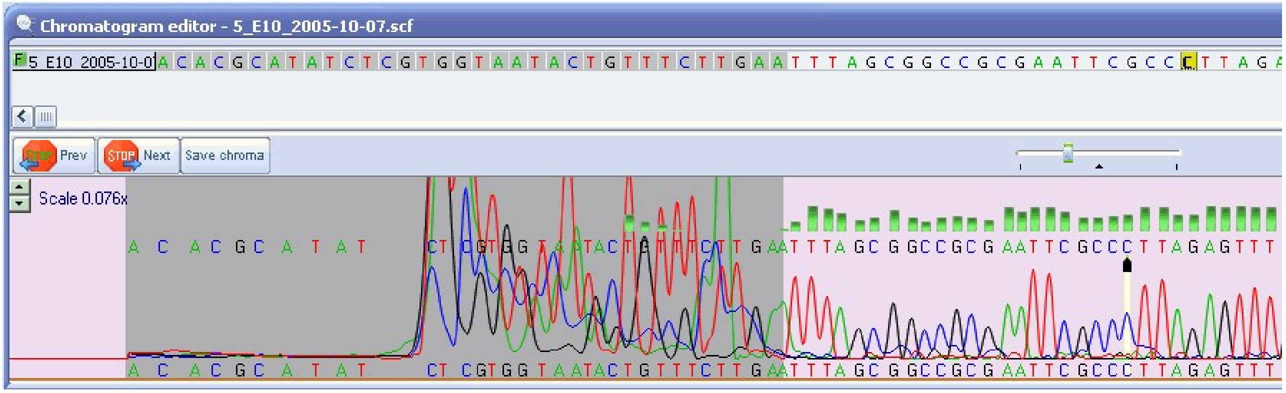
Long: Hours - Low tens of hours
Very long: Tens - Low hundreds of hours

* A computer architect perspective

Basecalling (3rd gen)



- Basecalling is the process of assigning bases to chromatogram peaks
  - Chromatogram is a visual representation of a DNA sample produced by a sequencing machine
- Earlier solutions use Hidden Markov Models
- Latest solutions use RNN (DeepNano, Albacore) or a combination of RNN and CNN (Chiron)
- Existing DNN accelerators can probably be employed to this end

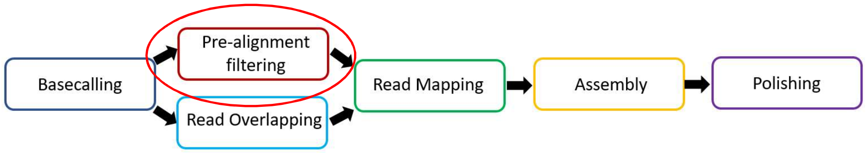


Chromatogram editor - 5_E10_2005-10-07.scf

F 5 E10 2005-10-0 ACACGCATATCTCGTGGTAATACTGTTTCTTGAATTTAGCGGCCGCGAATTCGCCCTTAGA

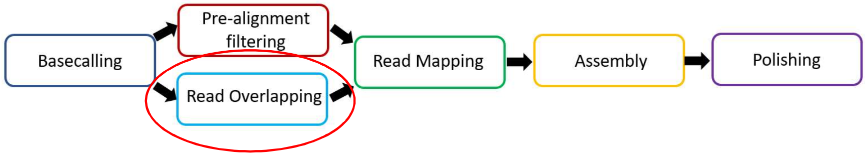
Source: https://www.dnabaser.com/help/snp%20mutation%20detection/base%20caller.html

Pre-alignment Filtering

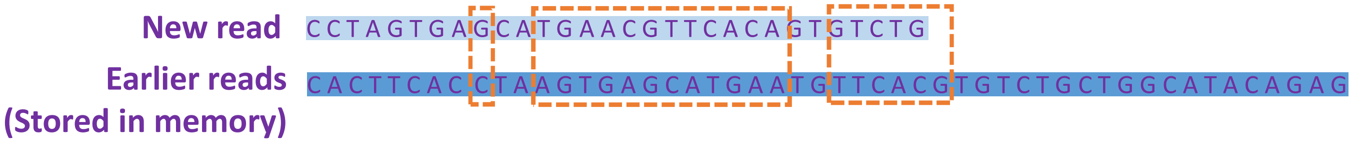


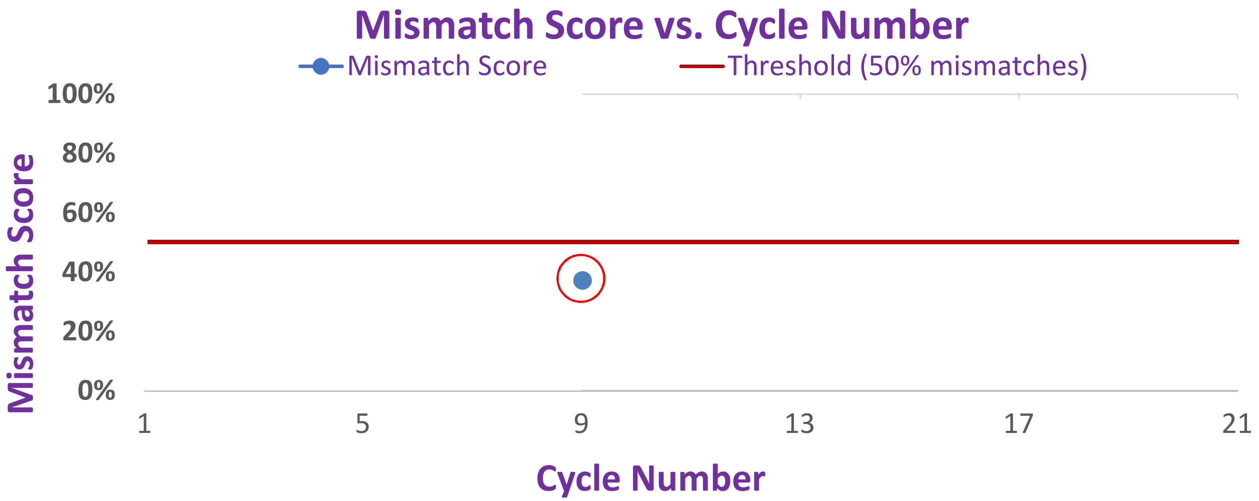
- Read mapping (alignment) complexity is $O(n^2)$
- Pre-alignment filtering is proposed to reduce the alignment complexity
  - Filters out unlikely matching positions
- Typically follows two approaches
  - Hash table based, good for reference guided assembly: GateKeeper, GRIM
  - Shifted Hamming Distance, good for de-novo assembly (RASSA)

Read Overlapping

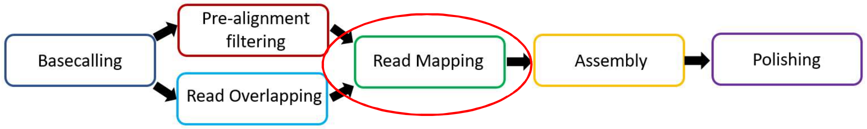


- Used in de-novo assembly
  - Stiches reads using prefix – suffix similarity
- There is no reference sequence → no hashing
- A solution was proposed: calculating Hamming distance using processing in associative memory (RASSA)


New read
CCTAGTGAGCATGAACGTTCACAGTGTCTG
Earlier reads
(Stored in memory)
CACTTCACCTAAGTGAGCATGAATGTTCACGTGTCTGCTGGCATACAGAG

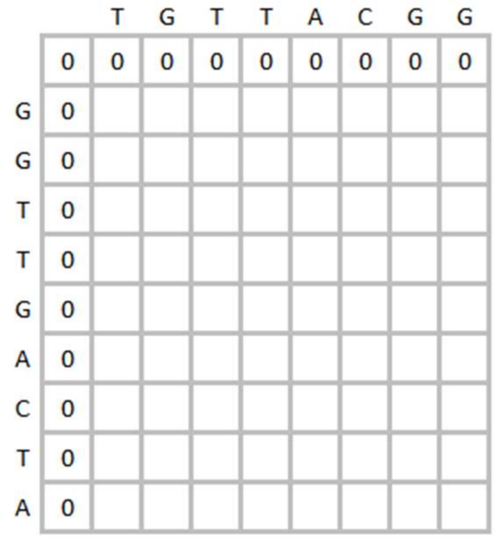


Read Mapping



- Probably the best-researched step
  - A well defined algorithm (sequence alignment using dynamic programming)
- A large number of accelerators have been proposed
  - Architecture: Conventional, systolic, 3D NDP, PIM
  - Implementation: ASIC and FPGA
- Just in the last year (both 2nd and 3rd gen): GenAx, MPU-BWM, MESGA, AligneR, BioSEAL, RADAR, DARWIN

Initialize the scoring matrix

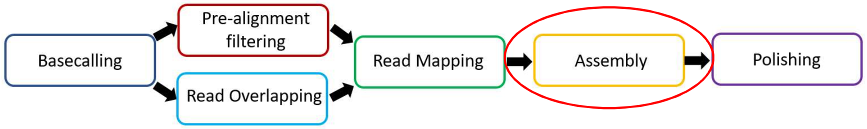


		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0								
G	0								
T	0								
T	0								
G	0								
A	0								
C	0								
T	0								
A	0								

Substitution matrix: $s(a_i, b_j) = \begin{cases} +3, & a_i = b_j \\ -3, & a_i \neq b_j \end{cases}$

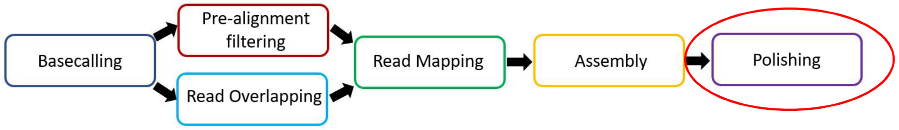
Gap penalty: $W_k = kW_1$
$W_1 = 2$

Assembly



- Connecting the mapped reads into an entire genome
- Sometimes the assembly is done by traversing the De Bruijn graph
- I’m not familiar with assembly accelerators

Polishing



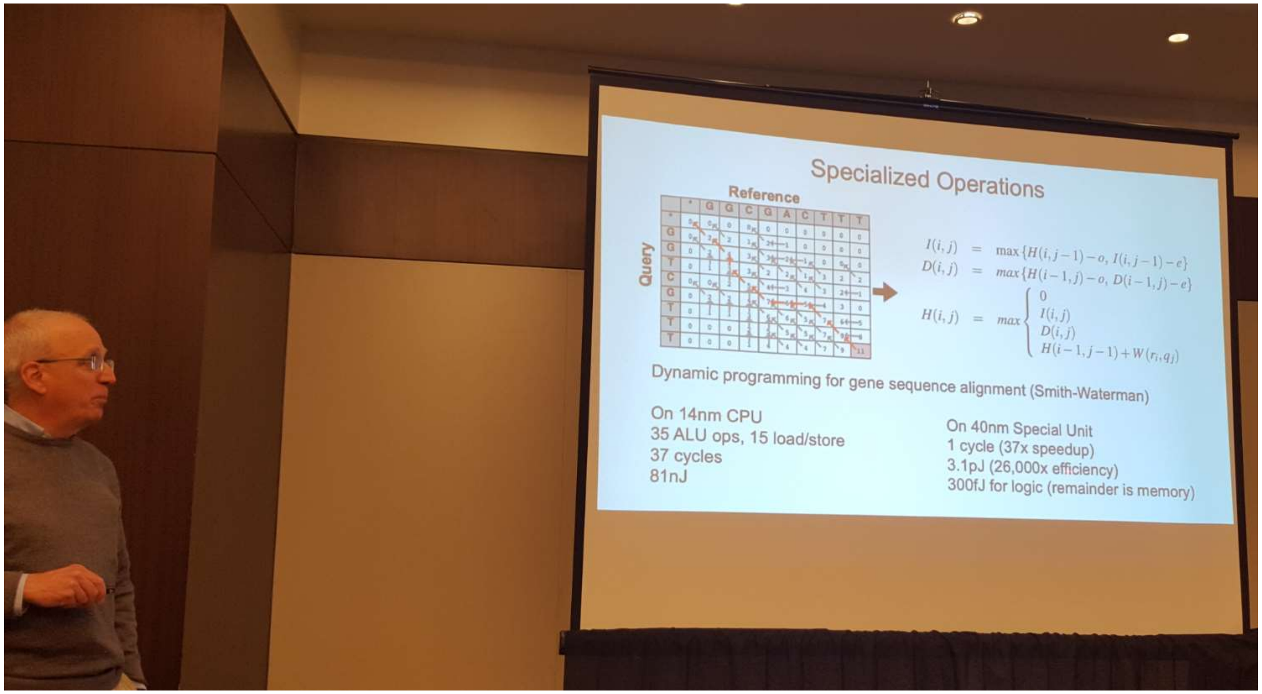
- Post-assembly error correction
  - Especially effective in 3rd gen sequencing, where error rates are very high
- Can go back to the raw signal to improve the final assembly
- I don’t know of polishing accelerators although polishing could be a very lengthy process

Insights from AACBB-2019

Mainly based on Bill Dally’s keynote

Accelerator specialization

- Accelerator specialization is great for energy efficiency
- For speedup, parallelism is mandatory
- Example: Darwin
  - Base op 37 cycles, 81nJ
  - Special unit: 1 cycle, 3.1pJ
  - 37x speedup
  - 26,000 energy efficiency



Specialized Operations

$$
\begin{aligned}
I(i,j) &= \max\{H(i,j-1)-o,\ I(i,j-1)-e\}\\
D(i,j) &= max\{H(i-1,j)-o,\ D(i-1,j)-e\}\\
H(i,j) &= max\begin{cases}0\\ I(i,j)\\ D(i,j)\\ H(i-1,j-1)+W(r_i,q_j)\end{cases}
\end{aligned}
$$

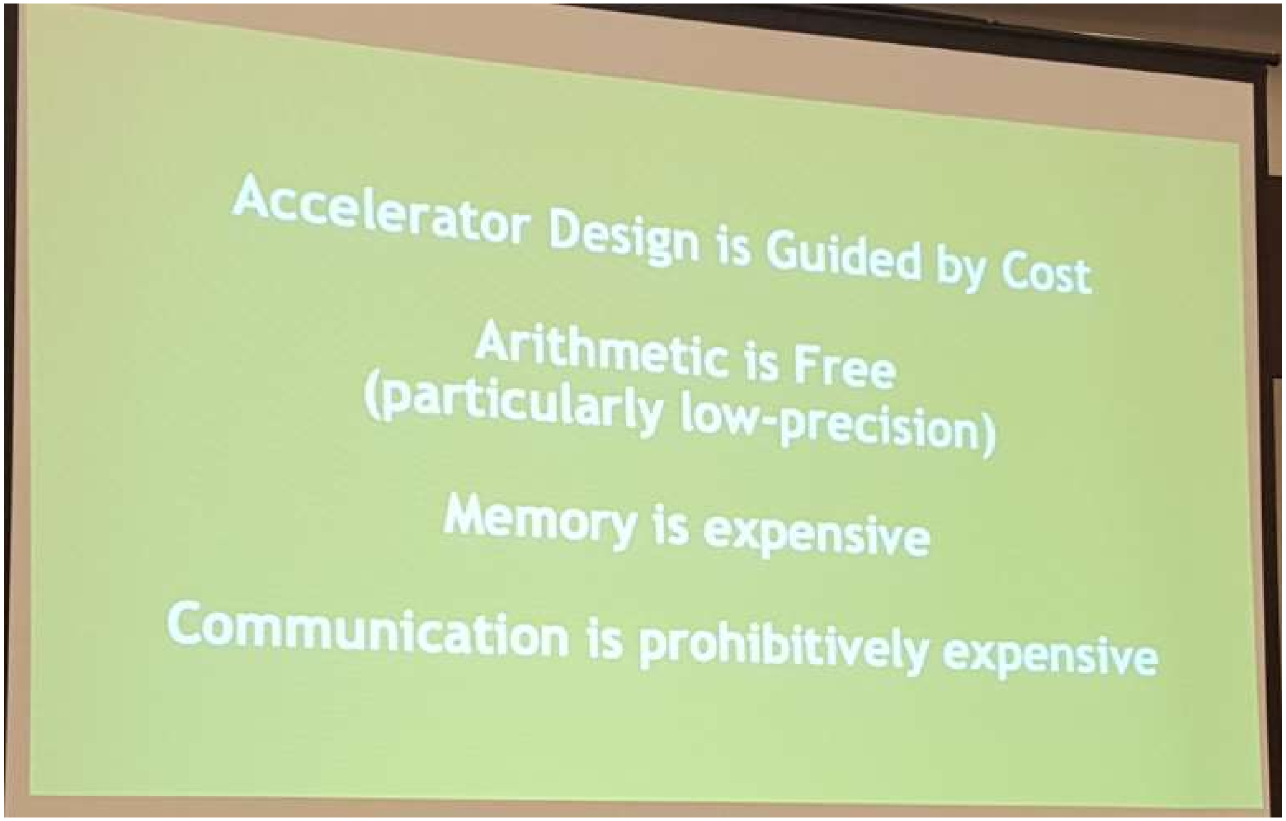
Dynamic programming for gene sequence alignment (Smith-Waterman)

On 14nm CPU
35 ALU ops, 15 load/store
37 cycles
81nJ

On 40nm Special Unit
1 cycle (37x speedup)
3.1pJ (26,000x efficiency)
300fJ for logic (remainder is memory)

Accelerator design is guided by cost

- Every memory hierarchy level increases the cost of access by at least an order of magnitude
- On-chip memory costs 10x-100x more per bit than DRAM but it’s often less expensive (because of comm costs)



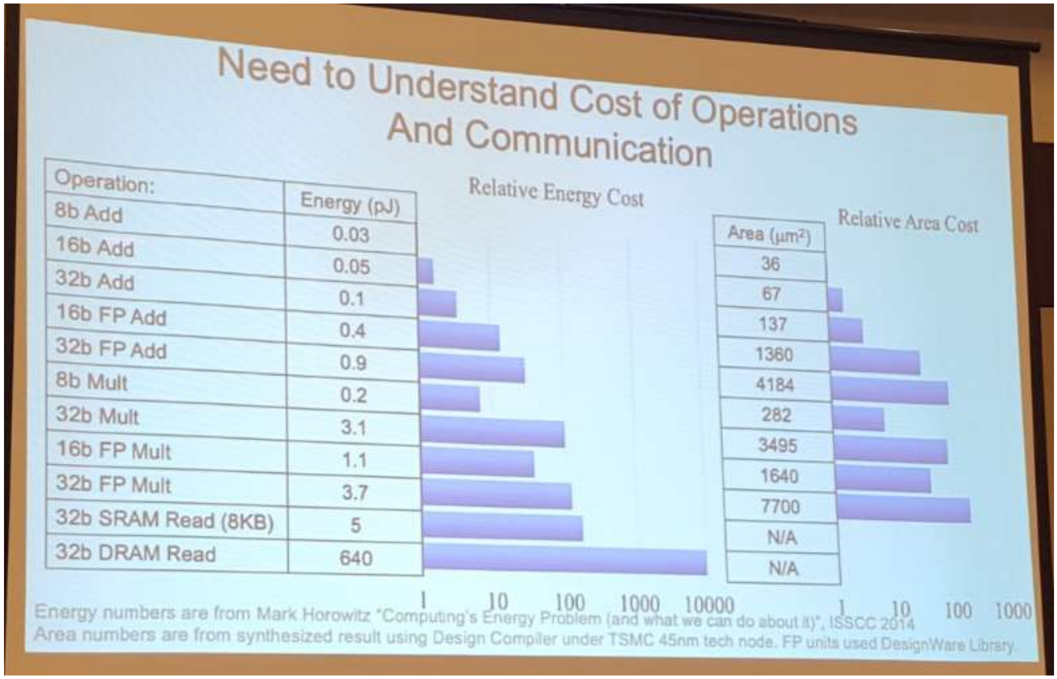
Accelerator Design is Guided by Cost

Arithmetic is Free
(particularly low-precision)

Memory is expensive

Communication is prohibitively expensive

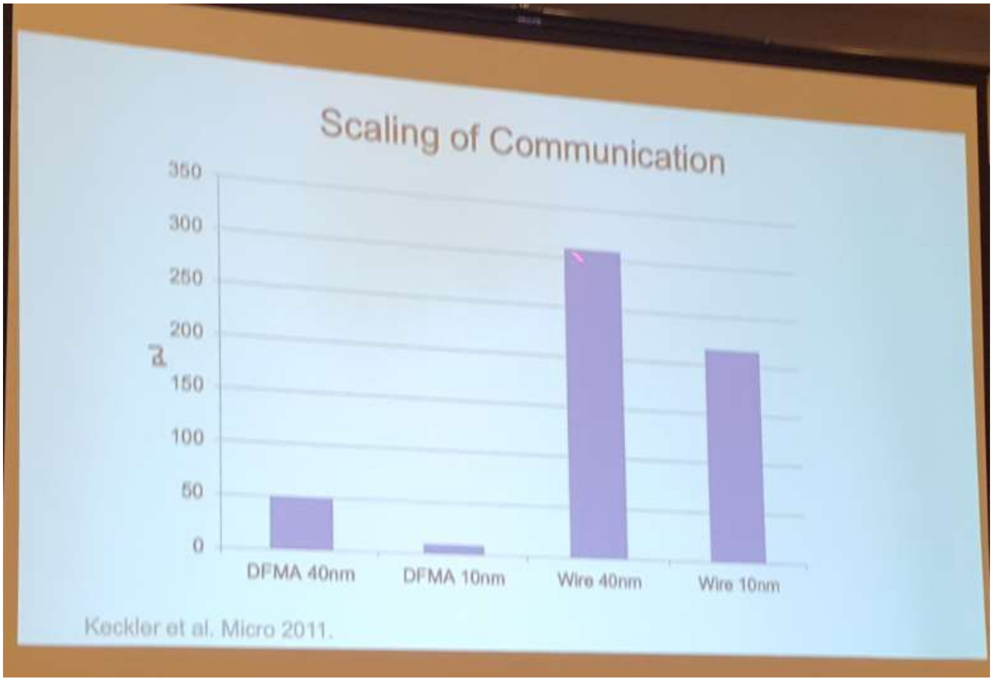
Once more on the costs of communication



Need to Understand Cost of Operations And Communication

Operation:	Energy (pJ)	Area (µm²)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A

Energy numbers are from Mark Horowitz "Computing's Energy Problem (and what we can do about it)", ISSCC 2014
Area numbers are from synthesized result using Design Compiler under TSMC 45nm tech node. FP units used DesignWare Library.



Scaling of Communication

Keckler et al. Micro 2011.

Hardware software co-design

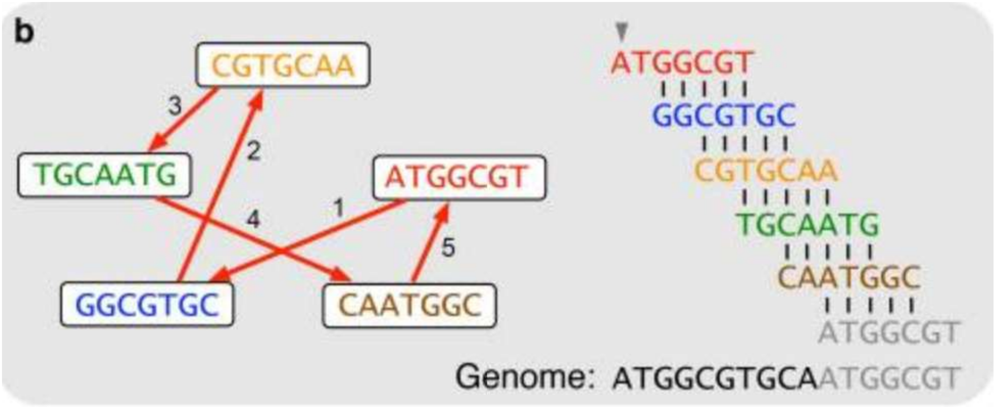
- The algorithm has to change
- Some algorithms can’t be sped up as is
- Hardware-software co-design is required to reach the speedup target

Misc

- Bioinformatics data can be mapped to graphs
  - Example: De Bruijn graph and its use in genome assembly

→ hence the relevance of accelerating graph processing

- Use of approximate computing to improve performance / energy efficiency



Ref slides

Bibliography

1. Turakhia, Y., Bejerano, G., & Dally, W. J. Darwin: A Genomics Co-processor Provides up to 15,000 X Acceleration on Long Read Assembly. ASPLOS 2018
2. Alser, M., Hassan, H., Xin, H., Ergin, O., Mutlu, O., & Alkan, C. (2017). GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping. *Bioinformatics*, *33*(21), 3355-3363.
3. Kim, J. S., Cali, D. S., Xin, H., Lee, D., Ghose, S., Alser, M., ... & Mutlu, O. (2018). GRIM-Filter: Fast seed location filtering in DNA read mapping using processing-in-memory technologies. *BMC genomics*, *19*(2), 89.
4. Kaplan, R., Yavits, L., & Ginosar, R. (2018). RASSA: Resistive Pre-Alignment Accelerator for Approximate DNA Long Read Mapping. *IEEE Micro*.