DNA Sequences Alignment in Multi-GPUs: Energy Payoff on Speculative Executions

Authors:

J. Pérez, M. Ujaldón
Computer Architecture Department
University of Malaga (Spain)

E. Sandes, A. Melo
Computer Science Department
University of Brasilia (Brazil)

Presenter:

M. Ujaldón
Full Professor @ Computer Architecture: >100 research papers published
CUDA Fellow @ NVIDIA: >100 talks/courses over the past 5 years

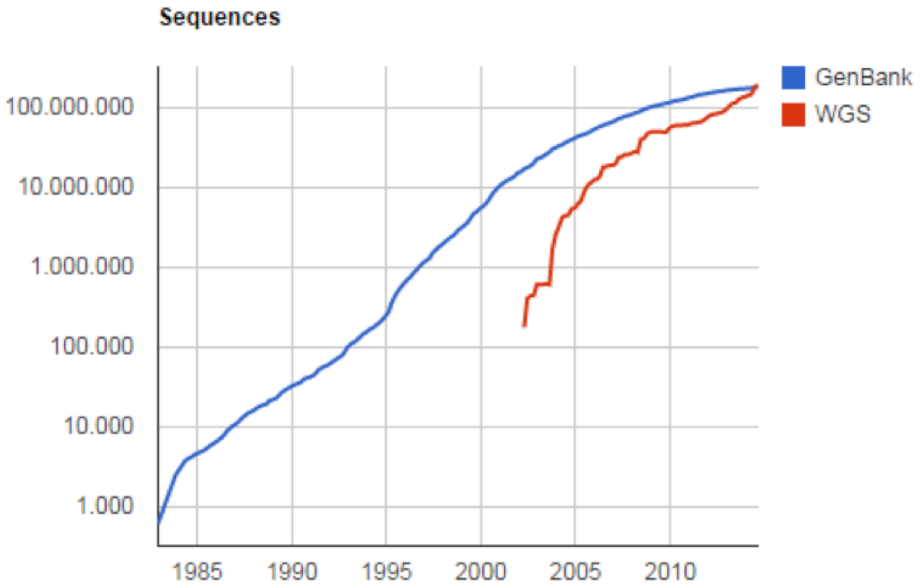
Talk outline [20 slides]

1. Background and motivation [4]
2. DNA sequence comparison [3]
3. CUDAlign [3]
4. Experimental setup [3]
5. Experimental results [4]
6. Speculative executions [3]

I. Background and motivation

The Smith-Waterman algorithm (SW)

- SW is a well known biomedical application to compute:
  1. The exact pairwise comparison of DNA/RNA sequences.
  2. A protein sequence (query) to a genomic database.
- Fine-grained parallelism applies better to 1.
- Already ported to multi-GPUs and Xeon Phis.
- Huge data volume ("big data"):
  - Tens-Hundreds of Million Base Pairs each sequence in our study.
  - Several Peta-cells (2^{50}) for the dynamic programming matrix used.



http://www.ncbi.nlm.nih.gov/genbank/statistics

Primary goals of this work

- We present a performance per watt analysis of SW using CUDAlign 4.0, identifying advantageous scenarios to maximize speed-up and minimize power consumption on GPUs.
- We evaluate:
  1. Speed-up, scalability and power on multi-GPU systems.
  2. How the workload size influences energy in data-intensive applics.
  3. The energy overhead on speculative executions.

GPU acceleration when energy matters: Identifying good and bad scenarios

- Example: Acceleration versus fuel consumption in my car. When is it worth to increase 10 MPH?

Driving at 60 MPH: 16.66% more speed. 10% more gas.

Driving at 100 MPH: 10% more speed. 25% more gas.

You can save fuel up to 4x if you press the throttle wisely.

Speculative executions: Evolution

- Back in the 80's and 90's:
  - Lots of successful stories (branch prediction, prefetching, look-ahead).
  - Because you gamble without risk, you are aggresive on bets.

- Now that energy matters:
  - Miss-predictions cause power consumption and no execution gains.
  - So you are undecided to play even with good cards.

II. DNA Sequence Comparison

DNA Sequence Comparison

- A DNA sequence is represented by an ordered list of nucleotide bases, strings of the {A, C, G, T} alphabet.
- Score function:
  - +1 for a match.
  - -1 for a mismatch.
  - -2 whenever you find a gap.

Example:

$$\begin{array}{cccccccc} A & T & A & C & T & C & C & A \\ A & T & A & - & T & C & C & A \\ \hline +1 & +1 & +1 & -2 & +1 & +1 & +1 & +1 \end{array}$$

$$score = 5$$

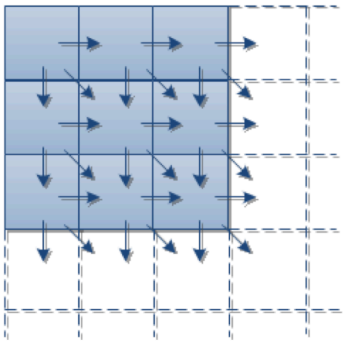
- Smith-Waterman algorithm obtains the optimal pairwise local alignment in quadratic time and space. Two phases:
  - Calculate the dynamic programming matrix.
  - Obtain the alignment (traceback).

Calculation of Dynamic Programming matrix: A toy example

- Alignment path:

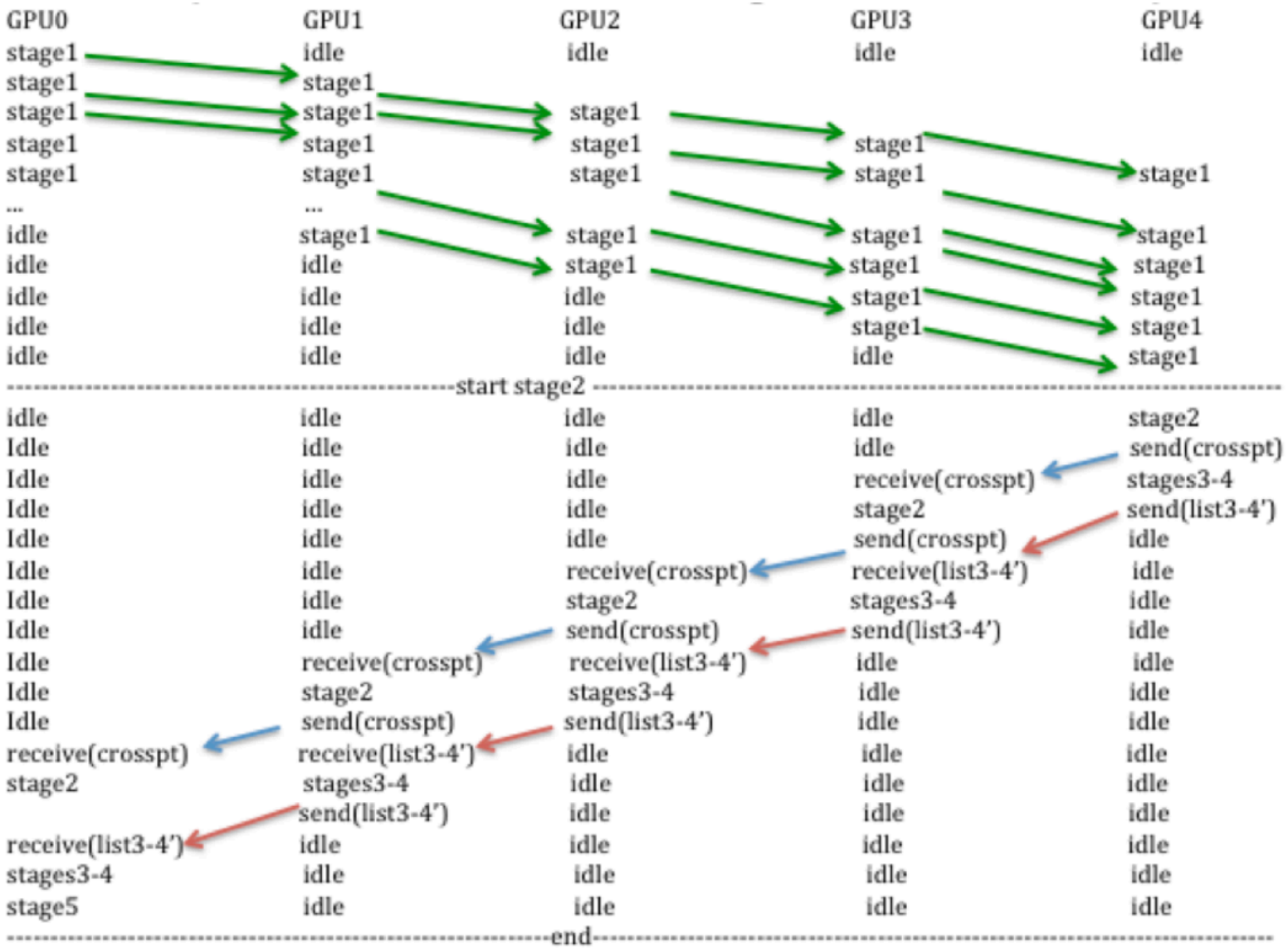
	*	C	T	C	G	A	T	A	C	T	C	C	A
*	0	0	0	0	**0**	0	0	0	0	0	0	0	0
A	0	0	0	0	0	**1**	0	1	0	0	0	0	1
T	0	0	1	0	0	0	**2**	0	0	1	0	0	0
A	0	0	0	0	0	1	0	**3**	**1**	0	0	0	1
T	0	0	1	0	0	0	2	1	2	**2**	0	0	0
C	0	1	0	2	0	0	0	1	2	1	**3**	1	0
C	0	1	0	1	1	0	0	0	2	1	2	**4**	2
A	0	0	0	0	0	2	0	1	0	1	0	2	**5**
A	0	0	0	0	0	1	1	1	0	0	0	0	3

- Score = 5.
- Size for the sequences: 8 and 12.
- Size of DP matriz: 9 x 13 elements.



Data dependencies in rows & columns

Time line for multi-GPU sequence alignment (chr13 and chr22 input data sets)



III. CUDAlign

Summary of CUDAlign versions

Version	Major contributions
1.0	Compares on GPUs sequences of unrestricted size using the affine gap model of SW. It provides score and end coordinates of the optimal alignment, but not the full alignment
2.0	Incorporates the Myers-Miller algorithm to retrieve the full alignment of two sequences in linear space
2.1	Improvements on six stages: 1-3 run on GPUs, 4-6 run on CPUs
3.0	Multi-GPU for SW phase 1 to distribute the DP matrix, and overlap computations with communications to the CPU
4.0	Multi-GPU for SW phase 2, with Pipeline Traceback (PT) and Incremental Speculative Traceback **(IST) to estimate points where optimal alignment crosses border columns**
MASA	Multi-platform Architecture for Sequence Aligner, enabling versions to run on: (1) a serial CPU, (2) multicore CPU using OmpsSs, (3) many-core GPUs using CUDA, and (4) Xeon Phi using OpenMP

Summary of CUDAlign stages

Stage	Description	Phase	Who
1	Obtains the optimal score	1	GPU
2	Partial traceback	2	GPU
3	Splitting partitions	2	GPU
4	Myers-Miller with balanced splitting and orthogonal execution	2	CPU
5	Obtaining the full alignment	2	CPU
6	External visualization (optional)	2	CPU

Comparison with:

Same device, different SW

Same SW, different device

Length of sequences	CUDAlign 3.0 (1) GPU GeForce GTX980	SW # (2) GPU GeForce GTX980	CUDAlign 3.0 (3) FPGA Altera Stratix V	
10K x 10K	0.03	0.3	14.47	FPGAs better
57K x 57K	1.08	7.62	30.35	
162K x 172K	8.18	33.33	33.45	
543K x 536K	45.89	64.53	35.51	GPUs better
1M x 1M	79.21	75.24	36.52	
3M x 3M	84.05	69.54	37.32	
5M x 5M	160.79	120.92	37.49	
7M x 5M	84.43	68.84	37.56	
10M x 10M	163.77	118.81	37.61	
23M x 25M	84.84	67.55	37.67	

- Performance in GCUPS (Giga-Cell Updates Per Second)
- (1) E. Sandes, G. Miranda, A. Melo, X. Martorell, E. Ayguadé. "CUDAlign 3.0 Parallel Biological Sequence Comparison in Large GPU Clusters". In CCGRID, pages 160-169 (2014).
- (2) M. Korpar, M. Sikic. "SW# - GPU-enabled exact alignments on genome scale". Bioinformatics, 29(19): 2494-2495 (2013)
- (3) E. Rucci, C. García, G. Botella, A. de Gusti, M. Naiouf, M. Prieto-Matías. "Accelerating Smith-Waterman Alignment on Long DNA Sequences with OpenCL on FPGA". In IWBBIO'17, pages 500-511, vol. II (2017)

IV. Experimental setup

Platforms used

Hardware resources	Nvidia GPUs used		Intel CPU
Commercial model	GeForce GTX 980	Titan Pascal	Xeon E5-2620
Number of cores	2048	3584	8
Cores frequency	1126 MHz	1531 MHz	2100 MHz
Memory size and family	4 Gbytes GDDR5	12 Gbytes GDDR5X	64 Gbytes DDR4
Memory frequency	7 GHz	10 GHz	2.4 GHz
Memory width	384 bit	384 bits	256
Memory bandwidth	336 GB/s.	480 GB/s.	76.8 GB/s.
Software installed	CUDA 8.0	CUDA 8.0	Ubuntu 14.04 LTS 64 bits

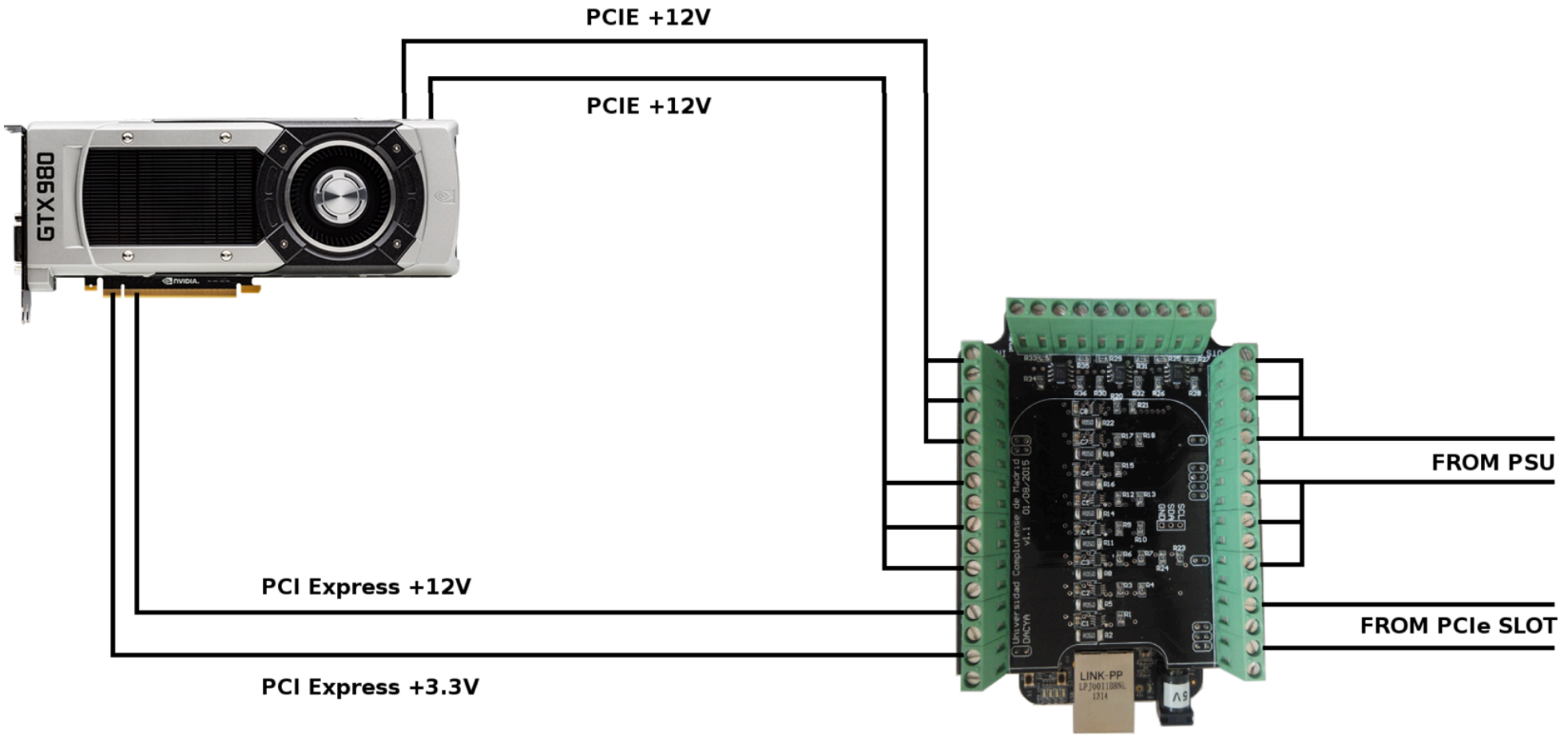
Input data set

- Real DNA sequences coming from the National Center for Biotechnology (NCBI) database. Comparison all human (GRCh37) and Chimpanzee (panTro4) homologous chromosomes. Among the 25 pairs of sequences, we have chosen:

Input seq.	Size		Peta Cells	Score	Length	Coverage	Matches	Mismatches	Gaps
	Human	Chimp.							
chr22	51M	50M	2.55	31.510.791	51.929.087	98.9%	88.5%	3.8%	7.7%
chr21	48M	46M	2.24	36.006.054	48.579.349	99.0%	91.9%	1.1%	7.1%
47M	47M	33M	1.54	27.206.434	33.583.457	70.5%	94.4%	1.5%	4.1%
chrY	59M	26M	1.56	1.394.673	2.283.191	6.0%	88.1%	2.0%	10.0%

Monitoring energy

- Beagleblone Black open-source hardware.
- Accelpower module with 8 sensors:



V. Experimental results

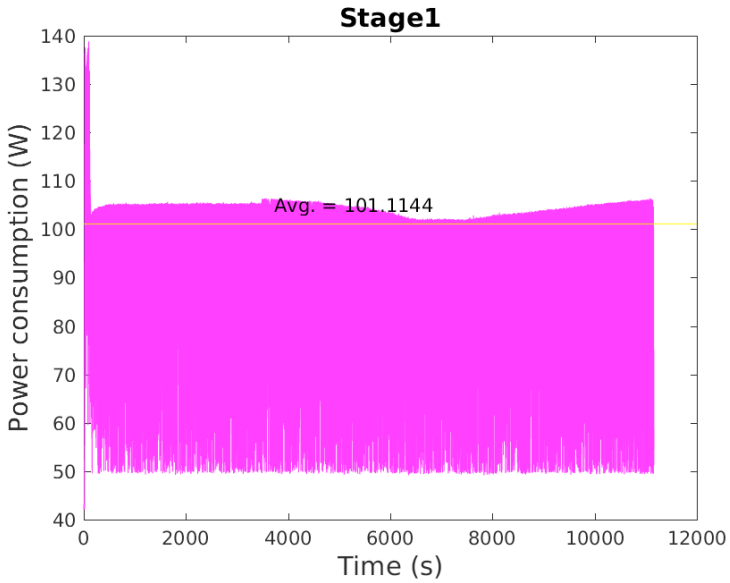
Power, time and energy on four GTX980 GPUs

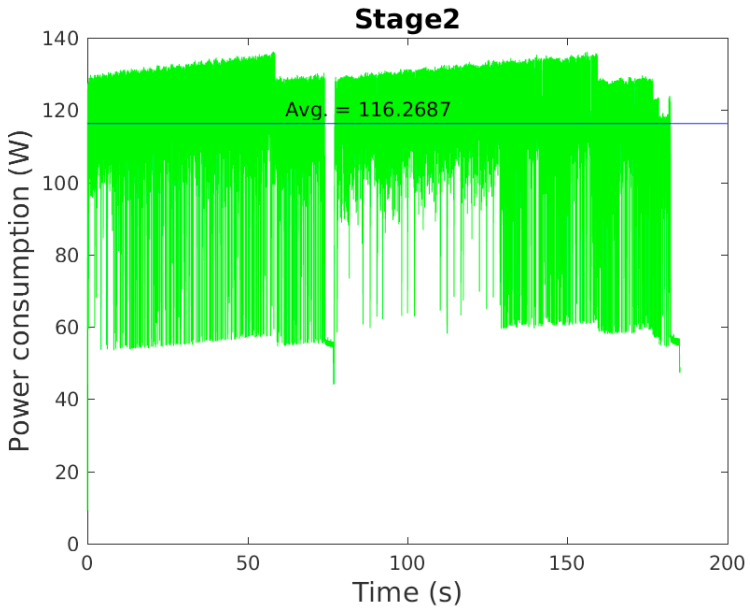
Sequence	Stage 1	Stage 2	Stage 3		
Average power (watts per GPU)					
chr22	101.11 W.	116.26 W.	77.27 W.		
chr21	102.11 W.	116.47 W.	78.89 W.		
47M	104.37 W.	117.12 W.	76.33 W.		
chrY	103.25 W.	119.63 W.	0.00 W.		
Execution time (seconds)				**Total time**	
chr22	11161.92 s.	185.20 s.	14.25 s.	11361.38 s.	
chr21	9687.36 s.	61.49 s.	11.03 s.	9759.89 s.	
47M	6694.95 s.	88.25 s.	9.05 s.	6792.26 s.	
chrY	6798.12 s.	3.99 s.	0.00 s.	6802.11 s.	
Energy consumption (kilojules per GPU)				**Total energy**	**Total cost**
chr22	1128.63 kJ.	21.53 kJ.	1.10 kJ.	1151.27 kJ.	0.1660 €
chr21	989.26 kJ.	7.16 kJ.	0.87 kJ.	997.29 kJ.	0.1440 €
47M	698.82 kJ.	10.34 kJ.	0.69 kJ.	709.85 kJ.	0.1024 €
chrY	701.94 kJ.	0.48 kJ.	0.00 kJ.	702.42 kJ.	0.1012 €

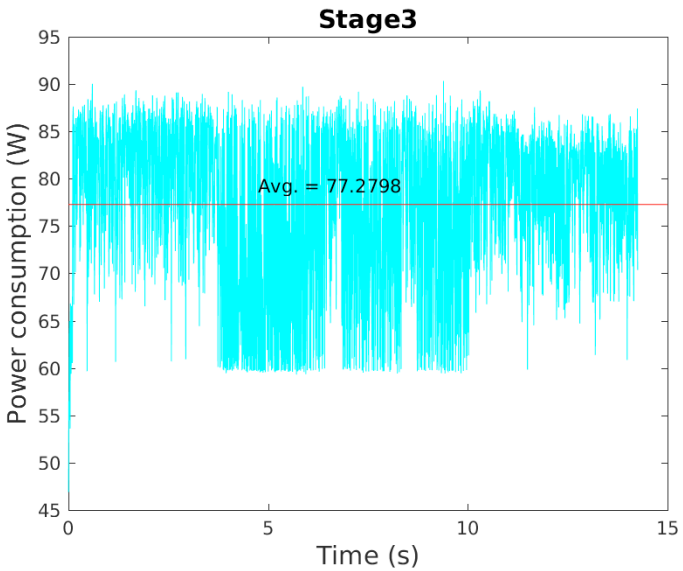
Power, time and energy for chr22 on multi-GPU

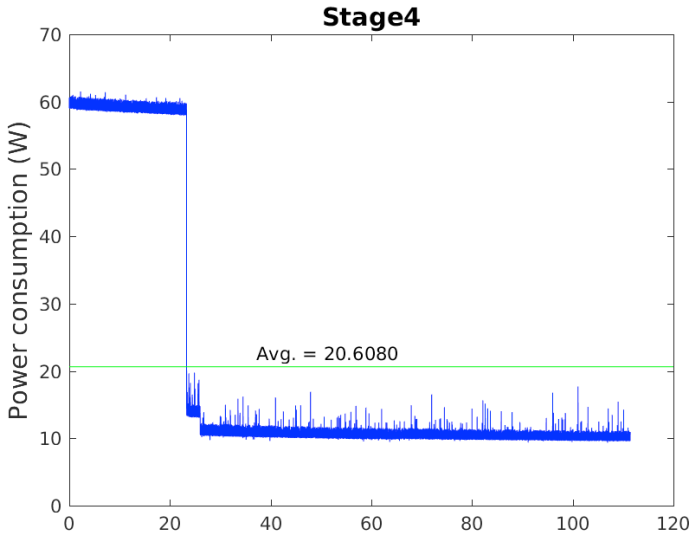
No. GPUs	Stage 1	Stage 2	Stage 3		
Average power (watts per GPU)					
4	101.11 W.	116.26 W.	77.27 W.		
3	101.53 W.	108.16 W.	78.79 W.		
2	100.30 W.	114.68 W.	76.74 W.		
1	102.95 W.	114.44 W.	81.27 W.		
Execution time (seconds)				**Total time**	
4	11161.92 s.	185.20 s.	14.25 s.	11361.38 s.	
3	14719.32 s.	253.72 s.	17.70 s.	14990.76 s.	
2	22080.04 s.	159.77 s.	23.17 s.	22262.99 s.	
1	22302.24 s.	291.50 s.	46.65 s.	22640.40 s.	
Energy consumption (kilojules per GPU)				**Total energy**	**Total cost**
4	1128.63 kJ.	21.53 kJ.	1.10 kJ.	1151.27 kJ.	0.1660 €
3	1494.60 kJ.	27.45 kJ.	1.40 kJ.	1523.44 kJ.	0.1650 €
2	2214.77 kJ.	18.32 kJ.	1.78 kJ.	2234.88 kJ.	0.1614 €
1	2296.22 kJ.	33.36 kJ.	3.79 kJ.	2333.37 kJ.	0.0842 €

Power consumption on 4 GTX 980 GPUs stage by stage







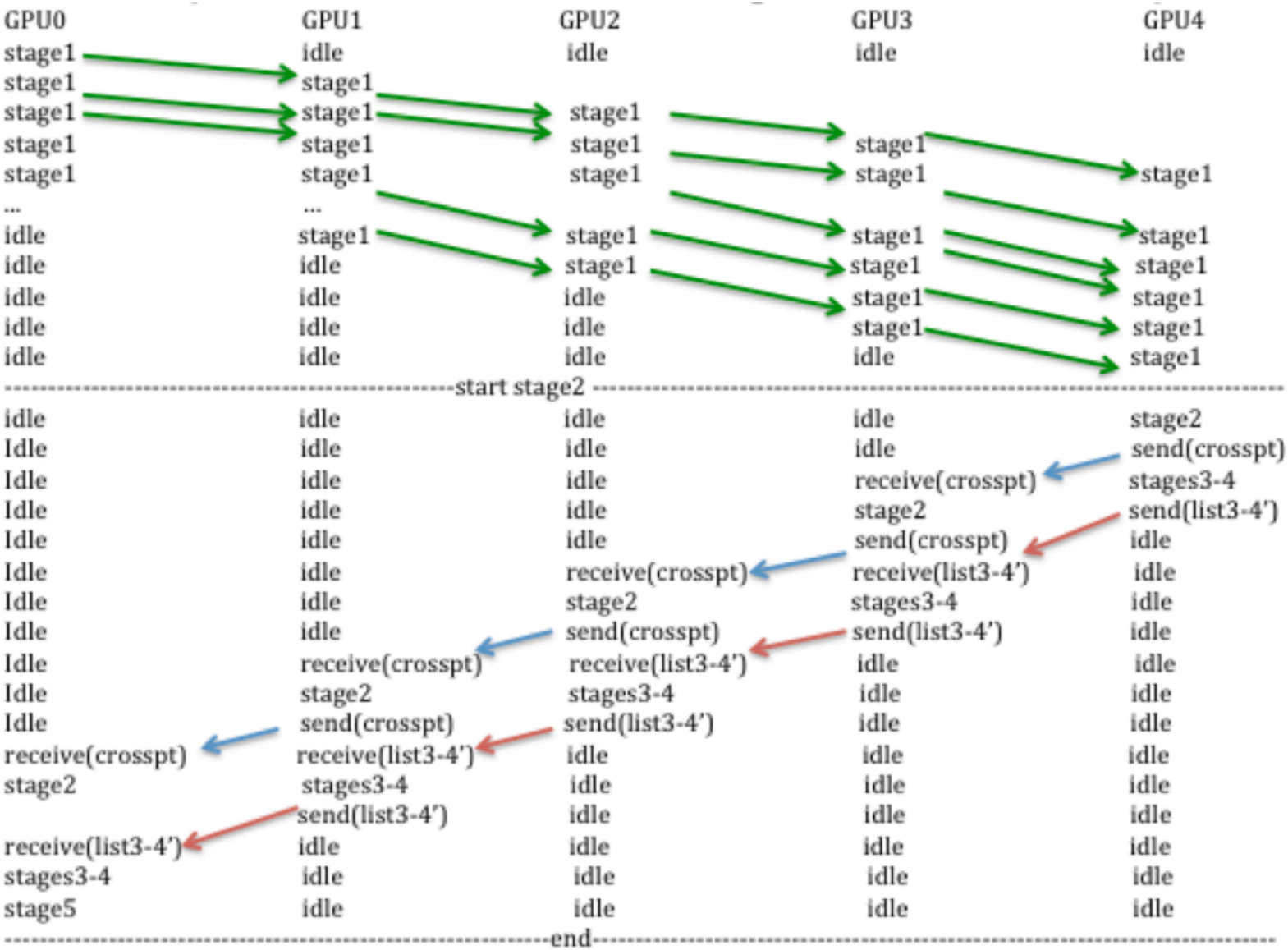


Time savings and energy penalties (chr22)

	Stage 1		Stage 2		Stage 3		Total	
No. GPUs	Savings (time)	Penalty (energy)	Savings (time)	Penalty (energy)	Savings (time)	Penalty (energy)	Savings (time)	Penalty (energy)
Four	49.96%	96.60%	36.47%	158.15%	69.46%	6.09%	49.82%	97.35%
Three	34.01%	95.26%	12.97%	146.85%	62.06%	0.81%	33.79%	95.86%
Two	1.00%	92.90%	45.20%	9.83%	50.34%	-6.07%	1.67%	91.55%

VI. Speculative executions

Speculate to anticipate results on idle GPUs

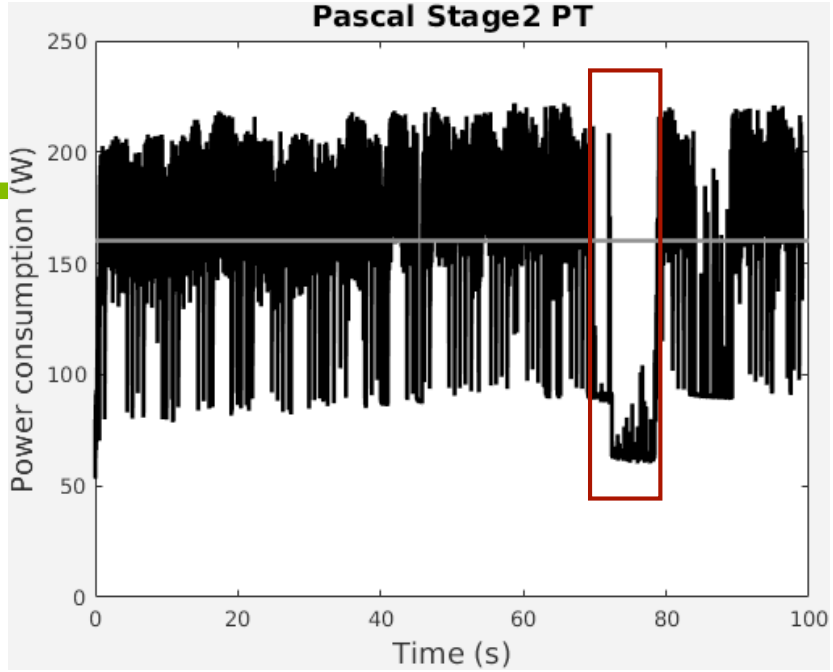


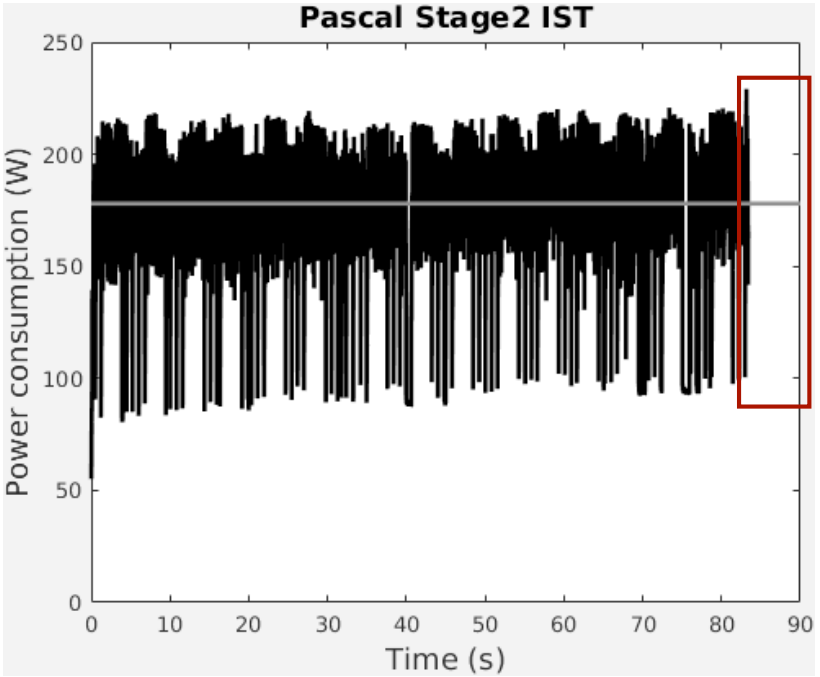
Benefits on 2 Titan Pascal: Stage 2 of Smith-Waterman

Execution	Time	Avg. power	Energy
Regular (PT)	99.04 s.	160.21 W	15867.19 J.
Speculative (IST)	83.39 s.	177.97 W	14840.91 J.
Comparison	-18%	+11%	-6.5%

Speculation	Time	Avg. power	Energy
Hit	83.39 s.	177.97 W	-6.5%
Miss	99.04 s.	177.97 W	+11%

- We need a 2 hit/miss ratio to waive energy penalties.
- And we would save 12% of execution time on average.





Regular versus speculative execution on GPU stages (1-3)

Stage	Time (s.)		Average power (W.)		Energy (kJ.)	
	Regular	Speculative	Regular	Speculative	Regular	Speculative
1	8743.63		164.19		1435.61	
2	99.04	83.39	160.21	177.97	15.86	14.84
3	18.55	18.97	103.92	105.52	1.92	2.00
Total	8861.22	8845.99	164.01	164.19	1453.39	1452.45

- Speculative executions gain in both:
  - Total execution time.
  - Total energy required.
- Benefits are much higher on large clusters of GPUs.
  - See paper (1) on page 15.

Summary

- CUDAlign comprises 6 stages, 1-3 are GPU-accelerated and 1 takes the bulk of computational time.
- Power consumption keeps stable across different alignment sequences, but we have seen deviations of up to 30% across different stages.
- Energy costs decrease on high number of GPUs.
- Speculative execs save energy from a 2/1 hit/miss ratio on.
- Good correlation between performance and extra energy, even where multi-GPUs do not show great scalability.
- **On multi-GPU, energy scales better than speed-up.**

Thanks!

Acknowledgements: Funds & sponsors

- Local. University of Malaga (Spain).
  - Support for presenting the paper here.
- Regional. Junta de Andalucía:
  - Research funds: Project of Excellence P12-1741 (2014-17).
- National. Ministry of Education of Spain:
  - Research funds: Project TIN2013-42253-P (2013-17).
- International. Nvidia corporation:
  - Travel expenses.
  - GTC invitation.

- QUESTIONS?