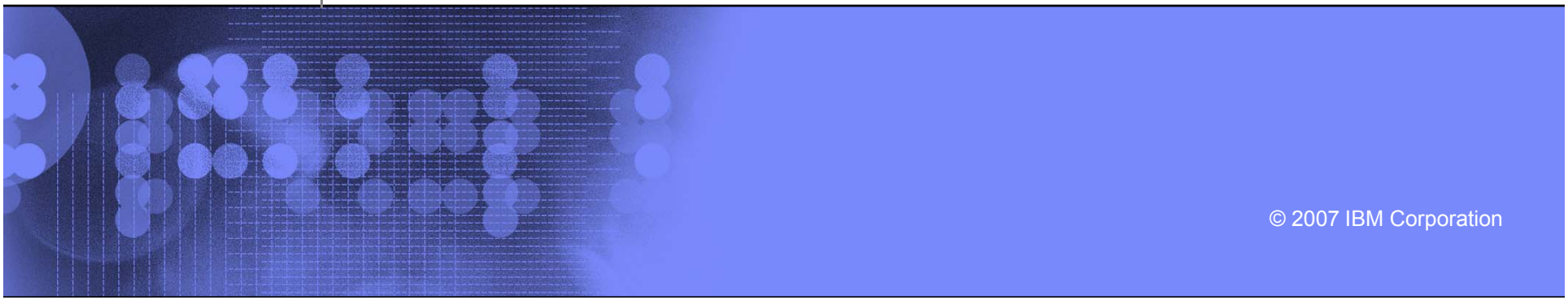
Biological Sequence Analysis on the Cell/B.E. -- HMMer-Cell

Kursad Albayraktaroglu (Univ. of Maryland, College Park), **Jizhu Lu**, Michael Perrone (IBM T.J. Watson Research Center), Manoj Franklin (Univ. of Maryland, College Park)



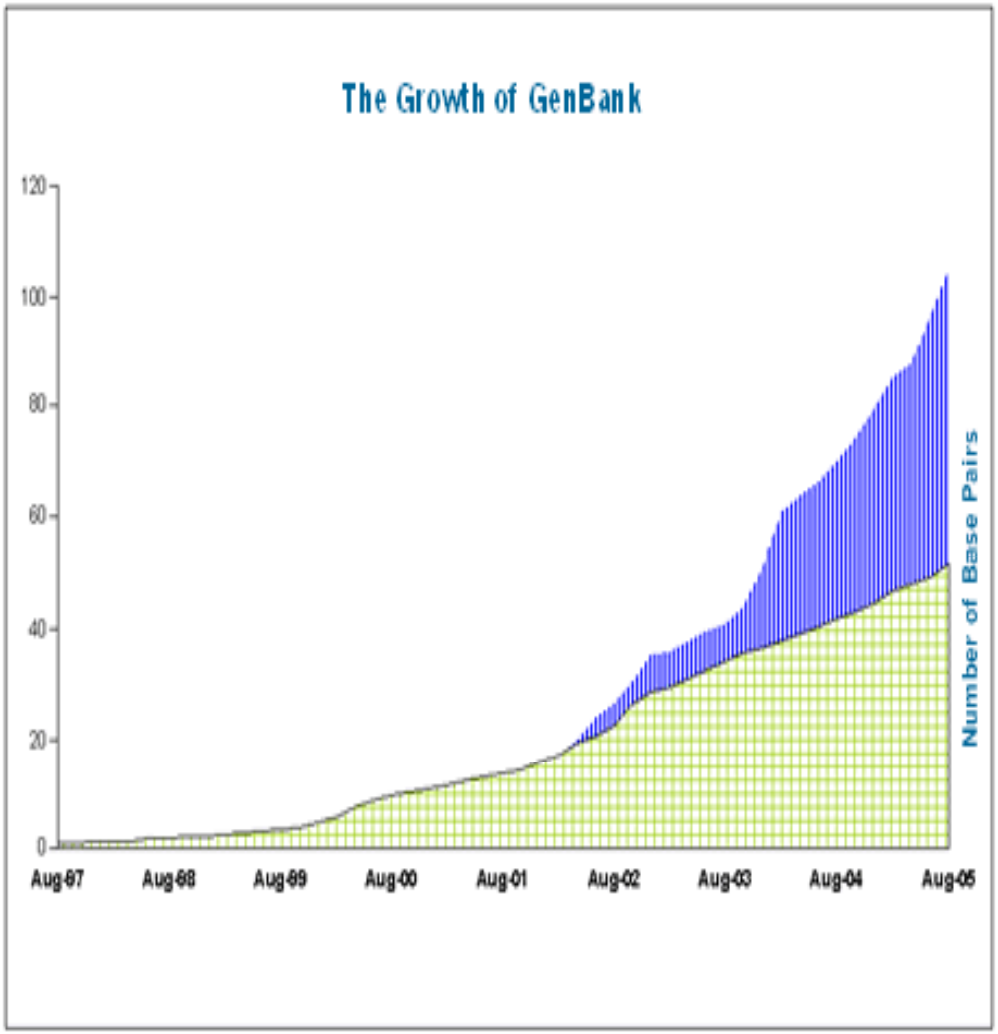
Outline

- Introduction
- Hidden Markov Models in Bioinformatics
- Parallelization Strategy
- Experiment & Results
- Conclusion & Future Work

Introduction

- HMMER: a widely used bioinformatics application software package
- Hmmsearch is one of 9 independent programs
- Hmmsearch searches a HMM profile against a sequence database and returns the most closely related sequences with higher scores
- Hmmsearch could be time consuming due to the huge volume of sequence DB.

Nucleotide Data Grow Exponentially



According to news from NCBI, GenBank, EMBL and DDBJ exceeds 100 billion base pairs!

Hidden Markov Models

HMM is a system M = (∑,Q,A,e)

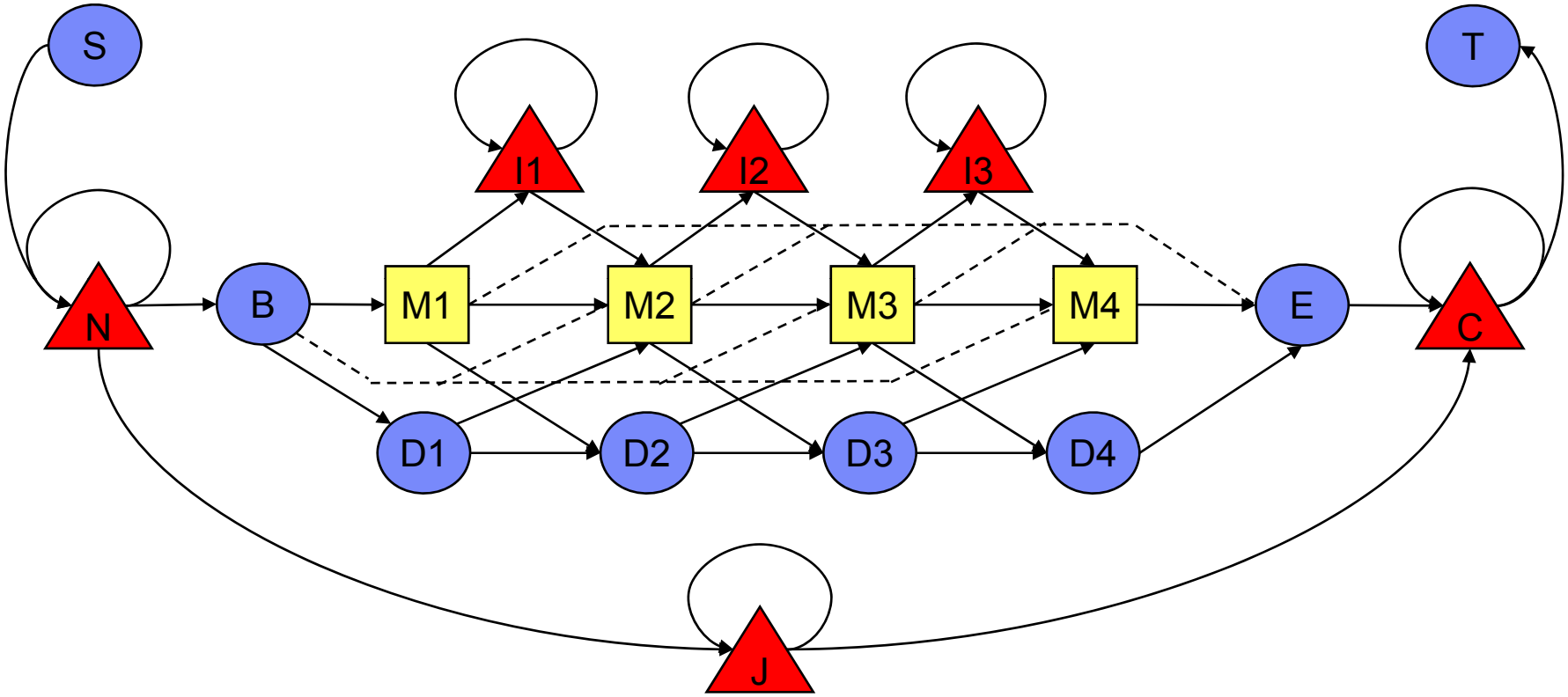
Where ∑: a set of alphabet

Q: a set of states

A = $\{a_{kl}\}$: a matrix of transition probability a_{kl} for k, l ℳ Q

e: an emission probability $e_k(b)$ for every k ℳ Q and b ℳ ∑

Hidden Markov Models in Bioinformatics

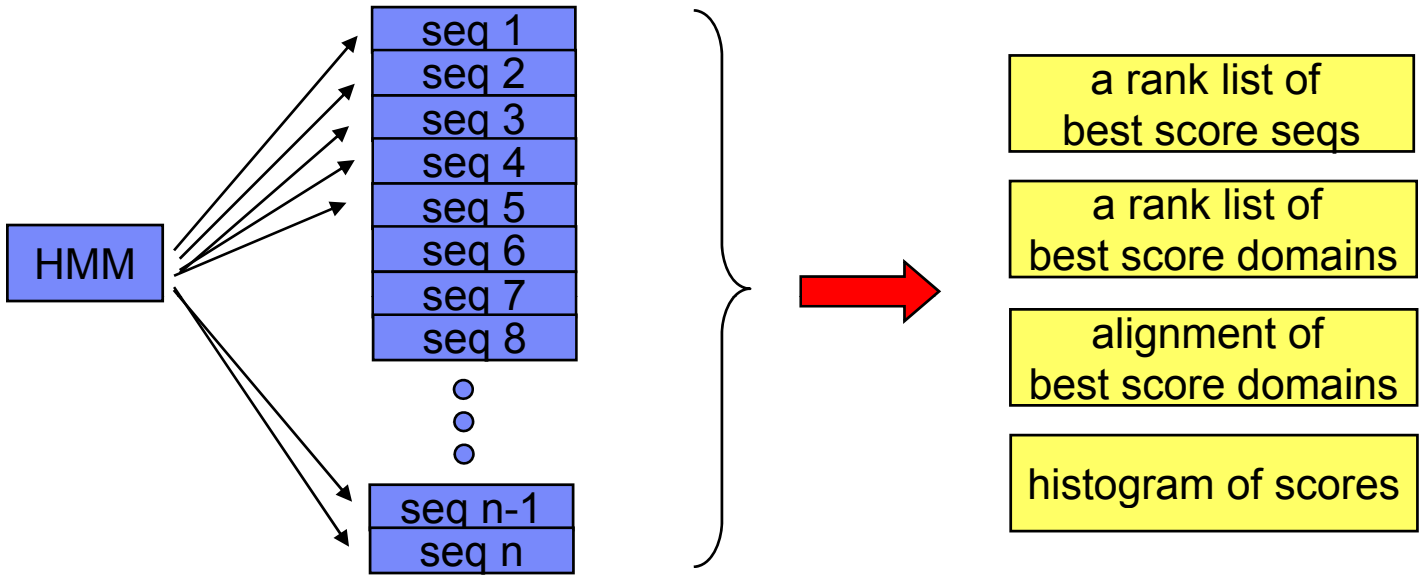


M:Match, D:Delete, I:Insert, B:Begin, E:End, S and T:beginning and end points of HMM

N and C:states which can emit non-motif sequences before and after the motif,

J:state can emit non-motif sequences between two copies of the motif

About HMMSearch

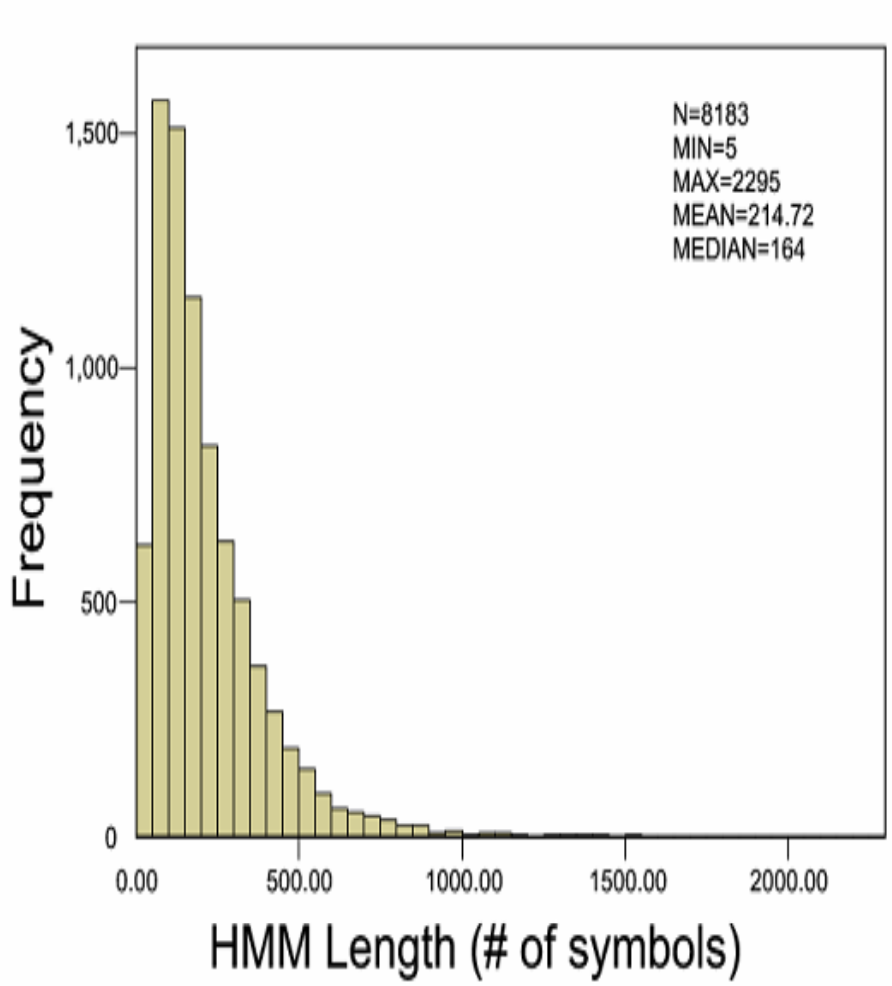


- Computationally intensive workload, 98% of execution time spent in a single function (Viterbi algorithm)
- Abundant parallelism
- Ideally suited for the Cell BE
- CPU bound

Challenges

- Viterbi algorithm on a Plan7 HMM requires memory in the order of O(N*M), and so for a typical HMM of length 200 (N), only sequences less than 1250 characters (M) could be processed without leaving any space in LS for the program!

Some Analysis



- 99.5% HMM in PFAM < 982 symbols, and the median is 164 symbols
- Only <0.1% sequences require a full traceback for commonly used threshold.
- The space complexity of Viterbi algorithm could be reduced from O(N*M) to O(N)

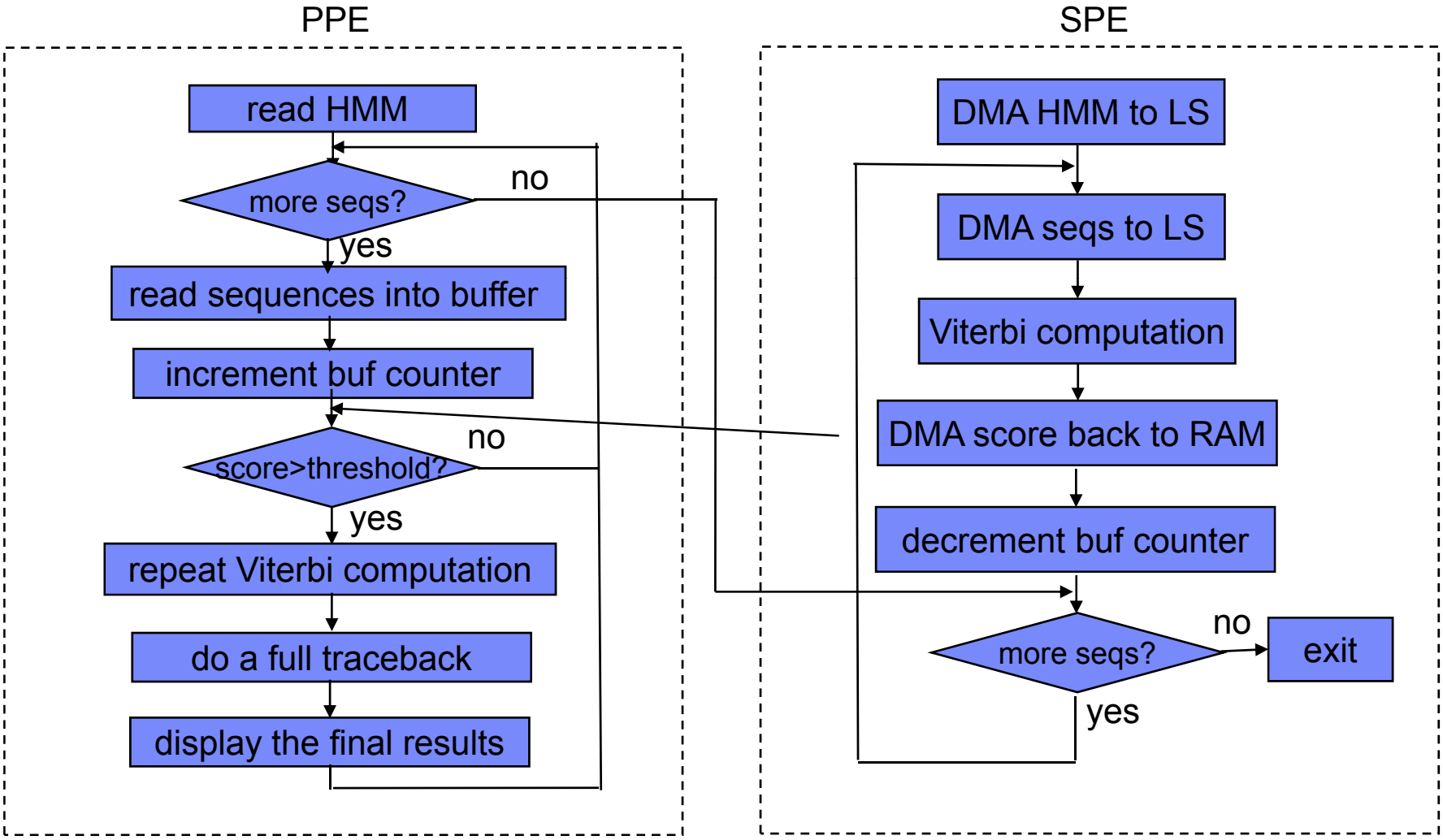
Solution

- SPE will perform Viterbi algorithm without keeping intermediate results, but just calculate the score and DMA the score back to PPE
- PPE will check whether a score is above the threshold, and if yes it will repeat the Viterbi computation and perform a full traceback

Parallelization Strategy

- “Manager-Workers” parallel design pattern
- “Double Buffering”
- “Huge page”
- Hand SIMDization

How Does HMMER-Cell Work on Cell?



Experiment Methodology

- Compared to 8 x86 configurations
- 6 HMMs (N=100..500), completed SwissPROT (250,000 seqs)
  - Maf1 (N=200)
  - COQ7 (N=100)
  - GerA (N=500)
  - Lipoprotein_1 (N=300)
  - APG9 (N=400)
  - Arfaptin (N=245)

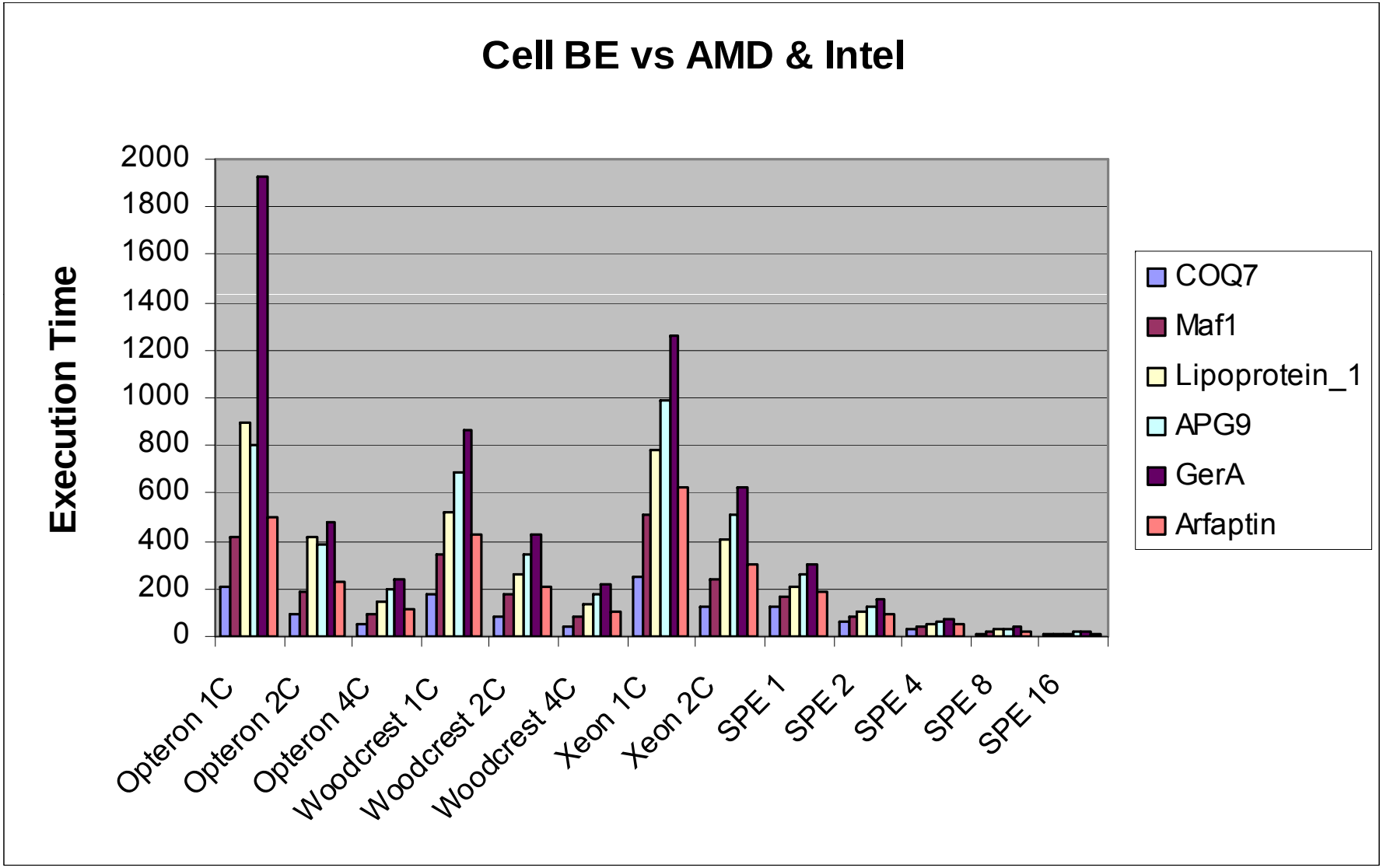
Experiment Environment

- QS21:
  - Cell BE 3.2GHz, 2GB RAM, 2GB swap
- Intel Xeon:
  - 2 x Dual-core processors, 3.0GHz, 4MB Cache, 8GB RAM (x1GB DIMMS), 4GB swap
- Intel Woodcrest:
  - 2 x Dual-core processors, 3.0GHz, 4MB Cache, 8GB RAM (x1GB DIMMS), 4GB swap
- AMD Opteron:
  - 2 x Dual-core Opteron 2220 SE, 2MB L2-Cache, 8GB RAM (x1GB DIMMS), 1GB swap

Experiment & Results (1)

SwissProt DB	COQ7	Maf1	Lipoprotein_1	APG9	GerA	Arfaptin
Opteron 1-core	206.57	413.46	893.23	806.18	1930.62	503.84
Opteron 2-core	97.47	187.8	412.86	383.22	475.06	234.09
Opteron 4-core	48.88	96.03	149.23	197.76	238.65	119.35
Woodcrest 1-core	176.82	346.48	520.86	689.03	862.8	424.7
Woodcrest 2-core	88.42	171.91	260.74	343.62	427.99	209.89
Woodcrest 4-core	44.5	86.77	131.87	175.37	219.33	106.13
Xeon 1-core	247.7	513.94	786.4	985.43	1260.68	624.81
Xeon 2-core	125.49	243.37	404.59	510.04	624.19	303.18
SPE 1	121.83	167.11	212.55	257.95	303.3	187.63
SPE 2	61.01	83.62	106.29	129.07	151.71	93.91
SPE 4	30.57	41.84	53.23	64.55	75.92	47
SPE 8	15.3	20.95	26.64	32.33	38.01	23.54
SPE 16	7.92 (*)	10.83 (*)	13.35	16.23	19.07	12.04

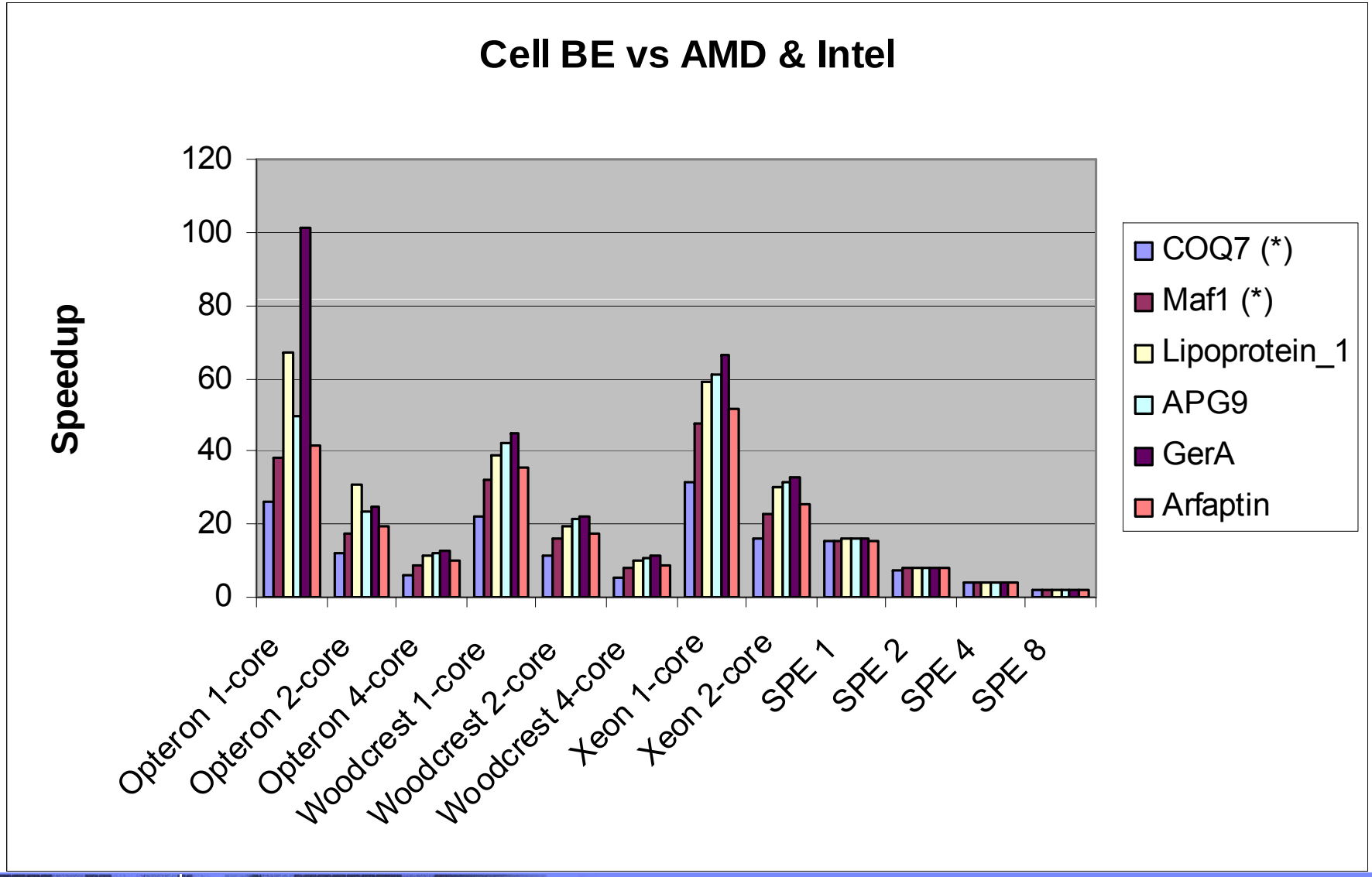
Experiment & Results (2)



Experiment & Results (3)

SwissProt DB	COQ7 (*)	Maf1 (*)	Lipoprotein_1	APG9	GerA	Arfaptin
Opteron 1-core	26.08	38.18	66.91	49.67	101.24	41.85
Opteron 2-core	12.31	17.34	30.92	23.61	24.91	19.44
Opteron 4-core	6.17	8.87	11.18	12.18	12.51	9.91
Woodcrest 1-core	22.31	31.99	39.02	42.45	45.23	35.27
Woodcrest 2-core	11.16	15.87	19.53	21.17	22.44	17.43
Woodcrest 4-core	5.62	8.01	9.88	10.81	11.5	8.81
Xeon 1-core	31.28	47.46	58.91	60.72	66.11	51.9
Xeon 2-core	15.85	22.47	30.31	31.43	32.73	25.18
SPE 1	15.38	15.43	15.92	15.89	15.9	15.58
SPE 2	7.7	7.72	7.96	7.95	7.96	7.8
SPE 4	3.86	3.86	3.99	3.98	3.98	3.9
SPE 8	1.93	1.93	1.996	1.99	1.99	1.96

Experiment & Results (4)



Experiment & Results (5)

HMMER-Cell Scalability on Cell BE

SPE 1	1	1	1	1	1	1
SPE 2	1.996	1.998	2	1.999	1.999	1.998
SPE 4	3.985	3.994	3.993	3.996	3.995	3.992
SPE 8	7.96	7.977	7.979	7.979	7.98	7.97
SPE 16	15.38	15.43	15.92	15.893	15.91	15.584

Results Summary

- Using all 16 SPEs, Cell-HMMER is up to 32.7x faster than dual-core Intel Xeon, 11.5x than 2-socket dual-core Intel Woodcrest and 12.5x than 2-socket dual-core Opteron
- Linear Scalability
- Should beat new quad-core x86 processors in the market (Clovertown tests under way)

Conclusion & Future Work (1)

- Perfectly linear scaling for large number of sequences
- Speedup improves as HMM size increases: less contention for sequence buffer as Viterbi operations take longer.
- Speedup with even 1 SPE

Conclusion & Future Work (2)

- Cell-HMMER outperforms HMMER on current dual-core x86 processors well
- The Cell BE architecture offers the potential of significant performance improvements for workloads that can utilize its features effectively.
- The low memory latency of the SPE local stores and the high memory bandwidth of the Cell BE architecture can be very effective.

Conclusion & Future Work (3)

- Explore performance on larger Sequence DB
- Enhance Cell-HMMER with a more efficient buffering and synchronization scheme

Thanks for your time!

Any Questions?