Computing Large-Scale Alignments on a Multi-Cluster

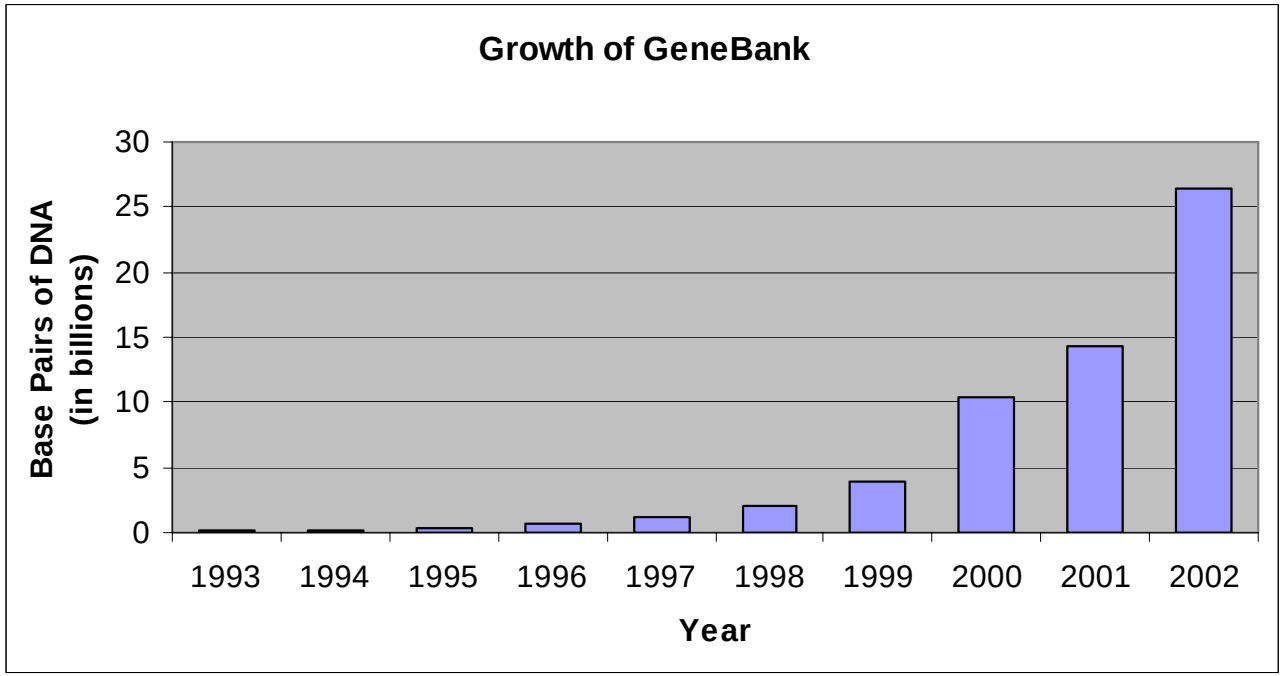
Chunxi Chen and Bertil Schmidt

Nanyang Technological University

Contents

- Motivation
- Smith-Waterman Algorithm
- Multi-Cluster Platform
- Parallelization
- Performance Evaluation
- Conclusions

Motivation



- Genetic sequence databases are growing exponentially
- Growth rate will continue, since multiple concurrent genome projects have begun with more to come

Motivation

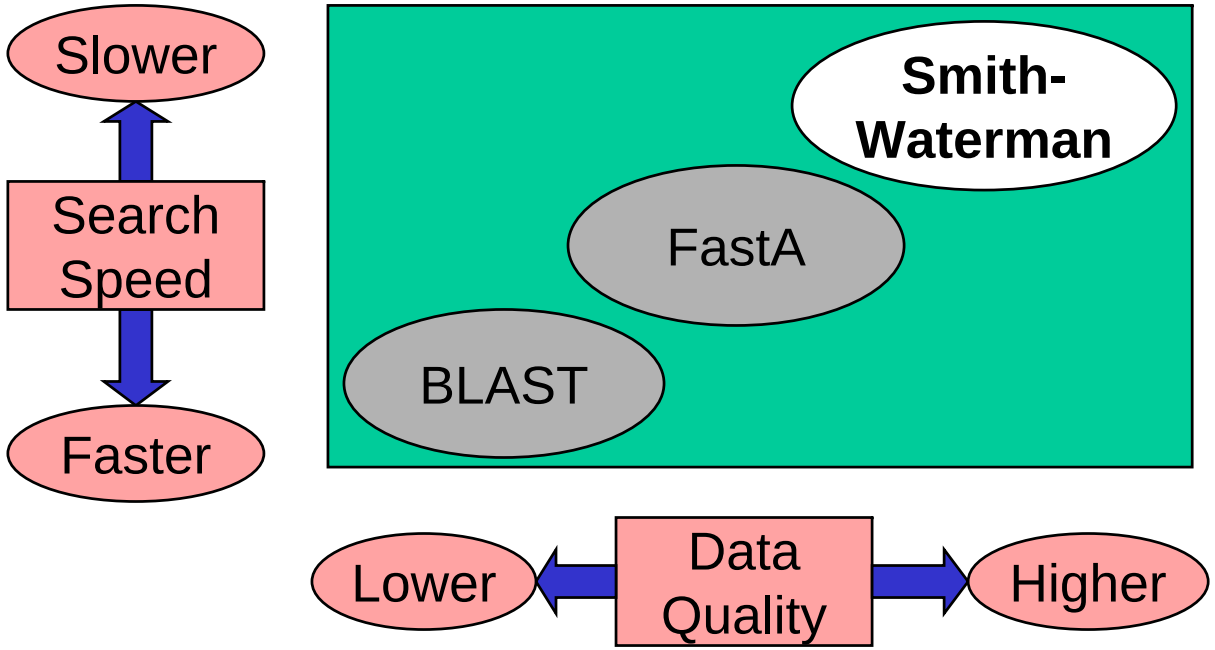
- Discovered sequences are analyzed by comparison with sequence databases
- Complexity of sequence comparison is proportional to the product of query size times database size
- ⇒ **Analysis too slow on sequential computers**
- There are two approaches:
  - Heuristics, e.g. BLAST, but the more efficient the heuristics, the worse the quality of the results
  - Parallel Processing, get high quality results in reasonable time

Motivation

- Number of completely sequenced genomes:
  - 105 microbial, 10 Eukaryotic
  - ~ 10 nearly completed eukaryotic genomes
  - > 300 genome projects planned or underway (http://www.nslij-genetics.org/seq/)
- Closely related genomes
  - Eukaryotic example: C. elegans and C. briggsae
  - Prokaryotic example: E. coli K12 and O157:H7
- What's Common and What's Unique
  - Identify unique & important proteins in pathogens as targets
  - Identify Genes and Predict Function based on Synteny
  - Susceptibility to disease (SNPs)

Sequence Alignment

- Several algorithms: BLAST, FastA, Smith-Waterman



Smith-Waterman Algorithm

- Optimal local alignment of two sequences
- Performs an exhaustive search for the optimal local alignment
  - Complexity $O(n \times m)$ for sequence lengths n and m
- Based on the 'dynamic programming' (DP) algorithm
  - Fill the DP matrix using a substitution (mutation) matrix
  - Find the maximal value (score) in the matrix
  - Trace back from the score until a 0 value is reached

Smith-Waterman Algorithm

- Aligning *S*1 and *S*2 of length *l*1 and *l*2 using Recurrences:

$$H(i,j)=\max\begin{cases}0\\ E(i,j)\\ F(i,j)\\ H(i-1,j-1)+Sbt(S1_i,S2_j)\end{cases} \qquad ,1\le i\le l1, 1\le j\le l2$$

$$H(i,0)=E(i,0)=0$$
$$H(0,j)=F(0,j)=0$$

$$E(i,j)=\max\begin{cases}H(i,j-1)-\alpha\\ E(i,j-1)-\beta\end{cases}, \quad F(i,j)=\max\begin{cases}H(i-1,j)-\alpha\\ F(i-1,j)-\beta\end{cases}$$

- Calculate three possible ways to extend the alignment
  - by one AminoAcid (AA) in each sequence
  - by one AA in the first sequence and align it with a gap in the second
  - by one AA in the second sequence and align it with a gap in the first

Smith-Waterman Algorithm

Align S1=**ATCTCGTATGATG** S2=**GTCTATCAC**

$$Sbt(x,y)=\begin{cases}2 \text{ if } (x=y)\\ -1 \text{ else}\end{cases}$$

α=1, β=1

	∅	A	T	C	T	C	G	T	A	T	G	A	T	G
∅	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	2	1	0	0	2	1	0	2
T	0	0	2	1	2	1	1	4	3	2	1	1	3	2
C	0	0	1	4	3	4	3	3	3	2	1	0	2	2
T	0	0	2	3	6	5	4	5	4	5	4	3	2	1
A	0	2	2	2	5	5	4	4	7	6	5	6	5	4
T	0	1	4	3	4	4	4	6	5	9	8	7	8	7
C	0	0	3	6	5	6	5	5	5	8	8	7	7	7
A	0	2	2	5	5	5	5	4	7	7	7	10	9	8
C	0	1	1	4	4	7	6	5	6	6	6	9	9	8

$$H(i,j)=\max\begin{cases}0\\ H(i-1,j)-1\\ H(i,j-1)-1\\ H(i-1,j-1)+Sbt(S1_i,S2_j)\end{cases}$$

A T C T C G T A T G A T G
G T C – T A T C A C

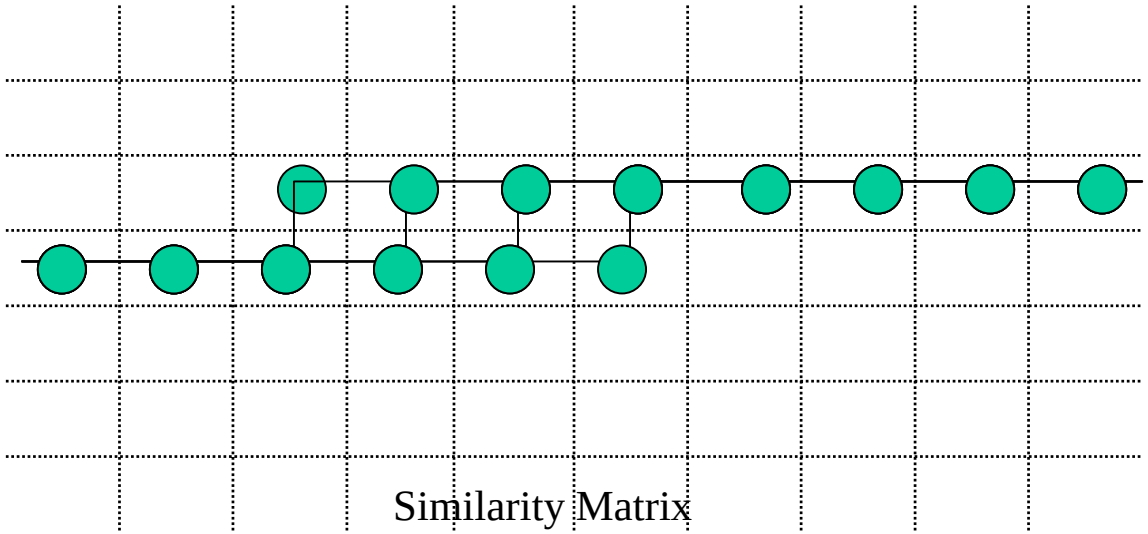
Nanyang Technological University

Smith-Waterman Algorithm

- Aligning two sequences of length *N* and *M*
  - Time Complexity: O(*N*×*M*)
  - Memory Complexity: O(*N*×*M*)
- Example
  - aligning *m. pneumoniae* (816,394 nucleotides) and *m. genitalium* (580,074 nucleotides)
  - around 500-G Cells in the Similarity Matrix

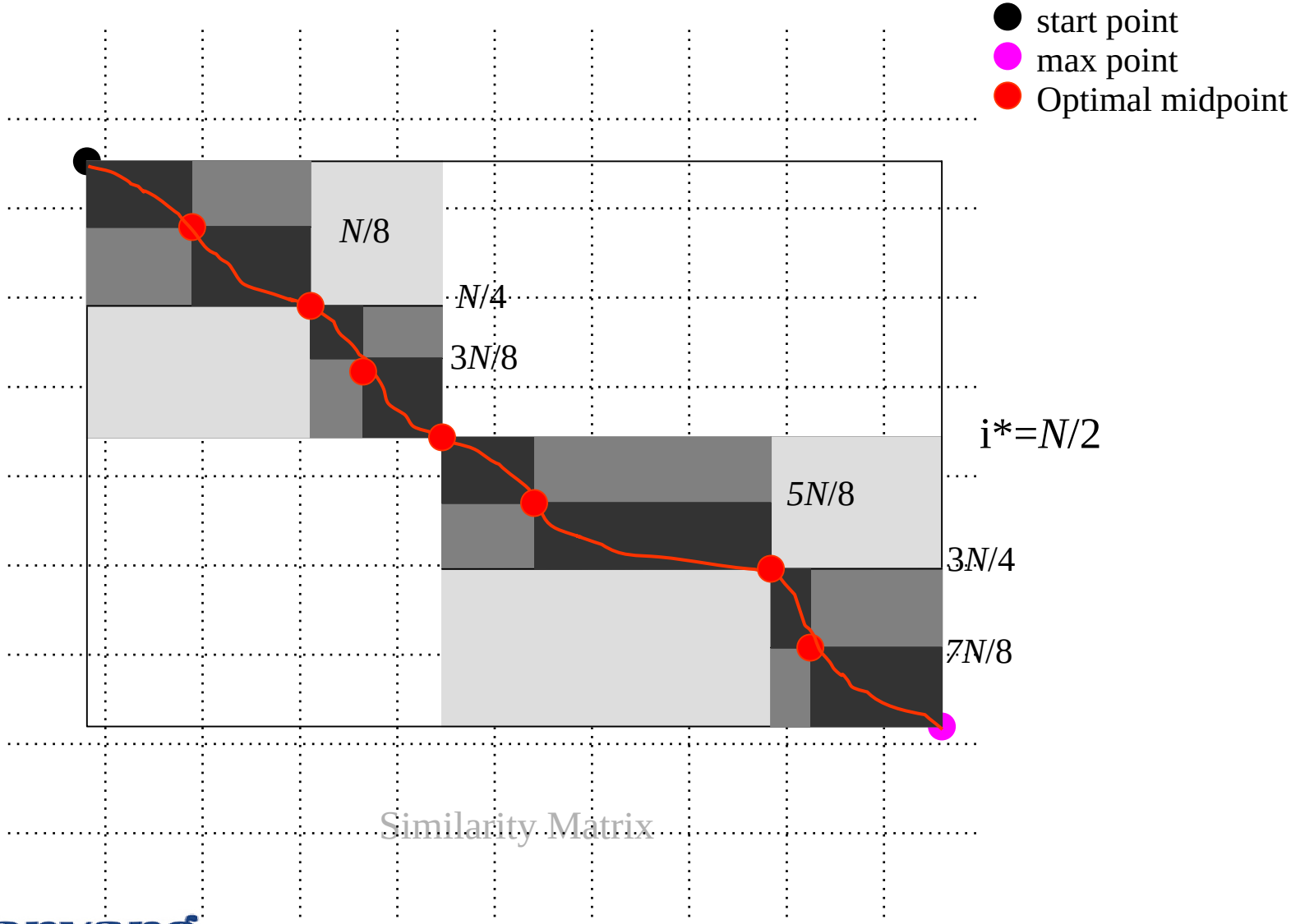
Computing Maximal Score in Linear Space

Idea: Store only one row in Similarity Matrix



- Compute maximal score in similarity matrix
- Computing the path (traceback) requires additional computation

Traceback in Linear Space

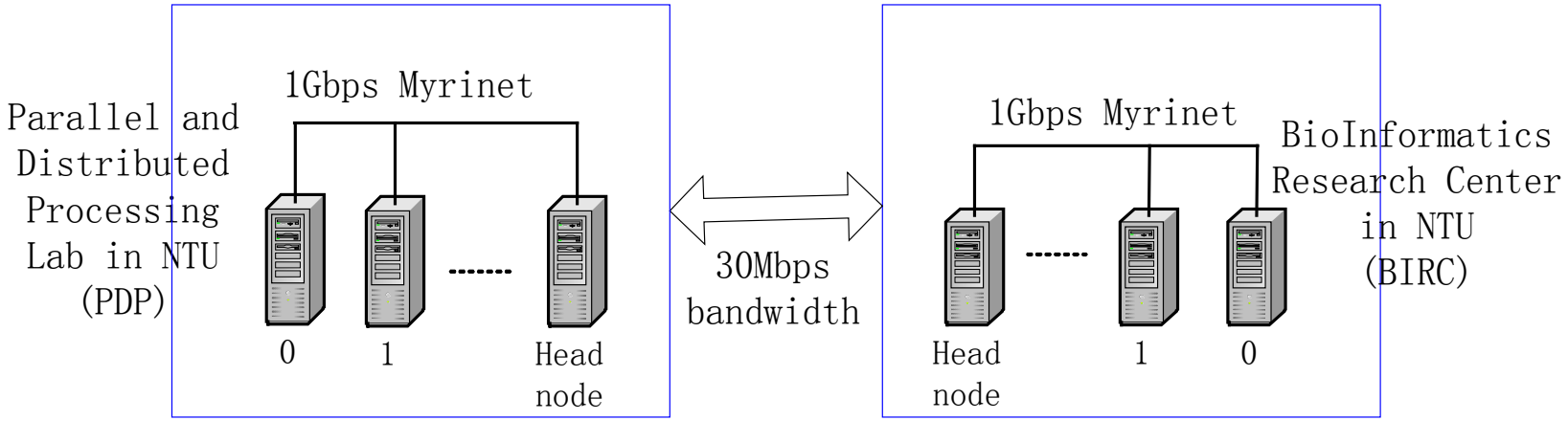


Finding Near-Optimal Non-Overlapping Alignments

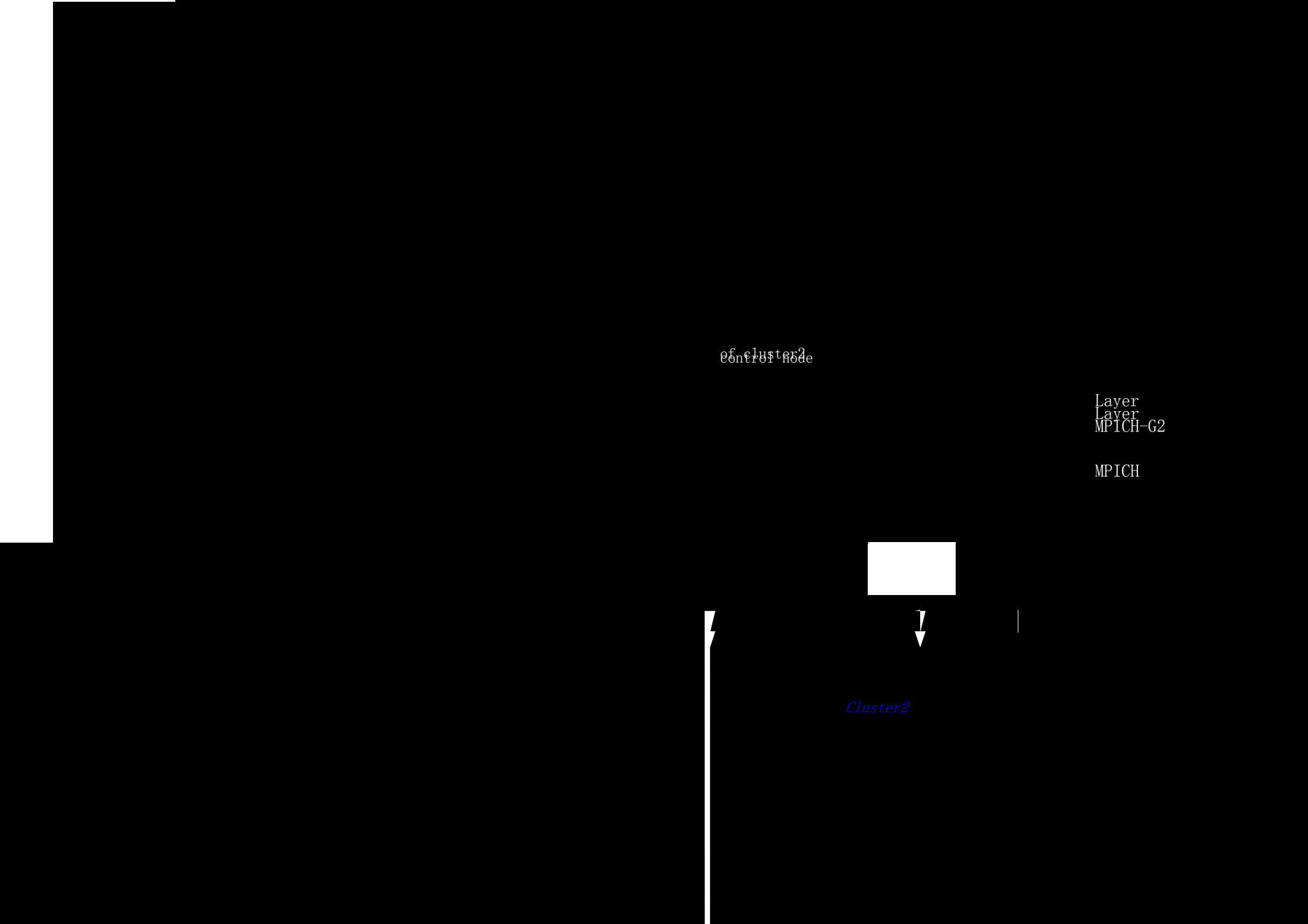
		C=0	C=1	C=2	C=3	C=4	C=5	C=6	C=7	C=8	C=9	C=10	C=11	C=12	C=13
		∅	**A**	**T**	**C**	**T**	**C**	**G**	**T**	**A**	**T**	**G**	**A**	**T**	**G**
R=0	∅	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R=1	**G**	0	0	0	0	0	0	2	1	0	0	2	1	0	2
R=2	**T**	0	0	2	1	2	1	1	4	3	2	1	1	3	2
R=3	**C**	0	0	1	4	3	4	3	3	3	2	1	0	2	2
R=4	**T**	0	0	2	3	6	5	4	5	4	5	4	3	2	1
R=5	**A**	0	2	2	2	5	5	4	4	7	6	5	6	5	4
R=6	**T**	0	1	4	3	4	4	4	6	5	9	8	7	8	7
R=7	**C**	0	0	3	6	5	6	5	5	5	8	8	7	7	7
R=8	**A**	0	2	2	5	5	5	5	4	7	7	7	10	9	8
R=9	**C**	0	1	1	4	4	7	6	5	6	6	6	9	9	8

- Aligning Determine *k* highest scores with different alignment start points.
- Use divide-conquer algorithm *k* times to determine the actual alignments.

Multi-Cluster Platform

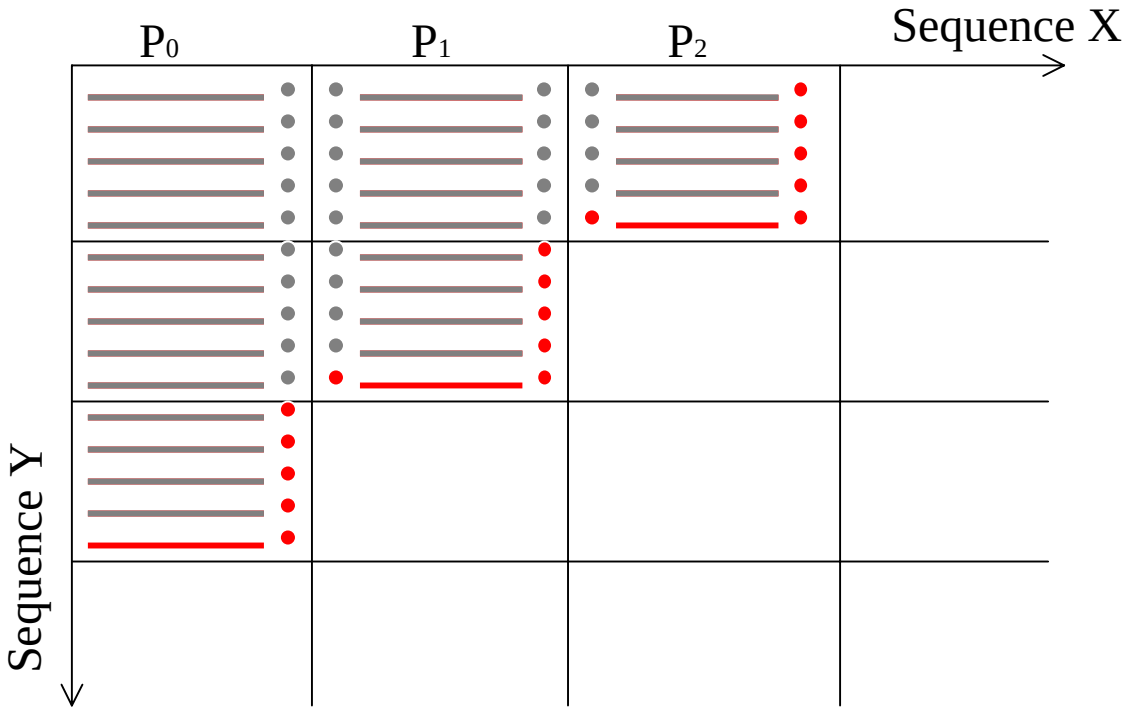


- High performance at low cost
- Geographically distributed
- Different Bandwidths


of cluster2
control node
Layer
Layer
MPICH-G2
MPICH
Cluster2

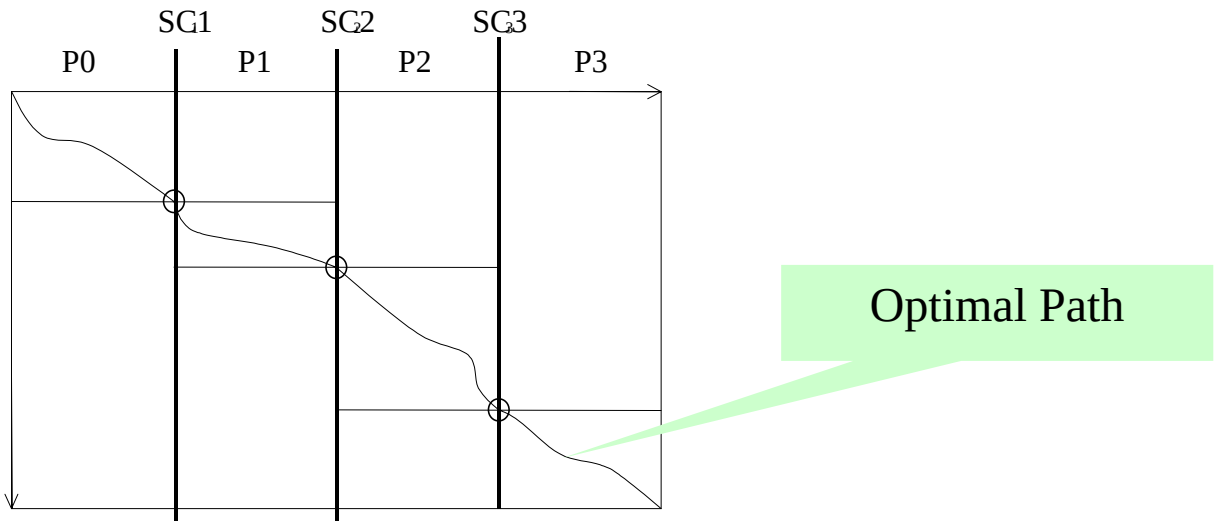
Parallelization on Multi-Cluster

- Wavefront Computation of DP matrix



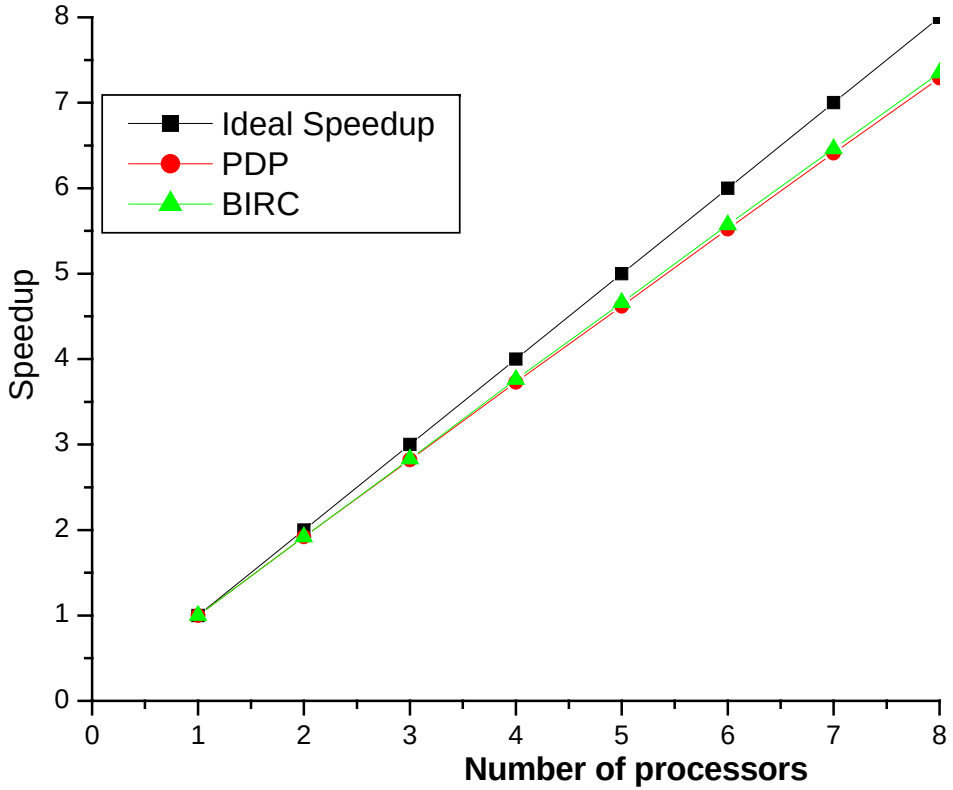
Parallelization on Multi-Cluster

- Divide-and-Conquer traceback computation
- Define special columns (SC's) as the last columns of the parts of the similarity matrix allocated to each processor, except the last processor
- Identify the intersection of an optimal path with the special columns



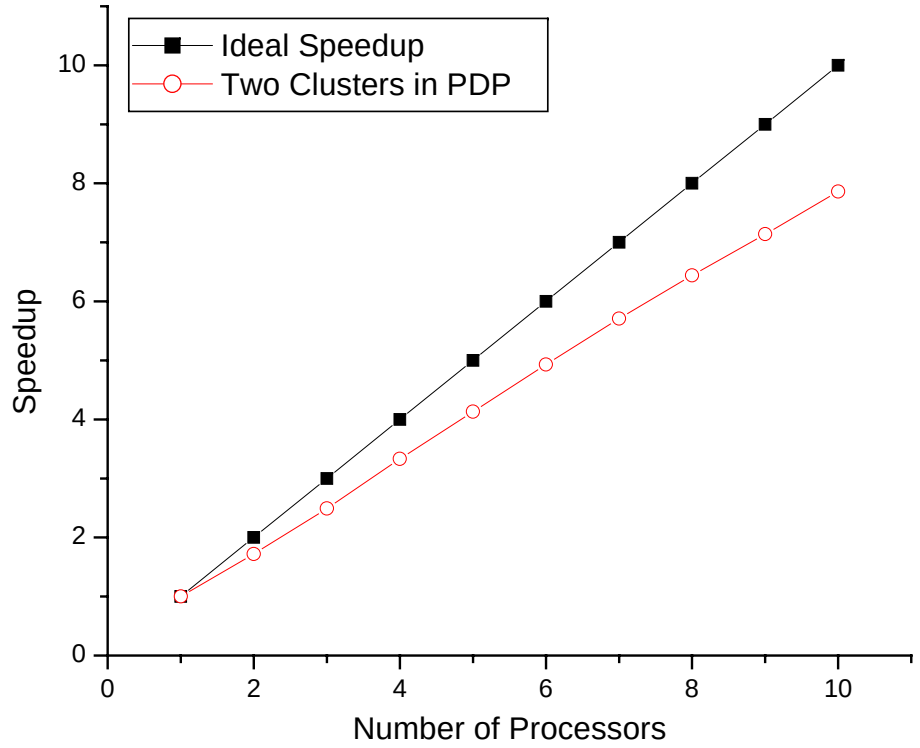
Performance Evaluation

- Using MPICH in PDP and BIRC to check the performance of every cluster



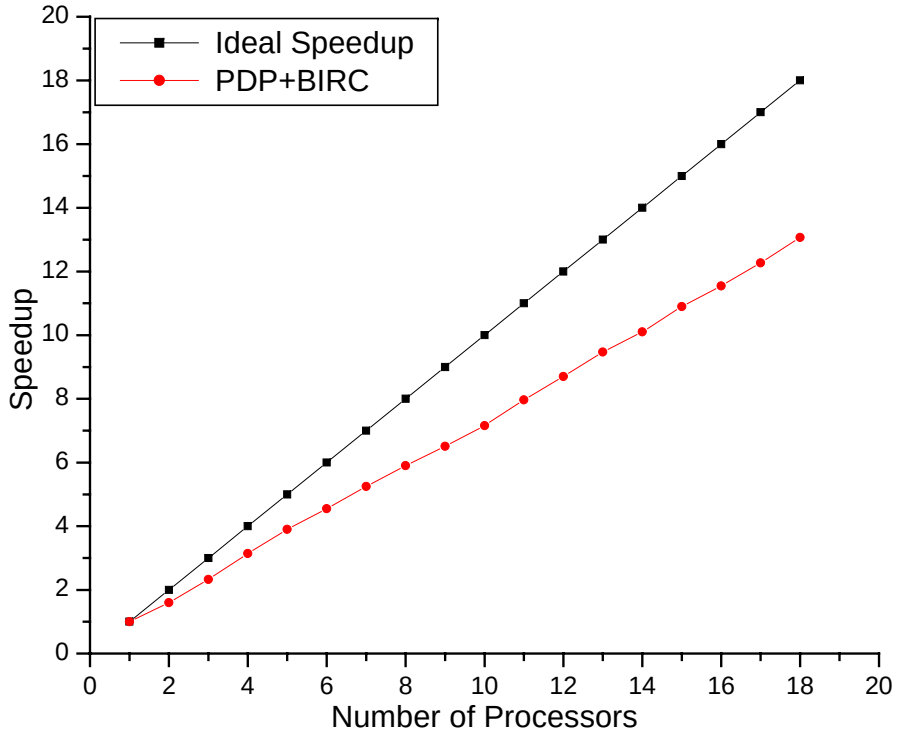
Performance Evaluation

- Speedups on two the clusters in PDP for aligning two sequences of length 100,000



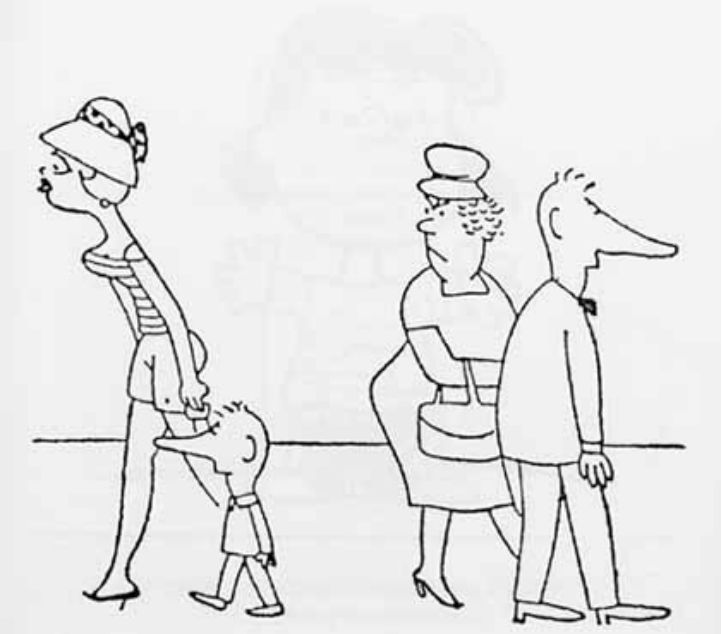
Performance Evaluation

- Speedups on the PDP and BIRC clusters for aligning two sequences of length 100,000



Performance Evaluation

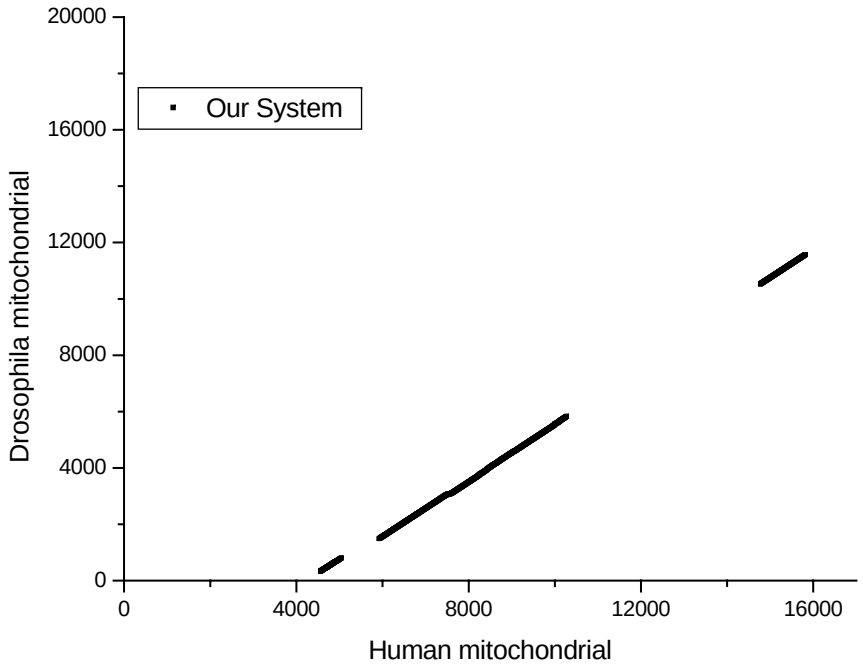
- Quality of Alignment Results
- Comparison to results of *MUMmer*
  - Assumes two sequences are closely-related (highly similar)
  - Can align two bacterial genomes under 1 minute
  - Uses Suffix Tree to find Maximal Unique Matches
  - Maximal Unique Match (*MUM*) Definition:
    - A subsequence that occurs in two exactly matching copies, once in each input sequence, and that can not be extended in either direction
  - *Key idea:* significant long MUM is certainly to be part of the global alignment

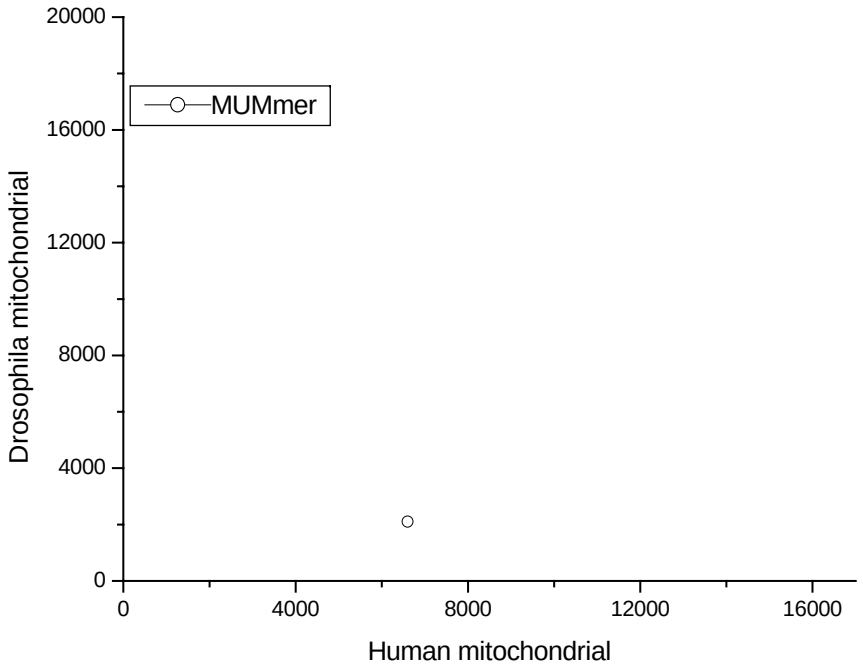


Can you find the MUM?

Performance Evaluation

- Quality of Alignment Results
- Align mitochondrial genomes of Human and Drosophila (not very related)





Conclusion and Future Work

- Presented how alignment of long DNA sequences based Dynamic Programming and Divide-and-Conquer can be efficiently parallelized on a Multi-Cluster Architecture
- Investigating which other applications can benefit from this type of computing power
- Investigating different bandwidths levels