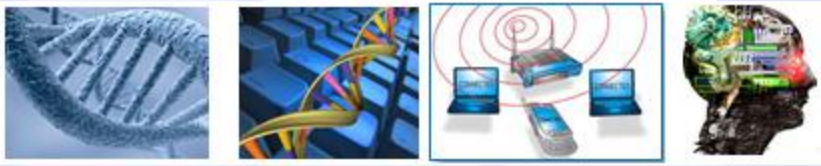


Sequence Alignment & Analysis

Linda Emujakporue and Ross Beckley

Department of Computer Science
College of Engineering
Tennessee State University

Advisor: Mr. Heh Miao and Dr. Wei Chen




OUTLINE

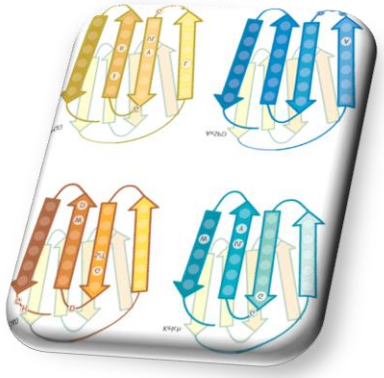
- Introduction
- Requirement analysis
  - Functional and non-functional
- Design and implementation
  - Architectural and detailed design
- Evaluation
- Conclusion

INTRODUCTION

- Bioinformatics develops computational theories and algorithms to analyze biological structures
  - Algorithms, databases, artificial intelligence, modeling and simulation, and more

Sequence Alignment

- Sequence Alignment and Analysis
  - Comparing pairs or groups of DNA sequences to find similarities
- Why is it important?
  - It can be used to find functional, structural, or evolutionary relationships between the sequences






Problem Statement

- Sequence alignment is critical in Bioinformatics for finding functional, structural, or evolutionary relationships.
- Aligning hundreds or thousands pairs/groups of DNA sequences is time costing.
- High performance computing (parallel computing) should be introduced into sequence alignment.




Project Goal and Objectives

- To design and implement parallel algorithms for efficiently finding regions of similarities in DNA and proteins
  - Implementing a ***global*** alignment algorithm
  - Implementing a ***local*** alignment algorithm
  - Converting those algorithms so that they can be implemented in parallel
    - To speed up the whole alignment process




Functional Requirements

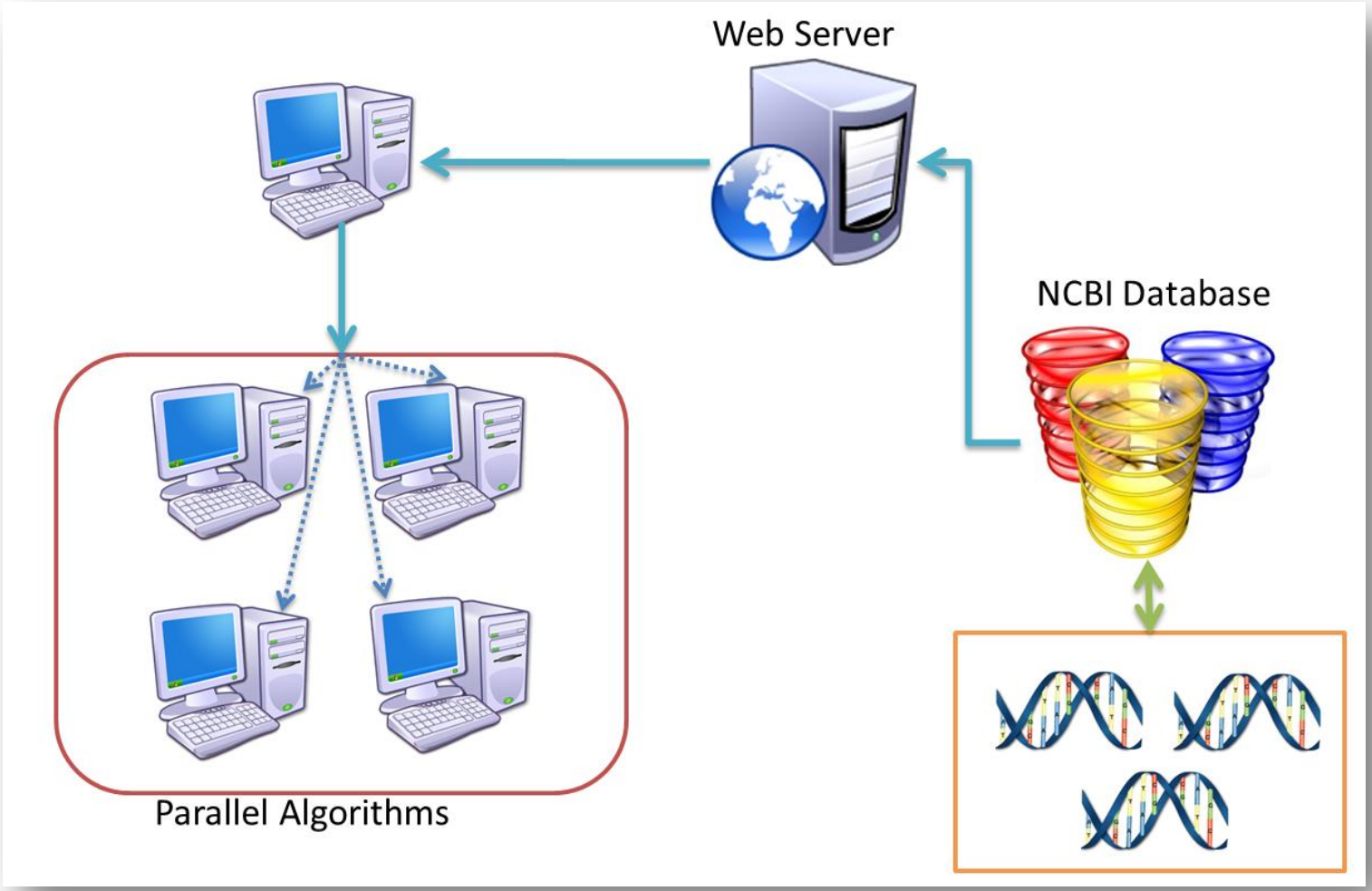
- Given a DNA/ Protein sequence, find the similarities/relationships in the same group of DNA/Proteins
  - Using global and local alignment techniques
- Develop strategies to convert those techniques into parallel implementation
- Compare non-parallel and parallel performances

Non-Functional Requirements

- NCBI database will be used to extract DNA and protein sequences
  - NCBI is widely accepted and used in bioinformatics community

- The algorithms will work with at least 80% accuracy
- .NET environment will be utilized
  - It provides a parallel processing library and the team is familiar with .NET

System Architecture






Global Alignment

- Needleman-Wunsh Algorithm
  - It produces an optimal alignment of two protein or DNA sequences allowing for the introduction of gaps
  - The Needleman-Wunsh algorithm aligns whole sequences (that is why it is called global alignment)

Global Alignment (cont.)

SCORING SCHEME can be

- Match Score = +1
- Mismatch Score = -1
- Gap penalty = -1

	A	C	T	G
A	1	-1	-1	-1
C	-1	1	-1	-1
T	-1	-1	1	-1
G	-1	-1	-1	1

The score of any cell C(i, j) is the maximum of:

scorediag = C(i-1, j-1) + S(i, j)
scoreup = C(i-1, j) + g
scoreleft = C(i, j-1) + g
where S(i, j) is the substitution score for letters i and j, and g is the gap penalty

Global Alignment (cont.)

- Scoring
  - Example (g = -1)

The calculation for the cell C(2, 2):

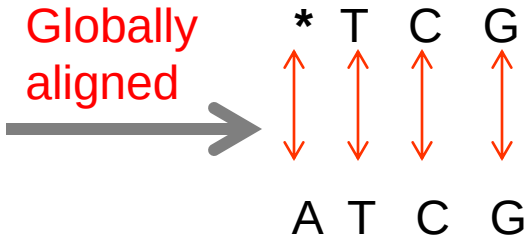
scorediag = C(i-1, j-1) + S(I, j) = 0 + -1 = -1

scoreup = C(i-1, j) + g = -1 + -1 = -2

scoreleft = C(i, j-1) + g = -1 + -1 = -2

		T	C	G
	0	-1	-2	-3
A	-1	-1	-2	-3
T	-2	0	-1	-2
C	-3	-1	1	0
G	-4	-2	0	2

Globally aligned






Local Alignment

- Smith-Waterman Algorithm
  - Smith-Waterman determines the optimal alignment of subsequences from a pair of sequences (that is why it is called as local alignment) .
  - For align subsequences there is no penalty when starting or stopping the alignment in the middle of the sequences.




Local Alignment (cont.)

SCORING SCHEME can be

- Match Score = +1
- Mismatch Score = -1
- Gap penalty = -1

	A	C	T	G
A	1	-1	-1	-1
C	-1	1	-1	-1
T	-1	-1	2	1
G	-1	-1	1	3

The score of any cell C(i, j) is the maximum of:

scorediag = C(i-1, j-1) + S(i, j)
scoreup = C(i-1, j) + g
scoreleft = C(i, j-1) + g
where S(i, j) is the substitution score for letters i and j, and g is the gap penalty

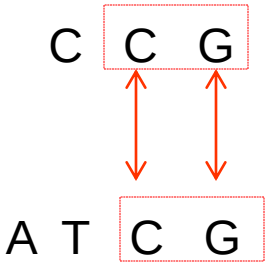
Local Alignment (cont.)

- Scoring
  - Example (g = -1)

- The scores are compared to zero, so that negative values are dismissed
- The trace-back method is then applied starting at the maximum score and tracing back to a zero.

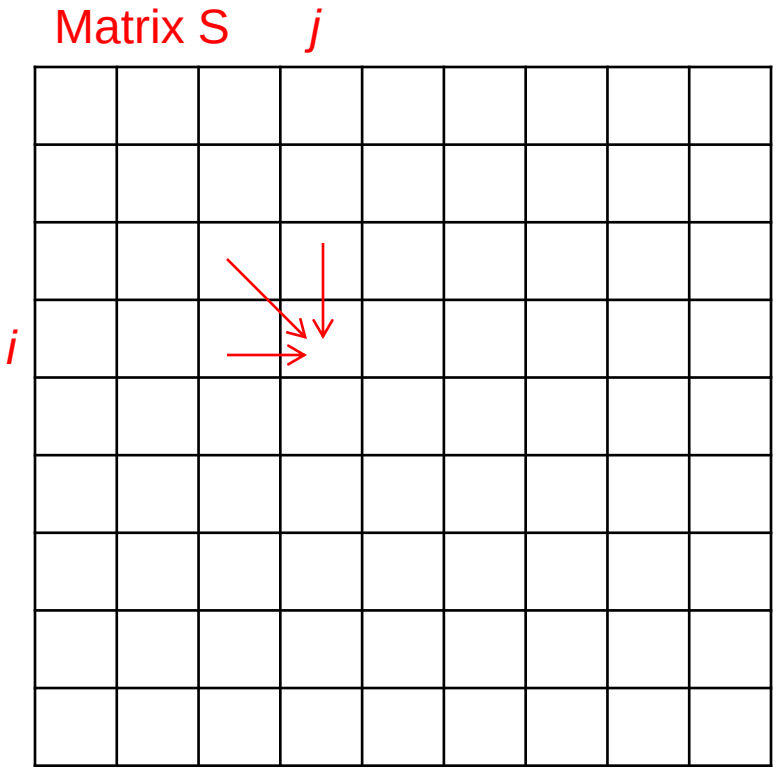
		C	C	G
	0	0	0	0
A	0	0	0	0
T	0	0	0	0
C	0	1	1	0
G	0	0	0	2

Locally aligned



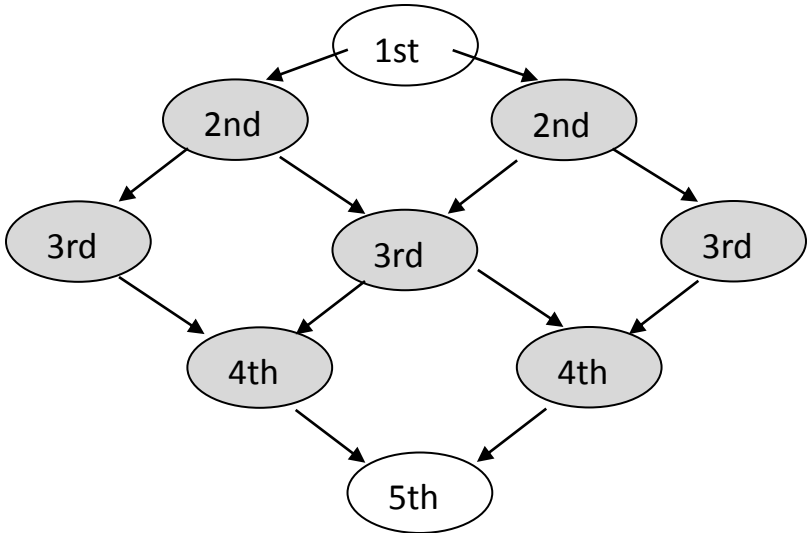
Parallel Strategies

When calculate score matrix S, the value of S(i,j) depends only on s(i-1,j) and S(i,j-1)



Parallelization (divide S to k×k submatrics)

Compute First	Compute Second	Compute Third
Compute Second	Compute Third	Compute Fourth
Compute Third	Compute Fourth	Compute Fifth



Evaluation – Theoretical Analysis

nxn Scoring Matrix

Divide nxn matrix to n/2 x n/2 matrix

1	2
2	3

Divide nxn matrix to n/k x n/k matrix

1	2	3
2	3	4
3	4	5

Sequential computing use one processor

calculate n^2 scores in O(n^2) time.

For 2x2 partition using 2 processors

First time: first processor calculates $(n/2 \times n/2)^2$ scores.

Second time: each of first and second processors calculate $(n/2 \times n/2)^2$ scores in parallel.

Third time: first processor calculate $(n/2 \times n/2)^2$ scores.

Totally, all tasks finish in O($3/4n^2$) time

Generally, for kxk partition using k processors

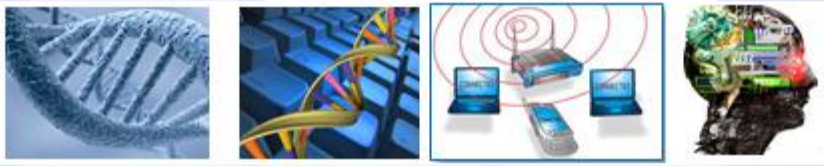
All tasks finish in $O((2k-1)(n/k \times n/k)) = O(\frac{2k-1}{k^2} n^2)$ time.




Evaluation – Simulation Results (1)

.NET Task Parallel Library

- TPL enables the user to express potential parallelism in form of lightweight tasks
- TPL schedules these tasks to run on parallel hardware and provides capabilities to cancel tasks and wait for completion

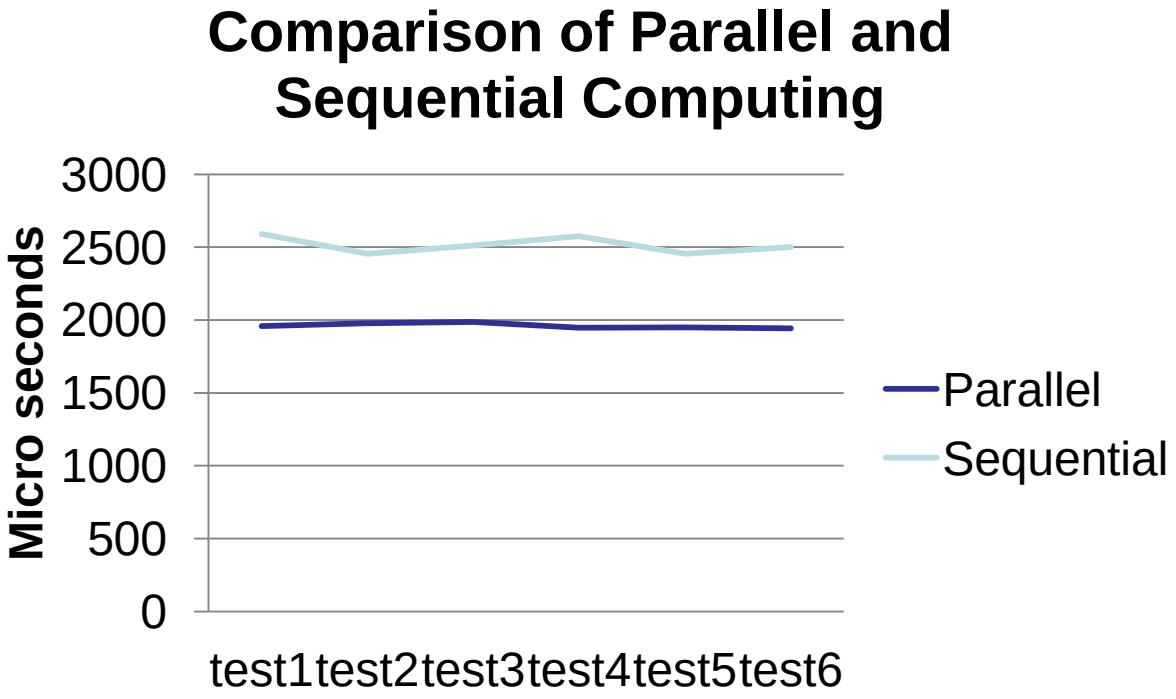


Evaluation – Simulation Results (2)

Result with 4 sub-tasks (computation time in microseconds)							
	test1	test2	test3	test4	test5	test6	Ave
sequential	2592	2455	2511	2576	2456	2501	2515.166667
parallel	1959	1979	1988	1947	1949	1944	1961.4
Improve	24%	19%	21%	24%	21%	22%	22%
Average Improvement: 22%							

Result with 9 sub-tasks (computation time in microseconds)							
	test1	test2	test3	test4	test5	test6	Ave
sequential	2592	2455	2511	2576	2456	2501	2515.166667
parallel	1771	1782	1776	1772	1781	1772	1776.6
Improve	32%	27%	29%	31%	27%	29%	29%
Average Improvement: 29%							

Evaluation – Simulation Results (3)



Acknowledgement

- Mr. Heh Miao
- Dr. Wei Chen
- Dr. Ali Sekmen

TNSCORE
TENNESSEE EPSCoR

NSF