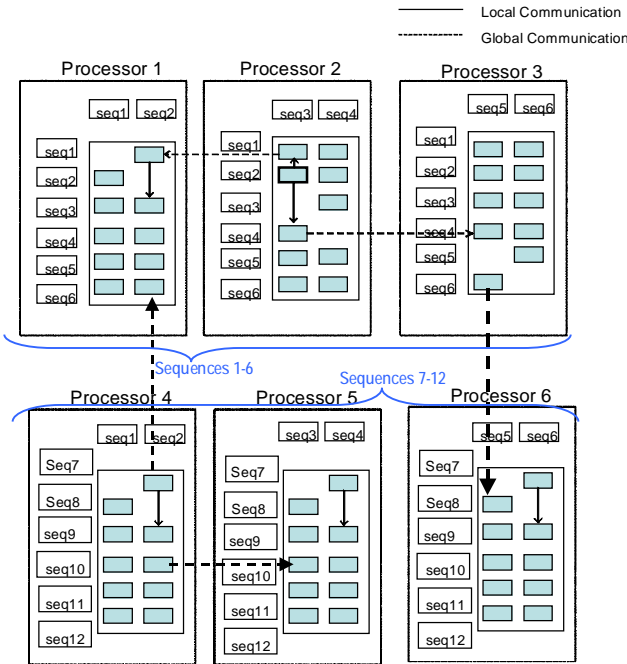


Multiple Genome Alignment on the Grid

Abstract

This application takes a number of genome sequences as input and gives an aligned sequence based on their structure by using a pairwise alignment algorithm. When run on grids like SURAgGrid, carefully designed and grid-enabled algorithms like this, which implement a memory efficient method for computation and are also parallelized efficiently so that the workload is well distributed on grids, afford bioinformatics users a performance comparable to cluster environments while giving them added flexibility and scalability.

Parallel load distribution among processors for multiple sequence alignment



Application Project Team

- Nova Ahmed**
Ph.D. student CS, Georgia Tech
- Victor Bolet**
Analyst Programmer Intermediate, Advanced Campus Services Georgia State
- Dharam Damani**
MS student CS, Georgia State
- Nicole Geiger**
Analyst Programmer Associate, Advanced Campus Services Georgia State
- Yi Pan**
Professor, Chair Computer Science, Georgia State
- Art Vandenberg**
Director Advanced Campus Services, Georgia State
- Chao "Bill" Xie**
Ph.D. student CS, Georgia State

Genome Alignment

Alignment 1
Sequence X: A T A - A G T
Sequence Y: A T G C A G T
Score: 1 1 -1 -2 1 1 1 Total Score = 2

Alignment 2
Sequence X: A T A A G T
Sequence Y: A T G C A G T
Score: 1 1 -1 -1 -1 -1 Total Score = -3

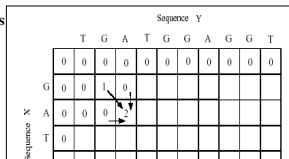
- Based on pairwise algorithm
 - Similarity Matrix, SM, built to compare all sequence positions
 - Observation that many "alignment scores" are zero value
- SM reduced by storing only non-zero elements
 - Row-column information stored along with value
 - Block of memory dynamically allocated as non-zero element found
 - Data structure used to access allocated blocks
- Parallelism introduced to reduce computation

Ahmed, N, Pan, Y, Vandenberg, A and Sun, Y, "Parallel Algorithm for Multiple Genome Alignment on the Grid Environment," 6th Intl Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC-05) in conjunction with (IPDPS-2005) April 4-8, 2005.

Similarity Matrix Generation

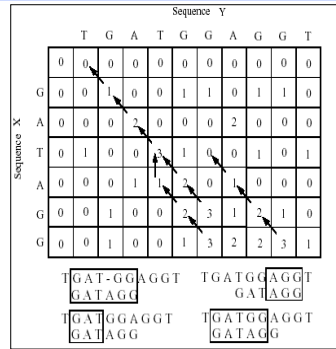
- Align Sequence X: TGATGGAGGT
Sequence Y: GATAGG
- 1 = matching; 0 = non-matching
- ss = substitution score; gp = gap score
- Generate SM max score with respect to neighbors:

$$SM[x, y] = \max \begin{cases} SM[x, y-1] + gp \\ SM[x-1, y-1] + ss \\ SM[x-1, y] + ss \\ 0 \end{cases}$$



Trace sequences

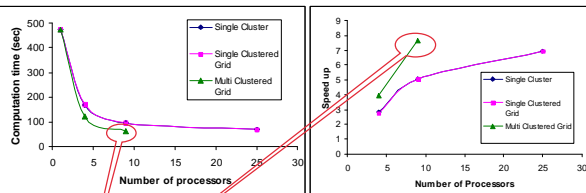
- Back trace matrix to find sequence matches



Computation Time

Speed up (1 cpu / N)

Job submission via SURAgGrid Portal



9 processors available in Multi Clustered Grid
32 processors for other configurations.

Interesting: When multiple clusters used (application spanned three separate clusters), performance improved additionally!?



Georgia State University



Application Driven Design for a Large-Scale, Multi-Purpose Grid Infrastructure