WinBioinfTools: Bioinformatics Tools for Windows Cluster

Done By:
Hisham Adel Mohamed
Objective

“Implement and Modify Bioinformatics Tools To run under Windows Cluster “

Project:

- Research Project between Nile University and CMIC (Cairo Microsoft Innovation Center)
Computer Cluster

Software:

- Supporting Operating System
- Job-scheduler
- Parallel Processing libraries/compilers

Compute node:

- PC
- Workstation
- HPC Server

Connectivity:

- Ethernet (10Mbps),
- Fast Ethernet (100Mbps),
- Gigabit Ethernet (1Gbps)
- Infiniband
Clusters Classification.....
Load balancing

- Used When there is a busy server processing client requests

Job scheduler
- SGE
- PBS
- W, HPC server component

Slide by: Dr Mohamed Abouelhoda Nile University (NU), Egypt 2009
(High Performance) Computer cluster

- All Nodes participate in solving the problem
- Nodes Need to communicate to each other

Internet Institution Enterprise

Head node

MPI/PVM

Communication between Nodes

Slide by: Dr Mohamed Abouelhoda Nile University (NU), Egypt 2009
High Availability Cluster

➢ To avoid falling of the services

Normal operation

Application A
Heart beat
Application B

After Failover

Application A
Heart beat
Application B

Slide by: Dr Mohamed Abouelhoda Nile University (NU), Egypt 2009
Parallel Programming

- **S I S D - Single Instruction, Single Data**
- **S I M D - Single Instruction, Multiple Data**
- **M I S D - Multiple Instruction, Single Data**
- **M I M D - Multiple Instruction, Multiple Data**
Shared Memory

- Processors can operate independently but share the same memory resources.
- Changes in a memory location or variable affected by one processor are visible to all other processors.
- Most common libraries: Pthreads and OpenMP
Distributed memory architecture

- Each Processors have local memory.
- A processor can't access Memory addresses of another processor.
- Programmer has to manage and synchronize the communication between nodes.
- Most Common libraries: **MPI** and **PVM**

[https://computing.llnl.gov/tutorials/parallel_comp/]
OpenMP

#include <omp.h>
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char *argv[])
{
    int nthreads, id;

    #pragma omp parallel private(nthreads, tid)
    {
        id = omp_get_thread_num();
        printf("Hello World from thread = %d\n", id);

        if (id == 0)
        {
            nthreads = omp_get_num_threads();
            printf("Number of threads = %d\n", nthreads);
        }
    }
}
```c
#include <stdio.h>
#include "mpi.h"
int main(int argc, char **argv)
{
    int rank, namelen, numprocs, x;
    char host[150];
    char msg[50];
    int tag = 1;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Get_processor_name(host,&namelen);
    if (rank == 0)
    {
        MPI_Status status;
        MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
        for (x = 1; x < numprocs; x++)
        {
            MPI_Recv(msg, 50, MPI_CHARACTER, MPI_ANY_SOURCE, tag, MPI_COMM_WORLD, &status);
            printf("Msg from %d: '%s'\n", status.MPI_SOURCE, msg);
        }
    }
    else
    {
        snprintf(msg, 50, "Hello from node rank %d.", rank);
        MPI_Send(msg, 50, MPI_CHARACTER, 0, tag, MPI_COMM_WORLD);
    }
    MPI_Finalize();
    return 0;
}
WinBioinfTools

- Parallel Sequence Alignment
- Parallel BLAST
- CoCoNUT
Resources

- **CMIC Lab:**
  - 4 nodes (2 Quad-core 2.6 GHz processors, 16GB RAM, 250 GB HD, 1G Ethernet Network)

- **Operating System:**
  - Windows HPC server 2008, with HPC Pack 2008
  - SUSE 11

- **Cluster management and monitoring tools**
  - Load balancing: Job scheduler for Windows and SGE for SUSE
  - Parallel computing: MS-MPI for Windows and MPICH2

- **Interoperability for Windows:**
  - SUA (Support for Unix Applications), Cygwin also works.
Dynamic programming algorithms take $O(n^k)$ time ($k=$ number of genomes, $n=$ average genome length)

Alignment of 8 Mbp of *C. briggsae* against 97 Mbp of *C. elegans* took 11 days using WABA (450 MHz CPU)

*Kent et al. 2000*
Sequence Alignment

- Sequence alignment aims at maximizing the similarities between sequences.
- Optimal sequence alignment can be computed using dynamic programming.
- For two sequences, the best alignment is computed by filling a 2D matrix, where the score at cell \((i,j)\) is computed as follows:

\[
score(i, j) = \min \begin{cases} 
score(i-1, j-1) + 1, & \text{if } S[i] = S[j] \\
score(i-1, j-1), & \text{if } S[i] \neq S[j] \\
score(i-1, j) - 1, & \\
score(i, j-1) - 1 & 
\end{cases}
\]
Parallel Sequence Alignment

- The data is divided on participating nodes.
- The cluster nodes cooperate in filling matrix (Compute Cluster Model)
- The filling proceeds diagonal-wise, and the master node synchronizes the filling
- The complexity reduces to $O(n^2/k+t_k')$, where $t$ is the communication time, $k$ is the number of cores, $k'$ is the number of cluster nodes
Results

- The running times (in seconds) for pairwise sequence alignment on one and 4 nodes.
BLAST

• BLAST (basic local alignment search tool): given a biological sequence it search for similar (sub) regions in the database

• The database size is extremely large

• The search time is proportional to the database length

• Computer cluster provides an ideal solution for speeding up BLAST search
Parallel BLAST

• We have Implemented Two Modules
  
  I. One is dividing the database equally on all participating nodes
  
  II. The other one is used to start up the Search on all nodes

Internet Institution Enterprise

queries

Database segmentation
Running Time for 1000 queries

Running times in hours
The first 3 databases are DNA while the others are proteins
The query sequence is of the same type as the database
CoCoNUT

• Pairwise comparison: Given two multi-chromosomal genomes, compare each chromosome in one genome to each chromosome of the other one.

• The pairwise comparison identifies the differences and similarity.

• CoCoNUT is written in Perl and C/C++ and it was intended to run under Linux/Unix

• CoCoNUT was ported to run under Windows
  • Parts of the code are compiled and runs directly on windows
  • Third party packages runs using SUA and Cygwin

• The large scale comparison runs
  • using the Job Scheduler
  • using our MPI-based script to save some computations
CoCoNUT Strategy

Fragment Generation Phase

- > Gi Genome 1
  CGCCACGCT..
  CGGCCGTGG
  CGGGCAAAAA

  Fragment generation tool

Based on GenomeTools, 3rd party package
Most time consuming

Chaining Phase

- Description Files
  Statistics files

  Chaining

Based on CHAINER
Recursive calls
Chaining fragments:
Many variations for many tasks

Post-processing Phase

Visuilaization
Dotplot
**Pairwise Comparison of multi-chromosomal**

**Load balancing**

Chromosome comparisons are independent of each other → Divide the comparisons among the cluster nodes

**Compute Cluster**

Speed-up the computation of CoCoNUT phases: fragment generation and chaining

---

**Slide by: Dr Mohamed Abouelhoda Nile University (NU), Egypt 2009**
Load Balancing Results

- Total time of 20 comparison on 4 nodes: approx. 12 h. on Windows and 9 h. on Linux.

- Total time of 20 comparison one node: approx. 47 h. on Windows and 40 h. on Linux.

- Estimated total time for the whole human-mouse comparison is 75 days on 1 node and 19 days on 4 nodes
Hisham Mohamed and Mohamed Abouelhoda

WinBioinfTools: Bioinformatics Tools for Windows Cluster

Abstract: Open source bioinformatics tools running under MS Windows are rare to find, and those running under Windows HPC cluster are almost nonexisting. This is despite the fact that Windows is the most popular operating system used among life scientists. Therefore, we introduce WinBioinfTools, an open source toolkit containing a number of bioinformatics tools running under Windows High Performance Computing Server 2008. We currently start with three programs from the area of sequence analysis: 1) CoCoNUT for pairwise genome comparison, 2) parallel BLAST for biological database search, and 3) parallel global pairwise sequence alignment. In this paper, we show how some components of these tools were ported from Linux/Unix environment to run under Windows. We also demonstrate by experiments the performance gain achieved when using a computer cluster against a single machine, highlighting the benefits of using the Windows HPC Cluster 2008. Furthermore, we show the results of comparing the performance of WinBioinfTools on the Windows and Linux Cluster.

Questions ?