International Conference
"Automatics and Informatics'2020"

An Approach for Parallel Reading in Multiple Sequence Alignment

Soon-Heum Ko

National Supercomputing Center, Linkoping Universit, Sweden

sko@nsc.liu.se

Veska Gancheva

Technical University of Sofia, Bulgaria

vgan@tu-sofia.bg

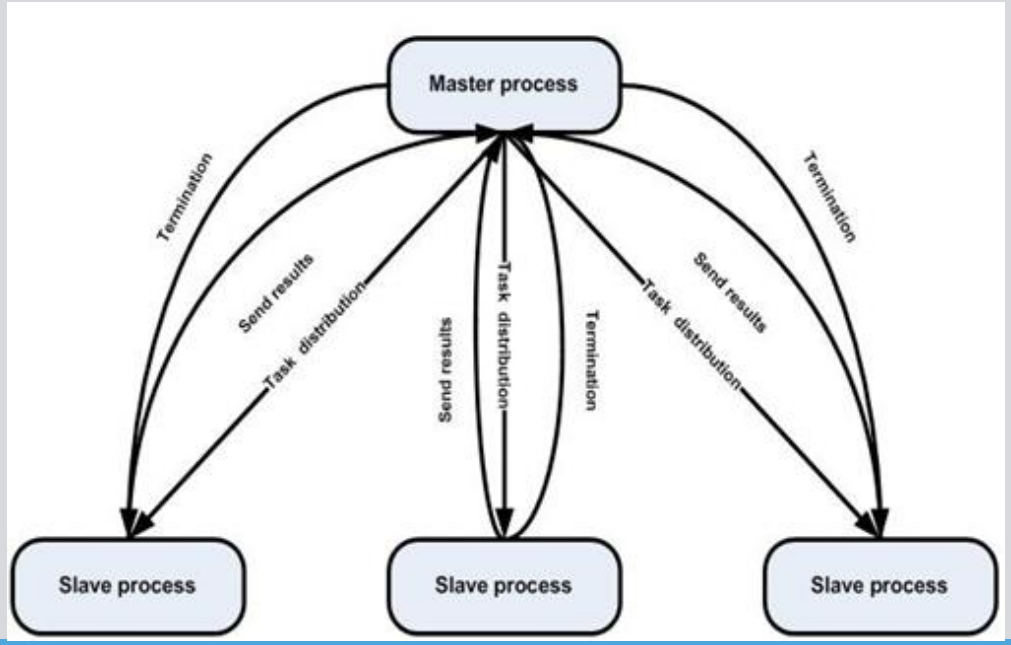
Project "Innovative Platform for Intelligent Management and Analysis of Big Data Streams Supporting Biomedical Scientific Research"
Financially supported by the National Science Fund, Bulgarian Ministry of Education and Science, project Grant KP-06-N37/24.

Introduction

- Multiple sequence alignment (MSA) is an important method for biological sequences analysis.
- This method is in search of evolutionary relationship among biological sequences.
- Amounts of genetic sequences of proteins, DNAs, RNAs are newly discovered every day, which require the adoption of advanced I/O techniques as well as efficient parallelism for sequence analyses.
- In this paper we propose a parallel I/O approach for multiple sequence alignment input reading.

ClustalW-MPI

- ClustalW-MPI is a parallel version of ClustalW software and applies the master-slave parallelism for multiple sequence alignment.
- ClustalW-MPI is capable of experimenting up to ~10K sequence sets.



Designing a Parallel I/O Interface for Multiple Sequence Alignment

- The structure of the parallel read routine is summarized as follows:
  - Predefine *group_size* and *read_chunk_size* parameters
  - Create an MPI Group '*processor_group*' whose entities are (i×*group_size*, i×*group_size*+1, …, (i+1)×*group_size*-1) for i^{th} group of processors
  - Create an MPI communicator '*group_comm*'
  - Create an MPI Group '*leader_group*' whose entities are (0, *group_size*, …, (N-1)×*group_size*)
  - Create an MPI communicator '*leader_comm*'
  - Create a string *read_buffer*[*file_size*]
  - *leader_comm* calls *MPI_file_open* sequence dataset
  - *leader_comm* iterates *MPI_read* calls to store the *read_chunk_size* chunk of input file to a *read_buffer*, until all sequence dataset are read
  - Broadcast operation of *read_buffer* between the *group_comm*
  - Reformulate *read_buffer* info to recognizable format of *seq*[*seq_id*][*each_seq_size*]
- Parameters of *group_size* and *read_chunk_size* are dependent of data file size, total number of processors, and the file system capacity. We determine these parameters heuristically after the synthetic benchmark of I/O performance.

Experimental Resource

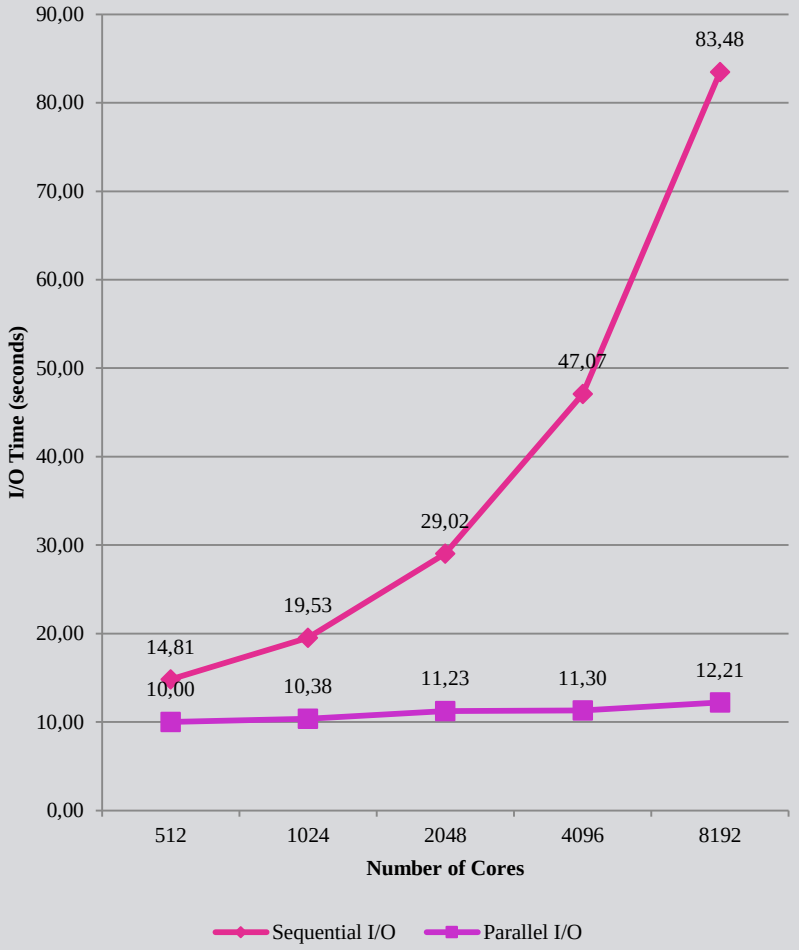
- JUQUEEN supercomputer.
- 458,752 cores in total.
- Each compute node contains 16 cores
- 4-way hyperthreading (use up to 64 cores-per-node)
- 16 GB RAM
- GPFS (General Parallel File System).

Sequence Dataset

- 4 different datasets ranging from tens of KiloBytes to a hundred MegaBytes.
- A subset of human influenza A virus type from NCBI.
- 8999 sequences approx. 16 MB.
- A dataset with 139 MB in size.

Parallel Reading Performance for Genetic Sequence Input

- Wall-time comparison between sequential and parallel file reading.
- Measured time is the total simulation time in seconds until all data inputs are broadcasted to all slaves.



Conclusion

- The use of MPI-I/O over a subset of MPI cores
- It is achieved by creating a number of subgroups under a global MPI communicator
- We verify the performance of our approach by comparing it with the traditional way of “sequential file reading and global broadcast”
- We apply it to the MPI version of multiple sequence alignment software ClustalW.
- In the production run over 8192 BlueGene/Q cores, the current approach provides 6.8 times speed-up than the original ClustalW-MPI implementation.

- Thank you for your attention!