

Accelerate Life Science Applications Using Grid Computing

Wang Junhong
(HPC, Computer Centre)

Many life sciences relating to bioinformatics applications that deal with processing of large data sets can be very time-consuming. A comprehensive biological sequence analysis is estimated to take months or even years on a powerful computer. Fortunately, those large data set analyses can be partitioned into multiple small data chunks and processed independently on normal computers. By running these processing tasks on small data chunks on a large pool of computers, the entire process can be shortened significantly. If the application can be grid parallelised and enabled on the grid computing, good acceleration of the computational task can be achieved easily with the job submission and sub-task dispatching to the clients for execution to be handled by the grid server automatically.

On NUS Grid, four open source bioinformatics application packages have been parallelised and enabled to date. Learn about the parallelisation of the application and the performance on the Grid in the following sections.

1. BLAST



URL: <http://www.ncbi.nlm.nih.gov/>

In bioinformatics, **Basic Local Alignment Search Tool**, or **BLAST**, is an algorithm for comparing primary biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences. A BLAST search enables a researcher to compare a query sequence with a library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold.

BLAST is often used as part of other algorithms that require approximate sequence matching. Examples of other questions that researchers use BLAST to answer are:

- Which bacterial species have a protein that is related in lineage to a certain protein with known amino-acid sequence?
- Where does a certain sequence of DNA originate?
- What other genes encode proteins that exhibit structures or motifs such as ones that have just been determined?

BLAST is one of the most widely used bioinformatics programs because it addresses a fundamental problem and the algorithm emphasises speed over sensitivity. This emphasis on speed is vital to making the algorithm practical on the huge genome databases currently available, although subsequent algorithms can be even faster.

Grid Parallelization

To accelerate the BLAST analysis, the huge database and sequences input can be split into many small pieces and with each pair of the DB piece and sequence piece bundled as one task to process, the original computational intensive sequencing can be replaced by hundreds or thousands of small

tasks (see Figure 1). Given a big pool of normal computers such as a desktop at your office, these small tasks can be completed within much a shorter time compared to running the original sequencing analysis on dedicated compute servers.

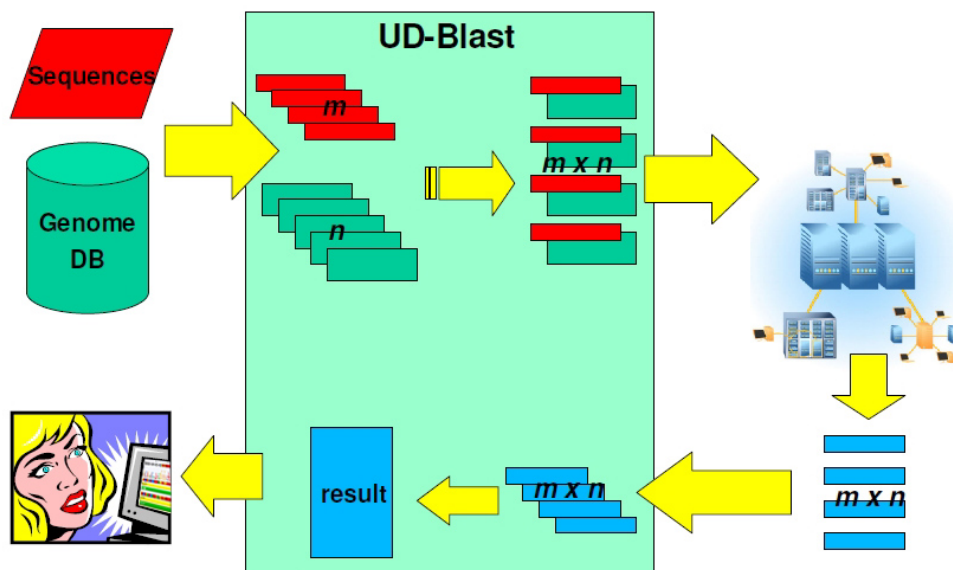


Figure 1 Parallelization of BLAST Sequencing Analysis

Example and Performance

This data parallel idea for BLAST was enabled and made available on NUS Grid platform. An actual project performed on the Grid showed that one analysis with 22,484 sequences to be aligned against 3,461,799 database sequences could be completed in two days, compared to an estimate of 30 days if it was run on a SunFire E6800 with 16 processors for the original analysis. With such accelerated performance, researchers can move towards a bigger sequencing analysis with 100,000 sequences and are able to evaluate different sequencing configurations.

2. HMMER



URL: <http://hmmer.janelia.org/>

Profile hidden Markov models (profile HMMs) can be used to do sensitive database searching using statistical descriptions of a sequence family's consensus. HMMER is a freely distributable implementation of profile HMM software for protein sequence analysis. HMMER is maintained at <http://hmmer.janelia.org/>.

Grid Parallelisation

HMMER is computational intensive too. Fortunately it works similarly to BLAST, as in the studied sequences can be separated into multiple sub sequences and then searched against the known protein database. Figure 2 illustrates the workflow for HMMER sequencing analysis.

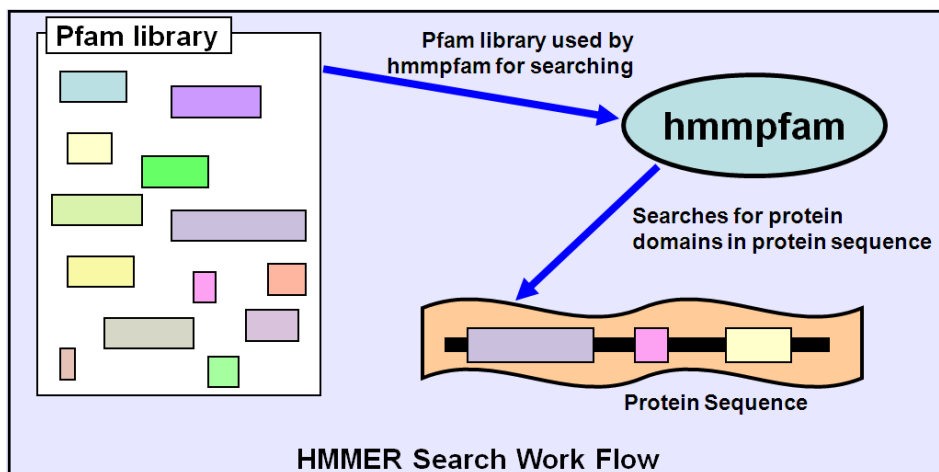


Figure 2 Diagram of HMMER Sequencing Analysis

Example and Performance

In a protein sequence study of *Drosophila melanogaster* (fruitfly) using HMMER on the NUS Grid platform, the whole analysis which was estimated to require 17 days to complete on the fastest machine at that time, could be finished in four days. The speedup performance improved the efficiency of researchers and enabled them to run the study on larger genomes that are much more challenging.

3. AutoDOCK



URL: <http://autodock.scripps.edu/>

AutoDock is a suite of automated docking tools. It is designed to predict how small molecules, such as substrates or drug candidates, bind to a receptor of known 3D structure. AutoDock consists of two main programs: AutoDock performs the docking of the ligand to a set of grids describing the target protein; AutoGrid pre-calculates these grids.

AutoDock is a useful tool for drug design and its applications also include:

- X-ray crystallography;
- lead optimisation;
- virtual screening (HTS);
- combinatorial library design;
- protein-protein docking;
- chemical mechanism studies.



AutoDock is free software and is distributed under the GNU General Public License. The webpage for AutoDock is at <http://autodock.scripps.edu/>.

Grid Parallelisation

To obtain better statistics and clustering for the docking result, more numbers of final conformations (ga_run) needs to be computed. Higher numbers indicate demand for more computational work. With each of the final conformations computed independently, the docking process can be parallelised easily by evenly distributing the multiple docking tasks with a relative small ga_run number to different computers to run.

Example and Performance

Using the AutoDock enabled on the NUS Grid, a researcher managed to complete three docking analyses concurrently within two months. Each of the analyses included 200,000 final conformations, which would take about 32 months if it was run on a desktop computer continuously.

4. PHYLIP



URL: <http://www.phylip.com/>

PHYLIP is a free Computational phylogenetics package of programs for inferring evolutionary trees (phylogenies). The name is an acronym for PHYLogeny Inference Package. Methods (implemented by each program) that are available in the package include parsimony, distance matrix, and likelihood methods, including bootstrapping and consensus trees. Data types that can be handled include molecular sequences, gene frequencies, restriction sites and fragments, distance matrices, and discrete characters.

Grid Parallelisation

Grid parallelisation of PHYLIP was driven by a researcher who needed to run large amounts of PHYLIP analyses as she could not do so with conventional computing servers or workstations that she had access to. The unique feature of the grid-enabled PHYLIP is that only the computational intensive core step “protdist” is grid-parallelised, and the very minor computational steps of 1st, 3rd, 4th and 5th can be performed on a normal computer within a couple of minutes. The typical PHYLIP analysis steps are illustrated in Figure 3.

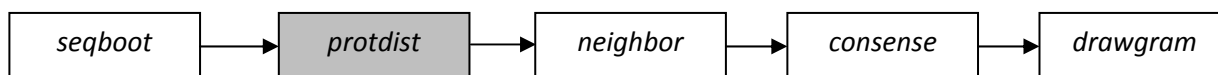


Figure 3: Typical pipeline for building a protein phylogentic tree with PHYLIP.

Example and Performance

Preliminary results of the grid parallelised PHYLIP on the Grid have achieved very good speedup rate at about 24 to 58 times. One of the largest jobs consuming 12.7 months of CPU time was able to be completed in a week on the Grid. With such a great improvement in efficiency, PHYLIP will be a significant power tool for researchers.