Big Data Parallel Processing of Personal Genomes

Shinichi Morishita (University of Tokyo)

In this talk, I will address two technical problems in processing personal genomes. The first problem is to search for single nucleotide variations (single letter changes) in personal genomes that are inconsistent with the reference human genome. One individual has approximately three million single nucleotide variations in its genome, and some of them can be associated with genetic disorders. For this search, we typically collect about one billion of short sequences of length 100 nucleotides at random (35-fold coverage of the human genome) from a personal genome, which takes about one day using a standard second-generation sequencer such as HiSeq2500. Afterwards, we align each of one billion sequences to the reference human genome of ~3 billion nucleotides in length with allowing a couple of mismatches. For this alignment, a variety of algorithms using hash tables, suffix arrays, and Burrows-Wheeler Transform have been proposed, and many efficient software programs that devise these ideas are also available. The alignment task can be executed in parallel by dividing the set of one billion sequences into subsets and by assigning each subset to a CPU core, which is a typical example of data parallelism. In order to complete the entire job in one day, we use 30 CPUs (Intel Xeon X5690) with 12 cores and 96GB of main memory in our university hospital. In near future, the time to collect data will become less than one hour. The data processing time should be reduced accordingly.

While searching for single nucleotide variations is now becoming common in medical genome analysis, large-scale structural variations such as long insertions and genome duplications in personal genomes are largely unexplored because uncovering large-scale structural variations is really challenging. We need to collect longer sequences from a personal genome, and we use Pacific Biosciences RS sequencer, the first commercialized third-generation single-molecule real-time sequencer that is able to output very long sequences of length ~5000 nucleotides on average. One serious problem inherent in single-molecule sequencers is a high sequencing error rate of ~15%, but fortunately, we can reduce the error ratio to 0.1-0.2% by aligning accurate short reads to erroneous long reads. A typical example is to align one billion sequences of 100 nucleotides in length to millions of long reads of length ~5000 nucleotides (on average) that amount to ~30 billion

nucleotides. To handle such big data efficiently, we need to carefully move data between parallel nodes while reducing the IO bottleneck. Moreover, because of a high error rate of long reads, we have to allow many mismatches when we align short sequences to long ones. Shoichiro Oishi, one of my colleagues, developed a program that could complete this task within one day using 60 CPUs (Intel Xeon X5690) with 12 cores. I will outline his algorithm in this talk.

I have tried to answer the following questions as possible.

1. Architecture:

- What architectural changes are needed for extreme computing storage systems to make them better suited for BD?
- What operational changes are needed to support new storage architectures?
- Looking at future technologies, what future architectures are possible?

It is quite important for us to run the system for many hours without failure of computations. Another pressing need is to accelerate data transfer between the main memory and secondary disk.

2. Workflows:

- For extreme computing and big data, describe a forwarding-looking workflow, from simulation to analysis.
- What software is missing to support your workflow?
- A plan for achieving interoperability among various systems that one might want to use.

3. Taxonomy:

- There are several forms of data-centric computing linked to extreme computing. One outcome of this workshop is to help describe these modes. Please outline how you use your data and how you answer questions about your science using your data.
- Do you have a data-driven mini-application that demonstrates a new usage model?
- What are cross-cutting concerns for BD (for example: data integrity)

See the two problems above that I mentioned.

4. Software:

- What software are you currently using to manage and explore your data?
- What algorithms and software libraries/tools need development and improvement to address your big data needs?
- As you look to the future, what are the holes/gaps that have no planned solution?

Our processing pipeline outputs huge temporary files to secondary disks several times, which should be avoided in some way.

5. Interoperability challenges:

- How to handle Data provenance (location, observed/simulated, type of system concerned) from a data representation and IT architectural point of view? How to annotate existing data sets and develop records for data citation and tracking?
- What Information systems are used for providing semantic capacity to provide effective translation between data and conceptual models used by different communities?
- What IT systems are used for providing information about the actual use of both observational data and simulated data?