Title
Efficient Graph Based Assembly of Short-Read Sequences on a Hybrid Core Architecture

Authors
Sczyrba, Alex
Pratap, Abhishek
Canon, Shane
et al.

Publication Date
2011-03-22

Efficient Graph Based Assembly of Short-Read Sequences on a Hybrid Core Architecture

Alex Sczyrba*[1,2], Abhishek Pratap[1,2], Shane Canon[2,3], James Han[1,4], Alex Copeland[1,2], Zhong Wang[1,2], Tony Brewer[5], David Soper[5], Mike D'Jamoos[5], Kirby Collins[5], George Vacek[5]

[1]DOE Joint Genome Institute, Walnut Creek, CA, USA
[2]Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[3]National Energy Research Scientific Computing Center (NERSC), Oakland, CA, USA
[4]Lawrence Livermore National Laboratory, Livermore, CA, USA
[5]Convey Computer Corporation, Richardson, TX, USA

March 2011

The work conducted by the U.S. Department of Energy Joint Genome Institute is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231

DISCLAIMER

Efficient Graph Based Assembly of Short-Read Sequences on a Hybrid Core Architecture

Alex Sczyrba*[1,2], Abhishek Pratap[1,2], Shane Canon[2,3], James Han[1,4], Alex Copeland[1,2], Zhong Wang[1,2], Tony Brewer[5], David Soper[5], Mike D'Jamoos[5], Kirby Collins[5], George Vacek[5]

[1]DOE Joint Genome Institute, Walnut Creek, CA, USA
[2]Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[3]National Energy Research Scientific Computing Center (NERSC), Oakland, CA, USA
[4]Lawrence Livermore National Laboratory, Livermore, CA, USA
[5]Convey Computer Corporation, Richardson, TX, USA

Advanced architectures can deliver dramatically increased throughput for genomics and proteomics applications, reducing time-to-completion in some cases from days to minutes. One such architecture, hybrid-core computing, marries a traditional x86 environment with a reconfigurable coprocessor, based on field programmable gate array (FPGA) technology. In addition to higher throughput, increased performance can fundamentally improve research quality by allowing more accurate, previously impractical approaches.

We will discuss the approach used by Convey's de Bruijn graph constructor for short-read, *de-novo* assembly. Bioinformatics applications that have random access patterns to large memory spaces, such as graph-based algorithms, experience memory performance limitations on cache-based x86 servers. Convey's highly parallel memory subsystem allows application-specific logic to simultaneously access 8192 individual words in memory, significantly increasing effective memory bandwidth over cache-based memory systems. Many algorithms, such as Velvet and other de Bruijn graph based, short-read, *de-novo* assemblers, can greatly benefit from this type of memory architecture. Furthermore, small data type operations (four nucleotides can be represented in two bits) make more efficient use of logic gates than the data types dictated by conventional programming models.

JGI is comparing the performance of Convey's graph constructor and Velvet on both synthetic and real data. We will present preliminary results on memory usage and run time metrics for various data sets with different sizes, from small microbial and fungal genomes to very large cow rumen metagenome. For genomes with references we will also present assembly quality comparisons between the two assemblers.

The work conducted by the U.S. Department of Energy Joint Genome Institute is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231