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

https://escholarship.org/uc/item/7sx491xs

Sczyrba, Alex

2011-06-01
Efficient Graph Based Assembly of Short-Read Sequences on a Hybrid Core Architecture

Alex Sczyrba*1,2, Abhishek Pratap1,2, Shane Canon2,3, James Han1,4, Alex Copeland1,2, Zhong Wang1,2, Tony Brewer5, David Soper5, Mike D’Jamoos5, Kirby Collins5, George Vacek5

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

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California.
Introduction
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.

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 offload computation from the CPU to the coprocessor, based on field programmable gate arrays (FPGAs).

The Crux of the Issue

• Throughput for genomics and proteomics
  • Bandwidth over cache-based memory systems. The memory subsystem allows application-specific logic to be moved from the CPU to the coprocessor, based on field programmable gate arrays (FPGAs).

De Bruijn Graph Assembly

• Popularity for short-read de novo sequence assembly
  • Sequences are parsed into “k-mers” as nodes of graph
  • Directed graph edge shows overlap between nodes
  • Graph implemented in memory as hash table-based binary tree
  - Require random access to memory
  - Can require large amounts of memory
  - Memory bandwidth is a limiting factor

Workflow using Velvet

De Bruijn Graph Constructor

• Written from scratch to get maximum use of host and coprocessor
  - Input and output file types compatible with Velvet
  - Graph cleanup approach similar to Velvet

Performance Metrics
We compare the performance of Convey’s GraphConstructor and Velvet using real Illumina data from different genome projects. GraphConstructor runs were performed on Convey’s HC-1 system (host Xeon L5408, 128GB RAM; coprocessor includes 4 Xilinx V5LX330 FPGAs). Velvet was run on a SunFire x4640 (Opteron 8435, 2.6GHz, 512GB RAM).

Microbial Genome Assemblies
Results on run time metrics for 6 small microbial and one fungal genomes. In general, a 2-fold speedup was observed. Assembly statistics in terms of number of contigs, n50, largest scaffold and total assembly size are in agreement with Velvet results.

Convey HC-1 Architecture

• “Commodity” Intel Server
• Convey FPGA-based coprocessor

Convey’s De-Bruijn GraphConstructor

• Copy on write, snapshot, and save
• De Bruijn graph nodes during error correction, which can result in different paths through the graph. This results in slightly different numbers of nodes, n50, and coverage values.
• Convey’s GraphConstructor does all concatenation and renumbering in a single pass at the end of error correction, rather than while corrections are being made. This can result in a different traversal order, which leads to different node numbers and ordering in the contig files.

Convey’s GraphConstructor and Velvet were run on different subsets (between 10Gbp and 160Gbp) of the cow rumen metagenome data set sequenced at the JGI. This version of the GraphConstructor (v0.4.1429) generates contigs directly, resulting in a speedup between 2.2x and 2.8x compared to Velvet. Convey’s implementation reduced the maximum memory usage to 18-71%.

Conclusion

• High performance memory
  - Highly parallel memory access (512 simultaneous)
  - 0.45DMAs optimized for single word memory access
  - Partition graph to fit into coprocessor memory
• Faster performance (up to 2.8x)
  - Small memory footprint (up to 82%)
  - Interface for Velvet
  - Constrains de Bruijn graphs
  - Potential for other assemblies as well

Future work

• Additional performance optimizations
  - Hardware acceleration of roadmap phase
  - Implement ability to read cg hygr binary sequence file directly in velvet for scaffolding
  - Specific optimizations for metagenomics
    - Prefiltering to eliminate low abundance kmers
    - Investigate metagenomes specific scaffolding

References
Huse SD, Development of minimally-degrading gates and genotypes from core rubins. Science 2011;331(6016):602-7

LLNL-POST-474851
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.