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

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]

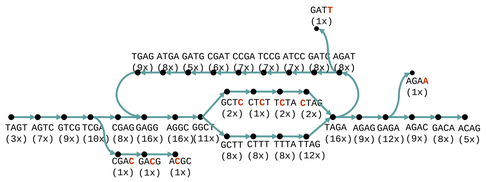
[1]DOE Joint Genome Institute, Walnut Creek, CA, USA [2]Lawrence Berkeley National Laboratory, Berkeley, CA, USA [3]National Energy Research Scientific Computing Center (NERSC), Oakland, CA, USA
[4]Lawrence Livermore National Laboratory, Livermore, CA, USA [5]Convey Computer Corporation, Richardson,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

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.

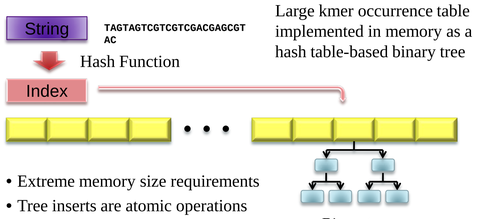
Bioinformatics applications that have random access patterns to large memory spaces, such as graph-based algorithms, experience memory performance limitations on cache-based x86 servers. Convey's highly parallel memory subsystem allows application-specific logic to simultaneously access 8192 individual 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.

De Bruijn Graph Assembly



- Popular 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 limiting factor

Crux of the Issue

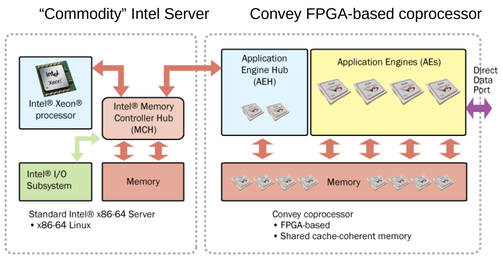


Large kmer occurrence table implemented in memory as a hash table-based binary tree

- Extreme memory size requirements
- Tree inserts are atomic operations
- Reads/writes can be in parallel if collisions avoided
- Memory bandwidth is critical

Binary tree manages hash table collisions

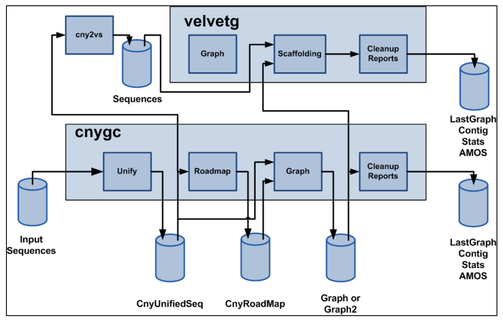
Convey HC-1 Architecture



Convey's De-Bruijn GraphConstructor

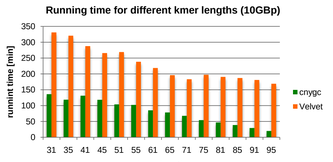
- **Written from scratch to get maximum use of host and coprocessor**
  - Input and output file types compatible with Velvet
  - Graph cleanup approach similar to Velvet
- **Objectives**
  - Accelerate execution
  - Reduce memory requirements
    - Partitionable Roadmap generation phase
    - Rewrite graph construction / read tracking to minimize memory usage

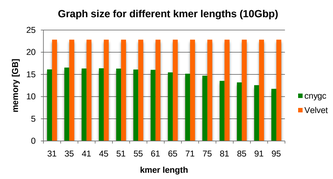
Workflow using Velvetg



Performance Metrics

We compared the performance of Convey's Graph Constructor 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).

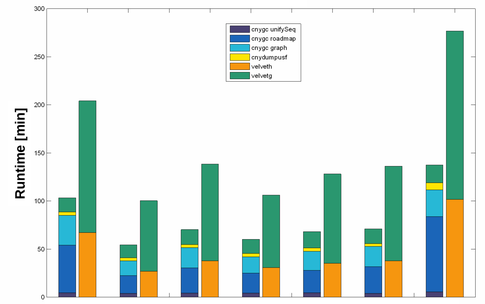




Microbial Genome Assemblies

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.

Cnygc (v0.2.1208) and Velvet (v1.0.18) running times

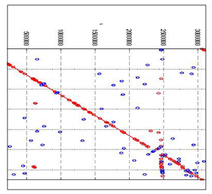


Genome	Arcanobacterium haemolyticum	Brachyspira murdochii	Cellulomonas flavigena	Spirochaeta smaragkinae	Haloterrigena turkmenica	Conexibacter woesei	Trichoderma reesei
GC content	53%	28%	74%	48%	64%	73%	54%
Genome size	2.0 Mb	3.2 Mb	4.1 Mb	4.7 Mb	5.4 Mb	6.4 Mb	33.5 Mb
Data Size	10 Gbp	7.6 Gbp	7.8 Gbp	8.1 Gbp	9.1 Gbp	7.1 Gbp	11.6 Gbp

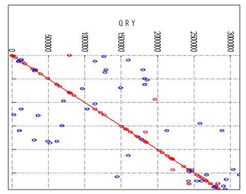
Cnygc Assembly						
contigs	55	262	148	128	308	276	512
n50	1,922,620	1,945,822	4,105,349	1,359,007	216,346	1,324,038	555,467
max contig	1,922,620	1,945,822	4,105,349	1,601,212	1,615,816	3,093,222	2,010,969
total bases	2,004,990	3,194,341	4,178,413	4,683,107	5,574,032	6,416,357	32,184,156
Velvet Assembly							
contigs	95	311	180	190	331	290	514
n50	1,777,978	1,831,953	4,104,164	1,579,090	566,562	3,923,496	587,983
max contig	1,777,978	1,831,953	4,104,164	2,292,977	1,766,707	3,923,496	1,637,422
total bases	2,028,841	3,193,928	4,161,316	4,663,909	5,504,000	6,394,983	32,216,902

Brachyspira murdochii DSM 12563

Reference vs. cnygc

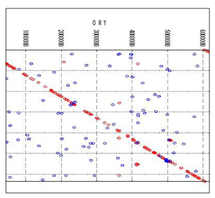


Velvet vs. cnygc

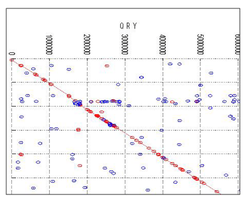


Conexibacter woesei DSM 14684

Reference vs. cnygc



Velvet vs. cnygc

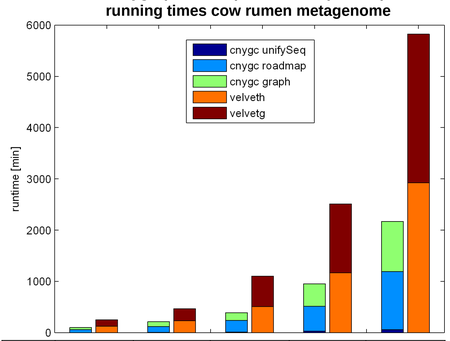


- Convey's GraphConstructor does not resize De-Bruijn graph nodes during error correction, which can result in different paths through the graph. This results in slightly different numbers of nodes, n50, and coverage values.
- Convey's GraphConstructor does all concatenation and renumbering in a single 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.

Cow Rumen Metagenome

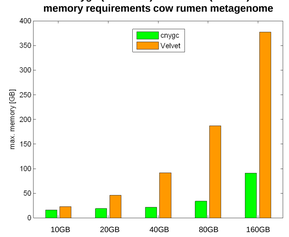
Convey's GraphConstructor and Velvet were run on different subsets (between 10Gbp and 160Gbp) of the cow rumen metagenome data set sequenced at the JGI. This version of the GraphConstructor (v0.4.1429) generates contigs directly, resulting in a speedup between 2.2x and 2.8x compared to Velvet. Convey's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

Cnygc (v0.4.1429) and Velvet (v1.0.19) running times cow rumen metagenome



	10GB	20GB	40GB	80GB	160GB
Cnygc Assembly					
contigs	316,080	608,334	1,102,746	3,008,877	7,769,909
n50	1,342	793	411	351	768
max contig	248,716	133,847	170,217	163,029	224,189
total bases	97,067,342	116,786,227	143,667,023	372,771,310	1,465,466,461
Velvet Assembly					
contigs	378,147	723,504	1,296,513	3,415,955	8,607,580
n50	1,328	779	393	339	757
max contig	185,486	162,087	170,099	163,029	215,607
total bases	97,988,187	117,791,746	144,695,296	373,390,969	1,466,550,531

Cnygc (v0.4.1429) and Velvet (v1.0.19) memory requirements cow rumen metagenome



Conclusion

- High performance memory
  - Highly parallel memory access (8192 simultaneous)
  - SG-DIMMs optimized for single word memory access maximizes bandwidth
- Faster performance (up to 2.8x)
- Smaller memory footprint (up to 82%)
  - Partition graph to fit into coprocessor memory
- Interface for Velvet
  - Constructs de Bruijn graphs
  - Potential for other assemblers as well

Future work

- Additional performance optimizations
  - hardware acceleration of roadmap phase (2x improvement overall for cow rumen)
  - implement ability to read cnygc binary sequence file directly in velvetg for scaffolding
- Specific optimizations for metagenomics
  - prefiltering to eliminate low abundance kmers
  - investigate metagenomics specific scaffolding

References

Zerbino DR, Birney E.: Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.* 2008 May;18(5):821-9
Hess et al.: Metagenomic discovery of biomass-degrading genes and genomes from cow rumen. Science 2011; 331(6016):463-7

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.