University of Virginia
High-Performance Low-Power Lab

Prof. Dr. Mircea Stan

Burrows-Wheeler Short Read Aligner on AWS EC2 F1 Instances

Smith-Waterman Extension on FPGA(s)

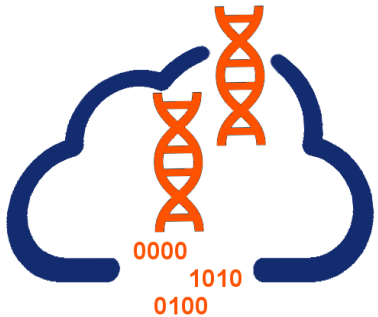
Sergiu Mosanu, Kevin Skadron and Mircea Stan

AACBB, February 23, 2018

Motivation

Why target the cloud for bioinformatics?

- On-demand scalability
  - Increase / decrease resources with demand
  - Lower up-front infrastructure investments
  - Reduced cost of ownership
- Increased performance
  - High-end server machines
  - Equipped with GPU / FPGA accelerators
- Security compliant [1]



Motivation

Why target FPGA acceleration?

- FPGAs are massively parallel
- More power efficient than CPU and GPUs
  - higher performance at lower cost?

Instance	**Accelerator**	**vCPU**	**Memory** [GiB]	**Cost** [USD/h]	
c5.2xlarge	-	8	16	0.34	(1x)
c5.18xlarge	-	72	144	3.06	
f1.2xlarge	1 FPGA	8	122	1.65	(5x)
f1.16xlarge	8 FPGA	64	976	13.20	
p3.2xlarge	1 GPU	8	61	3.06	(9x)
p3.16xlarge	8 GPU	64	488	24.48	

Table: AWS EC2 Instances and On-Demand Pricing

Burrows-Wheeler Short-Read Aligner
Smith-Waterman (SW) Extension

- Available under GPLv3 on `github.com/lh3/bwa`
- Highly optimized, accurate aligner
- Implements SW extension in *ksw_extend2* function
- Includes:
  - BWA-backtrack [2]
  - BWA-SW [3]
  - BWA-MEM [4]

Burrows-Wheeler Short-Read Aligner

Smith-Waterman (SW) Extension

- Iterative algorithm
  - Calculates scoring matrix *H*

```
ksw_extend2(query, target, s_mat, params)
{
   // init H, E, F
   // ...
   for i in [0 to length(target)]
      // ...
      for j in [begin to end]
         H(i,j)   = max{H(i-1,j-1)+S(i,j), E(i,j), F(i,j)}
         E(i+1,j) = max{H(i,j)-gapo, E(i,j)} - gape
         F(i,j+1) = max{H(i,j)-gapo, F(i,j)} - gape
         // ...
      }
      // update begin and end for the next round
      // ...
   }
   return max
}
```

Figure: Code structure of *ksw_extend2* function

Port of SW Extend to FPGA

Optimizations on *ksw_extend2* in SDAccel

- *ksw_extend2* kernel implemented in Xilinx SDAccel
  - Original code largely preserved
- Fixed *query* and *target* lengths to 256 symbols
- Similarity function implemented in logic
- Reduced variables from (u)int to (u)short
- Changed few variable declarations local to loop
  - Loop-carry dependency set to false with HLS pragmas
- Reduced BRAM accesses by storing previous iteration values
- Pipelined all but loop-*i* with HLS pragmas
- Achieved functional correctness

Port of SW Extend to FPGA

Utilization and Performance Results

- Max frequency of 330MHz, well above 250MHz
- Average kernel execution time: 0.17ms
- Host chrono results: FPGA 333ms vs 54ms CPU
  - CPU matched with 6 *ksw_extend2* parallel instances on FPGA
- Min 80 *ksw_extend2* instances to fit on single FPGA

	LUT	LUTMem	REG	BRAM	DSP
User Budget	890.6k	552.1k	1985k	1615	6828
ksw_ext2	6407 ($< 1\%$)	1550	11k	21 ($\approx 1.3\%$)	1

Table: FPGA utilization with 1 BWA *ksw_extend2* instance

Proposed Single-FPGA Multi-Threaded architecture

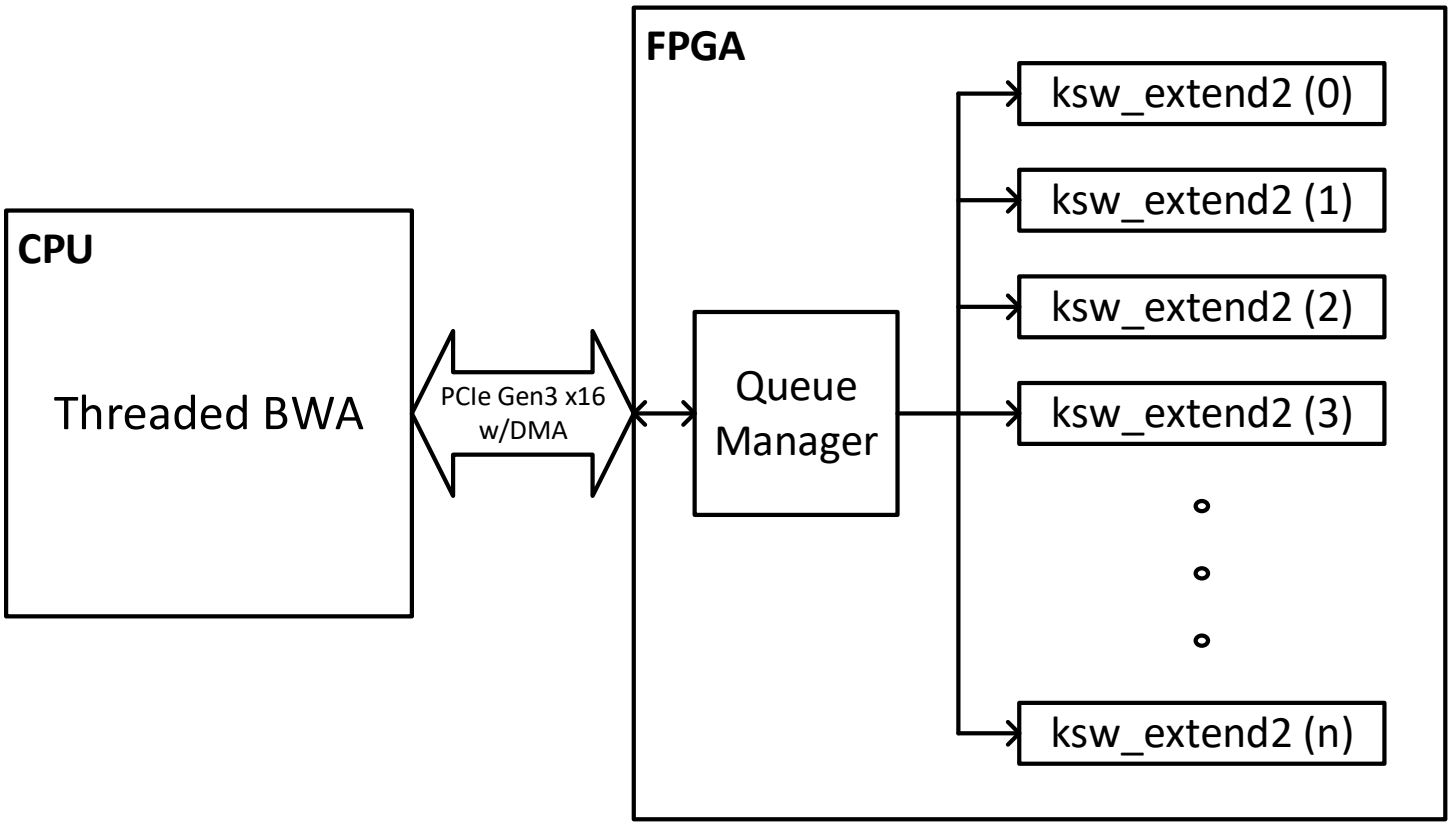


Figure: Multi-threaded single-FPGA architecture

Proposed Cross-FPGA Multi-Threaded architecture

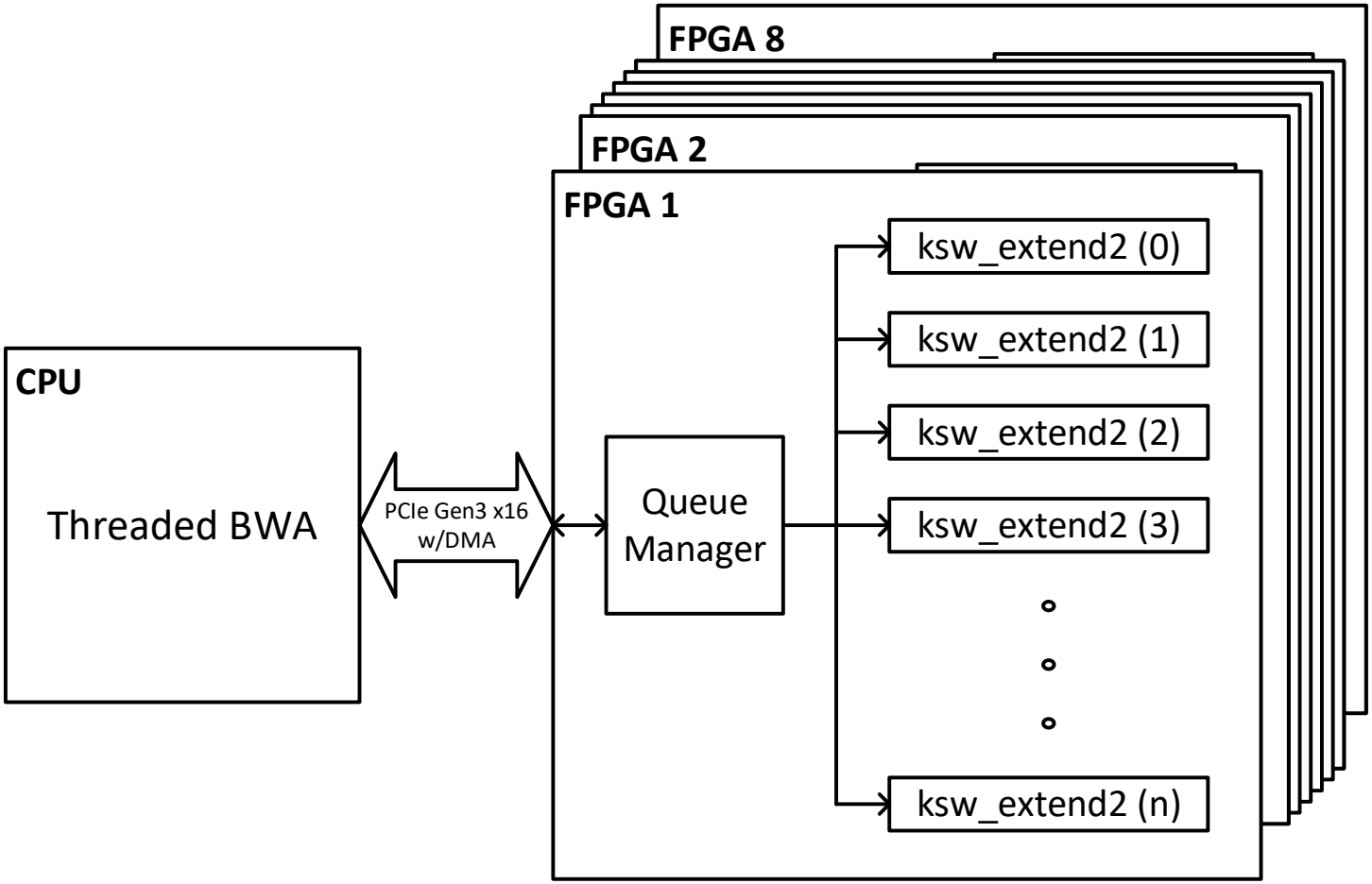


Figure: Multi-threaded cross-FPGA architecture

Estimated Benefits
Lighter SW Extend step

- ≈13x speedup for 80 BWA *ksw_extend2* instances on F1 2xLarge machine (single FPGA)
- ≈100x speedup for cross-FPGA multi-threaded architecture on F1 16xLarge machine (8 FPGAs)
  - Both result in ≈4x cost saving compared with equivalent EC2 machines with no accelerators

Conclusion and future work

- AWS EC2 F1 is a promising platform for bioinformatics
- SW Extension on FPGA with SDAccel

- Further optimize BWA *ksw_extend2*
- Complete multi-threaded architectures
- Integrate with rest of BWA and benchmark

Conclusion and future work

- AWS EC2 F1 is a promising platform for bioinformatics
- SW Extension on FPGA with SDAccel

- Further optimize BWA *ksw_extend2*
- Complete multi-threaded architectures
- Integrate with rest of BWA and benchmark

Thank you!

- Code available at: `github.com/hplp/BWA_HLS`

References


A. Pizarro, C. Whalley, "Architecting for Genomic Data Security and Compliance in AWS", Amazon Web Services, December 2014.


H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows-Wheeler transform", Bioinformatics, 2009


H. Li and R. Durbin, "Fast and accurate long-read alignment with Burrows-Wheeler transform", Bioinformatics, 2010


H. Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM", arXiv:1303.3997v2, 2013