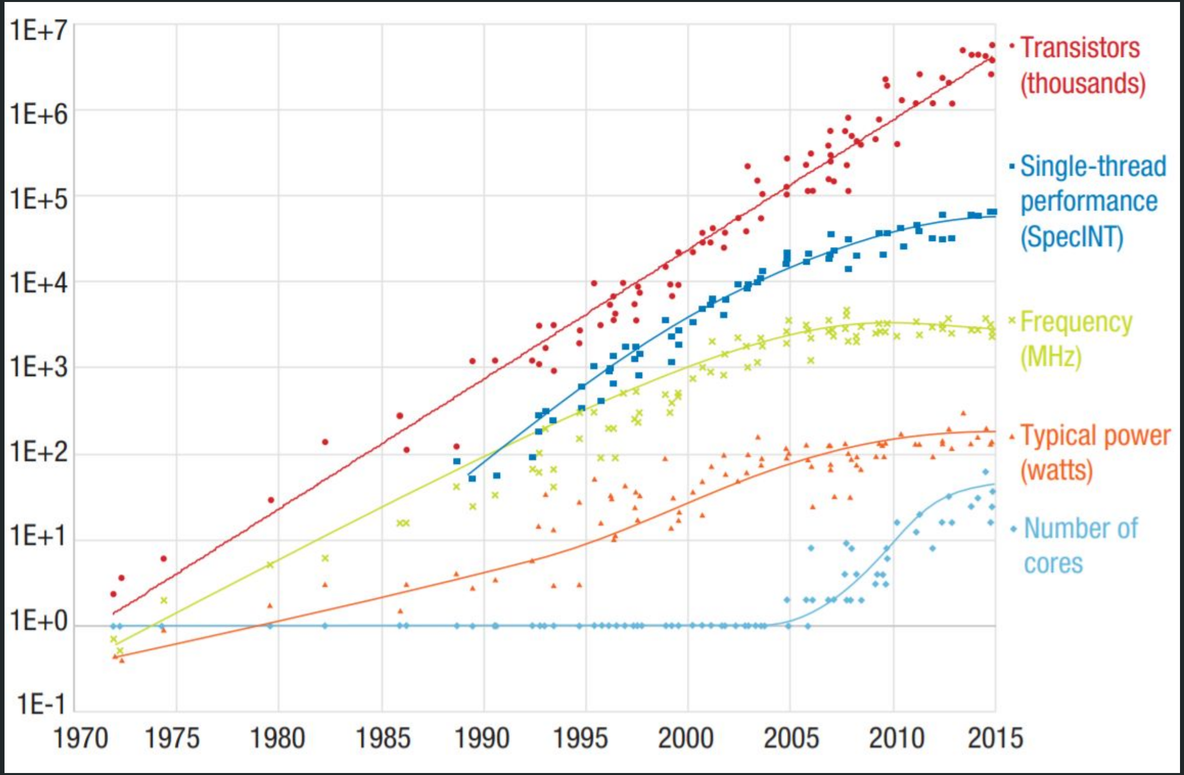
Domain Specific Computing in Tightly-Coupled Heterogeneous Systems

Anthony Cabrera
PhD Dissertation Defense
Department of Computer Science and Engineering
Washington University in St. Louis
July 22, 2020

It’s the end of **Moore’s Law** as we know it

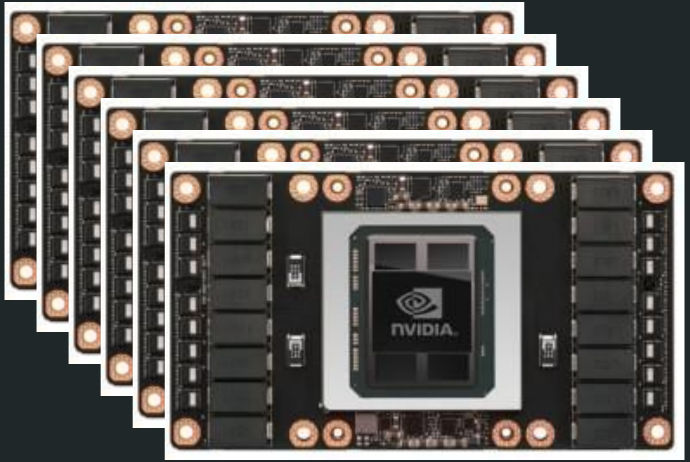


KM Bresniker et al., Computer ‘15

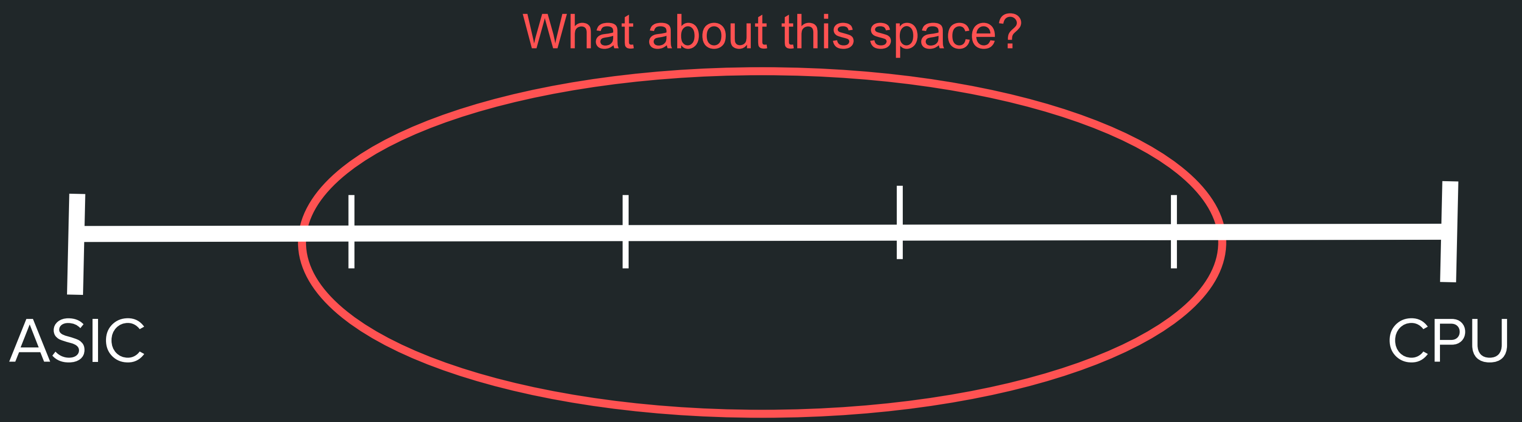
Queue Heterogeneity!

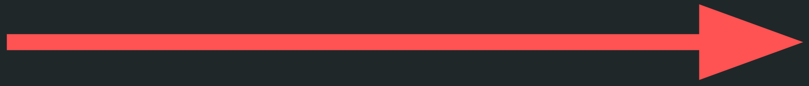
Summit Supercomputer @ Oak Ridge National Lab

6X NVidia V100 GPUs



Hardware Specialization Spectrum





More specialized

Less specialized

The Path Forward

Don’t try to build a general purpose processor that does everything well; build a processor that builds a few tasks incredibly well, and figure out how to build a heterogeneous architecture using those techniques.

David Patterson and John Hennessy

Our Vision

Enable the paradigm shift towards domain specific computing by identifying application domains and architecting performant domain-specific hardware.

Outline

$$\begin{tikzcd}
 & \text{Domain Specific Computing} \arrow[dl] \arrow[dr] & \\
\text{Domain Identification} \arrow[dr] & & \text{Hardware Platform Evaluation} \arrow[dl] \\
 & \text{Architecting Domain Specific Hardware} &
\end{tikzcd}$$

Outline

$$\begin{tikzcd}
 & \text{Domain Specific Computing} \arrow[dl] \arrow[dr] & \\
\text{Domain Identification} \arrow[dr] & & \text{Hardware Platform Evaluation} \arrow[dl] \\
 & \text{Architecting Domain Specific Hardware} &
\end{tikzcd}$$

How do you identify a domain?

How do you identify a domain?

Qualitatively

- Create the Data Integration Benchmark Suite

Quantitatively

- Characterize workload

Cabrera et al. *DIBS: A Data Integration Benchmark Suite*. ICPE `18

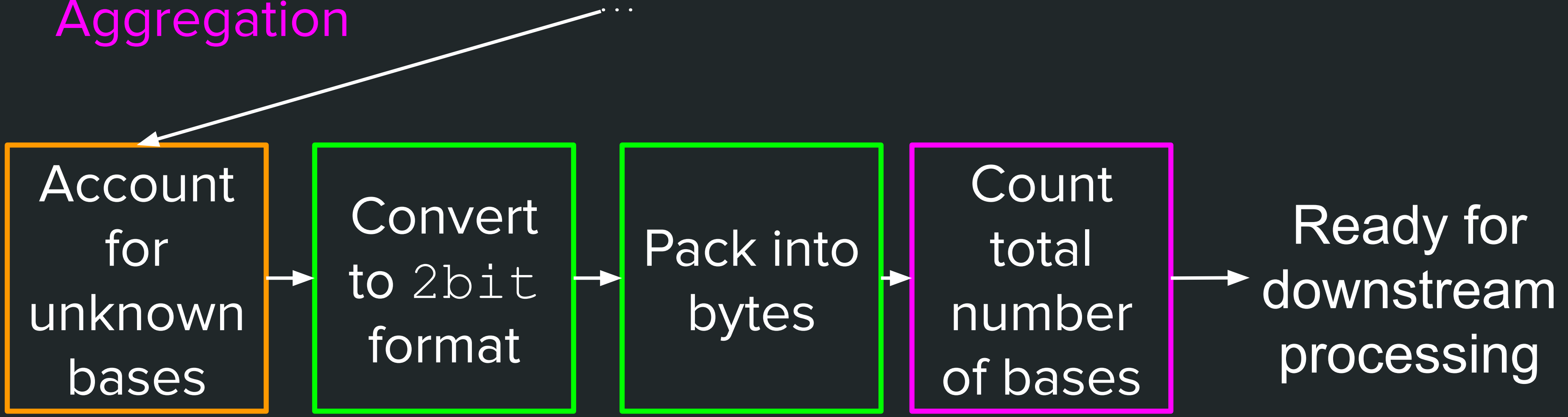
Data Integration

Parsing/Cleansing

Transformation

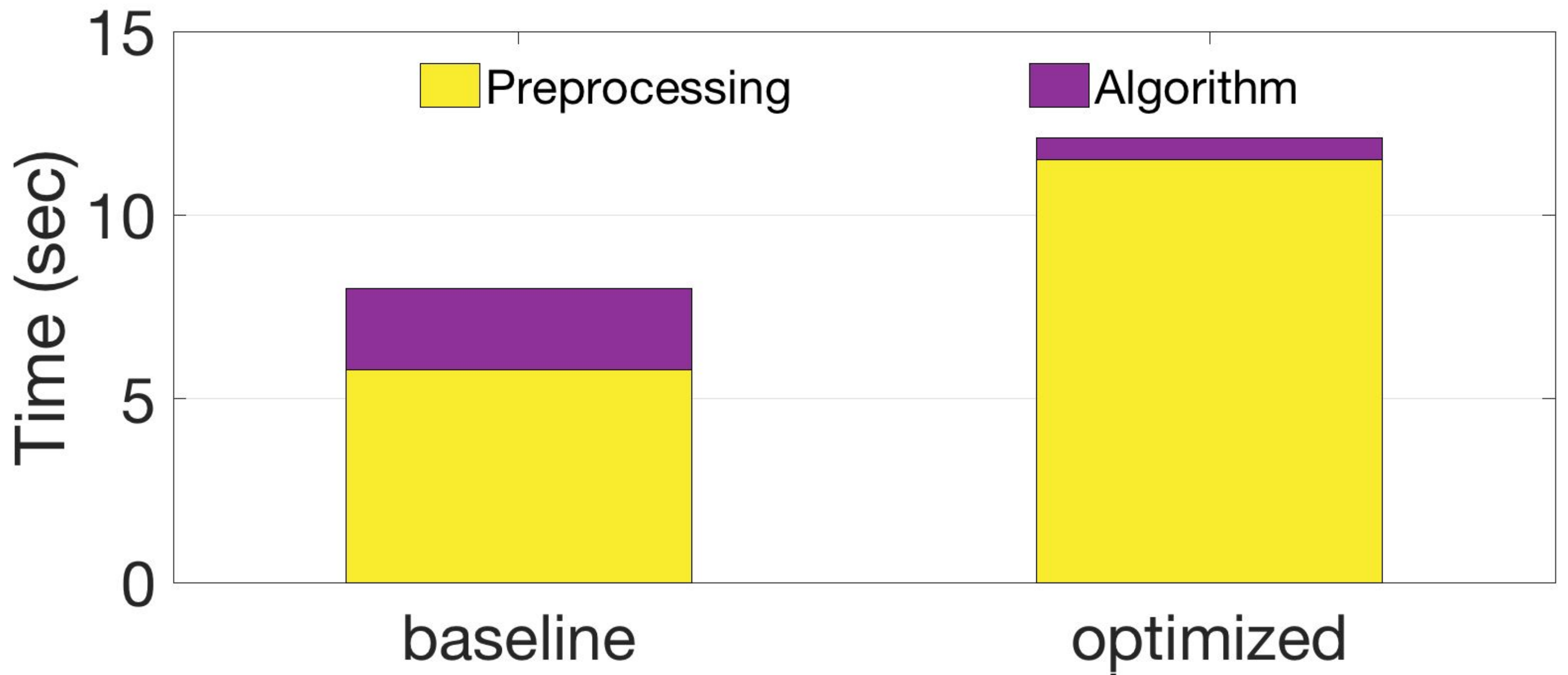
Aggregation

```
>Some Sequence of Interest
...
agcaagacttcatctcaaaaaaaaaaaaaaaGCTGCANATTTattattat
tattattagtttatttatttattttttgagacagagtctcgttctgtcg
cccaggctggagtgcggtggcgtgatcttggctcattgcaacctccacct
cccgggttcaagtgattctcctgcctcagcctcccgagtagctgggacta
caggcgtatgccaccatgcctggctaattttttgtactttagtagagac
Agagtttcacggtgttagccaggctggtcttgatctcctgacctcgtgat
...
```


Account for unknown bases
Convert to 2bit format
Pack into bytes
Count total number of bases
Ready for downstream processing

The Preprocessing Pain Point

BFS on Twitter Data



Characterization Conclusions

Quantitative Characterization

- Consistency in Locality
- Prevalence of Data Movement

Use this to inform domain specific hardware

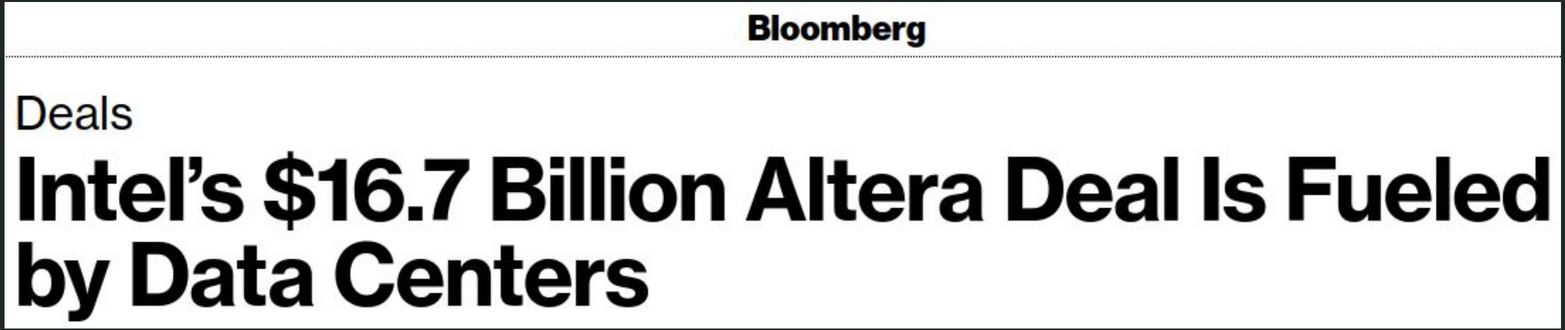
Outline

$$\begin{tikzcd}
 & \text{Domain Specific Computing} \arrow[dl] \arrow[dr] & \\
\text{Domain Identification} \arrow[dr] & & \text{Hardware Platform Evaluation} \arrow[dl] \\
 & \text{Architecting Domain Specific Hardware} &
\end{tikzcd}$$

Outline

$$\begin{tikzcd}
 & \text{Domain Specific Computing} \arrow[dl] \arrow[dr] & \\
\text{Domain Identification} \arrow[dr] & & \text{Hardware Platform Evaluation} \arrow[dl] \\
 & \text{Architecting Domain Specific Hardware} &
\end{tikzcd}$$

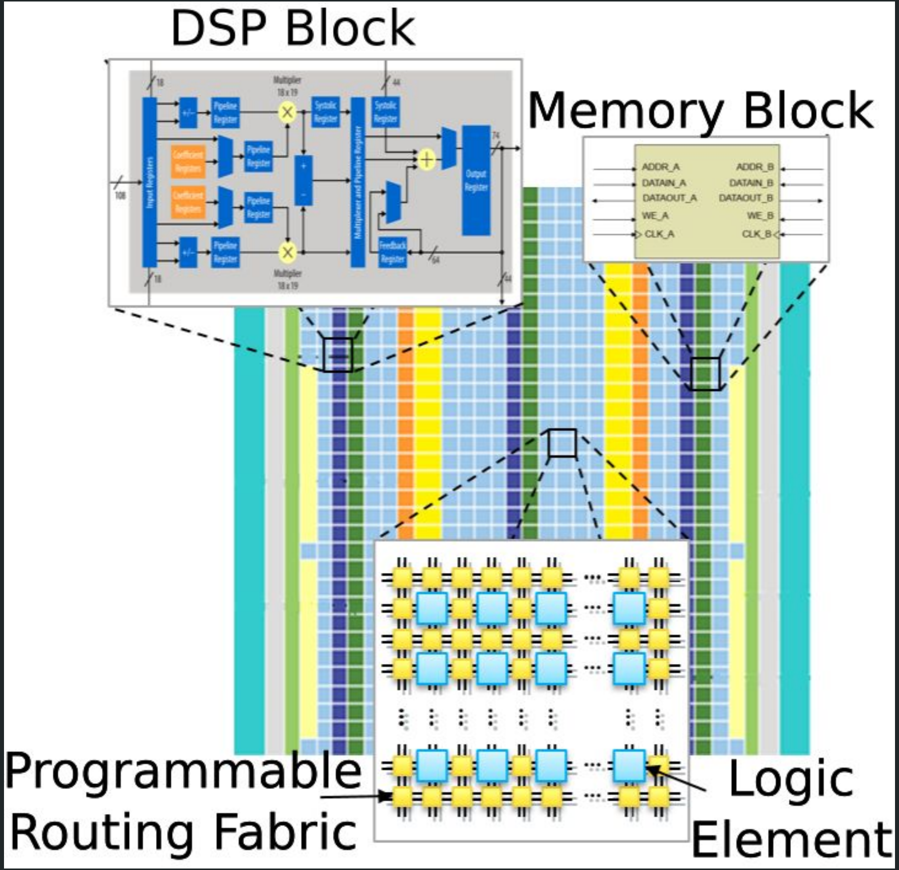
How about FPGAs as our platform?


Bloomberg

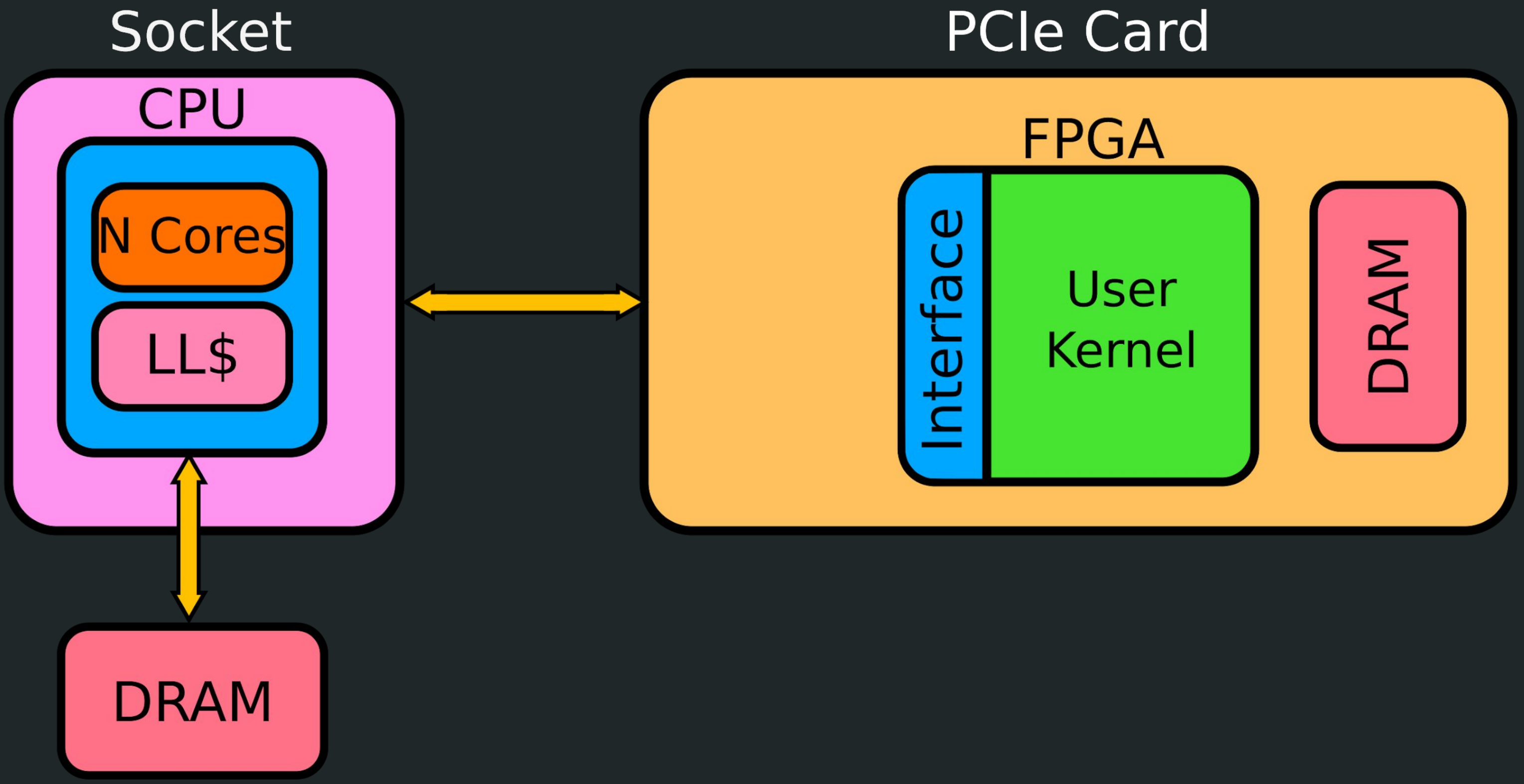
Deals

Intel's $16.7 Billion Altera Deal Is Fueled by Data Centers

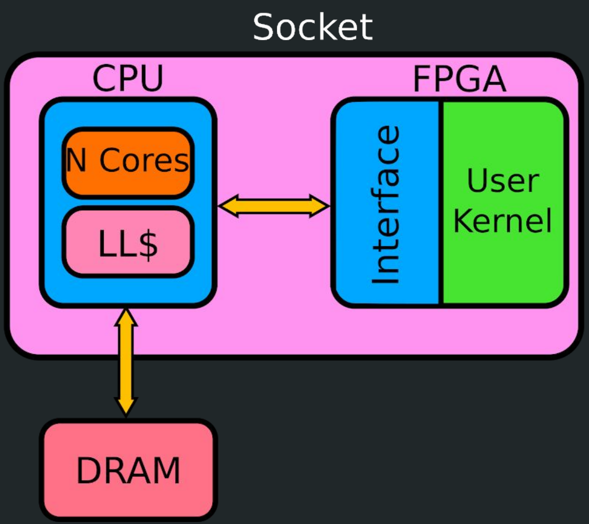
Wait, what's an FPGA, anyway?

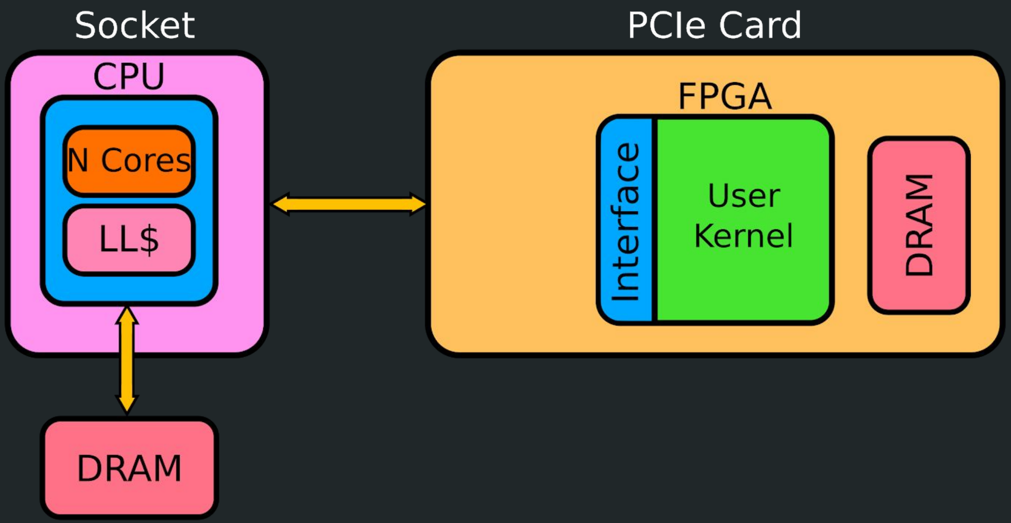


FPGA attached via PCIe card

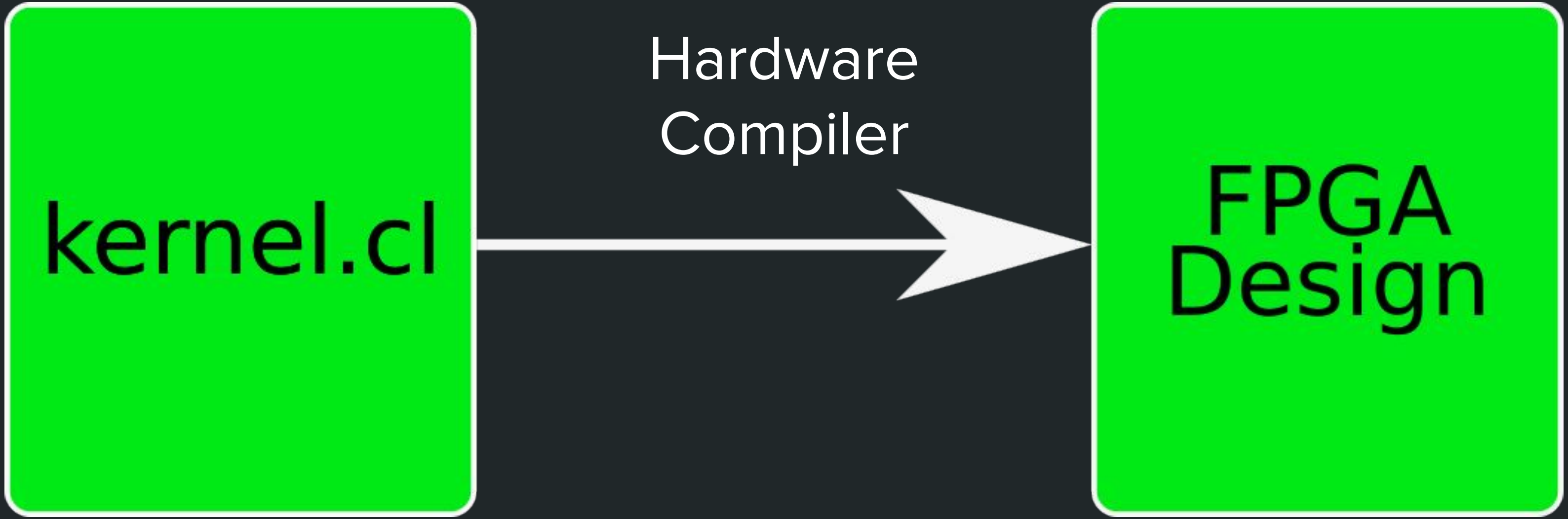


Intel HARPv2 (left) vs. PCIe card (right)





OpenCL to the Rescue!



How portable are OpenCL FPGA kernels to Intel HARPv2?

Results

Version	FPGA	Speedup

SVP = Stratix V, PCIe
HARP= Arria 10, HARP

🟦 Zohouri et al., SC`18
🟩 Cabrera et al., IWOCL`19

Results

SVP = Stratix V, PCIe
HARP= Arria 10, HARP

🟦 Zohouri et al., SC`18
🟩 Cabrera et al., IWOCL`19

Version	FPGA	Speedup
1	SVP	1.00
	HARP	0.74
2	SVP	0.05
	HARP	0.01
3	SVP	2.48
	HARP	3.90
4	SVP	3.55
	HARP	3.24
5	SVP	38.22
	HARP	34.27

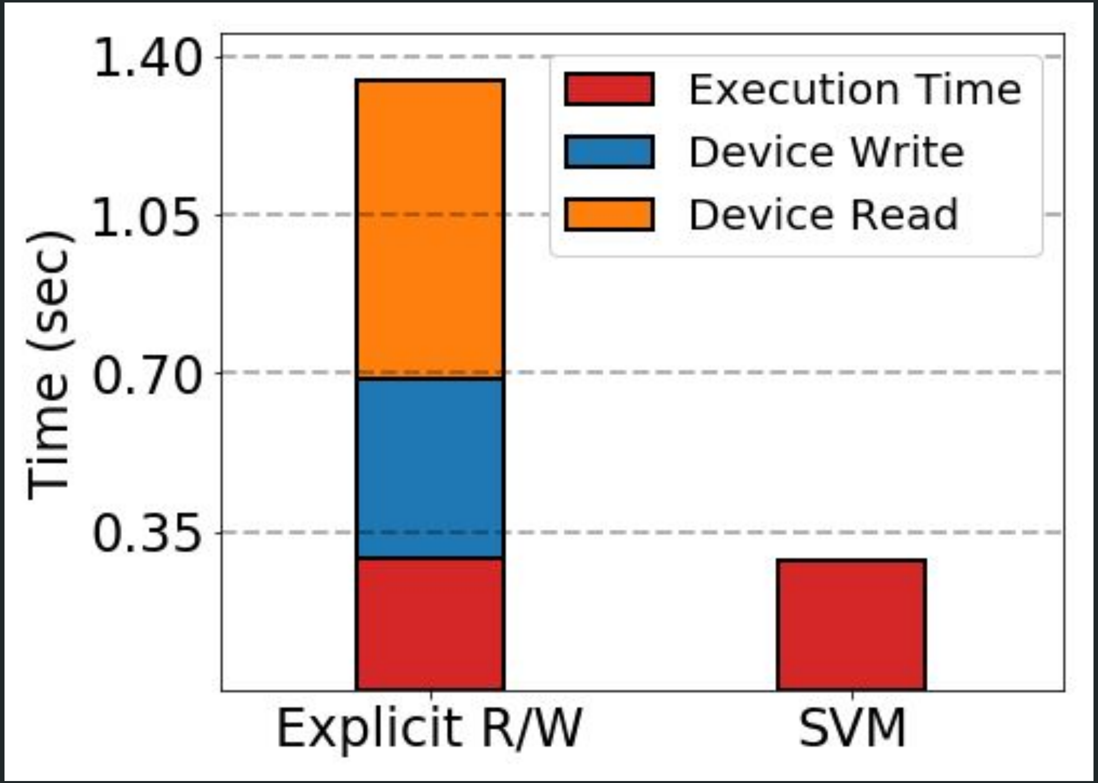
Results

SVP = Stratix V, PCIe
HARP= Arria 10, HARP

🟦	Zohouri et al., SC`18
🟩	Cabrera et al., IWOCL`19

Version	FPGA	Speedup
1	SVP	1.00
	HARP	0.74
2	SVP	0.05
	HARP	0.01
3	SVP	2.48
	HARP	3.90
4	SVP	3.55
	HARP	3.24
5	SVP	38.22
	HARP	34.27

Result of Exploiting Shared Virtual Memory



Results

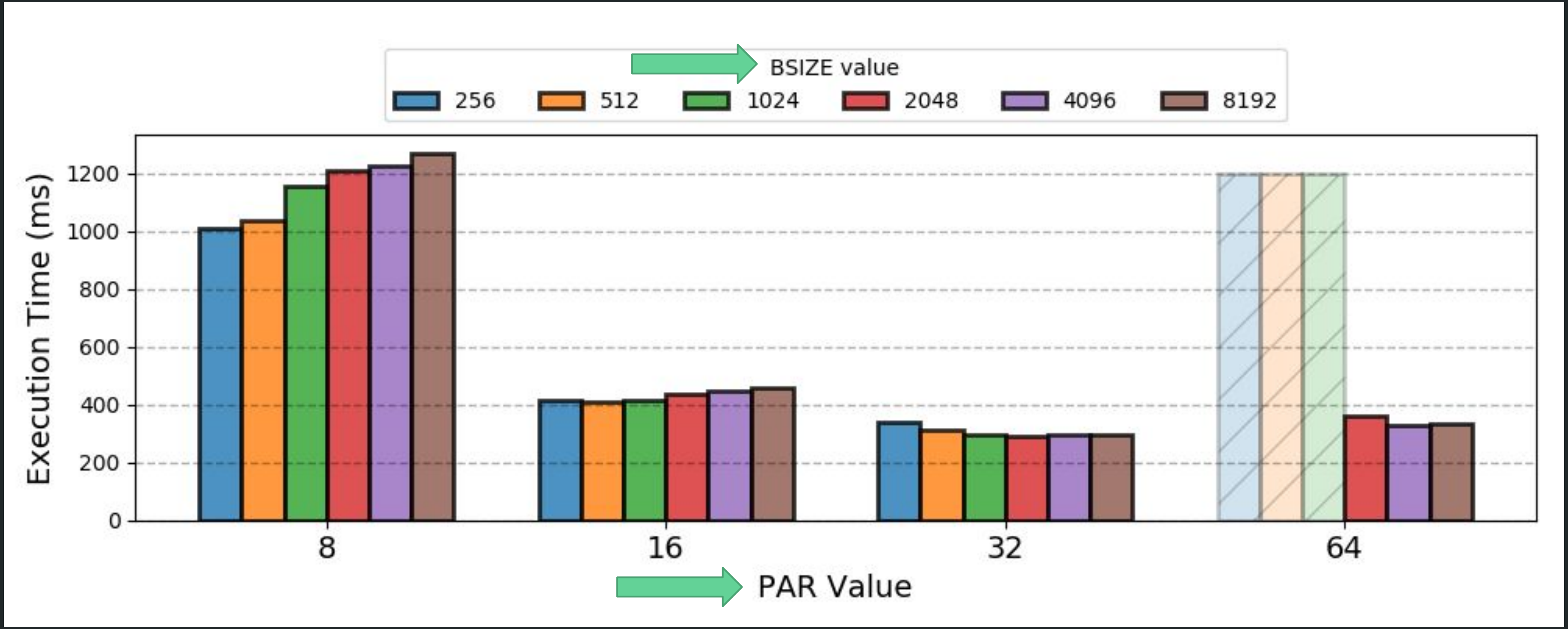
SVP = Stratix V, PCIe
HARP= Arria 10, HARP

🟦 Zohouri et al., SC`18
🟩 Cabrera et al., IWOCL`19

Version	FPGA	Speedup
5	SVP	38.22
	HARP	34.27

How do you find the most performant knob configuration?

Hardware Design Parameter Sweep



Outline

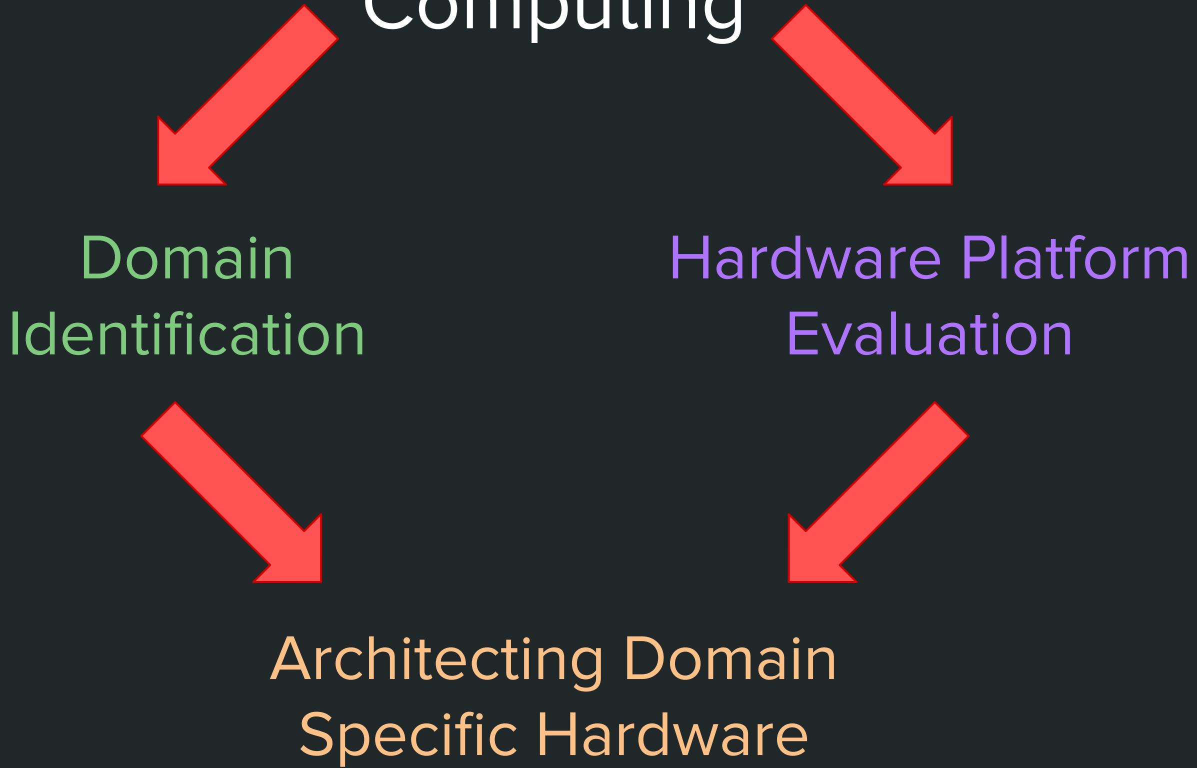
$$\begin{tikzcd}
 & \text{Domain Specific Computing} \arrow[dl] \arrow[dr] & \\
\text{Domain Identification} \arrow[dr] & & \text{Hardware Platform Evaluation} \arrow[dl] \\
 & \text{Architecting Domain Specific Hardware} &
\end{tikzcd}$$

Outline

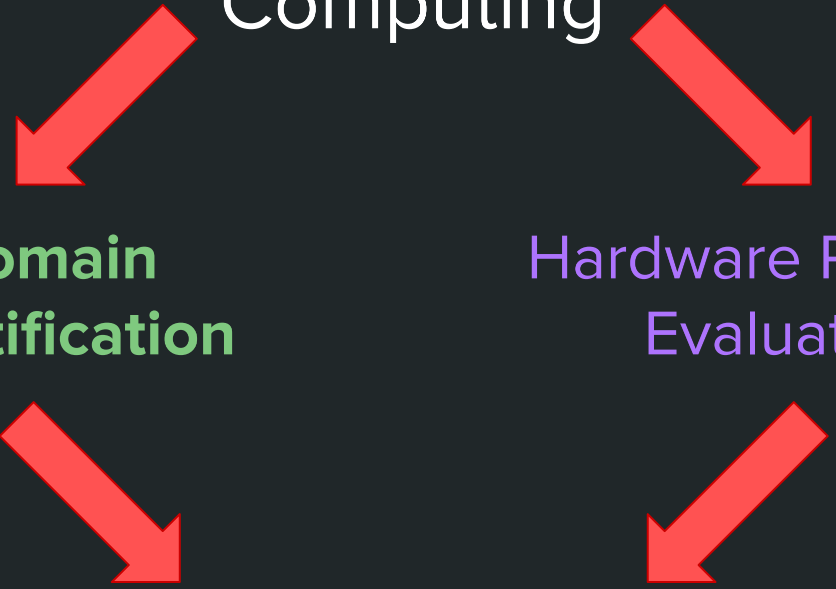
$$\begin{tikzcd}
& \text{Domain Specific Computing} \arrow[dl] \arrow[dr] & \\
\text{Domain Identification} \arrow[dr] & & \text{Hardware Platform Evaluation} \arrow[dl] \\
& \textbf{Architecting Domain Specific Hardware} &
\end{tikzcd}$$

How do you design performant hardware for a specific domain?

Domain Specific Computing


Domain Identification
Hardware Platform Evaluation
Architecting Domain Specific Hardware

Domain Specific Computing



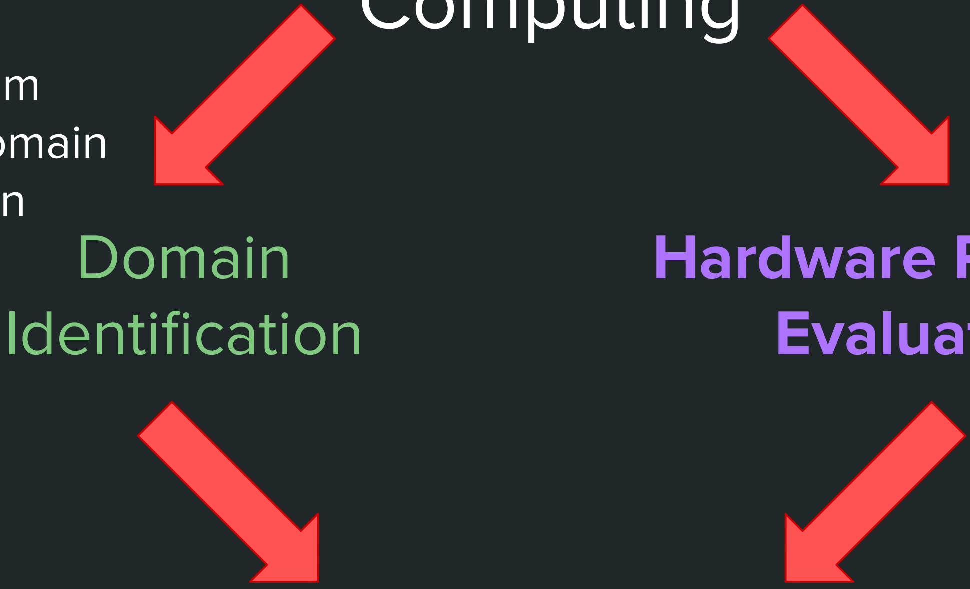
Use insights from data integration domain characterization

Domain Identification

Hardware Platform Evaluation

Architecting Domain Specific Hardware

Domain Specific Computing



Use insights from data integration domain characterization

Leverage results from HARPv2 portability and performance evaluation

Domain Identification

Hardware Platform Evaluation

Architecting Domain Specific Hardware

Domain Specific Computing

Use insights from data integration domain characterization

Leverage results from HARPv2 portability and performance evaluation

Domain Identification

Hardware Platform Evaluation

Design a data-driven method to guide HW design choices

Architecting Domain Specific Hardware

Multi-spectral Reuse Distance

Develop a tool to inform the relationship between spatial and temporal locality

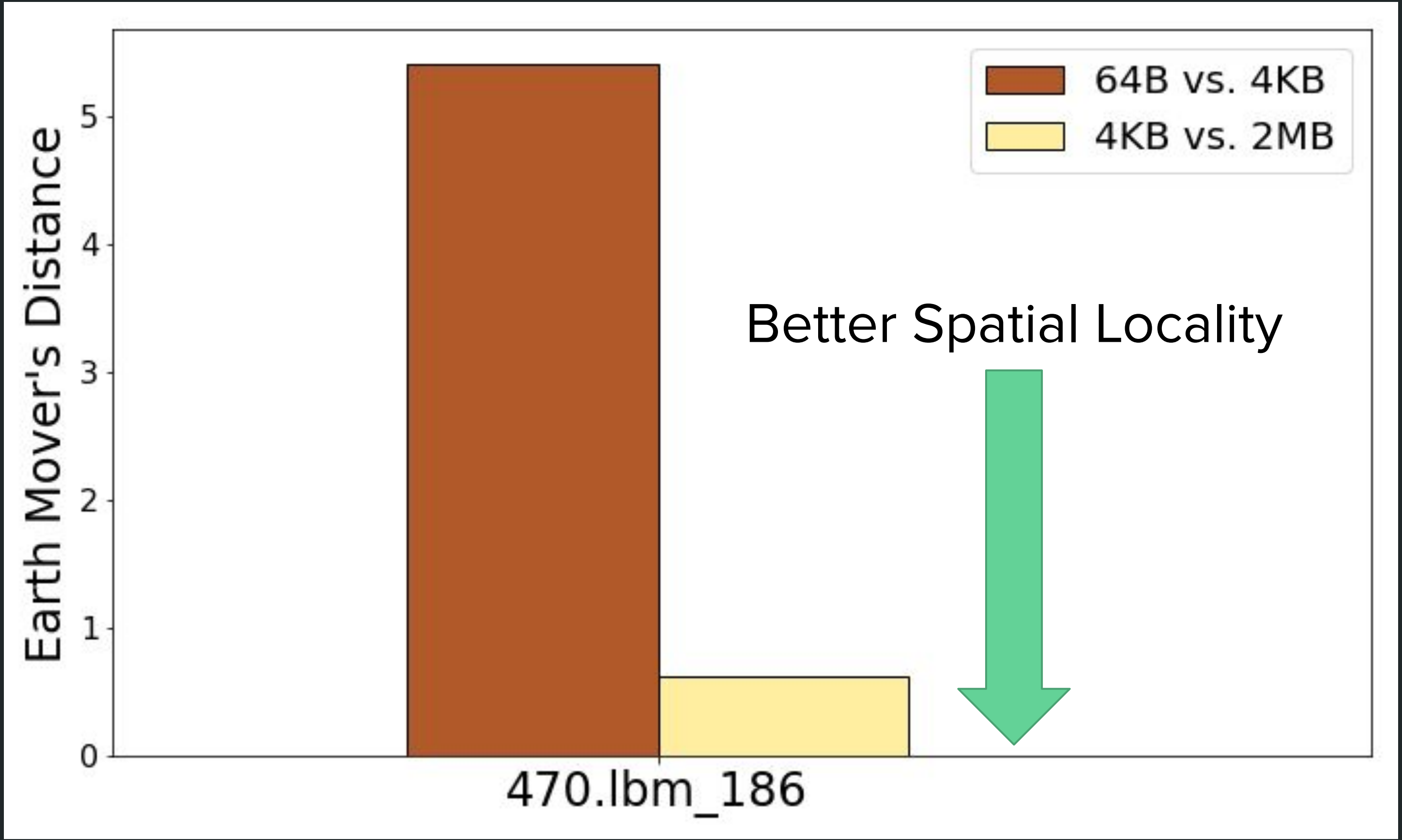
Cabrera et al. Multi-spectral Reuse Distance: Divining Spatial Information from Temporal Data. HPEC `19

Method Overview

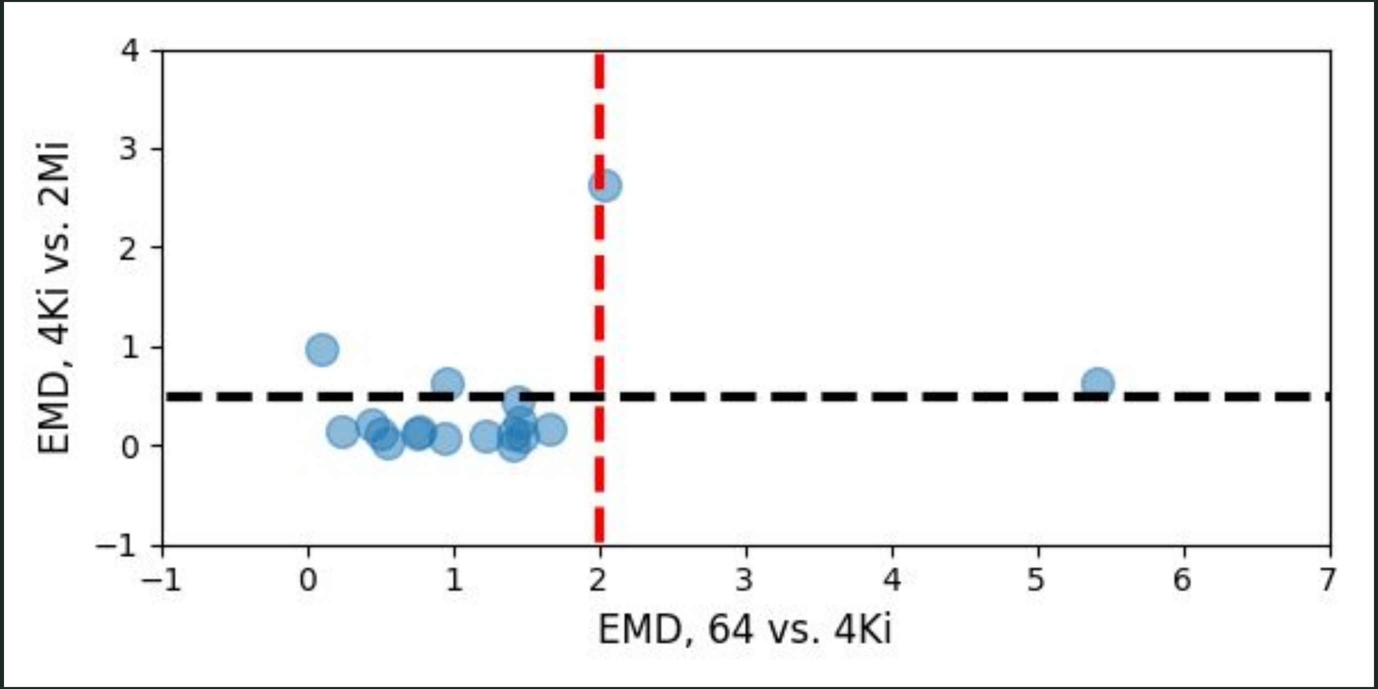
Code Regions of Interest + DynamoRIO The DR. is in. = Instruction Trace

Instruction Trace + Reuse Distance @ 64B, 4KiB, 2MiB Granularities + Earth Mover's Distance

Quantifying Spatial Locality with EMD



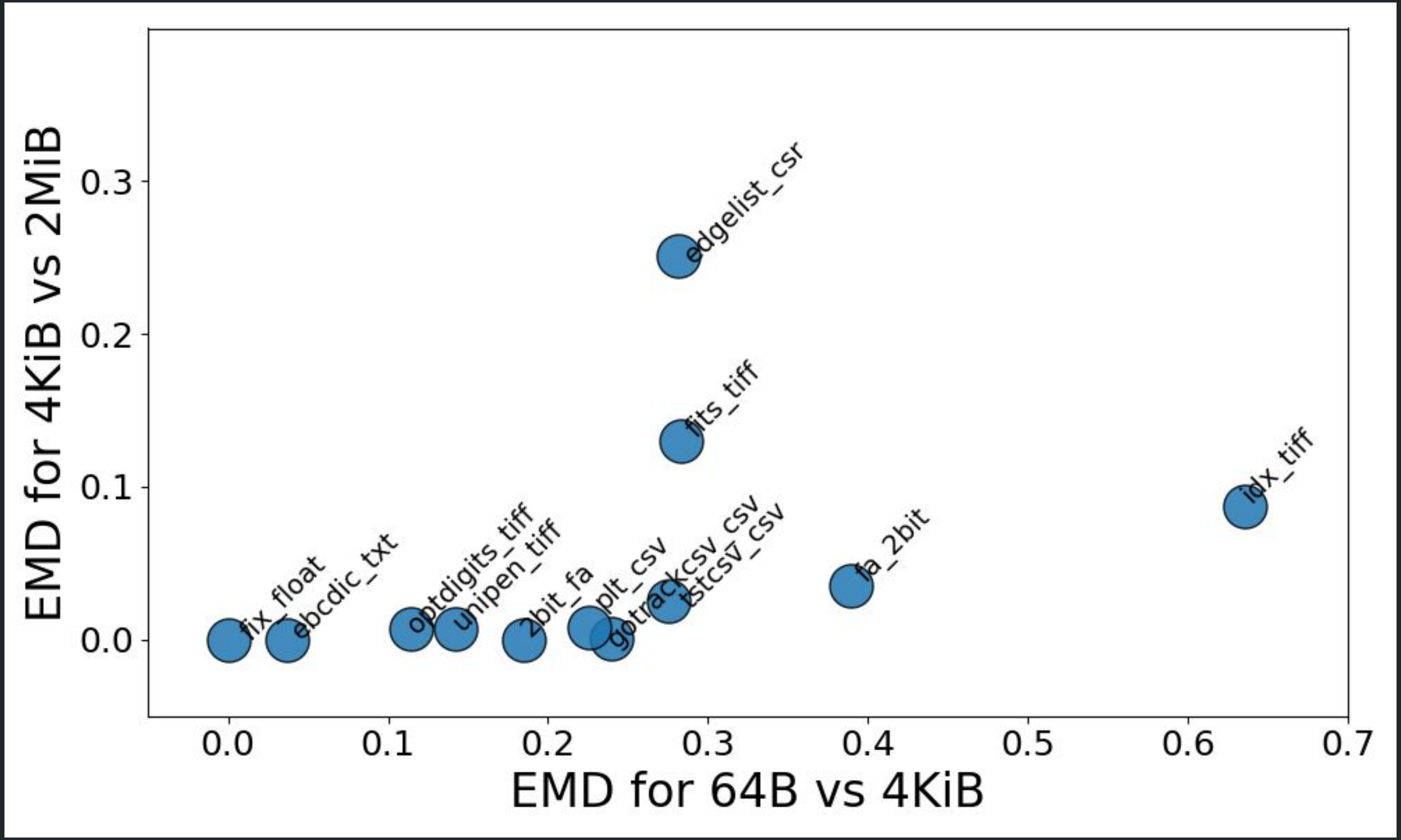
EMD for SPEC2006 Applications



Sub-Domain Creation

Multi-spectral Reuse Distance + DIBS Applications =

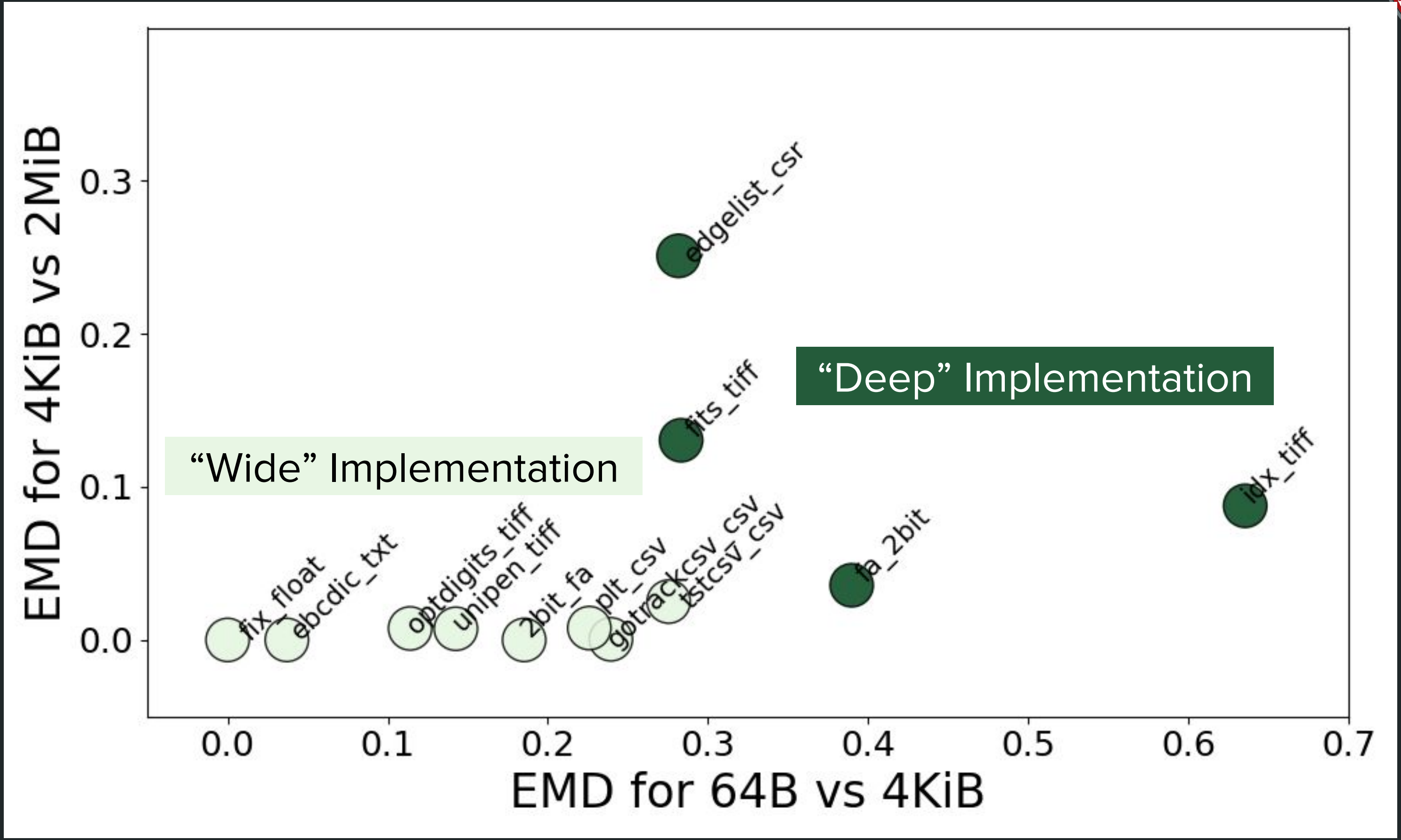
DIBS Multi-spectral Reuse Distance Results



Sub-Domain Creation

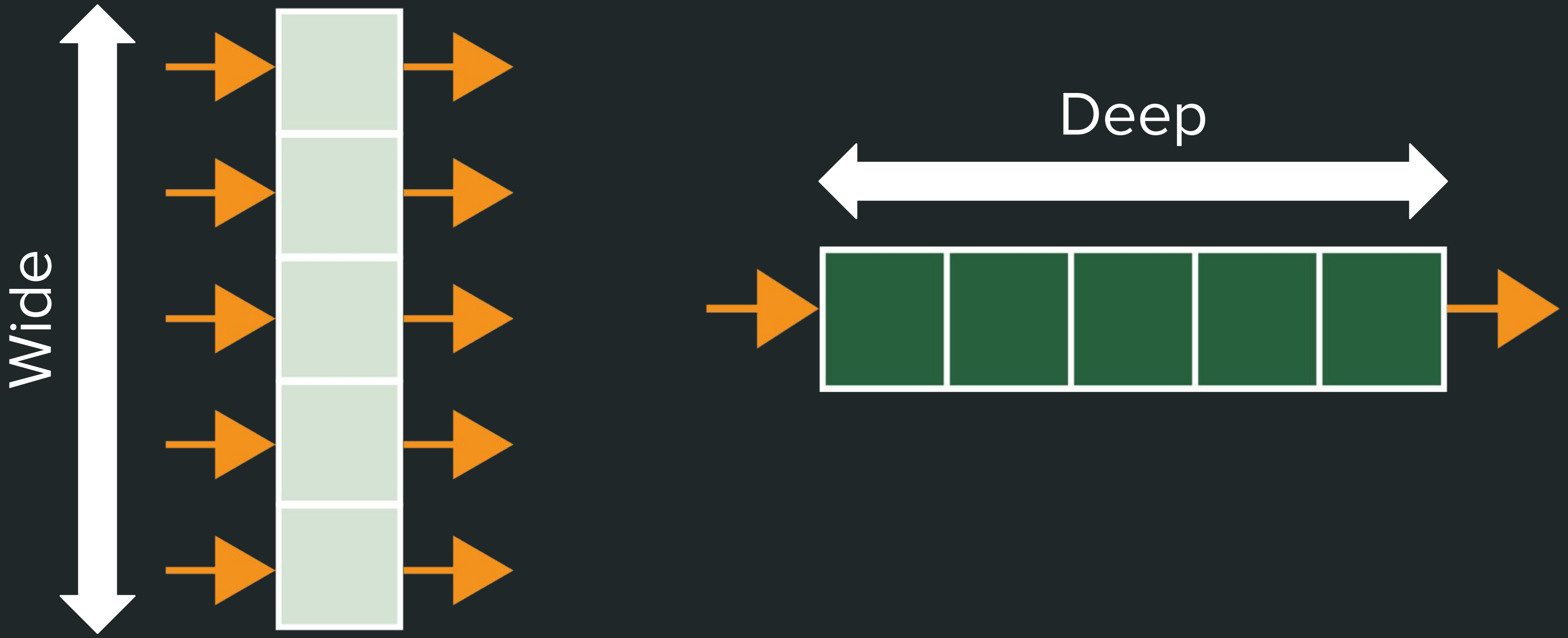
Multi-spectral Reuse Distance + DIBS Applications + *k*-means clustering

Clustering Applications with *k*-means



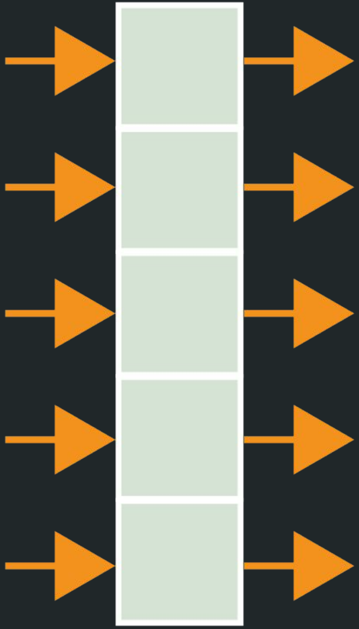
Width vs. Depth

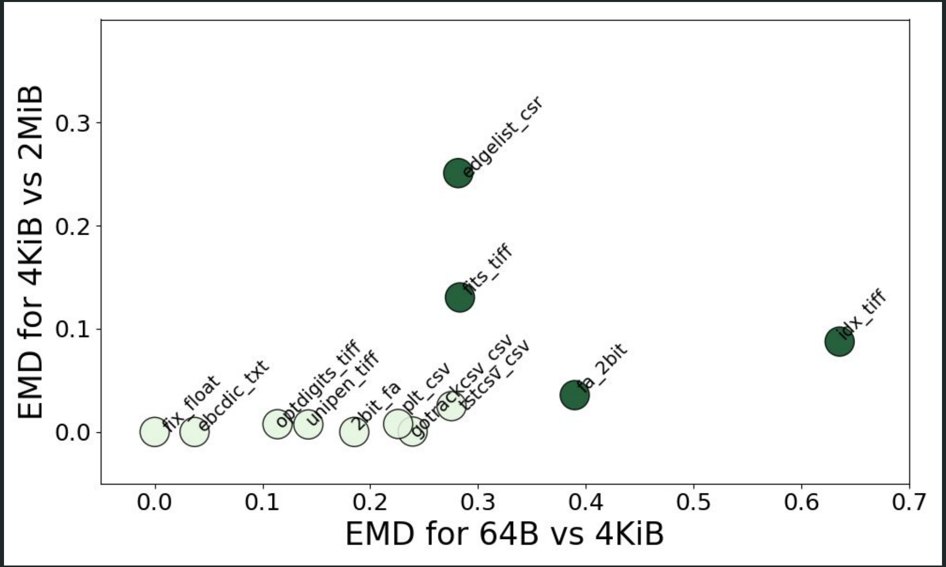
The two OpenCL FPGA Design Paradigms



k-means Clustering

The two OpenCL FPGA Design Paradigms





DIBS Subset

Application	Clustering Prediction
`ebcdic_txt`	Wide
`idx_tiff`	Deep
`fix_float`	Wide
`edgelist_csr`	Deep
`2bit_fa`	Wide
`fa_2bit`	Deep

Let’s Talk OpenCL Hardware Design

Wide Kernel Design

Deep Kernel Design

Case Study: `ebcdic_txt` Wide Kernel

3) Width Knobs

```
__kernel void e2a(
```

2) Kernel Arguments

```
{
```

1) Kernel Body

```
}
```

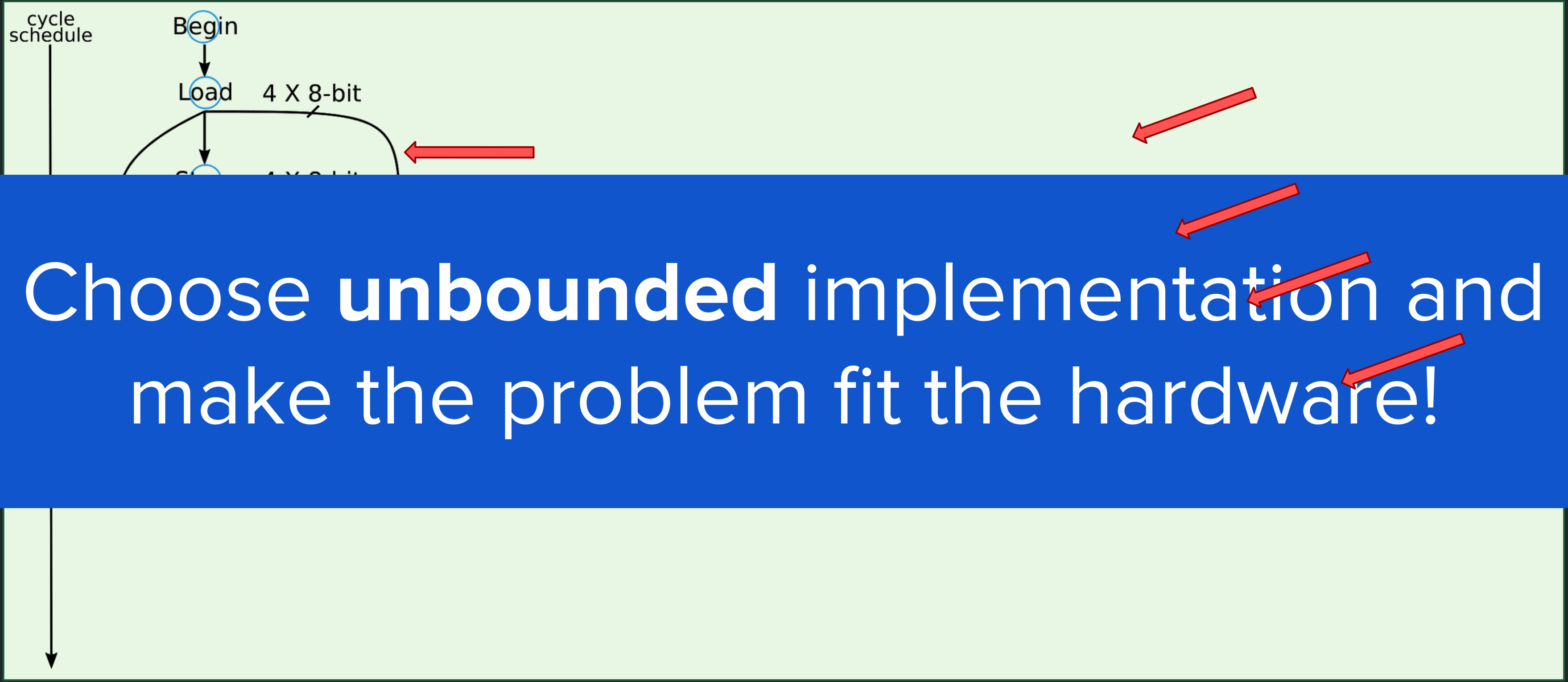
What About the “Loose Ends”?

```
__kernel void e2a(   __global const uchar* restrict src,
                     __global uchar* restrict dst,
                     ulong total_work_items)
{
    unsigned char e2a_lut[256] = { … };
    unsigned int i = get_global_id(0);
    uchar orig_char = src[i];
    uchar xformd_char;
    xformd_char = e2a_lut[orig_char];
    dst[i] = xformd_char;
}
```

```
__kernel void e2a(   __global const uchar* restrict src,
                     __global uchar* restrict dst)
{
    unsigned char e2a_lut[256] = { … };
    if (i < total_work_items) {
        unsigned int i = get_global_id(0);
        uchar orig_char = src[i];
        uchar xformd_char;
        xformd_char = e2a_lut[orig_char];
        dst[i] = xformd_char;
    }
}
```

Unbounded (left) vs. Bounded (right)

4 replicates



Choose **unbounded** implementation and make the problem fit the hardware!

Case Study: `ebcdic_txt` Wide Kernel

3) Width Knobs

```
__kernel void e2a(  __global const uchar* restrict src,
                    __global uchar* restrict dst)
{
    unsigned char e2a_lut[256] = { … };
    unsigned int i = get_global_id(0);
    uchar orig_char = src[i];
    uchar xformd_char;
    xformd_char = e2a_lut[orig_char];
    dst[i] = xformd_char;
}
```

`ebcdic_txt` Coarse-Grain Width Knobs

```
__attribute__((num_compute_units(NUMCOMPUNITS)))
__attribute__((reqd_work_group_size(WGSIZE,1,1)))
__attribute__((num_simd_work_items(NUMSIMD)))
```

NUMCOMPUNITS = # of replicated compute units

WGSIZE = work-group size of compute unit

NUMSIMD = # of times data path is replicated within a compute unit

`ebcdic_txt` Width Design Space

`WGSIZE` = {128, 256, 512, 1024}

`NUMCOMPUTEUNITS` = {1, 2, 4, 8}

`NUMSIMD` = {1, 2, 4, 8, 16}

Case Study: `ebcdic_txt` Deep Kernel

```
__kernel void e2a(  __global const uchar* restrict src,
                    __global uchar* restrict dst,
                    ulong num_elts)
{
    unsigned char e2a_lut[256] = { … };
    unsigned int i;
    #pragma unroll UNROLL
    for (i = 0; i < num_elts; ++i) {
        uchar xformd_char;
        xformd_char = e2a_lut[orig_char];
        dst[i] = xformd_char;
    }
}
```

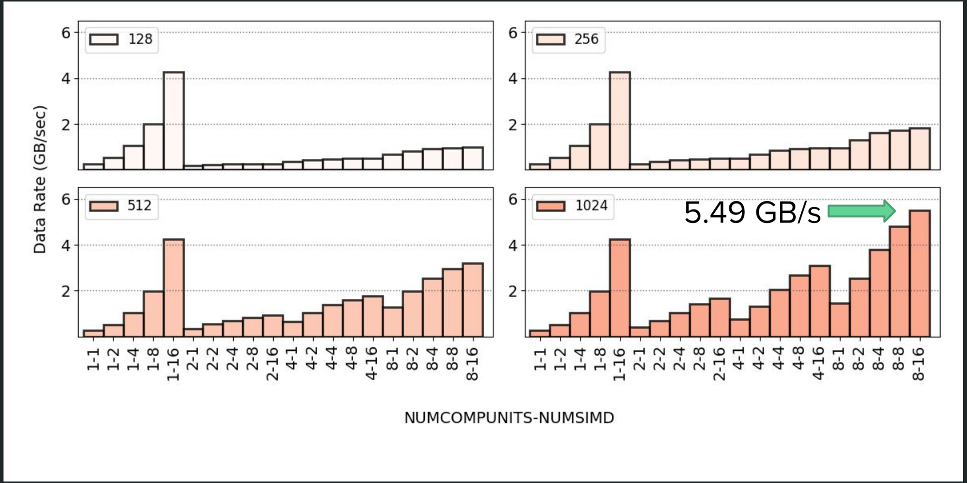
Loop termination condition

UNROLL = # of times to unroll the loop

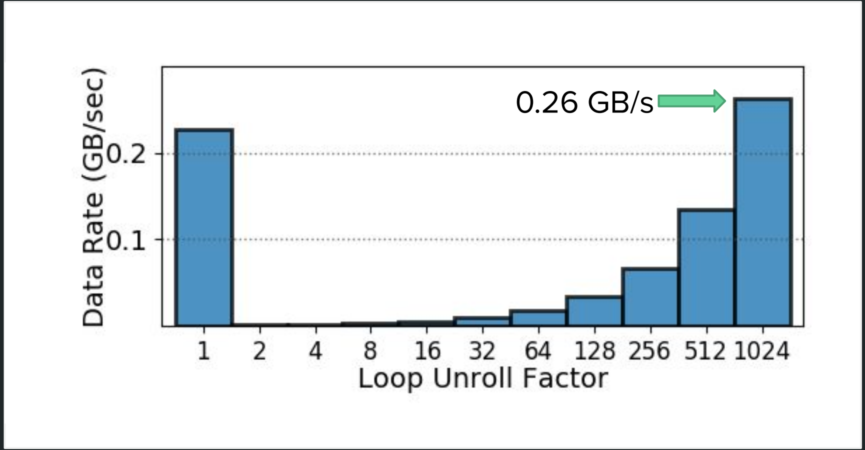
`ebcdic_txt` Deep Design Space

`UNROLL` = {1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024}

`ebcdic_txt` Width vs. Depth Results



Wide Result

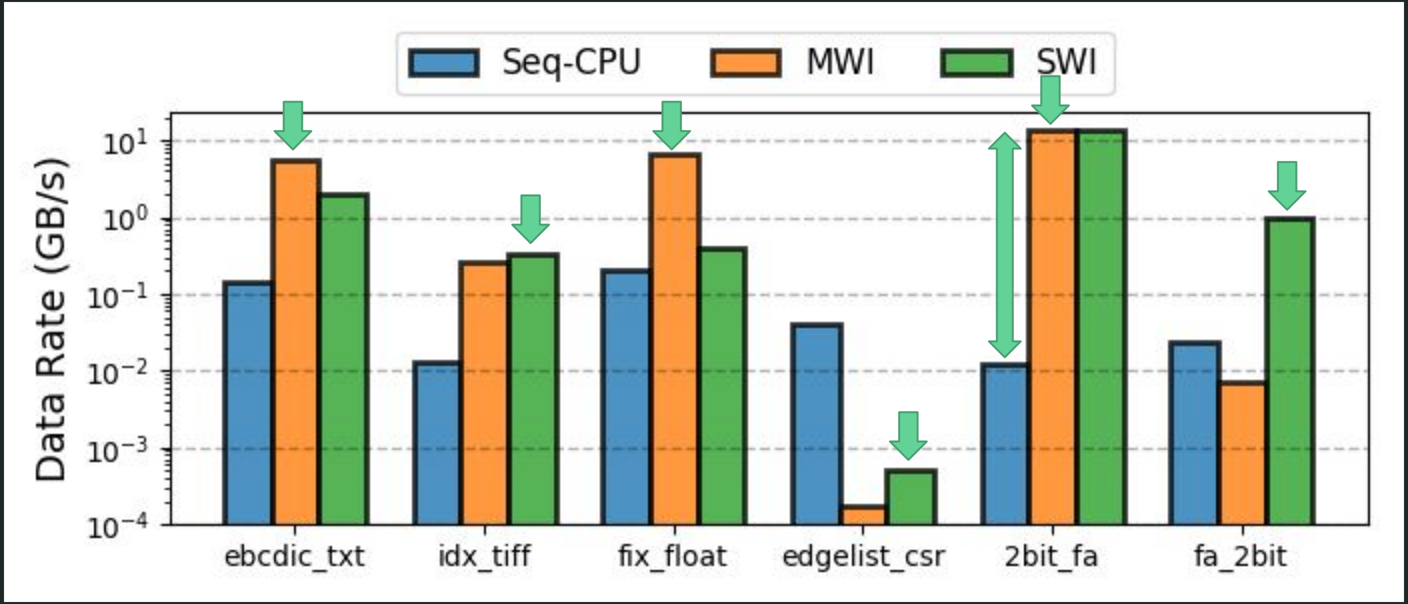


Deep Result

Prediction Results:

Application	Clustering Prediction	Correct?
`ebcdic_txt`	Wide	Yes!
`idx_tiff`	Deep	Yes!
`fix_float`	Wide	Yes!
`edgelist_csr`	Deep	Yes!
`2bit_fa`	Wide	Yes!
`fa_2bit`	Deep	Yes!

CPU vs Width (MWI) vs SWI (Depth) Results



Conclusion

We have presented our work towards designing domain specific hardware for a Post-Moore world. We do that through qualitative and quantitative domain identification, evaluating future compute technologies for domain specific computing, and architecting hardware that exploits the target domain and hardware platform.

Future Work

Intelligent Design Space Search

HLS Hardware Compiler Development

What/Where to Accelerate

A New Domain

We have presented our work towards designing domain specific hardware for a Post-Moore world. We do that through qualitative and quantitative domain identification, evaluating future compute technologies for domain specific computing, and architecting hardware that exploits the target domain and hardware platform.

Thanks!