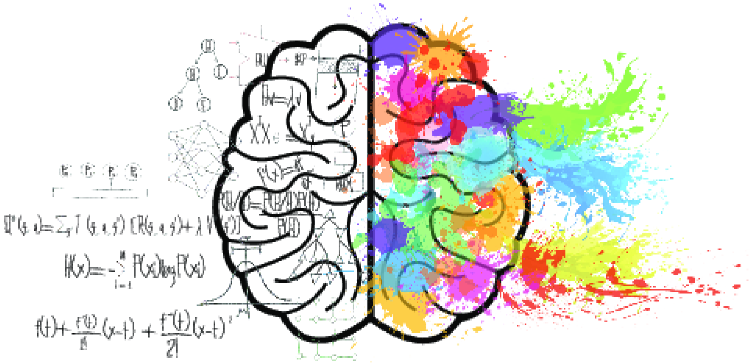
High Performance Machine Learning Workshop



Energy Efficient K-Means Clustering for an Intel® Hybrid Multi-Chip Package

Matheus Souza, Lucas Maciel, Pedro Penna, Henrique Freitas

PUC Minas

CArT

24/09/2018

Agenda

- Introduction
- Background
  - K-Means Clustering and the CPU-FPGA platform
- Proposed Implementation
- Experimental Methodology
- Results and Conclusion

PUC Minas CArT

Introduction

Introduction

- The demand for HPC
  - Massively parallel architectures
  - Large-scale multi-cores
  - Challenge: Power consumption

- Heterogeneous computing
  - GPUs and Co-processors (Xeon Phi)
  - New challenge: data movement between devices
  - Big data scenario

PUC Minas CArT

Introduction

- Field Programmable Gate Arrays (FPGAs)
- Is FPGA a good alternative for Big Data processing
  - Runs specific tasks in hardware (no general purpose units)
  - K-Means implementations [3][4]
  - Offers 10% of GPUs' power consumption [5]
  - Scales better than CPU and GPU if used properly [6]

PUC Minas CArT

Introduction

- Challenge: Data movement between host and device
  - Low bandwidth
- Multi-chip Packaging (MCP) [7]
  - Integrates heterogeneous devices or chips in a single board
- Intel launched their MCP platform
  - Xeon Broadwell CPU + Arria 10 FPGA

PUC Minas

Introduction

- Back to Big Data
  - Consider the ever growing amount of data to be processed
  - Can hybrid MCPs accelerate Machine Learning applications?
- Our goal is to propose and evaluate K-Means algorithm for the Intel Broadwell + Arria 10 platform
- Contributions
  - OpenCL-based implementation for the novel platform
  - Comparative analysis against other platforms

PUC Minas CArT

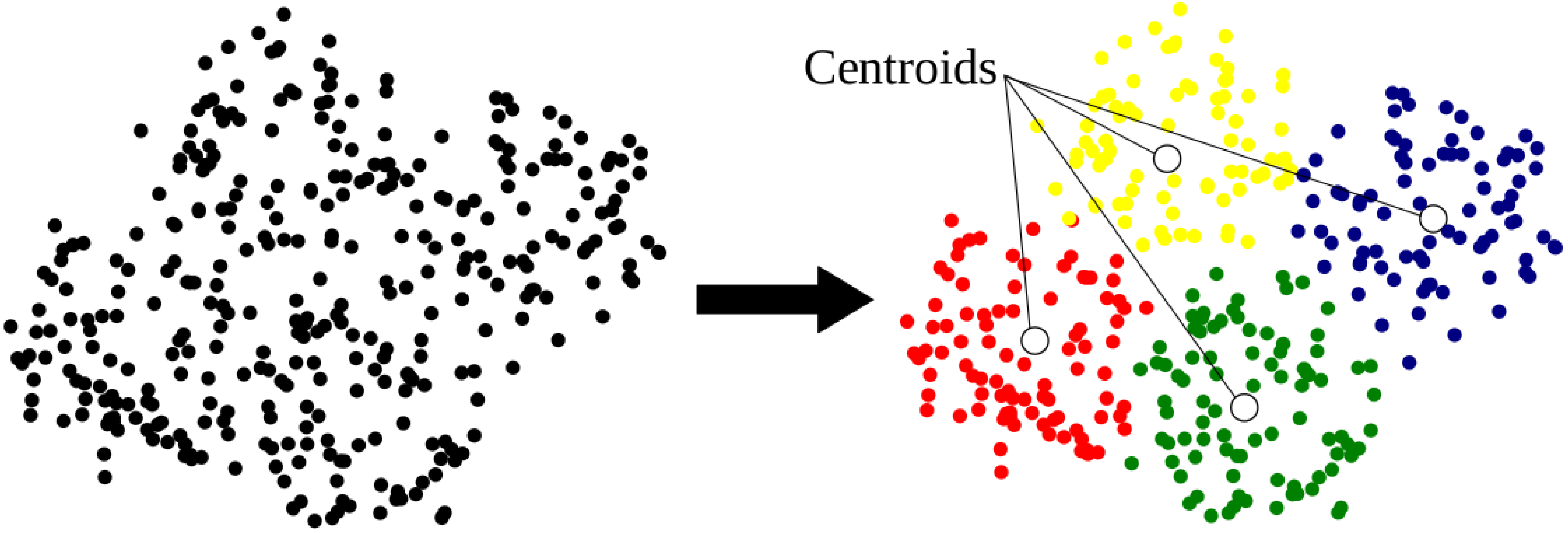
Background

Background: K-Means Clustering

- Unsupervised Machine Learning algorithm
- Partition and group data according to their features
- Euclidean distance

- Given:
  - A set of ***n*** data points
  - A real ***d***-dimensional space
  - ***k***-clusters

Background: K-Means Clustering



Background: CPU-FPGA platform

- Intel hybrid MCP
  - Host:

    14-core Xeon Broadwell @ 2.4GHz
  - Device:

    Arria 10 FPGA, GX1150
  - Interconnection:

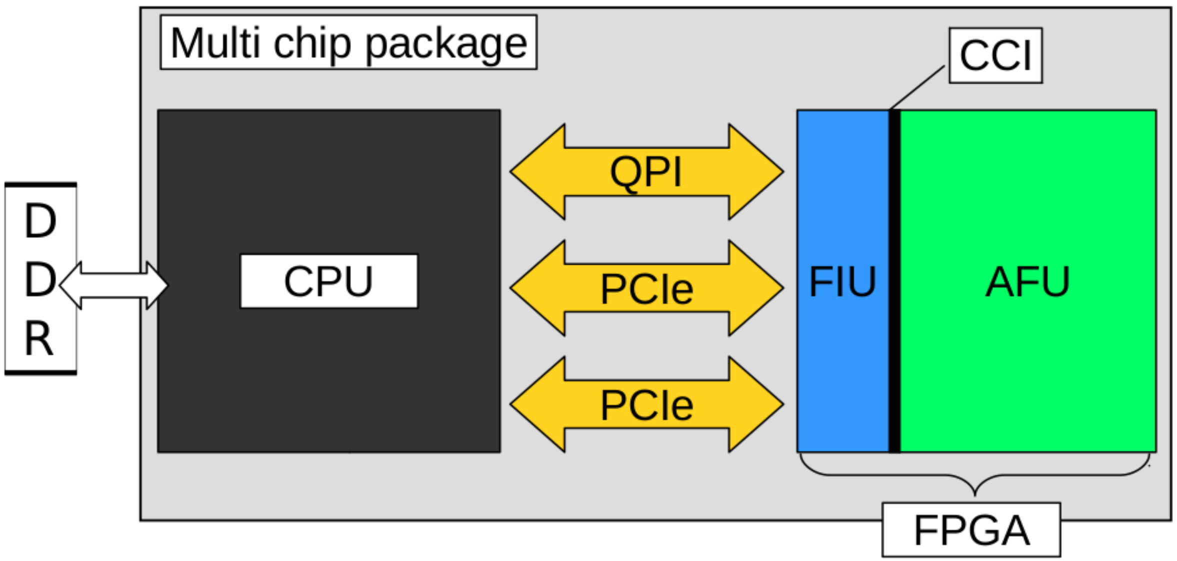
    2 PCIe (16 GB/s together)

    1 QPI (12.8 GB/s)

PUC Minas

CArT

Background: CPU-FPGA platform



- FIU: FPGA Interface Unit
- AFU: Accelerated Function Unit
- CCI: Core Cache Interface
  - Communication done in a coherent manner

PUC Minas CArT

Proposed Parallel K-Means

Proposed Parallel K-Means

- Baseline: OpenMP version from CAP Bench [17]
  - A benchmark suite for performance and energy evaluation of low-power many-core processors

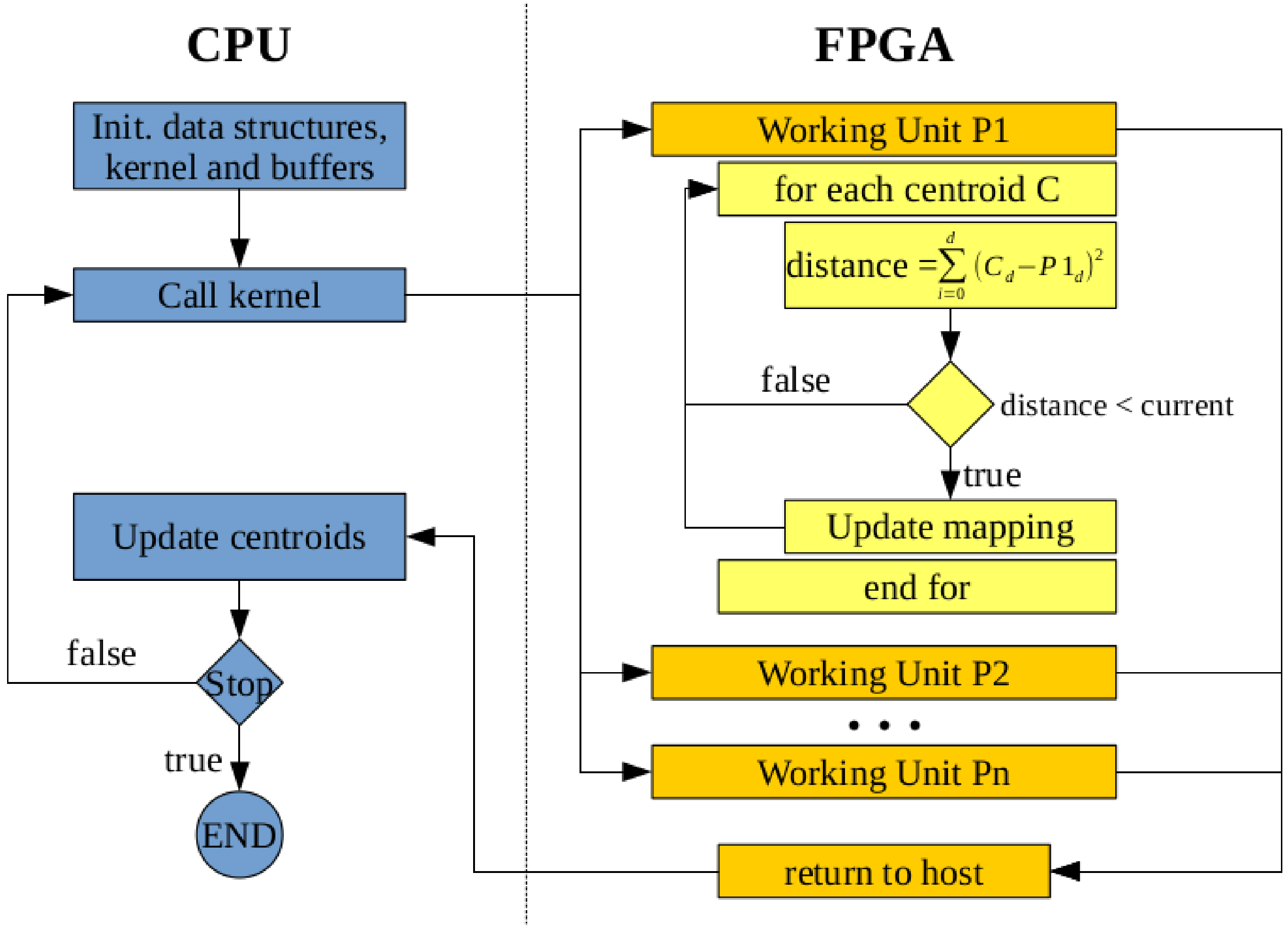
- Intel hybrid and heterogeneous MCP platform

  Host: C/C++ code to initialize the system, data structures and for synchronization

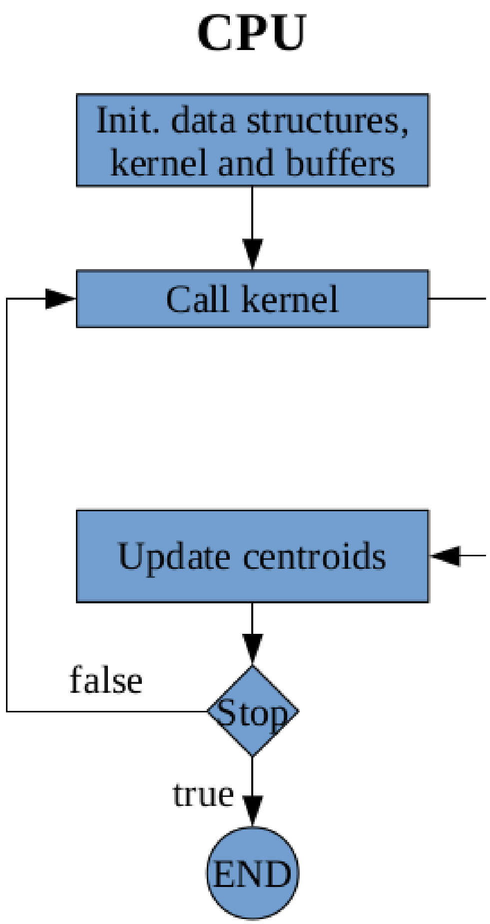
  Device: OpenCL kernel to perform the desired computation

PUC Minas CArT

Proposed Parallel K-Means



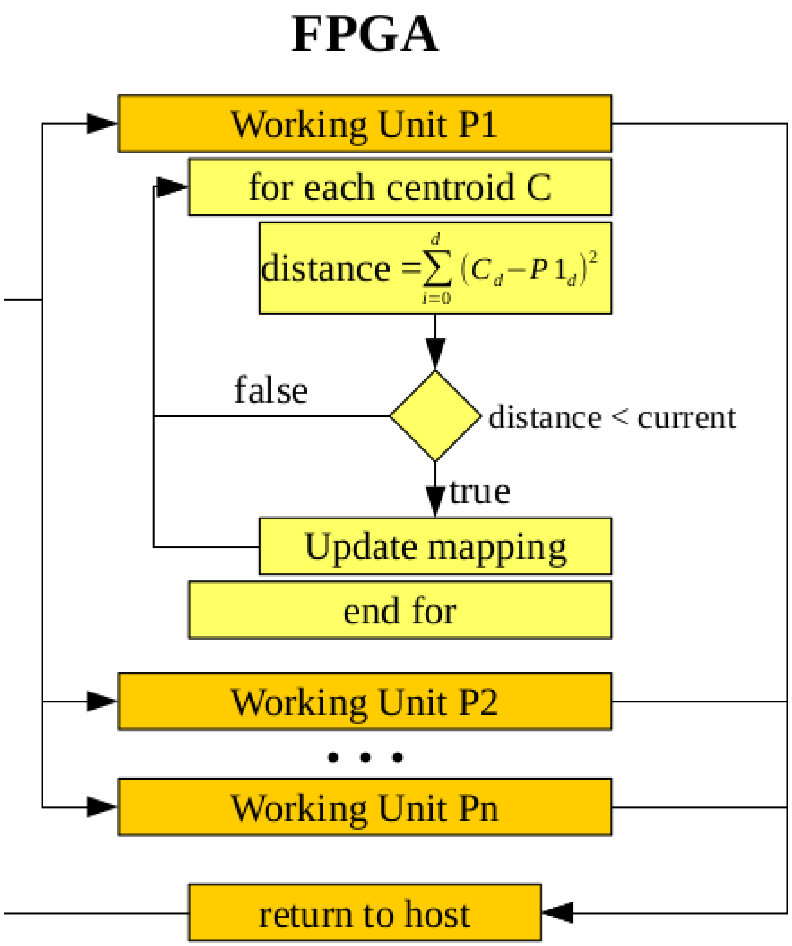
Proposed Parallel K-Means



- Update centroids at the host (C/C++)
  - Not much complex arithmetic operations
  - Conditional branches (not suitable for the FPGA)
  - Data movement (only centroids need to be offloaded)
  - OpenMP using 14-cores

PUC Minas CArT

Proposed Parallel K-Means



- Distance computing using *NDRange Kernel*
  - No gains when using *SimpleTask* pipeline
  - High data parallelism (no dependency between data points)
  - Number of *Working Units* equal to the amount of data points

PUC Minas CArT

Proposed Parallel K-Means

- Square root from Euclidean distance removed
  - Does not change the point-to-centroid mapping

- Loop unroll for features
  - Features are independent from each other at the first step of the Euclidean distance

PUC Minas CArT

Experimental Methodology

Experimental Methodology

- Compilers and tools
  - Intel SDK for OpenCL compiler (AOCL)
  - Quartus 16.0.2 (Power Play included)
  - GCC 4.9.2 (OpenMP 4.0)
  - PAPI (for measuring CPU energy consumption)

PUC Minas

Experimental Methodology

- Other platforms
  - CPU: Intel Xeon Sandy Bridge E5-2620 (12 cores @ 2.10 GHz)
  - Intel Xeon Phi Knights Corner (61 cores)
  - 8-node Raspberry Pi B2 cluster (32 cores)
  - Kalray MPPA-256 many-core (256 cores)

- 20 tests (standard error lower than 8%)

PUC Minas

Experimental Methodology

- Workload sizes

Workload	Data points	Centroids	Features	Iterations*
Tiny	4096	256	16	13
Small	8192	512	16	15
Standard	16384	1024	16	14
Large	32768	1024	16	25
Huge	65536	1024	16	48

* Measured after the tests

PUC Minas CArT

Results Evaluation

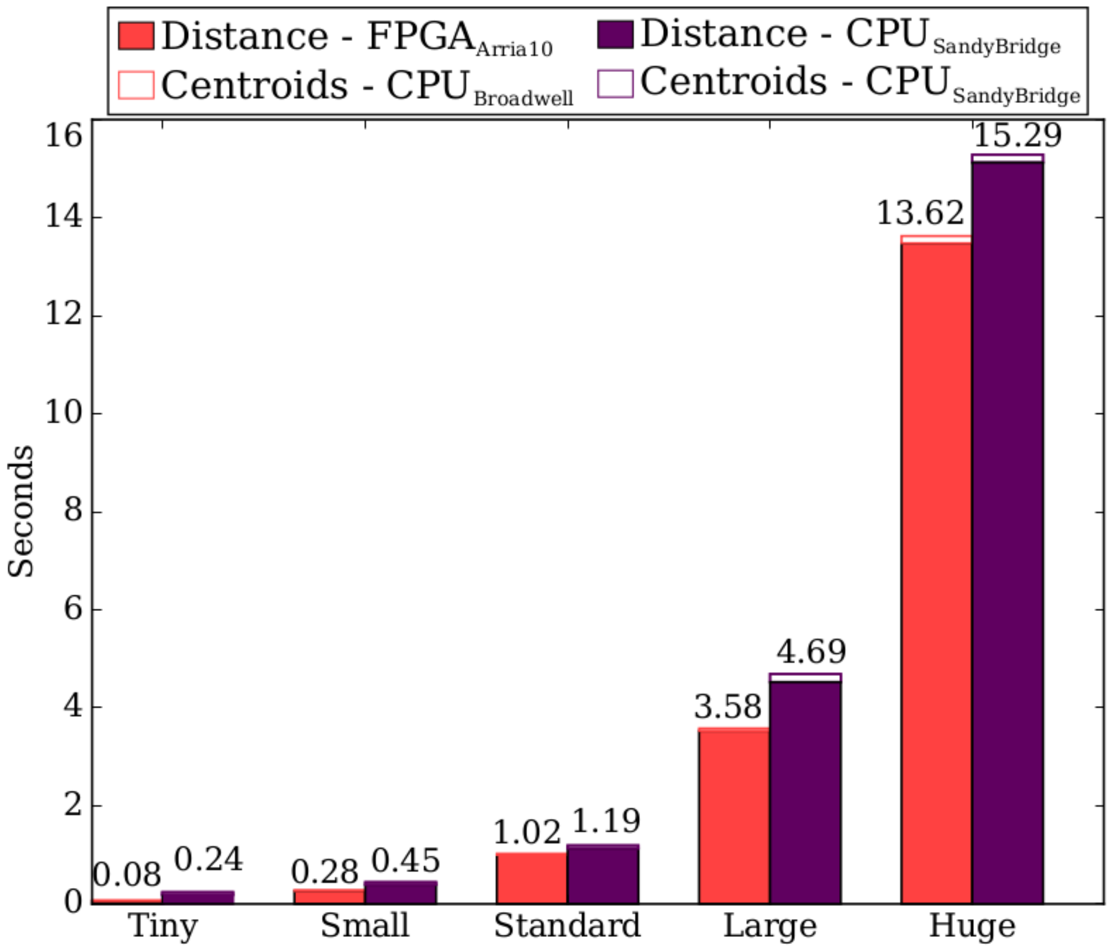
Results: Resource usage

Resource	Available	AFU
Logic utilization	1150	10% (115)
ALUTs	427200	4% (17088)
Registers	1708800	6% (102528)
Memory blocks	67244	12% (8069)
DSP blocks	1518	5% (76)

- There is still space for other kernels

PUC Minas CArT

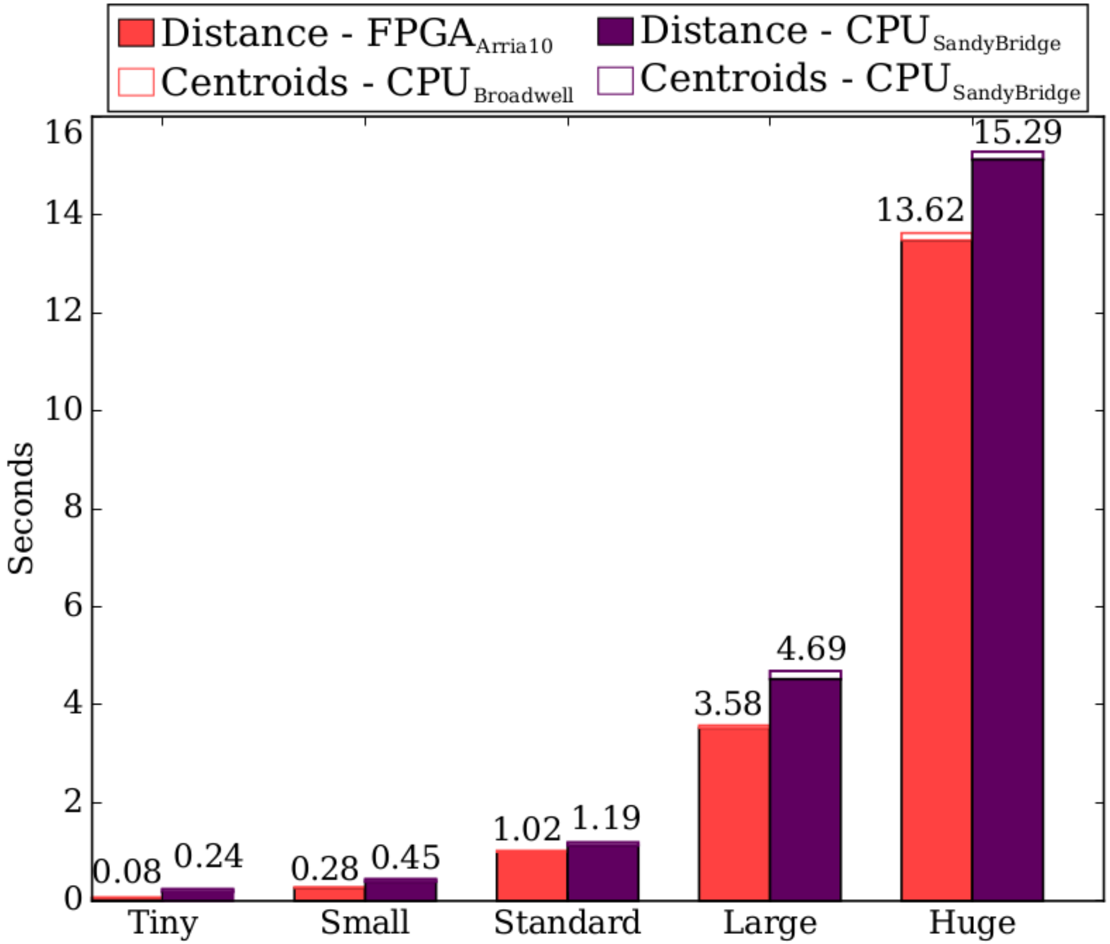
Results: Time to solution



- Centroids computing does not represent much
- Read/write to/from host/device represent, at most, 1.22% from overall time
- Time increases proportionally to input sizes and iterations
- CPU scales slightly better, however, the FPGA is faster.

PUC Minas
CArT

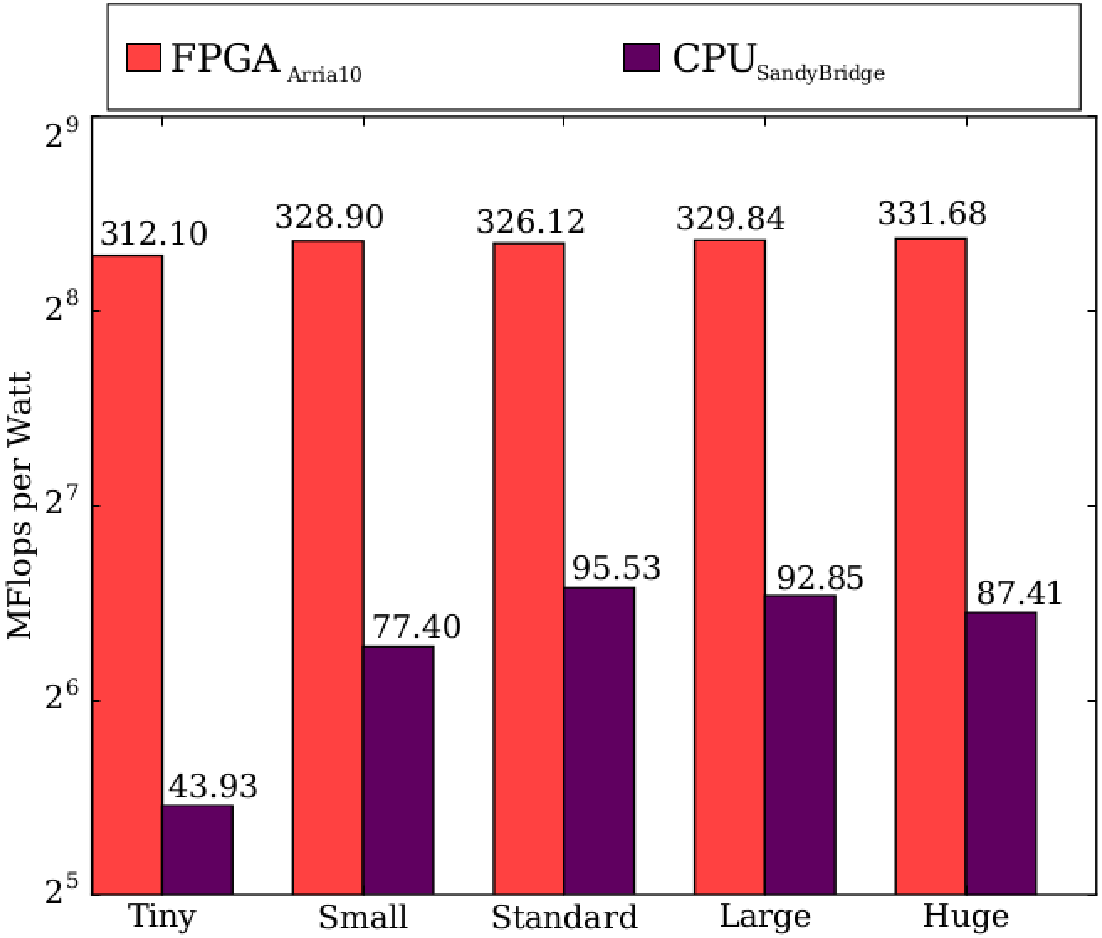
Results: Time to solution



- FPGA improvements:
  - Tiny: 68.21%
  - Small: 36.57%
  - Standard: 14.03%
  - Large: 23.59%
  - Huge: 10.82%

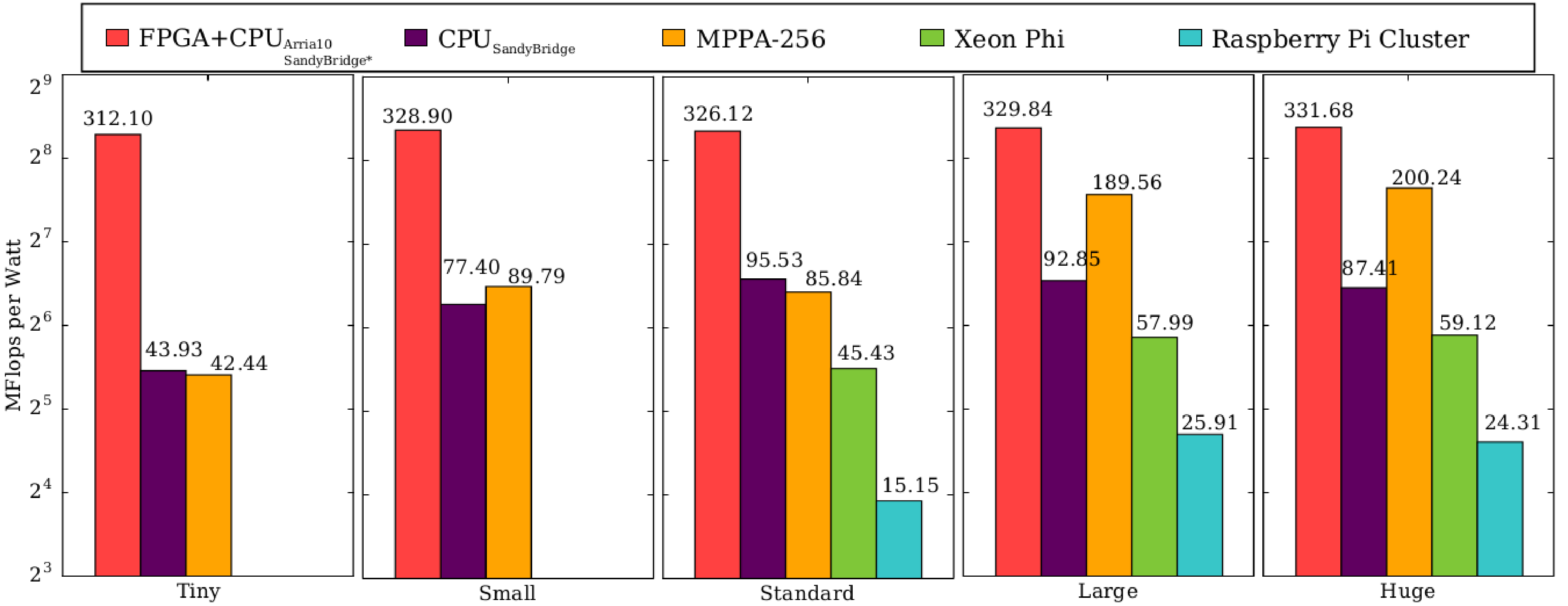
PUC Minas CArT

Results: Energy efficiency



- Tiny to Standard: CPU improves efficiency, but it starts to decrease with higher inputs
- CPU+FPGA is more energy efficient than the CPU
  - FPGA consumes less power
- Improvements:
  - Lowest: 70.71% (Standard)
  - Biggest: 85.92% (Tiny)

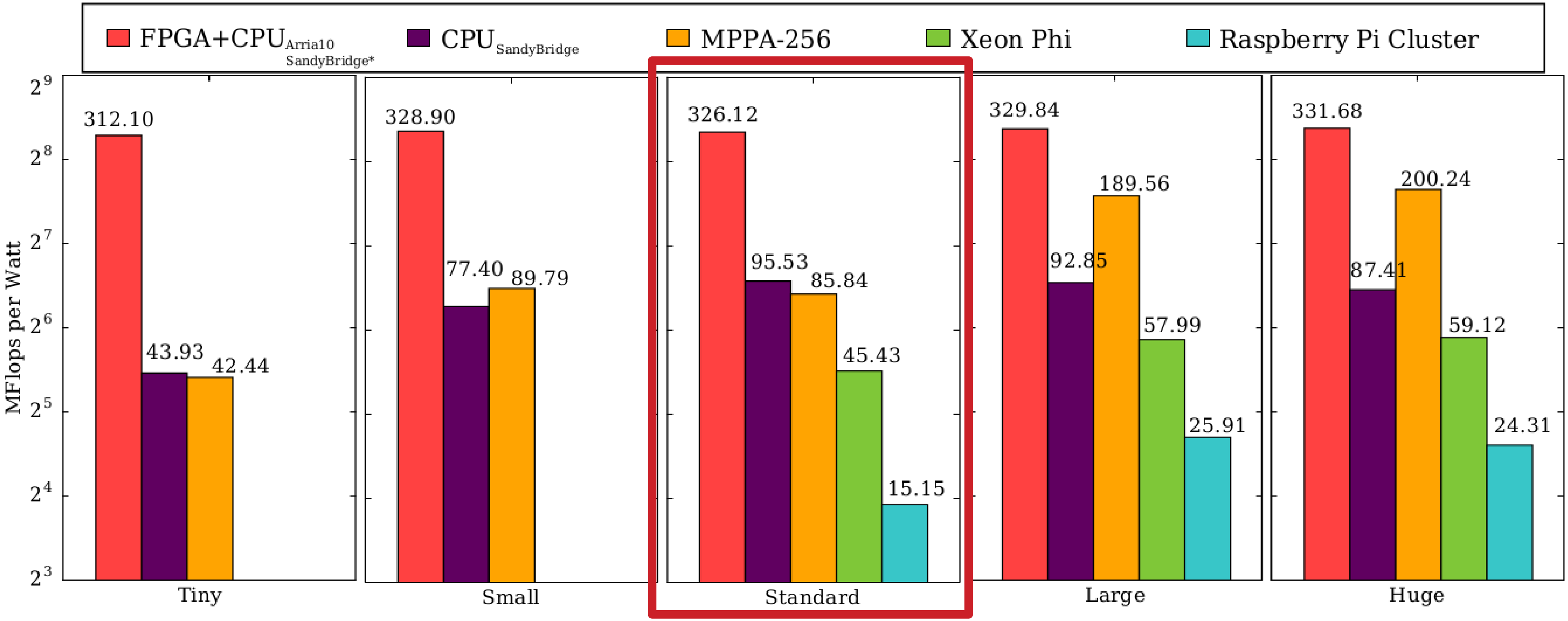
Results: Other platforms



- **Xeon Phi**: CPU+FPGA is 7.2x more energy efficient
- **Raspberry Cluster:** CPU+FPGA is 21.5x more energy efficient
- **Kalray MPPA-256:** CPU+FPGA is 3.8x more energy efficient

PUC Minas

Results: Other platforms



- **Xeon Phi**: CPU+FPGA is 7.2x more energy efficient
- **Raspberry Cluster:** CPU+FPGA is 21.5x more energy efficient
- **Kalray MPPA-256:** CPU+FPGA is 3.8x more energy efficient

PUC Minas

Conclusion

- Energy efficiency evaluation of K-Means
- Novel Intel hybrid MCP platform
  - Xeon Broadwell CPU + Arria 10 FPGA

- Up to 85.92% more energy efficient than CPU-only
- More energy efficient than other platforms

PUC Minas CArT

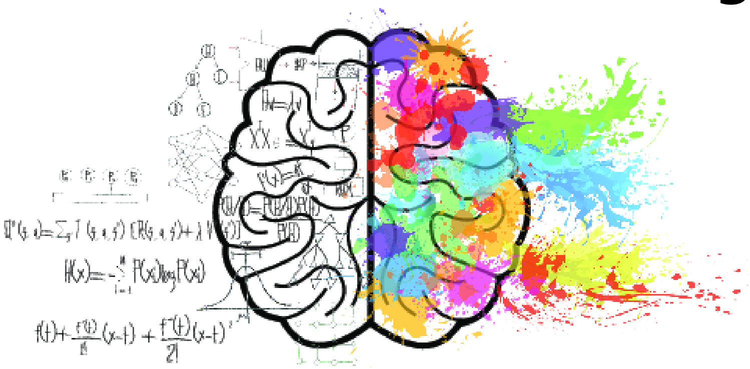
Future work

- Further K-Means evaluations
  - Real datasets
  - Other platforms (e.g. GPU)
- Other Machine Learning algorithms
  - Deep Neural Networks
  - Support Vector Machines
  - Decision Trees
- Load balance considering the system heterogeneity

PUC Minas CArT

Acknowledgement

High Performance Machine Learning Workshop



Energy Efficient K-Means Clustering for an Intel® Hybrid Multi-Chip Package

Matheus Souza, Lucas Maciel, Pedro Penna, Henrique Freitas

PUC Minas CArT

24/09/2018

Related Work

Related Work

- FPGA-based solutions for Deep Neural Networks
  - Memory restrictions: key problems

- Stream buffers to reduce data exchange [18]
- Pipeline of kernels to increase data reuse [19]
- Floating point precision reduction [20]

Related Work

- K-Means clustering
  - Hardware implementations vs. software ones

- Single FPGA: 95% less energy consumption [4]
- Single FPGA: Speedup of 10x [22]
- Multiple FPGAs: Speedup of 330x [24]

PUC Minas

CArT

References

- [3] Q. Y. Tang et al: Acceleration of k-Means Algorithm Using Altera SDK for OpenCL
- [4] L. A. Maciel et al: Projeto e Avaliação de uma Arquitetura do Algoritmo de Clusterização K-Means em VHDL e FPGA
- [5] J. Cong et al: Understanding performance differences of FPGAs and GPUs
- [6] K. O’Brien et al: Towards exascale computing with heterogeneous architectures
- [7] R. Tummala et al: Heterogeneous and homogeneous package integration technologies at device and system levels
- [17] M. A. Souza et al: CAP Bench: a benchmark suite for performance and energy evaluation of low-power many-core processors
- [18] U. Aydonat et al: An OpenCL deep learning accelerator on Arria 10
- [19] D. Wang et al: PipeCNN: An OpenCL-based FPGA accelerator for large-scale convolution neuron networks
- [20] J. Zhang et al: Improving the performance of OpenCL-based FPGA accelerator for convolutional neural network
- [22] J. Canilho et al: Multi-core for K-means clustering on FPGA
- [24] M. Vidal et al: Implementation of the K-means algorithm on heterogeneous devices: a use case based on an industrial dataset

PUC Minas

Background: K-Means Clustering

```
procedure KMEANS(k, points, d)
    centroids ← random_centroids(k, points, d)
    repeat
        map ← compute_distances(points,centroids, d)
        recalculate_centroids(points,centroids,map,d)
    until not check_if_should_stop()
    return map
end procedure
```

Results: Resource usage

Resource	Available	FIU + CCI	AFU
Logic utilization	1150	49% (563)	10% (115)
ALUTs	427200	23% (98256)	4% (17088)
Registers	1708800	27% (461376)	6% (102528)
Memory blocks	67244	23% (15466)	12% (8069)
DSP blocks	1518	12% (182)	5% (76)

- FIU + CCI consume a considerable resources
- There is still space for other kernels

PUC Minas CArT