HPES
High Performance Embedded Systems

EEE4120F

University of Cape Town • iYunivesithi yaseKapa • Universiteit van Kaapstad

High Performance Embedded Systems

Lecture 3:

Edge computing; Microprocessor vs. FPGA-based RC solutions

Lecturer:
Simon Winberg

Outline for Lecture

- Considerations for EDGE computing (connection to direction of this course)
- Computing solutions: HW vs SW vs RC
- Latest Intel chipsets
- Towards digital accelerators (e.g. Alveo)
- Tools that we will be using
- Closing Remarks & Intermission

Edge Computing

EEE4120F HPES

Edge Computing

- Edge computing (a precise definition):
  - A distributed computing paradigm that brings computation (and possibly data storage) closer to where it is needed, as a means to improve response times and save bandwidth. *
- Alternate definition:
  - "edge computing" is any type of computer program that delivers low latency nearer to the requests.
- It something of a swing away from the "in the cloud" thinking… there are problems with everything being in the cloud such as needing to get data (e.g. sensor data) to the cloud and results back from it.

* Ref and recommended read: https://www.cloudwards.net/what-is-edge-computing/

** Dilley, John, et al. "Globally distributed content delivery." *IEEE Internet Computing* 6.5 (2002): 50-58.

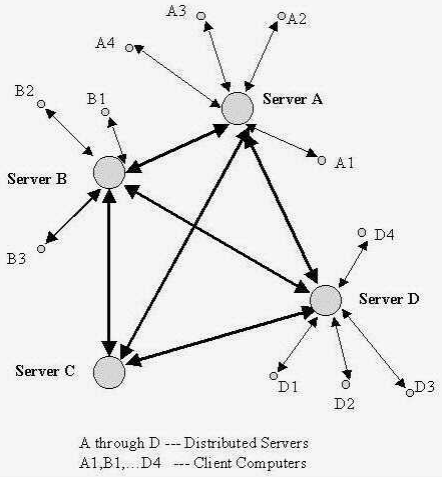
Edge Computing

"Another name for networked embedded systems?"

- A 'rediscovery' in a way that doing things at the edges (leaves) of a network may work out better than drawing it in to a central compute point
- But with the advantage of being connected to a network (e.g. bring in data from other nodes)
- The concept of a 'smart embedded sensor' or a sub-sensor net (i.e. a distribution of distributed nets) may work better in some cases.

Some benefits of edge, e.g.

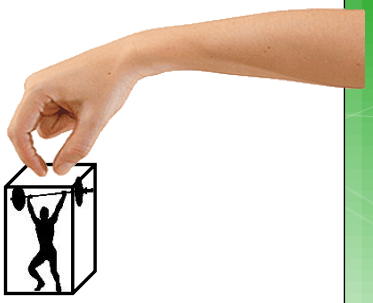
- Reduce risks (can do stuff without internet access)
- Improve response rates
- Better manage load (not having to do the processing centrally and distribute the results)
- Improved reliability



Distribution of distributed nets

HPES and Edge Computing

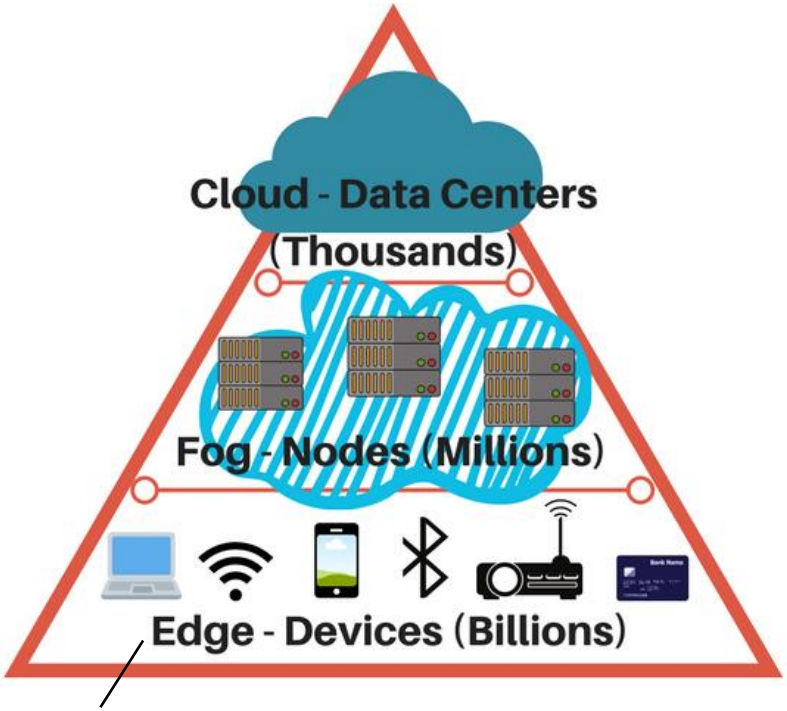
- Probably obvious to realize that many Edge Computing solutions involve HPES
- To achieve the 'smartness' of an edge computing device, you may need substantial compute power… and if you want small and low-power, then you're probably going to do it using HPES



HPES
Edge Sensor

Edge and Friends

- Fog computing:
  - Moves the computing closer to the network edge, reducing data traffic, latency etc.
  - A compromise between cloud and edge
- Cloud computing:
  - You already know…
  - It's centralizing the data and processing



e.g. proliferation of IoT devices

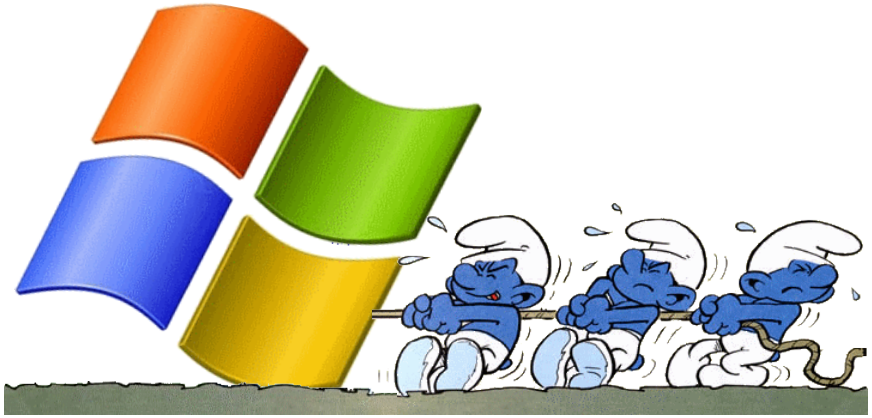
Nice to know about Edge Computing but… let's get

back to:

processing and computer hardware issues

Question:

Do you sometimes feel that despite having a wizbang multicore PC, it still just isn't keeping up well with the latest software demands?...



Major software

Processor cores

… It maybe so because:

- Your software doesn't leverage full potential of your hardware
  or maybe
- Your hardware should be leverageable to suite your software

heavily laden
single CPU

vs.

CPUs at idle

vs.

reconfigurable hardware
with specialized software

Special Hardware
vs.
Microprocessor-based
vs.
RC / FPGA-based
Solutions

Computation Methods

this term

Hardware

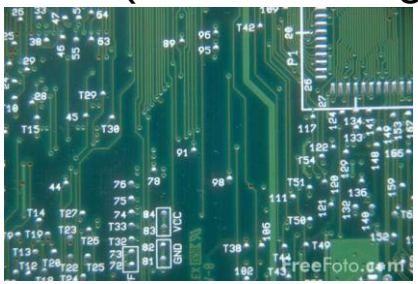
e.g. PCBs, ASICs

Advantages:

- High speed & performance
- Efficient (possibly lower power than idle processors)
- Parallelizable

Drawbacks:

- Expensive
- Static (cannot change)



Reconfigurable Computer

e.g. IBM Blade, FPGA-based computing platform

Advantages:

- Faster than software alone
- More flexible than software
- More flexible than hardware
- Parallelizable

Drawbacks:

- Expensive
- Complex (both s/w & h/w)

Software Processor

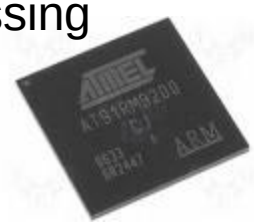
e.g. PC, embedded software on microcontroller

Advantages:

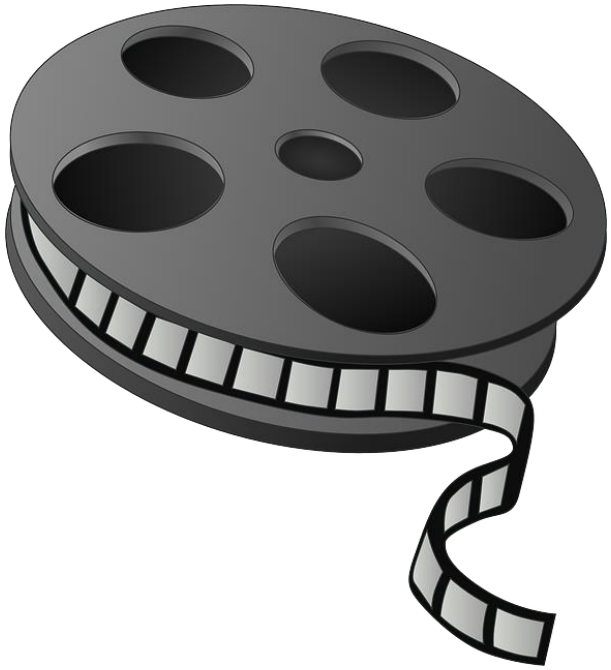
- Flexible
- Adaptable
- Can be much cheaper

Drawbacks:

- The hardware is static
- Limit of clock speed
- Sequential processing



Supplementary reading



Useful video:

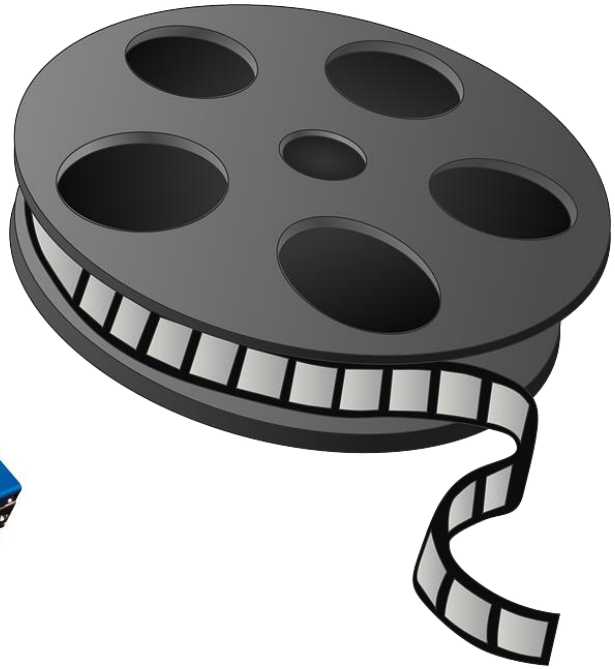
Latest Intel Chipsets

https://youtu.be/TK7eOrgOISc

"9th Gen Intel CPU Comparison — i5 vs i7 vs i9 Benchmarks"

Core i9: "Intel Core i9 Explained"
https://youtu.be/suQnh1TvGHw?list=TLPQMjIwMjIwMjAkrwJfYkr9qA

Supplementary reading



Intel's Xeon Phi

Exciting Stuff!

at a hefty price tag

which a GPU will probably beat dramatically for most things

(but some motivation for OpenCL)

https://youtu.be/I0U6ZMeVrB4?list=TLPQMjIwMjIwMjAkrwJfYkr9qA

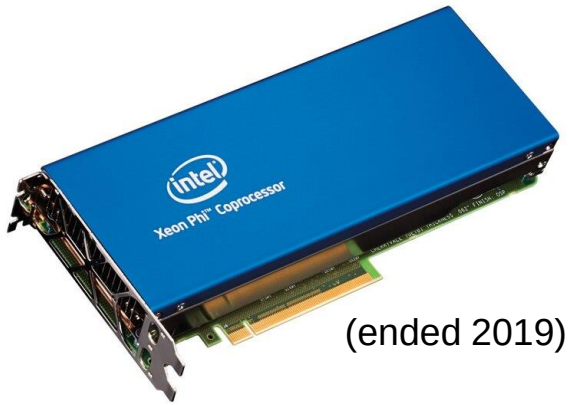
Digital Accelerators: New Practice

Digital accelerator cards (including GPGPUs) are ever increasing in popularity, including use in data centres.

(launched end 2018)

Xilinx Alevo: Adaptable Accelerator Cards for Data Centres
https://www.xilinx.com/products/boards-and-kits/alveo.html

Program with OpenCL or Xilinx's owns accelerator design suite.

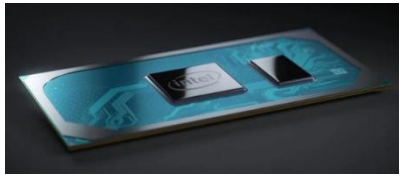


(ended 2019)

HP Intel 5110P Xeon Phi Coprocessor Kit
Intel Xeon Phi
Where and why it's no longer being made, replaced by Xeon Scalable Platform

Replaced by (from 2020)

Intel Ice Lake
(10 nm process) 10th generation core, successor for Xeon Phis.



Core i7 1068G7 (2020 debut) 4 CPU cores + 64 Iris+ GPCPU cores

And… some of this it may involve designing specialized compute architectures for the need (using a combination of languages and tools e.g. OpenCL / Verilog, C, R, etc.)

* What was it said in the "Berkeley Landscape" … "Small is beautiful" and "manycore is the future of Computing".

AMD Xilinx Alveo

Products Company

AMD XILINX

Xilinx is now part of AMD | Updated Privacy Policy

Latest awesome R.C. processing card

Adaptable Accelerator Cards for Data Center Workloads

← watch this!

YouTube https://youtu.be/IJDSclTjgBU

Feedback

Acceleration for Dynamic Workloads

Fast

Highest Performance

Adaptable

Accelerate Any Workload

- Machine learning inference to video processing to any workload using the same accelerator card
- As workload algorithms evolve, use reconfigurable hardware to adapt faster than fixed-function accelerator card product cycles

Accessible

Cloud ↔ On-Premises Mobility

ool Accelerator Card for Development Where to Buy

but now back to Earth …

Tools that we will be using

- Julia / OCTAVE ✓
- C / C++ (for host and coordination)
- OpenCL (for GP* digital accelerator)
- Verilog HDL (i.e. to design our own special-purpose processor or digital accelerator)

Hope you're suitable ready!

* GP = General Purpose (e.g. GPGPU is for running instructions on a GPU that aims to be useful for very general types of processing)

About Pracs & Prac 2

- Status of Prac1 completion?
- Prac2: OpenCL
- Recommend attempting to install nVidia or Intel or AMD OpenCL drivers on your own PC if have an OpenCL compatible device
  - For coding suggest
    Code::Blocks https://www.codeblocks.org/
    OR:
    DevC++ http://dev-cpp.com/
- Limited PC currently available in BlueLab, need to share operational machines.
  Will be provided Linux remote login.

closing
remarks & reminders…

Assigned Reading

Test Tomorrow!

Read L01b: Asanovic et al.,A view of the parallel computing landscape (recommended easier read)

Copy in resources

There will be a short quiz, and I will follow that with solutions

Read About page for Valgrind (very useful): https://valgrind.org/info/about.html

Intermission!

then…

Onwards to lecture 4

End of Lecture 3

Disclaimers and copyright/licensing details

I have tried to follow the correct practices concerning copyright and licensing of material, particularly image sources that have been used in this presentation. I have put much effort into trying to make this material open access so that it can be of benefit to others in their teaching and learning practice. Any mistakes or omissions with regards to these issues I will correct when notified. To the best of my understanding the material in these slides can be shared according to the Creative Commons “Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)” license, and that is why I selected that license to apply to this presentation (it’s not because I particulate want my slides referenced but more to acknowledge the sources and generosity of others who have provided free material such as the images I have used).

Image sources:

Wikipedia (open commons)
http://www.flickr.com
http://pixabay.com/
Forrest of trees: Wikipedia (open commons)