Evolution of Computing

Pedro Moura Trancoso
CASPER Research Group

Department of Computer Science
University of Cyprus
Computer Science & Engineering
Chalmers University of Technology

Evolution...
CPU Evolution

CPU Evolution (part 2)
Multicore - Manycore - Manytype

What to do?

- Ignore all!
- Continue to develop more complex and sophisticated architectures...
- Languages and systems that are more difficult to use...
- Crazy ideas:
  - Use GPUs to drive cars!
  - Use mobile phone CPUs to build supercomputers!

Do more using less energy and easier to use!
Possible Solutions

- Get lower - Near-Threshold Computing
- Get closer - Near-Memory Computing
- Get parallel - Lightweight Runtime
- Get tolerant - Approximate Computing
- Get smarter - Cognitive Computing

Get Parallel - Lightweight Runtime

- Dataflow... and Tasks on commodity systems
- Yet another language/system? Not really

- Research Work:
  - TFlux, TFluxSoft, TFluxHard, TFluxCell, TFluxSCC...
  - New scalable (manycore) lightweight system under review! (no impact on productivity)
Get lower - Near-Therhold Computing

- Reduce energy consumption
- Work with aggressive margins
- Logic voltage-frequency and Memory refresh rate
- Close feedback loops for error detection and correction
- May lead to errors, faults (see approximate computing)
- Project:
  - UniServer (H2020)

Get closer - Near-Memory Computing

- Improve memory bandwidth and latency
- Move computation into memory
- Challenge:
  - Computation elements - granularity, capabilities, placement, visibility, access
  - Detect regions of code that can operate in memory
- Project:
  - Memoryland (CF15), determine computation requirements and memory allocation/placement
  - Future directions: non-von Neumann in-memory processing with large number of small processing elements.
Get tolerant - Approximate Computing

- System aging, process, near-threshold operation may result in errors
- Tolerate errors at application level
- Avoid correction cost/overhead
- Projects:
  - Getting Ready for Approximate Computing: Trading Parallelism for Accuracy for DSS Workloads (CF14),
  - UniServer (H2020)

Get smarter - Cognitive Computing

- Use prediction and cognitive methods to drive all of the above!
  - Determine safe and efficient thresholds
  - Assign code for in memory execution
  - Parallelizing code
  - Estimate tolerable errors
  - Estimate error propagation and selection factor
  - ...

Tack så mycket!