

Performance and Power Impact of Issue-width in Chip-Multiprocessor Cores

Magnus Ekman Department of Computer Engineering,
Chalmers University of Technology

Per Stenstrom Department of Computer Engineering,
Chalmers University of Technology

Outline

- Problem statement
- Assumptions and studied system
- Methodology
- Results
- Conclusion

Problem

- What is the best trade-off between the number of cores and their complexity in a CMP?
- Wide design space ranging from very few very complex superscalar processors to lots of very simple single-issue cores.

Assumptions

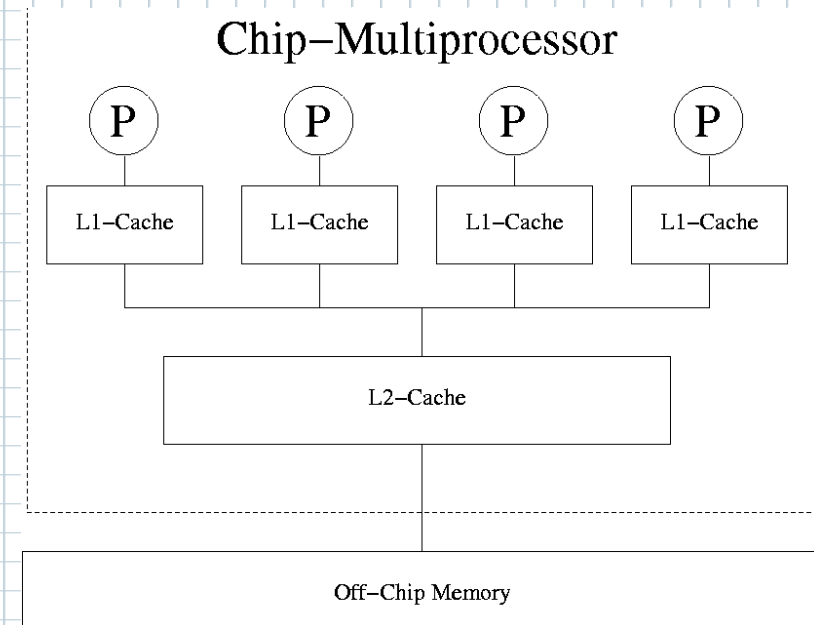
- Chip-area requirements are constant in all designs
- Clock frequency is constant in all designs
- Parallel applications

Assumptions & Disclamers

- Chip-area requirements are constant in all designs
Very rough area estimates
- Clock frequency is constant in all designs
Perhaps more realistic with faster clock for simpler designs
- Parallel applications
The world is not entirely parallel

Four basic systems studied

- 2 cores, 8-issue
- 4 cores, 4-issue
- 8 cores, dual-issue
- 16 cores, single-issue



Things that we study

- Total execution time of the same task on all systems

How do applications exploit ILP vs. TLP?

- Power consumption for the different systems

Gives hints about hot-spots in the designs

- Total energy consumption of executing the same task on all systems

How efficient is the system?

Simulation methodology (complexity effective?)

Multiprocessor version of SimWattch [1]

SimWattch is based on Simics [2] and Wattch [3] (which is based on SimpleScalar [4] and Cacti [5]).

- [1] SimWattch, 2003 IEEE International Symposium on Performance Analysis of Systems and Software
- [2] www.simics.net
- [3] ISCA 2000
- [4] www.simplescalar.org
- [5] research.compaq.com/wrl/people/jouppi/CACTI.html

How it works

- Simics generates traces dynamically
- Traces are fed into the detailed processor simulators, which tell Simics if they can handle more instructions or if they should stall.
- Activity counters are used in order to get an estimation of energy consumption

Simulation parameters all systems

- SimpleScalar pipeline
- Snoop-based MOESI protocol
- Shared bus, with contention modeled
- Shared L2-Cache: 2M, 8-way
- L1-latency: 1 cycle
- L2-latency: 12 cycles+bus-arb.
- Mem-latency: 128 cycles

Simulation parameters

8-issue core

Issue-width:	8
Window and ROB-size:	128
Load/Store-queue:	64
G-Share BP:	16K-entries
Branch Target Buffer:	4K-entries
Return Address Stack:	8 entries
L1I-Cache	64K, 2-way
L1D-Cache	64K, 4-way

Scaling methodology

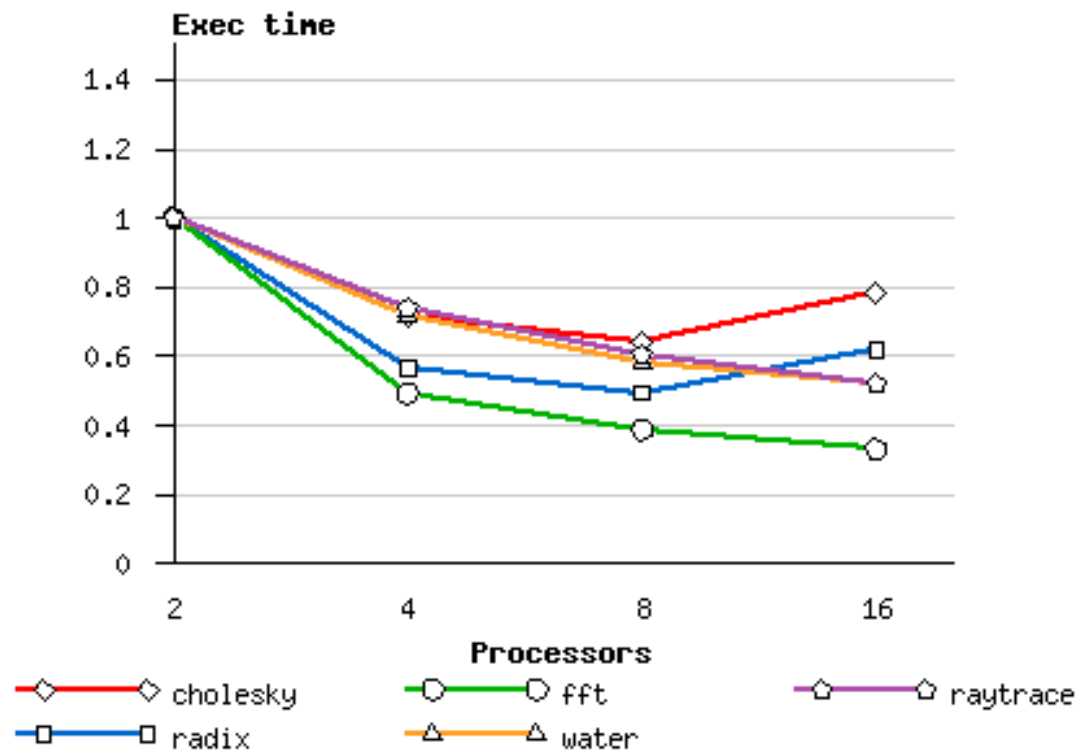
- Everything except return address stack is scaled linearly.
- Tend to favor systems with many cores.

Benchmarks

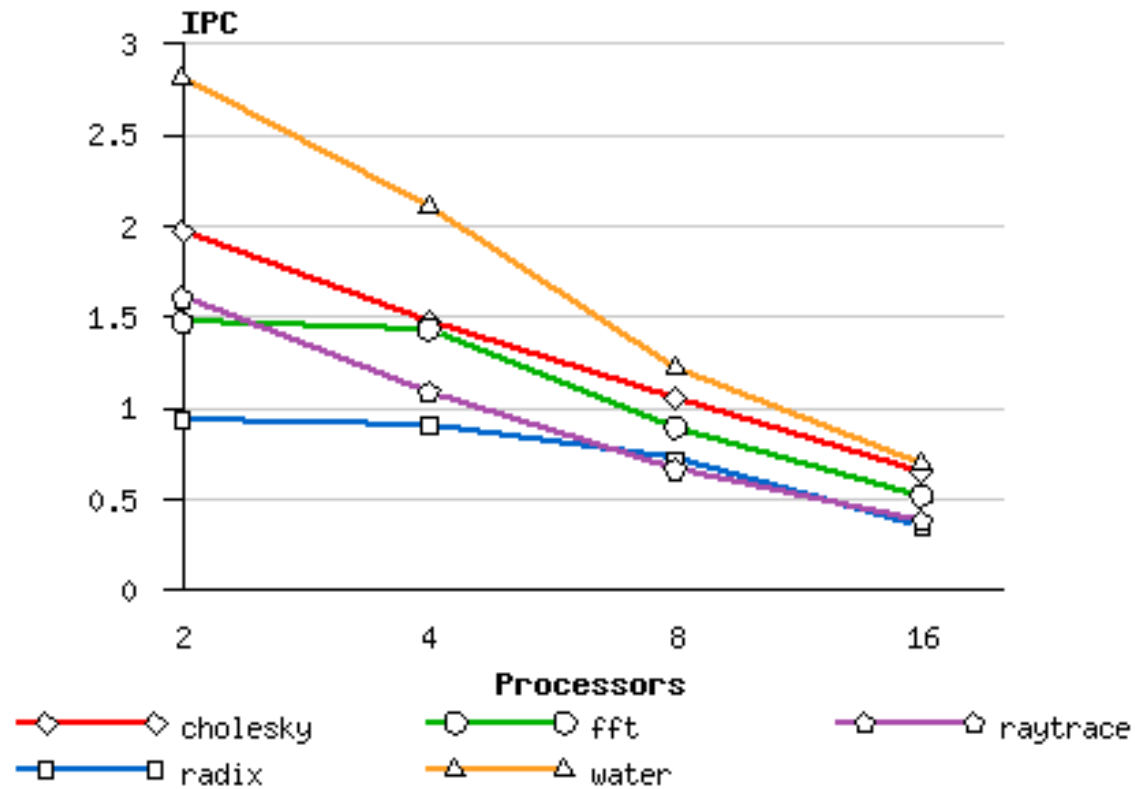
Parallel applications from Splash-2

- Cholesky
- Raytrace
- FFT
- Radix
- Water-sp

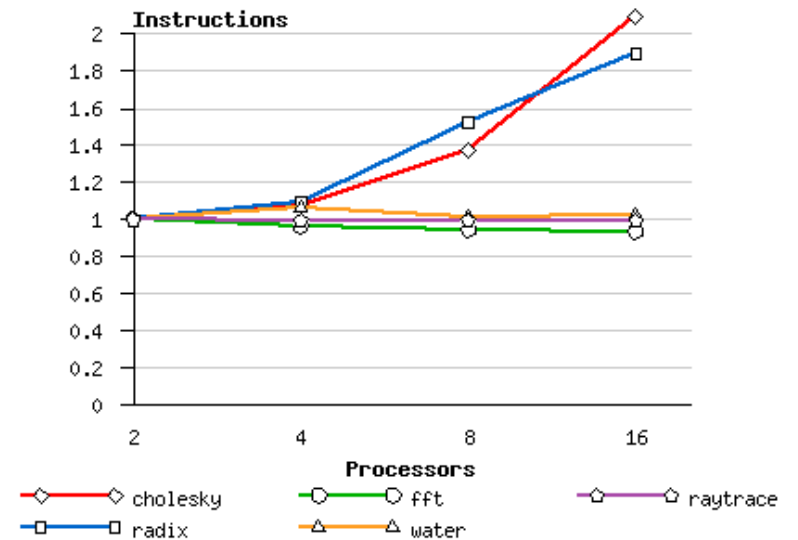
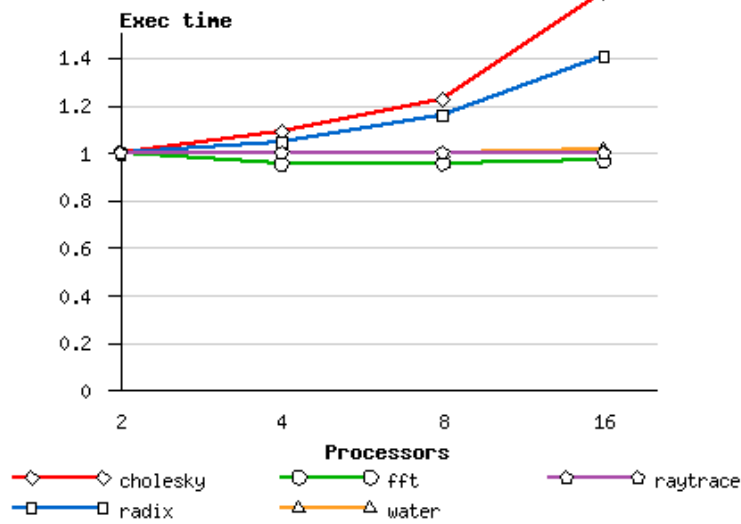
Execution time



Instructions per cycle



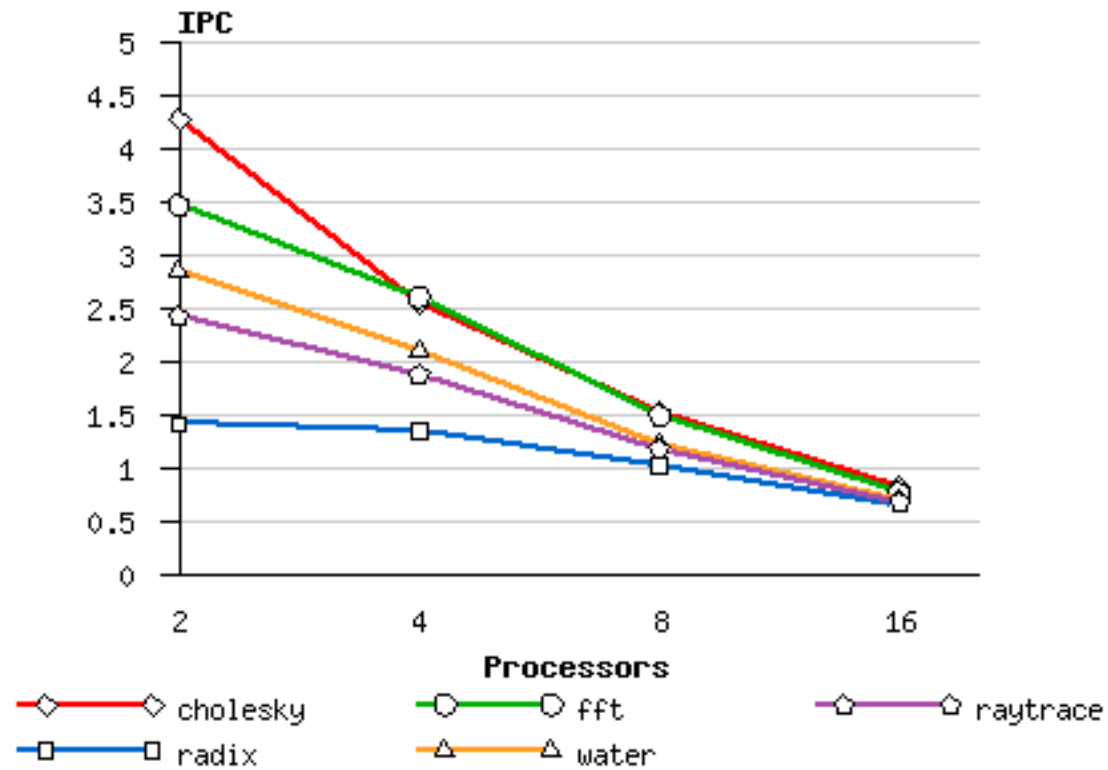
Executed instructions



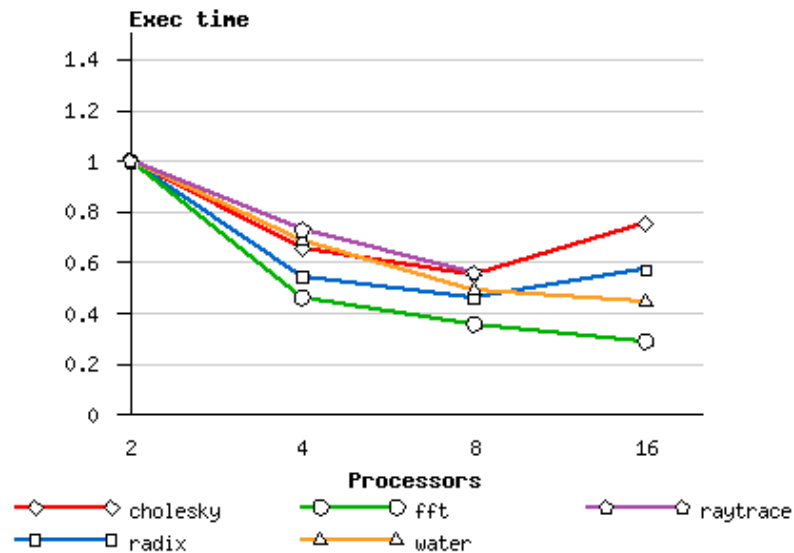
1IPC system

Baseline system

IPC with perfect memory



Execution time with longer memory latency (3x)



Increased execution time

Cholesky: 114%

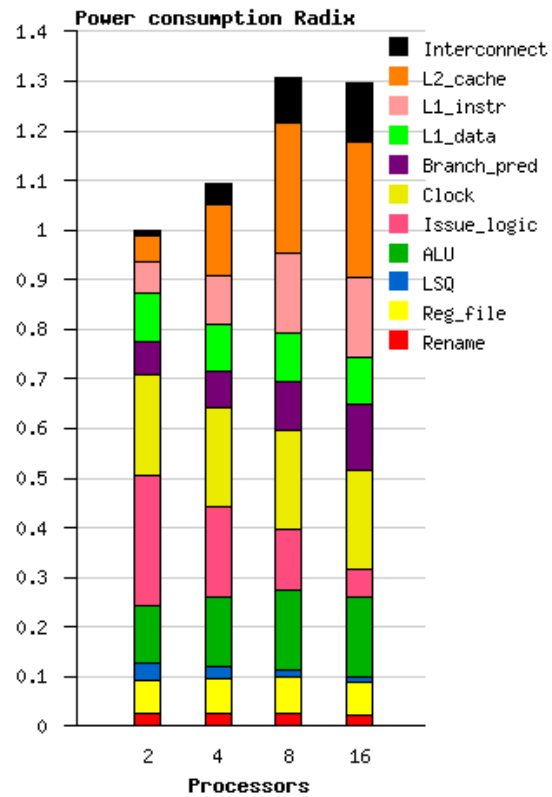
Radix: 112%

FFT: 103%

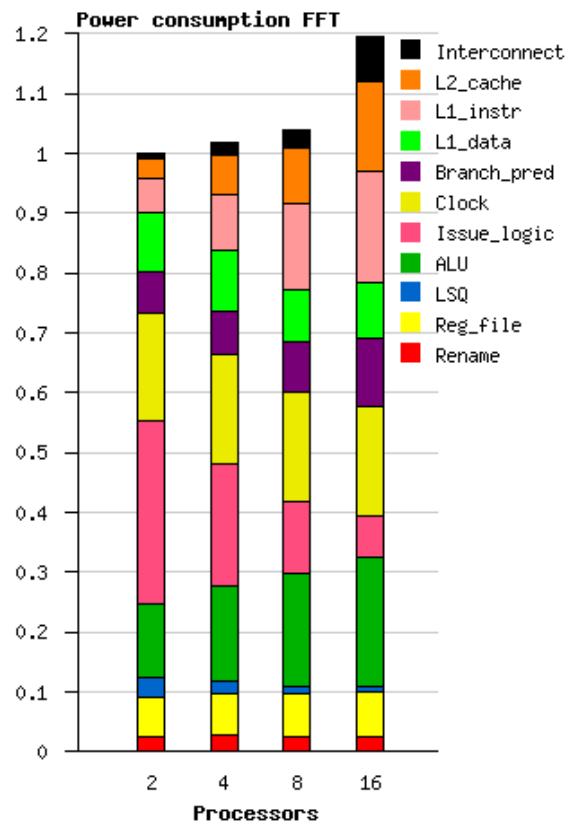
Water: 61%

Raytrace: 94%

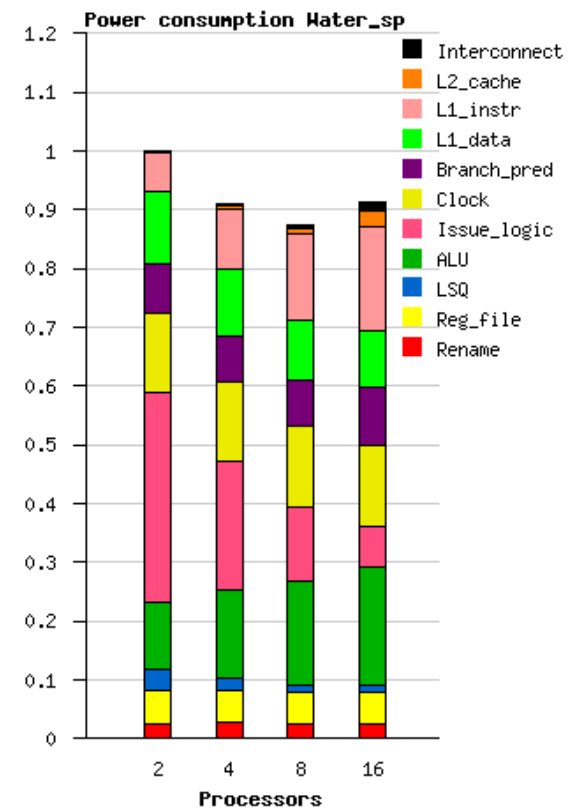
Power consumption



Radix

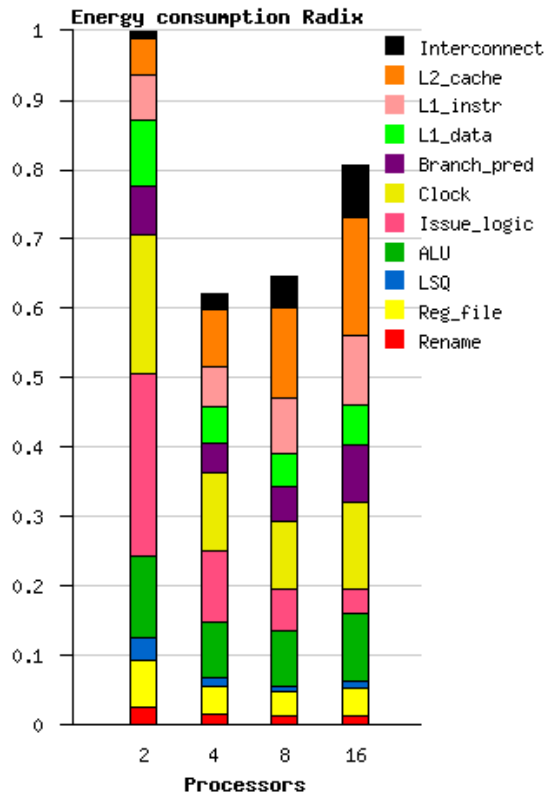


FFT

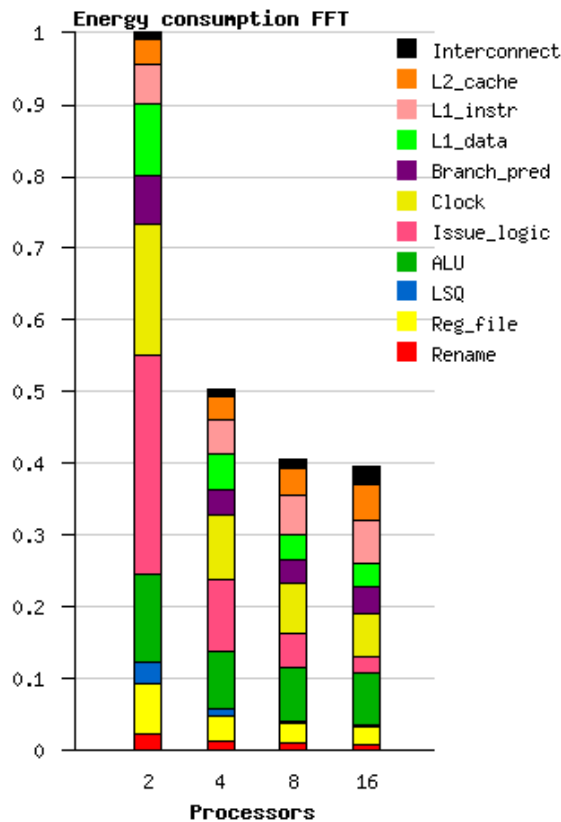


Water-sp

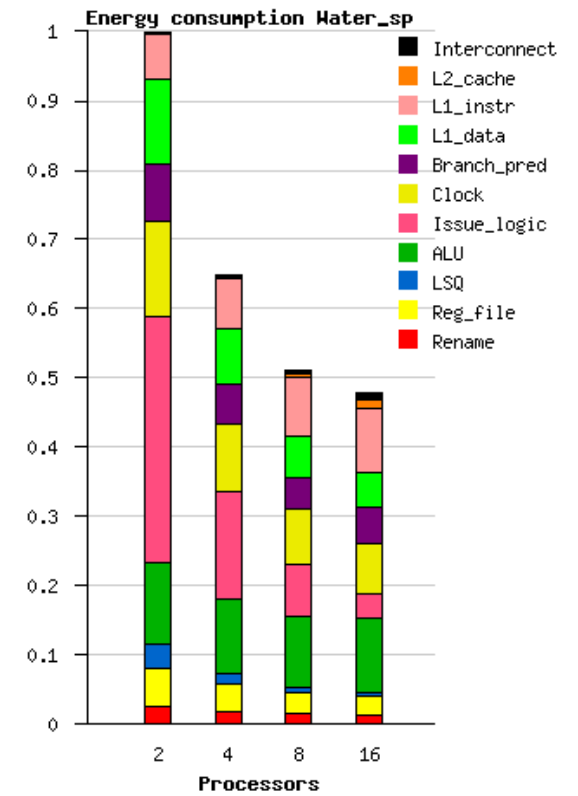
Energy consumption



Radix



FFT



Water-sp

Conclusions

- Four 4-issue cores seem to yield almost as good performance as more cores for these multi-threaded applications.
- Considering power and energy, four or eight cores seem beneficial.
- Choose four cores in order to achieve good single-thread performance!