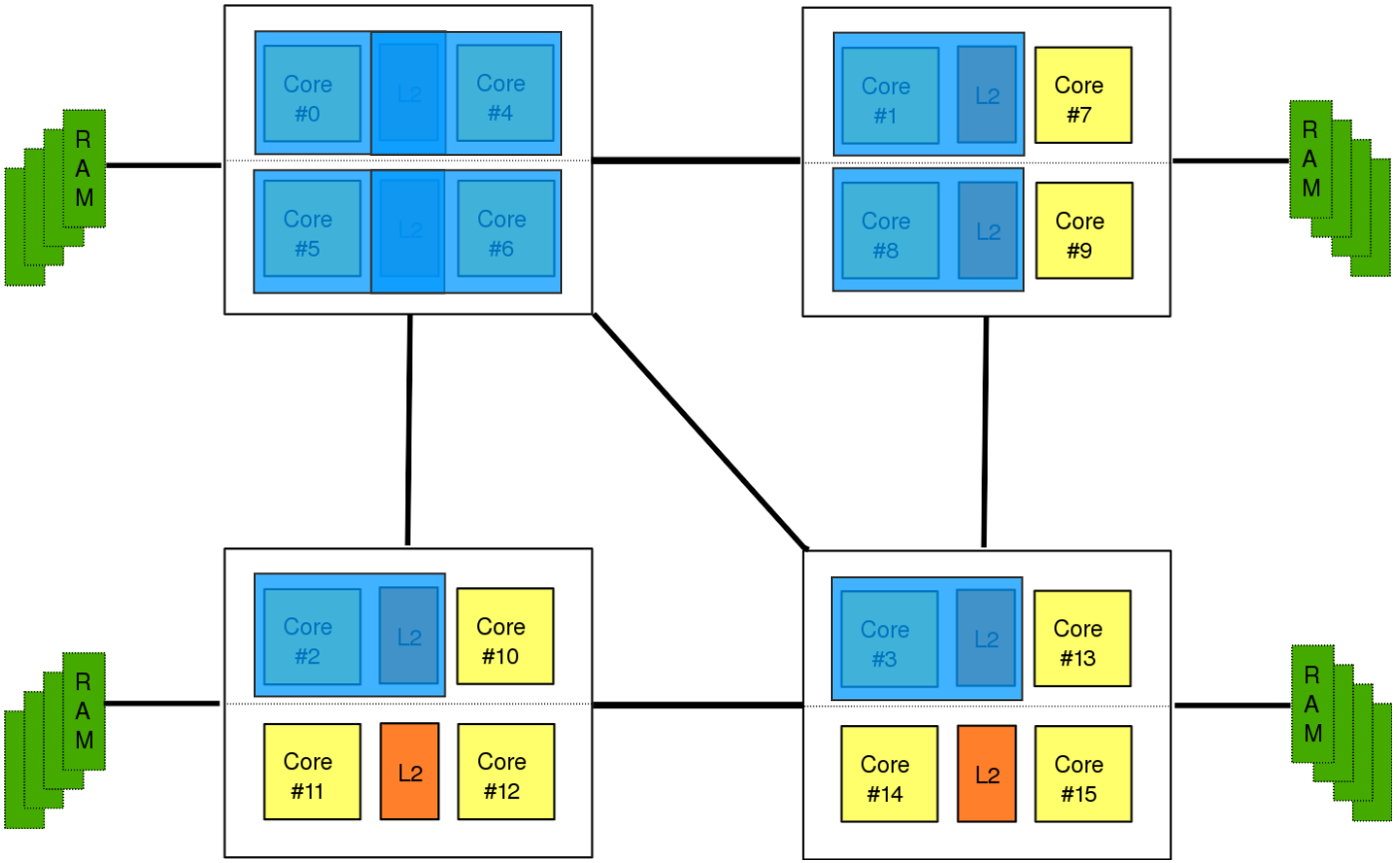
On the Importance of Thread Placement on Multicore Architectures

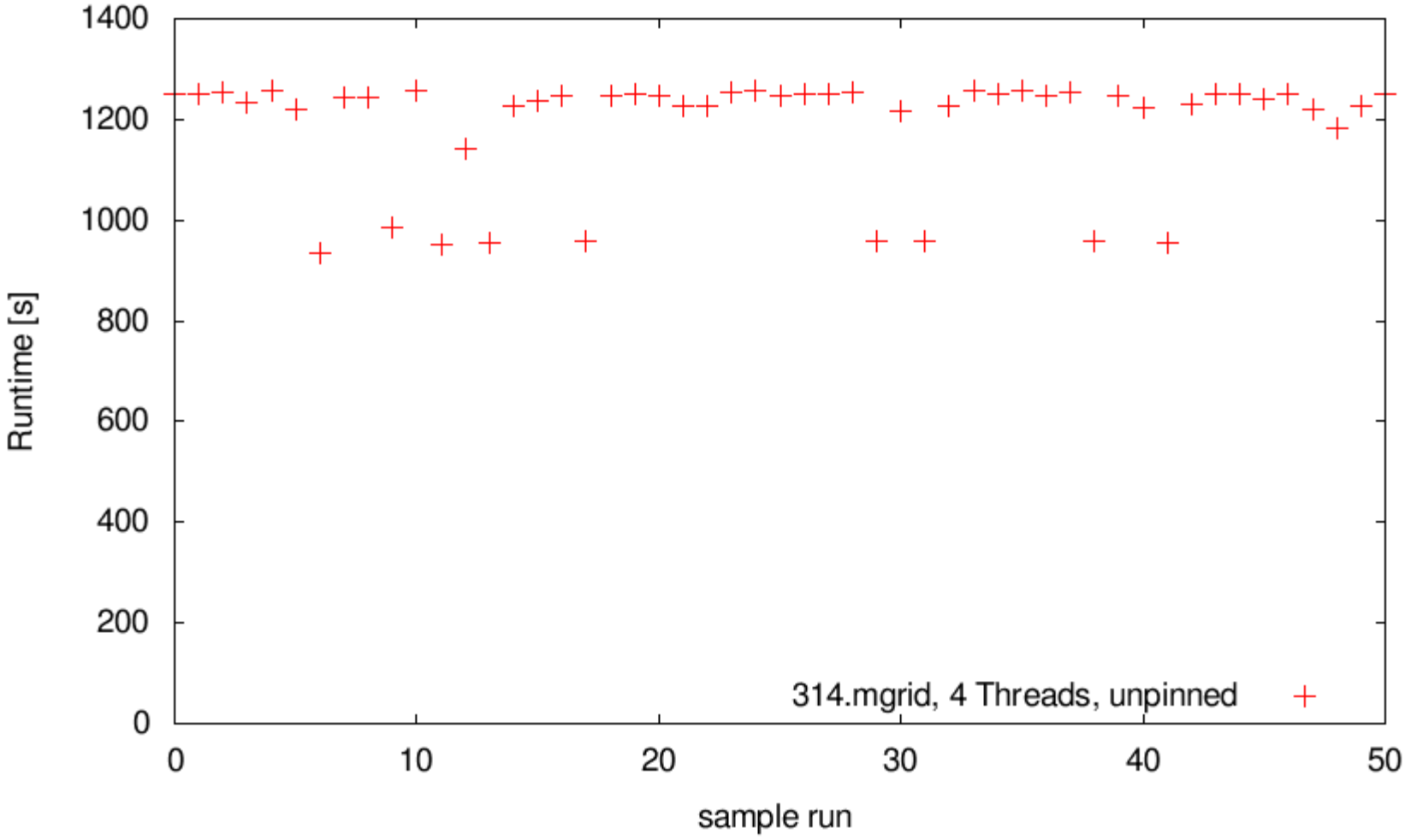
HPCLatAm 2011
Keynote
Cordoba, Argentina
August 31, 2011

Tobias Klug

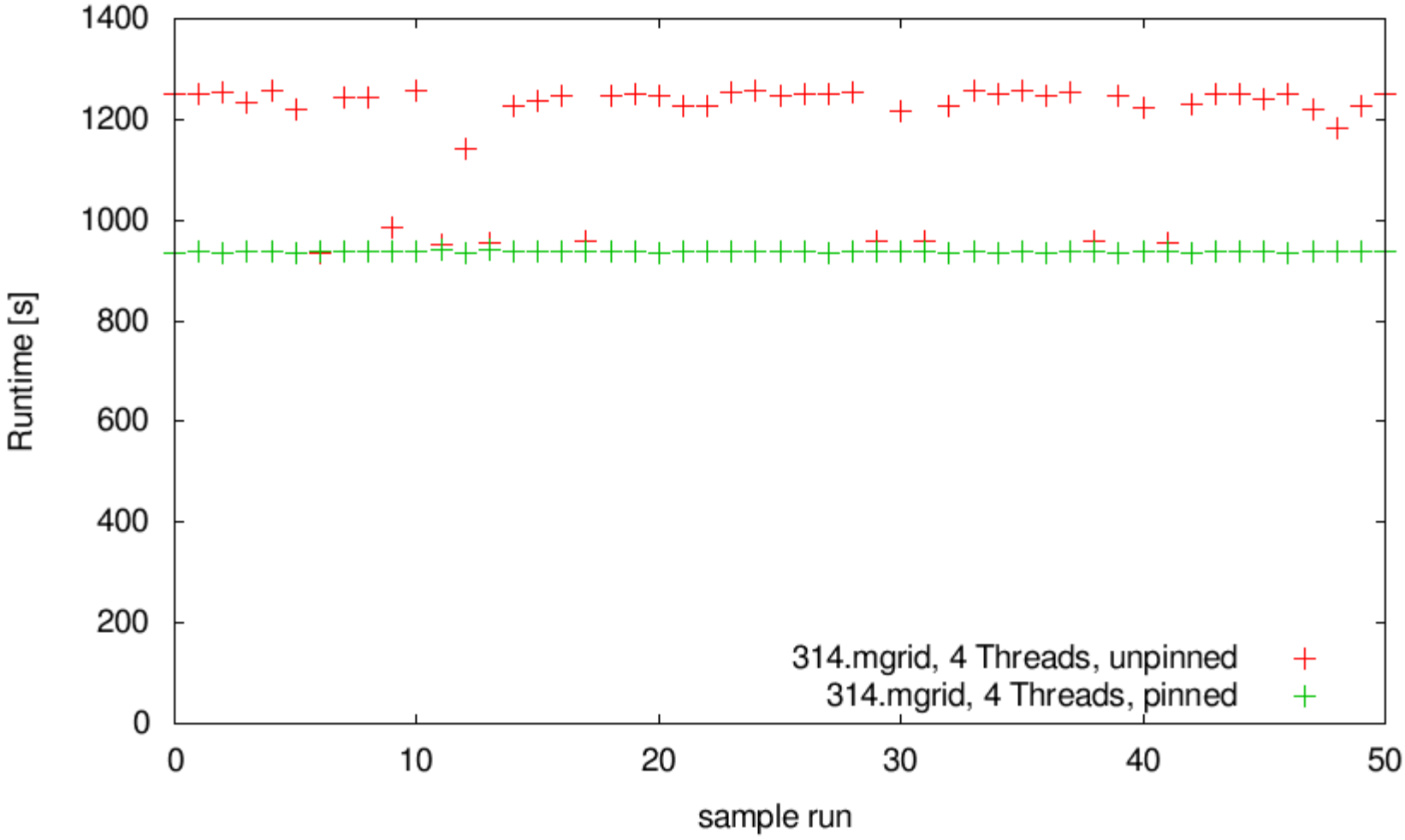
Motivation: Many possibilities…



… can lead to non-deterministic runtimes...



... but don‘t have to



The `autopin` Approach

- User-level tool
- Start multi-threaded application under `autopin` control
- User can specify pinnings of interest
- Pin threads to cores
- Assess performance of chosen pinning using performance counters
- Try alternative pinnings until optimal pinning is found

Performance Counters

- Multiple Event Sensors
  - ALU Utilization
  - Branch Prediction
  - Cache Events (L1/L2/TLB)
  - Bus Utilization
- Two Uses:
  - Read: Get Precise Count of Events in Code Regions => Counting
  - Interrupt on Overflow => Statistical Sampling
- Well-known tools:
  - Oprofile
  - Perfctr
  - Intel Vtune
  - Perfmon2

perfmon2

- Kernel-Patch + library (libpfm)
- Generic interface for PMU access
- Portable: implementations for IA32, x64, IA64, MIPS, Power
- Allows for per-thread and system-wide monitoring
- Support for counting and sampling

- pfmon:
  - attach to running threads
  - fork new processes and attach to them
  - fully exploit performance counters

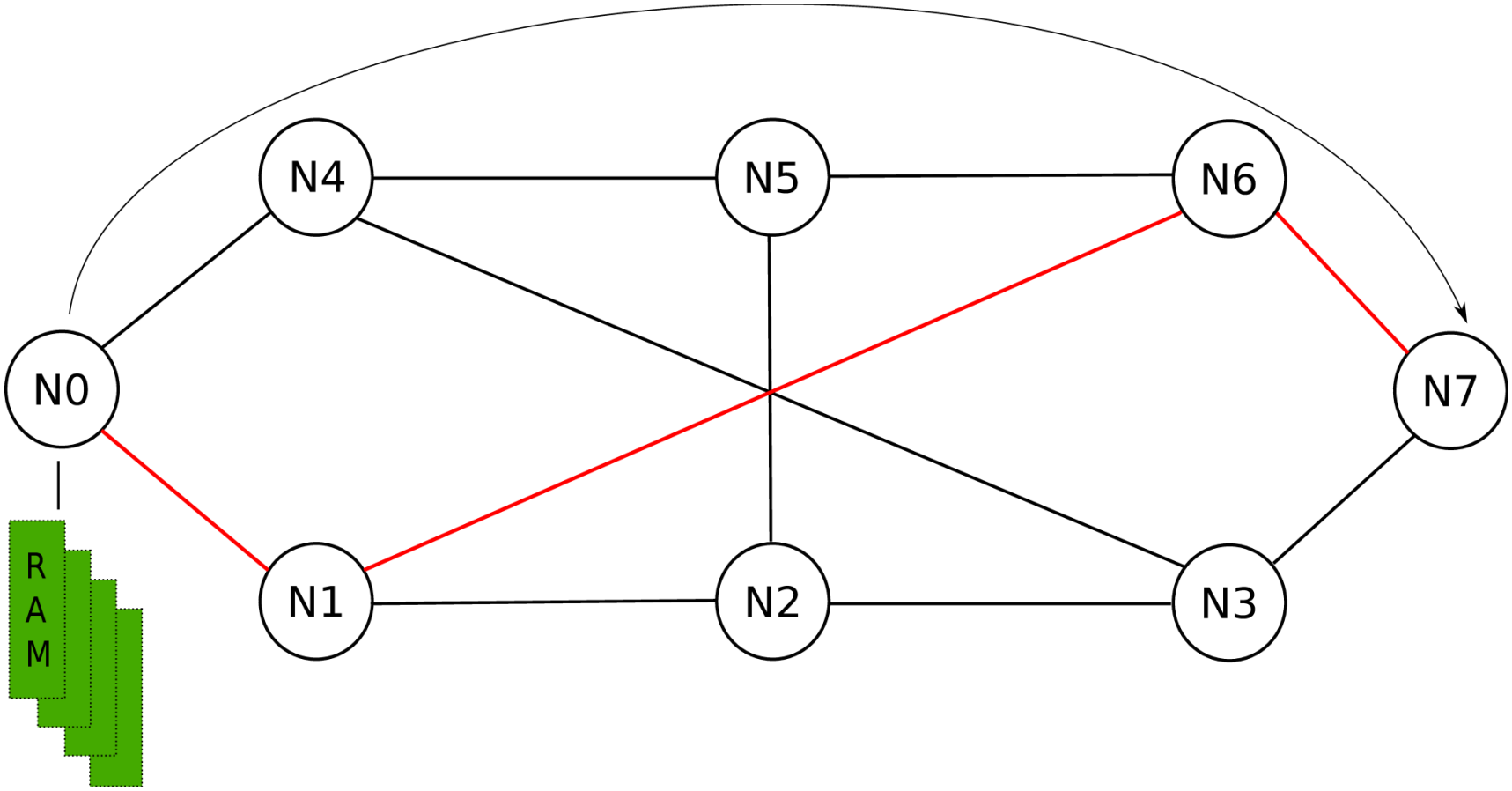
Algorithm

```
init_autopin (pinningList, initTime, program);

for (i=0; i<numOfPinnings; i++)
{
   pinThreads(pinningList[i]);
   runThreads(warmupTime);
   p1 = readPerformanceCounters();
   runThreads(sampleTime);
   p2 = readPerformanceCounters();
   performanceRate[i] = (p2-p1)/sampleTime;
}

pinThreads(bestPinning);
```

NUMA automatic page migration



Automatic Page Migration

- Kernel patch from Lee Schermerhorn
- Thread moves to new NUMA node:
  remove PTE references of “old” NUMA node
  pages are now unmapped
- Next access to page causes page-fault
- Modified kernel routines pull page local
  (migrate on fault)
- Update PTE
- Controlled via `cpusets`

Experimental Setup

- Caneland:
  - Intel Tigertown: Quad-Core, 2x4MB L2/socket, 2.93GHz clock rate
  - 4-way, 4x1066MHz FSB, 64MB snoop filter, UMA
- Clovertown:
  - Intel Clovertown: Quad-Core, 2x4MB L2, 2.66GHz clock rate
  - 2-way, 2x1333MHz FSB, UMA
- Barcelona:
  - AMD K10: Quad-Core, 4x512kB L2, 1x2MB L3, 1.9GHz clock rate
  - 2-way, 1000MHz Hypertransport, NUMA

- Linux Kernel 2.6.23 with perfmon2 patches
- Intel Compiler Suite

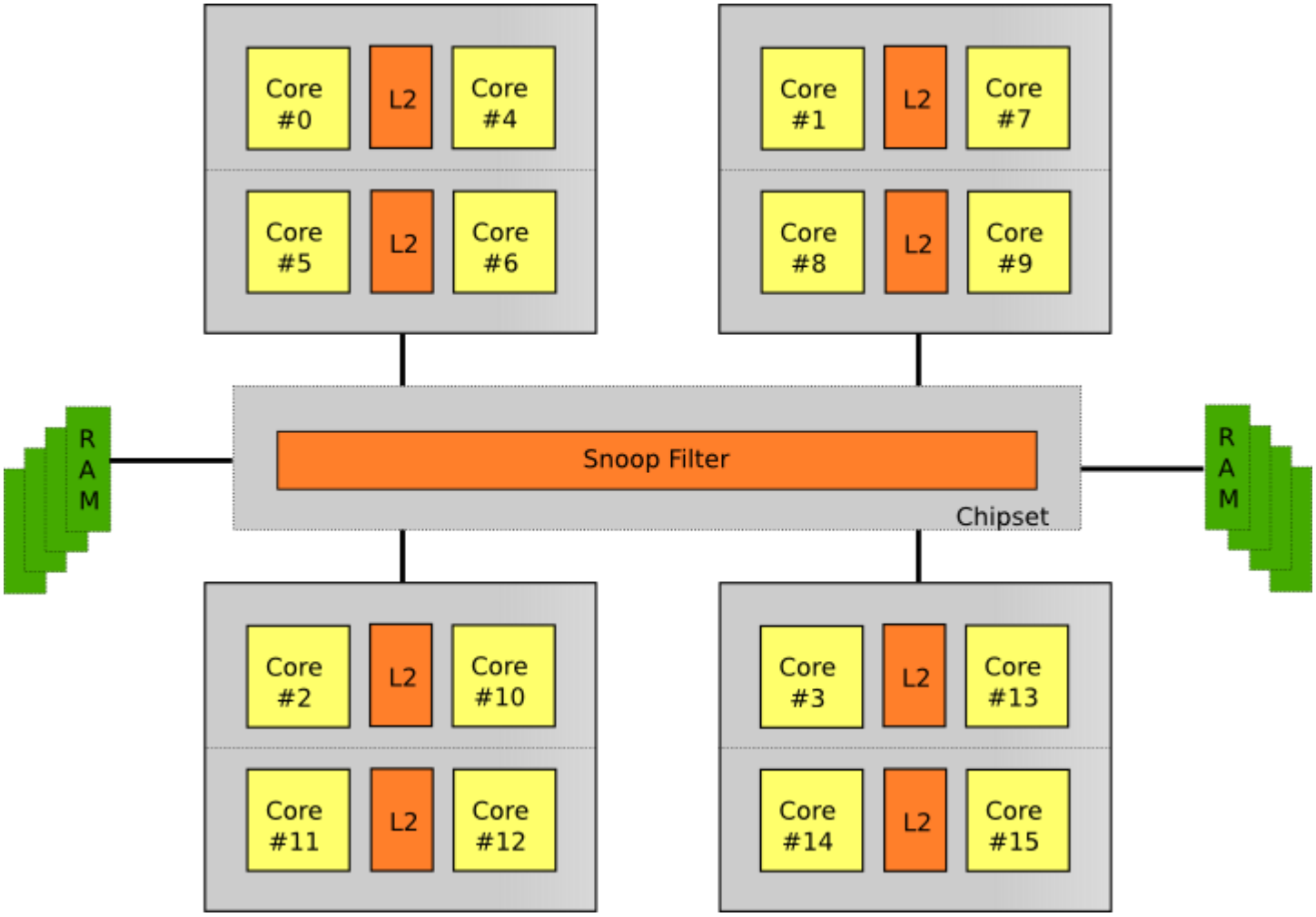
Evaluation

- SPEC OMP
  - Benchmark consists of real scientific applications
  - OpenMP
  - PC: INSTRUCTIONS_RETIRED
  - Several Multicore-Architekturen examined
- Memory Throughput
- MPI
- Electric power consumption

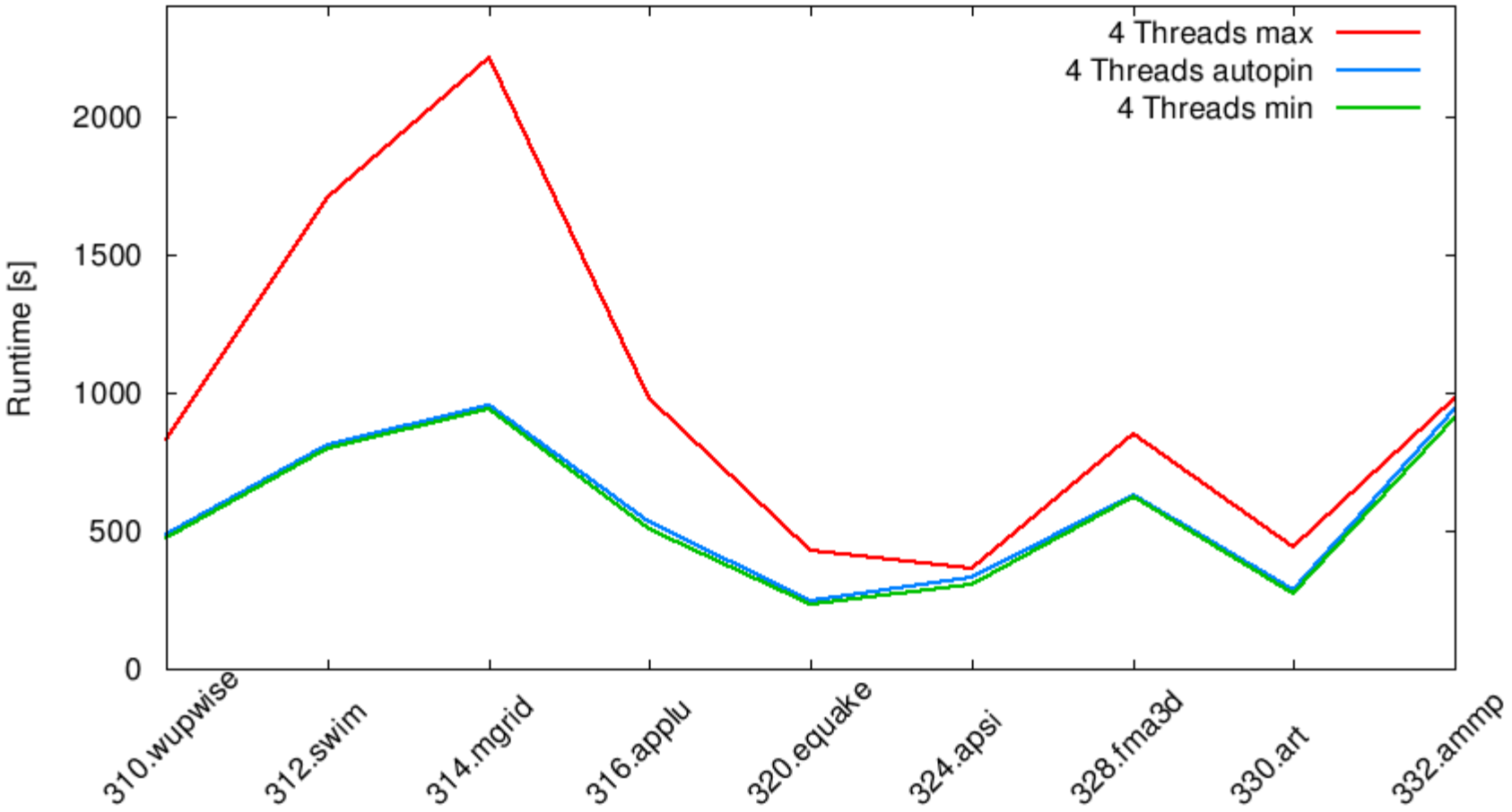
SPEC OMP

Benchmark	Description
310.wupwise	Quantum chromodynamics
312.swim	shallow water modeling
314.mgrid	multi-grid solver in 3D potential field
316.applu	parabolic/elliptic partial differential equations
318.galgel	fluid dynamics analysis of oscillatory instability
320.equake	finite element simulation of earthquake modeling
324.apsi	weather prediction
326.gafort	genetic algorithm code
328.fma3d	finite-element crash simulation
330.art	neural network simulation of adaptive resonance theory
332.Ammp	computational chemistry

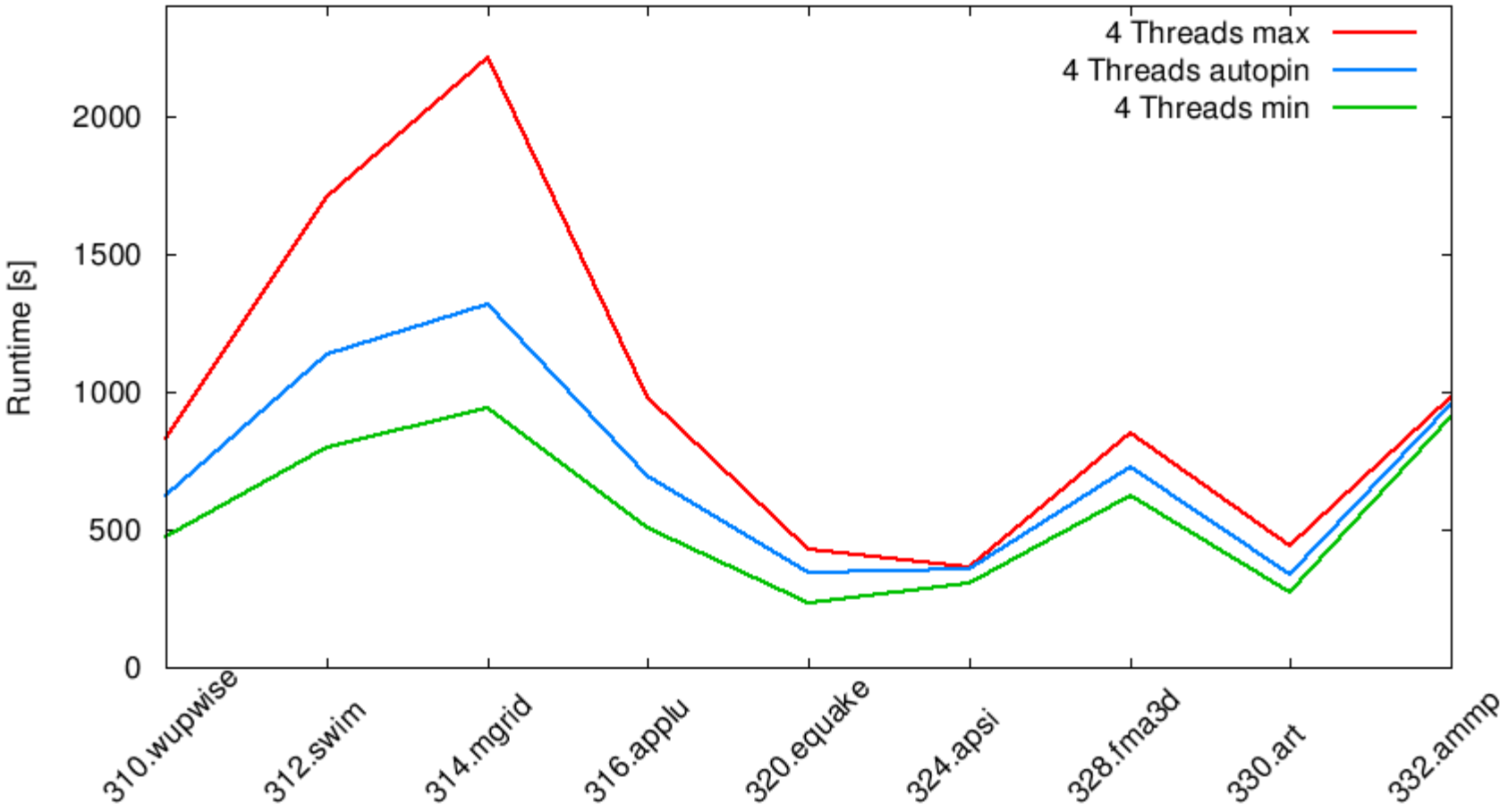
Caneland


Core #0
L2
Core #4
Core #5
L2
Core #6
Core #1
L2
Core #7
Core #8
L2
Core #9
RAM
Snoop Filter
Chipset
RAM
Core #2
L2
Core #10
Core #11
L2
Core #12
Core #3
L2
Core #13
Core #14
L2
Core #15

Caneland: Runtimes (10s sample time)



Caneland: Runtimes (30s sample time)



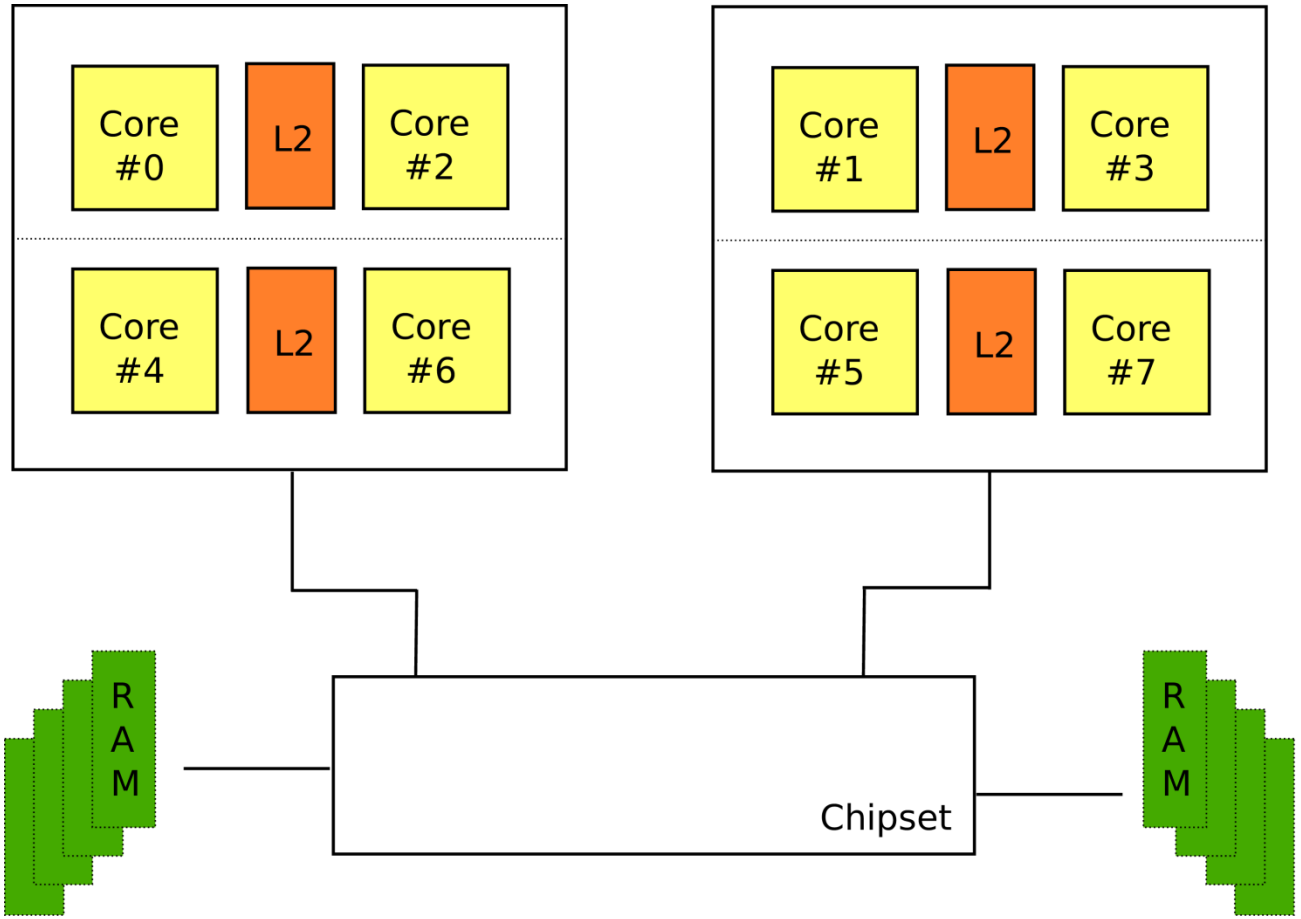
Results

	Caneland			Clovertown		Barcelona	
	2	4	8	2	4	2	4
310.wupwise							
312.swim							
314.mgrid							
316.applu							
320.equake							
324.apsi							
328.fma3d							
330.art							
332.ammp							

Results

	Caneland			Clovertown		Barcelona	
	2	4	8	2	4	2	4
310.wupwise							
312.swim							
314.mgrid							
316.applu							
320.equake							
324.apsi							
328.fma3d							
330.art							
332.ammp							

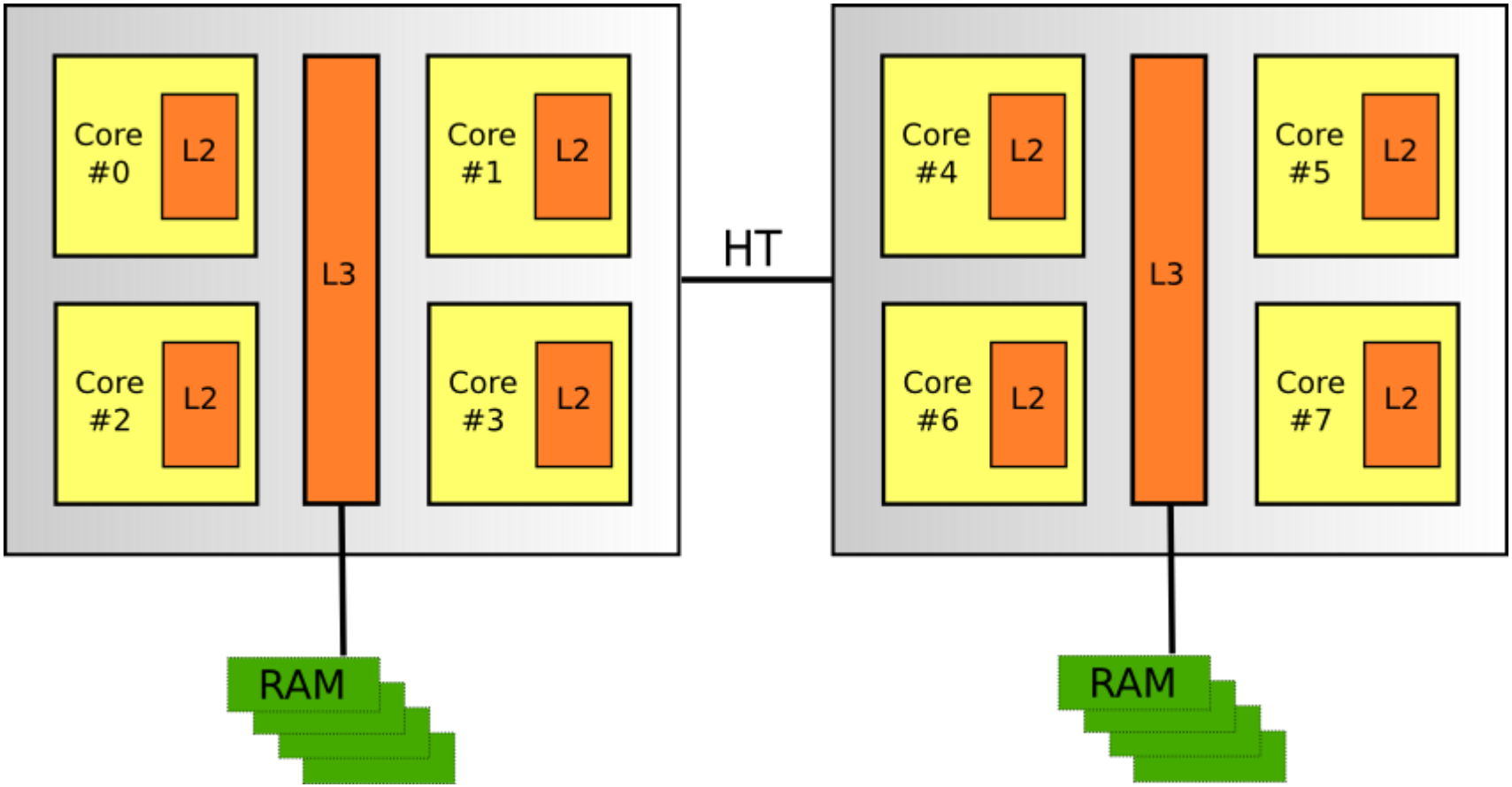
Clovertown


Core #0
L2
Core #2
Core #4
L2
Core #6
Core #1
L2
Core #3
Core #5
L2
Core #7
R A M
Chipset
R A M

Results

	Caneland			Clovertown		Barcelona	
	2	4	8	2	4	2	4
310.wupwise							
312.swim							
314.mgrid							
316.applu							
320.equake							
324.apsi							
328.fma3d							
330.art							
332.ammp							

Barcelona


Core #0
L2
Core #1
L2
Core #2
L2
Core #3
L2
L3
HT
Core #4
L2
Core #5
L2
Core #6
L2
Core #7
L2
L3
RAM
RAM

Results (w/o NUMA patch)

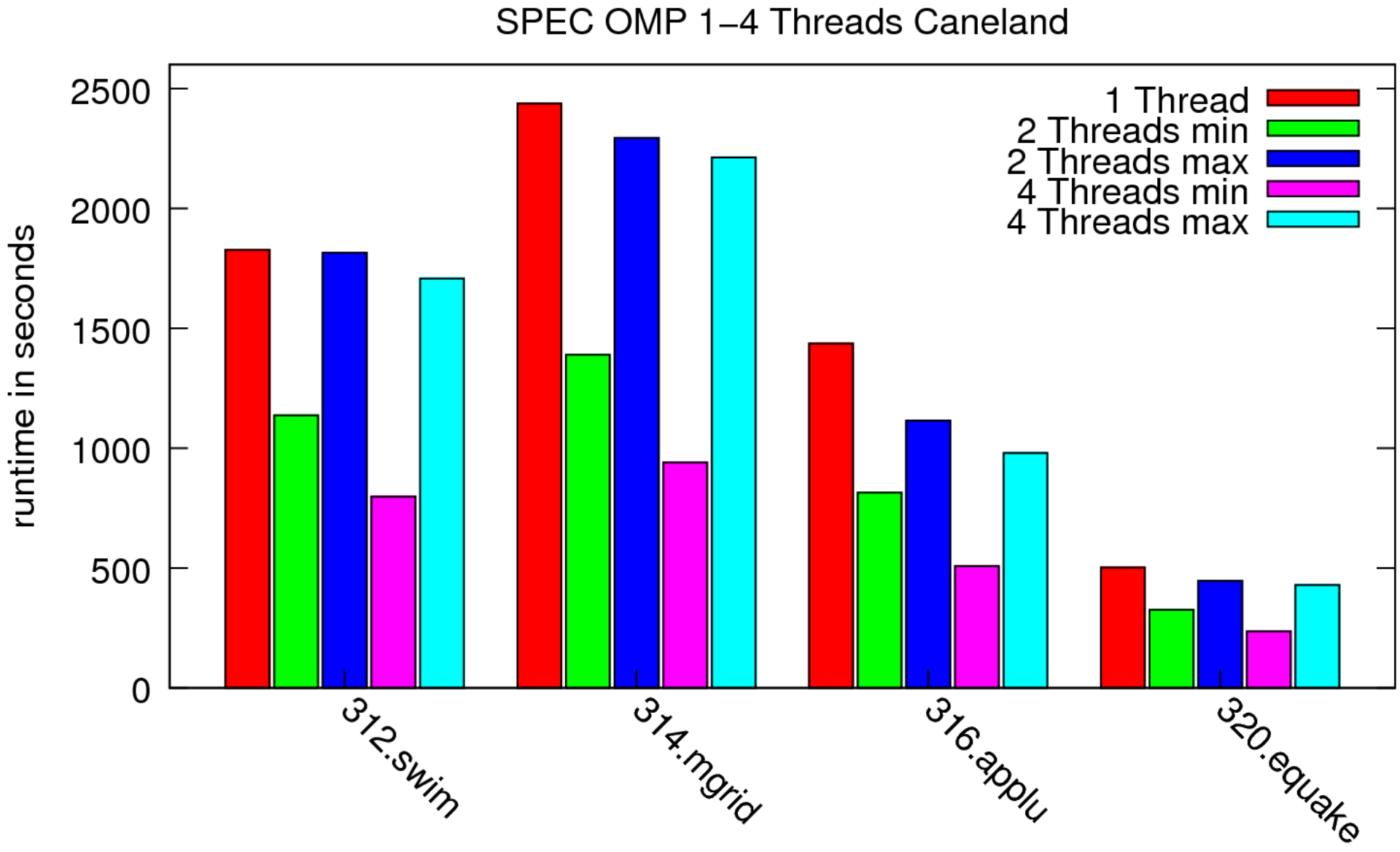
	Caneland			Clovertown		Barcelona	
	2	4	8	2	4	2	4
310.wupwise							
312.swim							
314.mgrid							
316.applu							
320.equake							
324.apsi							
328.fma3d							
330.art							
332.ammp							

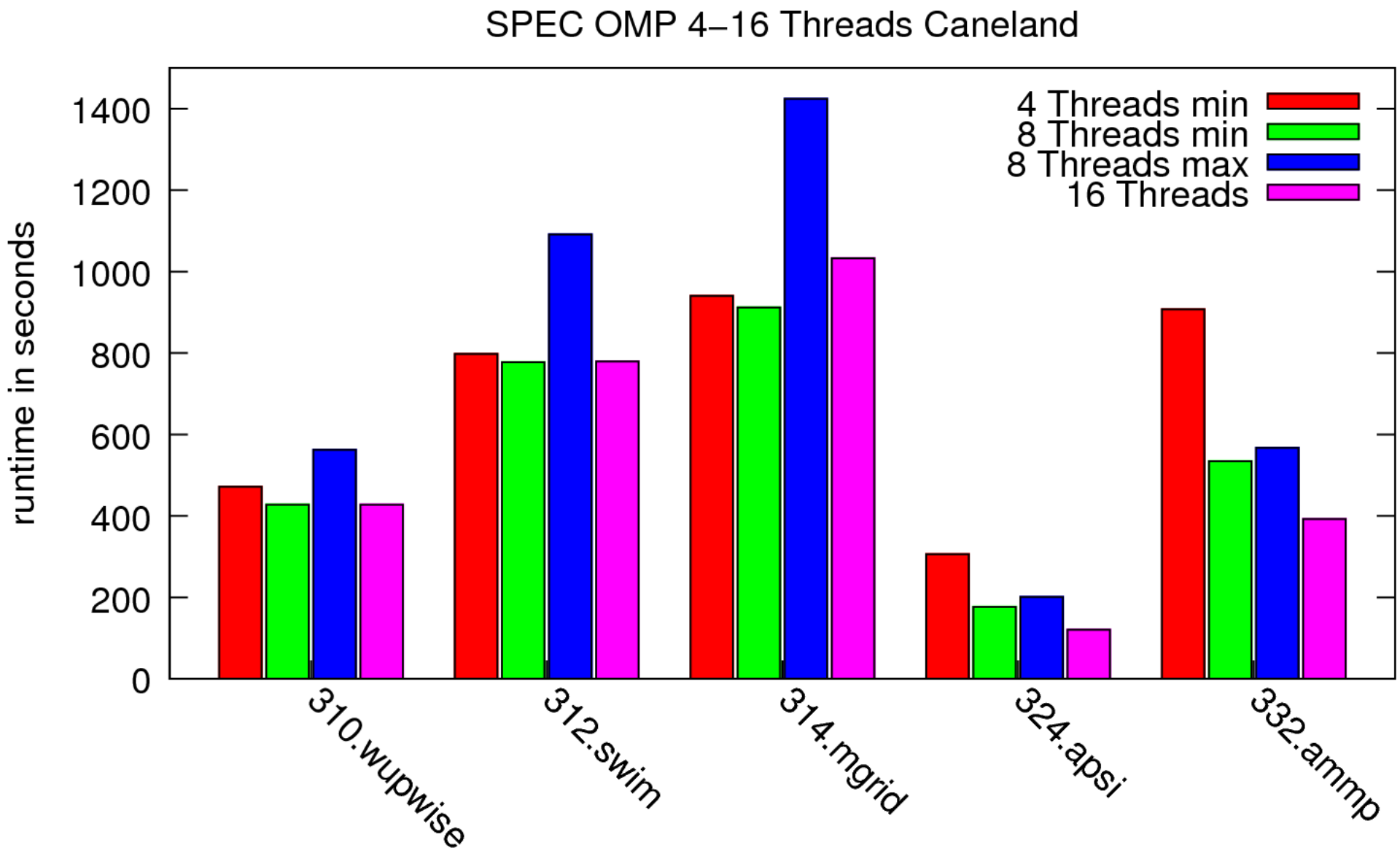
Results (with NUMA patch)

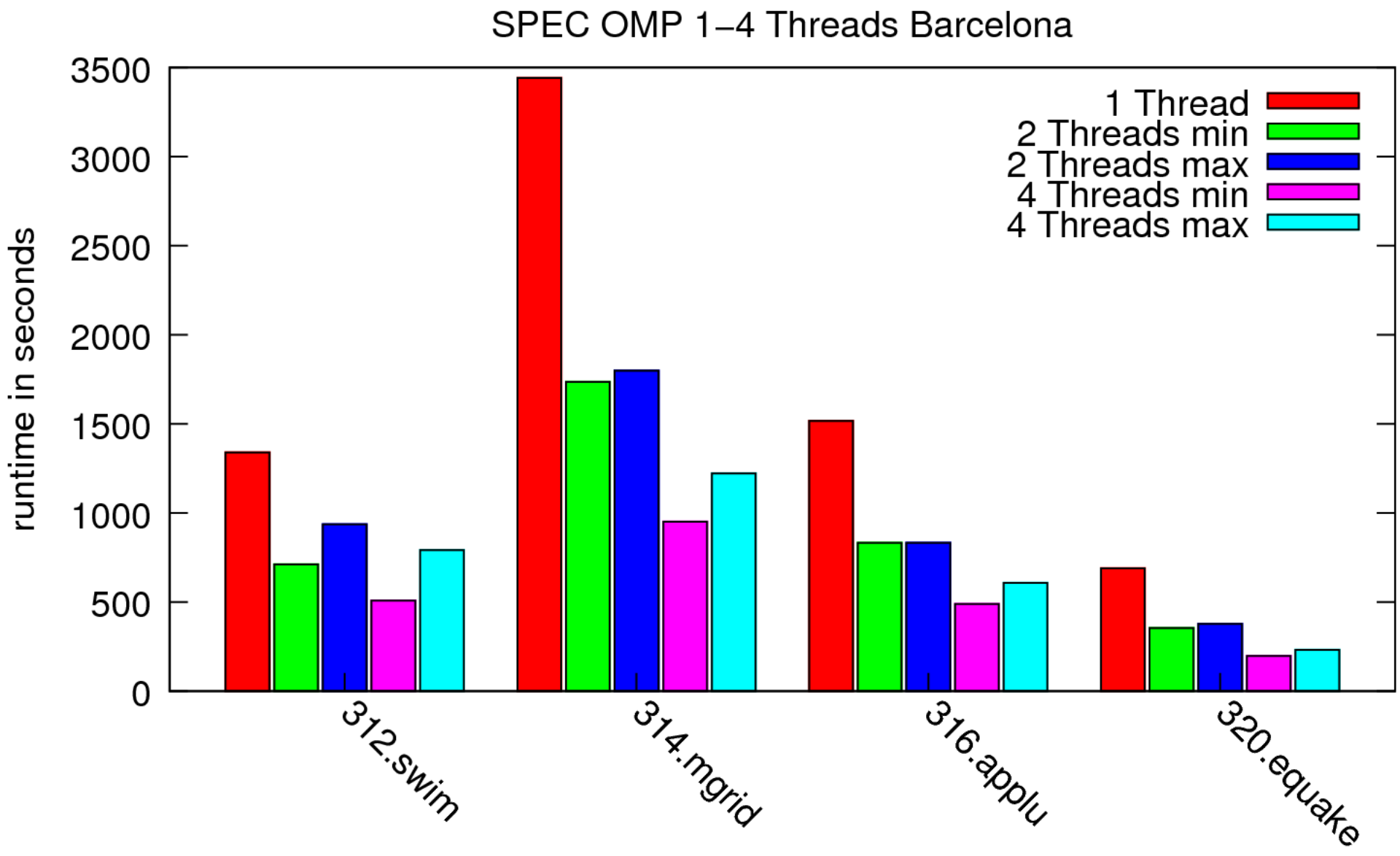
	Caneland			Clovertown		Barcelona	
	2	4	8	2	4	2	4
310.wupwise							
312.swim							
314.mgrid							
316.applu							
320.equake							
324.apsi							
328.fma3d							
330.art							
332.ammp							

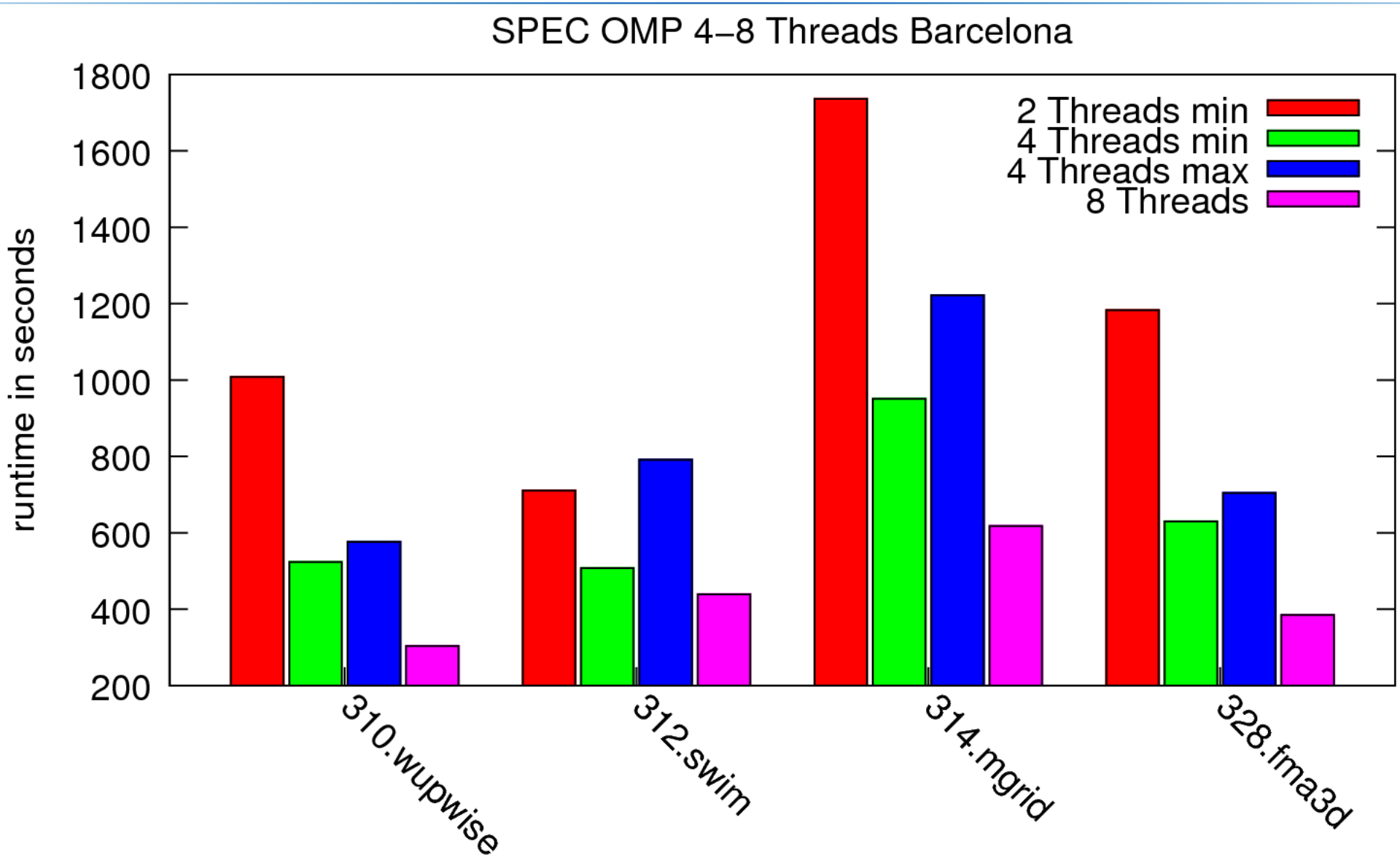
Results SPEC OMP

- Optimal pinning found in all but 2 cases autopin's alternative less than 5% slower
- Overhead less than 3% on UMA platform
- Overhead less than 7,5% on NUMA platform (Kernel level page migration)


SPEC OMP 1–4 Threads Caneland
runtime in seconds
2500
2000
1500
1000
500
0
1 Thread
2 Threads min
2 Threads max
4 Threads min
4 Threads max
312.swim
314.mgrid
316.applu
320.equake


SPEC OMP 4–16 Threads Caneland
runtime in seconds
0
200
400
600
800
1000
1200
1400
4 Threads min
8 Threads min
8 Threads max
16 Threads
310.wupwise
312.swim
314.mgrid
324.apsi
332.ammp

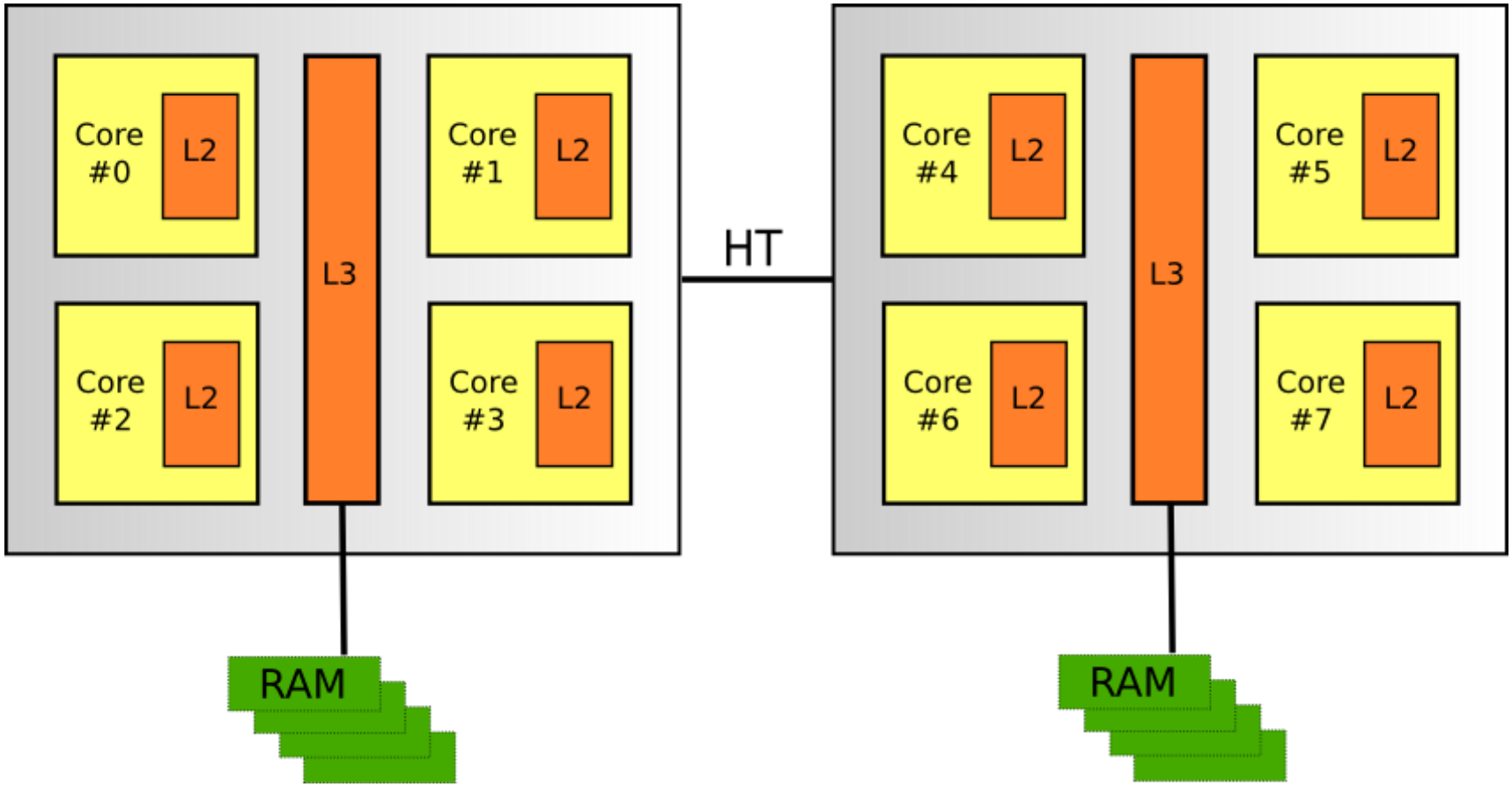

SPEC OMP 1–4 Threads Barcelona
runtime in seconds
3500
3000
2500
2000
1500
1000
500
0
1 Thread
2 Threads min
2 Threads max
4 Threads min
4 Threads max
312.swim
314.mgrid
316.applu
320.equake


SPEC OMP 4–8 Threads Barcelona
runtime in seconds
1800
1600
1400
1200
1000
800
600
400
200
2 Threads min
4 Threads min
4 Threads max
8 Threads
310.wupwise
312.swim
314.mgrid
328.fma3d

Evaluation

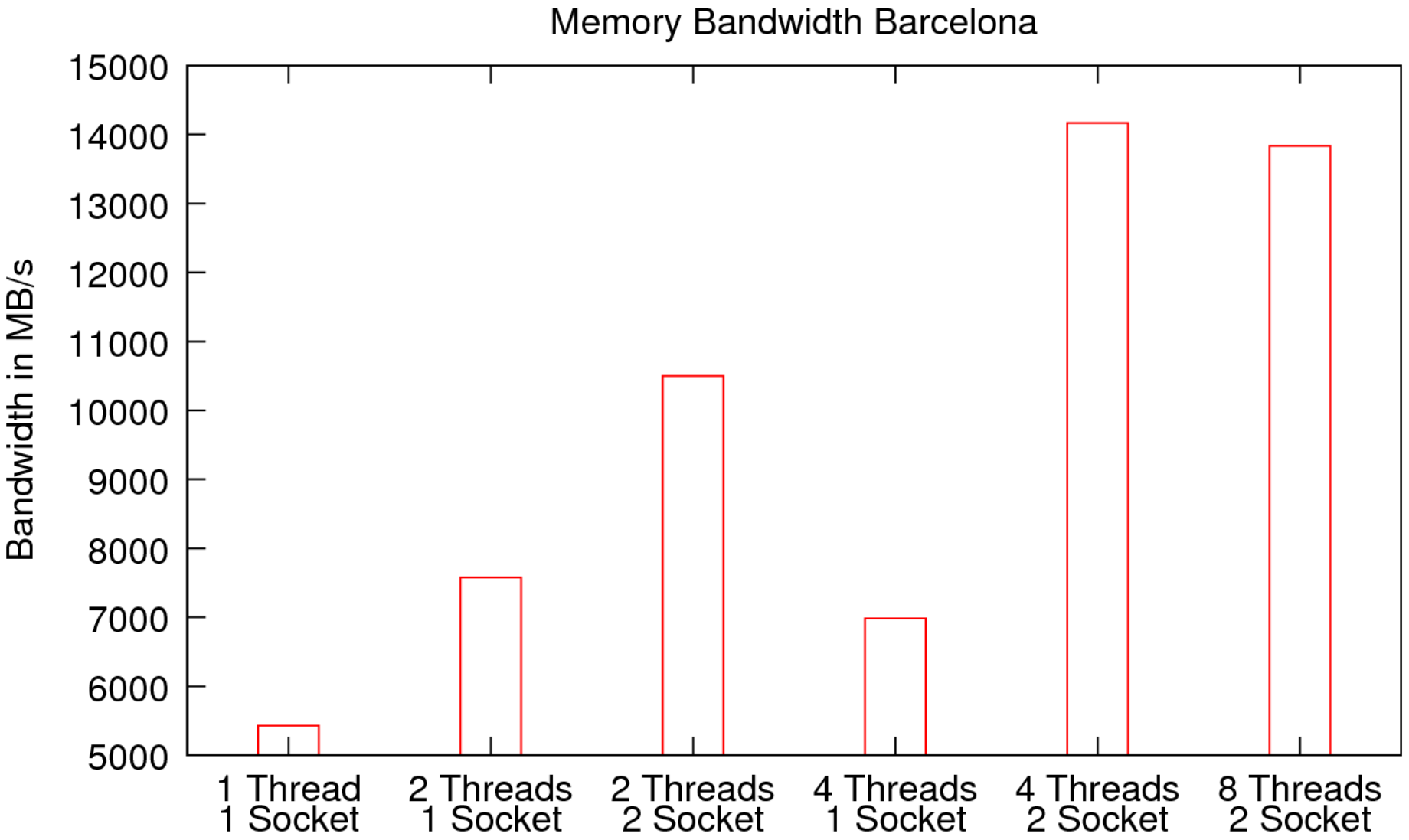
- SPEC OMP
  - Benchmark consists of real scientific applications
  - OpenMP
  - PC: INSTRUCTIONS_RETIRED
  - Several Multicore-Architekturen examined
- Memory Throughput
- MPI
- Electric power consumption

Barcelona


Core #0
L2
Core #1
L2
Core #2
L2
Core #3
L2
L3
HT
Core #4
L2
Core #5
L2
Core #6
L2
Core #7
L2
L3
RAM
RAM

Memory Bandwidth

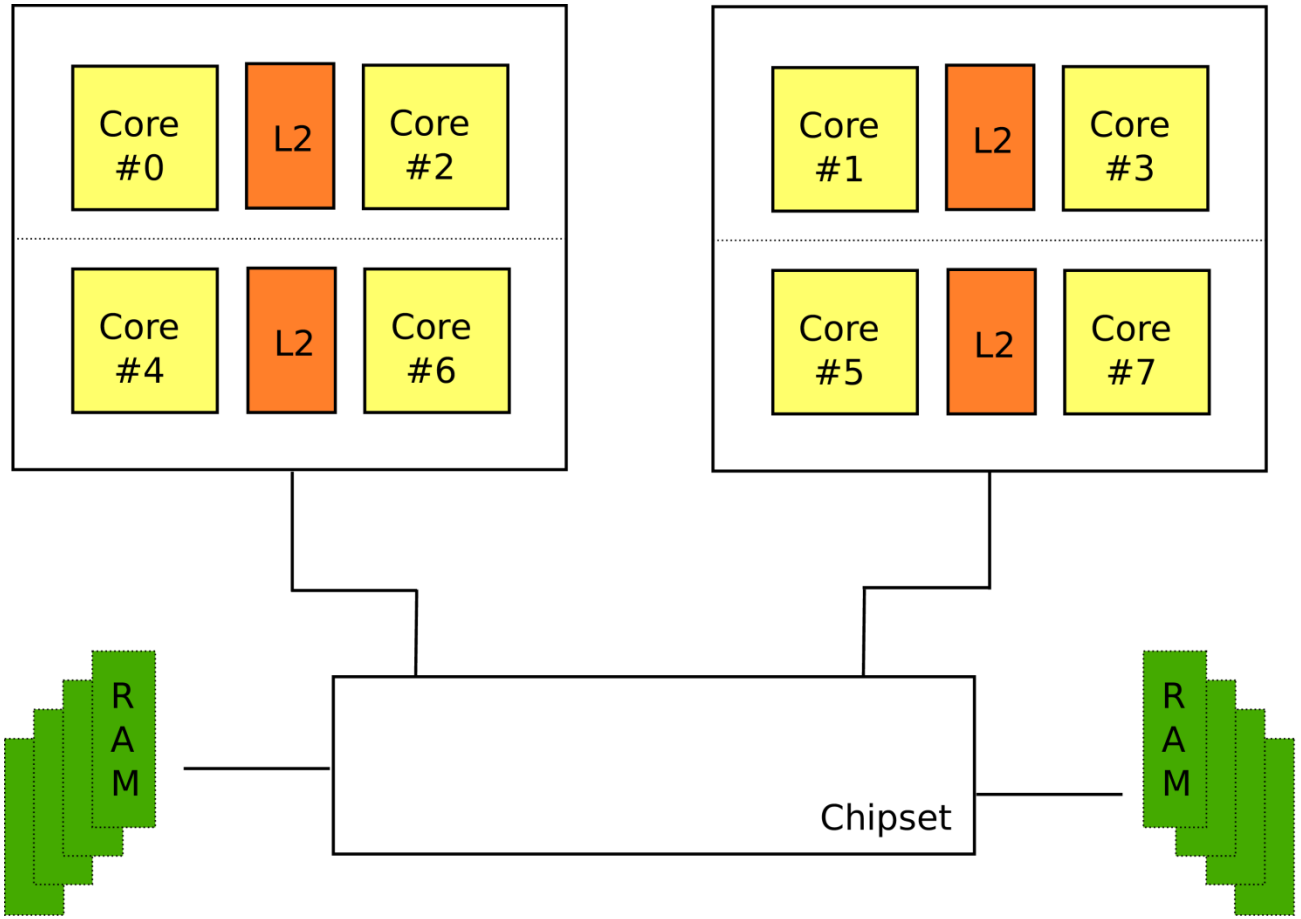
- STREAM John McCalpin
- synthetic benchmark
- copy and computation operations on large FP arrays
- Reusage of data avoided
- copy: `a[i] = b[i]`
  scale: `a[i] = q*b[i]`
  sum: `a[i] = b[i] + c[i]`
  triad: `a[i] = b[i] + q*c[i]`

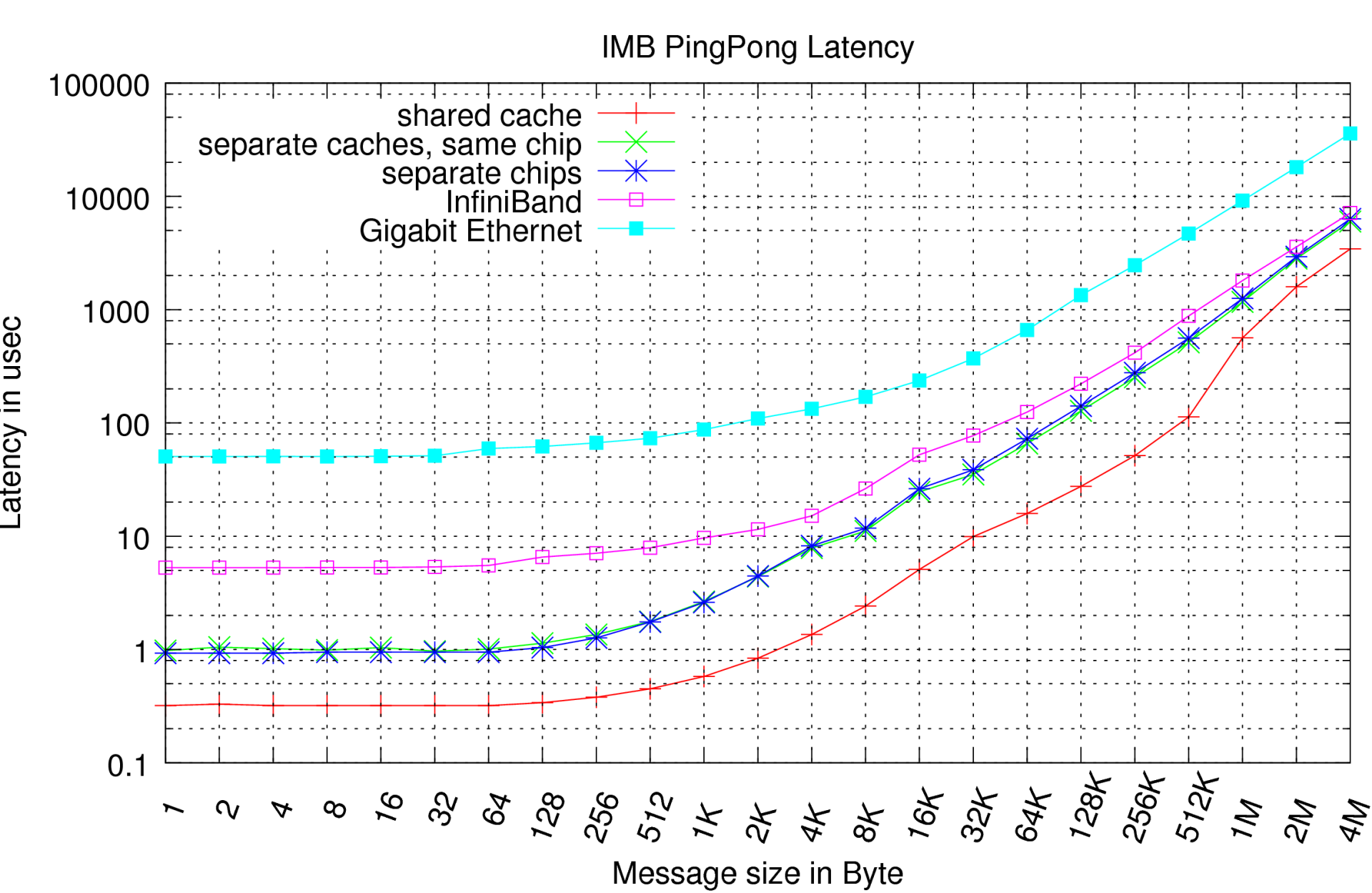

Memory Bandwidth Barcelona
Bandwidth in MB/s
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
1 Thread
1 Socket
2 Threads
1 Socket
2 Threads
2 Socket
4 Threads
1 Socket
4 Threads
2 Socket
8 Threads
2 Socket

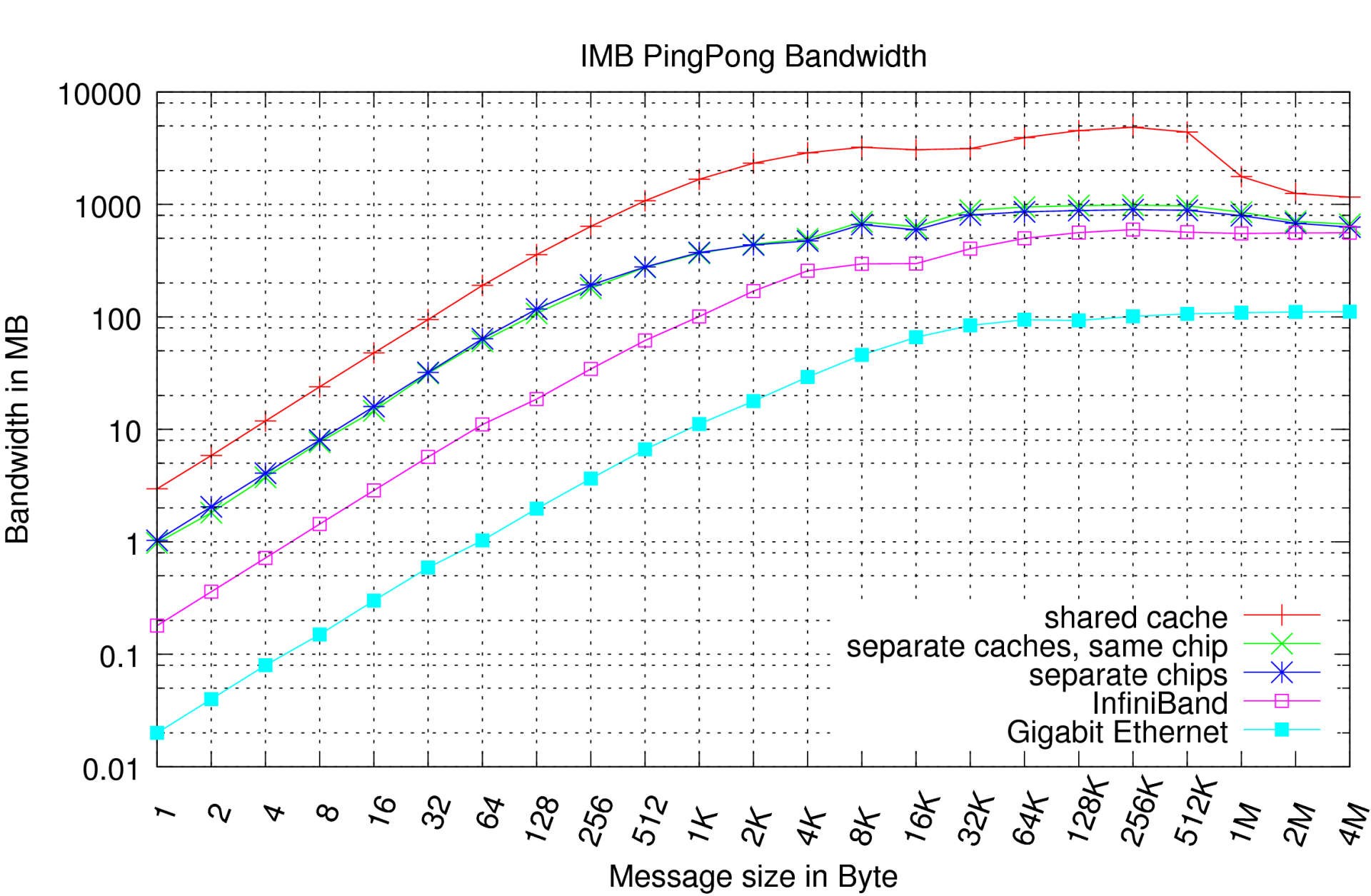
Evaluation

- SPEC OMP
  - Benchmark consists of real scientific applications
  - OpenMP
  - PC: INSTRUCTIONS_RETIRED
  - Several Multicore-Architekturen examined
- Memory Throughput
- MPI
- Electric power consumption

Clovertown


Core #0
L2
Core #2
Core #4
L2
Core #6
Core #1
L2
Core #3
Core #5
L2
Core #7
R A M
Chipset
R A M


IMB PingPong Latency
shared cache
separate caches, same chip
separate chips
InfiniBand
Gigabit Ethernet
Latency in usec
100000
10000
1000
100
10
1
0.1
1
2
4
8
16
32
64
128
256
512
1K
2K
4K
8K
16K
32K
64K
128K
256K
512K
1M
2M
4M
Message size in Byte


IMB PingPong Bandwidth
Bandwidth in MB
10000
1000
100
10
1
0.1
0.01
1
2
4
8
16
32
64
128
256
512
1K
2K
4K
8K
16K
32K
64K
128K
256K
512K
1M
2M
4M
Message size in Byte
shared cache
separate caches, same chip
separate chips
InfiniBand
Gigabit Ethernet

Evaluation

- SPEC OMP
  - Benchmark consists of real scientific applications
  - OpenMP
  - PC: INSTRUCTIONS_RETIRED
  - Several Multicore-Architekturen examined
- Memory Throughput
- MPI
- Electric power consumption

PET (Positron Emission Tomography)

- Nuclear medicine imaging
- Visualizes functional processes
  (e.g. tumor diagnostics)
- fixed detector ring around patient
- radioisotopes injected into body
- Positron vs. electron → 2 photons 180 degree
- coincidence circuit

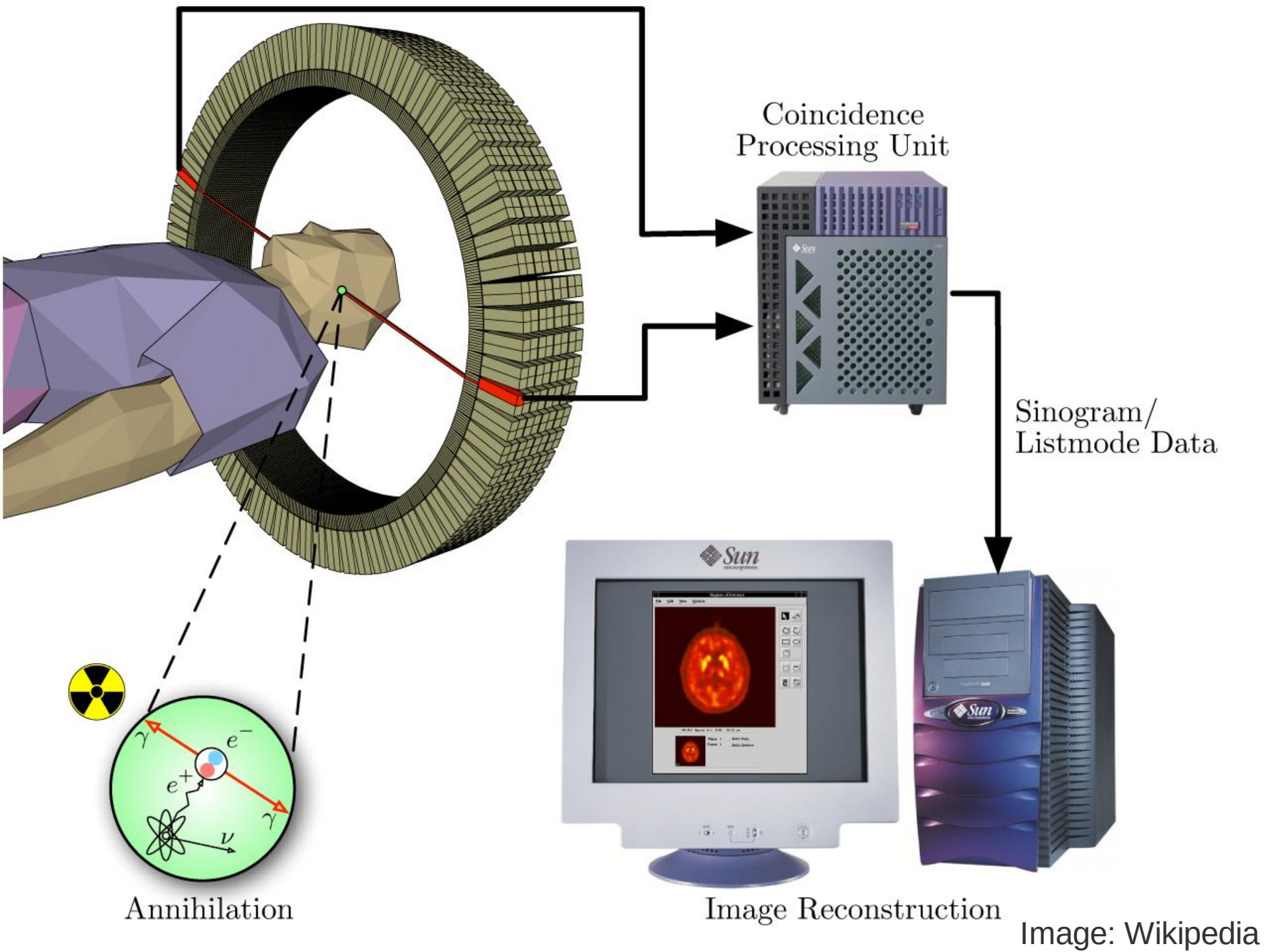

Coincidence
Processing Unit
Sinogram/
Listmode Data
Annihilation
Image Reconstruction
Image: Wikipedia

Image Reconstruction

$$\underline{g} = \underline{\underline{A}}\,\underline{f}$$

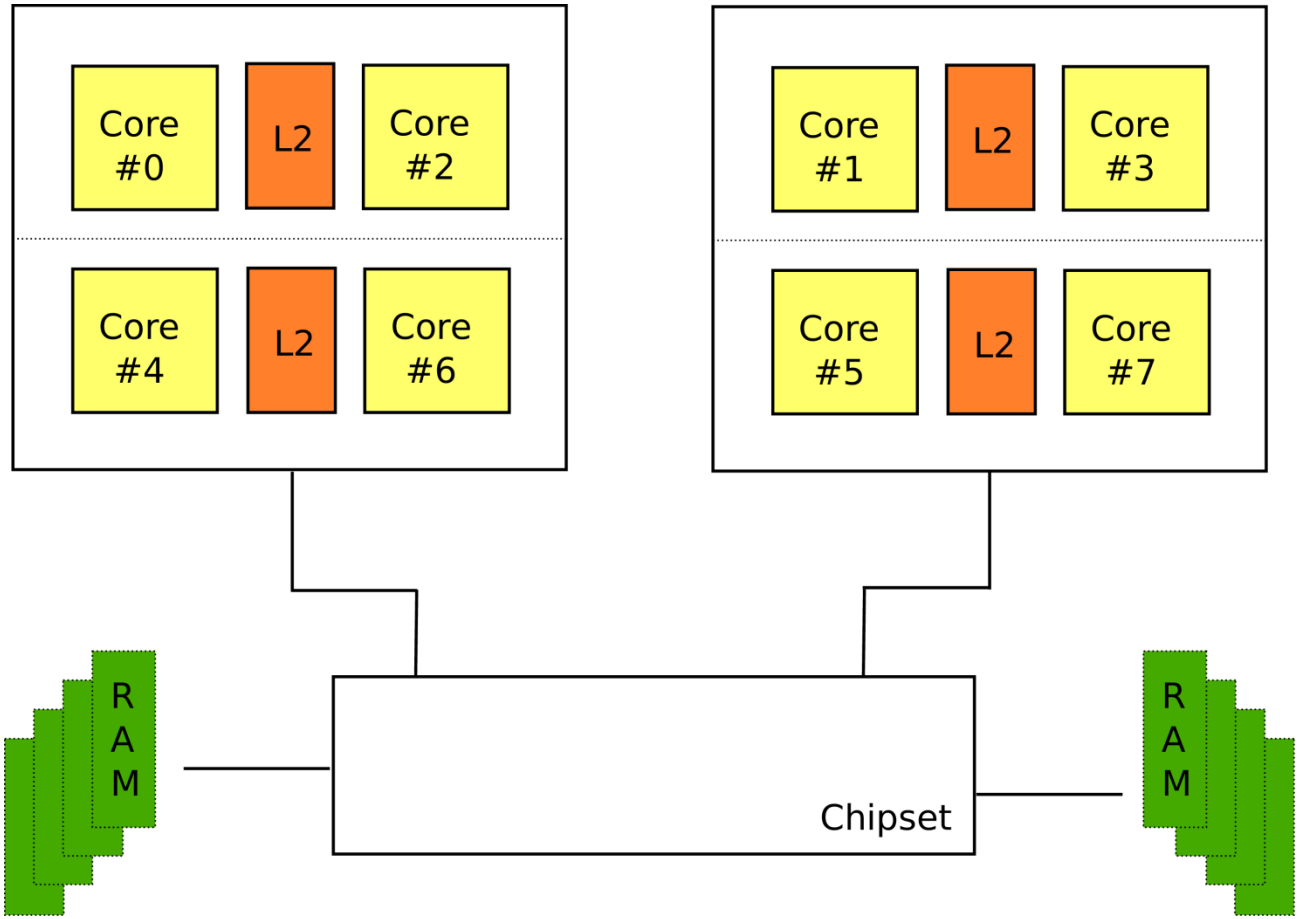
$\underline{g}$ known measurement vector

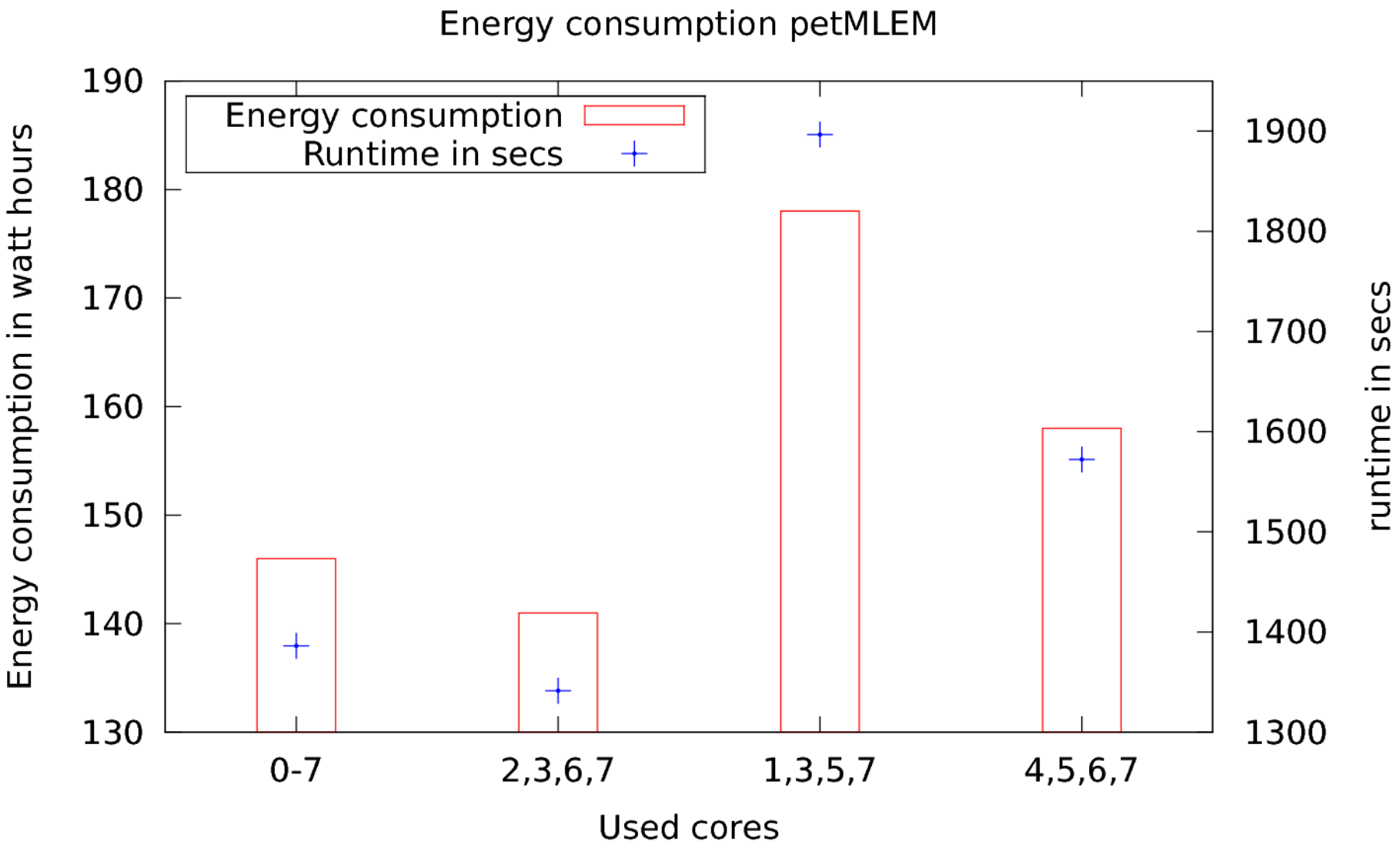
$\underline{f}$ unknown image vector

$\underline{\underline{A}}$ system matrix
(describes characteristics of detector ring)

MLEM approximates linear system

Clovertown


Core #0
L2
Core #2
Core #4
L2
Core #6
Core #1
L2
Core #3
Core #5
L2
Core #7
R A M
Chipset
R A M


Energy consumption petMLEM
Energy consumption
Runtime in secs
Energy consumption in watt hours
runtime in secs
190
180
170
160
150
140
130
1900
1800
1700
1600
1500
1400
1300
0-7
2,3,6,7
1,3,5,7
4,5,6,7
Used cores

Conclusion and Outlook

- Pinning is essential on multicore systems
- Will become even more important on many core architectures
- Tools can reliably find optimal pinnings on UMA and NUMA architectures

- Outlook – `autopin2`
  - new design: perf performance counters subsystem
  - Flexible and modular Design:
    - perfmon, perf
    - Energy, runtime, user defined objective functions
    - Back channel from application to `autopin2`

Questions?