Can We Understand Performance Counter Results?

Vince Weaver

ICL Lunch Talk

23 July 2010

How Do We Know if Counters are Working?

Three common failures:

- Wrong counter (PAPI, Kernel, User)
- Counter works but gives wrong values
- Counter is giving "right" values but documentation is wrong

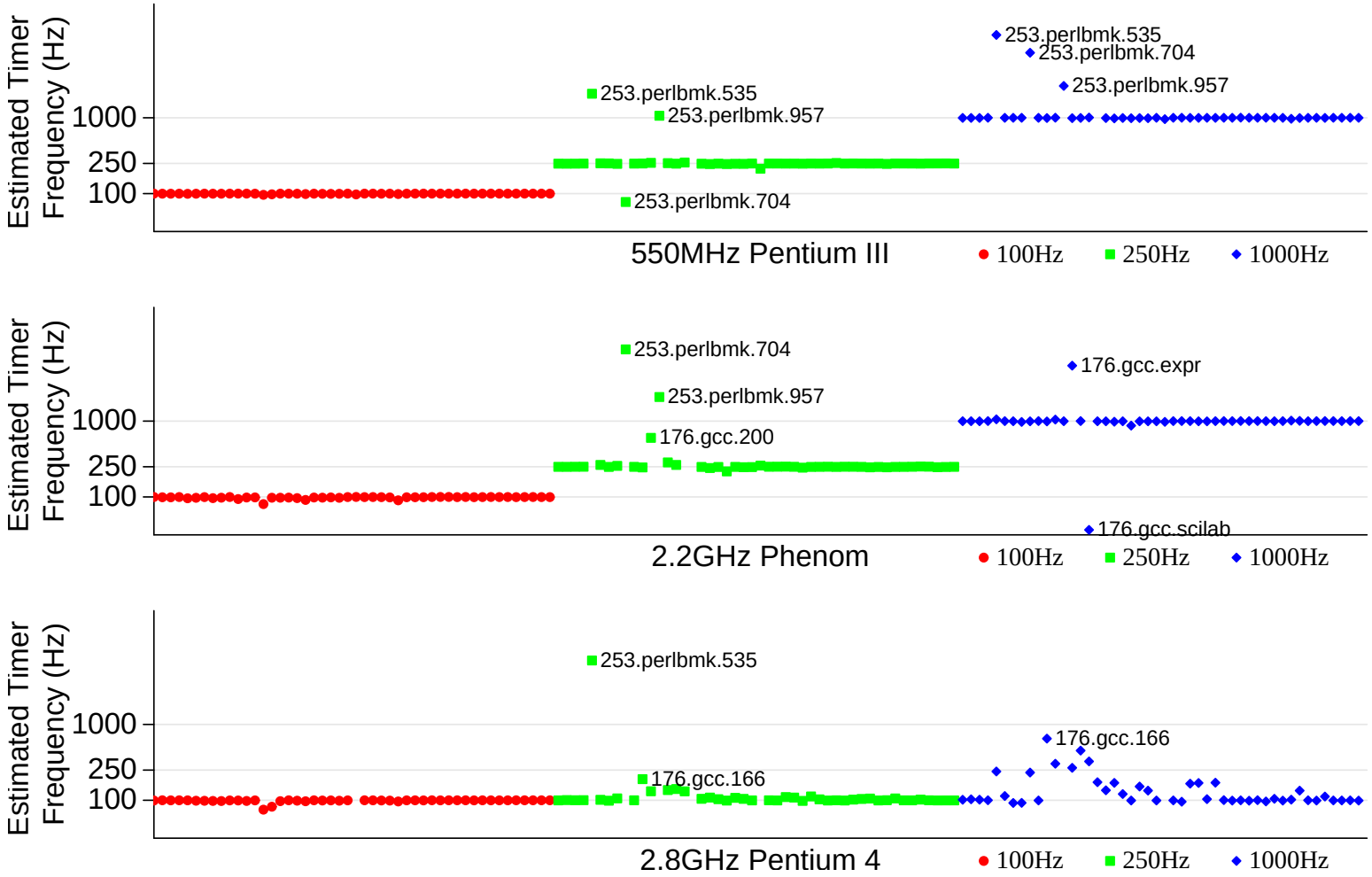
Deterministic Events Easiest to Validate

- Retired Instructions
- Retired Branches
- Retired Loads and Stores
- Retired Multiplies and Divides
- Retired μops
- Retired Floating Point and SSE
- Other (`fxch`, `cpuid`, move operations, serializing instructions, memory barriers, and not-taken branches)

Ideal Deterministic Events

- Results are same run-to-run
- Event is frequent enough to be useful
- The expected count can easily be determined by code inspection
- Available on many processors

Retired Instruction Overcount



Tracking Down the Source of Overcounts

- Work backward from existing benchmarks?
- Assembly Language!

Contributors to Instruction Count on x86_64

Expected Count
+1 for every Hardware Interrupt
+1 for each memory page touched
+1 for first floating point ins
Processor Errata
Undocumented processor quirks

Retired Instruction Results

machine	Raw Results	Adjusted Results	Adjustments Made
Expected	226,990,030	226,990,030	
Core2	10,793±40	12±1	HW Int
Atom	11,601±495	-43±12	HW Int
Nehalem	11,794±1316	2±7	HW Int
Nehalem-EX	11,915±9	6±2	HW Int
Pentium D R	2,610,571±8	200,561±8	Instr Double Counts
Pentium D C	10,794±28	-52±5	HW Int
Phenom	310,601±11	11±0	HW Int, FP Except
Istanbul	311,830±78	9±1	HW Int, FP Except
Pin	2.51868e9±0	0±0	Count rep string as 1
Qemu	-16,410,000±0		
Valgrind	-6,909,896±0		

Retired Stores Results

machine	Raw Results	Adjusted Results	Adjustments Made
Expected	24,060,000	24,060,000	
Core2	0±0	0±0	
Atom	n/a	n/a	
Nehalem	411,632±1483	410,014±1	HW Int
Nehalem-EX	411,914±6	410,018±1	HW Int
Pentium D	-12,880,000±0		
Phenom	n/a	n/a	
Istanbul	n/a	n/a	
Pin	802,180,000±0	980,000±0	Count rep string as 1
Qemu	n/a	n/a	
Valgrind	-7,542,176±0		

Retired Floating Point

machine	FP1	FP2	SSE
Core2	73,500,376±140	40,299,997± 0	23,200,000± 0
Atom	38,800,000± 0	0± 0	88,299,597± 792
Nehalem	50,150,648±140	17,199,998± 1	24,201,639± 957
Nehalem-EX	50,155,704±562	17,199,998± 2	24,007,005±197,401
Pentium D	100,400,262± 9	140,940,555±39,287	53,149,435±522,879
Phenom	26,600,001± 0	112,700,001± 0	15,800,000± 0
Istanbul	26,600,001± 0	112,700,001± 0	15,800,000± 0

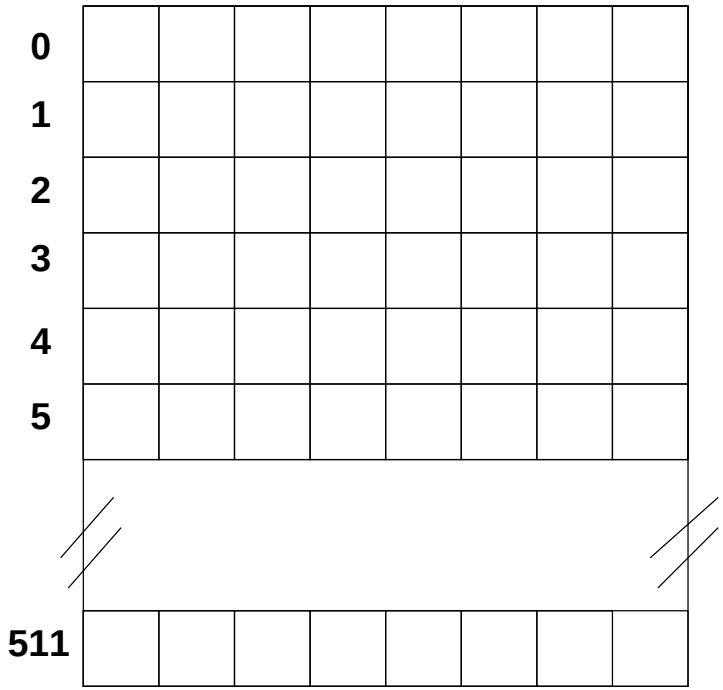
Other Architectures

- ARM – Cannot select only userspace events
- ia64 – Loads, Stores, Instructions all deterministic
- POWER – On Power6 Instructions is deterministic, Branches is not
- SPARC – on Niagara1, Instructions is deterministic

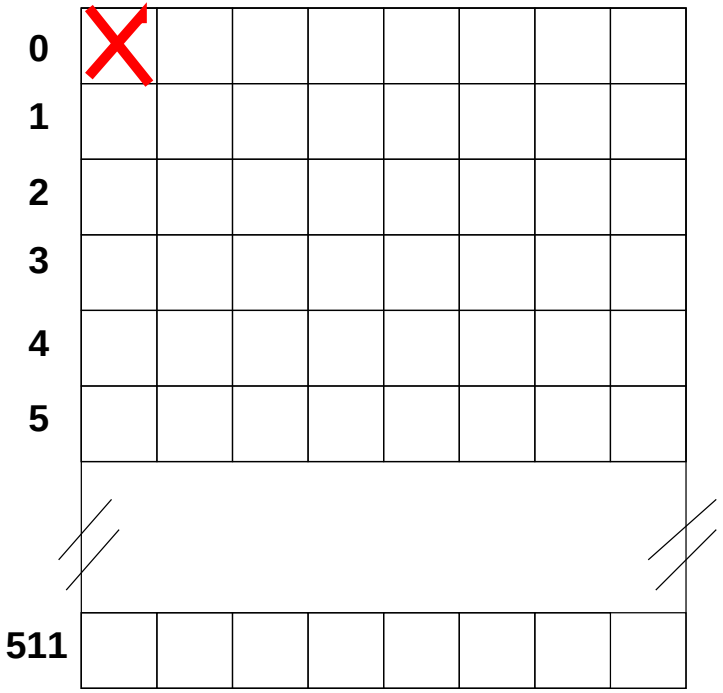
Non-deterministic Events

- Cache and Memory Related
- Branch Predictor
- Cycles
- Stalls

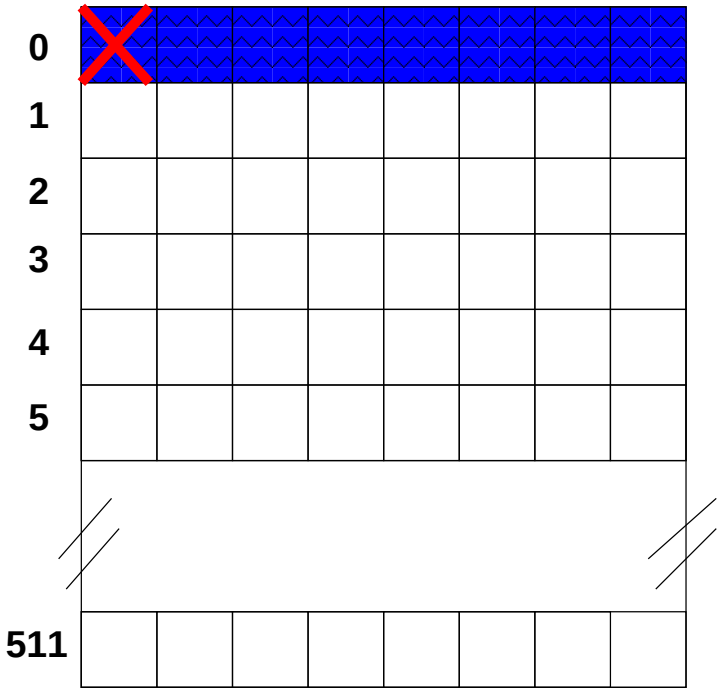
Simplistic Cache Model



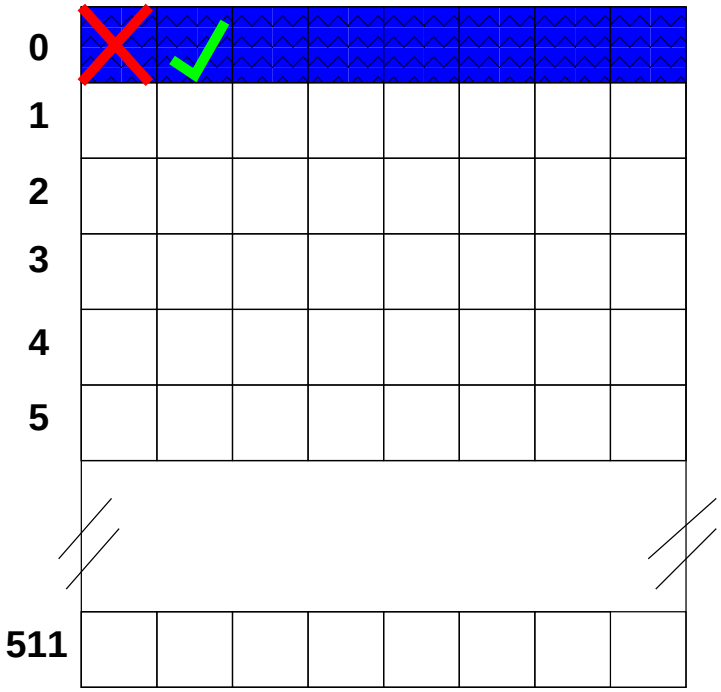
Simplistic Cache Model



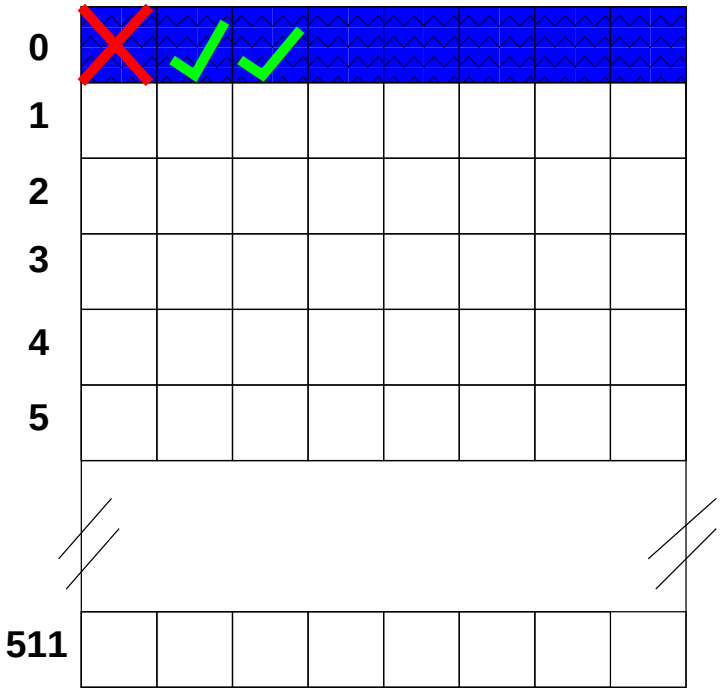
Simplistic Cache Model



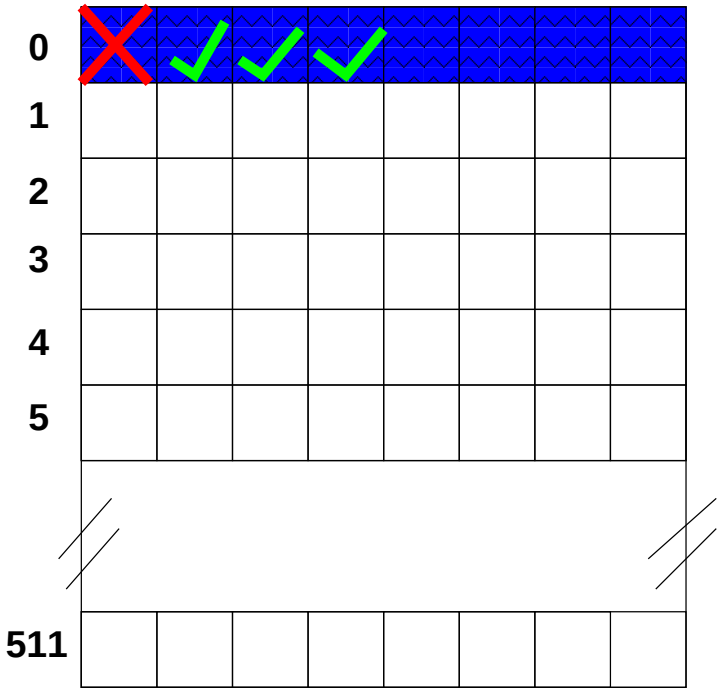
Simplistic Cache Model



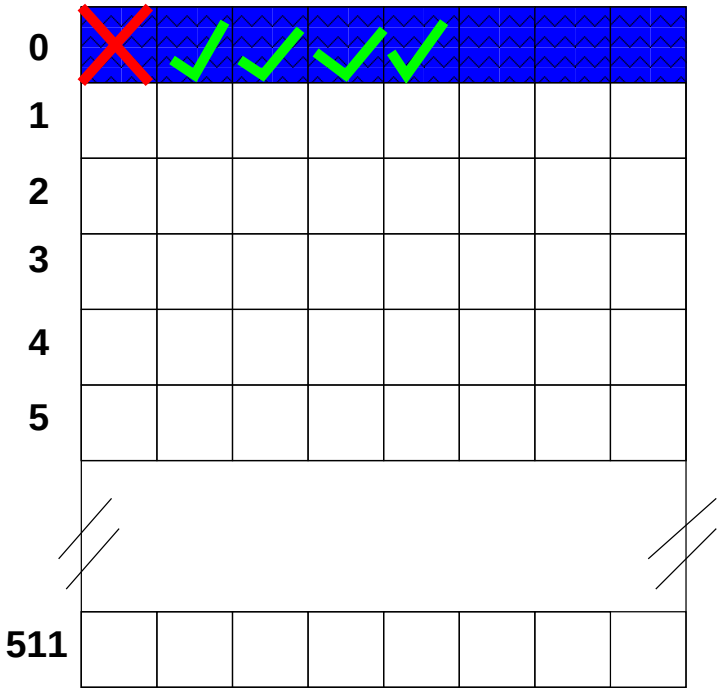
Simplistic Cache Model



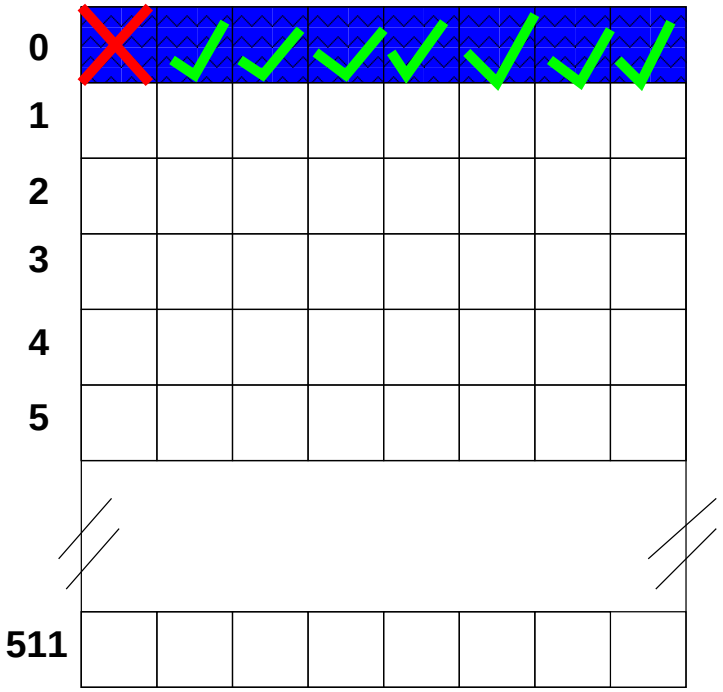
Simplistic Cache Model



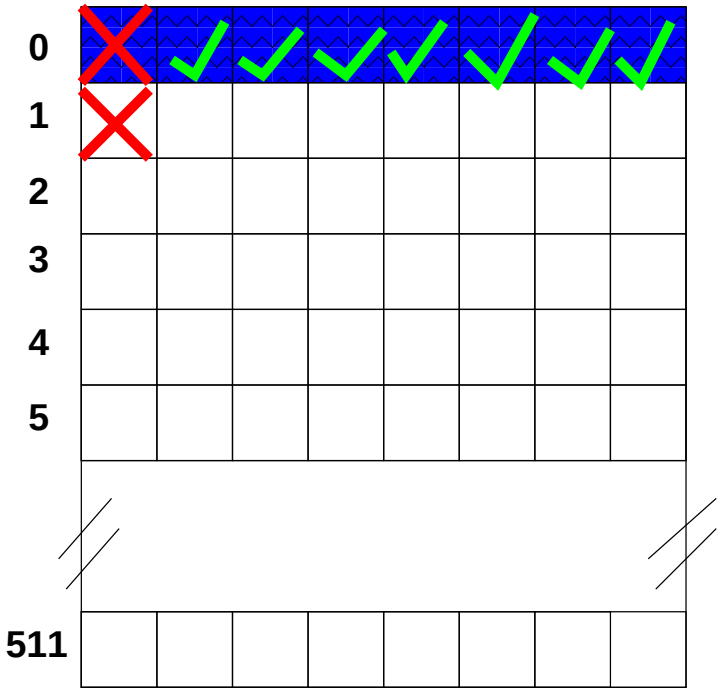
Simplistic Cache Model



Simplistic Cache Model



Simplistic Cache Model



L1 Data Cache Accesses

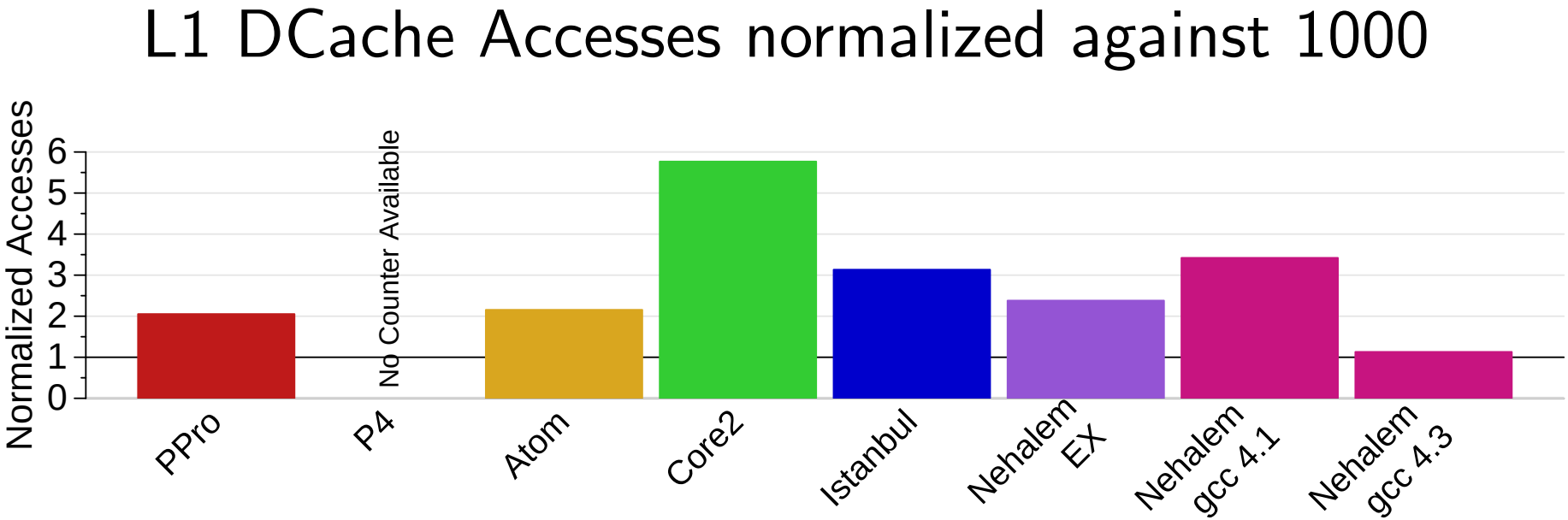
```
float array[1000],sum = 0.0;

PAPI_start_counters(events,1);

for(int i=0; i<1000; i++) {
   sum += array[i];
}

PAPI_stop_counters(counts,1);
```

PAPI_L1_DCA


L1 DCache Accesses normalized against 1000
Normalized Accesses
0
1
2
3
4
5
6
No Counter Available
PPro
P4
Atom
Core2
Istanbul
Nehalem EX
Nehalem gcc 4.1
Nehalem gcc 4.3

PAPI_L1_DCA

Expected Code

```
* 4020d8:       f3 0f 58 00             addss  (%rax),%xmm0
  4020dc:       48 83 c0 04             add    $0x4,%rax
  4020e0:       48 39 d0                cmp    %rdx,%rax
  4020e3:       75 f3                   jne    4020d8 <main+0x328>
```

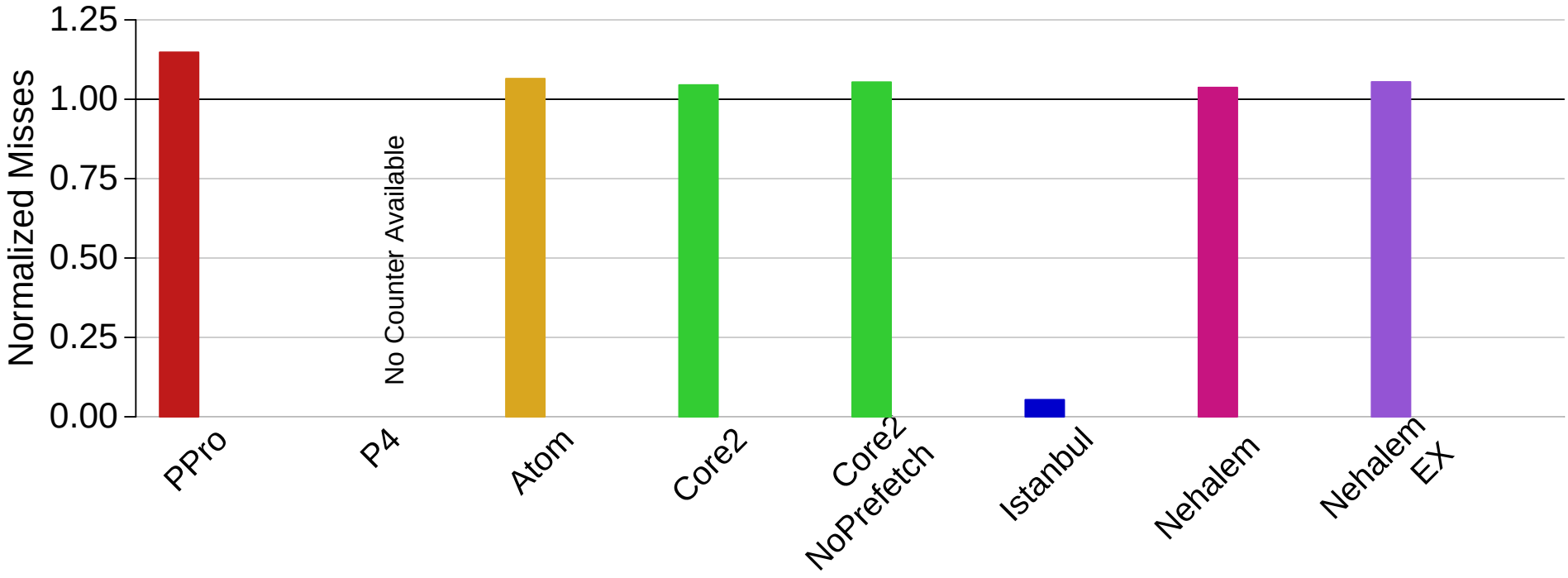
Unexpected Code

```
* 401e18:       f3 0f 10 44 24 0c       movss  0xc(%rsp),%xmm0
* 401e1e:       f3 0f 58 04 82          addss  (%rdx,%rax,4),%xmm0
  401e23:       48 83 c0 01             add    $0x1,%rax
  401e27:       48 3d e8 03 00 00       cmp    $0x3e8,%rax
* 401e2d:       f3 0f 11 44 24 0c       movss  %xmm0,0xc(%rsp)
  401e33:       75 e3                   jne    401e18 <main+0x398>
```

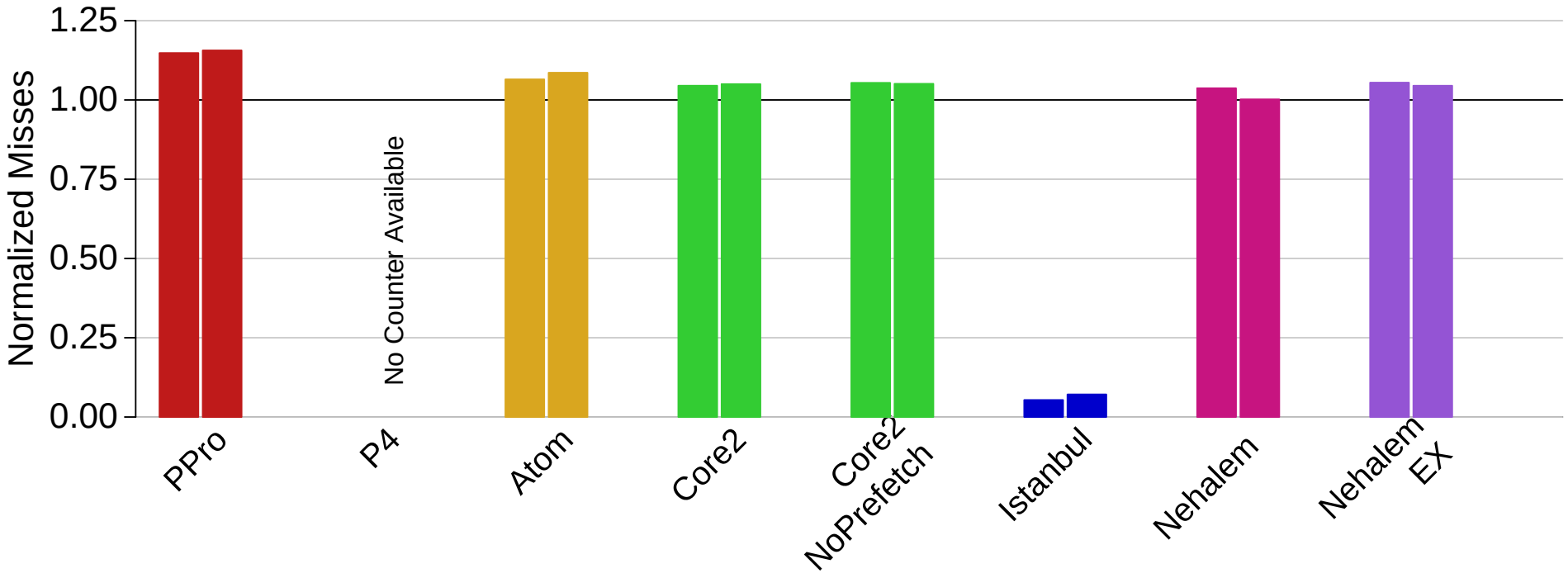
L1 Data Cache Misses

- Allocate array as big as L1 DCache
- Walk through the array byte-by-byte
- Count misses with `PAPI_L1_DCM` event

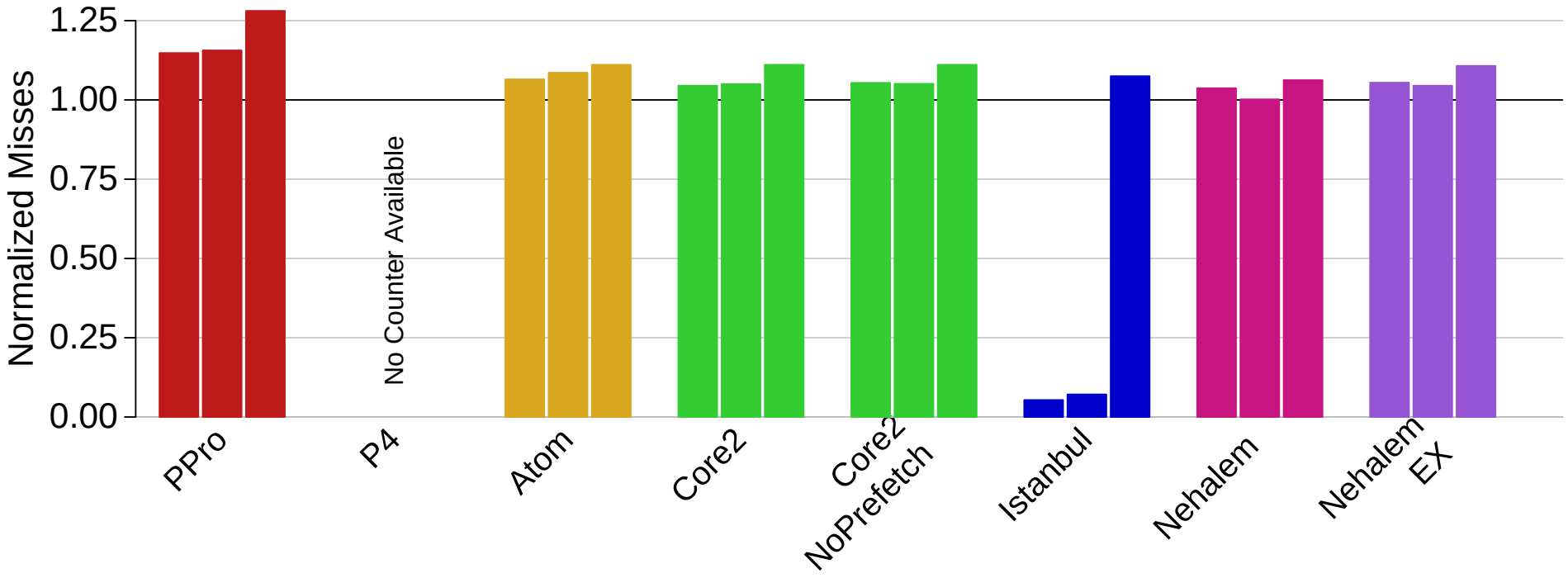
PAPI_L1_DCM – Forward



PAPI_L1_DCM – Reverse



PAPI_L1_DCM – Random



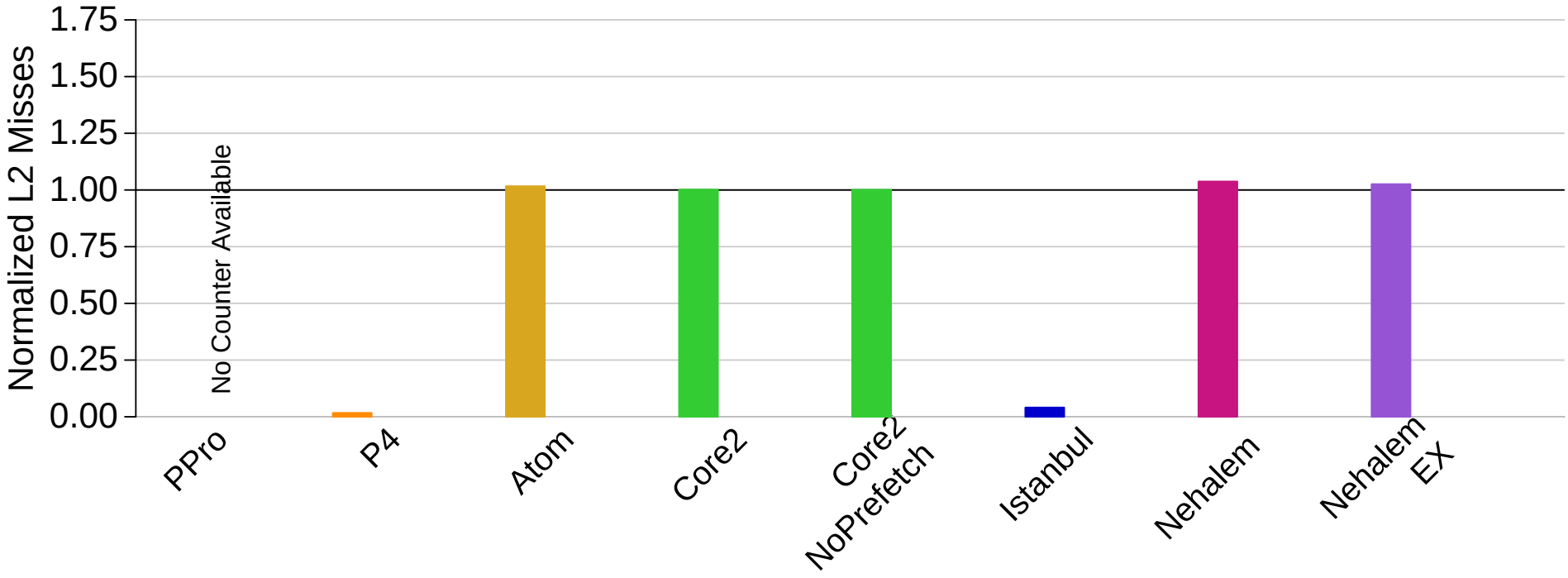
L1D Sources of Divergences

- Hardware Prefetching
- PAPI Measurement Noise
- Operating System Activity
- Non-LRU Cache Replacement

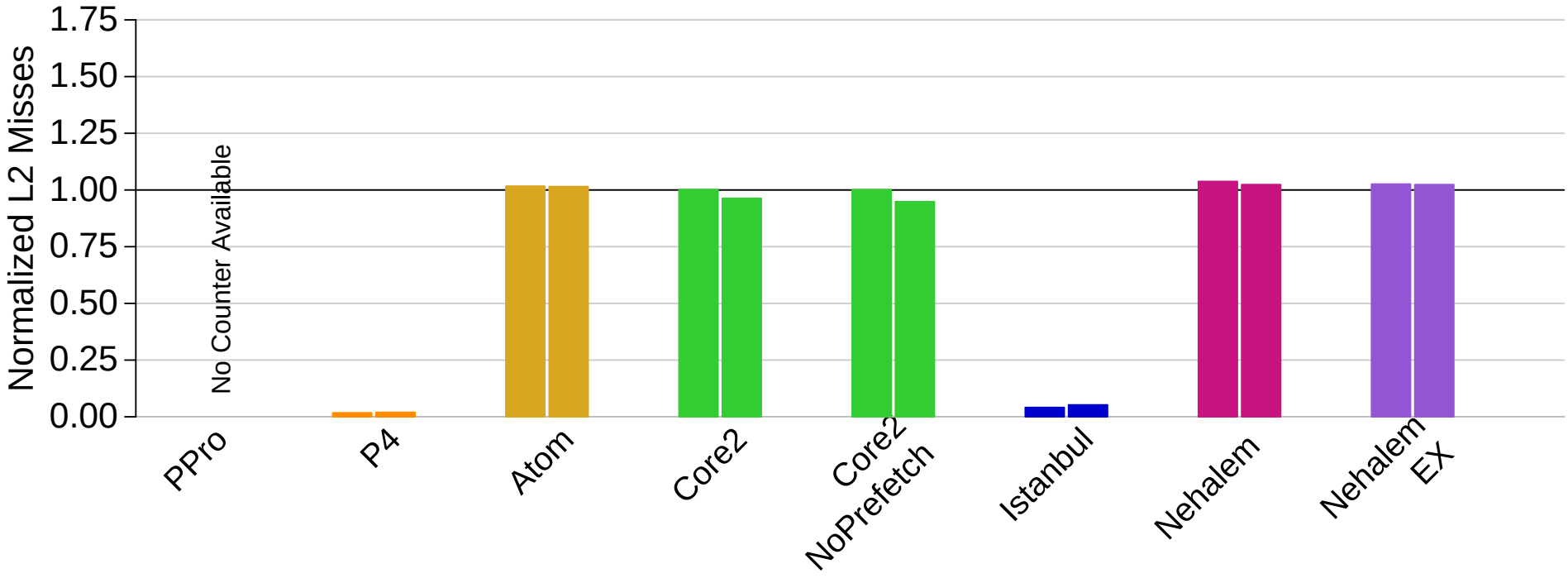
L2 Total Cache Misses

- Allocate array as big as L2 Cache
- Walk through the array byte-by-byte
- Count misses with `PAPI_L2_TCM` event

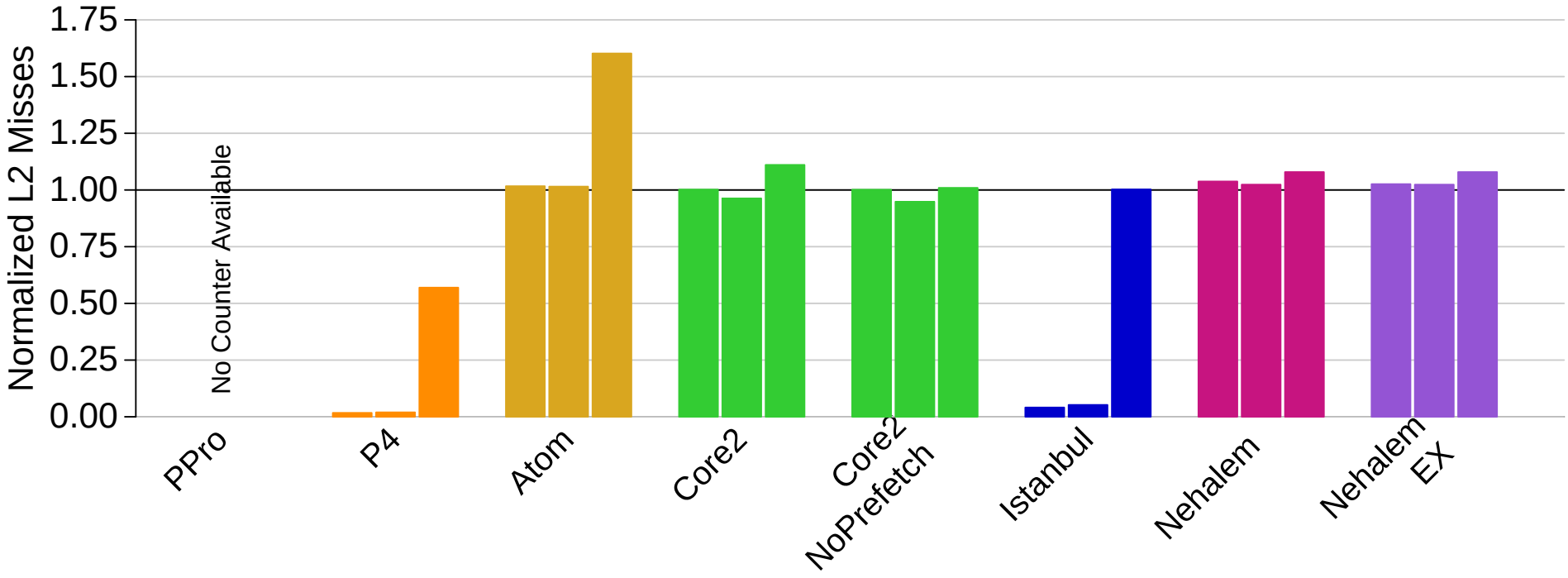
PAPI_L2_TCM – Forward



PAPI_L2_TCM – Reverse



PAPI_L2_TCM – Random



L2 Sources of Divergences

- Hardware Prefetching
- PAPI Measurement Noise
- Operating System Activity
- Non-LRU Cache Replacement
- Cache Coherency Traffic

Conclusions

- Counters are hard to understand
- Automated testing is possible

Questions?

Pictures from a Maryland-style Crab Feast