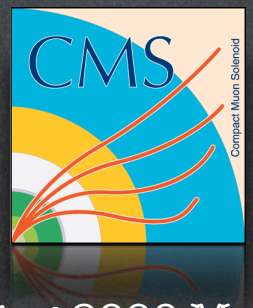
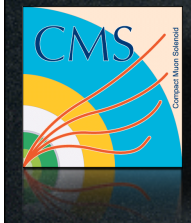
Benchmarking CMS applications

[Using CMSSW to benchmark machines]

Gabriele Benelli
CERN



HEPiX Spring 2008 May 8th 2008



HEPiX cpu performance

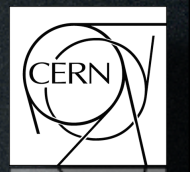
- Test the validity of the industry-standard benchmarks (SPEC CPU) when compared with HEP experiments code
- Provide some recommendation to guide institutional purchases
- Use CMSSW applications to benchmark a number of machines with different architecture

Benchmarked machines

- Used 10 machines with different architectures, frequencies, memory:
  - 7 machines at CERN from the lxbench cluster
    - Cluster TWiki: https://twiki.cern.ch/twiki/bin/view/FIOgroup/TsiLxbench
  - 1 machine at DESY Zeuten (hpbl1)
  - 2 machines at INFN Padua (lxcmssrv7, lxcmssrv8)

	lxbench01	lxbench02	lxbench03	lxbench04	lxbench05	lxbench06	lxbench07	lxcmssrv07	lxcmssrv08	hpbl1
Number of cores	2	2	4	4	4	4	8	8	8	8
Frequency (GHz)	2.8	2.8	2.2	2.66	3.0	2.6	2.33	2.33	2.1	2.83
Cache (could be L2/L3) (MB)	1	2	2	4	4	2	8	12	2	12
Memory (GB)	2	4	2	8	8	8	16	16	16	16
Processor	Nocona	Irvingdale	Opteron 275	Woodcrest	Woodcrest	Opteron 2218 Rev.F	Clovertown	Xeon HarperTown E5410	Opteron Barcelona 2352	Xeon HarperTown E5440
Vendor	Intel	Intel	AMD	Intel	Intel	AMD	Intel	Intel	AMD	Intel

CMSSW benchmarking



- All tests run with CMSSW_2_0_0_pre5 release
- Used 7 different physics processes ("candles"):
    1. HiggsZZ4LM190
    2. MinBias
    3. QCD_80_120
    4. SingleElectronE1000
    5. SingleMuMinusPt10
    6. SinglePiMinusE1000
    7. TTbar

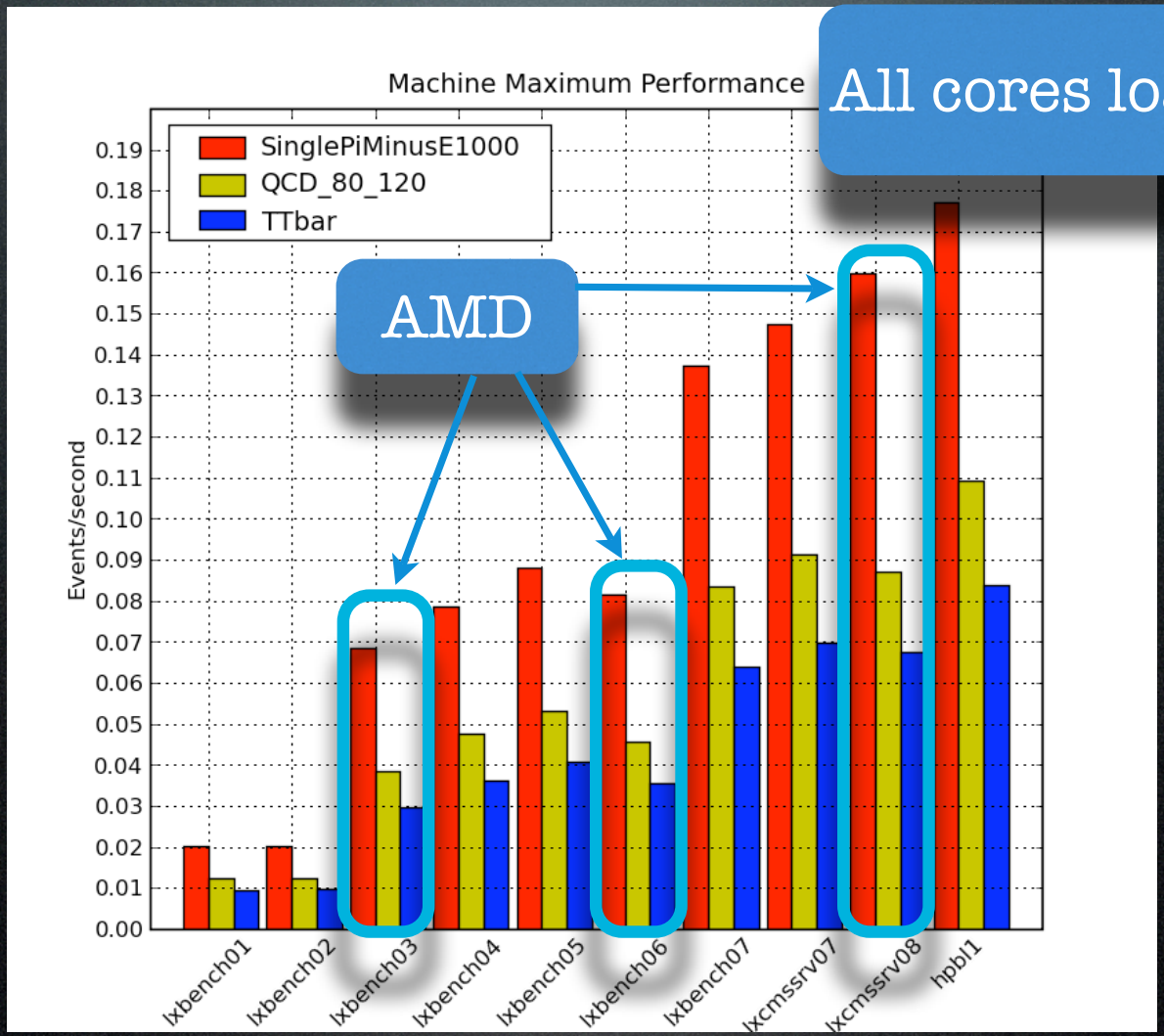
CMSSW Benchmarking

- Run 100 events per candle
- Run GEN+SIM, DIGI, RECO steps separately
- Run the 7 candles sequentially on each core
- Four tests ran up to now:
  - Loading all cores simultaneously
  - Loading 1, 3 (only for 4 cores machines) and 5 cores (only for 8 cores machines) with our application while running a cpu-intensive, cache-contained benchmarking tool (cmsScimark2) on the other cores

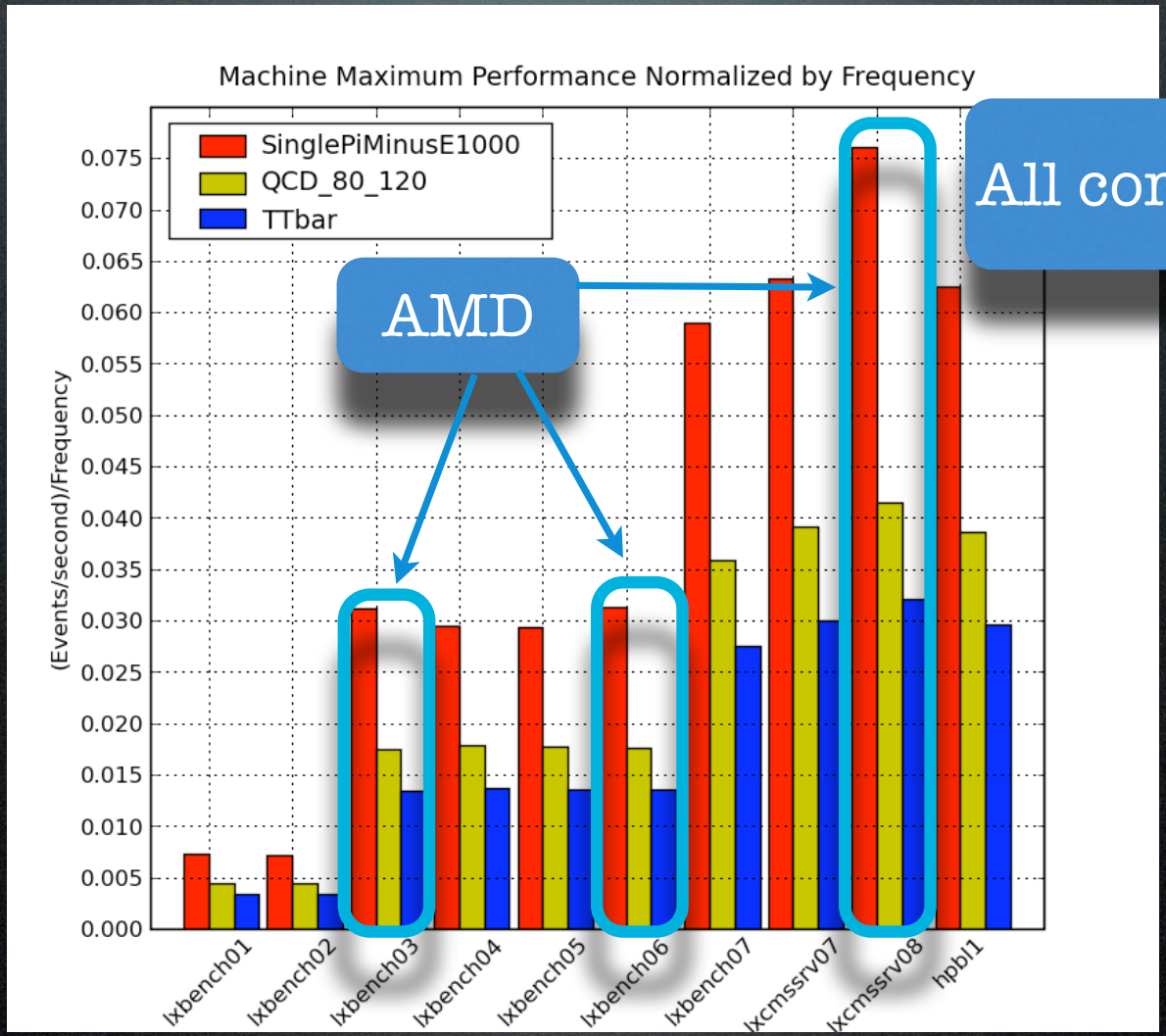
CMSSW Benchmarking

- The result of the benchmarking is seconds/event averaged on the 99 events (skipping the first one to avoid biases due to initialization)
- The results are reported in 3 formats:
  - seconds/events per core
  - events/seconds per core
  - events/seconds per machine
- Link: https://hepix.caspur.it/processors/dokuwiki/doku.php?id=benchmarks:cms

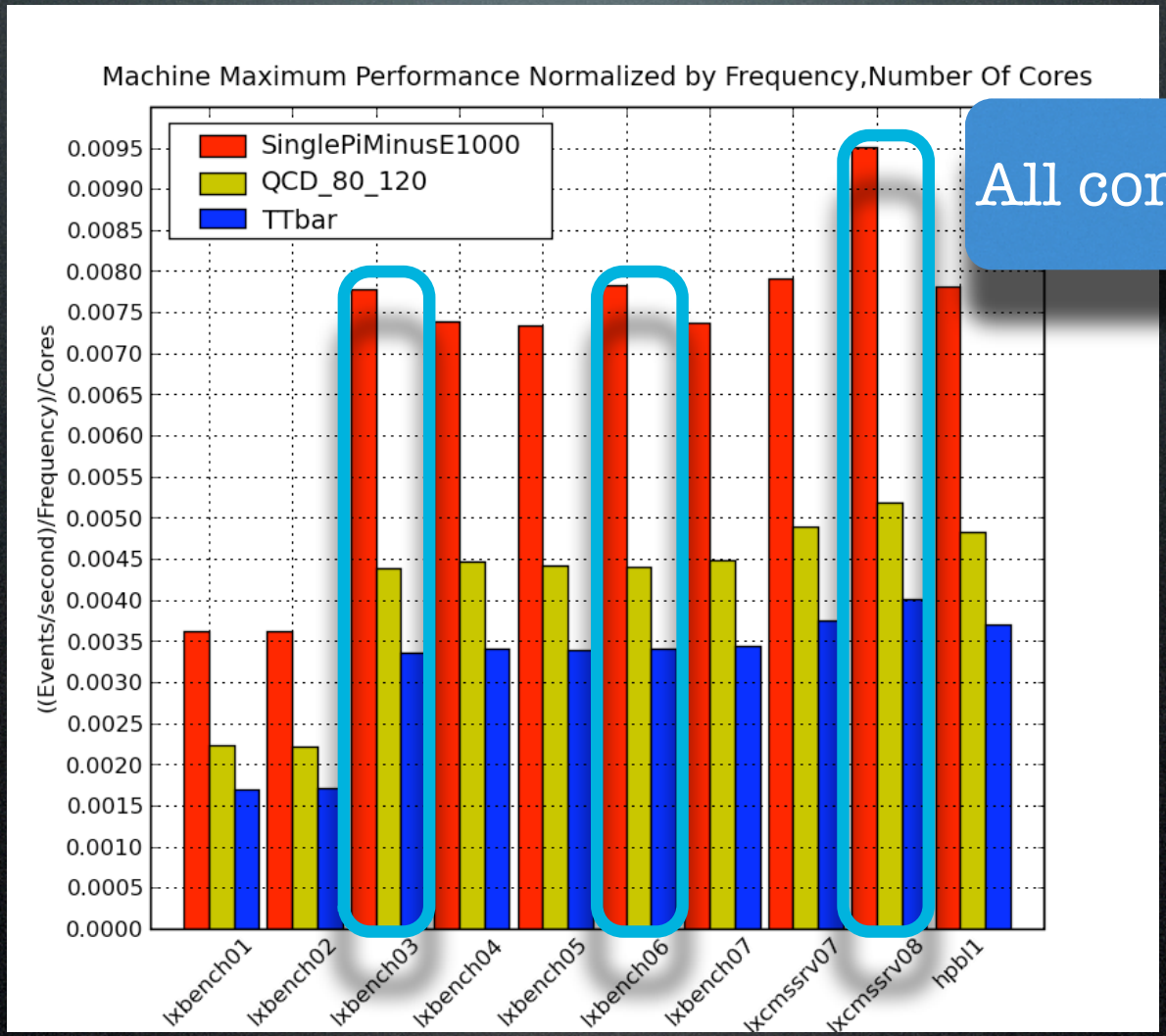
Comparing GEN+SIM



GEN+SIM normalized by frequency

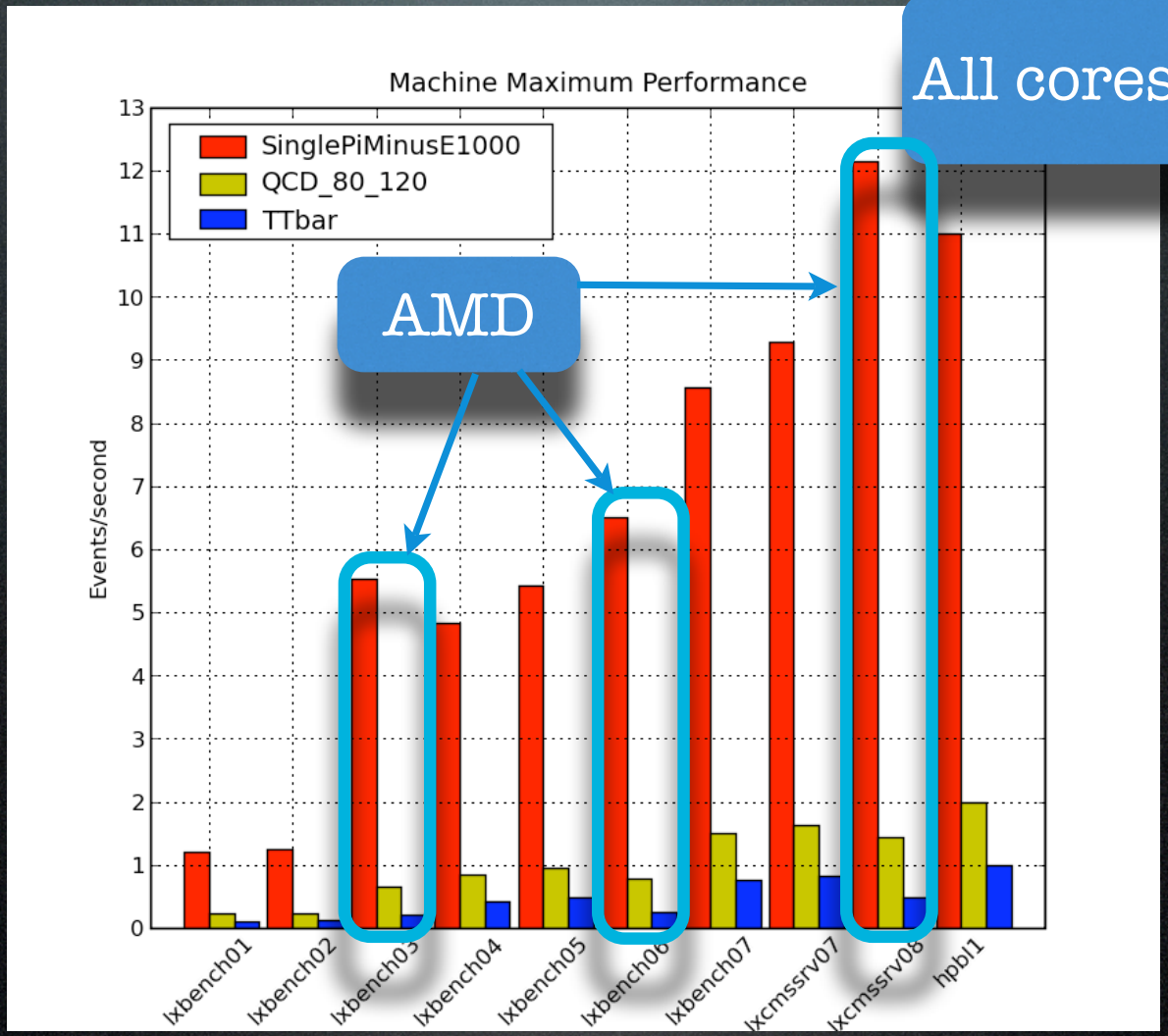


GEN+SIM normalized by frequency and # cores

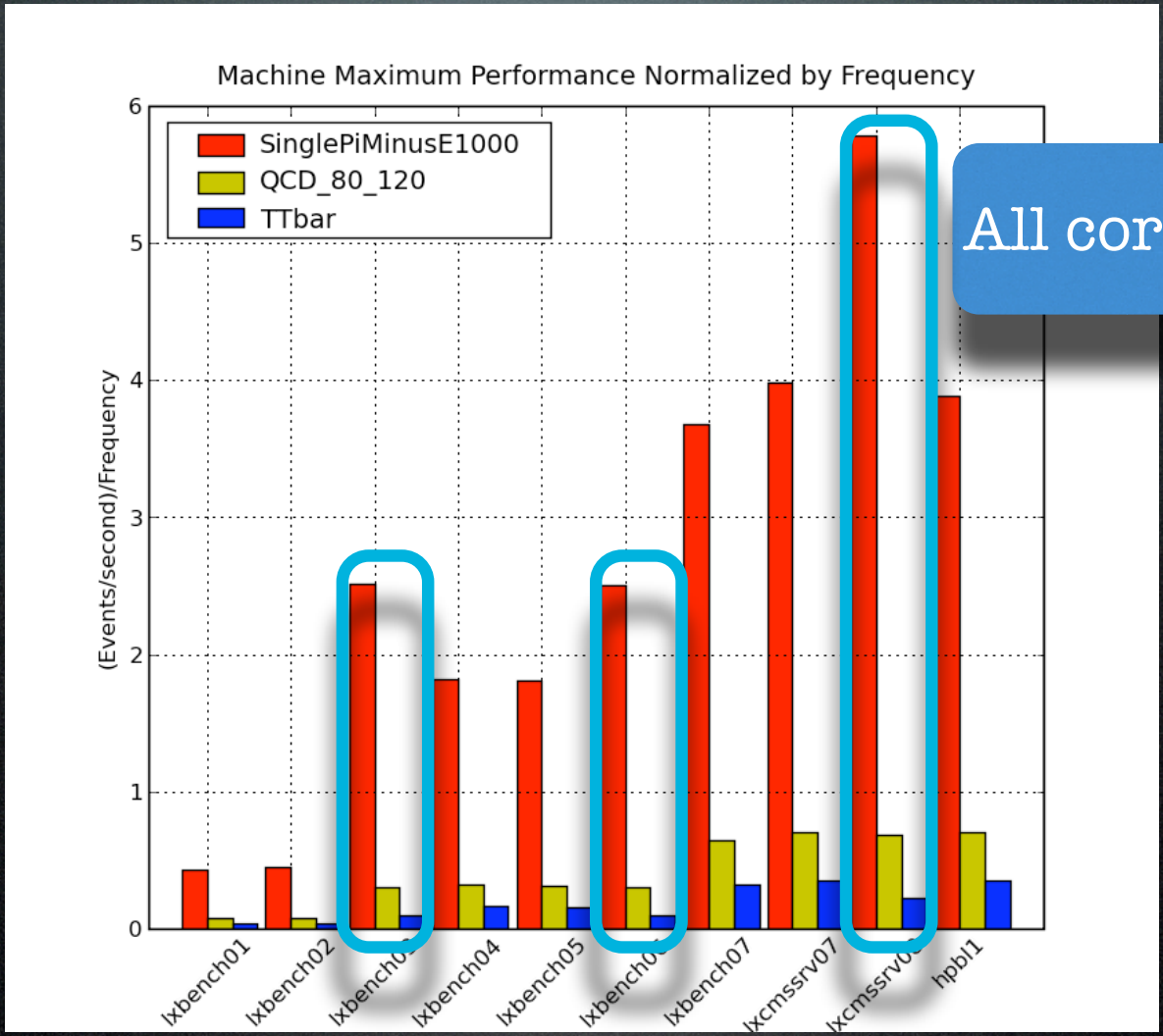


All cores loaded

Comparing RECO

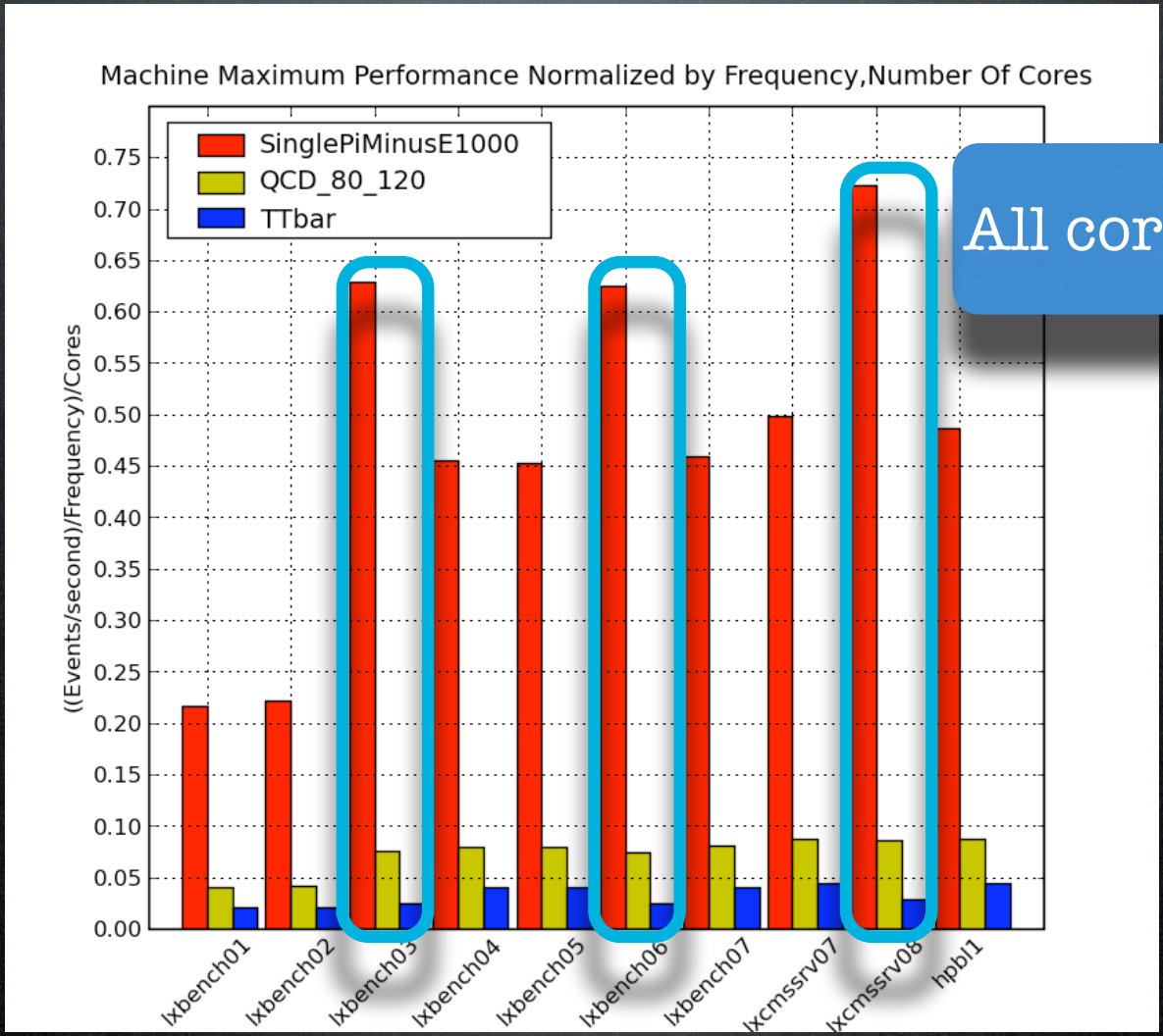


RECO normalized by frequency



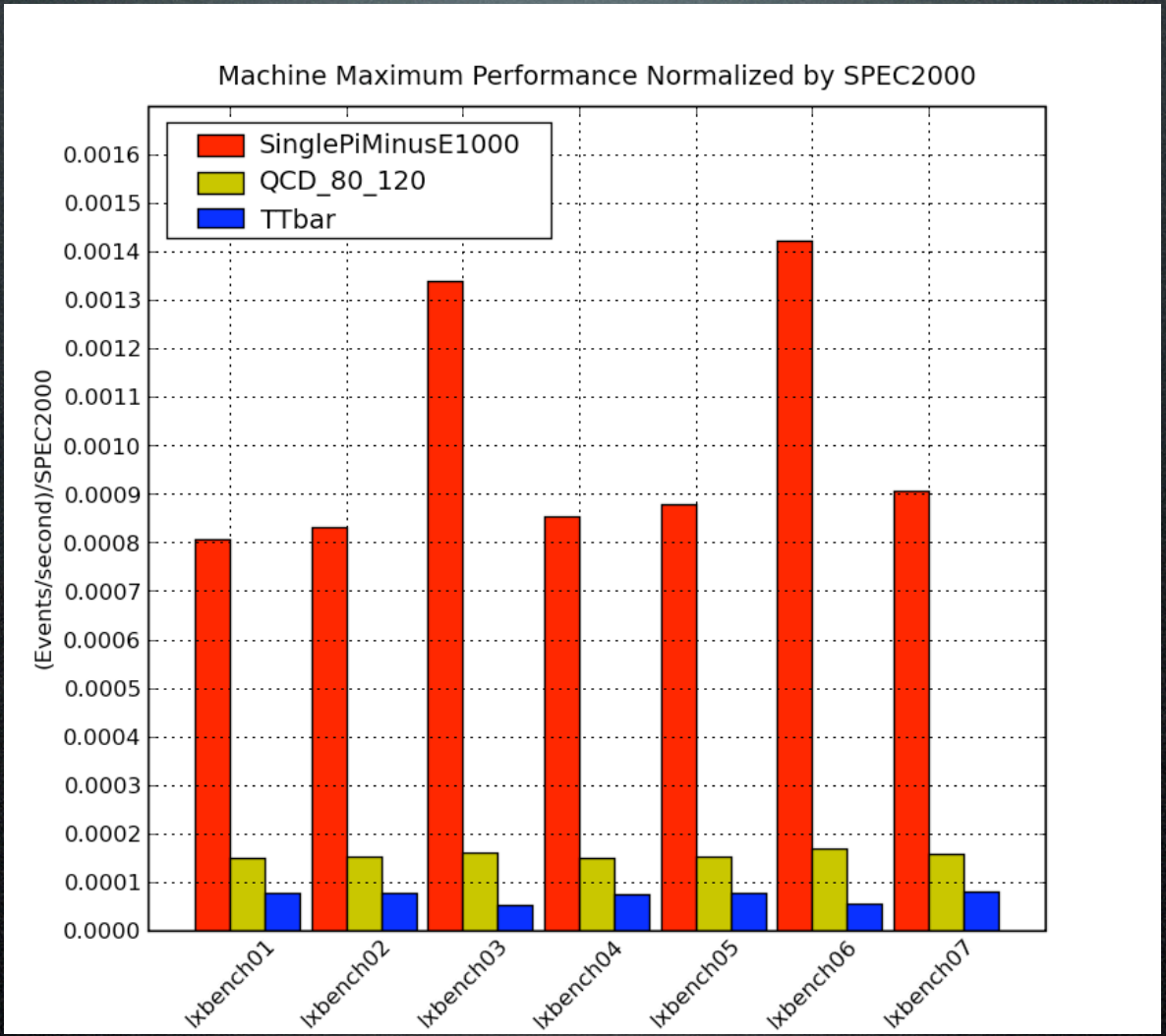
All cores loaded

RECO normalized by frequency and # cores

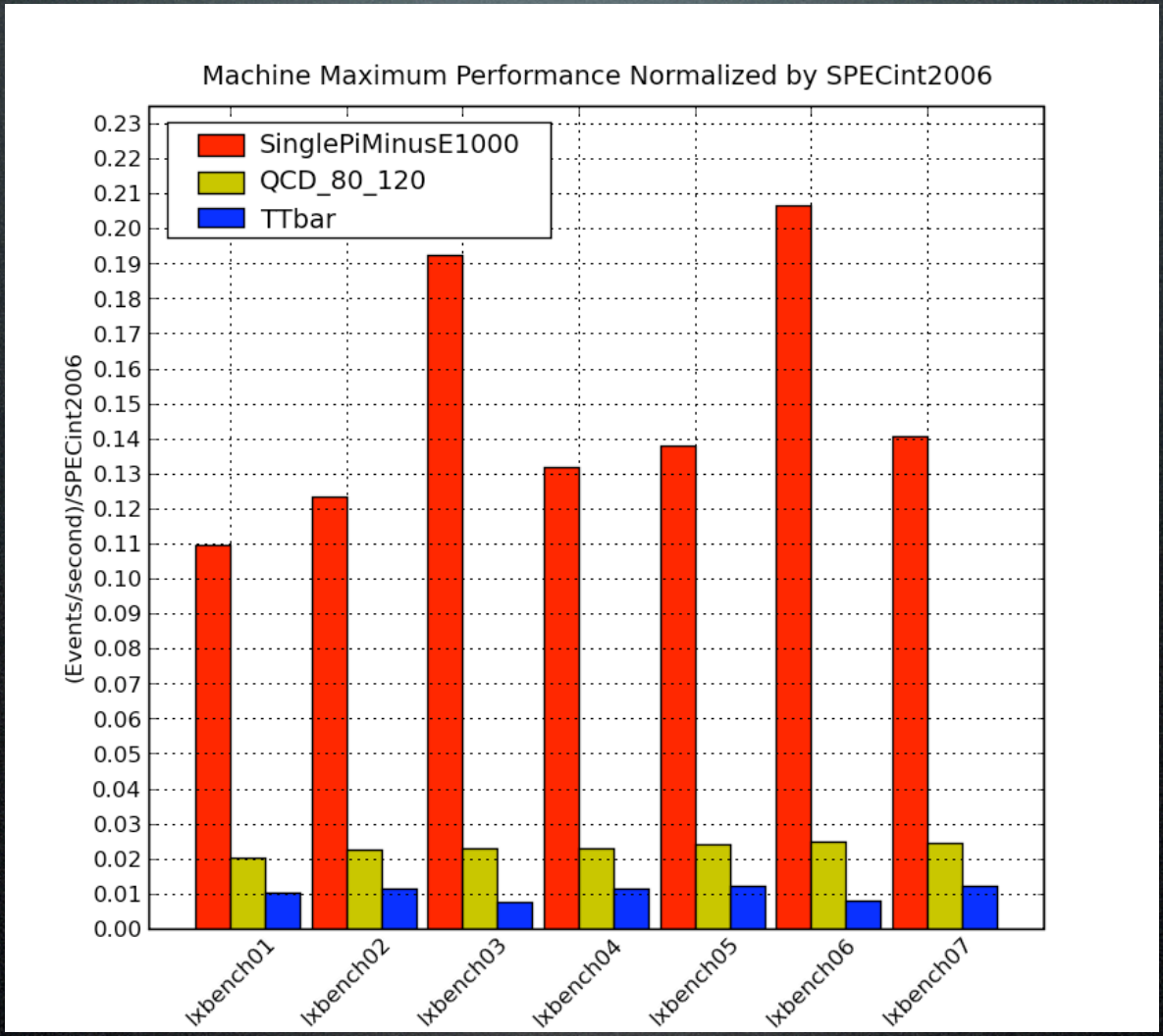


All cores loaded

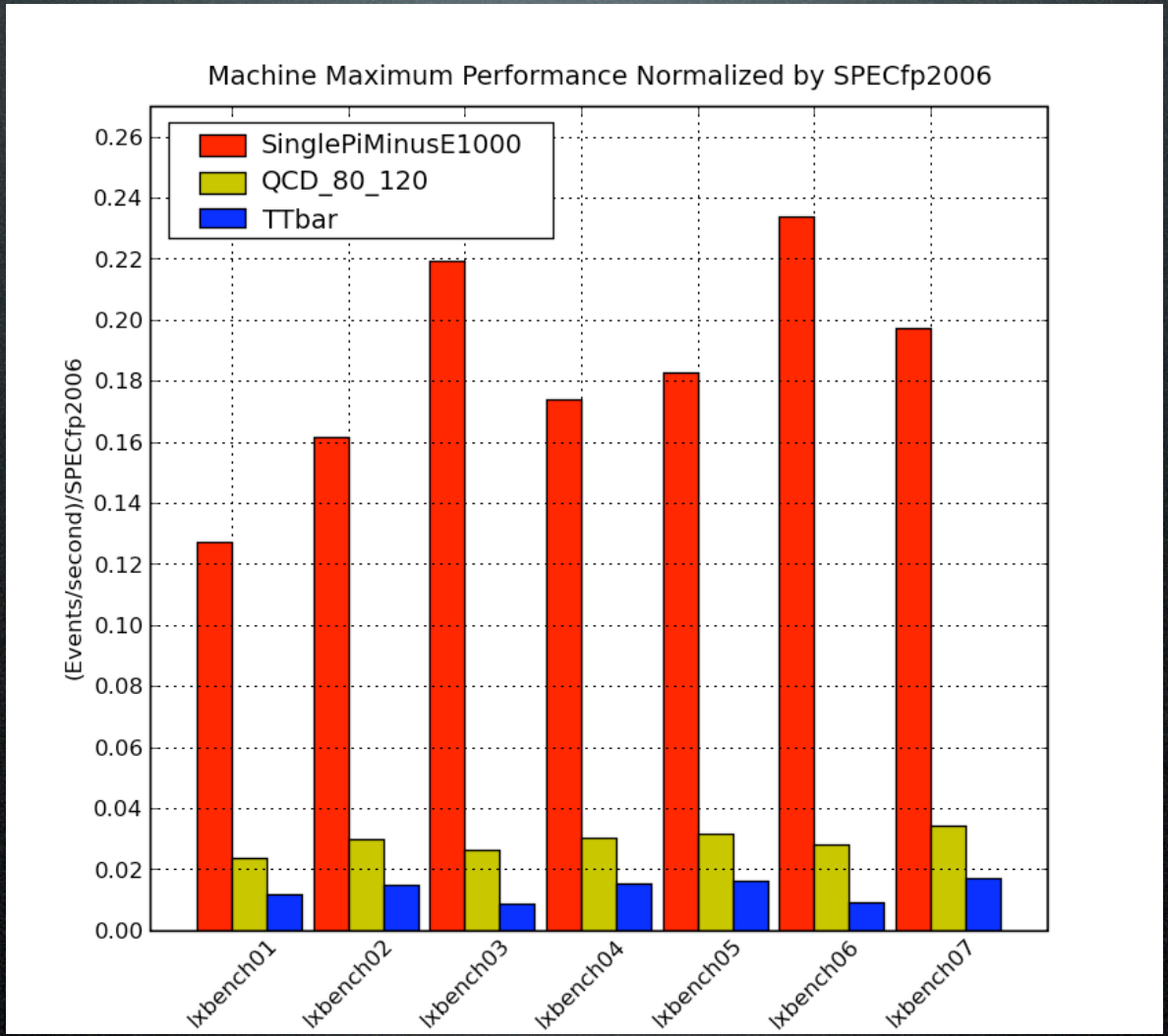
RECO vs SPEC2000



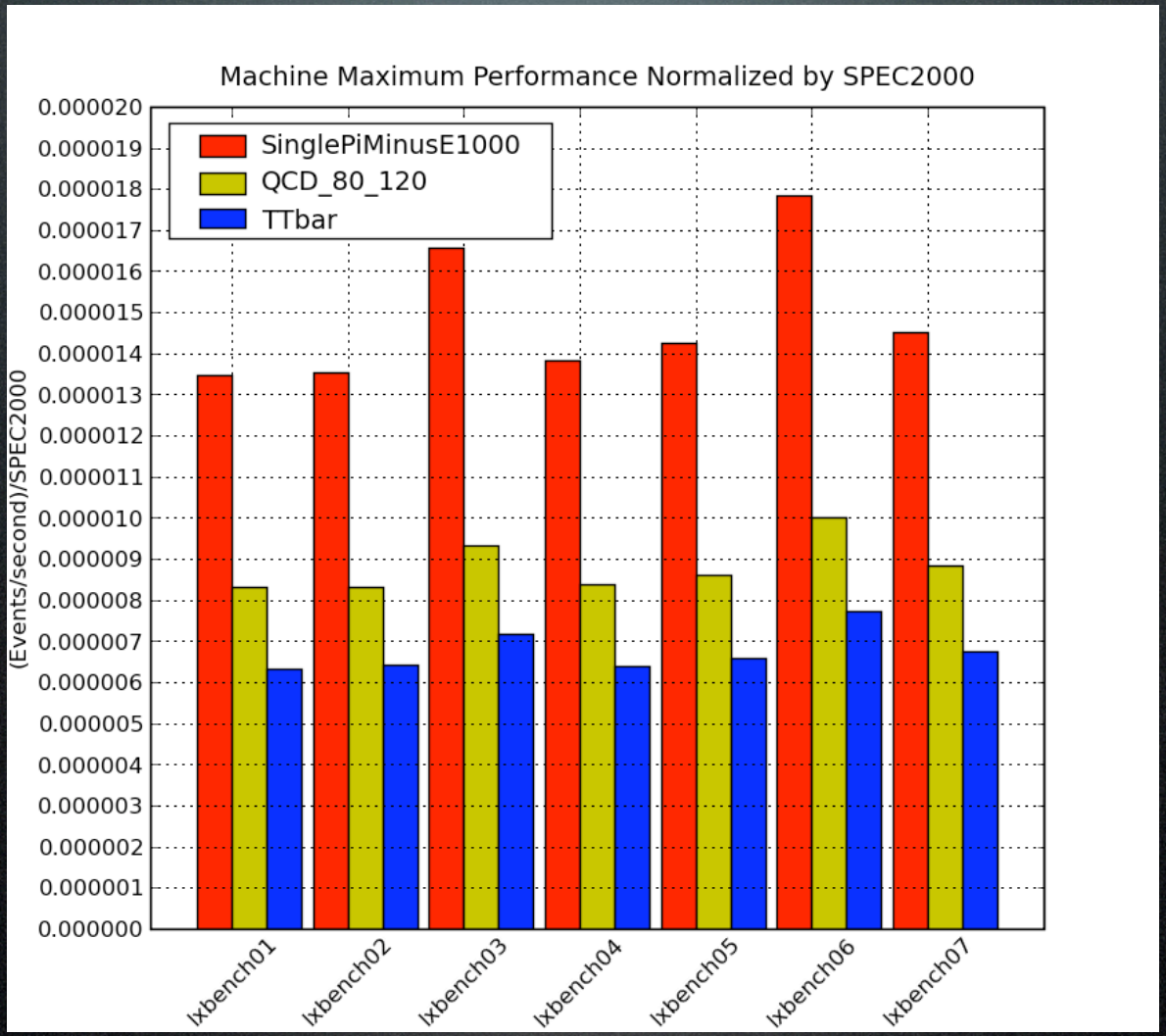
RECO vs SPECint2006



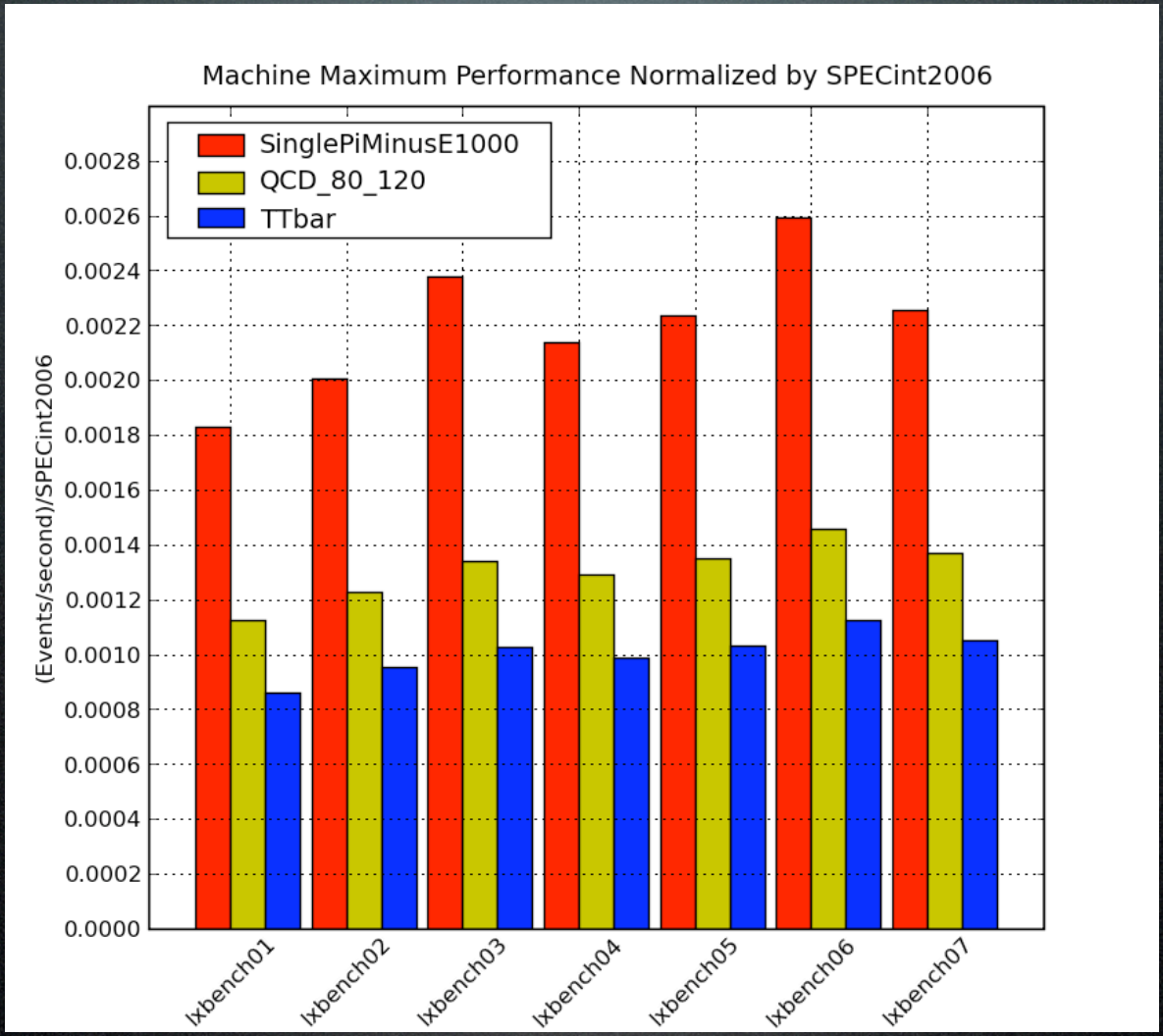
RECO vs SPECfp2006



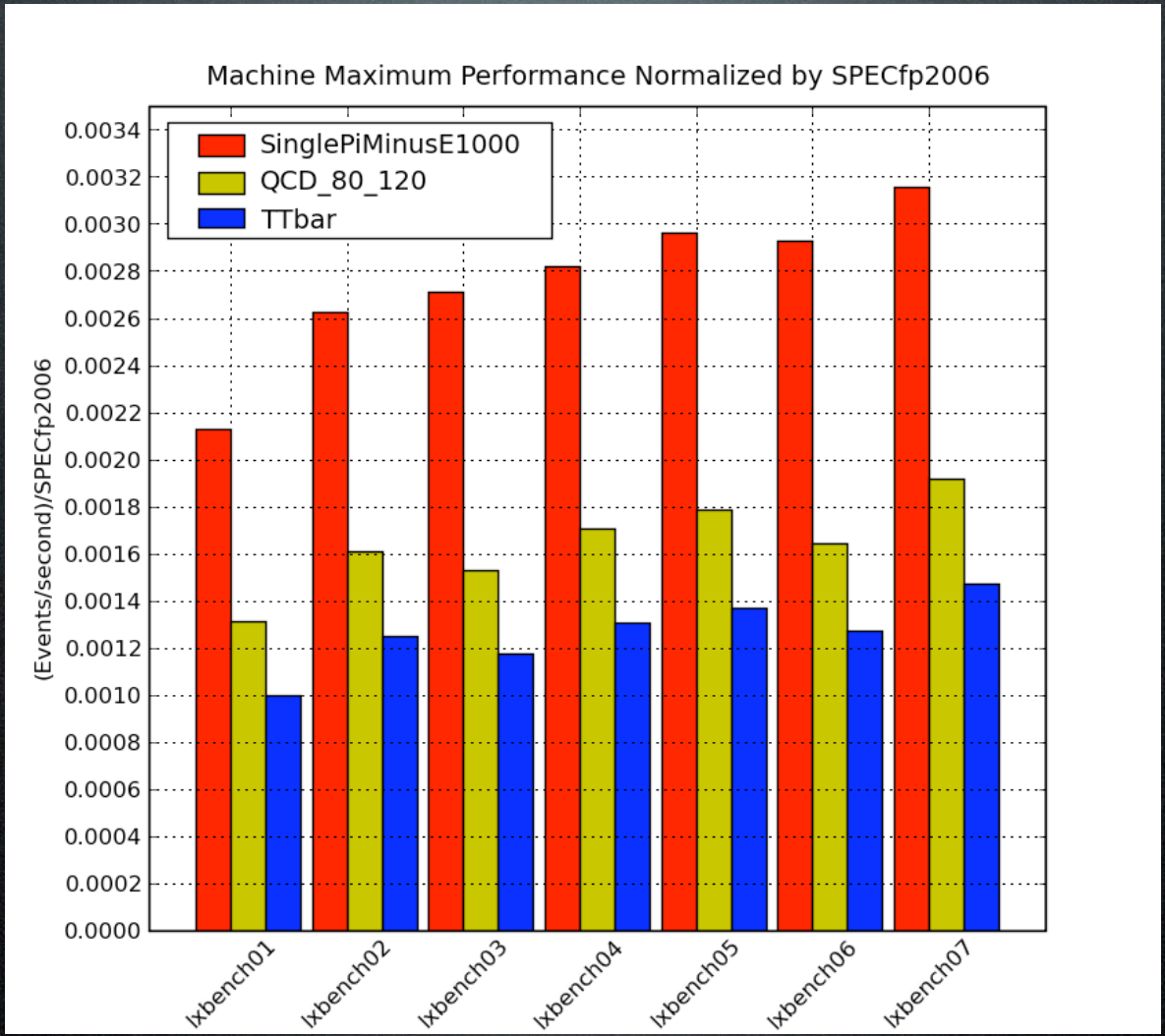
SIM vs SPEC2000



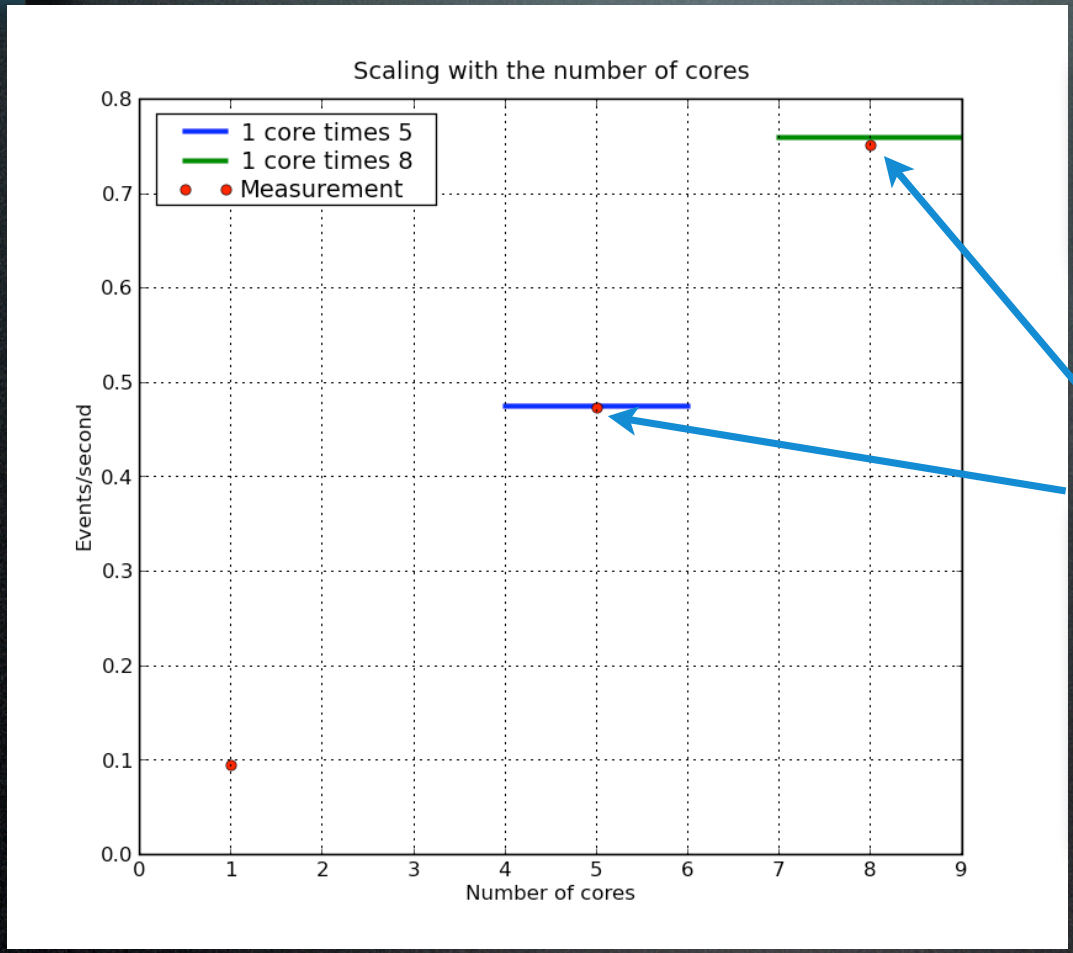
SIM vs SPECint2006



SIM vs SPECfp2006



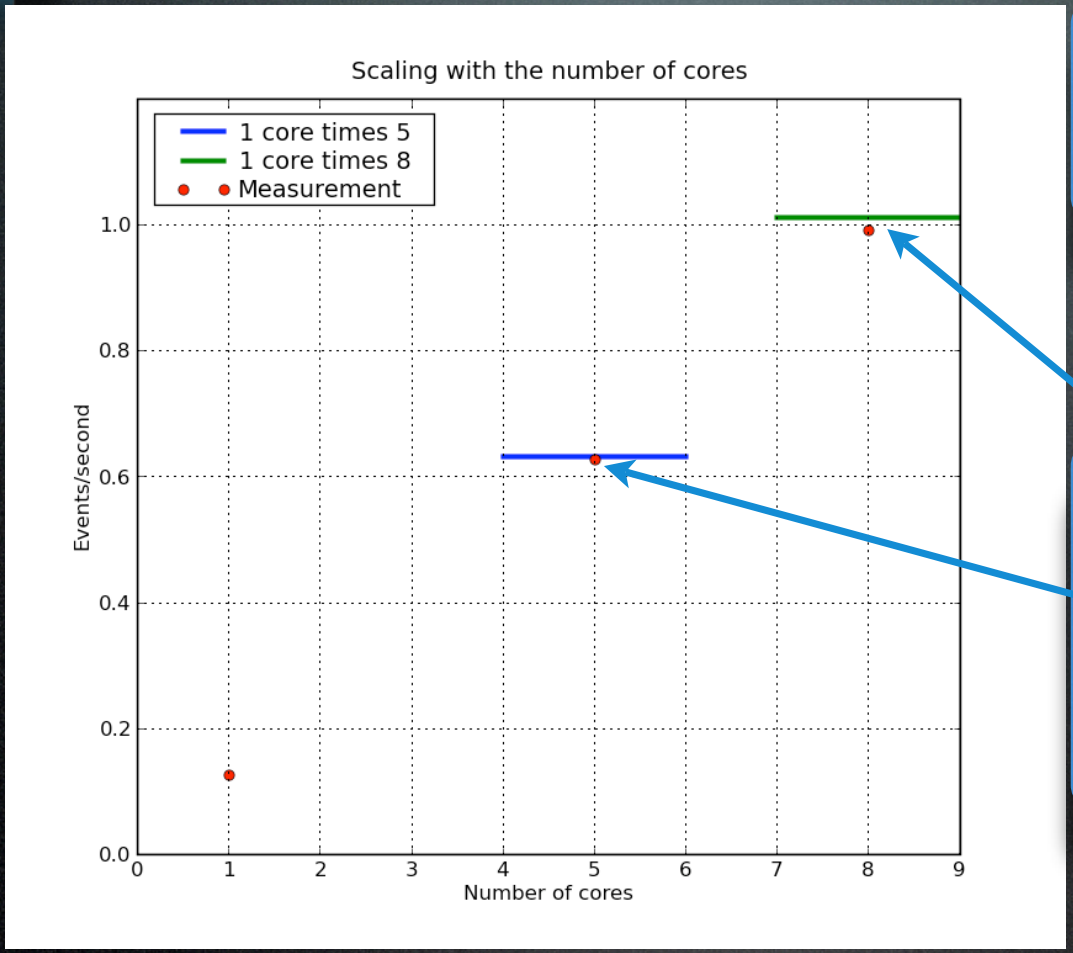
Scaling with multicores



lxbench07
TTbar RECO

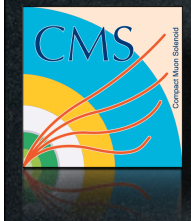
Excellent scaling to the last core (cmsScimark2)

Scaling with multicores



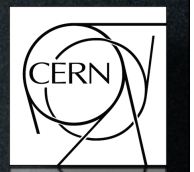
hpbl1
TTbar RECO

Excellent scaling to the last core (cmsScimark2)



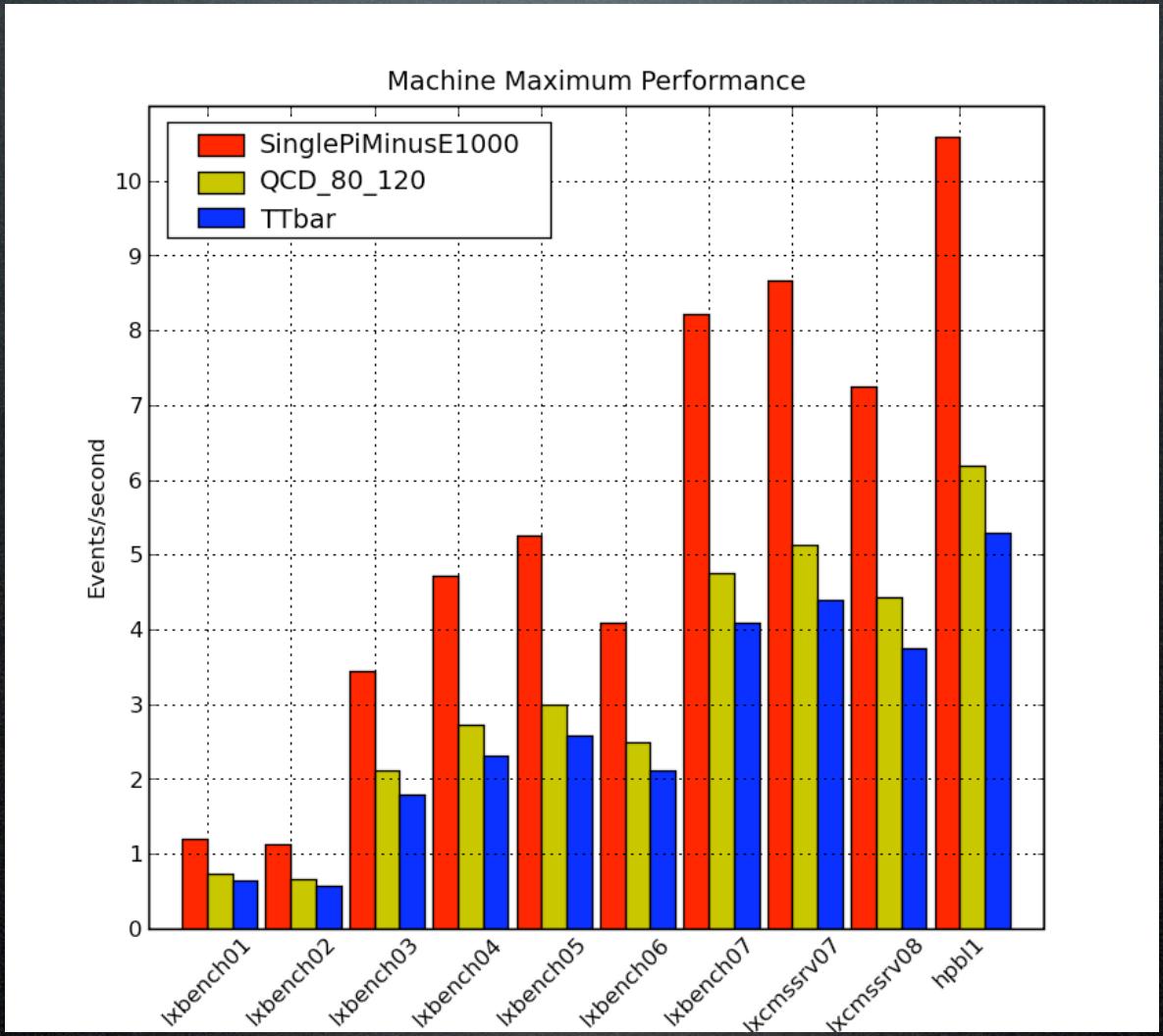
Conclusions

- Used CMSSW to benchmark 10 machines
- Observed a different behavior in AMD vs. Intel machines for complex vs. simple events at the RECO step
- Compared CMSSW applications with SPEC benchmarks: differences due to architecture/type of event are larger than the differences between different SPEC benchmarks
- The CMSSW application scales nicely with the current multicore architectures
- Work is ongoing (data analysis, more tests, developing a benchmarking suite to distribute)

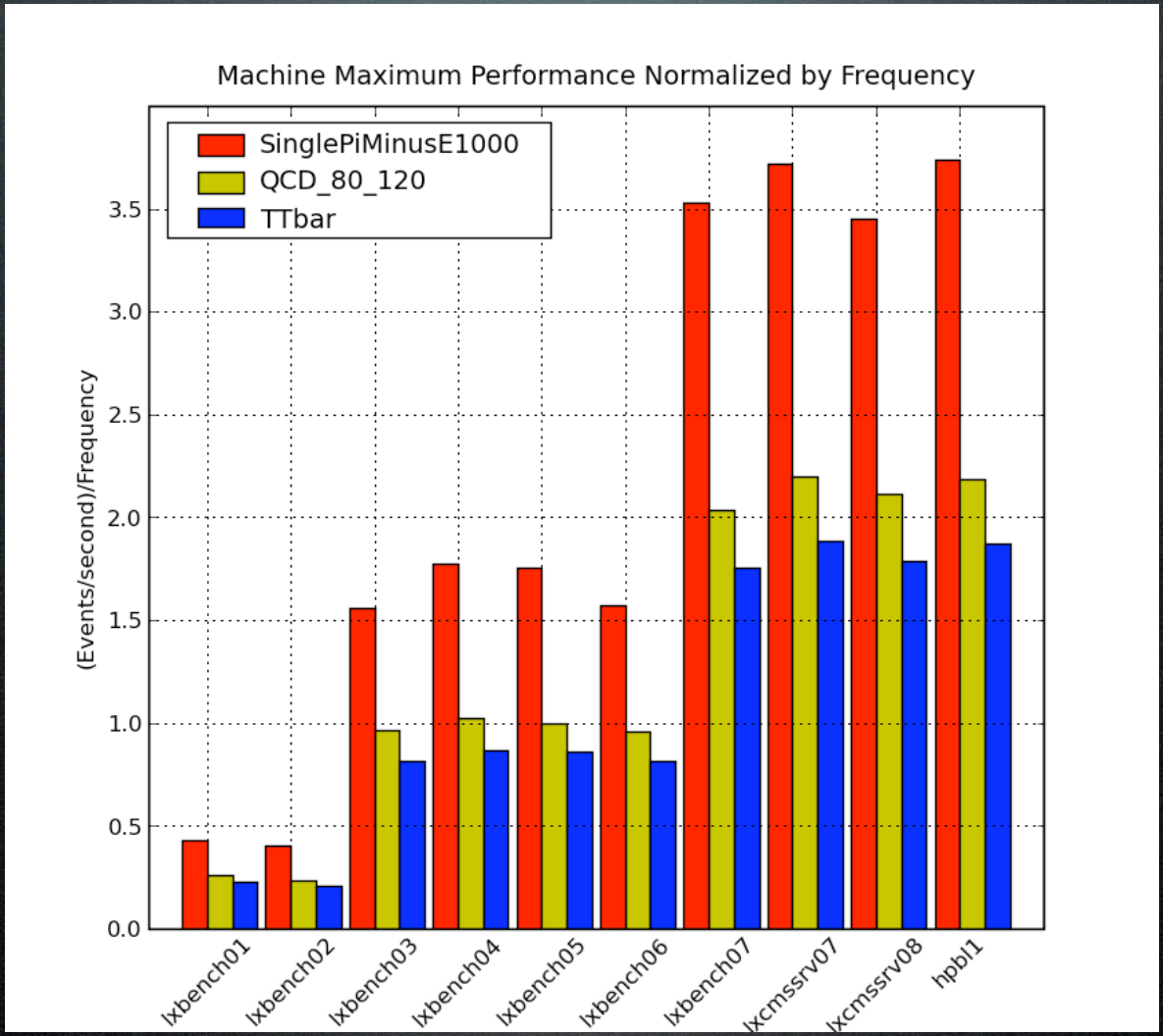


Back-up

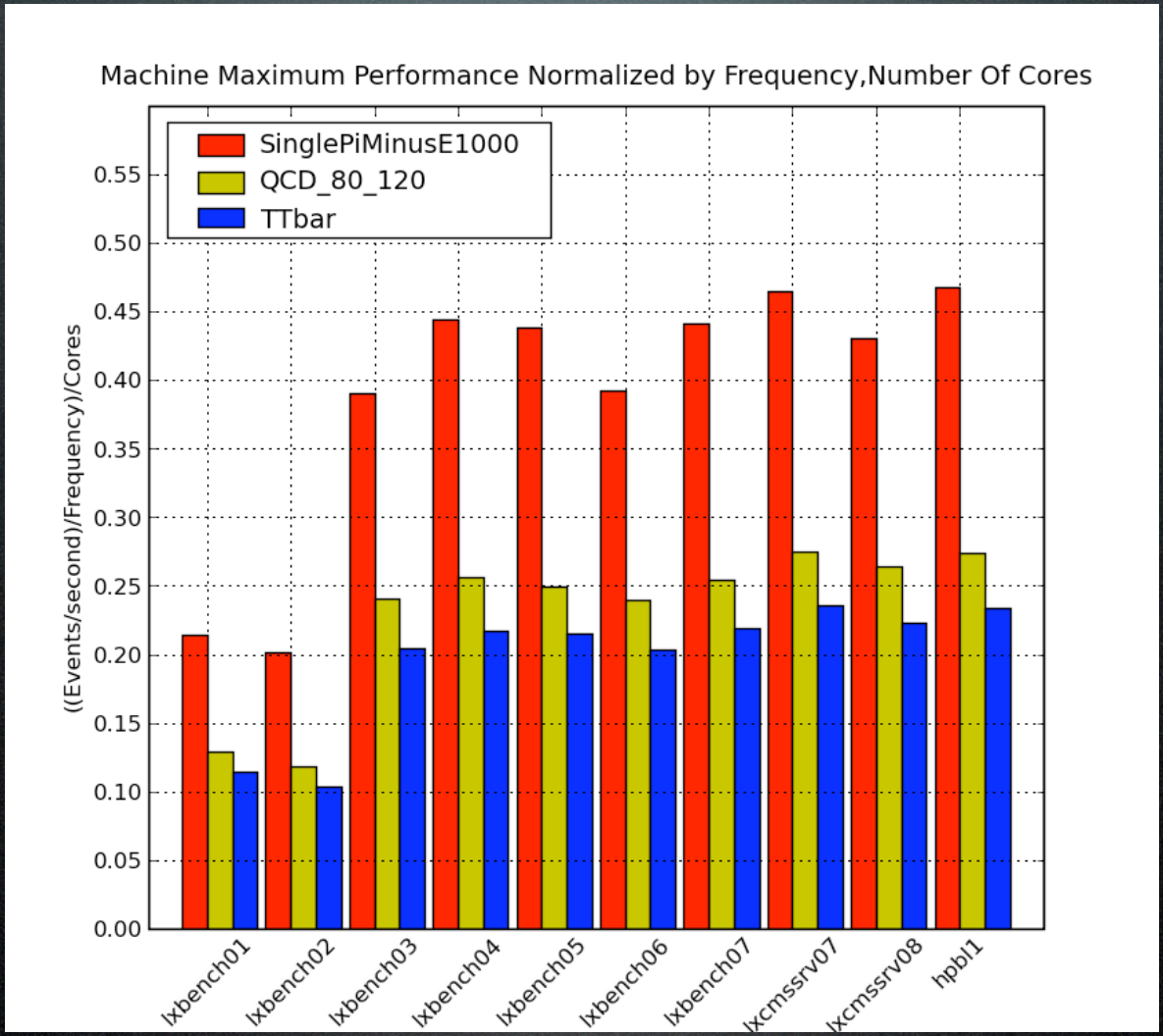
All Cores DIGI



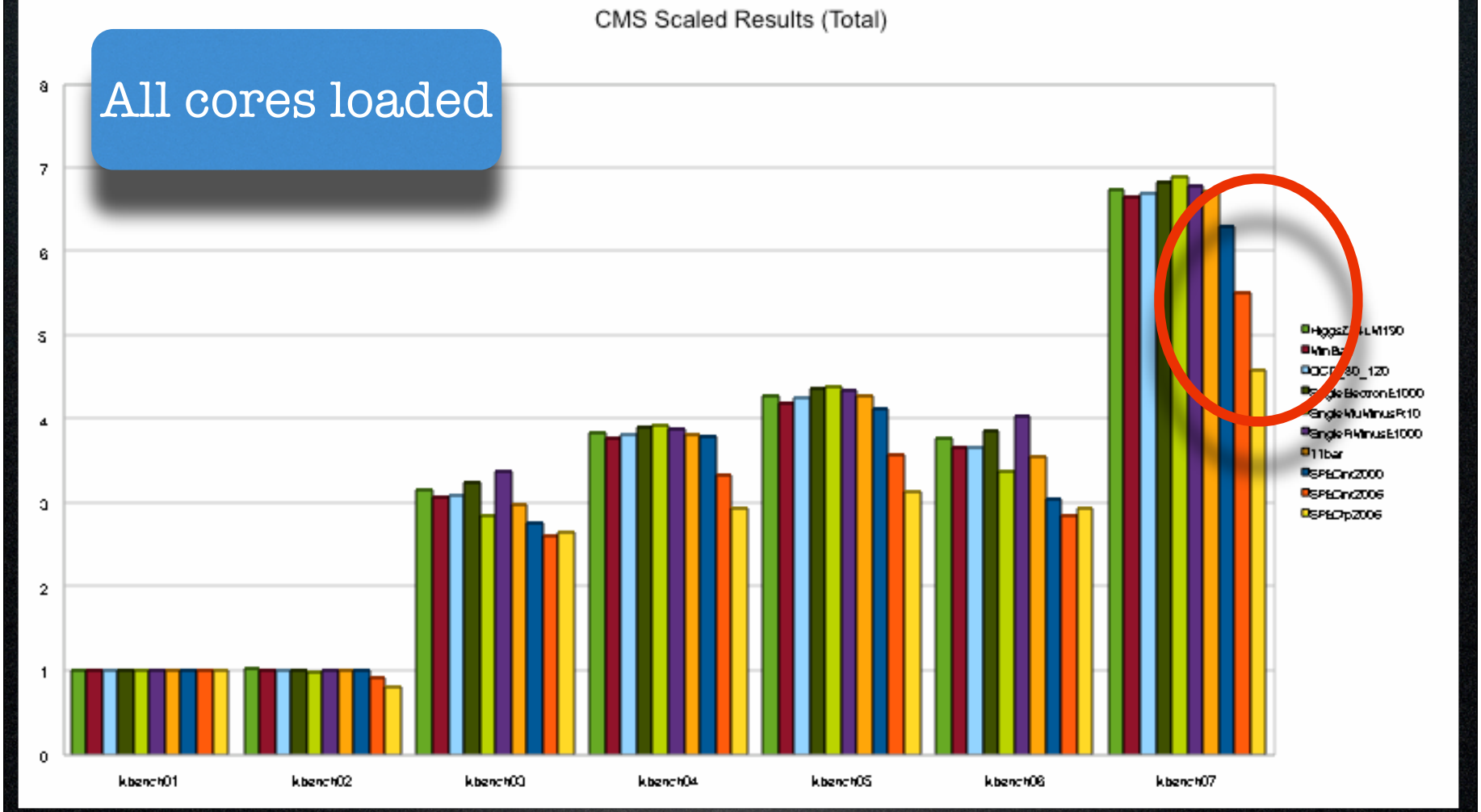
All Cores DIGI freq



All Cores SIM freq core



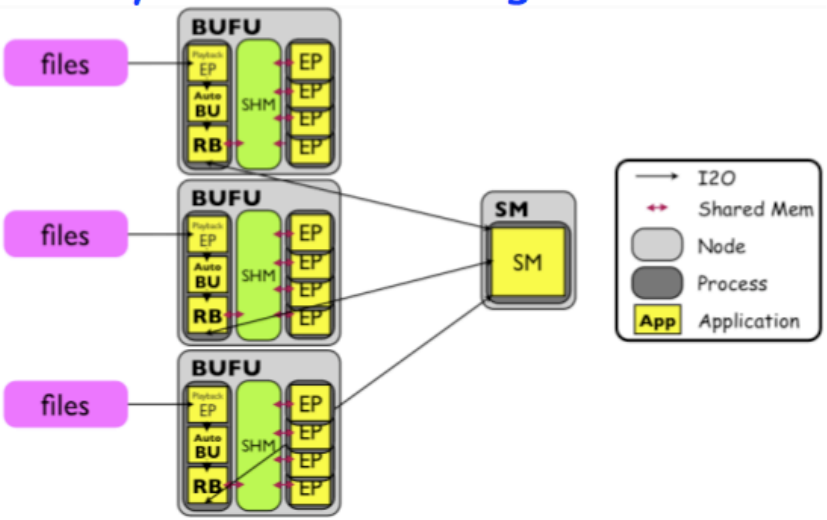
CMSSW vs SPEC



HLT benchmarking

Data Playback

- ✓ Emulation of High Level Trigger mode of operation
- ✓ Several processes (EP's) per node analyzing data as they were provided by the DAQ
- ✓ "Building" of the events performed in the same CPU
- ✓ Shared Memory used to exchange event data



Marco Zanetti CERN PH

HLT benchmarking

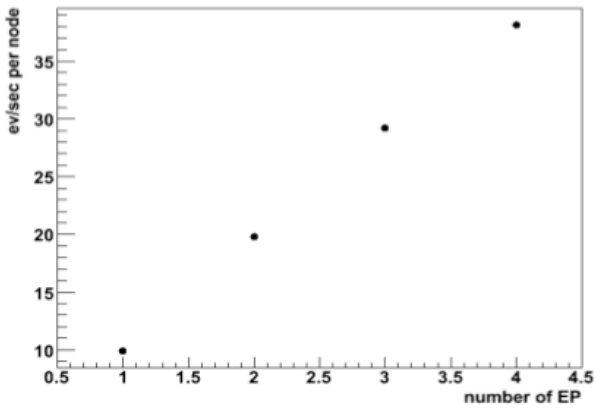
CMS HLT validation farm

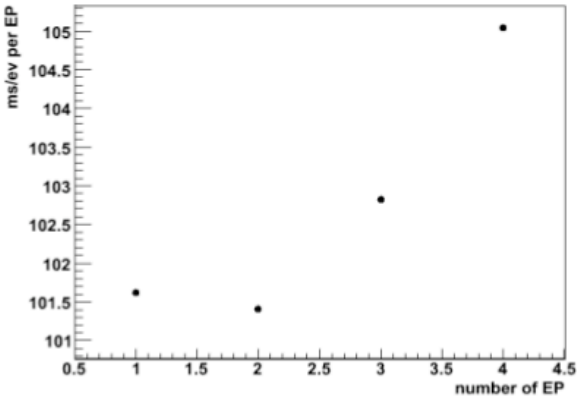
- ✓ 1 rack (20 PC's) of the CMS DAQ farm at LHC P5:
  - Dual dual-core
  - CPU: Intel Xeon 5130 @2.00 GHz, 4 MB L2 cache
  - 8 GB memory
- ✓ Up to 4 EventProcessor's (cmsRun equivalent) per node. 1000 events per node

Marco Zanetti CERN PH

HLT benchmarking

Results: HiLTon



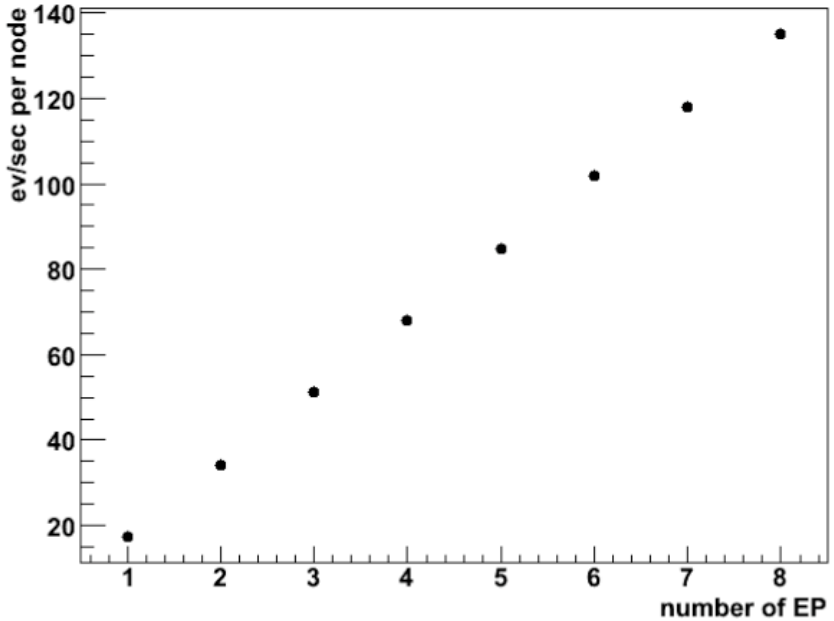


- ✓ Number of events processed per second per node scaled nicely with number of EP's
- ✓ Hilton machines have 4 cores, but even with 5 processes per node (4 EP + 1 BU) the EP's take ~100% of the CPU

Marco Zanetti CERN PH

HLT benchmarking

Results: Bench07

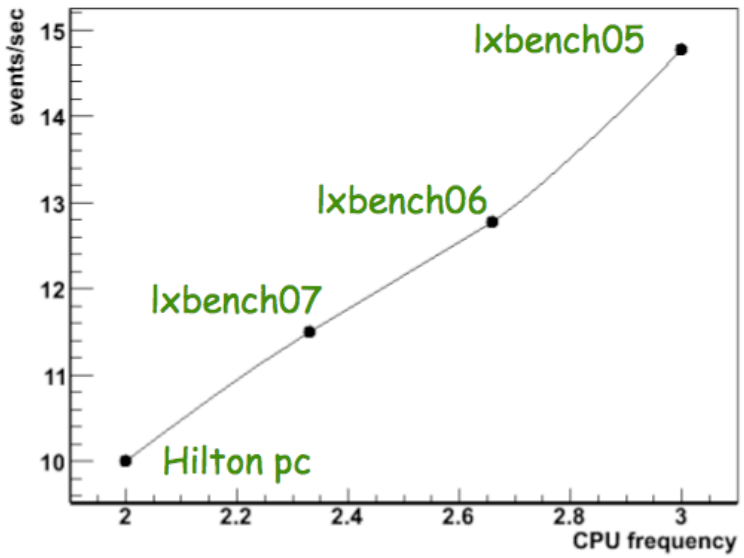


- ✓ Scaling valid up to 8 EP's on lxbench07 (dual quad-core)
- ✓ Better performances due to higher CPU frequency

Marco Zanetti CERN PH

HLT benchmarking

Scaling with frequency



- ✓ Nice scaling with CPU frequency (1 EP per node)
- ✓ (not the same configuration w.r.t what shown before, i.e. not same timing)

TTbar numbers

TTbar **per core** (when running on all cores at once)

	lxbench01	lxbench02	lxbench03	lxbench04	lxbench05	lxbench06	lxbench07
	2 cores	2 cores	4 cores	4 cores	4 cores	4 cores	8 cores
	s/evt	s/evt	s/evt	s/evt	s/evt	s/evt	s/evt
GEN+SIM	210.207	208.060	134.939	109.988	98.194	112.939	124.910
DIGI	3.361	3.688	2.408	1.869	1.674	2.047	2.112
RECO	17.987	17.739	18.482	9.610	8.570	15.782	10.906
TOTAL(SUM)	231.555	229.487	155.829	121.467	108.438	130.768	137.928

TTbar **per core** (when running on all cores at once)

	lxbench01	lxbench02	lxbench03	lxbench04	lxbench05	lxbench06	lxbench07
	2 cores	2 cores	4 cores	4 cores	4 cores	4 cores	8 cores
	evts/s	evts/s	evts/s	evts/s	evts/s	evts/s	evts/s
GEN+SIM	0.004757	0.004806	0.007411	0.009092	0.010184	0.008854	0.008006
DIGI	0.297530	0.271150	0.415282	0.535045	0.597372	0.488520	0.473485
RECO	0.055596	0.056373	0.054107	0.104058	0.116686	0.063363	0.091693
TOTAL(SUM)	0.004319	0.004358	0.006417	0.008233	0.009222	0.007647	0.007250

TTbar **per machine** (when running on all cores at once)

	lxbench01	lxbench02	lxbench03	lxbench04	lxbench05	lxbench06	lxbench07
	2 cores	2 cores	4 cores	4 cores	4 cores	4 cores	8 cores
	evts/s	evts/s	evts/s	evts/s	evts/s	evts/s	evts/s
GEN+SIM	0.009514	0.009612	0.029644	0.036368	0.040736	0.035416	0.064048
DIGI	0.595060	0.542300	1.661128	2.140180	2.389488	1.954080	3.787880
RECO	0.111192	0.112746	0.216428	0.416232	0.466744	0.253452	0.733544
TOTAL(SUM)	0.008638	0.008716	0.025668	0.032932	0.036888	0.030588	0.058000

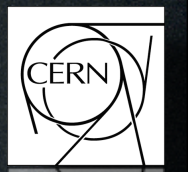
TTbar numbers

TTbar **per core** (when running on 1 core only, cpu1, while running cmsScimark on the other cores)

	lxbench01	lxbench02	lxbench03	lxbench04	lxbench05	lxbench06	lxbench07
	2 cores	2 cores	4 cores	4 cores	4 cores	4 cores	8 cores
	s/evt	s/evt	s/evt	s/evt	s/evt	s/evt	s/evt
GEN+SIM	209.565	210.290	134.687	110.259	98.406	112.645	122.560
DIGI	3.335	3.673	2.468	1.798	1.622	2.098	2.021
RECO	17.782	17.653	18.477	9.488	8.460	15.766	10.794
TOTAL(SUM)	230.682	231.616	155.632	121.545	108.488	130.509	135.375

Events/Second per core figure for candle TTbar:

GEN+SIM	0.004772	0.004755	0.007425	0.009070	0.010162	0.008877	0.008159
DIGI	0.299850	0.272257	0.405186	0.556174	0.616523	0.476644	0.494805
RECO	0.056237	0.056648	0.054121	0.105396	0.118203	0.063428	0.092644
TOTAL(SUM)	0.004335	0.004317	0.006425	0.008227	0.009218	0.007662	0.007387



lxbench01	lxbench02	lxbench03	lxbench04	lxbench05	lxbench06	lxbench07	lxcmssrv07	lxcmssrv08	hpbl1
2	2	4	4	4	4	8	8	8	8
2.8	2.8	2.2	2.66	3.0	2.6	2.33	2.33	2.1	2.83
1	2	2	4	4	2	8	12	2	12
2	4	2	8	8	8	16	16	16	16
Nocona	Irvingdale	Opteron 275	Woodcrest	Woodcrest	Opteron 2218 Rev.F	Clovertown	Xeon HarperTown E5410	Opteron Barcelona 2352	Xeon HarperTown E5440
Intel	Intel	AMD	Intel	Intel	AMD	Intel	Intel	AMD	Intel