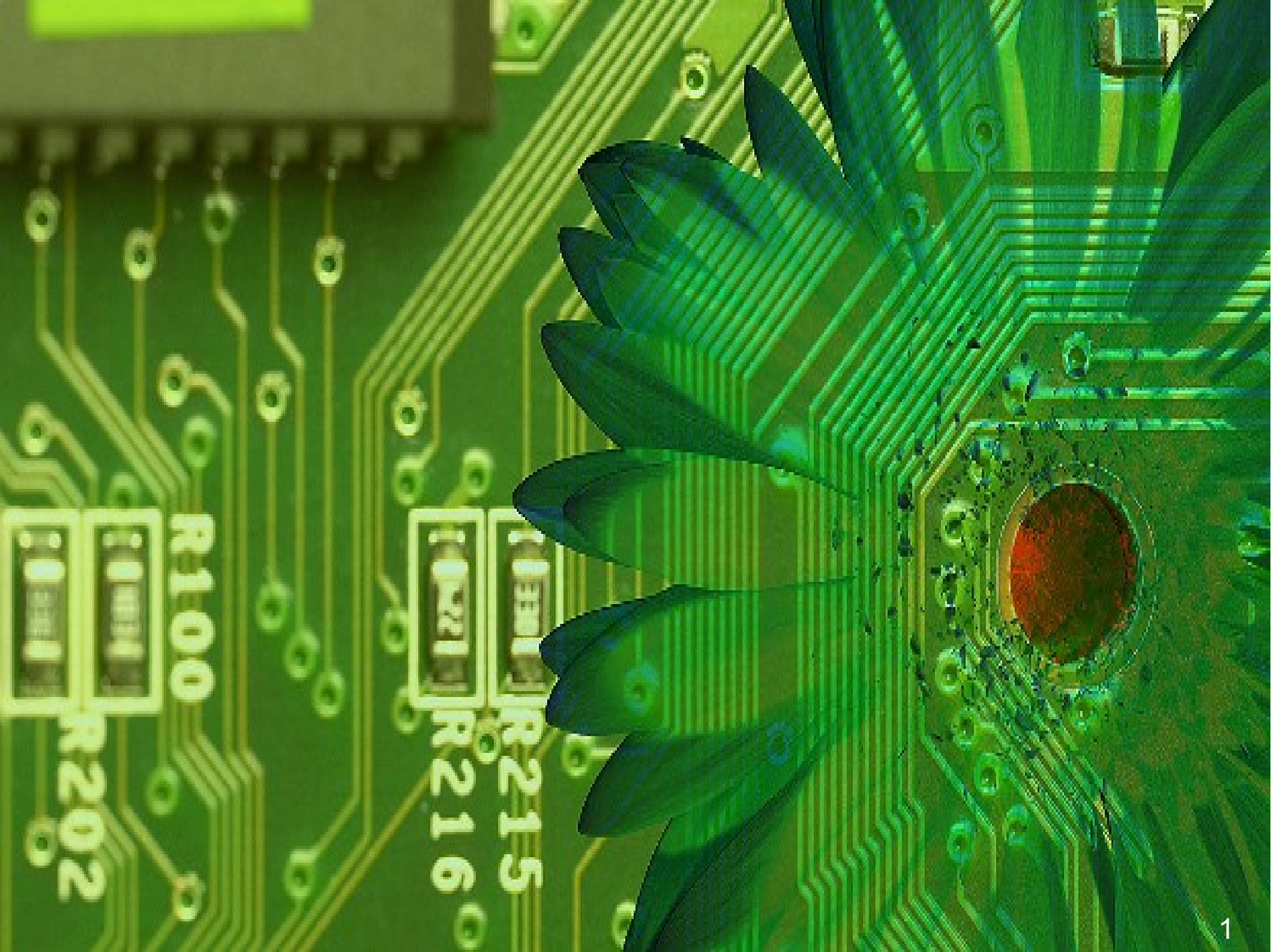
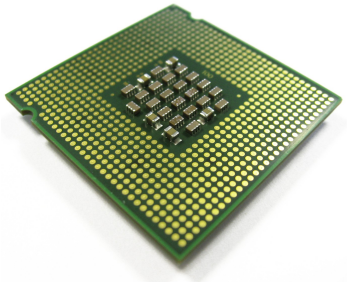
Reducing the cost of power monitoring with DC wattmeters

M. Asunción Castaño, Sandra Catalán, Rafael Mayo,
Enrique S. Quintana-Ortí

UNIVERSITAT JAUME·I

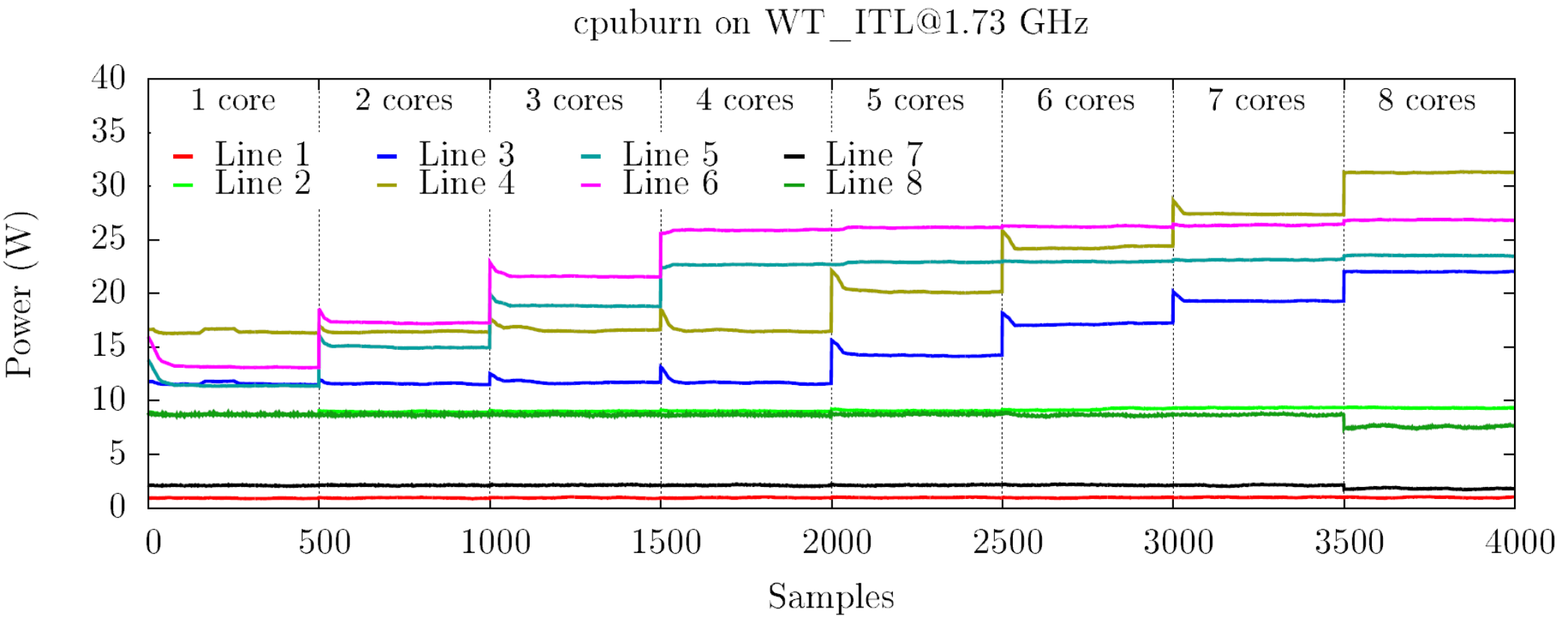

R100
R202
R219
R215


UNDERSTANDING



Target

Is it possible to ***estimate the total instantaneous power*** by using just a ***few lines***?


cpuburn on WT_ITL@1.73 GHz
1 core
2 cores
3 cores
4 cores
5 cores
6 cores
7 cores
8 cores
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6
Line 7
Line 8
Power (W)
Samples
0
5
10
15
20
25
30
35
40
0
500
1000
1500
2000
2500
3000
3500
4000

$$p_T(t) = p_1(t) + p_2(t) + \cdots + p_l(t)$$

$$p_T(t) = p_1(t) + p_2(t) + \cdots + p_l(t)$$

$$\hat{p}_T(t) = w_1 \cdot p_{l_1}(t) + w_2 \cdot p_{l_2}(t) + \cdots + w_r \cdot p_{l_r}(t)$$

$$r \ll l$$

$$p_T(t) = p_1(t) + p_2(t) + \cdots + p_l(t)$$

$$\hat{p}_T(t) = w_1 \cdot p_{l_1}(t) + w_2 \cdot p_{l_2}(t) + \cdots + w_r \cdot p_{l_r}(t)$$

$$p_T(t) = p_1(t) + p_2(t) + \cdots + p_l(t)$$

$$\hat{p}_T(t) = w_1 \cdot p_{l_1}(t) + w_2 \cdot p_{l_2}(t) + \cdots + w_r \cdot p_{l_r}(t)$$

Calibrating the reduced power model

Experimentation

Benchmarks

cpuburn

stream

hdparm

Experimentation

Frequencies

Benchmarks

cpuburn

stream

hdparm

Experimentation

Frequencies

Benchmarks

cpuburn

stream

hdparm

Experimentation

Number of cores

Frequencies

Benchmarks

cpuburn

stream

hdparm

Experimentation

AMD

Multicore platforms

ITL

Number of cores

Frequencies

Benchmarks

cpuburn

stream

hdparm

Experimentation

AMD

Multicore platforms

Number of cores

ITL

12V

Collection of power samples

benchmark	freq	# cores	P_1	P_2	P_3	. . .	P_I	P_T

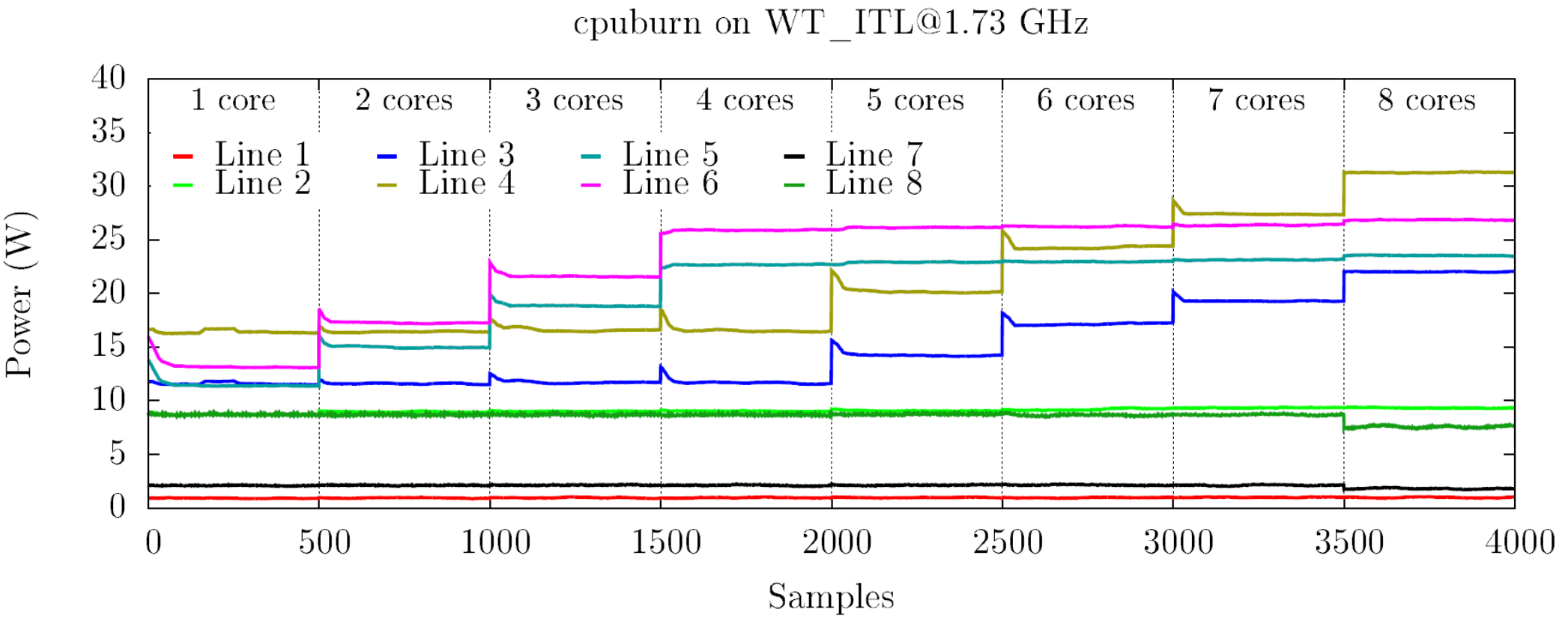
≈ 4,000,000 samples
72 minutes

Computation of the correlation matrix

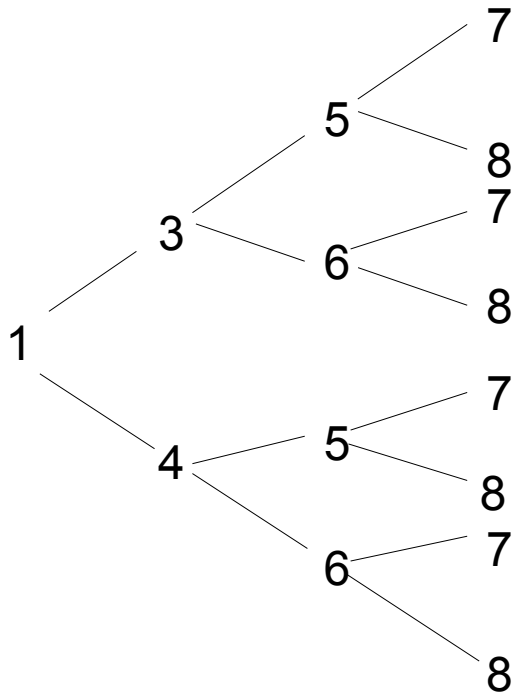
Line	1	2	3	4	5	6	7	8
1	1.00	0.92	0.20	0.20	0.30	0.30	0.58	0.59
2	0.92	1.00	0.28	0.28	0.35	0.36	0.65	0.66
3	0.20	0.28	1.00	0.99	0.70	0.70	0.39	0.40
4	0.20	0.28	0.99	1.00	0.70	0.70	0.39	0.40
5	0.30	0.35	0.70	0.70	1.00	0.99	0.39	0.39
6	0.30	0.36	0.70	0.70	0.99	1.00	0.39	0.39
7	0.58	0.65	0.39	0.39	0.39	0.39	1.00	0.98
8	0.59	0.66	0.40	0.40	0.39	0.39	0.98	1.00

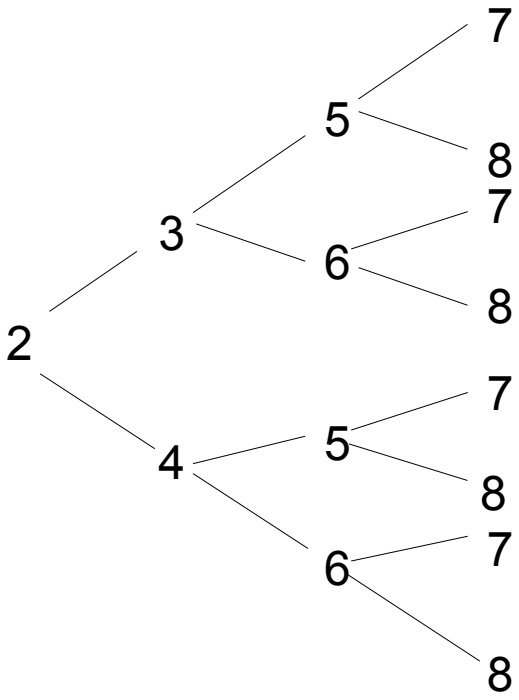
Classification of the *l* lines into *r* clusters

Line	1	2	3	4	5	6	7	8
1	1.00	0.92	0.20	0.20	0.30	0.30	0.58	0.59
2	0.92	1.00	0.28	0.28	0.35	0.36	0.65	0.66
3	0.20	0.28	1.00	0.99	0.70	0.70	0.39	0.40
4	0.20	0.28	0.99	1.00	0.70	0.70	0.39	0.40
5	0.30	0.35	0.70	0.70	1.00	0.99	0.39	0.39
6	0.30	0.36	0.70	0.70	0.99	1.00	0.39	0.39
7	0.58	0.65	0.39	0.39	0.39	0.39	1.00	0.98
8	0.59	0.66	0.40	0.40	0.39	0.39	0.98	1.00


cpuburn on WT_ITL@1.73 GHz
1 core
2 cores
3 cores
4 cores
5 cores
6 cores
7 cores
8 cores
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6
Line 7
Line 8
Power (W)
Samples
0
5
10
15
20
25
30
35
40
0
500
1000
1500
2000
2500
3000
3500
4000

Selection of the representative lines





c_T possible combinations of representative lines

Selection of the representative lines

$$\min_{\{c_k, w_k\}} \|A_k w_k - p\|_2$$

$$c_k = (l_1^k, l_2^k, \ldots, l_r^k),\ k = 1, 2, \ldots, c_T$$

$$p_T(t) = p_1(t) + p_2(t) + \cdots + p_l(t)$$

$$\hat{p}_T(t) = w_1 \cdot p_{l_1}(t) + w_2 \cdot p_{l_2}(t) + \cdots + w_r \cdot p_{l_r}(t)$$

$$p_T(t) = p_1(t) + p_2(t) + \cdots + p_l(t)$$

$$\hat{p}_T(t) = w_1 \cdot p_{l_1}(t) + w_2 \cdot p_{l_2}(t) + \cdots + w_r \cdot p_{l_r}(t)$$

Selection of the representative lines

$$\mathcal{R}(\{c_k, w_k\}) = \frac{\|A_k w_k - p\|_2}{\|p\|_2}$$

$$p_T(t) = p_1(t) + p_2(t) + \cdots + p_l(t)$$

$$\hat{p}_T(t) = w_1 \cdot p_{l_1}(t) + w_2 \cdot p_{l_2}(t) + \cdots + w_r \cdot p_{l_r}(t)$$

k	c_k	$\mathcal{R}(w_k)$	k	c_k	$\mathcal{R}(w_k)$
1	(1,3,5,7)	0.029	2	(1,3,5,8)	0.023
3	(1,3,6,7)	0.029	4	(1,3,6,8)	0.023
5	(1,4,5,7)	0.029	6	(1,4,5,8)	0.023
7	(1,4,6,7)	0.028	8	(1,4,6,8)	0.022
9	(2,3,5,7)	0.014	10	(2,3,5,8)	0.009
11	(2,3,6,7)	0.014	12	(2,3,6,8)	0.009
13	(2,4,5,7)	0.013	14	(2,4,5,8)	0.008
15	(2,4,6,7)	0.013	16	(2,4,6,8)	0.008

k	c_k	$\mathcal{R}(w_k)$	k	c_k	$\mathcal{R}(w_k)$
1	(1,3,5,7)	0.029	2	(1,3,5,8)	0.023
3	(1,3,6,7)	0.029	4	(1,3,6,8)	0.023
5	(1,4,5,7)	0.029	6	(1,4,5,8)	0.023
7	(1,4,6,7)	0.028	8	(1,4,6,8)	0.022
9	(2,3,5,7)	0.014	10	(2,3,5,8)	0.009
11	(2,3,6,7)	0.014	12	(2,3,6,8)	0.009
13	(2,4,5,7)	0.013	14	(2,4,5,8)	0.008
15	(2,4,6,7)	0.013	16	(2,4,6,8)	0.008

Experimental validation

PARSEC

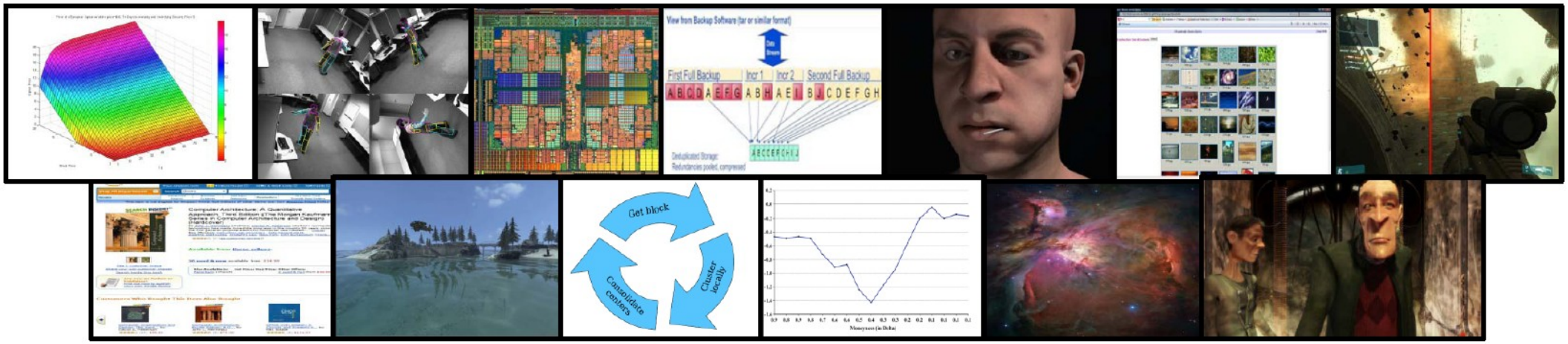
['pär-''sek] A unit of measure

PARSEC

['pär-''sek] A unit of measure

Evaluate multi-core and multiprocessor systems

Evaluate multi-core and multiprocessor systems



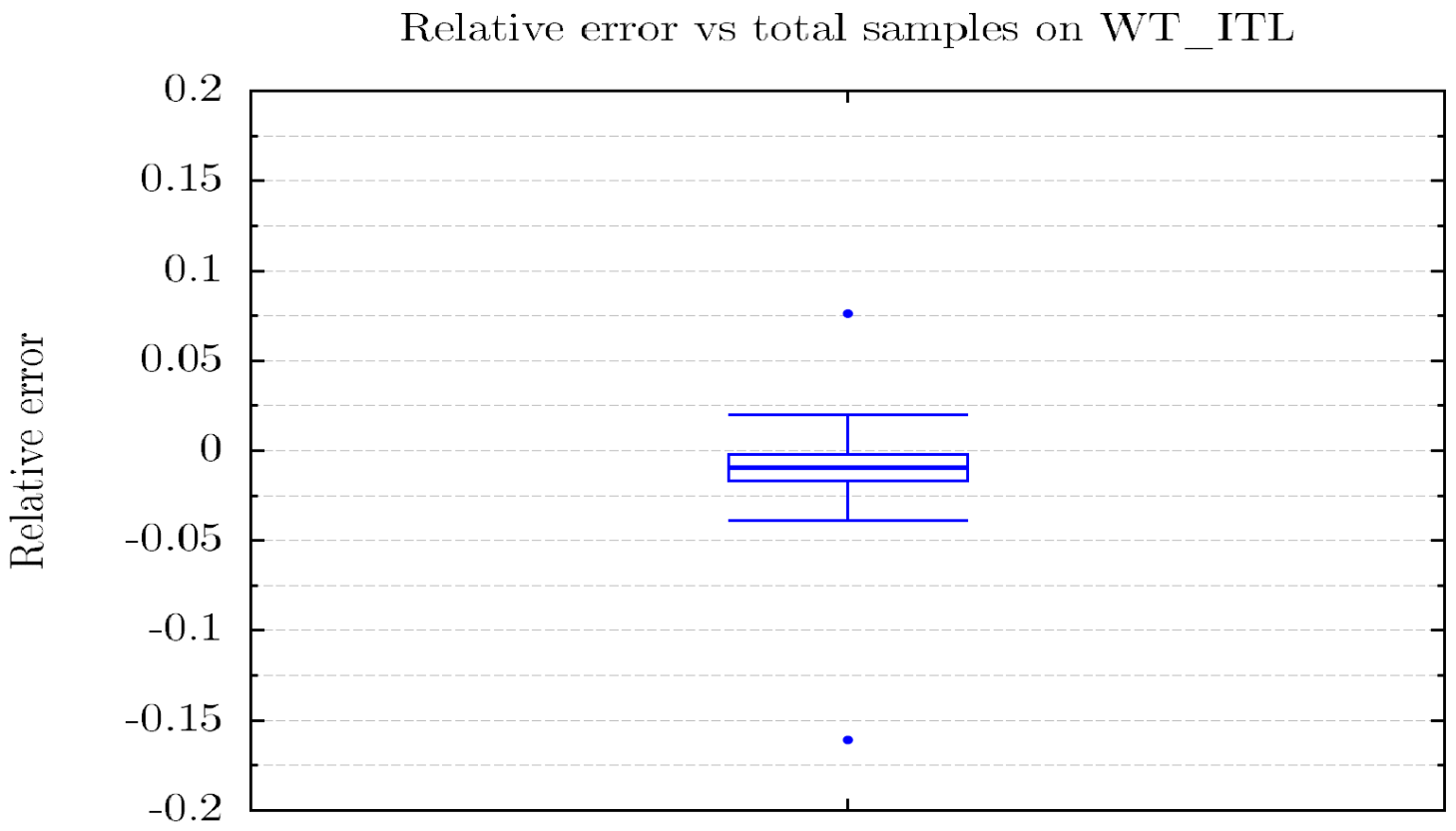
	ITL	AMD
Frequencies	1.60	1.00
	1.73	1.20
	2.00	1.50

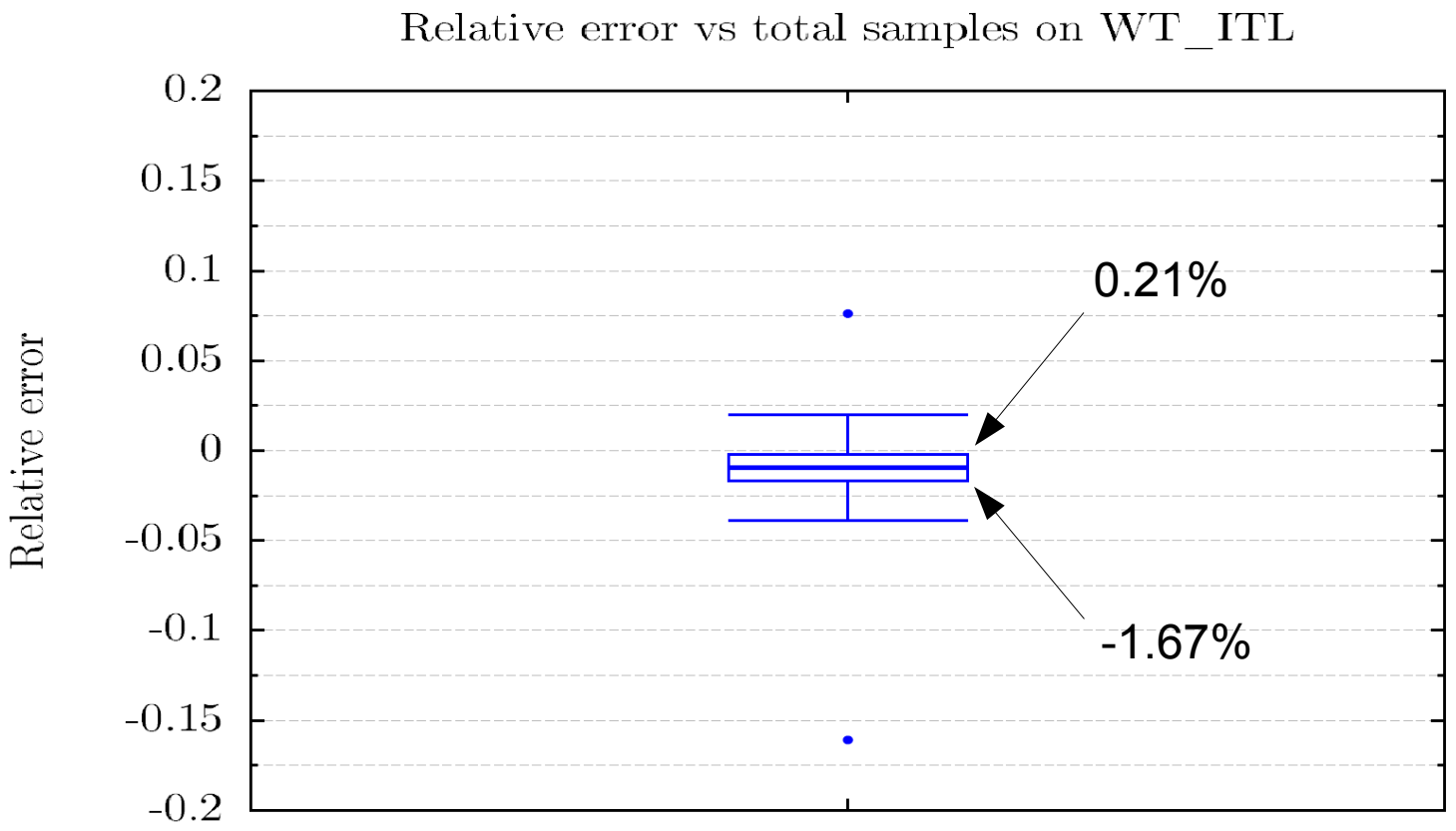
	ITL	AMD
Frequencies	1.60 1.73 2.00	1.00 1.20 1.50
# Cores	1, 2, 4, 6, 8	1, 2, 4, 6, …, 16

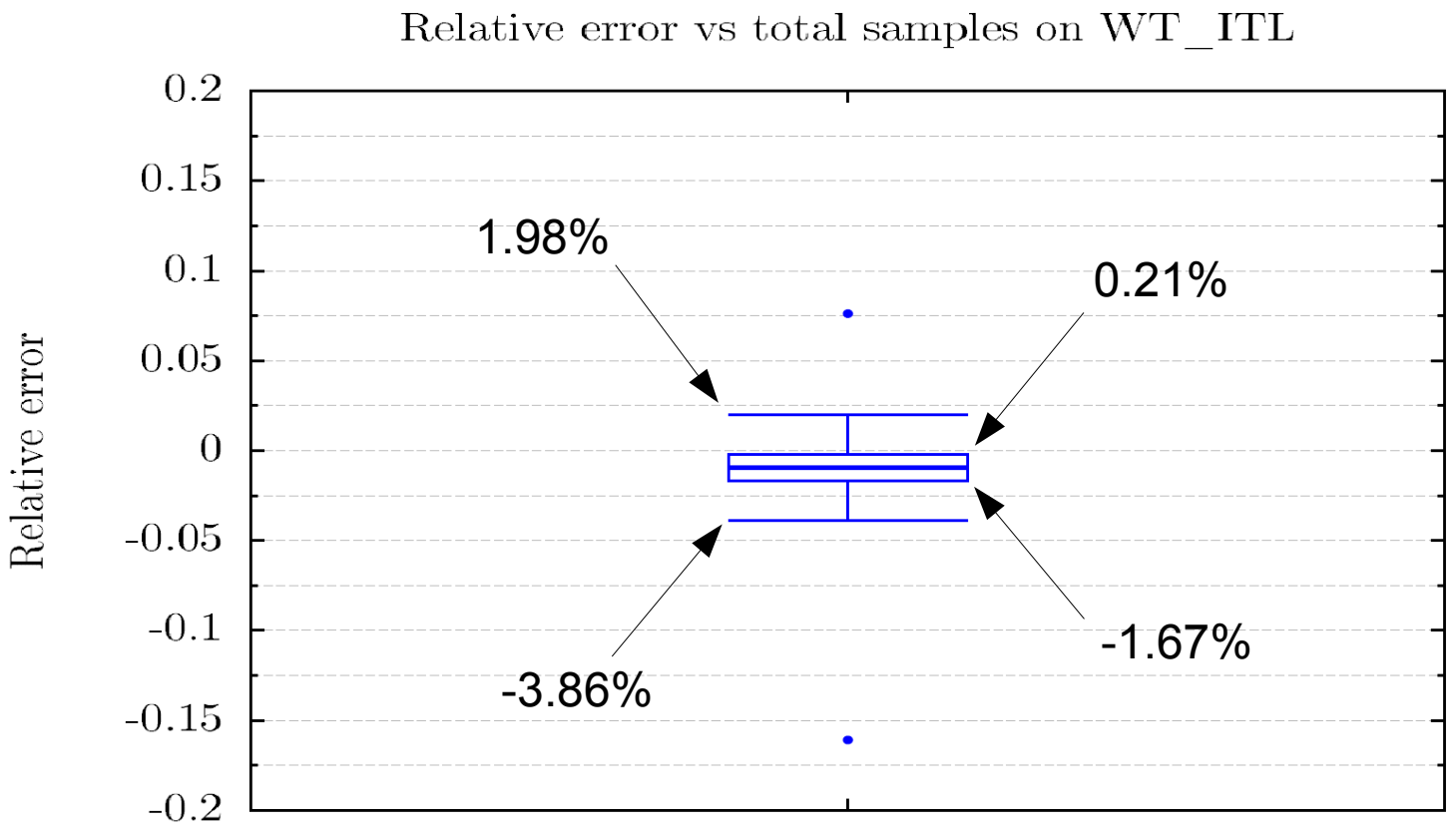
	ITL	AMD
Frequencies	1.60 1.73 2.00	1.00 1.20 1.50
# Cores	1, 2, 4, 6, 8	1, 2, 4, 6, …, 16
#Samples	≈ 12,000,000	≈ 22,000,000

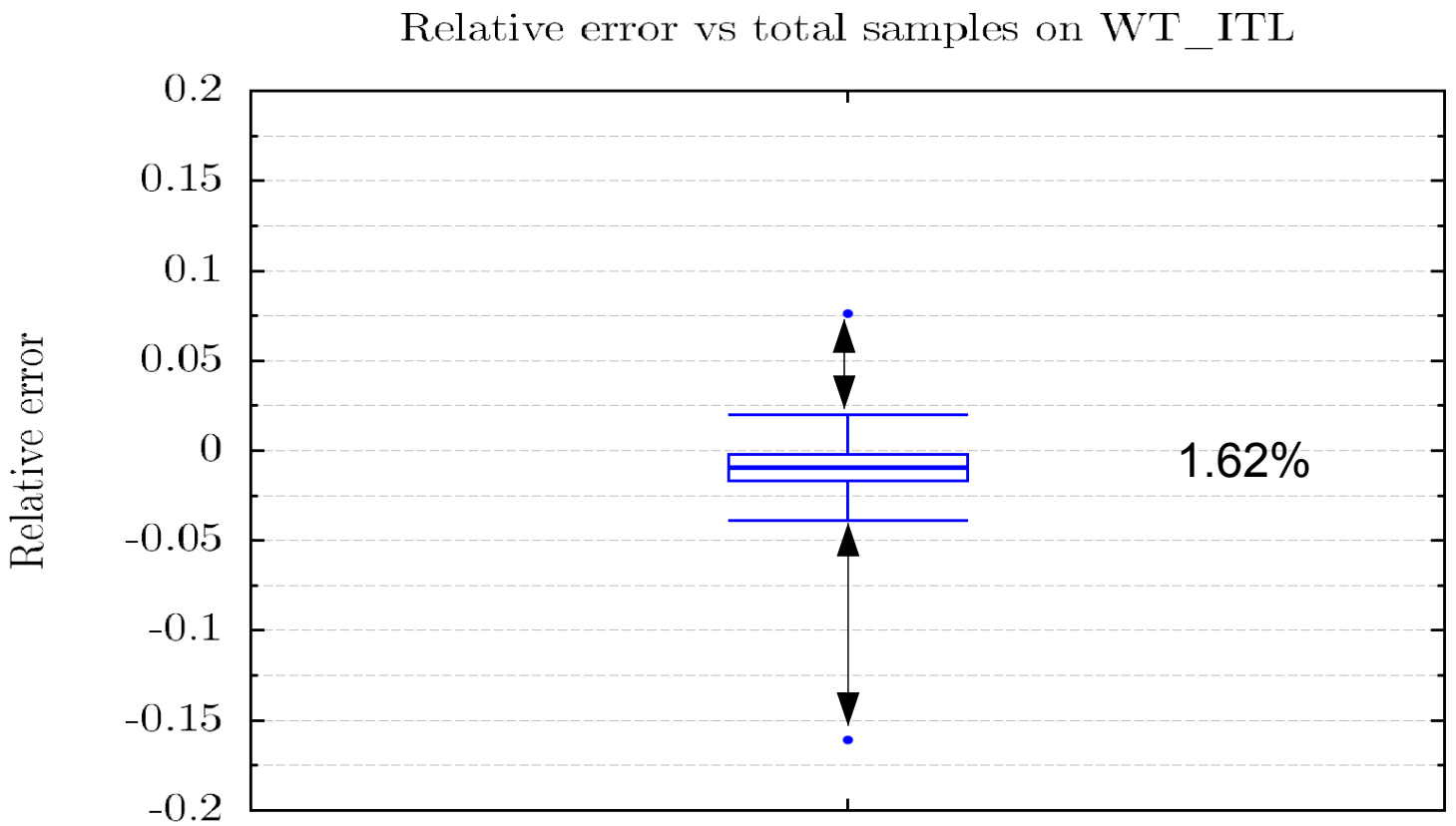
Reliability of the reduced model

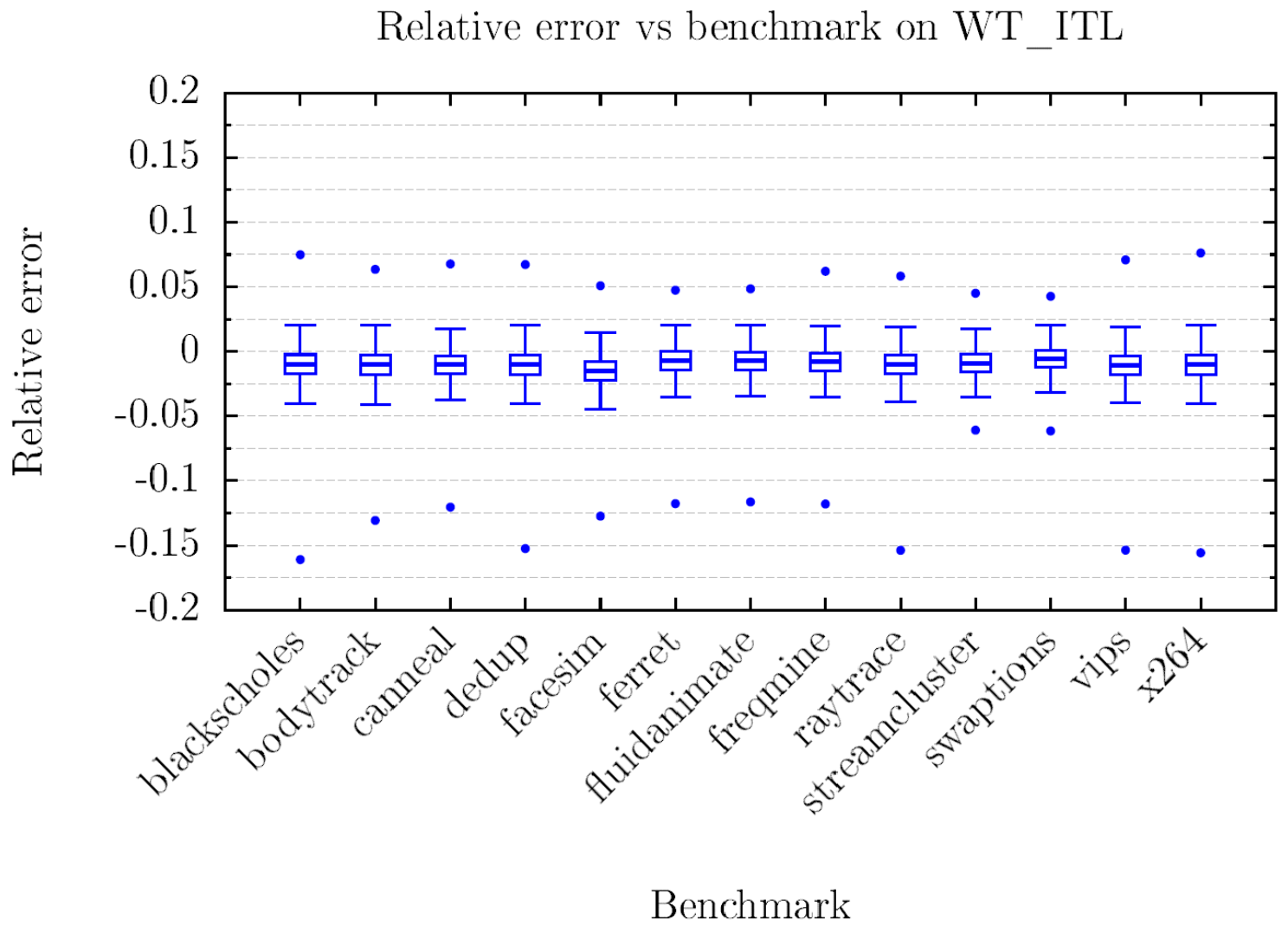
$$\frac{p_T(k) - \hat{p}_T(k)}{p_T(k)}, \quad k = 0, 1, 2, \ldots, \texttt{max\_samples}$$

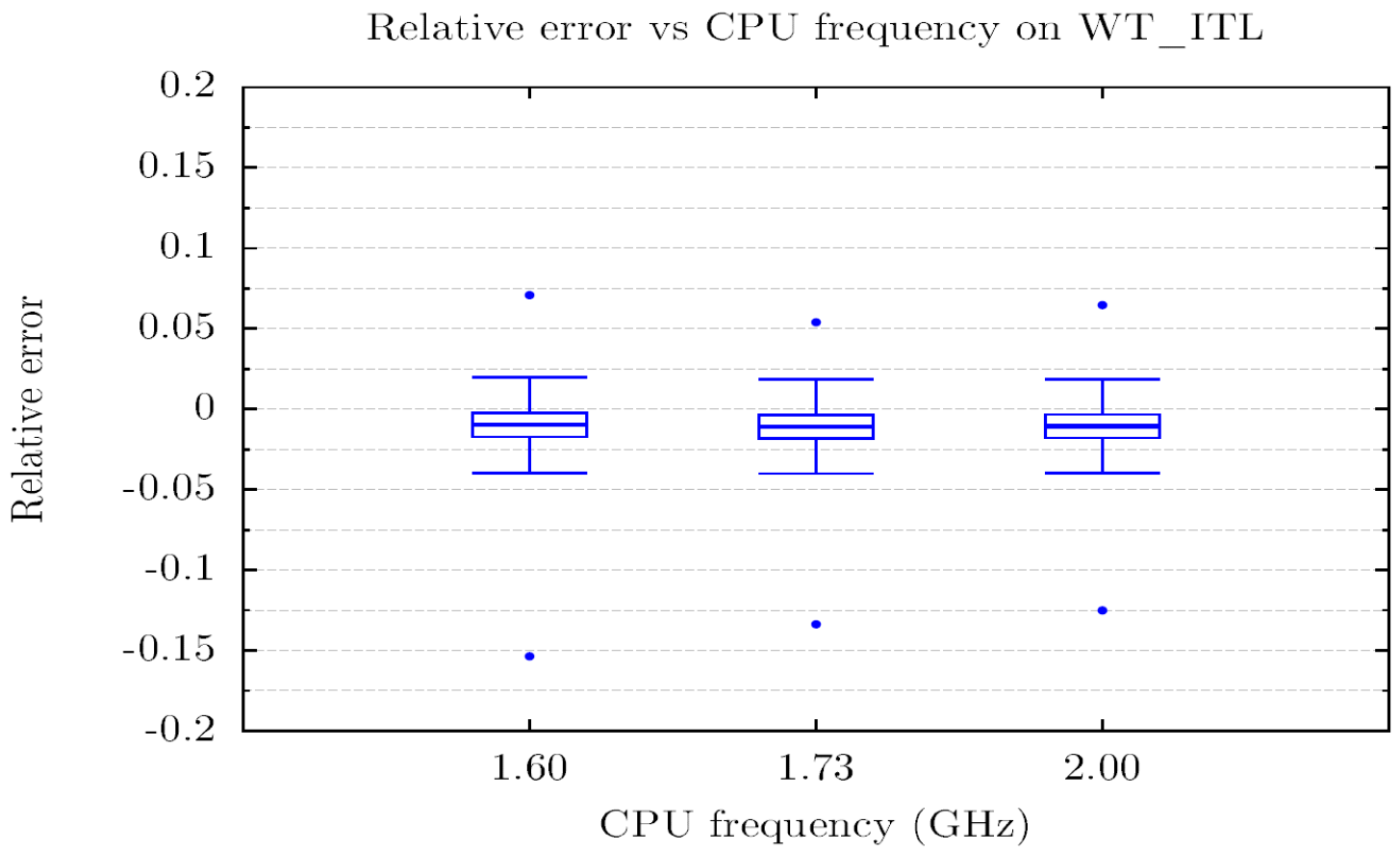

Relative error vs total samples on WT_ITL
Relative error
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2

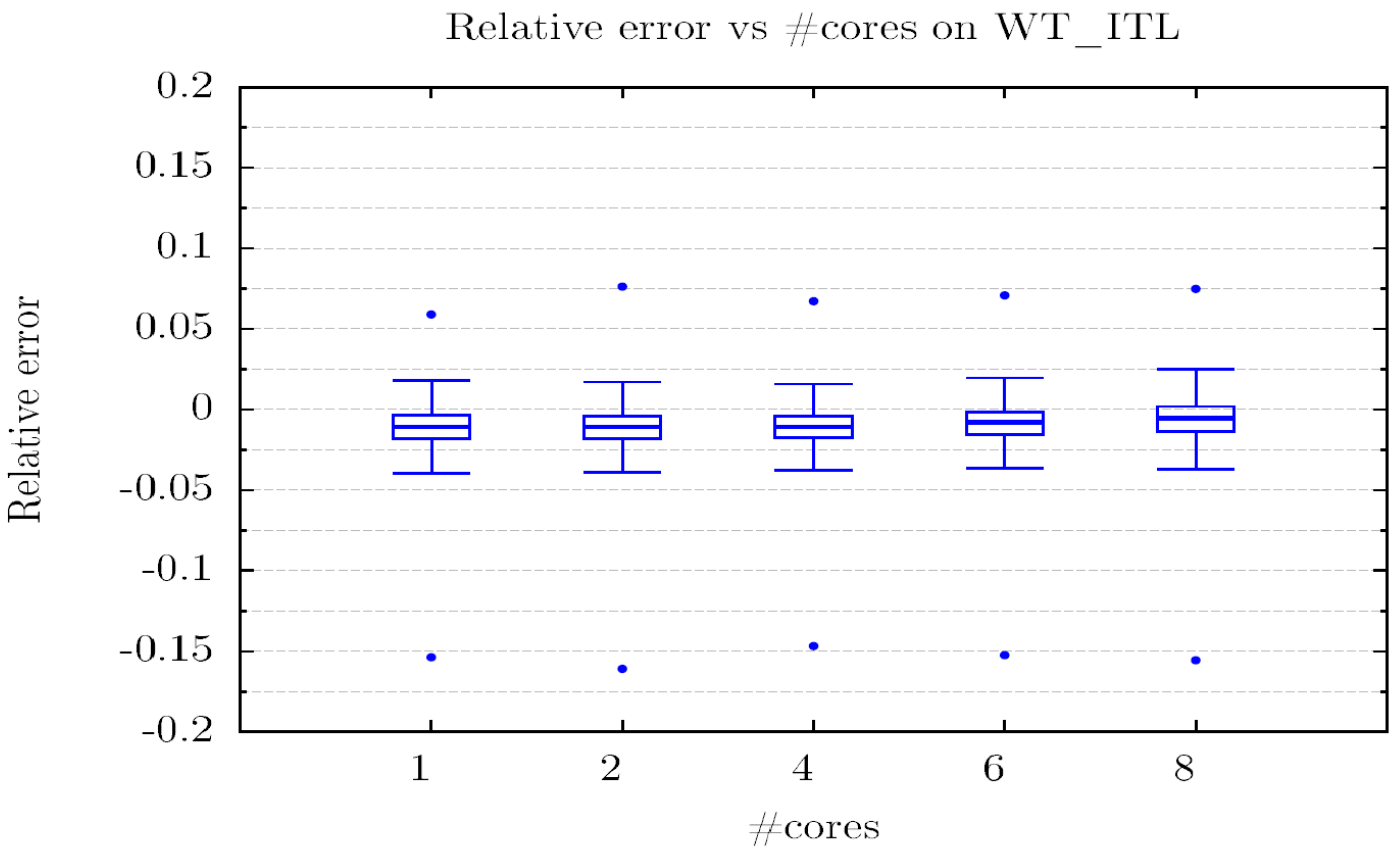

Relative error vs total samples on WT_ITL
Relative error
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0.21%
-1.67%


Relative error vs total samples on WT_ITL
Relative error
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
1.98%
0.21%
-3.86%
-1.67%


Relative error vs total samples on WT_ITL
Relative error
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
1.62%


Relative error vs benchmark on WT_ITL
Relative error
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
blackscholes
bodytrack
canneal
dedup
facesim
ferret
fluidanimate
freqmine
raytrace
streamcluster
swaptions
vips
x264
Benchmark


Relative error vs CPU frequency on WT_ITL
Relative error
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
1.60
1.73
2.00
CPU frequency (GHz)


Relative error vs #cores on WT_ITL
Relative error
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
1
2
4
6
8
#cores

Conclusions

Is it possible to ***estimate the total instantaneous power*** by using just a ***few lines***?

It is possible to ***estimate the total instantaneous power*** by using just a ***few lines***?

Calibration of the reduced model

Simple, fast and automatic

Results

Independent of the application, frequency and number of cores

Highly accurate

Take into account lines with different voltage and platforms with GPU

Investigate the minimum number of calibration runs required

Build a portable tool