

Manual and Automatic Energy Tuning for HPC Codes

The Projects Score-E and READEX



Horizon 2020 European Union funding for Research & Innovation

SPONSORED BY THE



Federal Ministry of Education and Research Kai Diethelm GNS Gesellschaft für numerische Simulation mbH Michael Gerndt TU München

Outline

- Motivation
- Score-E and READEX
- Energy Monitoring for HPC applications
- Application analysis
- Automatic energy tuning
- Applications



Exascale Energy Wall

#	Site	System	Cores (M)	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (MW)	Exascale (Factor)	Exascal e (MW)
1	National Supercomputing Center, Wuxi, China	Sunway TaihuLight - Sunway 1.45GHz, Sunway, NRCPC	10.6	93.0	125.4	15.4	8	123
2	National Super Computer Center in Guangzhou, China	Tianhe-2 (MilkyWay-2), Intel Xeon E5-2.2GHz, Intel Xeon Phi 31S1P, NUDT	3.1	33.9	54.9	17.8	18	324
3	DOE/SC/Oak Ridge National Laboratory, United States	Titan - Cray XK7 , Opteron 2.2GHz, NVIDIA K20x, Cray Inc.	0.5	17.6	27.1	8,2	37	303
4	DOE/NNSA/LLNL, United States	Sequoia - BlueGene/Q, Power BQC 1.60 GHz, IBM	1.6	17.2	20.1	7,9	50	392
5	DOE/SC/LBNL/ NERSC, United States	Cori - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Cray Inc.	0.6	14.0	27.9	3,9	36	141

GA Gauß-Allianz

Score-E



Federal Ministry of Education and Research

- Partners:
 - GNS Gesellschaft f
 ür numerische Simulation musich, Braunschweig (Coordinator)
 - RWTH Aachen
 - German Research School for Simulation Sciences, Aachen (until February 2015), TU Darmstadt (from March 2015)
 - TU Dresden
 - FZ Jülich
 - TU München
- Associated Partners:
 - Engys GmbH (Rostock), Munters Euroform GmbH (Aachen),
 - U of Oregon
- Funded by BMBF (Grant No. 01IH13001)
- October 2013 September 2016



Semi-automatic and Automatic Energy Tuning



Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing

Funded by the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 671657.



Horizon 2020 European Union funding for Research & Innovation



READEX Project partners

- TU Dresden (Coordinator), Germany
- NTNU, Norway
- IT4I, Czech Republic
- TUM, Germany
- Intel Exascale Lab, France
- GNS Braunschweig, Germany
- ICHEC, Ireland





Score-E: Energy Monitoring with Score-P

Metric Plugins:

- Energy: Intel RAPL, AMD APM, HDEEM
- Temperature
- C-/P-states
- Synchronization Adapter
- Shared resources





Manual Analysis of Energy Usage with Vampir



NAS Parallel Benchmarks (MG)



Manual Analysis with Cube on JURECA





Deselected "Compute loop synced"

Energy Consumption Projection

Goal: Prediction of energy consumption on larger systems

- Based on Extra-P (GRS, TU Darmstadt, ETH Zürich, LLNL, FZ Jülich)
- Analytical model deduced from application profiles
- Visualization of predicted energy consumption in Cube



MILC: QCD code



Resource-Aware Visualization

• • •

Severity	Self	F	Perform	mance Metr	ic				
4.96e+04	-3.08e-	11 sec	▶ Tir	ne					
6.07e+08	6.07e+0	08 000	Vis	sits					
0		0 000	▶ Sy	nchroniza	tions				
0		0 000	► Co	mmunica	tions				
8.27e+10		0 bytes	▶ By	tes transf	erred				
0		0 000	▶ MI	Pl file oper	rations				
0		0 sec	► Co	mputation	nal imb	alance			
0		0 sec	Mi	nimum Inc	clusive	Time			
4.03e+04	4.03e+0)4 sec	Ma	aximum In	clusive	Time			
0		0 bytes	by	tes_put					
0		0 bytes	by	tes_get					
				0					
		Hie	rarch	• y Tree	map				
Se	everity 🔻	Hie	rarch	y Tree Code Reg	map				
Se 4	everity ▼ .96e+04	Hie Self 1.1	erarch sec	● Tree Code Reg ▼ MAIN	map ion				
Se 4 4	everity V 96e+04 94e+04	Hie Self 1.1 0.0887	sec sec	● Tree Code Reg ● MAIN ● di	map ion I river_				
Se ■ 4 ■ 4	everity ▼ .96e+04 .94e+04 .81e+04	Hie Self 0.0887 0.737	sec sec sec	y Tree Code Reg MAIN di	map ion I river_ cps0	2_			
Se 4 4 4 4 3	everity V 96e+04 94e+04 .81e+04 .74e+04	Hie Self 1.1 0.0887 0.737 11.1	sec sec sec sec	y Tree Code Reg WAIN V di	ion I river cps0 ▼ lo	2_ peser_			
Se 4 4 4 3 3 3	everity ▼ 96e+04 94e+04 .81e+04 .74e+04 .73e+04	Hie Self 0.0887 0.737 11.1 0.242	sec sec sec sec sec sec	y Tree Code Reg MAIN v di	map ion I river_ cps0 v lo	2_ eser_ lindir_			
S6 4 4 4 3 3 3 3 3 3 3 3 3 3 3	everity ▼ 96e+04 94e+04 .81e+04 .74e+04 .73e+04 .73e+04	Hie Self 0.0887 0.737 11.1 0.242 23	sec sec sec sec sec sec sec sec	y Tree Code Reg MAIN V di	map ion I river_ cps0 v lo	2_ neser_ ∫ lindir_ ▼ lin	dik_		
S6 4 4 4 3 3 3 3 3 2 2	everity ▼ 96e+04 .94e+04 .74e+04 .73e+04 .73e+04 .91e+04	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911	sec sec sec sec sec sec sec sec sec	y Tree Code Reg WAIN V di	map ion I river_ cps0 ▼ lo	2_ veser_ ∎ lindir_ ∎ lin	dik_ feti	i_s(blve_
S6 4 4 3 3 3 3 3 3 2 2 2 2 2	everity ▼ 96e+04 94e+04 .81e+04 .74e+04 .73e+04 .73e+04 .91e+04 .91e+04	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg MAIN V du	map ion I river_ cps0 V lo	2_ reser_ ▼ lindir_ ▼ lin	dik_ feti	i_so	blve_ vel1_feti_
Se 4 4 3 3 3 3 2 2 2	everity ♥ 96e+04 94e+04 .81e+04 .74e+04 .73e+04 .73e+04 .91e+04 .91e+04 .11e+04	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197 20.8	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg MAIN v di	map ion I river_ cps0 V lo	2_ peser_ lindir_ ▼ lin ▼	dik_ fet	i_so lev	blve_ vel1_feti_ interfac.
S6 4 4 3 3 3 3 2 2 2 2 7 7	everity ♥ 96e+04 94e+04 .81e+04 .73e+04 .73e+04 .91e+04 .91e+04 .11e+04 2.47e+03	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197 20.8 7.21e+03	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg MAIN v di	map ion ↓ river_ cps0 ▼ lo	2_ peser_ Iindir_ ▼ Iin ▼	dik_ feti ▼	i_so lev ▶	blve_ vel1_feti_ interfac. fwbw_tri
Se 4 4 3 3 3 3 2 2 2 2 2 2 7 5 5	everity ▼ 96e+04 94e+04 .81e+04 .73e+04 .73e+04 .91e+04 .91e+04 .11e+04 .11e+04 .47e+03 .72e+03	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197 20.8 7.21e+03 5.49e+03	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg WAIN V di	map ion I river_ cps0 v lo	2_ peser_ ▼ lindir_ ▼ lin	dik_ fetï ▼	i_sc lev	olve_ rel1_feti_ interfac. fwbw_tri fwbw_tri
Se 4 4 3 3 3 3 3 2 2 2 7 5 5 2 2 7 5 5 2	everity ♥ 96e+04 94e+04 .81e+04 .73e+04 .73e+04 .91e+04 .91e+04 .11e+04 .47e+03 .72e+03 .96e+03	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197 20.8 7.21e+03 5.49e+03 12.5	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg WAIN V di	map ion I river_ ⊂cps0 ▼ lo	2_ peser_ ▼ lindir_ ▼ lin	dik_ feti ▼	i_sc lev	olve_ rel1_feti_ interfac. fwbw_tri fwbw_tri interfac.
Se 4 4 3 3 3 3 2 2 5 5 2 2 5 5 2 2 5 5 2 2 5 5 5 5 5 5 5 5 5 5 5 5 5	everity ♥ 96e+04 94e+04 .81e+04 .73e+04 .73e+04 .91e+04 .91e+04 .11e+04 .47e+03 .72e+03 .96e+03 .650	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197 20.8 7.21e+03 5.49e+03 12.5 2.55	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg WAIN V di	map ion J river_ cps0 v lo	2_ beser_ lindir_ ▼ lin ▼	dik_ fetï ▼	i_so lev	olve_ vel1_feti_ interfac. fwbw_tri fwbw_tri interfac. sparse.
Se 4 4 3 3 3 3 3 3 2 2 2 7 5 5 2 2	everity ♥ 96e+04 94e+04 .81e+04 .73e+04 .73e+04 .91e+04 .91e+04 .11e+04 .47e+03 .96e+03 .96e+03 .650 .416	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197 20.8 7.21e+03 5.49e+03 12.5 2.55 19.8	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg WAIN V di	map ion J river_ cps0 v lo	2_ beser_ r lindir_ ▼ lin ▼	dik_ fet		olve_ rel1_feti_ interfac. fwbw_tri interfac. sparse_ globalsu
Se 4 4 4 3 3 3 2 2 7 5 5 2 2 5 2 2 5 5 2 2 5 5 5 5 5 5 5 5 5 5 5 5 5	everity ♥ 96e+04 94e+04 .81e+04 .73e+04 .73e+04 .91e+04 .91e+04 .11e+04 .47e+03 .72e+03 .96e+03 .650 .416 .233	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197 20.8 7.21e+03 5.49e+03 5.49e+03 12.5 2.55 19.8 7.67	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg WAIN V di	map ion I river cps0 V lo	2_ beser_ ▼ lindir_ ▼ lin	dik_ feti ▼		olve_ el1_feti_ interfac. sparse globalsu half_exp
Sec 4 4 4 3 3 3 3 3 3 2 2 2 7 5 5 2 2	everity ♥ 96e+04 94e+04 .81e+04 .73e+04 .73e+04 .91e+04 .91e+04 .11e+04 .47e+03 .72e+03 .96e+03 .650 .416 .233 .145	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197 20.8 7.21e+03 5.49e+03 12.5 2.55 19.8 7.67 9.9	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg WAIN V di	map ion I river_ © cps0 ♥ lo	2_ beser_ ▼ lindir_ ▼ lin	dik_ feti ▼		olve_ rel1_feti_ interfac. fwbw_tri interfac. sparse globalsu half_exp interfac.
Sec 4 4 4 3 3 3 3 3 3 2 2 2 7 5 5 2 2	everity ♥ 96e+04 94e+04 .81e+04 .73e+04 .73e+04 .91e+04 .91e+04 .11e+04 .47e+03 .96e+03 .650 .416 .233 .145 .39.3	Hie Self 1.1 0.0887 0.737 11.1 0.242 23 0.0911 197 20.8 7.21e+03 5.49e+03 12.5 2.55 19.8 7.67 9.9 39.3	sec sec sec sec sec sec sec sec sec sec	y Tree Code Reg WAIN di	map ion I cps0 V lo	2_ beser_ ▼ lindir_ ▼ lin	dik_ feti ▼		olve_ rel1_feti_ interfac. fwbw_tri interfac. sparse globalsu half_exp interfac. mtxv_



Geometry-Aware Visualization

....

Severity	Self		Per	formance Metric
4.96e+04	-3.08e-11	sec	►	Time
6.07e+08	6.07e+08	occ		Visits
0	0	occ	►	Synchronizations
0	0	occ	►	Communications
8.27e+10	0	bytes	►	Bytes transferred
0	0	000	►	MPI file operations
0	0	sec	►	Computational imbalance
0	0	sec		Minimum Inclusive Time
4.03e+04	4.03e+04	sec		Maximum Inclusive Time
0	0	bytes		bytes_put
0	0	bytes		bytes_get

				_			
	Hie	rarch	У	Treemap			
Severity 🔻	Self		Code	e Region			
4.96e+04	1.1	sec		MAIN_			
4.94e+04	0.0887	sec		driver_			
4.81e+04	0.737	sec		▼ cps0	2_		
3.74e+04	11.1	sec		▼ lo	eser_		
3.73e+04	0.242	sec			lindir_		
3.73e+04	23	sec			▼ line	dik_	
2.91e+04	0.0911	sec				feti_s	olve_
2.91e+04	197	sec				▼ lev	/el1_feti_
1.11e+04	20.8	sec				•	interfac
7.47e+03	7.21e+03	sec				►	fwbw_tri
5.72e+03	5.49e+03	sec				►	fwbw_tri
2.96e+03	12.5	sec				►	interfac
650	2.55	sec				►	sparse
416	19.8	sec				►	globalsu
233	7.67	sec				►	half_exp
145	9.9	sec				►	interfac
39.3	39.3	sec					mtxv_
 	0.1						



Geometry-Aware Visualization

Severity	Self		Per	formance Metric
4.96e+04	-3.08e-11	sec	►	Time
6.07e+08	6.07e+08	occ		Visits
0	0	000	►	Synchronizations
0	0	000	►	Communications
8.27e+10	0	bytes	►	Bytes transferred
0	0	occ	►	MPI file operations
0	0	sec	►	Computational imbalance
0	0	sec		Minimum Inclusive Time
4.03e+04	4.03e+04	sec		Maximum Inclusive Time
0	0	bytes		bytes_put
0	0	bytes		bytes_get



	Hie	rarchy	/	Treemap					
Severity V	Self		Cod	de Region					
4.96e+04	1.1	sec	▼	MAIN_					
4.94e+04	0.0887	sec		driver_					
4.81e+04	0.737	sec		▼ cps	s02_				
3.74e+04	11.1	sec			loes	er_			
3.73e+04	0.242	sec			▼ I	indir_			
3.73e+04	23	sec			,	🛛 lin	dik_		
2.91e+04	0.0911	sec					feti	_S	olve_
2.91e+04	197	sec					W	lev	/el1_feti_
1.11e+04	20.8	sec						►	interfac.
7.47e+03	7.21e+03	sec						▶	fwbw_tri
5.72e+03	5.49e+03	sec						►	fwbw_tri
2.96e+03	12.5	sec						►	interfac.
650	2.55	sec						►	sparse_
416	19.8	sec						►	globalsu
233	7.67	sec						►	half_exp
145	9.9	sec						►	interfac.
39.3	39.3	sec							mtxv_
20.0	01								

Energy Tuning with Periscope Tuning Framework

Automatic application analysis & tuning

- Tune performance and energy (statically)
- Plug-in-based architecture
- Evaluate alternatives online
- Scalable and distributed framework
- Support variety of parallel paradigms
 - MPI, OpenMP, OpenCL, Parallel pattern
- Developed in the AutoTune EU-FP7 project
 - Integrated with Score-P in Score-E





Tuning Plugins for Static Energy Tuning

- PCAP: Optimization of the thread number
 - Exploits scalability limitations
 - Reduces static and dynamic energy
 - Supports tuning of individual parallel
 - regions
 - MPICAP: Optimizes the number of MPI tasks
 - In addition: reduces communication energy
 - **DVFS**: Computes the optimal core frequency
 - Exploits wait cycles (IO-/Memory Bound)
 - Reduces dynamic energy
 - Taurus-Energy: Combined tuning (3D)





READEX: Beyond Static Tuning

HPC

• Automatic Tuning

Embedded

• System Scenarios



Systems Scenario based Methodology





Exploit Intra- and Inter-Phase Dynamism

```
int main(void) {
    // Initialize application
    // Initialize experiment variables
    int num_iterations = 2;
```

```
for (int iter = 1; iter <= num_ite
    // Start phase region
    // Read PhaseCharct
    laplace3D(); // significant regi
    residue = reduction(); // insign
    fftw_execute(); // significant r
    // End phase region
}</pre>
```

```
// Post-processing:
// Write noise matrices to disk fo
// Terminate application
```

```
MPI_Finalize();
return 0;
```





Scenario-Based Tuning



Periscope Tuning Framework (PTF)

READEX Runtime Library (RRL)



Lulesh: Individual, 3 cores per task

individual, 3 cores/task	Worst	Static Tuning	READEX	#Threads	Freq
	16; 1,2	8; 2,4			
CalcCourantConstraintForE	6075	5375	5232	10	2,4
CalcKinematicsForElems	11170	8260	8250	9	2,4
CalcMonotonicQGradients	4977	3920	3370	6	2,4
CalcMonotonicQRegionFor	11714	12470	11714	16	2,4
CalcVolumeForceForElems	56821	48111	47804	9	2,4
EvalEOSForElems	28335	18311	17082	4	2,4
SUM	119092	96447	93452		
		19,0%	3,1%		
Energy for phase	163891	138508			
		15,5%			



Score-E Application Example: Analysis of Indeed

- Finite Element code for sheet metal forming simulation
- SMP version (OpenMP) & DMP version (hybrid: OpenMP + MPI) available
- Two different application scenarios:
 - Classical use case: direct simulation of forming process
 - Recent variation: determination of optimal process parameters (in particular, tool geometry)
- Fundamental difference: mesh size for discretization of tools
- Consequences:
 - Significance of contact search algorithms changes drastically
 - Classification changes between "compute bound" and "memory bound"



Score-E Application Example: Analysis of Indeed

- Potential tuning parameters:
 - CPU frequency
 - Number of OpenMP threads
 - Number of MPI processes
 - Code path switching



Score-E Application Example: Analysis of Indeed

- Potential tuning parameters:
 - CPU frequency
 - Number of OpenMP threads
 - Number of MPI processes
 - Code path switching

Accepted by customers

Undesired by customers



Measurement Results for Indeed

Relative values w. r. t. default clock frequency (Sandy Bridge, classical example)



Measurement Results for Indeed

Relative values w. r. t. default clock frequency (Haswell, classical example)



Measurement Results for Indeed

Relative values w. r. t. default clock frequency (Haswell, nonclassical example)



Conclusions for Indeed

- Choose high clock frequency
- Exact optimal value depends on
 - Processor architecture
 - Characteristics of input deck
 - Concrete choice of objective function
 - Static tuning very attractive on Sandy Bridge;
- further improvements possible with dynamic tuning
- Optimization on Haswell requires further methods and tools

(automatic dynamic tuning at runtime \rightarrow READEX)

 Viability of platform-dependent approach from commercial point of view?



Future Work in READEX and Expected Outcome

- Use tools based on PTF for design-time analysis (search for significant regions, dynamism detection, ...)
 - Runtime application tuning based on READEX Runtime Library (RRL)
 - Alpha version for full READEX tool suite scheduled for spring 2017
 - Later step:

Programming paradigm for expressing application dynamism

 \rightarrow further improve automatic dynamic energy tuning

 Significant increase in energy efficiency without prohibitively high effort for software developers



Acknowledgements

We thank

- BMBF for providing the funding for Score-E,
- DLR PT-SW for the support and cooperation with respect to Score-E,
- the European Union for the funding for READEX.

Further information

- www.vi-hps.org/projects/score-e
- www.readex.eu

