

ENERGY EFFICIENT HIGH PERFORMANCE COMPUTING

A Case Study on EAR



Max Lübke, Dorian Stoll, Alexander Hacke,
Bettina Schnor
max.luebke@uni-potsdam.de

Institute of Computer Science
University of Potsdam

PARS 2026 @ Innsbruck
September 17, 2026



MOTIVATION: ENERGY EFFICIENCY IN HPC

The Energy Challenge in HPC:

- Supercomputers consume **megawatts** of electric power → massive operational costs and carbon emissions.

Leverage for Energy Saving:

- **Cluster Level (On-Off):** Power down idle nodes during periods of low cluster load (up to **30 % savings** ^{†1}).
- **Node Level (DVFS):** Dynamically adapt CPU frequency to the running workload.

DVFS Principle in HPC:

- Power scales linearly with frequency.
- **HPC Opportunity:** Memory stalls and MPI communication allow frequency scaling with **minimal slowdown!**

Evolution of MPI DVFS Frameworks:

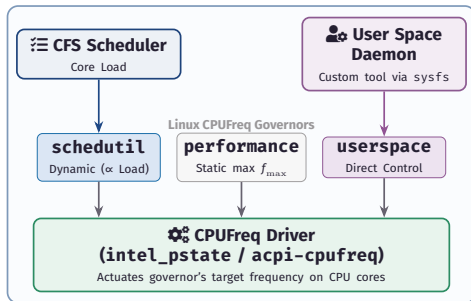
- **Adagio (2009)**
- **Energy Aware MPI (2015)**
- **Energy Aware Runtime (EAR) (2020+):**
Installed on SuperMUC and MareNostrum 5.

^{†1}S. Kiertscher, B. Schnor: *Optimizing Energy Efficiency and QoS*, GreenCom 2016.

WHY EAR?

Linux CPUFreq Architecture:

🔧 OS-Controlled Scaling



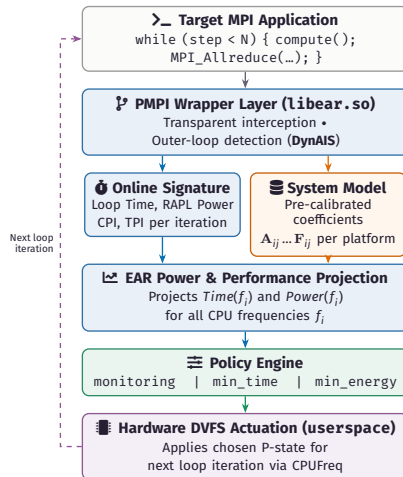
Motivation: Standard Linux as Baseline

- Linux provides DVFS mechanisms out-of-the-box (schedutil, HWP).
- **The Core Question:** Can standard OS mechanisms save energy in HPC, or do we need tools like EAR?

ENERGY AWARE RUNTIME (EAR): ARCHITECTURE

Core Architecture:

- **PMPI Interception (libear.so):**
 - Transparently wraps MPI calls via LD_PRELOAD.
 - Automatically identifies outer loops using DynAIS.
- **Online Application Signature:**
 - Measures *Time*, *Power* (RAPL), *CPI*, and memory bandwidth (*TPI*) per iteration.
- **Offline System Signature:**
 - Pre-calibrated regression coefficients.
- **Performance Model Projection:**
 - Predicts runtime and energy across all available P-states/frequencies.



EAR ENERGY POLICIES

Performance–Energy Trade-Off via Hyperparameter X

Implemented Energy Policies:

- **monitoring:**
 - Runs statically at default frequency.
 - Isolates EAR's pure telemetry overhead.
- **min_time (Bottom-Up):**
 - Starts at low clock ($f_{\text{def}} = 2.0$ GHz).
 - Steps up only if $\text{PerfGain} \geq \text{FreqGain} \cdot X$.
 - $X = 0.70$ (**Efficiency Factor**): +10% frequency jump requires $\geq 7\%$ speedup.
- **min_energy (Top-Down):**
 - Starts at max clock ($f_{\text{def}} = 3.2$ GHz).
 - Steps down if energy drops and $\text{Slowdown} \leq X$.
 - $X \in \{0.05, 0.20\}$ (**Penalty Budget**): At most +5% or +20% allowed slowdown.

Evaluated Configurations:

Class	Strategy	Governor	Def. Freq.	Hyperparam. (X)
HWP	bal. perf.	—	—	—
Linux	performance schedutil	performance schedutil	— —	— —
EAR	monitoring	userspace	3.2 GHz	—
	min_time	userspace	2.0 GHz	0.70
	min_energy	userspace	3.2 GHz	0.05
	min_energy	userspace	3.2 GHz	0.20

EXPERIMENTAL SETUP: TESTBED

Hardware Specifications and Measurement Methodology

Hardware Specifications:

- **Cluster Architecture:**
 - 2 compute nodes + 1 head node.
 - Gigabit Ethernet (GbE) interconnect.
- **Processor:**
 - Intel® Core™ i7-8700 (Coffee Lake).
 - 6 cores / socket → **12 physical cores total**.
 - Clock Range: 0.8 GHz to 3.2 GHz.
- **Memory:** 32 GB DDR4 RAM per node.

Configuration:

- **EAR version:** 5.1
- **min_time** starting at 2.0 GHz
- **min_energy** starting at 3.2 GHz
- **Turbo Boost off.**
- **SMT (Hyper-Threading) off.**
- **Energy Measurement:**
 - Intel Running Average Power Limit (**RAPL**).
 - Measures both **Package (CPU)** and **DRAM** power domains.

EXPERIMENTAL SETUP: EVALUATED HPC WORKLOADS

Benchmark Characteristics

Evaluated Workloads:

Benchmark	Domain	Training?
BT-MZ.D	Block Tri-diag. Solver	Yes
LU-MZ.D	LU Gauss-Seidel Solver	Yes
SP-MZ.D	Scalar Penta. Solver	Yes
POET	Reactive transport	No
GROMACS	Molecular dynamics	No

NPB-MZ with Class D dataset; 1 MPI rank per physical core.

Methodology:

- NPB-MZ used to train EAR's system signature.
- POET & GROMACS test generalization to unseen codes.
- Workload chosen to achieve **15–30 minutes** per run.
- **3 repetitions** per configuration.

RESULT: OVERHEAD OF EAR

performance Baseline vs. EAR's monitoring

Overhead Measurement:

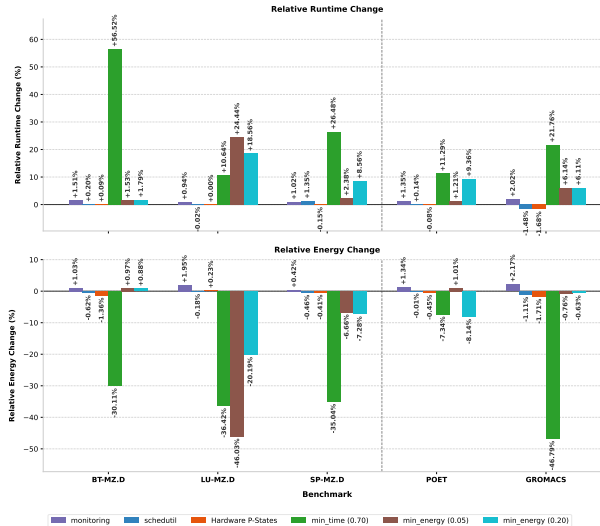
- Both configurations execute at an identical static 3.2 GHz.
- Difference represents overhead of EAR.
- **Average Runtime Overhead: +1.36 %**
 - From +0.94% (LU-MZ) to +2.02% (GROMACS).
- **Average Energy Overhead: +1.48 %**
 - From +0.42% (SP-MZ) to +2.17% (GROMACS).

Overhead Summary:

Benchmark	Runtime Overhead	Energy Overhead
BT-MZ.D	+1.51%	+1.03%
LU-MZ.D	+0.94%	+1.95%
SP-MZ.D	+1.02%	+0.42%
POET	+1.35%	+1.34%
GROMACS	+2.02%	+2.17%
Mean	+1.36 %	+1.48 %

→ Confirms that EAR's profiling overhead is remarkably low (< 2%).

RESULTS: RUNTIME AND ENERGY MEASUREMENTS



TAKEAWAYS: POLICY BEHAVIOR

Savings vs. Penalties

Linux Schedutil & HWP:

- Differed from performance by $< 1\%$ in runtime/energy.
- **No energy savings** on workloads.
- **OS “Blindness” in HPC:**
 - **Memory Stalls:** Counted as 100% core load by CFS.
 - **MPI Busy-Waiting:** MPI_Recv spins in user space.

→ Governors stay pinned at $f_{\max}$!

EAR min_time (0.70):

- Massive energy cuts for:
 - **GROMACS:** -46.79 % Energy (+21.8% Time).
 - **LU-MZ.D:** -36.42 % Energy (+10.6% Time).
 - **SP-MZ.D:** -35.04 % Energy (+26.5% Time).
- Higher runtime with less energy savings:
 - **BT-MZ.D:** -30.1% Energy (+56.5 % Time)
 - **POET:** -7.34% Energy (+11.3 % Time)

EAR min_energy:

- **0.05 budget:** -46.0% energy on LU-MZ (+24.4% time), but limited gains or energy increases elsewhere.
- **0.20 budget:** Failed to beat 0.05 despite higher allowed runtime penalty.

THE NORMALIZED ENERGY RUNTIME (NER) SCORE

Metric & Ranking

Metric Definition:

Geometric mean of normalized runtimes

$\tilde{R}_i = R_i / R_{\text{perf},i}$ and energies $\tilde{E}_i = E_i / E_{\text{perf},i}$ across $N = 5$ workloads:

$$\text{NER}(\text{Strategy}) := \sqrt[2N]{\prod_{i=1}^N \tilde{R}_i \cdot \prod_{i=1}^N \tilde{E}_i}$$

- **Baseline Reference:** performance $\equiv$ 100.00 %.
- **Lower is Better:** < 100 % indicates net improvement over baseline.

Desktop Testbed Results ($N = 5$):

Configuration	NER-Score	Assessment
monitoring	101.37 %	Profiling overhead (+1.37%)
performance	100.00 %	Static Baseline
schedutil / HWP	99.44–99.78%	Marginal ($\Delta < 0.6\%$)
min_time (0.70)	91.74 %	Rank 1 (Best Trade-Off)
min_energy (0.05)	96.61 %	Moderate Gain (+3.39%)
min_energy (0.20)	100.35 %	Worse than Baseline!

→ **min_time** achieves the best global balance (**91.74 %** net score).

→ **min_energy Paradox:** Looser runtime budget (0.20 vs. 0.05) degrades efficiency!

ANALYSIS: MPI SLEEP BENCHMARK

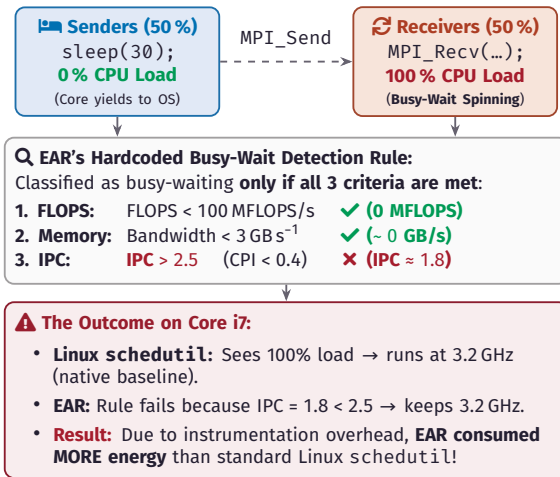
The MPI Sleep Benchmark:

- 50% ranks sleep for 30 s, then send a message.
- 50% ranks call blocking MPI_Recv.
- Total 60 iterations (30 minutes).

CPU Energy (2 Nodes) [kJ]:

Configuration	Run 1	Run 2	Run 3
schedutil	60.51	60.54	60.66
min_energy	62.59	47.33	62.48

→ **EAR consumed more energy** than Linux in 2 of 3 runs.



SERVER CLUSTER RERUN: TESTBED

Cluster Setup:

- **Nodes:** 3 compute nodes, 1 login node.
- **CPU:** Intel[®] Xeon[®] E5-2650 v4 (Broadwell).
 - 2 sockets per node
 - 12 cores per socket
 - Base: 2.2 GHz
 - Idle: 1.2 GHz
- **Memory:** 128 GB DDR4 RAM per node.
- **Workload:** Same 5 workloads as before.
- **Energy Measurement:** Intel RAPL Package + DRAM.
- **No HWP measurements.**

EAR parameters:

- **min_time:**
 - Starting frequency set to 1.4 GHz
 - X set to 0.7
- **min_energy:**
 - Starting frequency set to 2.2 GHz
 - X set to 0.05 and 0.20

RESULTS: EVALUATION ON SERVER HARDWARE

Server Results vs. performance:

Configuration	Time Δ	Energy Δ	NER Score
performance	0.00 %	0.00 %	100.00 %
schedutil	-0.16 %	-0.22 %	99.81 %
min_energy (0.20)	+1.62 %	-1.45 %	100.05 %
min_energy (0.05)	+1.94 %	-1.51 %	100.18 %
monitoring	+0.30 %	+0.42 %	100.36 %
min_time (0.70)	+36.41 %	-14.27 %	108.14 %

Equal-weighted geometric mean across all 5 workloads.

Key findings:

- **Savings Evaporate:**
 - min_time saves **14.27%** energy, but at a **+36.41%** time penalty (NER = 108.14).
 - All EAR policies have NER ≥ 100 .
- **schedutil:** Numerically best, but practically at parity with performance (within noise margin).

BUGFIX: DYNAIS LOOP DETECTION

Adjusting the IPC threshold did not resolve the aforementioned busy-wait detection bug in the MPI Sleep Benchmark.

The root cause was architectural in DynAIS:

- **Role:** Online loop detection from MPI call traces (essential for periodic phase classification).
- **Defect:** AVX2 vectorized routines in `dynais` violated C99 strict-aliasing rules.
- **Impact (GCC 15 -03):** Detection logic is removed by GCC → Always return `NO_LOOP`.
- **(Hot-)Fix:** Compile with `-fno-strict-aliasing` → restored detection.

→ achieving **22.8% energy savings** on the MPI Sleep Benchmark.

CONCLUSION

- **Low Overhead:** EAR runtime overhead confirmed at **0.30 % to 1.36 %**.
- **OS Blindness:** Linux schedutil $\approx$ performance in HPC environments.
- **Energy savings are possible** with min_time by trading off execution time $\rightarrow$ Is it worth it?
- **Several Bugs in EAR:** DynAIS GCC 15 aliasing, thread races, RAPL overflows, ... $\rightarrow$ Setting up EAR might be painful ...
-  [eas4dc/EAR#5](#) min_energy bug issue open since March 2026 

Modified Source Code



[gitup.uni-potsdam.de/ ... /ear-up](https://gitup.uni-potsdam.de/.../ear-up)

Thank you for your attention!

BACKUP: FULL BENCHMARK COMPARISON (DESKTOP VS. SERVER)

Normalized Runtime and Energy Changes vs. Respective Performance Baseline

Benchmark	Configuration	Time (Old %)	Time (New %)	Time Δ (pp)	Energy (Old %)	Energy (New %)	Energy Δ (pp)
	schedutil	100.20%	100.15%	-0.05 pp	99.38%	100.00%	+0.62 pp
	monitoring	101.51%	100.62%	-0.89 pp	101.03%	100.71%	-0.32 pp
	min_time (0.70)	156.52%	101.76%	-54.76 pp	69.89%	101.07%	+31.18 pp
	min_energy (0.05)	101.53%	101.78%	+0.25 pp	100.97%	100.97%	$\pm$ 0.00 pp
	min_energy (0.20)	101.79%	101.83%	+0.04 pp	100.88%	101.26%	+0.38 pp
	schedutil	99.98%	99.14%	-0.84 pp	99.82%	99.30%	-0.52 pp
	monitoring	100.94%	98.30%	-2.64 pp	101.95%	99.14%	-2.81 pp
	min_time (0.70)	110.63%	98.13%	-12.50 pp	63.58%	99.26%	+35.68 pp
	min_energy (0.05)	124.44%	100.61%	-23.83 pp	53.97%	99.42%	+45.45 pp
	min_energy (0.20)	118.56%	98.26%	-20.30 pp	79.81%	99.33%	+19.52 pp
	schedutil	101.35%	99.59%	-1.76 pp	99.54%	99.67%	+0.13 pp
	monitoring	101.02%	100.76%	-0.26 pp	100.42%	100.33%	-0.09 pp
	min_time (0.70)	126.48%	101.84%	-24.64 pp	64.96%	99.12%	+34.16 pp
	min_energy (0.05)	102.38%	102.01%	-0.37 pp	93.34%	99.60%	+6.26 pp
	min_energy (0.20)	108.56%	101.86%	-6.70 pp	92.72%	99.25%	+6.53 pp
	schedutil	100.13%	99.43%	-0.70 pp	99.99%	99.42%	-0.57 pp
	monitoring	101.35%	100.52%	-0.83 pp	101.34%	100.86%	-0.48 pp
	min_time (0.70)	111.29%	101.00%	-10.29 pp	92.66%	100.96%	+8.30 pp
	min_energy (0.05)	101.21%	102.07%	+0.86 pp	101.01%	93.18 %	-7.83 pp
	min_energy (0.20)	109.36%	102.89%	-6.47 pp	91.86%	93.64%	+1.78 pp
	schedutil	98.52%	100.89%	+2.37 pp	98.89%	100.51%	+1.62 pp
	monitoring	102.02%	101.31%	-0.71 pp	102.17%	101.05%	-1.12 pp
	min_time (0.70)	121.76%	103.06%	-18.70 pp	53.21%	98.99%	+45.78 pp
	min_energy (0.05)	106.14%	103.24%	-2.90 pp	99.24%	99.26%	+0.02 pp
	min_energy (0.20)	106.11%	103.27%	-2.84 pp	99.37%	99.29%	-0.08 pp

BACKUP: COMPLETE BUGFIX AND PATCH CATALOG

Summary of Identified Defects, Platform Adaptations, and Patches

ID	Class	Subsystem / Scope	Root Cause and Resolution
F1	EAR Bug	DynAIS Loop Detection	GCC 15 strict-aliasing bug → <code>-fno-strict-aliasing</code> restored detection.
F2	Porting/Bug	Broadwell PMU Counters	4-PMC limit oversubscribed; pinned <code>CPU_CLK_UNHALTED.THREAD</code> (r003c).
F3	Porting/Bug	Broadwell FLOP Events	Incomplete AVX2 256-bit mask in <code>0x0fc7</code> → compact classifier <code>0x3fc7 * 8</code> .
F4	EAR Bug	EARD Learning Mode	Daemon lost policy name in <code>authorized_context.name</code> → fixed crash.
F5	EAR Bug	<code>MPI_THREAD_MULTIPLE</code>	Concurrent threads corrupted timestamps → added non-blocking process mutex.
F6	EAR Bug	RAPL Wraparound Limits	Hardcoded bounds caused fake 19.8 kW power spikes → powercap bounds.
F7	EAR Bug	Busy-Wait CPI Evaluation	Random master caused 3% vs. 23% jitter → node-wide instruction summation.
F8	EAR Bug	Busy-Wait Actuation	Overlapping CPU domains caused <i>last writer wins</i> → unified actuation.
F9	EAR Bug	Rank CPU Affinity	OpenMPI fuse helper inflated mask to 24 cores → <code>cpumask_get_rankmask()</code> .
F10	EAR Bug	<code>min_energy</code> Reference	<code>curr_freq = def_freq</code> ; broke reference projection → removed overwrite.
A1	Platform	Intel IMC / Uncore	Direct Uncore MSRs locked by BIOS → dummy IMC (UFS disabled).
A2	Platform	RAPL Access Interface	Linux powercap prioritized over direct writable MSR access.
A3	Platform	CPUFreq Governor Backend	Switched from <code>intel_cpufreq</code> to <code>acpi_cpufreq</code> for discrete P-states.
C1	Calibration	Busy-Wait CPI Threshold	Broadwell spinning CPI is ~ 0.50 → calibrated threshold from 0.4 → 0.6.

BACKUP: DRAM ENERGY SHARE ACROSS BENCHMARKS

Measured DRAM Contribution to Total RAPL Energy (Package + DRAM)

Configuration	BT-MZ	LU-MZ	SP-MZ	POET	GROMACS	Overall
performance	5.93%	16.17%	12.62%	2.96%	5.41%	8.14%
schedutil	5.98%	16.23%	12.72%	2.96%	5.41%	8.18%
HWP	6.02%	16.14%	12.66%	2.97%	5.41%	8.18%
monitoring	5.94%	15.98%	12.66%	2.96%	5.33%	8.09%
min_time	11.18%	26.54%	21.40%	3.56%	11.68%	12.84%
min_energy (0.05)	5.94%	32.77%	13.69%	2.97%	5.62%	9.59%
min_energy (0.20)	5.96%	21.52%	14.05%	3.53%	5.61%	9.24%

DRAM share of total RAPL energy on the Core i7 testbed.

RANKING STRATEGIES

The Trade-Off Dilemma:

- DVFS trades execution time for energy:
 - **GROMACS: -46.8 % energy** at +21.8 % time.
 - **BT-MZ.D: -30.1 % energy** at **+56.5 % time**.
 - **LU-MZ.D: -46.0 % energy** at +24.4 % time.
- **Core Question:** How to rank strategies when time penalties and energy gains diverge wildly across workloads?

Pitfalls of Conventional Metrics:

- **Raw Energy-Delay Product ($E \cdot T$):** Biased by absolute runtime (hours vs. seconds).
- **Arithmetic Means:** Vulnerable to extreme outliers and scale bias.
- **Requirements for an Ideal Metric:**
 - **Scale Invariance:** Equal weight per benchmark.
 - **Balanced:** 1 % time $\approx$ 1 % energy.
 - **Normalized:** performance $\equiv$ 100 %.

The Objective

A unified, scale-invariant metric that combines normalized runtime and energy across heterogeneous workloads into a single, lower-is-better score.

THE ROLE OF DRAM POWER

DRAM Share of Total RAPL Energy:

Benchmark	Perf	min_time	Δ
POET	2.96 %	3.56 %	+0.6 pp
GROMACS	5.41 %	11.68 %	+6.3 pp
BT-MZ.D	5.93 %	11.18 %	+5.3 pp
SP-MZ.D	12.62 %	21.40 %	+8.8 pp
LU-MZ.D	16.17 %	26.54 %	+10.4 pp

Key Insights:

- **Strong Workload Dependency:** DRAM energy ranges from 3.0 % (POET) up to **26.5 %** (LU-MZ.D).
- **Amplified Share Under DVFS:** Lower core power makes DRAM a **much larger fraction** of total energy.
- **Accumulation Over Time:** Longer runtimes continuously accumulate baseline DRAM draw and erode CPU savings.

ANALYSIS: min_energy BUG

Theoretical Expectation:

- Configured penalty (e.g. 5%) is intended to cap maximum slowdown relative to the baseline frequency (T_{def} at 3.2 GHz).

Observed Anomaly (LU-MZ.D):

- min_energy (0.05) caused a **+24.44 % slowdown** (nearly 5× higher than permitted!).
- Relaxing penalty to min_energy (0.20) ironically resulted in *less* slowdown (+18.56 %).

Possible Cause (Moving Reference Target):

- Overwriting `curr_freq = def_freq`; in `min_energy.c` bypassed backward projection.
- EAR assumed data was collected at default frequency → set $T_{\text{ref}} = T_{\text{curr}}$ (already degraded!).
- Compounding Penalty:** Each step allowed +5 % on top of the already slowed-down runtime ($T_{\text{limit}} = 1.05 \times T_{\text{curr}}$) → ratcheted frequency down to the lowest P-state.

Upstream Issue:

-  [eas4dc/EAR#5](https://github.com/eas4dc/EAR/issues/5)