DURHAM DAYS HPC 2026

QUEEN'S UNIVERSITY BELFAST

A Frequency-Invariant Approach to Energy-Efficient Computing

Syed I. Tauhidi, Hans Vandierendonck

Kelvin Living Lab | Queen's University Belfast

Motivation

Global data centre energy consumption is projected to reach 1100 to 2000 TWh by 2030

This energy demand will surpass the aggregate usage of **Canada** or **Japan**

The End of Dennard Scaling

We must rely on **software-level solutions** to manage this computing energy footprint

DURHAM DAYS HPC 2026

QUEEN'S UNIVERSITY BELFAST

Default Linux Governors

Governors are **software that manages frequency and energy** in a CPU.

Traditional Heuristics

Linux governors like **ondemand** and **schedutil** rely on coarse CPU utilization metrics.

- **Race-to-idle philosophy:** Operates with the goal of finishing tasks quickly to enter low-power states.
- **The Core Assumption:** High CPU utilization always implies productive computation.

The Efficiency Collapse

This logic fails during **memory-bound phases** where the CPU is stalled.

- **Memory Wall:** Core stalls while waiting for data from cache misses.
- **False Utilization:** CPU utilization stays near 100% during these idle stalls.
- **Wasted Energy:** Systems blindly boost frequency to "wait faster," wasting significant power.

Gaps in the literature

Previous software-based solutions have significant **deployment barriers**:

Intrusive Modifications

Requires extensive changes to application source code for integration.

Expensive Training

Relies on long, computationally expensive offline training phases before deployment.

Runtime Overhead

Usage of profiling tools introduces prohibitive performance penalties during execution.

The URJA Framework

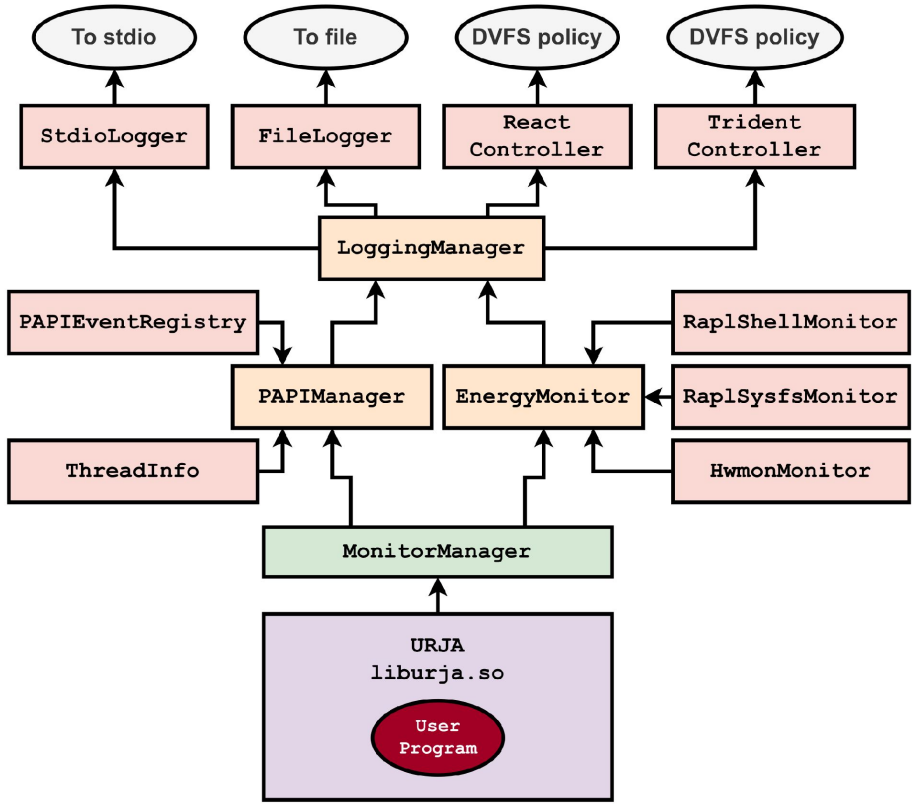
URJA is a **non-intrusive, user-space DVFS framework** built for online operation.

Core Characteristics

Non-intrusive Integration: Operates without modifying the application binary or source code.

Strictly Online Operation: Adapts to runtime behavior without needing offline training.

Hardware-Aware: Utilizes lightweight hardware performance counters.



DURHAM DAYS HPC 2026

QUEEN'S UNIVERSITY BELFAST

MPKI as a Metric

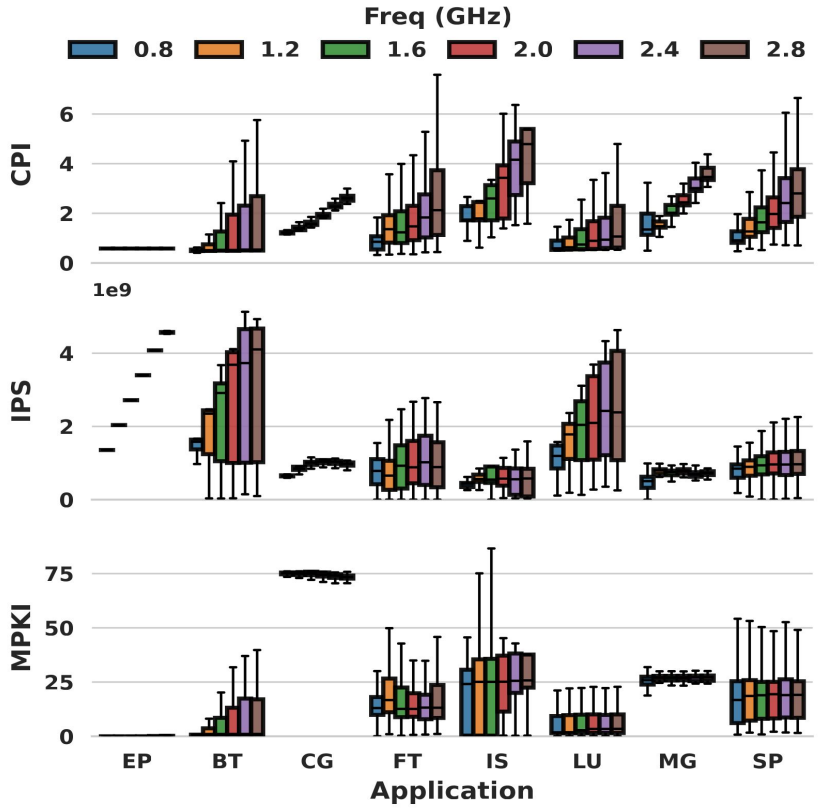
MPKI is a **robust, frequency-invariant** alternative to traditional performance counters.

Why not CPI or IPS?

- **Frequency-variant,** i.e., value change based on clock speed (frequency).
- Difficult to use as independent variables for DVFS.

MPKI Advantages:

- Reflects workload structure, not hardware speed.
- Enables phase detection without recalibration.



DVFS Controller

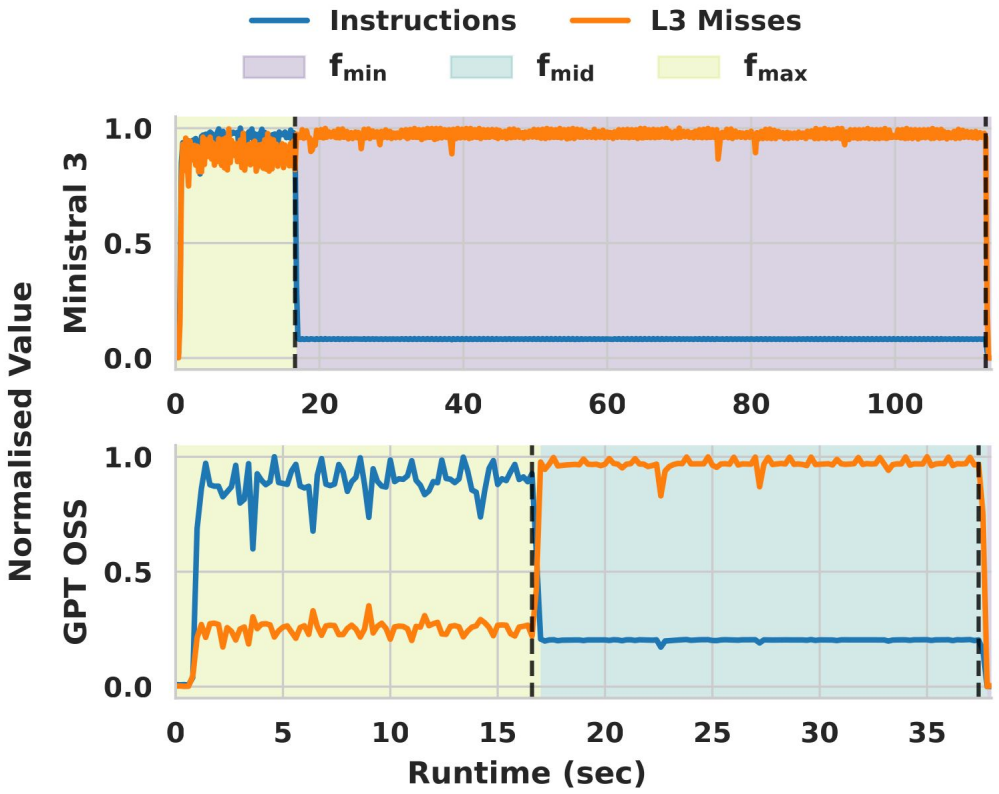
Ternary control strategy **with three discrete frequency caps**.

Frequency Strategy:

- **Caps:** f_{max}, f_{min}, and f_{mid} (buffer).
- **Buffer** amortises switching penalties.

Key Characteristics

- **Adaptive Window:** Reacts instantly to phase shifts while preventing thrashing
- **Hysteresis:** Includes a margin for stability



Choosing a Profiling Interval

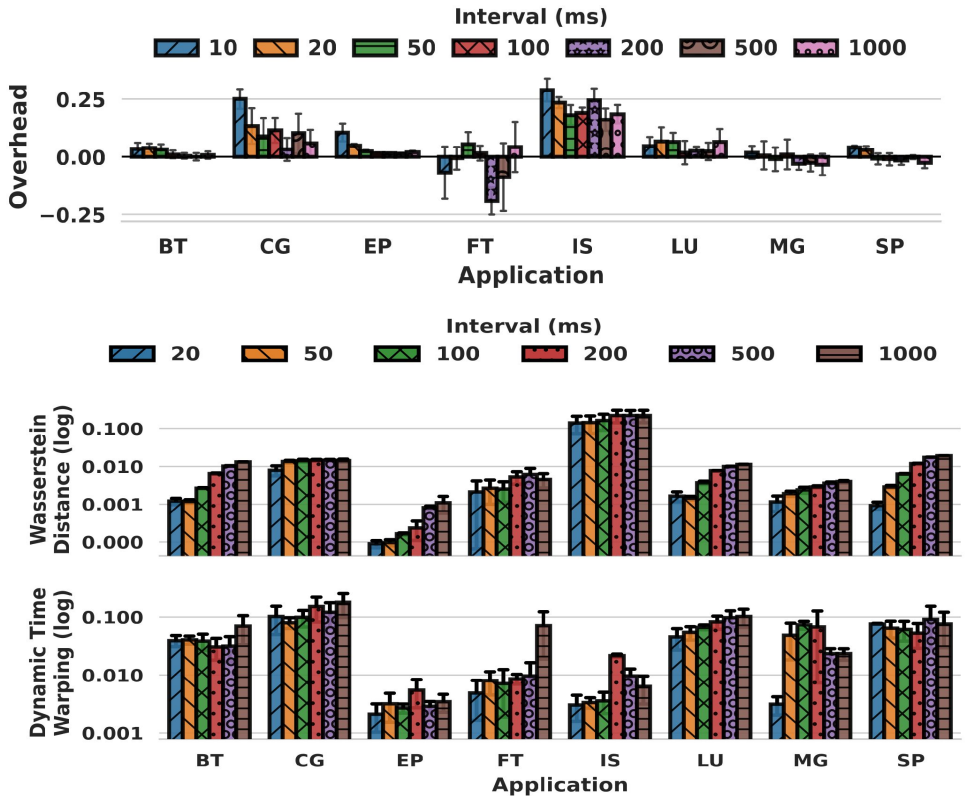
Balancing **runtime overhead** vs **signal fidelity**.

TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) Evaluation:

- **Runtime Overhead**
- **Wasserstein Distance:** Distribution error
- **DTW Score:** Temporal alignment errors

200 ms optimal

Score: 0.786 (vs. 50ms: 0.76, 100ms: 0.64)



Choosing Decision Thresholds

Thresholds (T_L and T_U) derived from the distribution of MPKI data.

k-means Clustering (k=3)

- Represents three ternary states
- Clusters optimized for MPKI distribution

Noise Prevention & Weighting

Confidence Weight assigned based on compactness using **Silhouette score** & **standard deviation**

Final thresholds calculated as the **weighted average** between cluster centroids.

Brute-force Sensitivity Analysis

Grid search used to validate the global optimality of chosen thresholds.

Performance on HPC Workloads

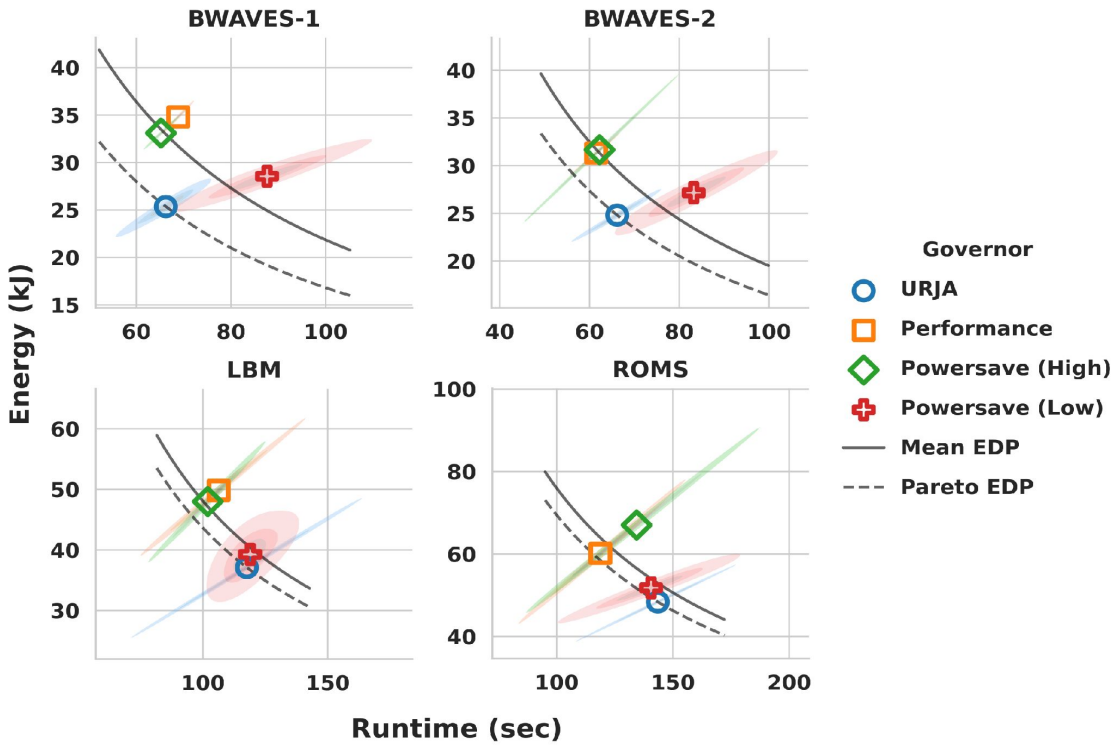
Evaluated against **Intel drivers** using **NAS Parallel Benchmarks** and **SPEC CPU2017**.

Pareto Optimization

- Moves execution toward the **optimal EDP Pareto frontier**.
- Achieves **~10% improvement** in EDP over baseline governors.

29% Energy Savings

With a minimal overhead of only ~1%.



Efficacy in AI Inference

DURHAM DAYS HPC 2026

QUEEN'S UNIVERSITY BELFAST

Tested modern AI workloads using **Ministral 3** and **GPT-OSS**.

Two-Phase Inference Architecture

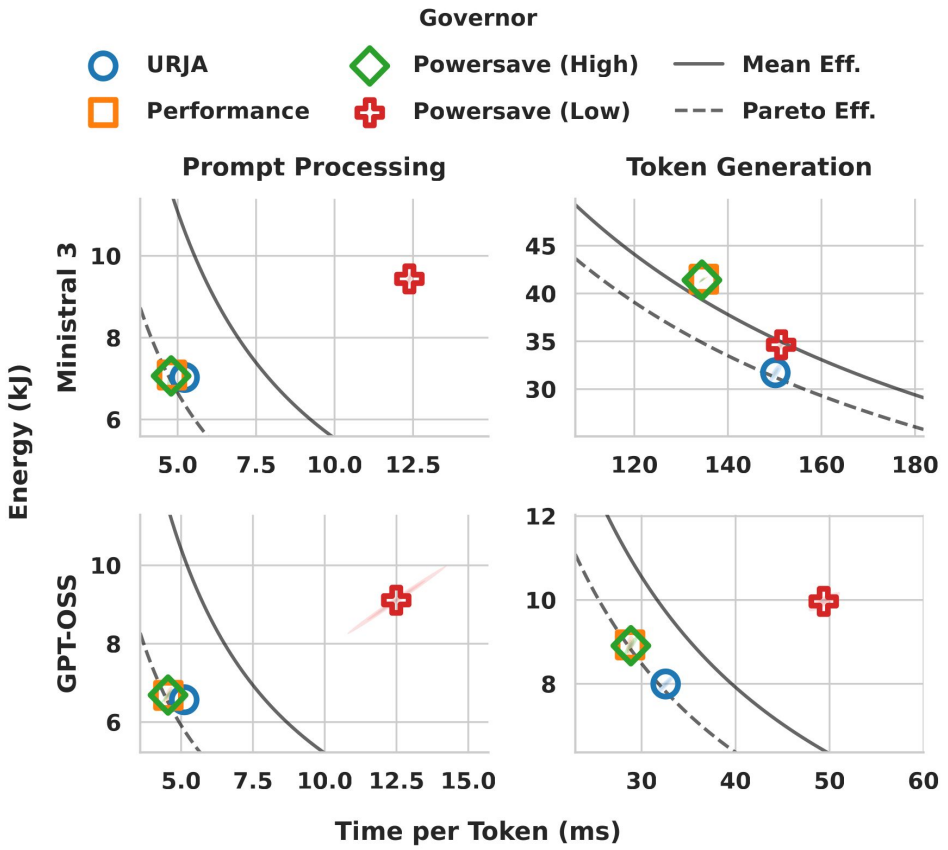
- **Prompt Processing:** Compute-bound phase.
- **Token Generation:** Memory-bound phase.

Frequency Management

Maintains $\mathbf{f}_{max}$ for prompt processing, lowers during decoding to save power.

~21% Energy Savings

Reducing Ministral 3 consumption from ~48 kJ to ~38 kJ.



Deployment & Conclusion

Available via a **Slurm SPANK plugin** for seamless activation.

Shared Cluster Advantages

- No source code or binary alterations required.
- Operates entirely in user-space.
- No disruption to workflows or thread pinning.

Future Work

Addressing challenges under **rapid heterogeneity** in performance and thread counts.

Conclusion

URJA's lightweight, micro-architecture-aware DVFS delivers major sustainability gains for next-generation data centers.

DURHAM DAYS HPC 2026

QUEEN'S UNIVERSITY BELFAST

Thank You

Research Group Blog:
blogs.qub.ac.uk/dipsa/

Personal LinkedIn:
linkedin.com/in/ibtisamtauhidi/