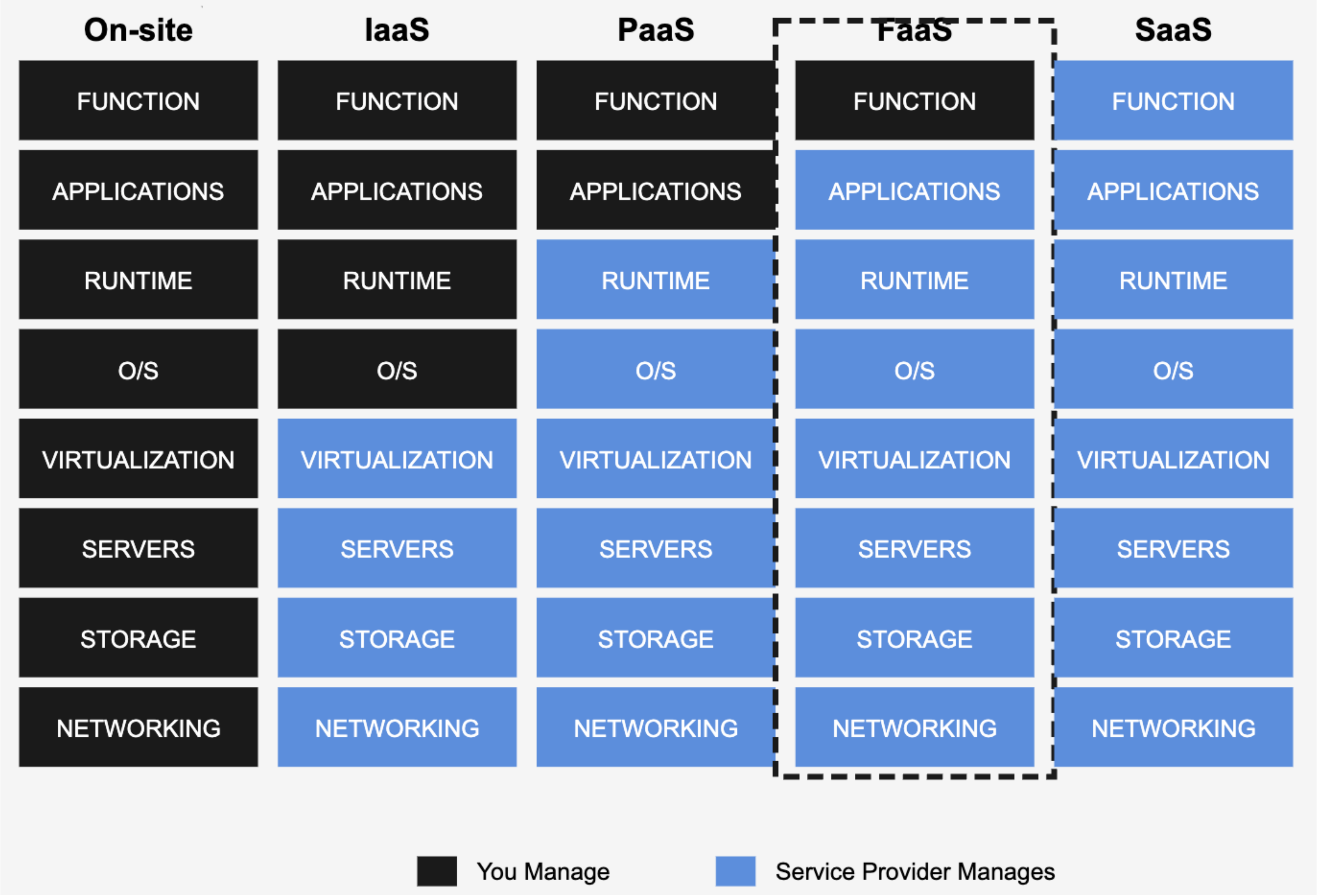
Meta TaskWave

Meta-Selection for Hybrid Dynamic Scheduling in FaaS Environments

Theresa To, MSc Computing and Information Systems

What is Function-as-a-Service?



What is Function-as-a-Service?

Event Driven

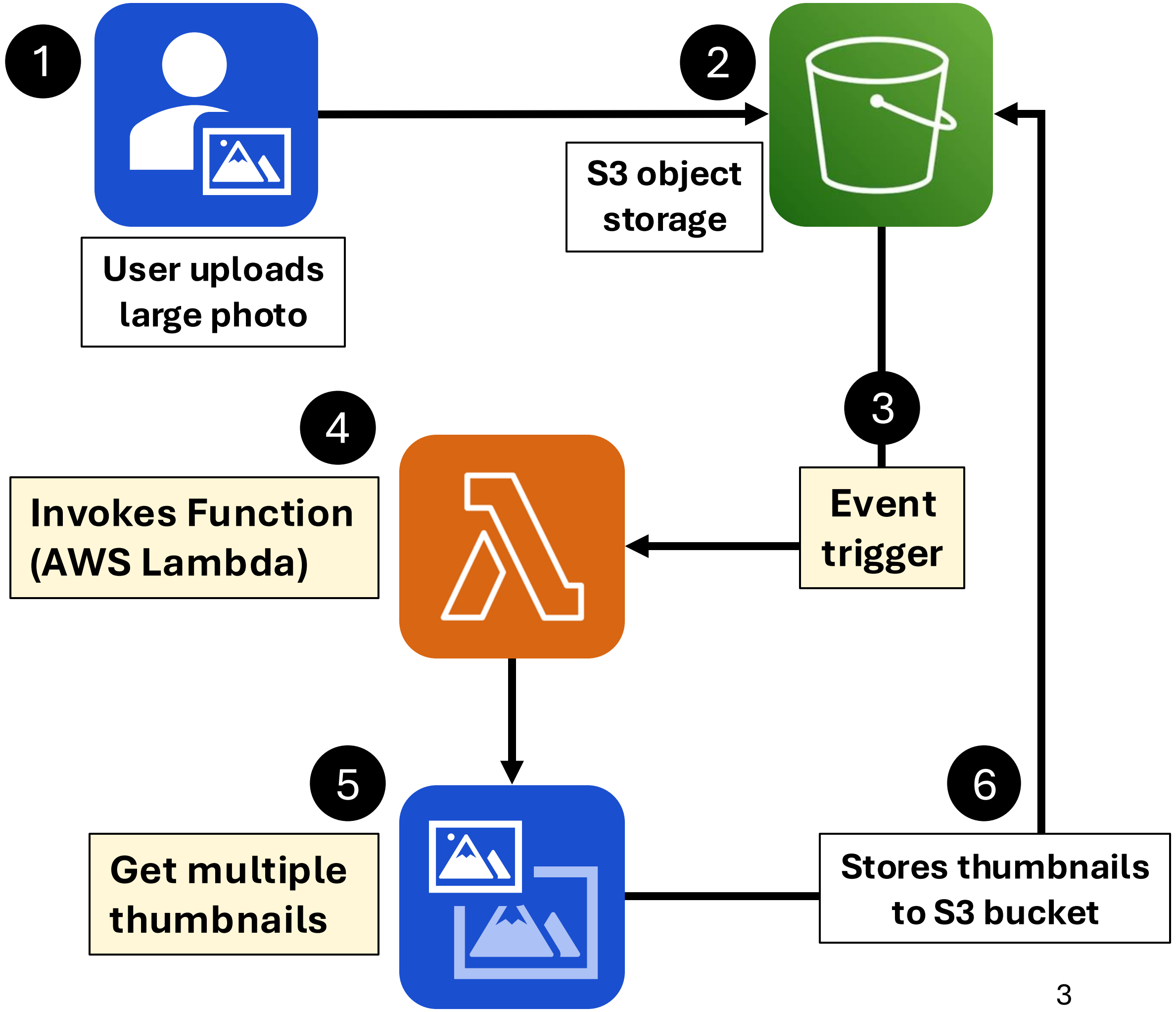
It's a micro-architecture responsive service that is only invoked when functions are called

Zero-Operational Requirements

Upload your function to a cloud provider and instantly access elastic scaling for computing, storage, and deployment.

Pay-per-execution

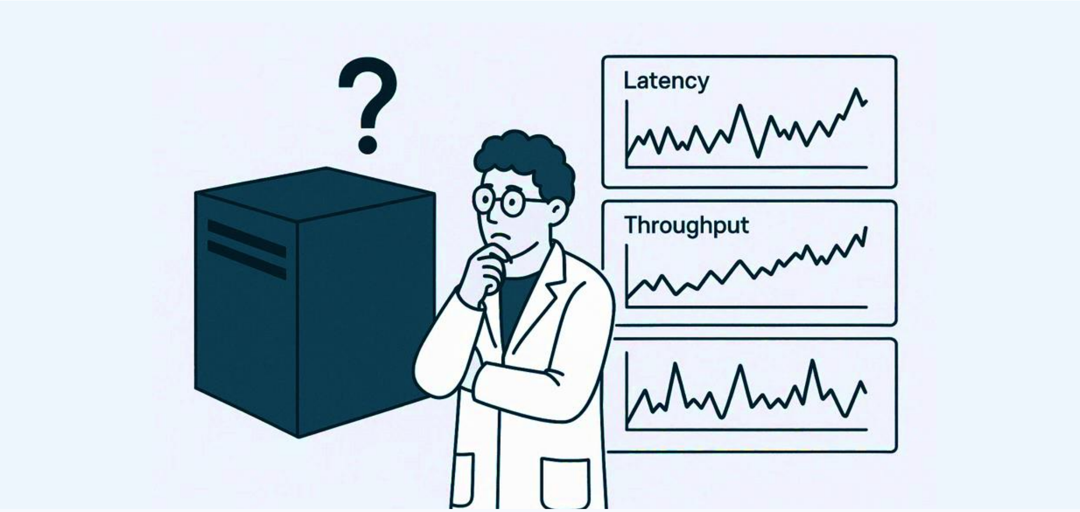
It's cost effective — you only pay per millisecond of compute time when your function is called

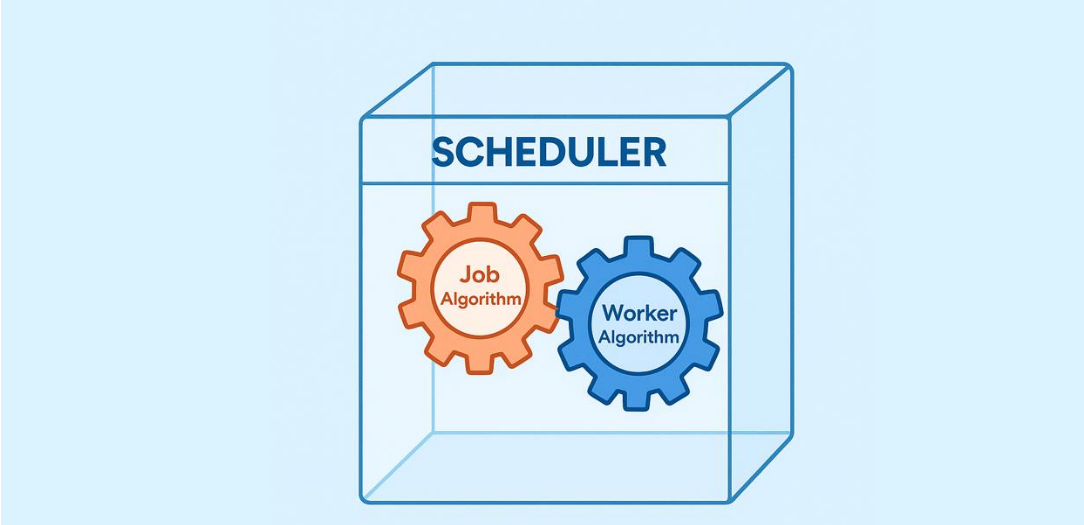


The Problem

Growing Market	**27.8% annually → $62.5B** Expected year-on-year growth for FaaS Market (Grand View Research Inc, 2024)
Proprietary	**Black box platforms — no transparency** No systematic way to choose optimal scheduling algorithms
So what?	**Need a transparent framework** For algorithm selection and systemic behaviour understanding

What's Missing?





Research Gap

While adaptive scheduling exists in production FaaS platforms, **academic research lacks a transparent framework** for evaluating hybrid scheduling strategies.

Our contribution

First **systematic benchmarking** of hybrid scheduling combinations via a **literature backed simulation** with **interpretable performance analysis**.

Research Questions

Main Research Question

How do scheduling algorithms differ in observable behaviours within FaaS platforms, and when do these differences matter for performance?

Sub-Questions

1. Can we **build a framework** to characterise scheduling trade-offs despite system-level noise?
2. What **distinct behavioural patterns** emerge when algorithms are properly characterised?
3. How can these patterns **inform adaptive scheduling** decisions?

System Design & Methodology

Foundations of the distributed system through literature

7

Distributed System Architecture

Job Generator (1)

Synthetic workload creation with configurable patterns

- Batch size control
- Inter-arrival timing
- Priority distribution

Central Scheduler

Hybrid algorithm implementation and job assignment

- 3 job algorithms
- 5 worker algorithms
- 15 combinations total

Worker Pool (N)

Heterogeneous workers with realistic constraints

- Capacity variation
- Network simulation
- Cold start penalties

- Mahgoub, A., Yi, E.B., Shankar, K., Minocha, E., Elnikety, S., Bagchi, S. & Chaterji, S. (2022) WISEFUSE: Workload Characterization and DAG Transformation for Serverless Workflows. Proc. ACM Meas. Anal. Comput. Syst. 6 (2), 26:1-26:28. doi:10.1145/3530892.
- Shahrad, M., Fonseca, R., Goiri, Í., Chaudhry, G., Batum, P., Cooke, J., Laureano, E., Tresness, C., Russinovich, M. & Bianchini, R. (2020) Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider. In: 2020 pp. 205–218. https://www.usenix.org/conference/atc20/presentation/shahrad.
- Vandebon, J., Coutinho, J.G.F. & Luk, W. (2021) Scheduling Hardware-Accelerated Cloud Functions. Journal of Signal Processing Systems. 93 (12), 1419–1431. doi:10.1007/s11265-021-01695-7.
- Amazon Web Services, Inc. (2025) AWS Lambda - Developer Guide. https://docs.aws.amazon.com/pdfs/lambda/latest/dg/lambda-dg.pdf.
- Chu, K., Li, X. & Qin, X. (2025) Optimizing Resource Utilization in Edge Computing Environment with Dynamic Load Balancing Scheduling Algorithm. In: 2025 6th International Conference on Computer Science, Engineering, and Education (CSEE). February 2025 pp. 49–56. doi:10.1109/CSEE64583.2025.00017.
- Copik, M., Kwasniewski, G., Besta, M., Podstawski, M. & Hoefler, T. (2021) SeBS: a serverless benchmark suite for function-as-a-service computing. In: Proceedings of the 22nd International Middleware Conference. Middleware '21. 2 October 2021 New York, NY, USA, Association for Computing Machinery. pp. 64–78. doi:10.1145/3464298.3476133.
- Mahgoub, A., Yi, E.B., Shankar, K., Minocha, E., Elnikety, S., Bagchi, S. & Chaterji, S. (2022) WISEFUSE: Workload Characterization and DAG Transformation for Serverless Workflows. Proc. ACM Meas. Anal. Comput. Syst. 6 (2), 26:1-26:28. doi:10.1145/3530892.
- Thinakaran, P., Gunasekaran, J.R., Sharma, B., Kandemir, M.T. & Das, C.R. (2019) Kube-Knots: Resource Harvesting through Dynamic Container Orchestration in GPU-based Datacenters. In: 2019 IEEE International Conference on Cluster Computing (CLUSTER). September 2019 pp. 1–13. doi:10.1109/CLUSTER.2019.8891040.

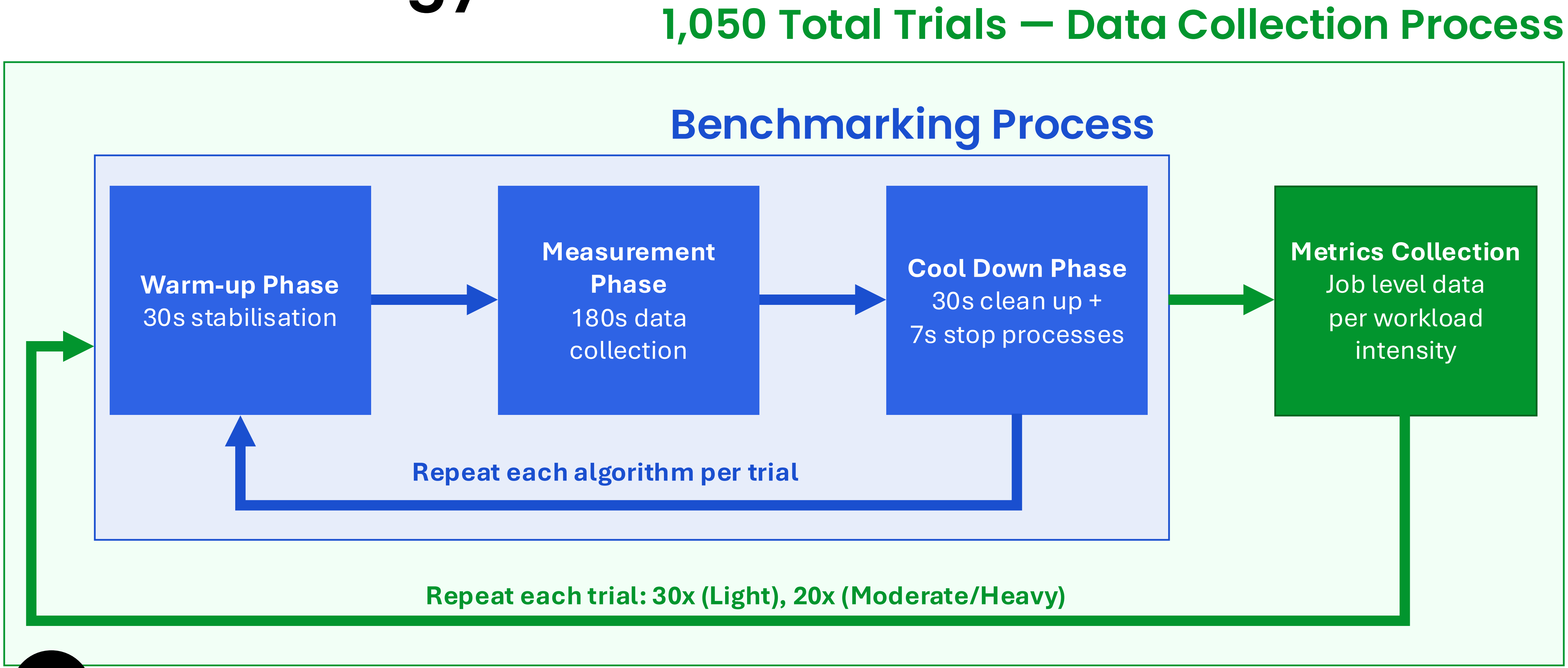
8

Distributed System Architecture

Worker Algorithm

Job Algorithm	Random	Round Robin	Least Loaded Fair	Fastest Worker Fair	Network Optimal Fair
Round Robin	RR + Ran	RR + RR	RR + LLF	RR + FWF	RR + NOF
EDF	EDF + Ran	EDF + RR	EDF + LLF	EDF + FWF	EDF + NOF
Urgency First	UF + Ran	UF + RR	UF + LLF	UF + FWF	UF + NOF

Methodology



Experimental Design

What we test

What we measure

What we control for

Variable Type	Variable	Values/Range	Literature Support
Independent	Algorithm Combinations	15 hybrid strategies	Foundational algorithms
Independent	Workload Intensity	**Light:** 8.0 – 15.0s **Moderate:** 2.0 – 5.0s **Heavy:** 0.5 – 1.5s	Joosen et al. (2023); Shahrad et al. (2020)
Dependent	Job Level Metrics	Total **Response** Time, Job **Processing** Time, **Cooling** Penalties, **Queue** Penalties	Copik et al. (2021)
Control	Worker Configuration	**4 WORKERS** 600-1200 MB ~ 1vCPU	Shahrad et al. (2019); AWS Lambda standards
Control	Number of Trials per Algorithm	**Light :** 30 trials **Moderate:** 20 trials **Heavy:** 20 trials	

Anatomy of E2E Execution

$$\text{Total Response Time} = \overbrace{\text{Job Processing Time} + \overbrace{\text{Cooling Penalties} + \text{Queue Delays}}^{\text{Latencies}}}^{\text{End-to-End Execution}}$$

If elastic scaling works
Queue Delay → 0

In properly-provisioned FaaS, elastic scaling would mean:

- There are enough backup worker containers
- And there are negligible queue delays

Anatomy of E2E Execution

$$\underbrace{\textbf{Total Response Time}}\ = \overbrace{\textbf{Job Processing Time} + \overbrace{\textbf{Cooling Penalties} + \cancel{\textbf{Queue Delays}}}^{\textbf{Latencies}}}^{\textbf{End-to-End Execution}}$$

If elastic scaling works
Queue Delay → 0

So Total Response Time (TRT) comes from either:

- Processing time variations
- Cooling penalty variations

$$\text{Total Response Time} = \text{Job Processing Time} + \overbrace{\underbrace{\text{Cooling Penalties}}_{\text{Full initialisation + Partial Initialisation}} + \text{Queue Delays}}^{\text{Latency}}$$

Cooling Penalties

Delay experienced when a **container** must be **initialised** (cold start) or **reactivated** (warm start) **before executing the function**, rather than running immediately from an active state.

In our simulation:

- **Cold start:** Full container boot (300ms)
- **Warm penalty:** Partial reactivation (=>60ms)
- **Hot state**: No penalty (0ms)

Hypothesis: Cooling penalties drive FaaS performance variance

Statistical Evaluation Framework

Distribution Analysis
Coefficient of Variation

Variance Component Analysis
Two-/Three-way ANOVA
Cohen’s d

Categorical Association Analysis
Chi-square
Cramérs V
Standardised Residuals

Thermal Efficiency Analysis
Thermal Degradation %
Thermal Stability Chain
Thermal Efficiency Ratio

Cross Validation (Categorical Analysis x Thermal Efficiency)

Pearson’s r,
Statistical Significance
Trend Line

Results

Results

1. Distribution and Variance Component

What's causing variation?

2. Categorical Association

Where do algorithms differ in penalties?

3. Thermal Efficiency

How do they create these differences

4. Cross Validation

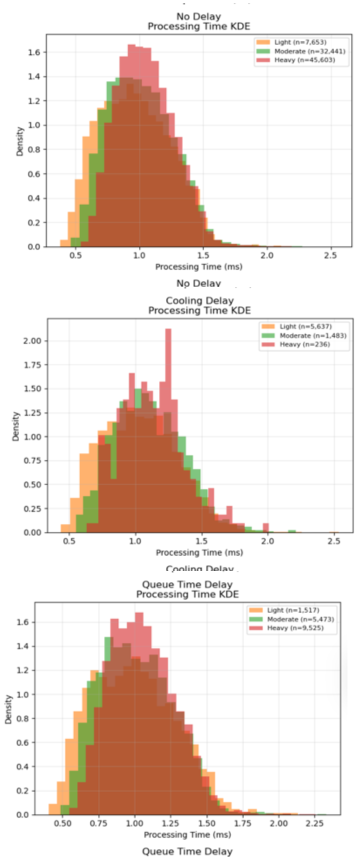
Testing our hypothesis

Total Response Time = Job Processing Time + Cooling Penalties + ~~Queue Delays~~

CV & ANOVA Discovery

Penalties vs Processing

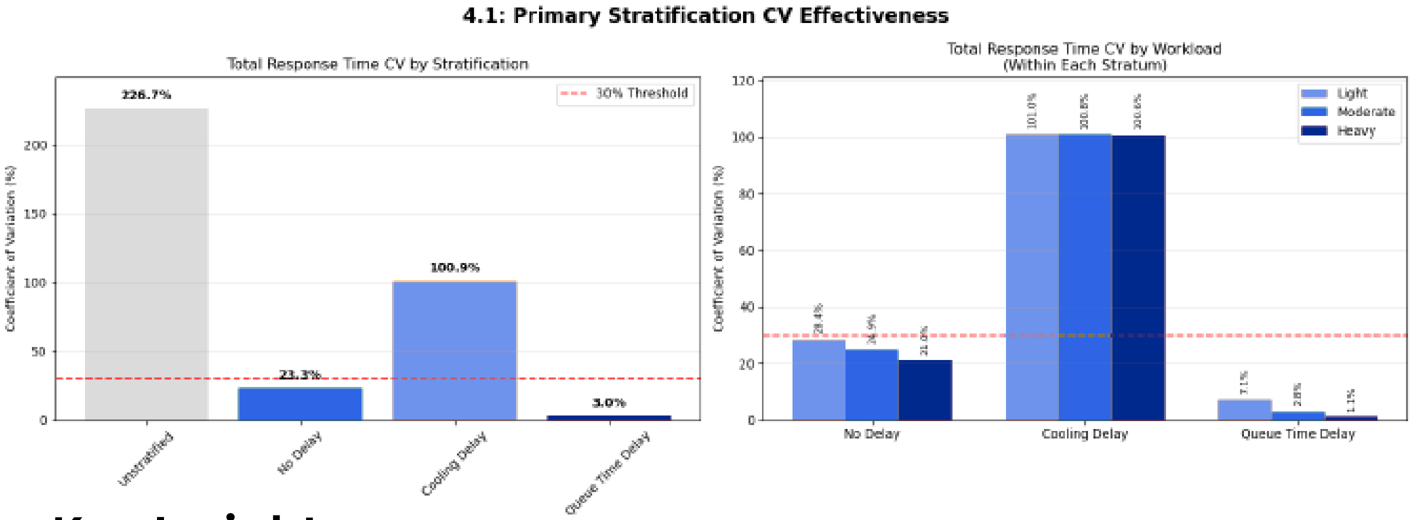
Job Processing Time KDE Plots



No delay

Cooling Delay

Queue Delay



Key Insights

✓ Total Response Time CV: 227%

✓ After controlling penalties: 23%

✓ Processing time effect: <2%

✓ Algorithms **statistically significant**

😱 **Max impact: 0.083ms** only!

Conclusion: Penalties drive 98% of variance

Processing Time is Uniform

ANOVA confirms negligible algorithmic effect

Penalties Drive All Variance

Results

Penalties must be the driver of variation

1. Distribution and Variance Component

What's causing variation?

2. Categorical Association

Where do algorithms differ in penalties?

3. Thermal Efficiency

How do they create these differences

4. Cross Validation

Testing our hypothesis

Chi-Square, Cramérs V, Standardised Residuals

Total Response Time = Job Processing Time + Cooling Penalties + ~~Queue Delays~~

Algorithm-Penalty Associations

Hybrid Algorithm – Categorical Analysis Table

Workload	Chi-Square Result	Sample Size	Significance	Cramér's V	Effect Size
Overall	$\chi^2 = 825.19$, $p = 0.0000$	109,568 jobs	SIGNIFICANT	0.0614	Negligible
Light	$\chi^2 = 3277.62$, $p = 0.0000$	14,807 jobs	SIGNIFICANT	0.3327	Moderate
Moderate	$\chi^2 = 382.82$, $p = 0.0000$	39,397 jobs	SIGNIFICANT	0.0697	Negligible
Heavy	$\chi^2 = 805.86$, $p = 0.0000$	55,364 jobs	SIGNIFICANT	0.0853	Negligible

Hybrid Algorithms show statistical significance on penalties

But whether it's the job queueing or worker queueing, we don't know.

Job algorithms?

Worker algorithms?

Are there patterns?

Chi-Square, Cramérs V, Standardised Residuals

Algorithm-Penalty Associations

Job Algorithm – Categorical Analysis Table

Workload	Chi-Square Result	Sample Size	Significance	Cramér's V	Effect Size
Overall	$\chi^2 = 3.30$, $p = 0.5090$	109,568 jobs	NOT SIGNIFICANT	0.0039	Negligible
Light	$\chi^2 = 0.96$, $p = 0.9164$	14,807 jobs	NOT SIGNIFICANT	0.0057	Negligible
Moderate	$\chi^2 = 0.47$, $p = 0.9764$	39,397 jobs	NOT SIGNIFICANT	0.0024	Negligible
Heavy	$\chi^2 = 4.26$, $p = 0.3726$	55,364 jobs	NOT SIGNIFICANT	0.0062	Negligible

Job Algorithms have no statistical effect on cooling penalties

All p-values show that job algorithm choices don't affect whether it will incur penalties on processing — and we now realise that container heterogeneity when looking at the whole system affects this.

Job algorithms: No Association

Worker algorithms?

Are there patterns?

Chi-Square, Cramérs V, Standardised Residuals

Algorithm-Penalty Associations

Worker Algorithm – Categorical Analysis Table

Workload	Chi-Square Result	Sample Size	Significance	Cramér's V	Effect Size
Overall	$\chi^2 = 781.94$, $p = 0.0000$	109,568 jobs	SIGNIFICANT	0.0597	Negligible
Light	$\chi^2 = 3255.30$, $p = 0.0000$	14,807 jobs	SIGNIFICANT	0.3315	Moderate
Moderate	$\chi^2 = 352.59$, $p = 0.0000$	39,397 jobs	SIGNIFICANT	0.0669	Negligible
Heavy	$\chi^2 = 741.57$, $p = 0.0000$	55,364 jobs	SIGNIFICANT	0.0818	Negligible

Meanwhile, Worker Algorithms show strong correlation to startup delays.

All p-values show that worker algorithm choices (**$p < 0.0001$**) significantly affect cold start penalties across every workload condition — revealing that container assignment is the primary performance driver.

Chi-Square, Craméřs V, Standardised Residuals

Algorithm-Penalty Associations

Stat. Sig. and Effect

Job

Workload	Chi-Square Result	Sample Size	Significance	Cramér's V	Effect Size
Overall	$\chi^2 = 3.30$, $p = 0.5090$	109,568 jobs	NOT SIGNIFICANT	0.0039	Negligible
Light	$\chi^2 = 0.96$, $p = 0.9164$	14,807 jobs	NOT SIGNIFICANT	0.0057	Negligible
Moderate	$\chi^2 = 0.47$, $p = 0.9764$	39,397 jobs	NOT SIGNIFICANT	0.0024	Negligible
Heavy	$\chi^2 = 4.26$, $p = 0.3726$	55,364 jobs	NOT SIGNIFICANT	0.0062	Negligible

Worker

Workload	Chi-Square Result	Sample Size	Significance	Cramér's V	Effect Size
Overall	$\chi^2 = 781.94$, $p = 0.0000$	109,568 jobs	SIGNIFICANT	0.0597	Negligible
Light	$\chi^2 = 3255.30$, $p = 0.0000$	14,807 jobs	SIGNIFICANT	0.3315	Moderate
Moderate	$\chi^2 = 352.59$, $p = 0.0000$	39,397 jobs	SIGNIFICANT	0.0669	Negligible
Heavy	$\chi^2 = 741.57$, $p = 0.0000$	55,364 jobs	SIGNIFICANT	0.0818	Negligible

Key Insights

✓ Hybrids significant ($p<0.001$)
✓ Jobs: No effect ($p>0.05$)
✓ Workers: Drive penalties
Light workload: V = 0.33 (moderate effect)
Moderate workload: V = 0.07 (small effect)

Conclusion:

Confirms **Light-to-Moderate** as **FaaS operating zone**

Percentage of jobs with cooling delays:

- Light = 40% cold starts
- Moderate = 3.8%)

Job algorithms: No Association

Worker algorithms: Penalty drivers

Are there patterns?

Chi-Square, Cramérs V, Standardised Residuals

Algorithm-Penalty Associations

Cooling Penalties as % of Total Jobs:

Light:	38.1%	5,637 out of 14,807
Moderate:	3.8%	1,483 out of 39,397
Heavy:	0.4%	236 out of 55,364

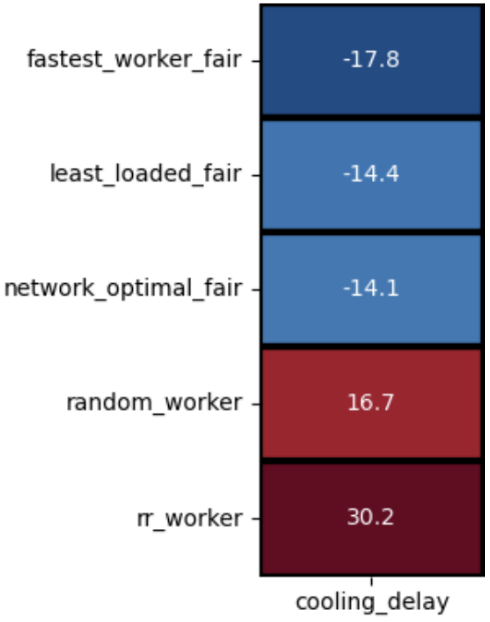
Job algorithms: No Association

Worker algorithms:
Penalty drivers

**Standardised residuals:
Penalty patterns uncovered**

Chi-Square, Cramérs V, Standardised Residuals

Algorithm-Penalty Associations



Sophisticated ✅

- Fastest Worker Fair: -17.8
- Least Loaded Fair: -14.4
- Network Optimal Fair: -14.1

Simple ❌

- Random Worker: 16.7
- Round Robin Worker: 30.2

Light Workload

Simple ✅

- Random Worker: -7.4
- Round Robin Worker:- 7.7
- Least Loaded Worker: -7.7

Too saturated!
Add more containers

Heavy Workload

Job algorithms: No Association

Worker algorithms:
Penalty drivers

**Standardised residuals:
Penalty patterns uncovered**

Results

Penalties must be the driver of variation

Worker algorithms drive penalties

1. Distribution and Variance Component

What's causing variation?

2. Categorical Association

Where do algorithms differ in penalties?

3. Thermal Efficiency

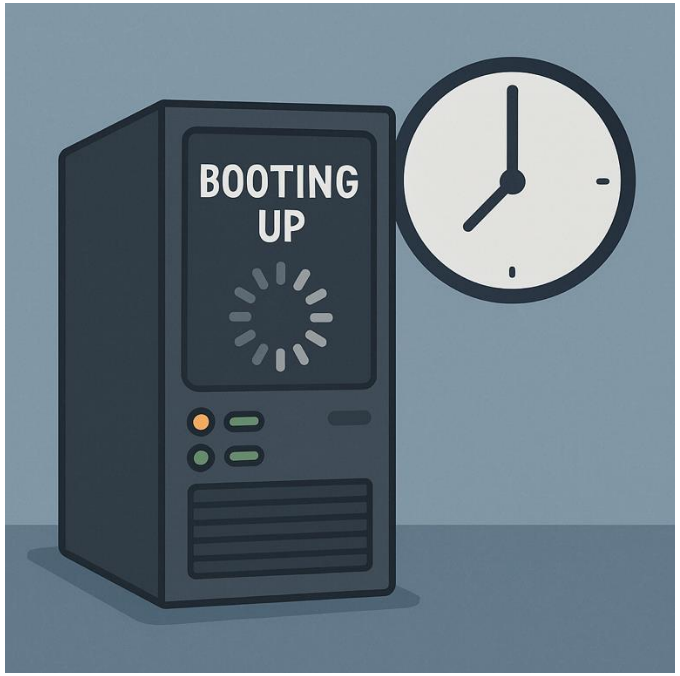
How do they create these differences

4. Cross Validation

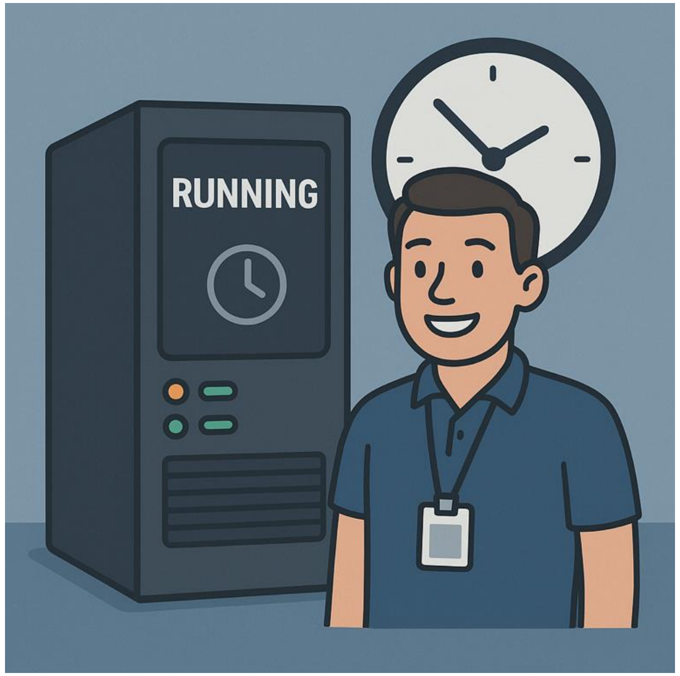
Testing our hypothesis

Container Degradation, Stability Chain, & Thermal Efficiency Ratio

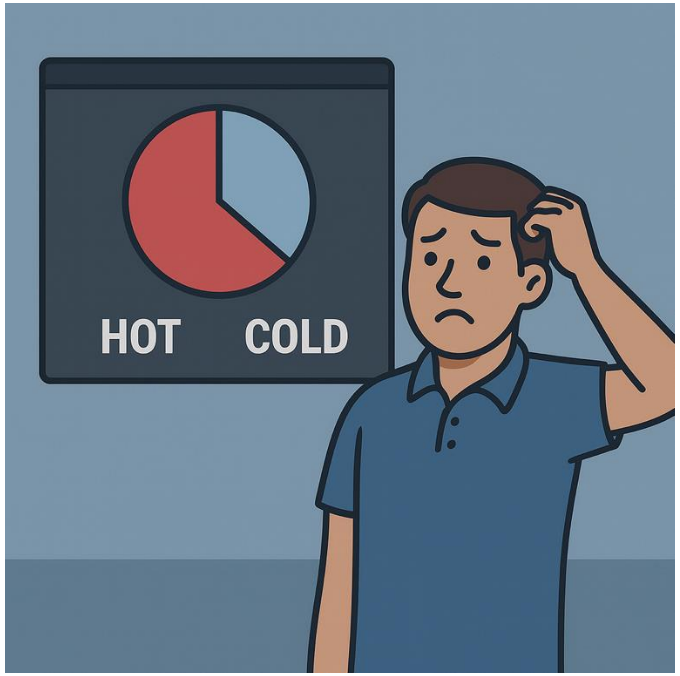
How do they create differences?



Thermal Degradation
- How often containers fail



Thermal Stability Chain
- How long they stay hot for



Thermal Efficiency (TE)
- The ratio between them

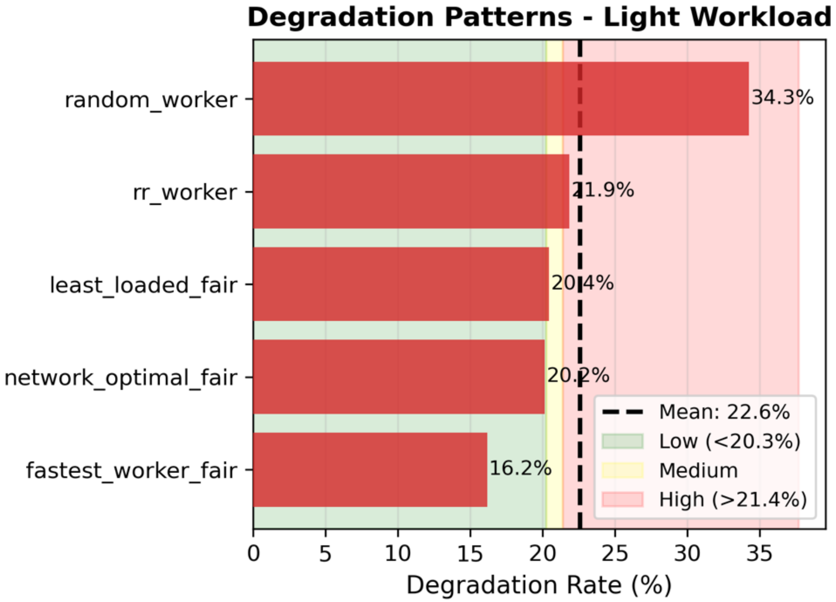
Thermal Degradation Rate

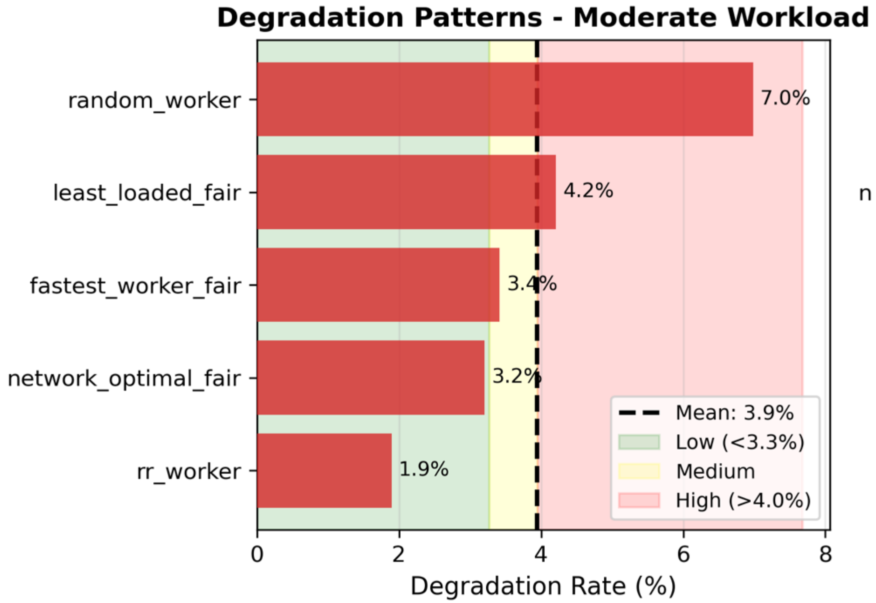
Thermal Stability Chain

Thermal Efficiency Ratio

Thermal Degradation

Container restarts ↑ = wasted warmth | Red zone = 67th percentile+





Pattern:
Smart selection
to
Equal distribution

Who restarts the most?

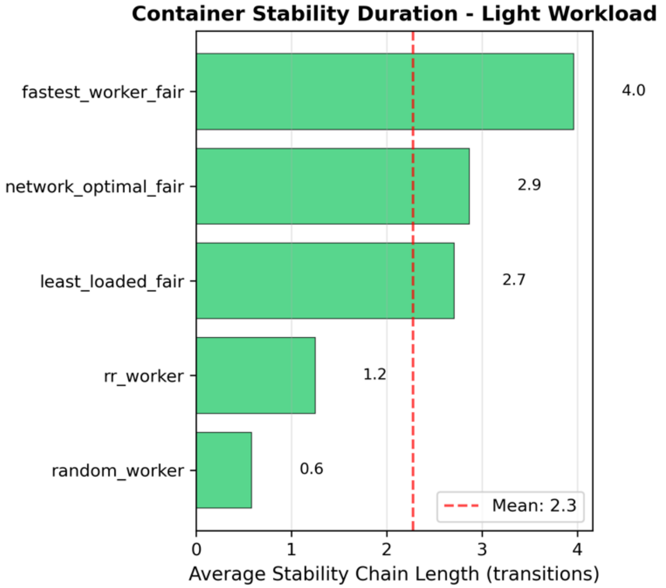
- **Light**: Random worst, FWF best
- **Moderate**: Random worst, RR best

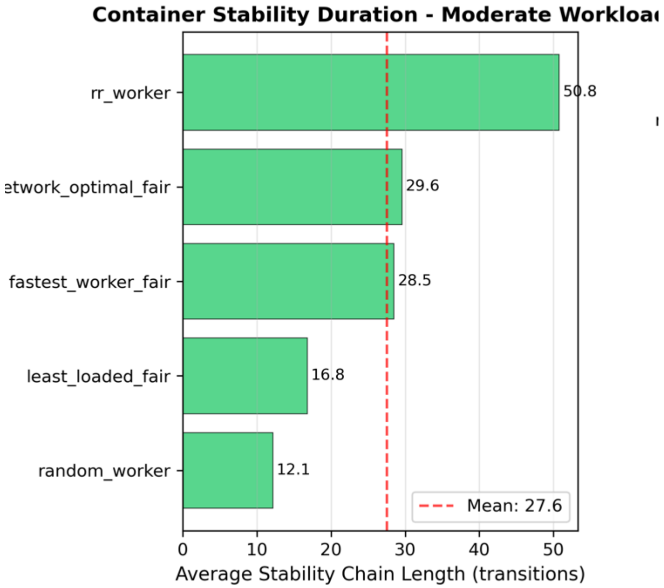
Thermal Degradation Rate | Thermal Stability Chain | Thermal Efficiency Ratio

28

Thermal Stability Chain

Longer Chains ↑ = better container utilisation | Red line = Mean chain across all jobs



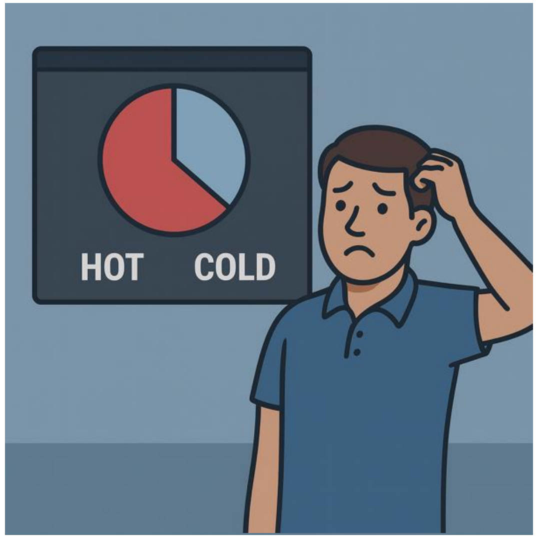


Pattern:
Intelligent reuse
to
Fair rotation

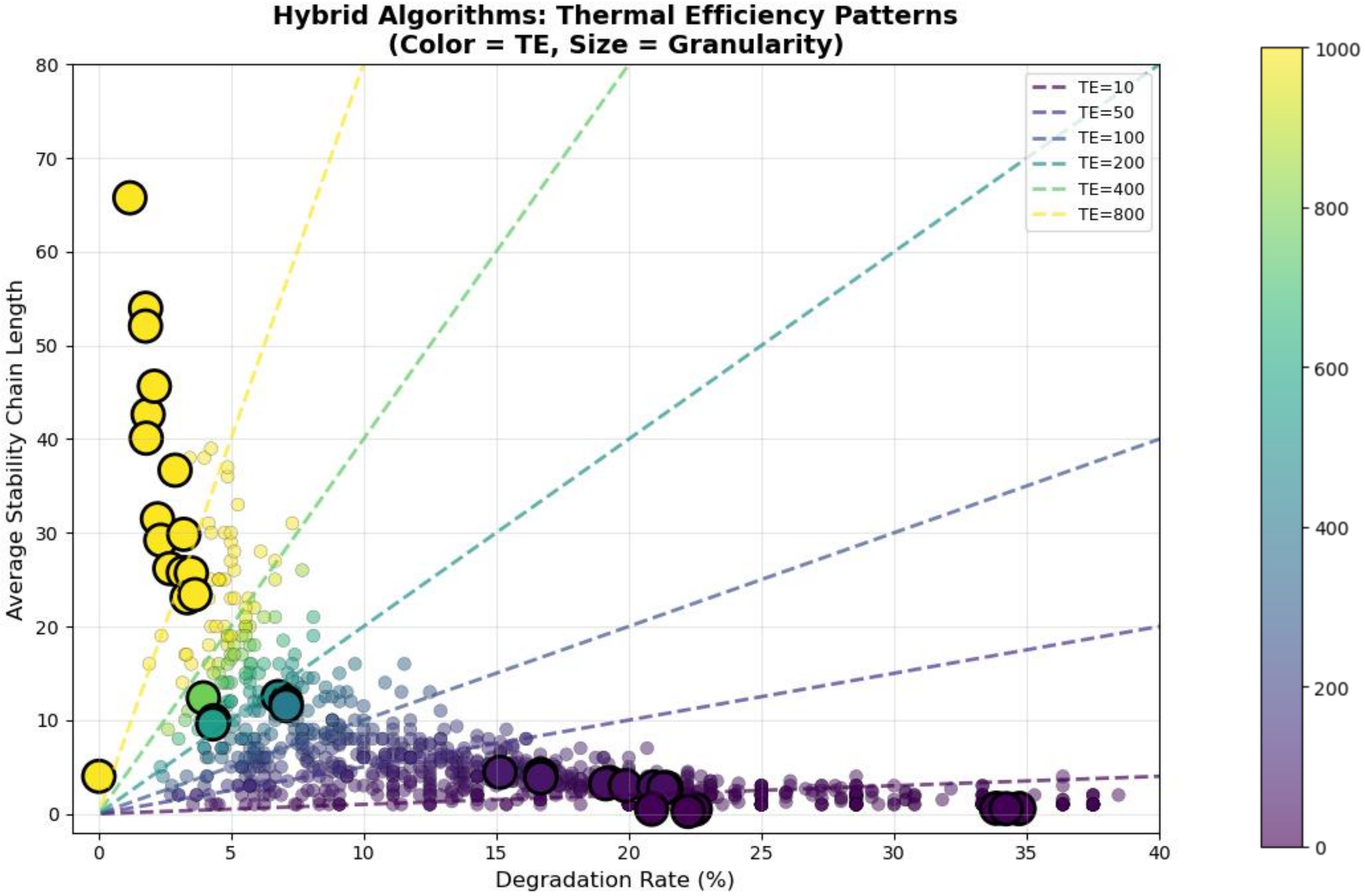
Who sustains containers longest?

- **Light**: FWF best, Random worst
- **Moderate**: RR worst, Random best

Thermal Efficiency



Thermal Efficiency
Mapping out the landscape between DR and SC



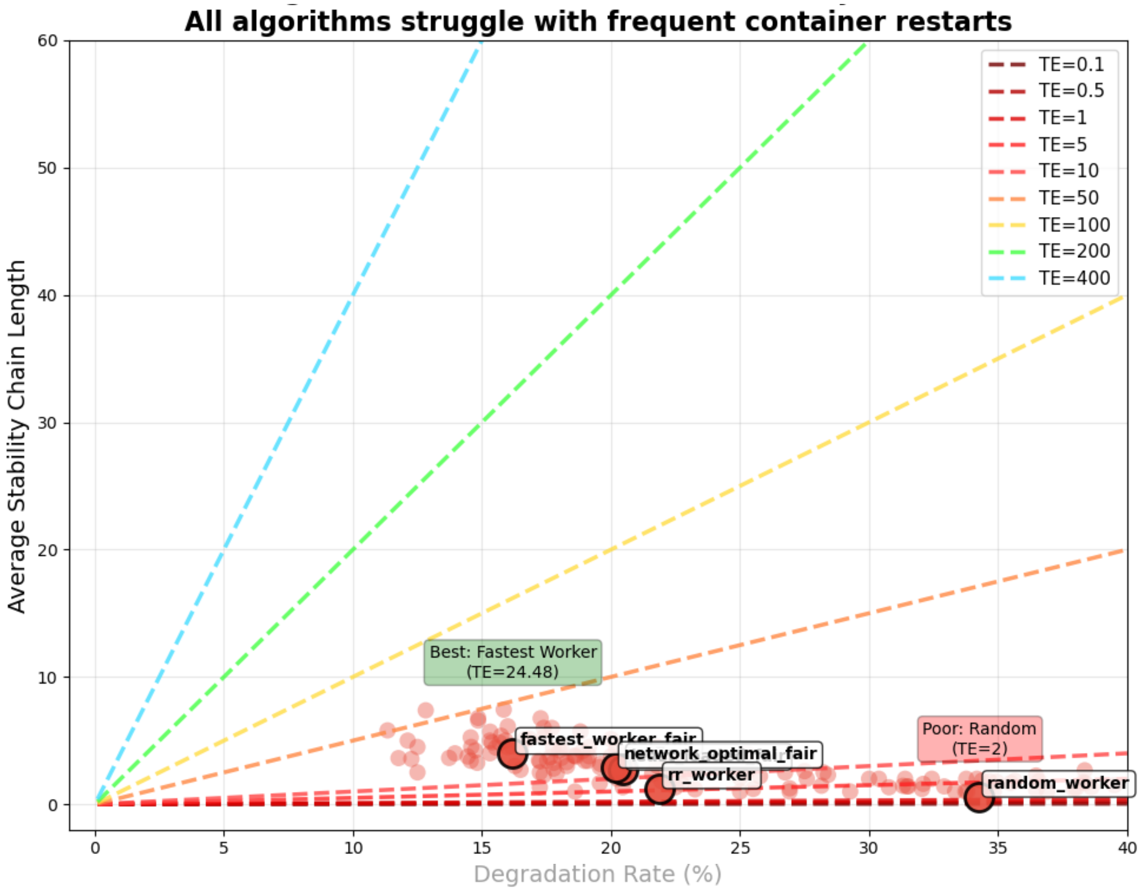
Thermal Degradation Rate

Thermal Stability Chain

Thermal Efficiency Ratio

Thermal Efficiency

LIGHT WORKLOADS



Light workloads:

- Best TE: 24.48 (FWF) - very low
- Clustering near x-axis
- High restart overhead

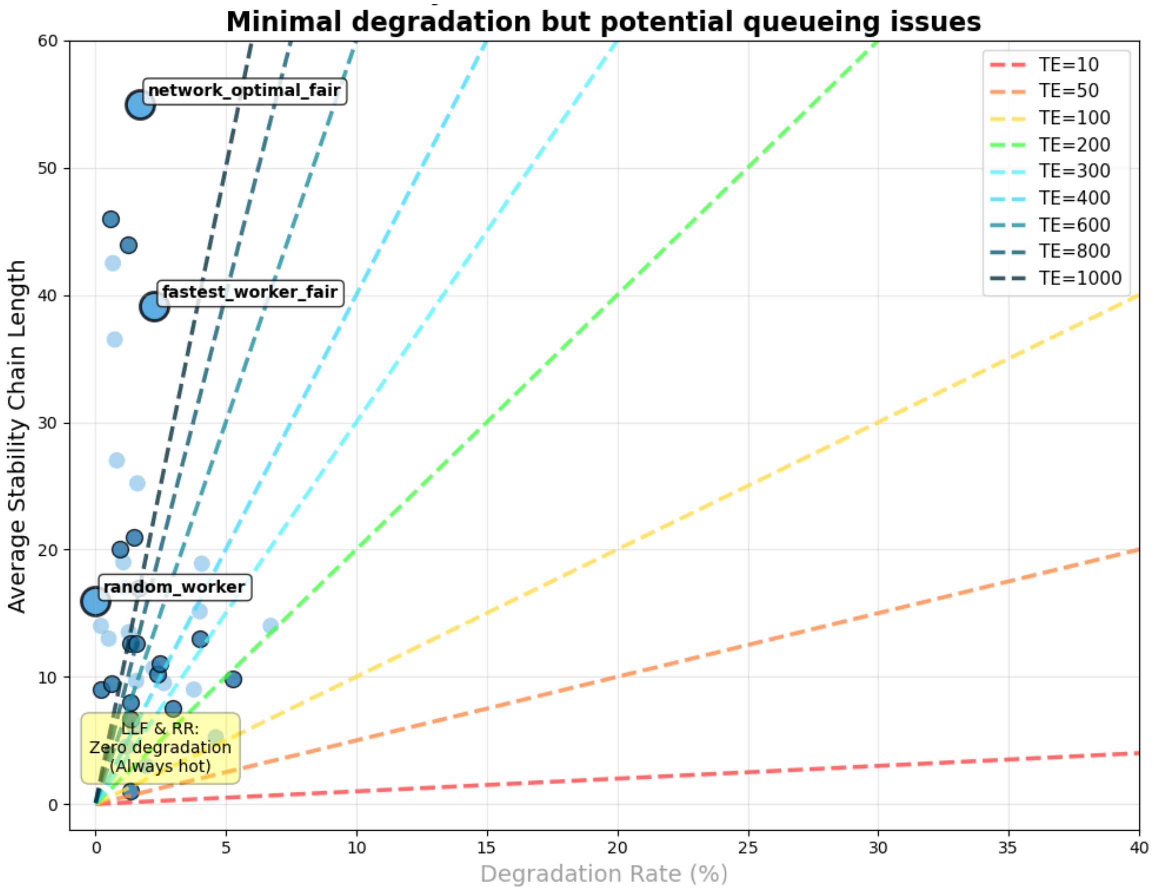
Signal: Room for algorithmic optimisation

Thermal Eff. (TE) Scale:

0-50:	Poor (high restart overhead)
50-100:	Fair (approaching the elbow)
100-200:	Good (elbow/tipping point area)
200-400:	Very good utilisation
400++:	Excellent but may cause bottlenecks

Thermal Efficiency

HEAVY WORKLOADS



Heavy workloads:

- Peak TE: 3,220 (NOF)
- But dark dots cluster at zero
- Near-vertical = bottlenecked

Signal: System saturated, not optimised

Thermal Eff. (TE) Scale:

0-50:	Poor (high restart overhead)
50-100:	Fair (approaching the elbow)
100-200:	Good (elbow/tipping point area)
200-400:	Very good utilisation
400++:	Excellent but may cause bottlenecks

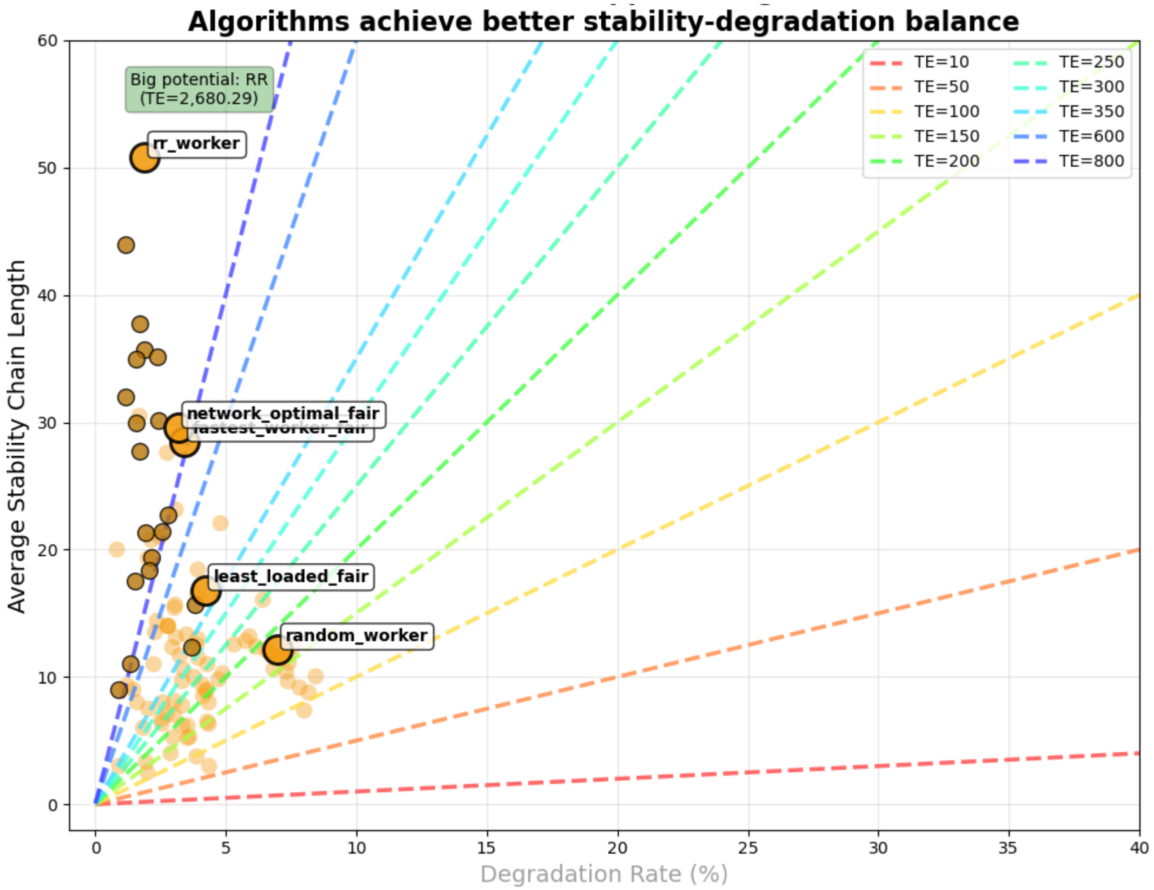
Thermal Degradation Rate

Thermal Stability Chain

Thermal Efficiency Ratio

Thermal Efficiency

MODERATE WORKLOADS



Moderate workloads:

- Peak: 2,680 TE (NOF)
- Iteration dots spreading evenly
- Moving toward y-axis

Signal: Load balancing becoming critical

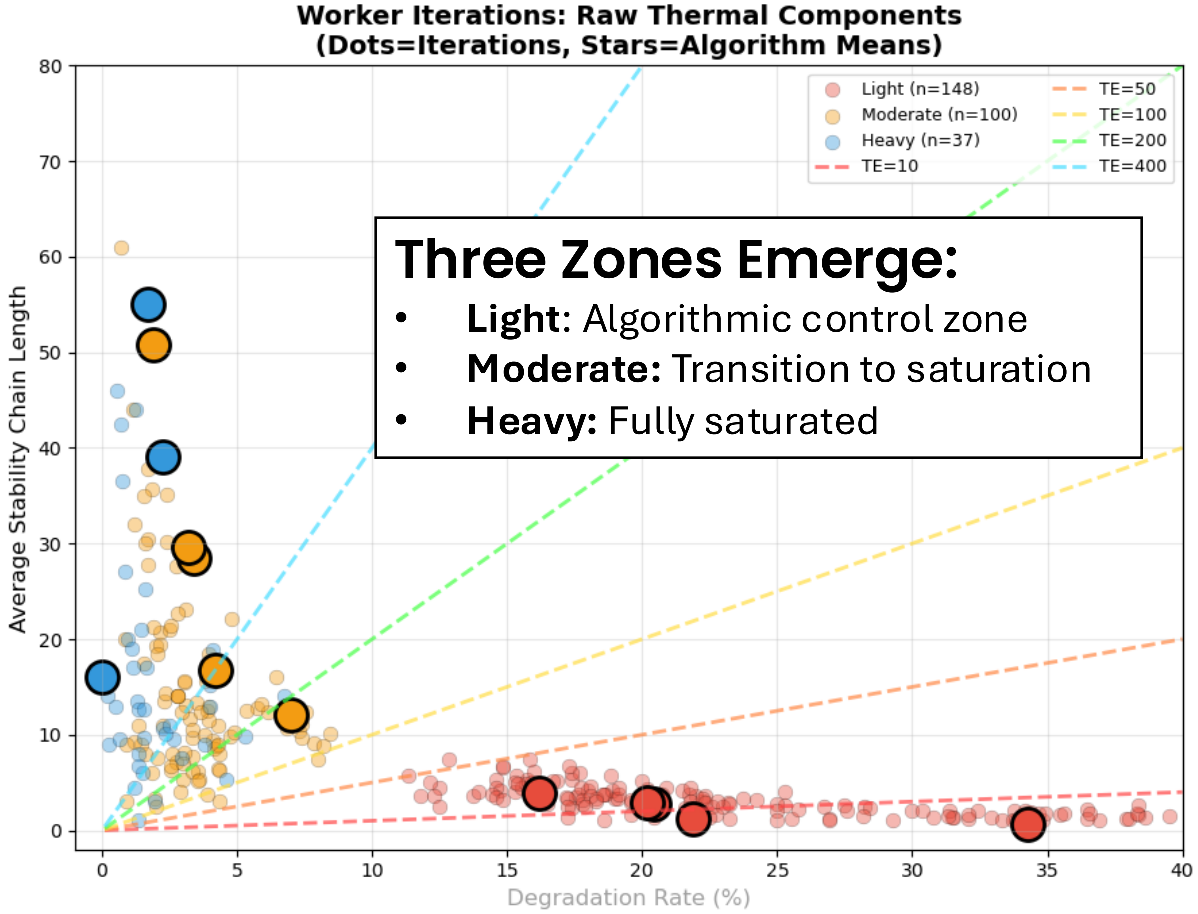
Thermal Eff. (TE) Scale:	
0-50:	Poor (high restart overhead)
50-100:	Fair (approaching the elbow)
100-200:	Good (elbow/tipping point area)
200-400:	Very good utilisation
400++:	Excellent but may cause bottlenecks

Thermal Degradation Rate

Thermal Stability Chain

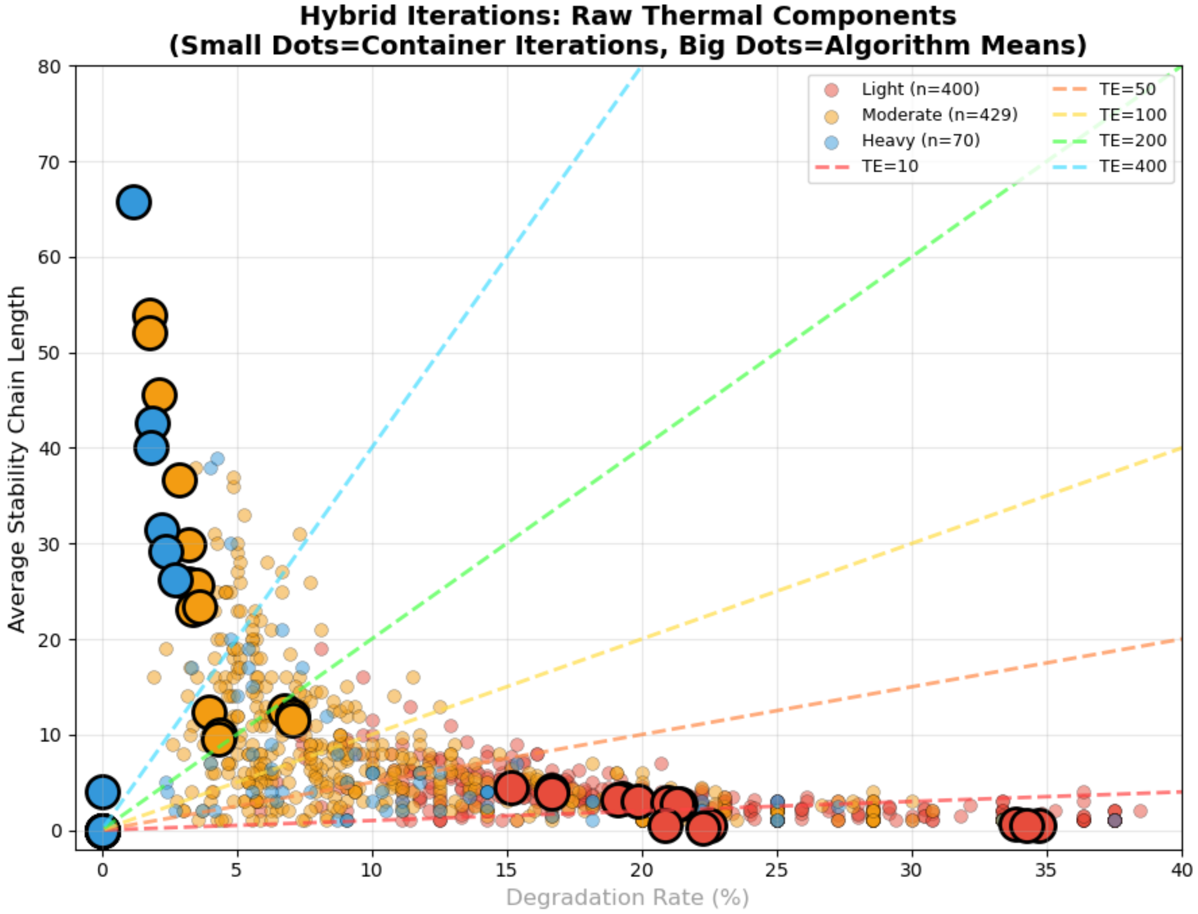
Thermal Efficiency Ratio

Thermal Efficiency Landscape



Three Zones Emerge:

- **Light**: Algorithmic control zone
- **Moderate:** Transition to saturation
- **Heavy:** Fully saturated

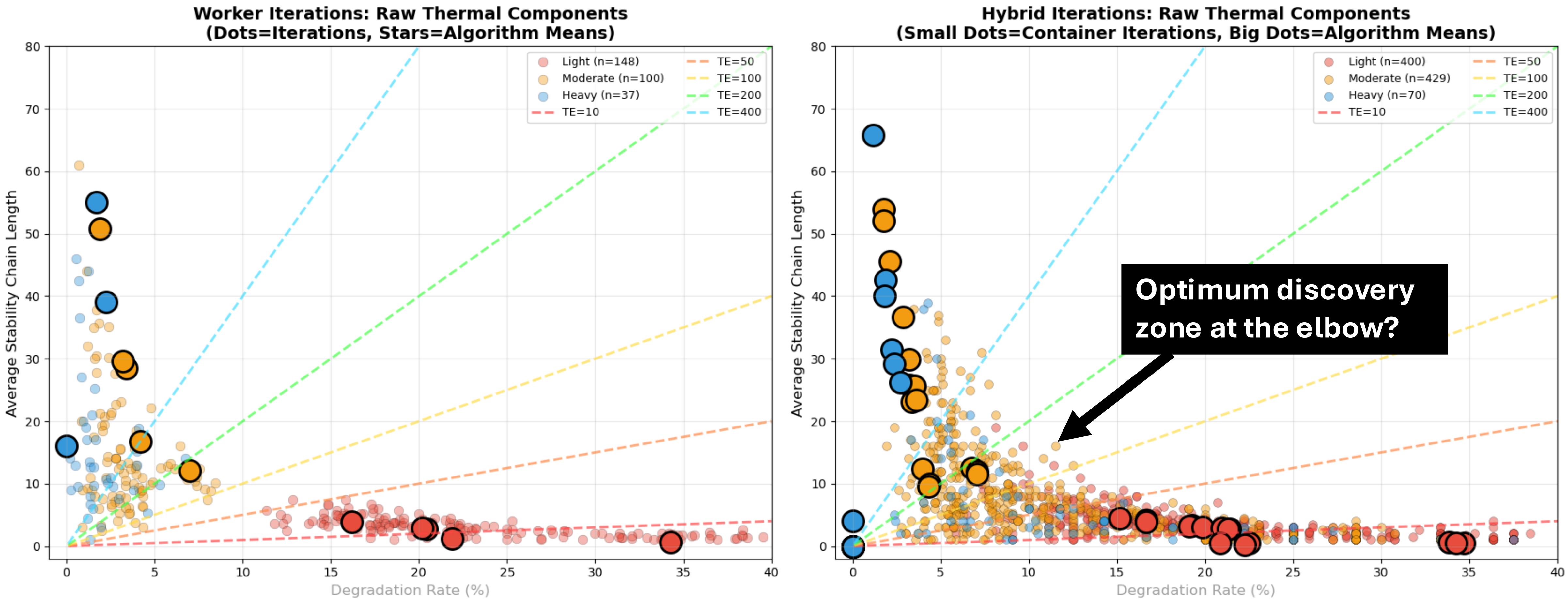


Thermal Degradation Rate

Thermal Stability Chain

Thermal Efficiency Ratio

Thermal Efficiency Landscape



Thermal Degradation Rate

Thermal Stability Chain

Thermal Efficiency Ratio

Results

Penalties must be the driver of variation

Worker algorithms drive penalties

Thermal management hypothesis formed

1. Distribution and Variance Component

What's causing variation?

2. Categorical Association

Where do algorithms differ in penalties?

3. Thermal Efficiency

How do they create these differences

4. Cross Validation

Testing our hypothesis

Cross Validation

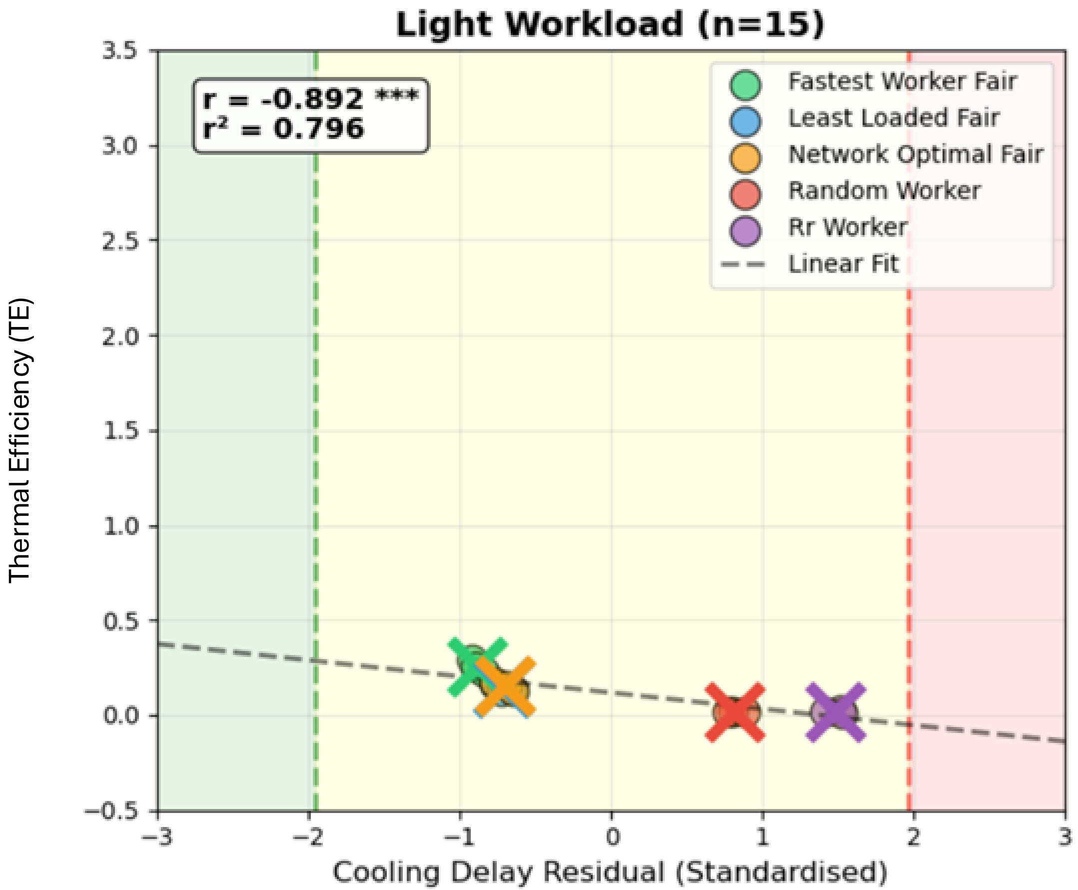
Pattern checking

Algorithm Type	Workload	r	p	Sig.
Worker	Light	-0.825	0.085	ns
	Moderate	-0.798	0.106	ns
	Heavy	-0.385	0.523	ns
Hybrid	Light	-0.892	0	***
	Moderate	-0.77	0.001	***
	Heavy	0.01	0.971	ns

Pattern: Strong → Significant → None

Cross Validation

LIGHT WORKLOADS: Maximum Elasticity



Light workloads: Maximum Algorithmic Control

- **Statistical**: Strong correlation (r = -0.892)
- **Practical**: 38.1% cooling penalties
- **Shallow slope:** y = -0.086x

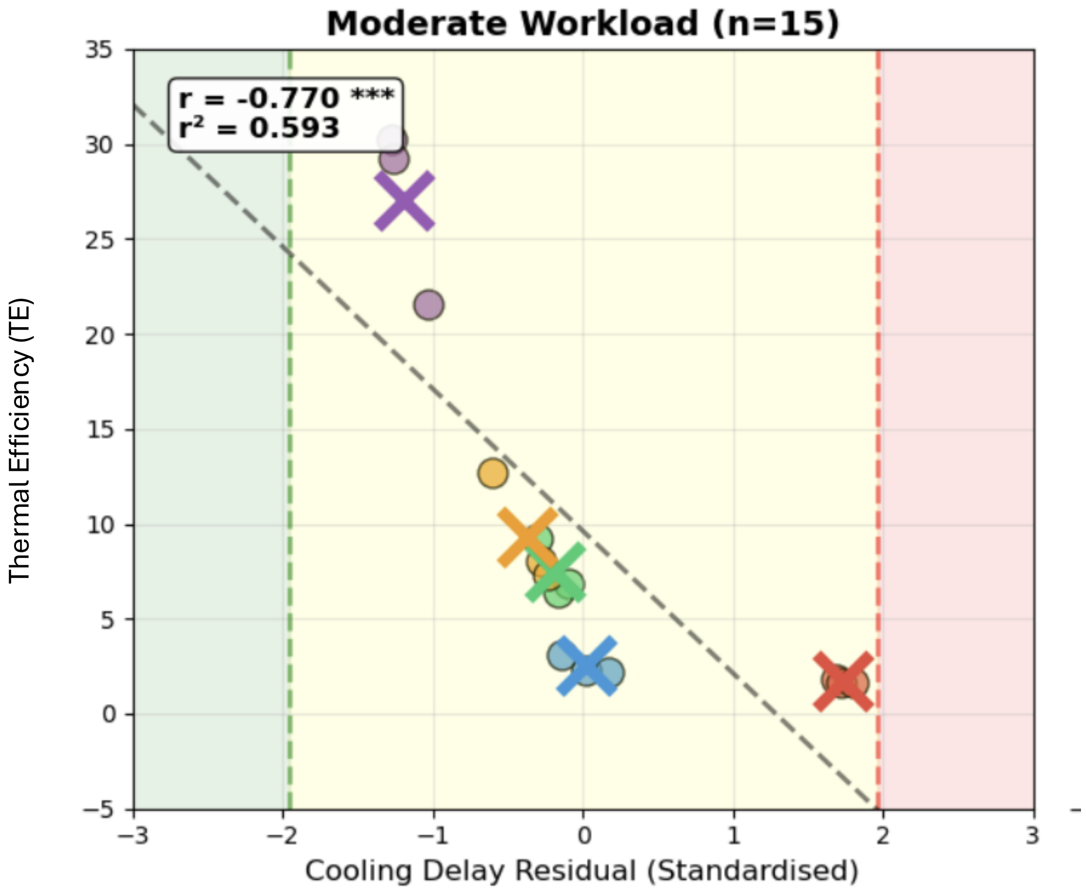
Small TE improvements → Large SR gains = Maximum algorithmic leverage

Trend lines (linear regression)

Light:	y = -0.0855x + 0.1178
Moderate:	y = -7.4906x + 9.6236
Heavy:	y = -40.0907x + 68.9546

Cross Validation

MODERATE WORKLOADS: Approaching Saturation



Moderate Workload Warning Signs

- **Still Significant** (r = -0.770)
- **Only 3.8%** cooling penalties (10× reduction)
- **Steep slope**: y = -7.49x (87× steeper!)

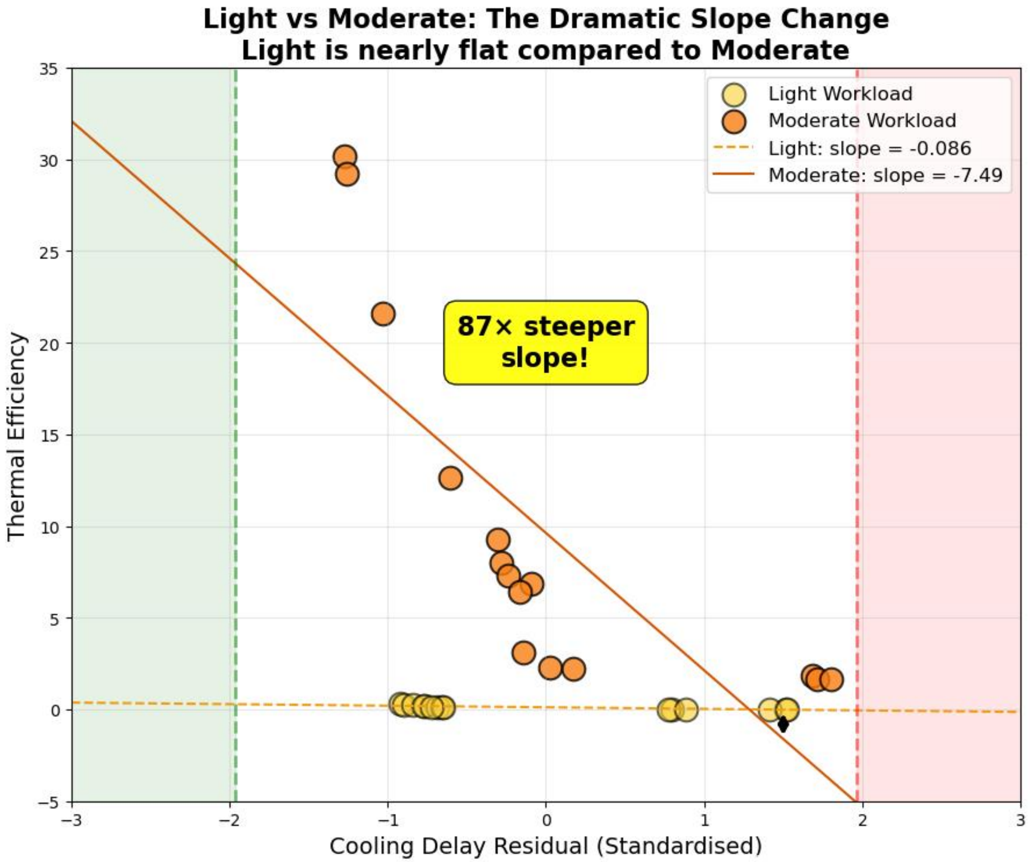
Large TE changes → Small SR gains
= Diminishing returns setting in

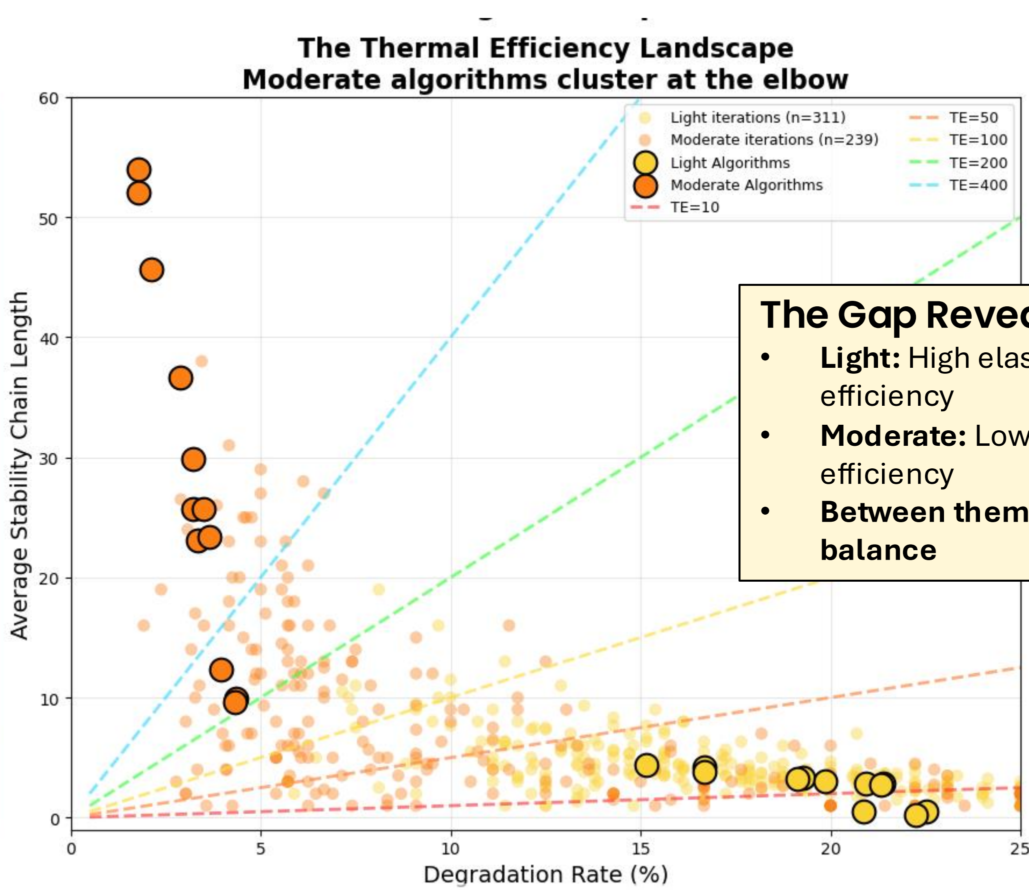
Trend lines (linear regression)	
Light:	y = -0.0855x + 0.1178
Moderate:	y = -7.4906x + 9.6236
Heavy:	y = -40.0907x + 68.9546

Slope is 87x steeper than light!

Cross Validation

The Optimal Zone Discovery





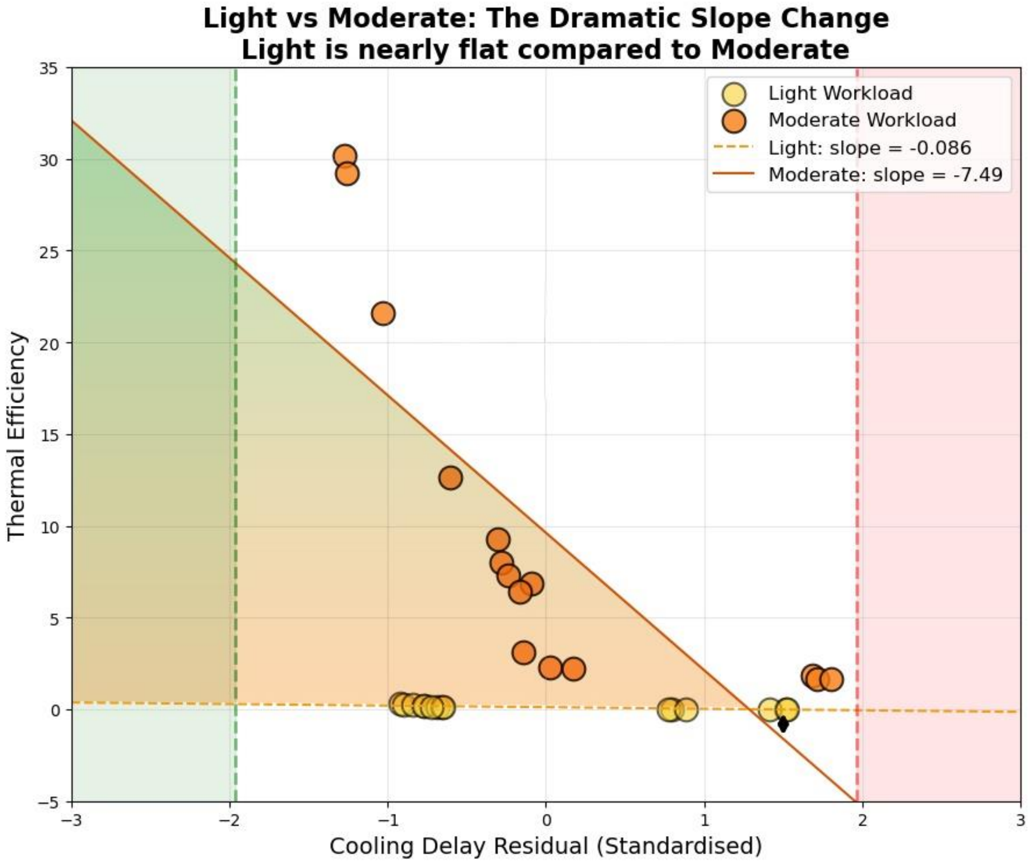
The Gap Reveals:

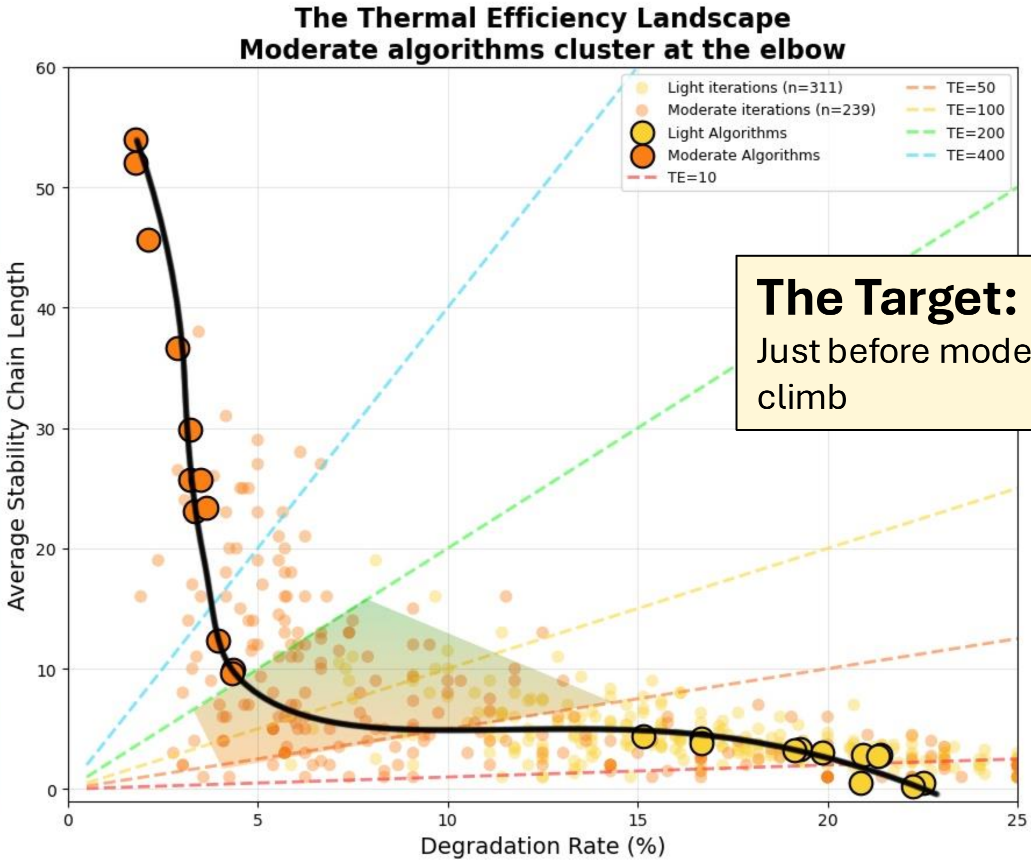
- **Light:** High elasticity, low efficiency
- **Moderate:** Low elasticity, higher efficiency
- **Between them: The optimal balance**

40 Correlation Discovery | Sensitivity Discovery | Scheduling Breakdown

Cross Validation

The Optimal Zone Discovery





The Target:
Just before moderate's steep climb

Correlation Discovery

Sensitivity Discovery

Scheduling Breakdown

Results

Penalties must be the driver of variation	Worker algorithms drive penalties	Thermal management hypothesis	TE and Std. Res. Correlation
1. Distribution and Variance Component	**2. Categorical Association**	**3. Thermal Efficiency**	**4. Cross Validation**
What's causing variation?	Which algorithms differ in penalties?	How do they create these differences	Testing our hypothesis

What did we discover?

Discovery # 1

Penalties Drive Performance

- **Processing time: negligible** (0.083ms)
- **Penalties:** drive all variability (227% → 23% CV)
- **Container lifecycle > Algorithmic sophistication**

What did we discover?

Discovery # 2

Worker Selection Hierarchy

- **Worker** algorithms: 15× **stronger** associations
- **Job** algorithms: **zero** statistical effect
- **BUT** only **when resources allow** differentiation
- **Heavy** workloads in our experiment are oversaturated

What did we discover?

Discovery # 3

Optimal Sensitivity Zone

- **Light**: Maximum algorithmic control
- **Moderate**: Approaching saturation limits
- **Heavy**: Fully saturated (algorithms irrelevant)

Optimal zone: Between Light and Moderate

What did we discover?

Discovery # 1

Penalties Drive Performance

Discovery # 2

Worker Selection Hierarchy

Discovery # 3

Optimal Sensitivity Zone

Answering the Main Question: Transparent Framework Key Insights

1. **Optimal** zone exists **BEFORE full** container **utilisation**

2. **Systematic evaluation** of black-box schedulers is **possible**

Worker selection >> Job prioritisation

Thermal patterns predict performance at scale

47

Future Research Questions

Meta-Scheduling Opportunities

- Can we dynamically target the elbow zone (TE 20-50)?
- How do we adapt as workload intensity shifts?
- Is there an optimal path along this curve?
- Can predictive models guide this transition?

Thank You!

Meta TaskWave

Theresa To
MSc Computing and Information Systems

49