

The Computational Sprinting Game

Benjamin C. Lee

benjamin.c.lee@duke.edu

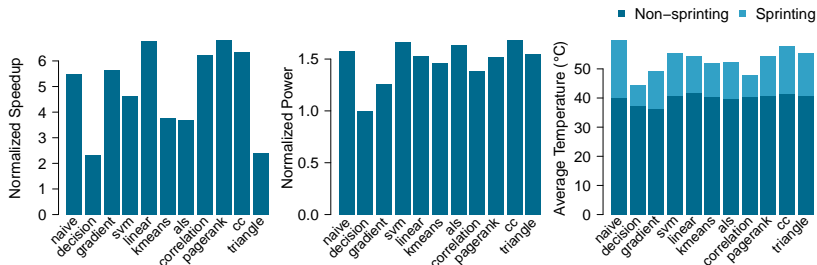
Songchun Fan, Andrew Hilton, Ziqiang Huang,
Jose Joao, Alejandro Rico, Seyed Majid Zahedi



Computational Sprints

Enhance performance briefly with extra power

3 cores @1.2GHz → 12 cores @ 2.7GHz

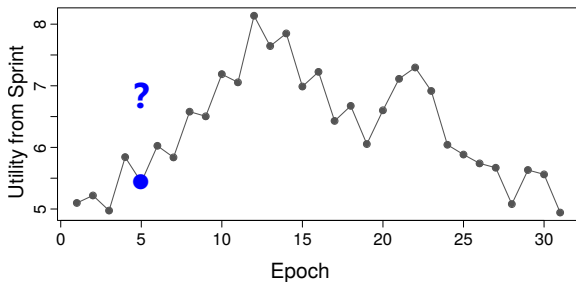


Sprint Management

When should processors sprint?

Which processors should sprint?

Phases and processors that benefit most

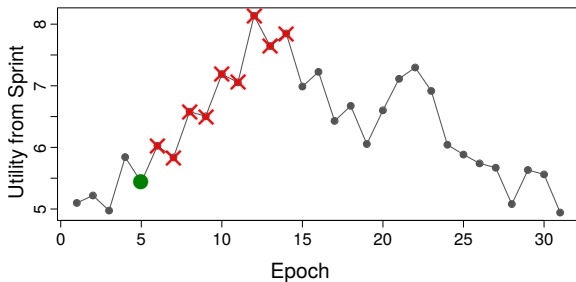


Sprint Management

Sprint consumes thermal headroom

Sprint constrains future action

Sprint requires recovery time



DynaSprint

Software Run-Time

Predict workload's sprint utility, cost

Track system's thermal headroom

Decide whether to sprint

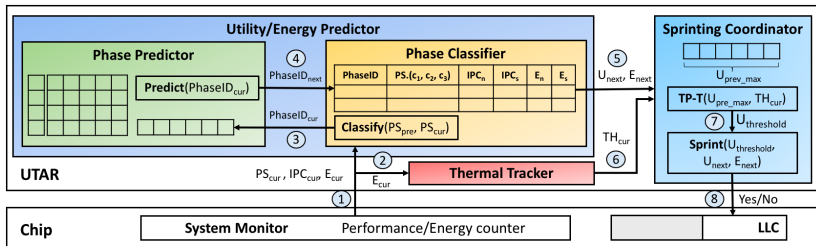
Hardware Support

Allocate resources (*e.g.*, cache capacity)

Monitor performance, energy

Implement safety fallback

DynaSprint

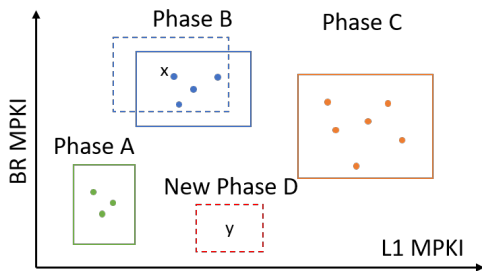


(1-2) **Classify** $events_i \rightarrow phase_i$

(3-5) **Predict** $phase_i \rightarrow phase_{i+1} \rightarrow utility_{i+1}$

(6-8) **Control** sprint if $utility_{i+1} > utility_{thresh}$

Phase Classifier

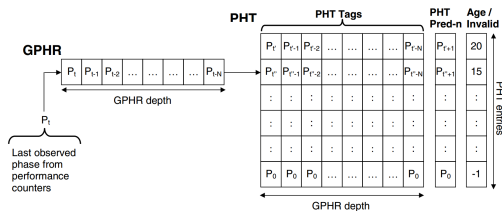


$\text{events}_i \rightarrow \text{phase}_i$

Define signature
with events

Cluster signatures to
define phase

Phase Predictor

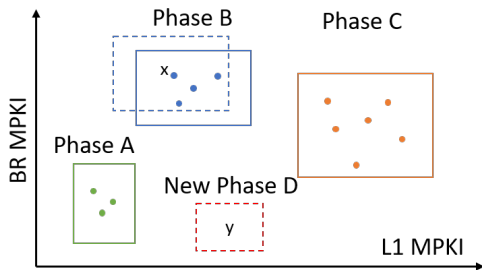


$\text{phase}_i \rightarrow \text{phase}_{i+1}$

Track phase history

Use history to predict phase

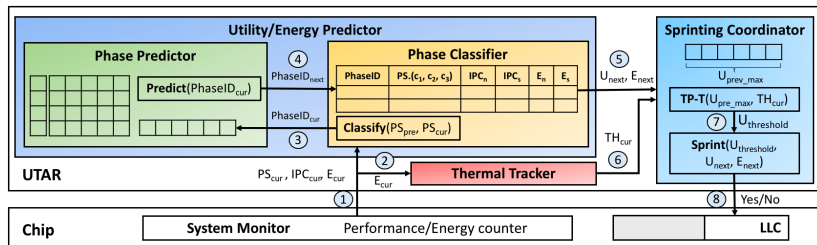
Utility Predictor



phase_{*i*+1} → **utility**_{*i*+1}

Use phase average
to predict utility

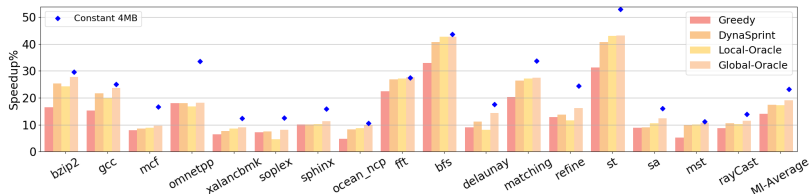
Sprint Decision



$$U_{i+1} > U_{max} \times H_{consumed} / H_{total}$$

U_{i+1} Predicted utility
 U_{max} Max utility in recent epochs
 $H_{consumed} / H_{total}$ Fraction of thermal headroom consumed

Performance

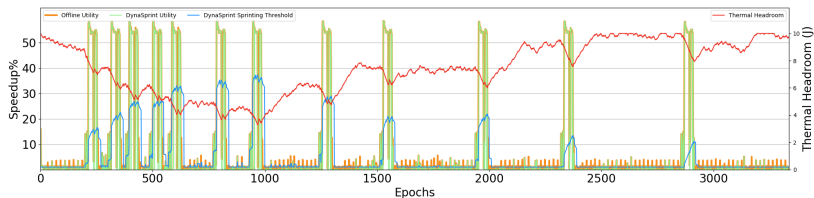


Improve performance by 17% (average), 40% (max)

Outperform heuristics

Compete with oracles

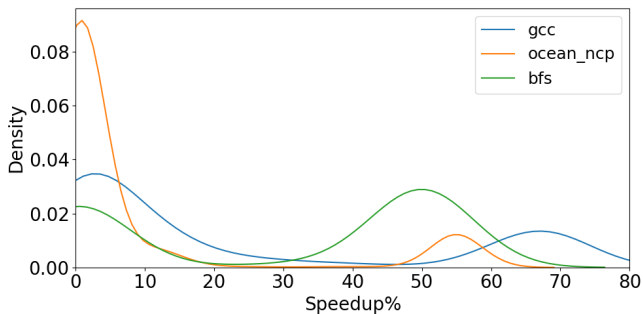
Dynamics



Sprint during high-utility epochs

Tune management policy dynamically

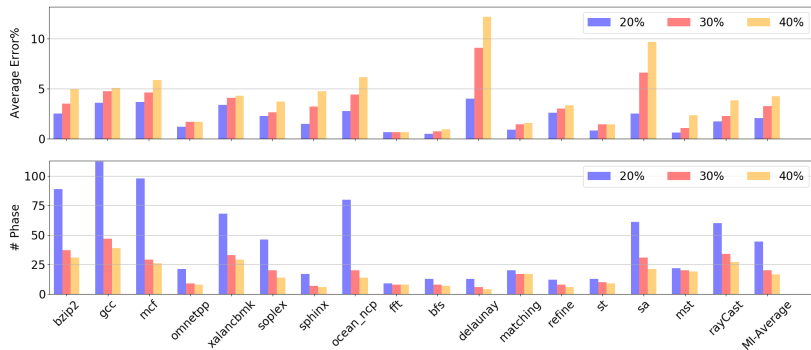
Utility Distributions



Perform well for dynamic workloads

Sprint judiciously given bimodal speedups

Prediction Accuracy



Predict utility from phase with low error

Track modest number of unique phases

Conclusion



Dynamic Phase Analysis

Manage sprints online

Sprint with cache capacity

Algorithmic Game Theory

Arbitrate multi-programmed sprints

Decentralize power management

Future Directions

Coordinate multi-resource sprints

Define sprints dynamically

Huang, Joao, Rico, Hilton, Lee. **“DynaSprint: Microarchitectural sprints with dynamic utility and thermal management.”** MICRO 2019.

Fan, Zahedi, Lee. **“Distributed strategies for computational sprints.”** CACM 2019.

Fan, Zahedi, Lee. **“The computational sprinting game.”** ASPLOS 2016.

The Computational Sprinting Game

Benjamin C. Lee

benjamin.c.lee@duke.edu

Songchun Fan, Andrew Hilton, Ziqiang Huang,
Jose Joao, Alejandro Rico, Seyed Majid Zahedi

