Performance Predictability in Heterogeneous Memory

Jinshu Liu, Hanchen Xu, Daniel S. Berger*, Marcos K. Aguilera†, Huaicheng Li
jinshu@vt.edu

Heterogenous Memory

DRAM	NUMA/CXL
Fast-tier	*Slow-tier*
Limited size Fast, low latency	Memory expansion Slow, High latency

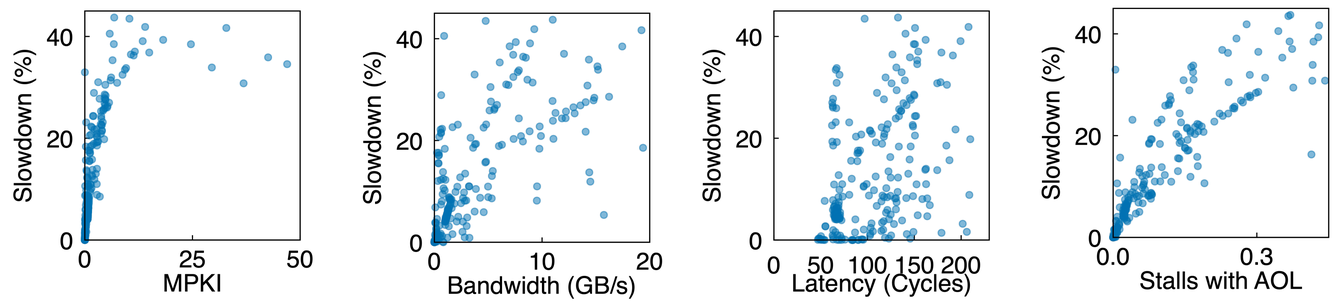
Without accurate predictors

Cloud operators overprovision DRAM to minimize risk
OS/runtimes correct placement errors after performance has degraded

Predictors are required

Accurate slowdown predictors can tell how a placement decision will affect application performance

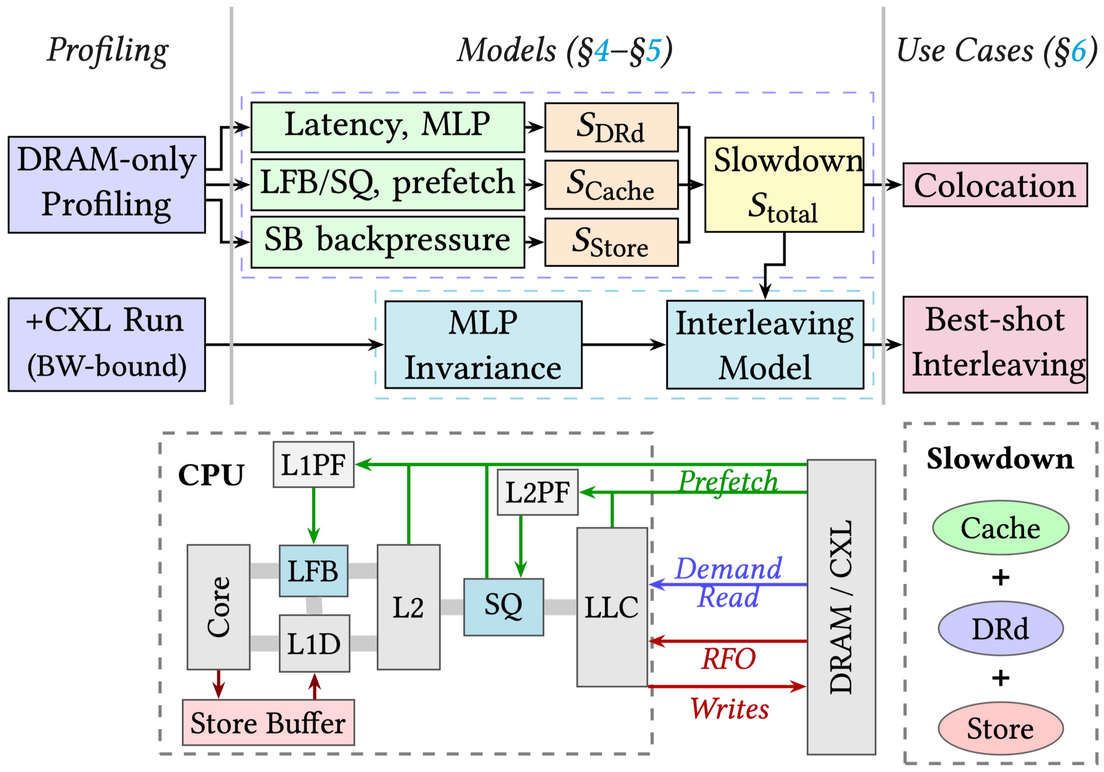
Existing Metrics Fall Short



Simple proxy metrics fail to accurately capture microarchitectural behavior

Is it possible to predict the workload performance on CXL using intrinsic workload signatures?

CAMP: Causal Analytical Memory Prediction



Slowdown Prediction

The microarchitectural causes:

The dynamics of MLP and latency

The reliance on LFB and prefetching

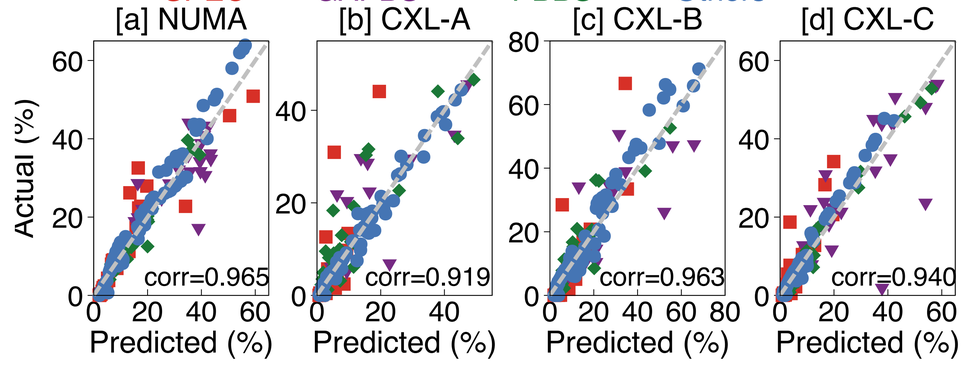
The pressure on Store Buffer

The predictors:

$$S_{\text{DRd}} \approx k \times \frac{1}{p + q \cdot \frac{\text{MLP}_{\text{DRAM}}}{L_{\text{DRAM}}}} \times \frac{s_{\text{LLC}}}{c}$$

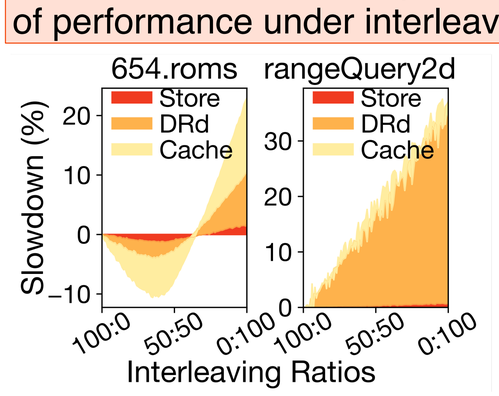
$$S_{\text{Cache}} \approx k \times R_{\text{LFB-hit}} \times R_{\text{Mem}} \times \frac{s_{\text{Cache}}}{c}$$

$$S_{\text{Store}} \approx k \times \frac{s_{\text{SB}}}{c}$$

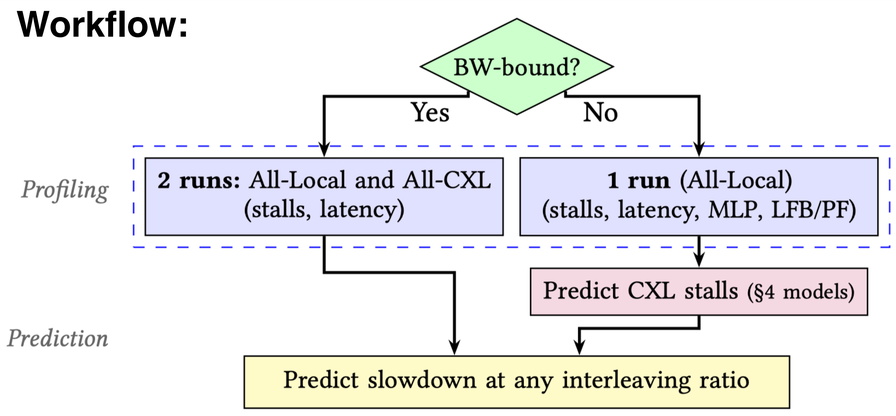


Interleaving model

How is the continuous spectrum of performance under interleaving?

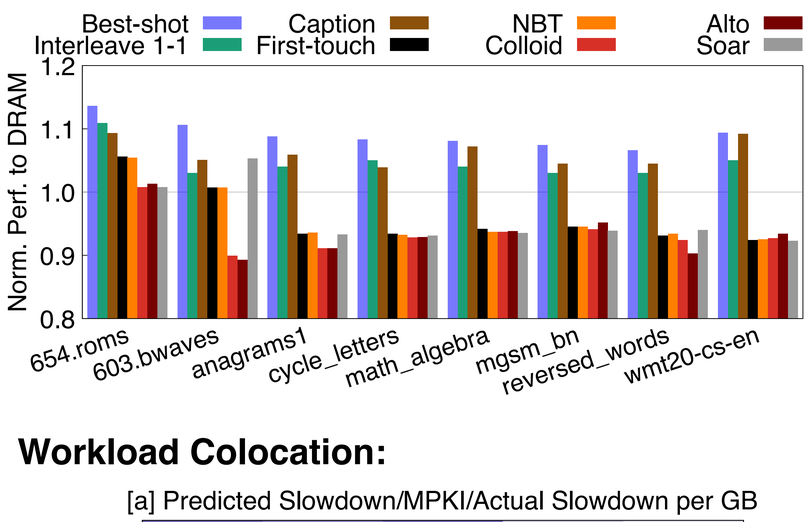


Workflow:



Use Cases

"Best-shot" Interleaving:



Goal:
Predicting performance at any interleaving ratio (x)

Challenge:
Changing the ratio alters the traffic load on each tier, which non-linearly impacts contention and latency

$$C = \frac{N \times L}{MLP}$$

The variance of C is determined by how N, L and MLP change with x

-> MLP invariance

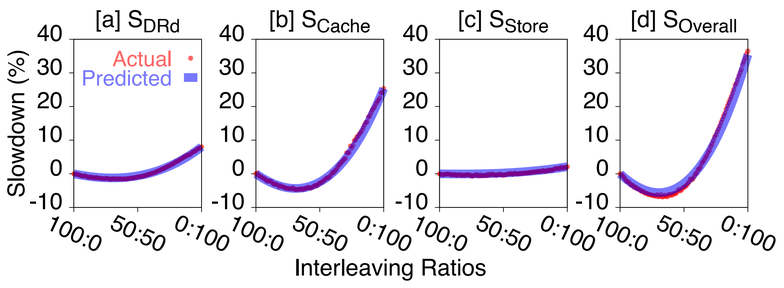
-> Modeling latency curve

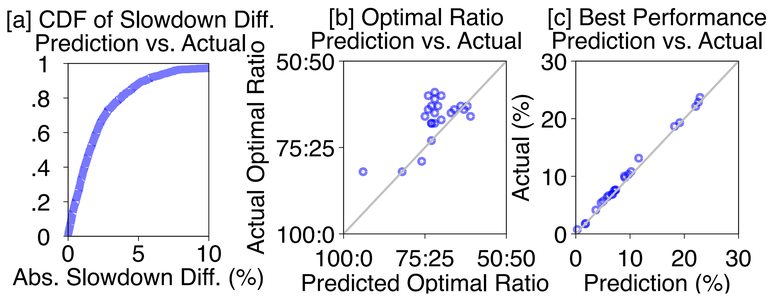
The predictor:

$$S(x) \approx \frac{\mathcal{M}(x) \cdot s_{\text{DRAM}} + \mathcal{M}(1-x) \cdot s_{\text{CXL}} - s_{\text{DRAM}}}{c}$$

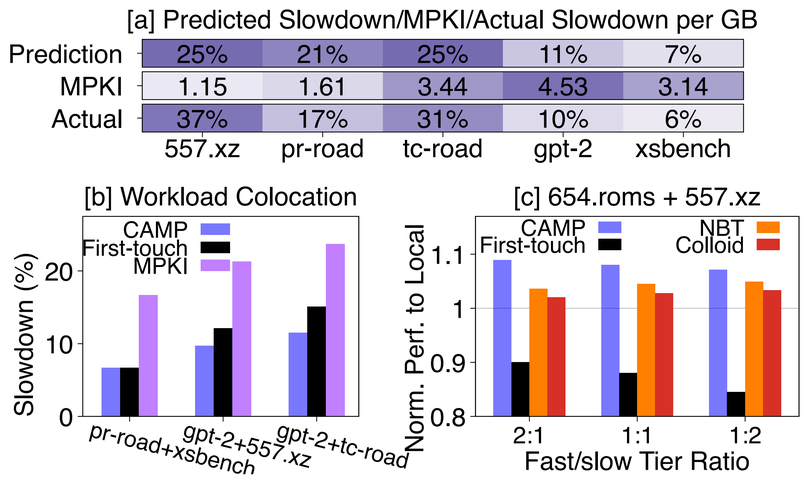
$$\mathcal{M}(x') = \frac{x' \cdot [L_{\text{idle}} + (L_{\text{full}} - L_{\text{idle}}) \cdot x'^2]}{L_{\text{full}}}$$

Prediction results:





Workload Colocation:



More in the paper:

Appears in the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'26)