VisSched: An Auction based Scheduler for Vision Workloads on Heterogeneous Processors

Diksha Moolchandani*, Anshul Kumar*,
José F. Martínez¥, and Smruti R. Sarangi*
*Indian Institute of Technology Delhi, India
¥Cornell University, Ithaca, NY, USA

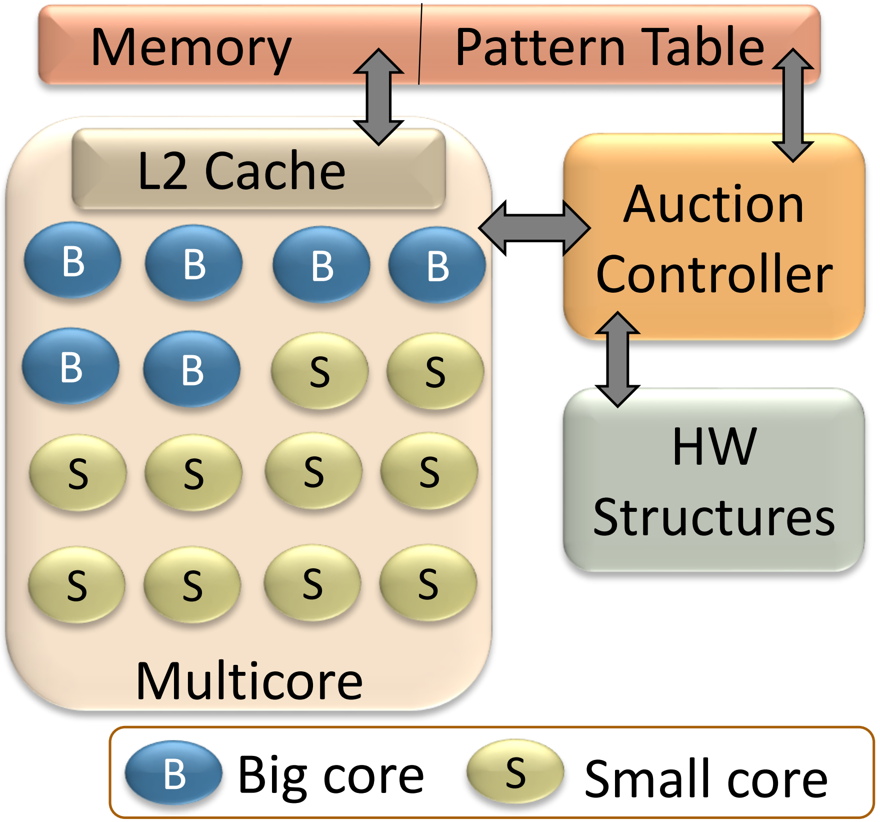
Biography

- I (Diksha Moolchandani) am a Ph.D. Research Scholar in the School of Information Technology, IIT Delhi since 2016
    - Bachelors in Electronics and Communication Engineering, IIITDM Jabalpur, 2015
- My research areas are designing novel architectures and architectural optimizations for image processing and computer vision applications
- Email: diksha.moolchandani@cse.iitd.ac.in
- Website: http://www.cse.iitd.ac.in/~diksha/

Problem Motivation

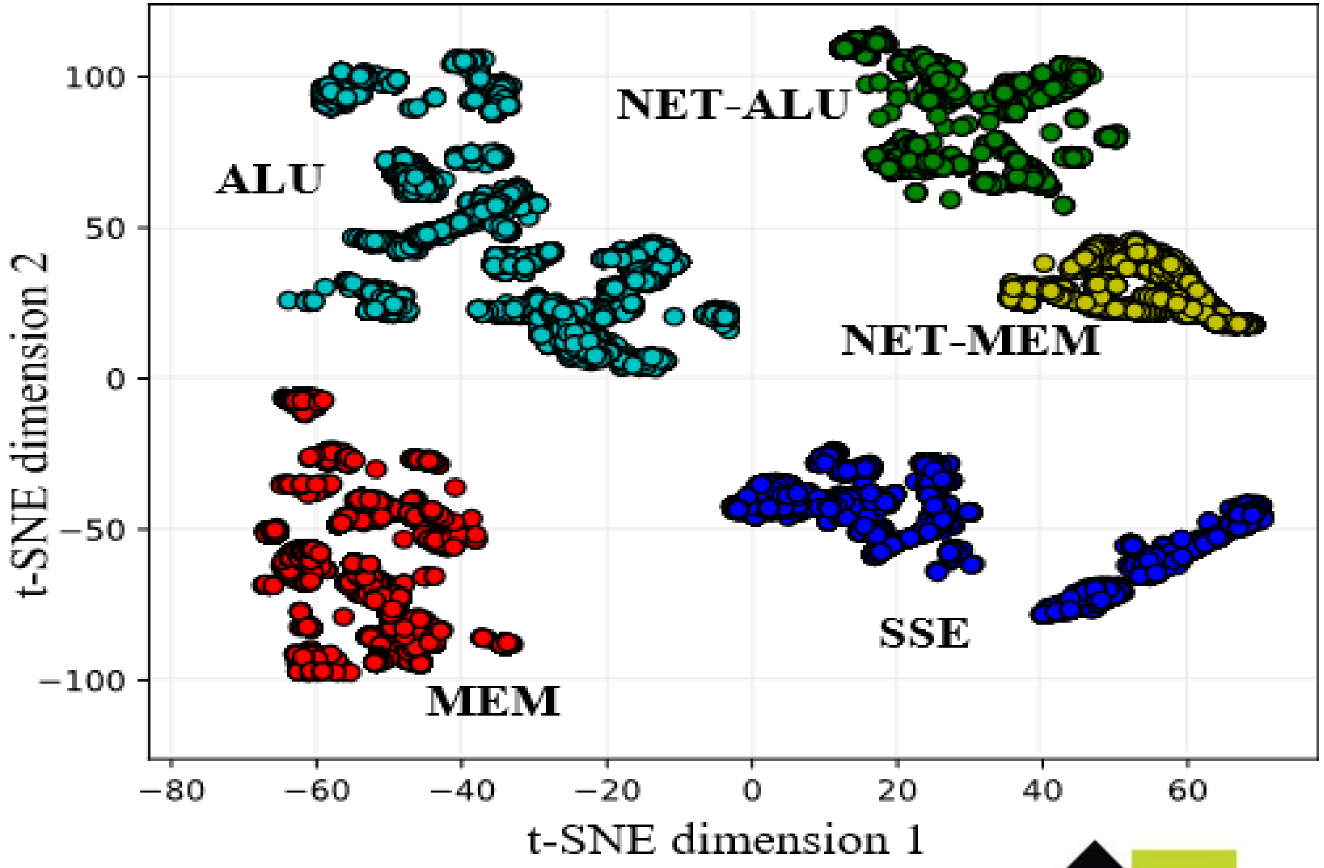
- Application specific benchmarks share common kernels. Eg: MEVBench, SD-VBS, CortexSuite, CAVBench
  - Traditional scheduling algorithms do not exploit this property
- Multiple applications run concurrently on edge, cloud, server
- Conventional way is to design accelerators
  - FPGAs & GPUs do not support multi-application concurrency
- Design specialized scheduling algorithms for scheduling these workloads on multicores that exploit the correlation among them

System Diagram

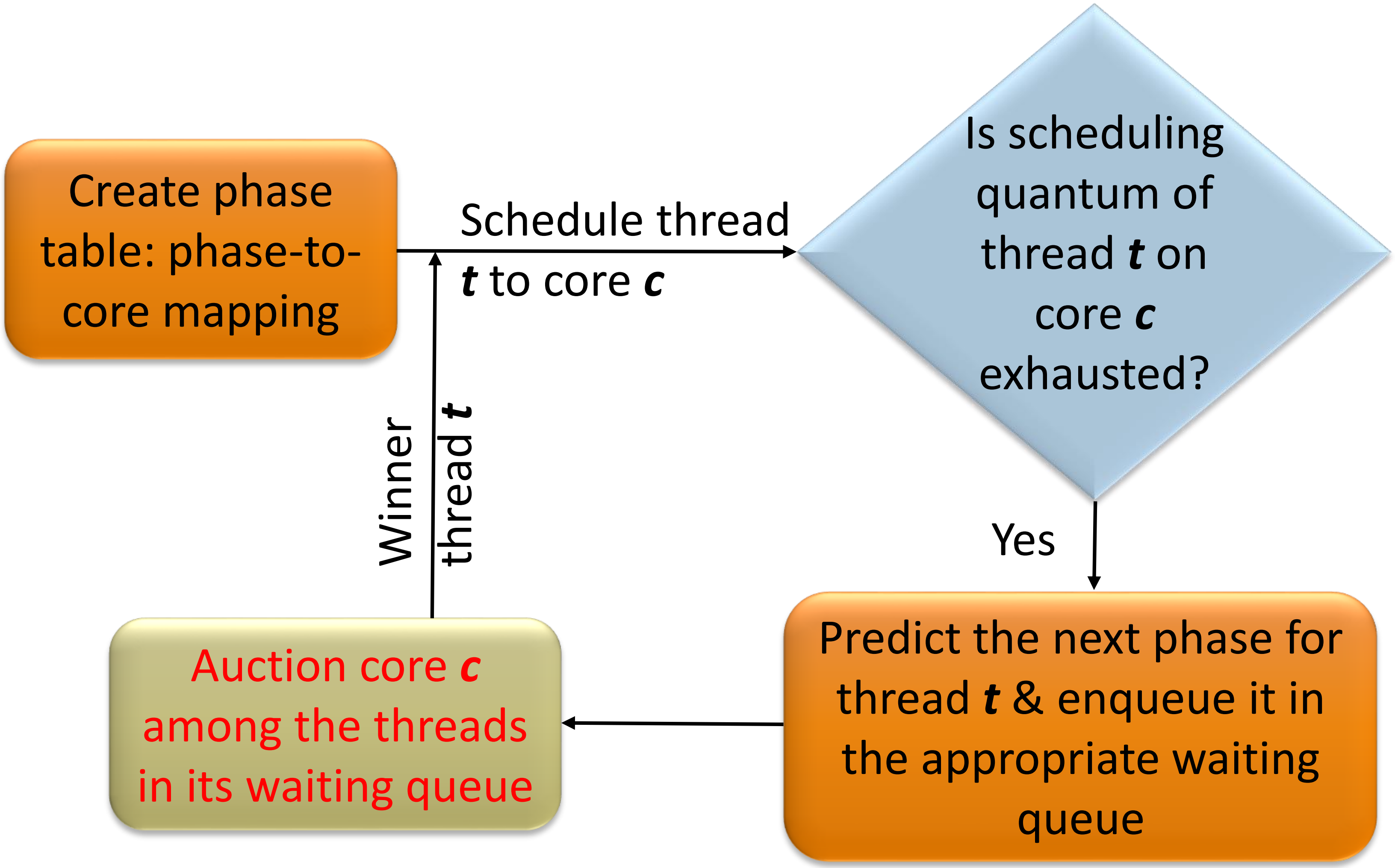


Characterization of Workloads

- Use the instruction mix in an execution interval as a feature vector
- Perform clustering
- Observations:
    - Any interval belongs to one of the five clusters
    - Strong positive correlation between the same phases of different workloads
    - Strong correlation between the phase and the IPC of an interval



VisSched: Flow Chart



Experimental Setup

Parameter	Value
Cores	10 small + 6 big
Frequency	1.55 GHz & 3.1 GHz
L1 I/D Cache	32 KB, 64 bytes block size, associativity 4
Shared L2	256 KB per bank, 16 banks, 64 bytes block size, associativity 8

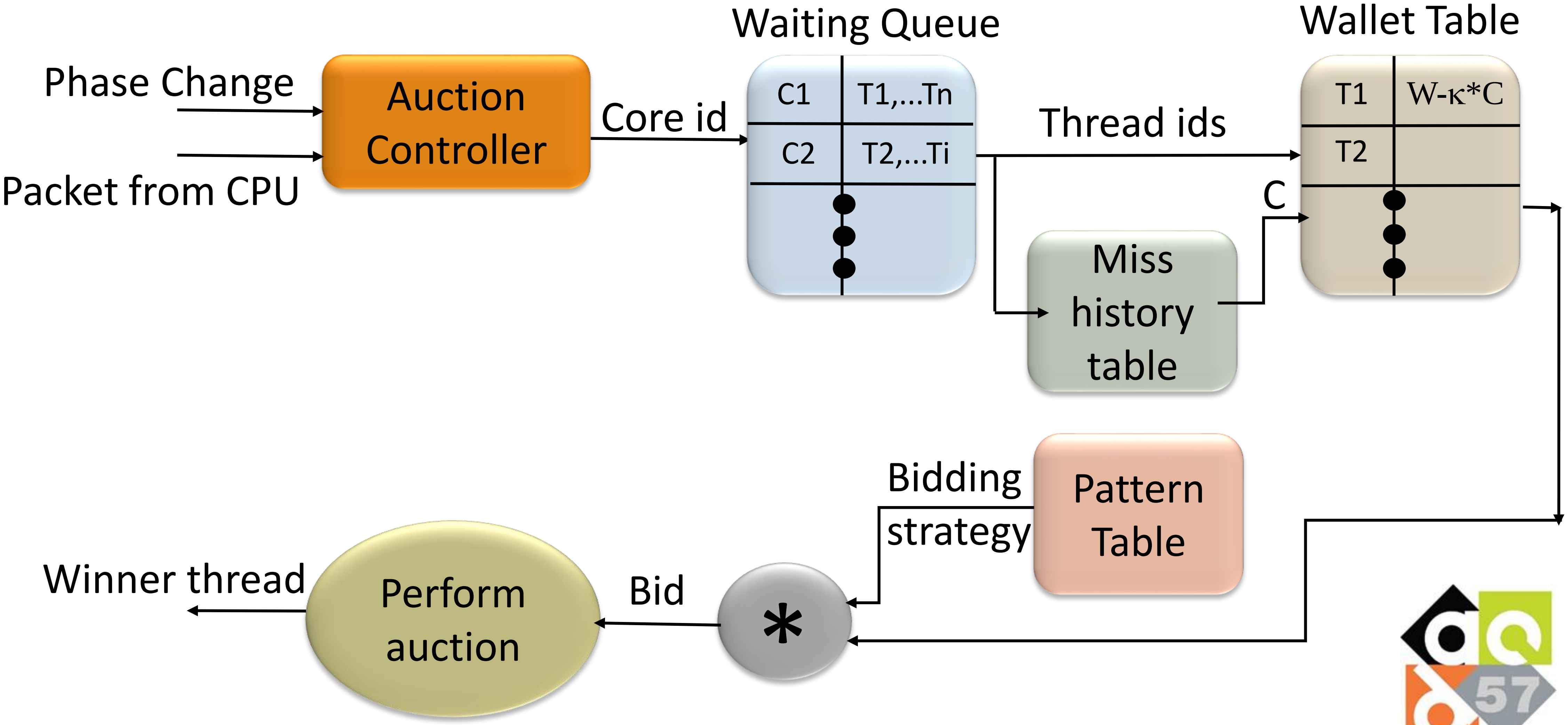
VisSched: Auction Process

Terms	Symbols	VisSched
Wallet	W	W
Auctioneer's fee	F	$\kappa * C$
Bid	$\mu*(W - F)$	$\mu*(W - \kappa * C)$
Bonus	B	$IPC/Energy + W_{base}$
Loser's subsidy	L	$L = \lambda * W_{base}$

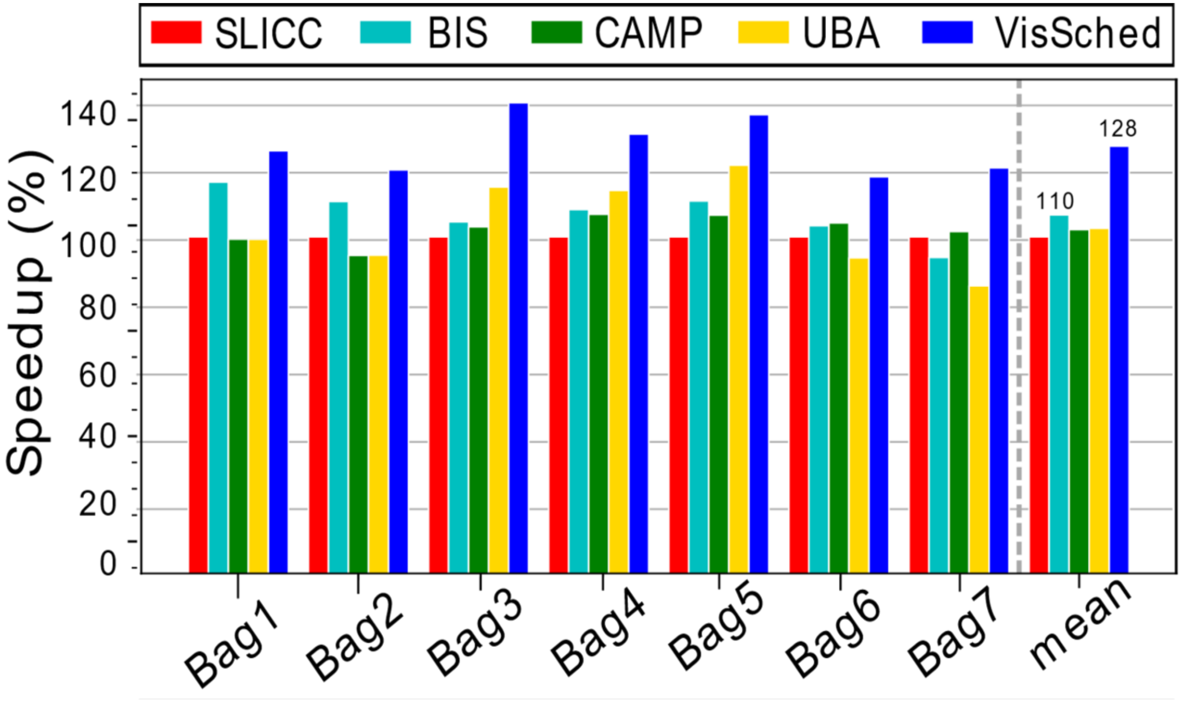
Updating the metrics	Winner	Loser
Utility	$(1 - \mu)(W_{old} - F) + B$	0
Wallet (W_{new})	$(1 - \mu)(W_{old} - F) + B$	$W_{old} + L$

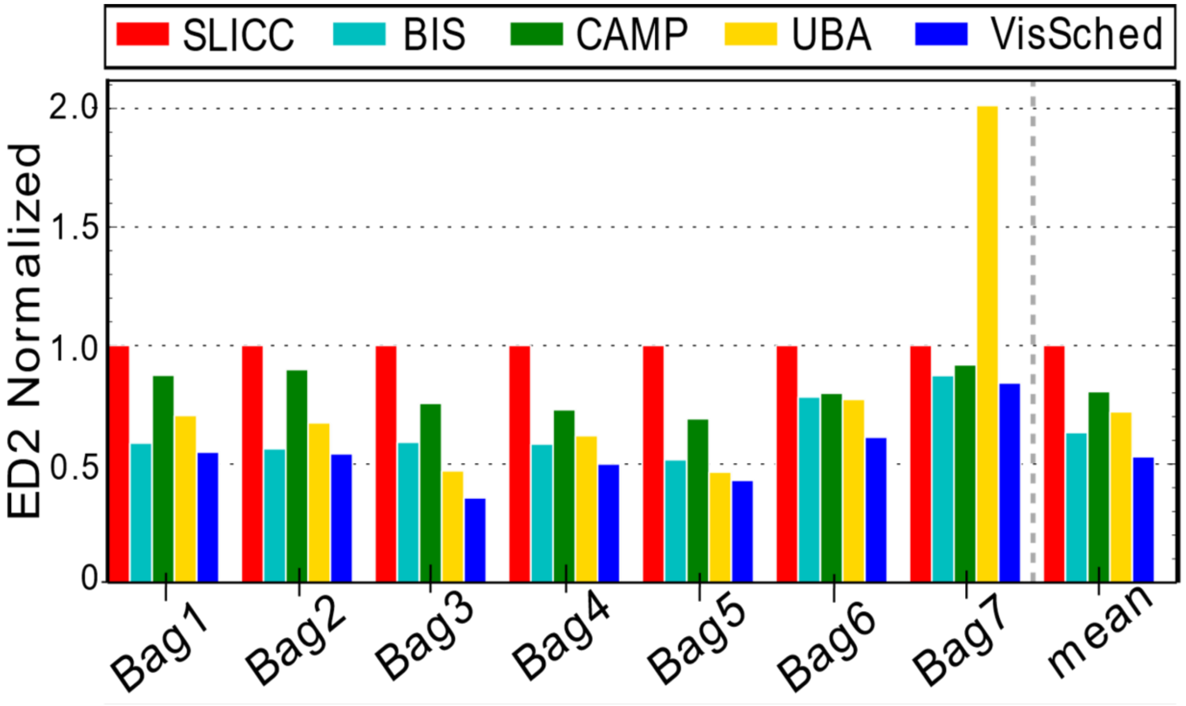
Symbol	Meaning
W_{base}	Base Wallet balance
C	Migration cost calculated as a function of cache misses
IPC	Instructions per cycle
Energy	Energy consumed by the system
κ	Normalization constant
μ	Bidding Strategy
λ	Constant
W_{new}	Wallet balance in next auction round
W_{old}	Wallet balance in prev. auction round

VisSched: Auction Process



Results





- **Performance improvement 18%**
- **ED^2 improvement 17%**

Conclusion

- Proposed a clustering-based technique to divide the execution intervals of a workload into phases
- Demonstrated the correlation of phases across workloads
- Proposed an auction-based scheduling scheme where each thread has a replenishable virtual wallet to bid for cores
- Used hill climbing to reach a Nash solution and store the bidding strategies of the bidders in the pattern table
- 18% performance improvement
- 17% ED^2 improvement

Questions