

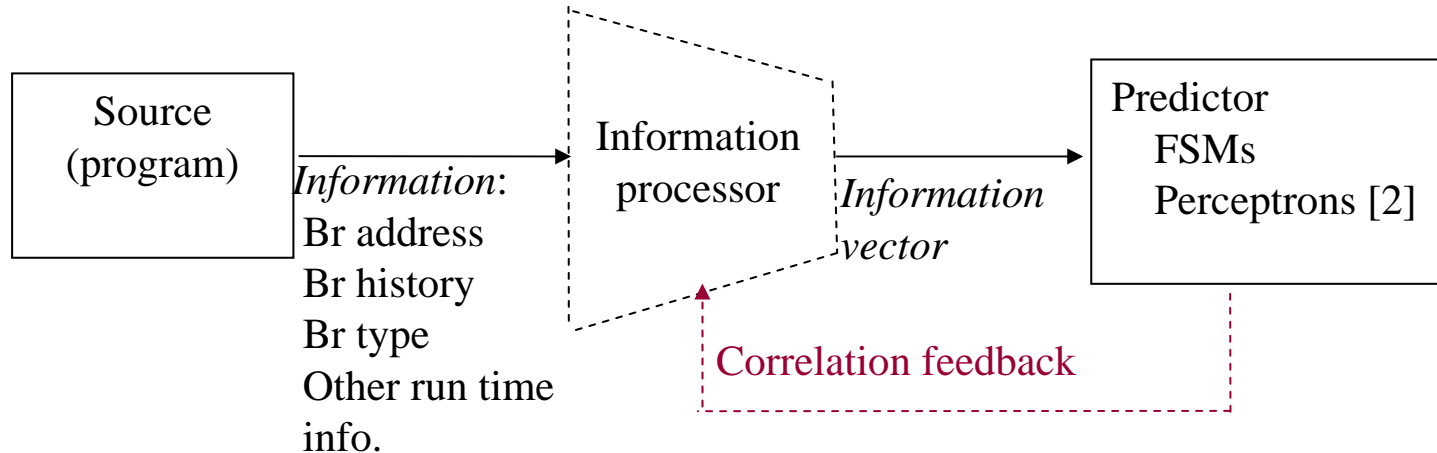
Adaptive Information Processing: An Effective Way to Improve Perceptron Predictors

Hongliang Gao Huiyang Zhou



Computer Science Department
University of Central Florida

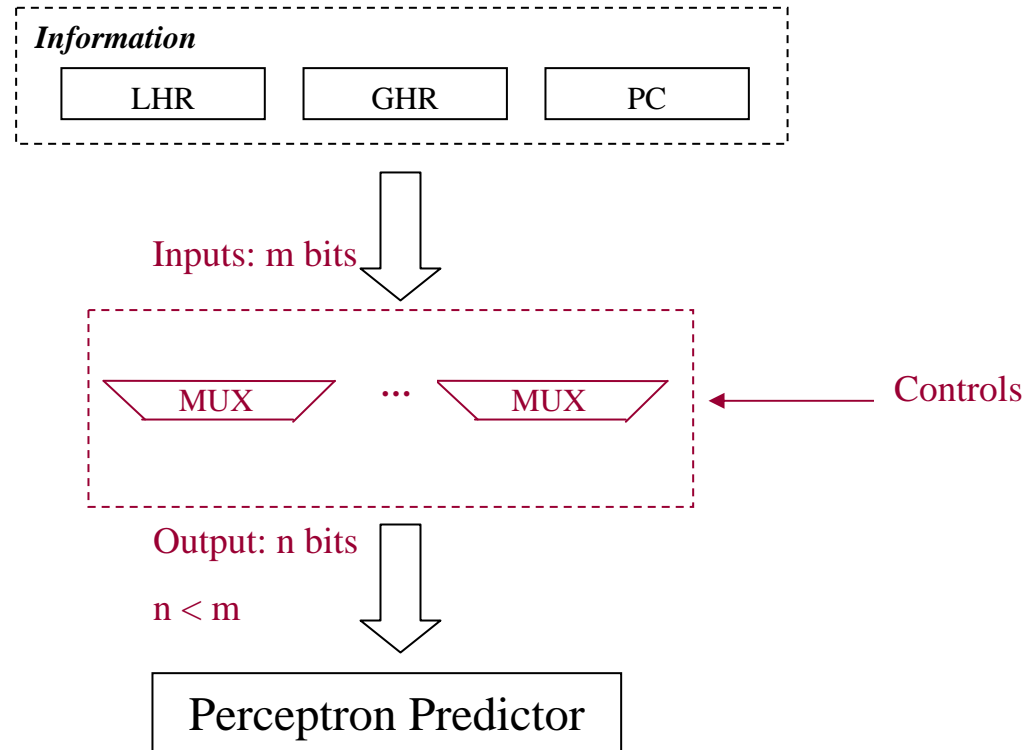
Information system model



- Information system model by Chen et. al. [ASPLOS-VII].
- Key observations
 - Shortcomings:
 - Fixed information vector while different workloads/branches need different information data.
 - Perceptron weights \leftrightarrow Correlation
 - Assemble information vector to maximize correlation
- **Our contribution**
 - Re-assemble the information vector based on correlation (weights)
 - Performed at a coarse grain, so it is not latency critical



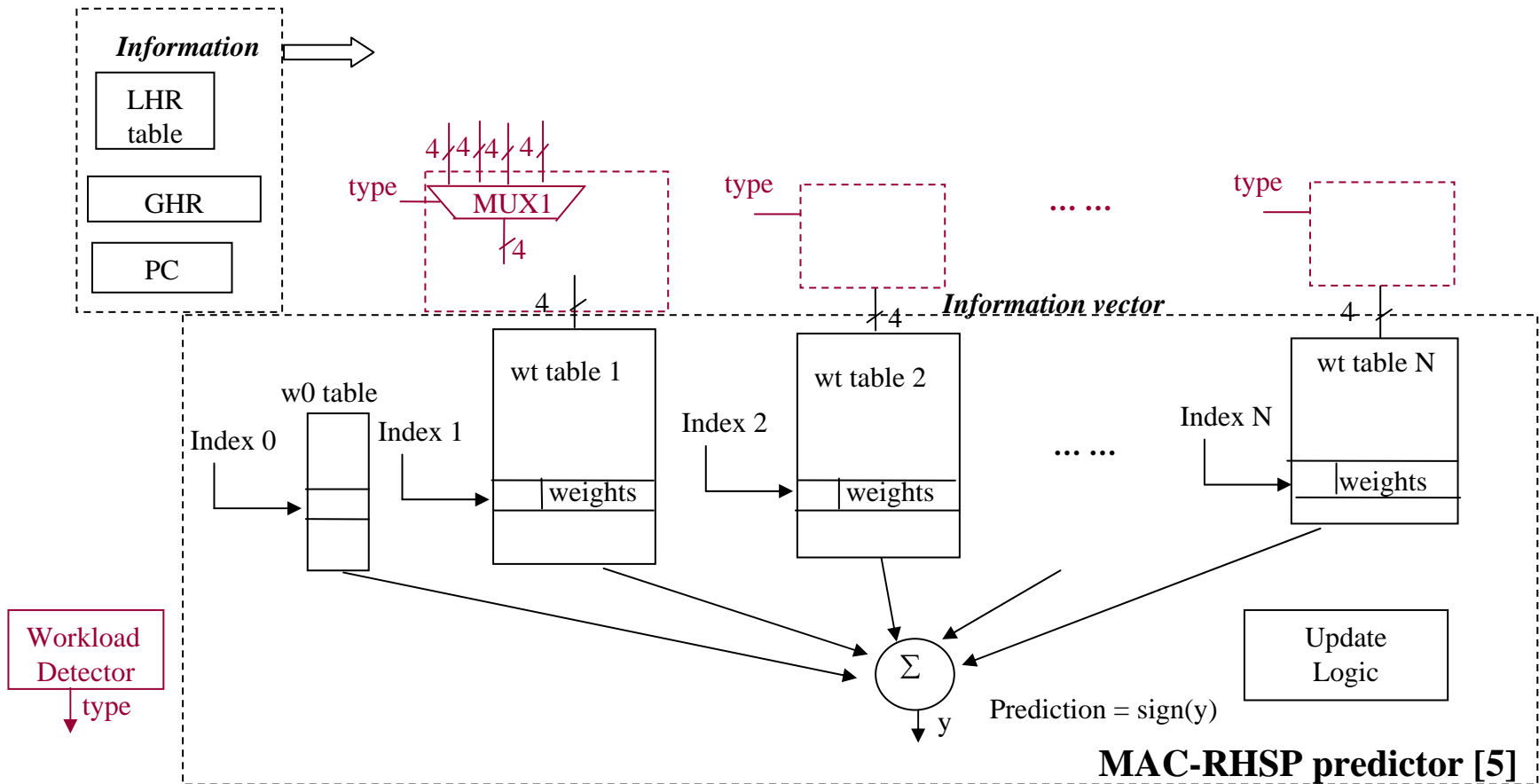
Adaptive Information Processing



- Profile-directed adaptation
- Correlation-directed adaptation



Profile-directed adaptation





Profile-directed adaptation

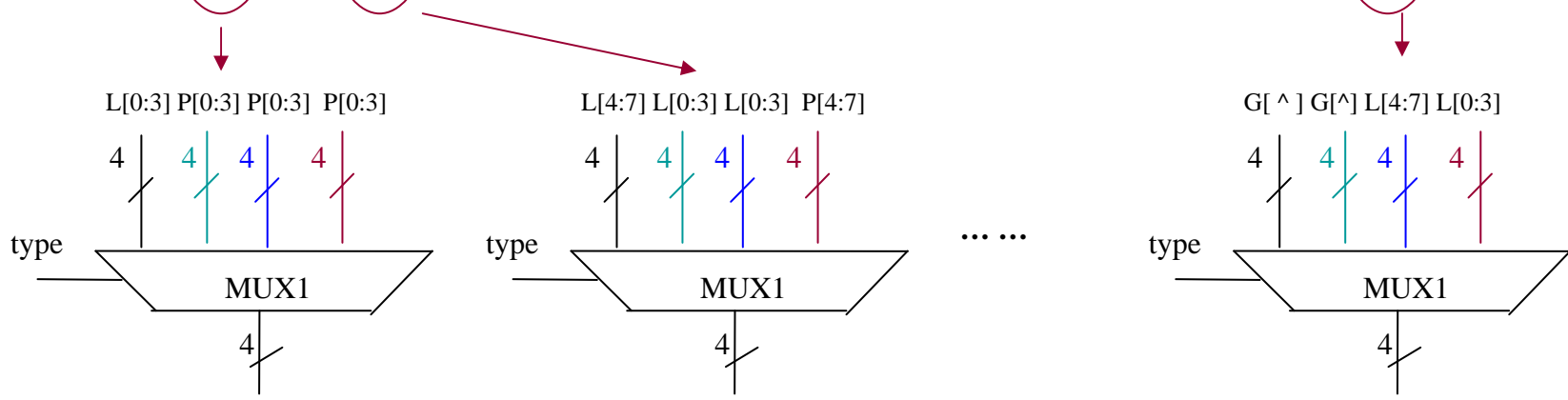
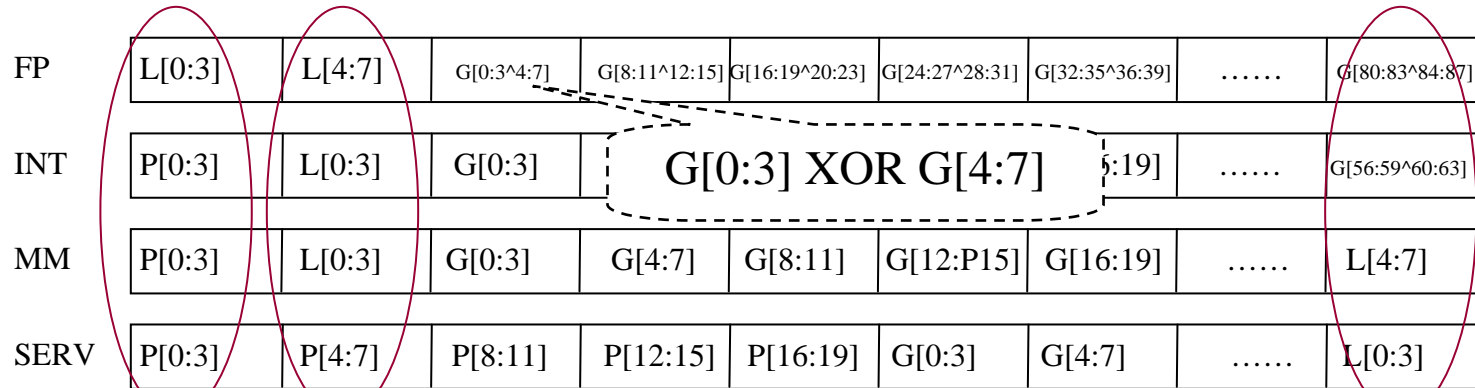
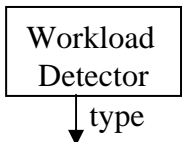


Table 1

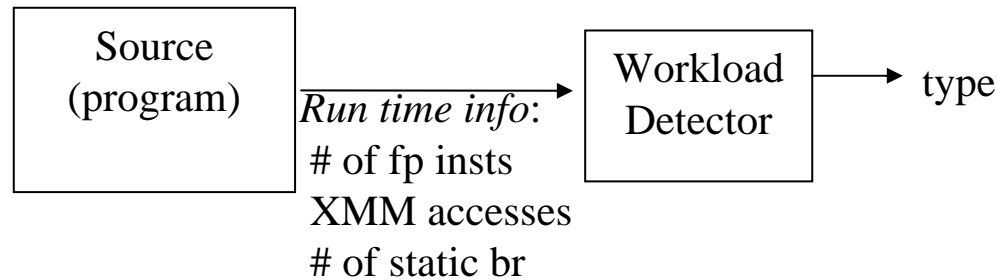
Table 2

Table N



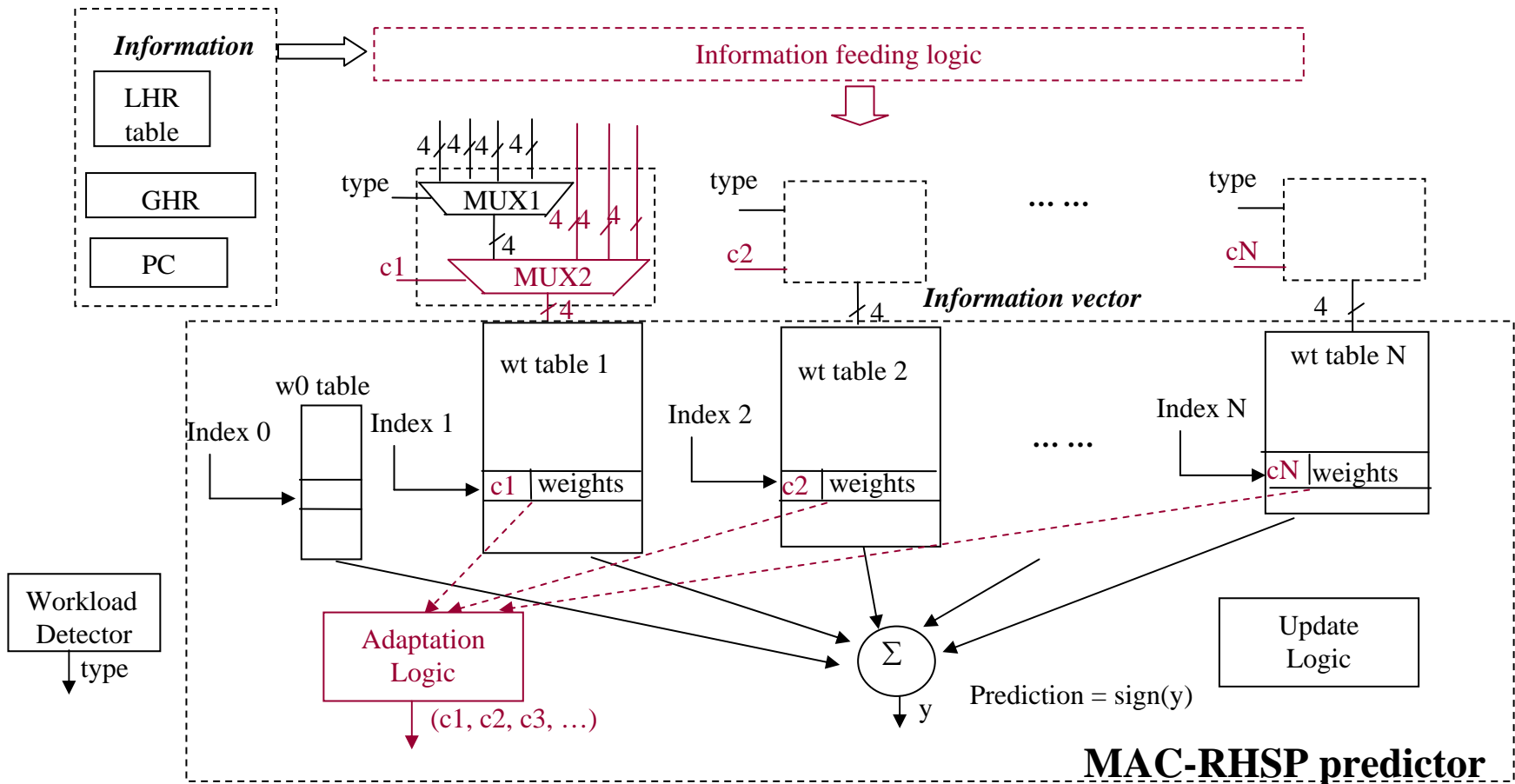


Workload Detector

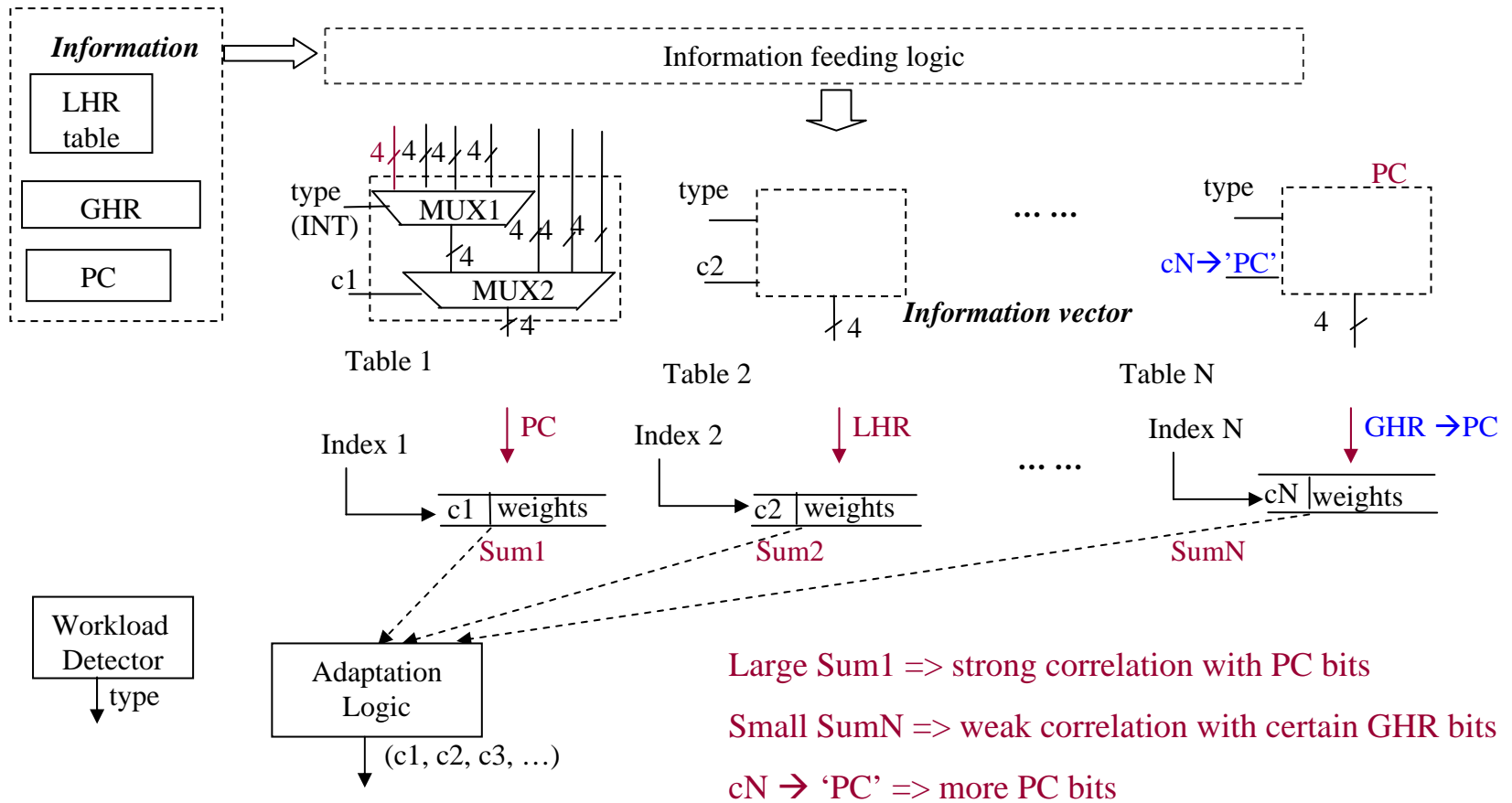


- Detection criteria
 - **SERV**: a **large** number of **static branches**
 - **FP**: a **small** number of **static branches**, a **high** number of **floating point operation**, and a **high** number of instructions using **XMM registers**
 - **MM**: a **medium** number of **static branches**, a **medium** number of **floating point operation**, and a **medium** number of instructions using **XMM registers**
 - **INT**: default

Correlation-directed adaptation

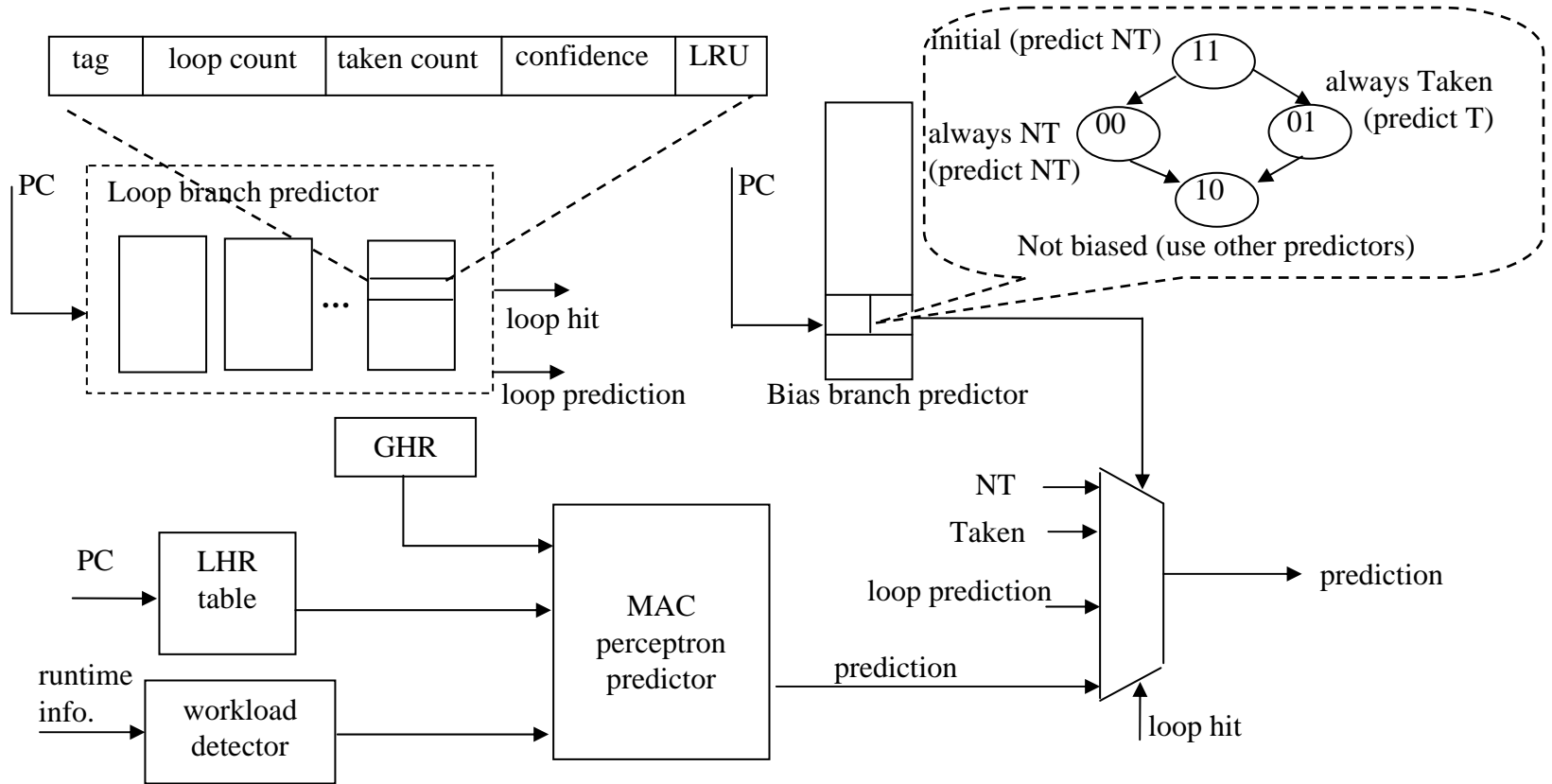


Correlation-directed adaptation





Overall scheme





Summary

- Observations
 - Different workloads/branches need different information.
 - Perceptron weights \leftrightarrow Correlation
- Contributions
 - Profile-directed adaptation
 - Correlation-directed adaptation
 - Reducing aliasing from bias and loop branches
- Result
 - Significant improvement

Thank you and Questions?



Computer Science Department
University of Central Florida



References

- [1] I. K. Chen, J. T. Coffey, and T. N. Mudge, “Analysis of branch prediction via data compression”, *Proc. of the 7th Int. Conf. on Arch. Support for Programming Languages and Operating Systems (ASPLOS-VII)*, 1996.
- [2] D. Jimenez and C. Lin, “Dynamic branch prediction with perceptrons”, *Proc. of the 7th Int. Symp. on High Perf. Comp. Arch (HPCA-7)*, 2001.
- [3] D. Jimenez and C. Lin, “Neural methods for dynamic branch prediction”, *ACM Trans. on Computer Systems*, 2002.
- [4] S. MacFarling, “Combining branch predictors”, *Technical Report*, DEC, 1993.
- [5] A. Sez nec, “Revisiting the perceptron predictor”, *Technical Report*, IRISA, 2004.
- [6] T.-Y. Yeh and Y. Patt, “Alternative implementations of two-level adaptive branch prediction”, *Proc. of the 22nd Int. Symp. on Comp. Arch (ISCA-22)*, 1995.



Predictor configuration

COMPONENT	CONFIGURATION	COST
Bias branch predictor	2293 entries	$2293 \times 2 = 4586$ bits.
Loop branch predictor	24 entries, 8-way set associative	1344 bits
Information	PC GHR: 100 bits LHR: 8 bits, 63 entries	$32 + 100 + 8 * 63 = 636$ bits
MAC perceptron predictor	W0 table: 61 entries, 8 bits each Other table sizes: 63, 55, 53, 53, 51, 49, 43, 41, 41, 39, 37, 37, and 35. Total MAC entries: 597 Each entry: 16 weights, 6 bits each Control bits: 2 bits each entry	$61 * 8 +$ $597 * 16 * 6 +$ $597 * 2$ $= 58994$ bits
Adaptation	Adaptation interval : $100000 * 2$ conditional branches	22 bits
Workload detection	Interval: 10000 (instructions / conditional branches)	82 bits
		Total : 65649 bits (less than $64 \times 1024 + 256 = 65792$ bits)