

MLP-Aware Dynamic Instruction Window Resizing for Adaptively Exploiting Both ILP and MLP

Yuya Kora
Kyohei Yamaguchi
Hideki Ando

Nagoya University

Problem to Solve

- Difficult to improve single-thread performance in memory-intensive programs
 - Memory wall
- Very large instruction window can overcome this problem by exploiting MLP
 - This degrades the clock cycle time
 - Can be solved by pipelining, but...
 - Pipelining prevents ILP from being exploited, degrading IPC in compute-intensive program

Dynamic Instruction Window Resizing

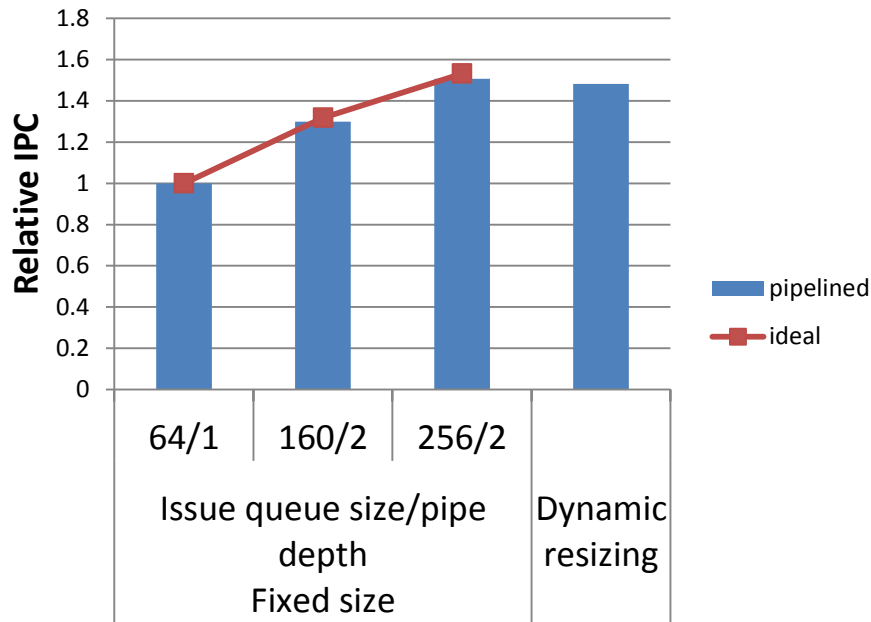
- Adapt window size to available parallelism
 - ILP or MLP
- As more exploitable MLP is predicted
 - Window resources are enlarged and pipeline depth is increased
- If prediction indicates less MLP is exploited (= ILP is more valuable)
 - Window resources are shrunk and pipeline depth is decreased

Prediction when MLP is Exploitable

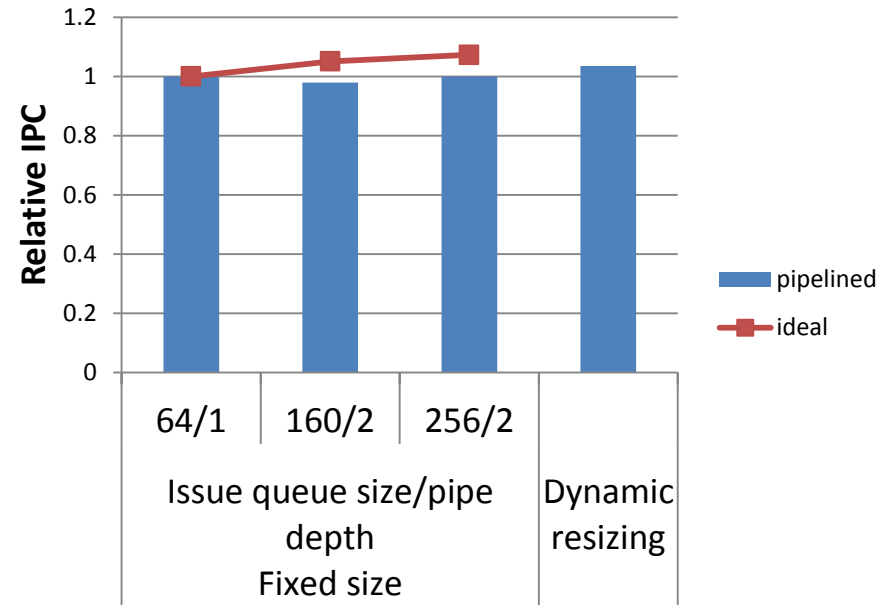
- If an LLC miss occurs once
 - Predict that MLP is exploitable for a while
- If memory latency has lapsed after the last LLC miss
 - Predict that MLP will not be exploitable
- Rationale
 - LLC misses are clustered in terms of time

IPC

GM memory-intensive

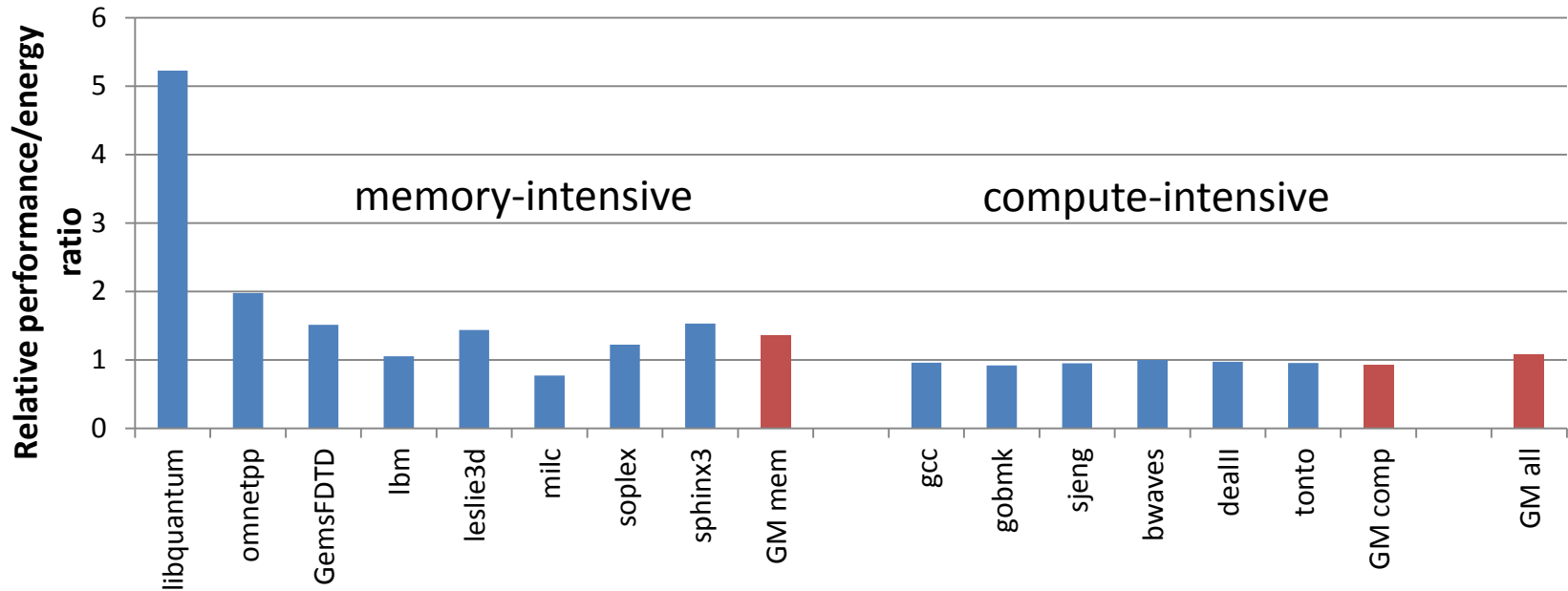


GM compute-intensive



- Dynamic resizing model achieves as good as **best** performance for levels 1 to 3 of fixed size model.
- It achieves similar performance to ideal model (no pipelined).
- Imply good adaptability
- **21%** speedup for all programs


Energy Efficiency



- Power is increased, but perf is improved \Rightarrow Better energy efficiency
- Memory-intensive: **36% better**
- Compute-intensive: **8% worse**
- Overall: **8% better**

Cost Efficiency

| | | |
|-----------------|---------------------------|--------------------|
| Additional cost | value (per core) | 1.6mm ² |
| | vs. base core | 6% |
| | vs. Sandy Bridge core | 8% |
| | vs. Sandy Bridge chip | 3% |
| Speedup | achieved | 21% |
| | expected by Pollack's law | 3% |
| | augmented L2 cache | 1% |

 2MB, 4-way → 2.5MB, 5-way
(increased cost is 1.3x greater than the additional cost)

Good cost/performance ratio,
that far exceeds that based on Pollack's law

Conclusion

- Dynamic instruction window resizing
 - Exploit ILP and MLP adaptively
 - Based on prediction of available parallelism
- Features
 - Very simple
 - Very practical
- Our scheme achieves
 - Performance level similar to the **best** performance achieved with fix-sized resources
 - **21%** speedup
 - **6%** extra cost of a core, or **3%** of an entire proc chip
 - **8%** better energy efficiency