Limit of Thread-Level Parallelism on Partitioning Levels and Speculations in Non-Numerical Programs

Akio Nakajima[†‡], Ryotaro Kobayashi[†],
Hideki Ando[††], Toshio Shimada[†]

[†] Department of Electrical Engineering and Computer Science, Nagoya University

[††] Department of Computational Science Engineering, Nagoya University

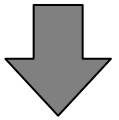
[‡] Currently with Hitachi Ltd.

Outline

- Background
- Goal
- Models of Thread Partitioning Level
- Constraint Relaxing Techniques
- Evaluation
- Conclusion

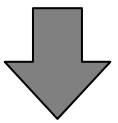
Background

Superscalar processor is reaching its limit



Chip multiprocessors (CMPs)

- Available with the advance of LSI technology
- Exploit thread-level parallelism (TLP)



Insufficient speedup in non-numerical programs

Goal

Beneficial techniques to obtain high TLP in a non-numerical program

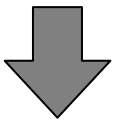
- Approach
  - Explore the TLP limit
  - Impose only constraints associated with a technique
- Techniques
  - Level of partitioning
  - Constraint relaxing techniques

Models of Thread Partitioning Level

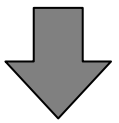
- SP model (no partitioning)
- FC model (function level)
  - Thread contains a callee function
- LP model (loop level)
  - Each thread contains each loop iteration
- PD model (basic block level)
  - Threads have no control-dependences on their fork point

Speculative Thread Execution

Branches frequently appear in non-numerical programs



Control dependences severely limit TLP



Speculative thread execution

- Threads are created and start execution soon after the fork point is speculatively fetched

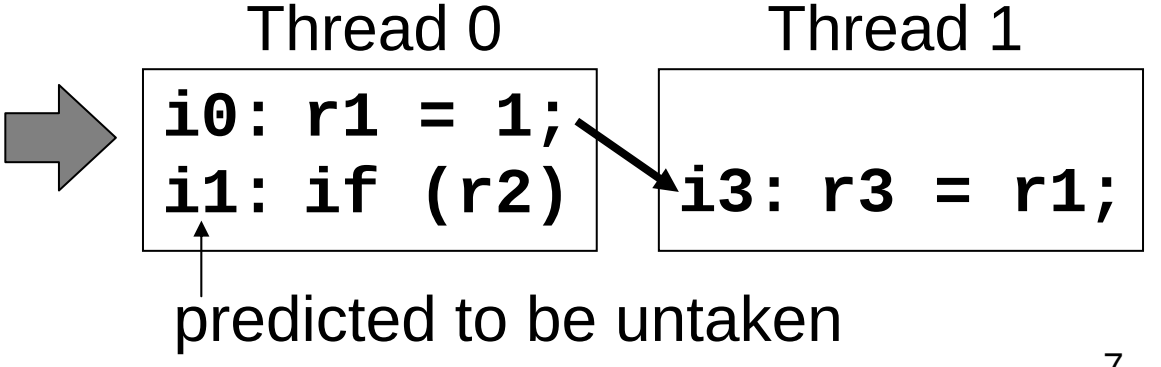
Speculative Register Communication

- Constraint
  - Register communication must wait until the definition is determined to reach the consumer
- Speculative Register Communication
  - Rely on branch prediction

Code example

```
i0: r1 = 1;
i1: if (r2)
i2:    r1 = 2;
i3: r3 = r1;
```

Execution of threads

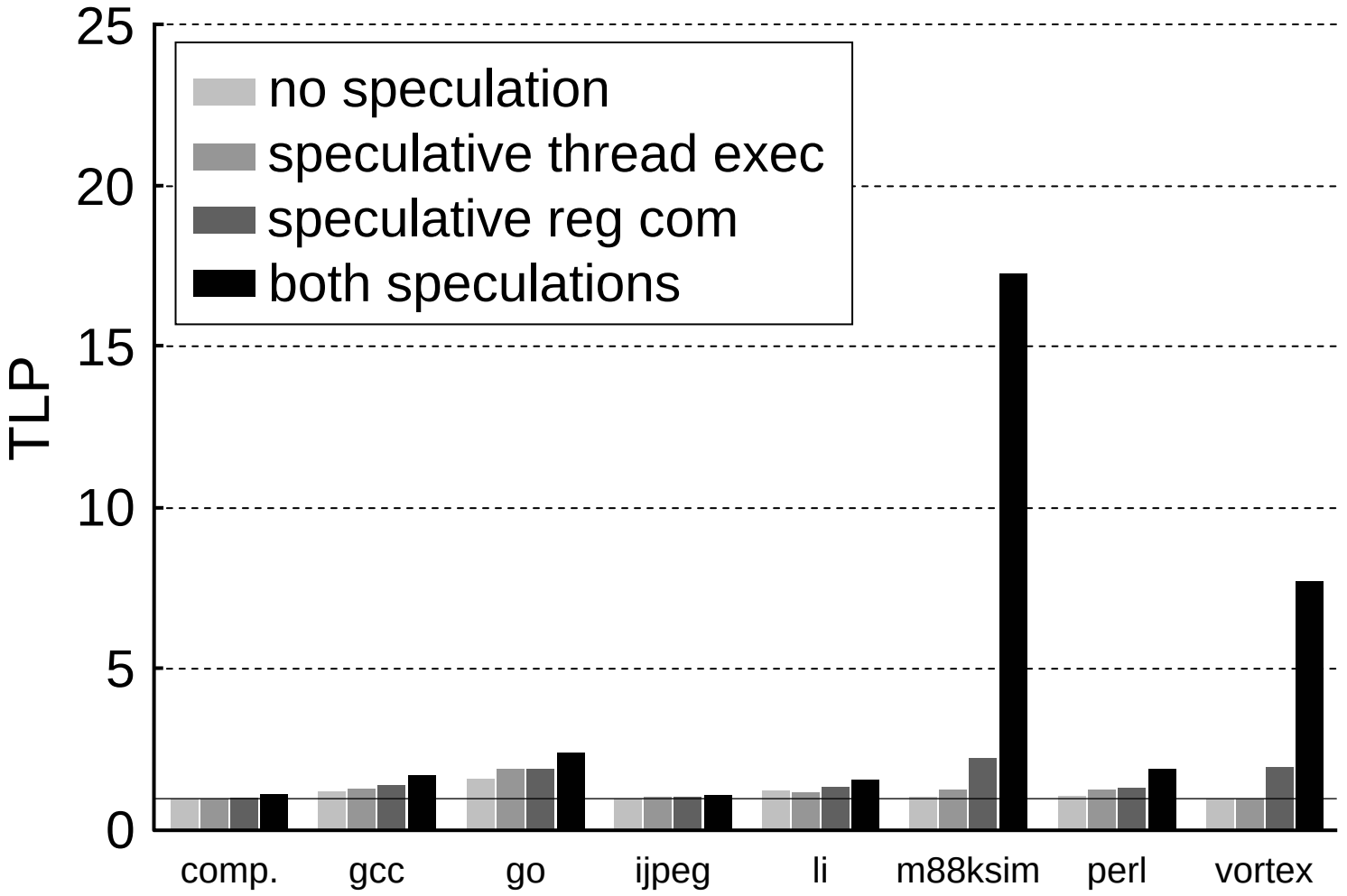


Evaluation Environment

- Benchmarks: 8 programs of SPECint95
- Latency of any instruction: a single cycle
- Branch predictor: PAs with large tables
- Memory disambiguation is ideally removed
- No resource constraints

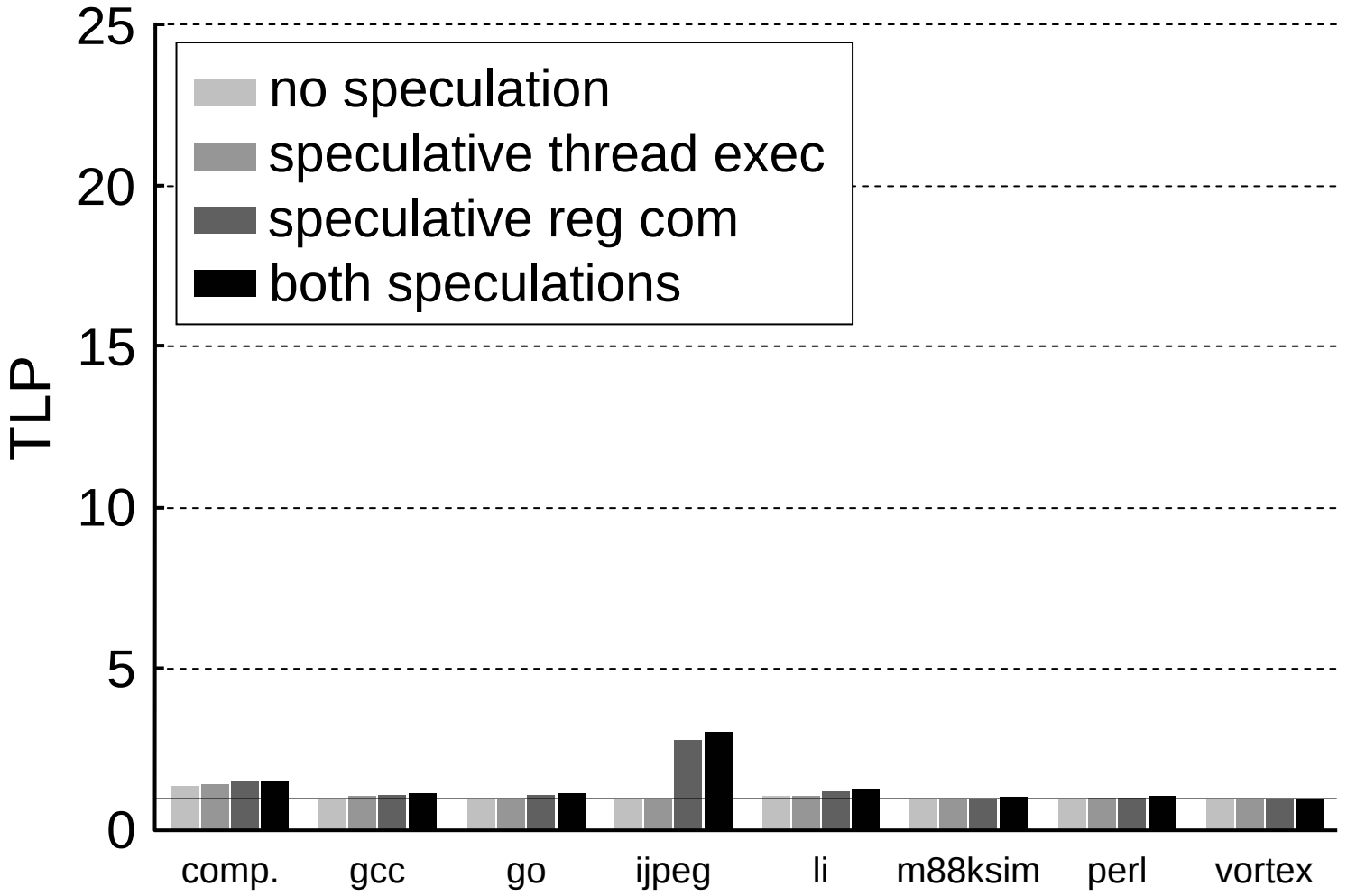
  Issue width, function units, etc: infinite
- No overhead of executing parallel threads

TLP of FC Model



TLP is severely limited due to control dependence constraints

TLP of LP Model



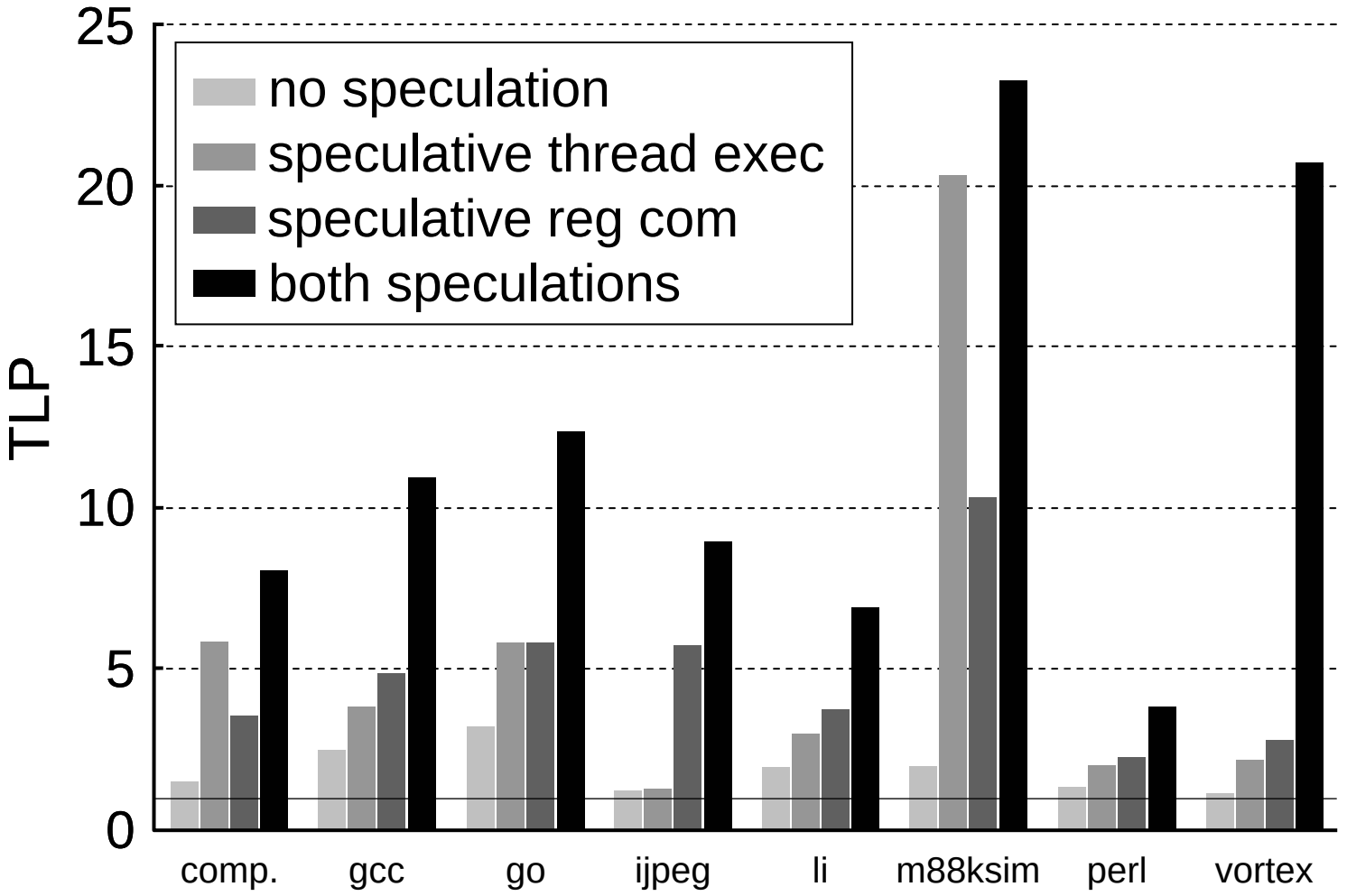
TLP is severely limited due to small number of loop iterations

Number of loop iterations

Benchmark	# loop iterations
compress95	8.2
gcc	2.4
go	2.9
ijpeg	16.8
li	2.4
m88ksim	2.8
perl	3.2
vortex	2.4

The number of loop iterations is very small

TLP of PD Model



Speculations significantly increase the TLP to 10.3

Conclusion

- Speedup of CMPs is insufficient in non-numerical programs
- We evaluated TLP limit to find the effect of the techniques:
  - Thread partitioning
  - Speculative thread execution
  - Speculative register communication
- Evaluation results
  - Loop and function level partitioning is not useful
  - Basic block level partitioning with the speculations is essential to obtain high TLP