道 ... a 'way', 'path', often used to signify the true nature of the world

TAO: *T*wo-level *A*tomicity for Dynamic Binary *O*ptimizations

Edson Borin, **Youfeng Wu,** Cheng Wang, Wei Liu,
Mauricio Breternitz Jr., Shiliang Hu, *Esfir Natanzon, *Shai Rotem, *Roni Rosner

MPR/Programming Systems Lab, Intel Labs

*MCD/Microprocessor Architecture, Intel Architecture Group

Intel Corporation

April 26, 2010

Agenda

- Atomic execution support
- Optimization scope vs rollback penalty
- Two level atomicity
- Preliminary results
- Conclusions & Future Work

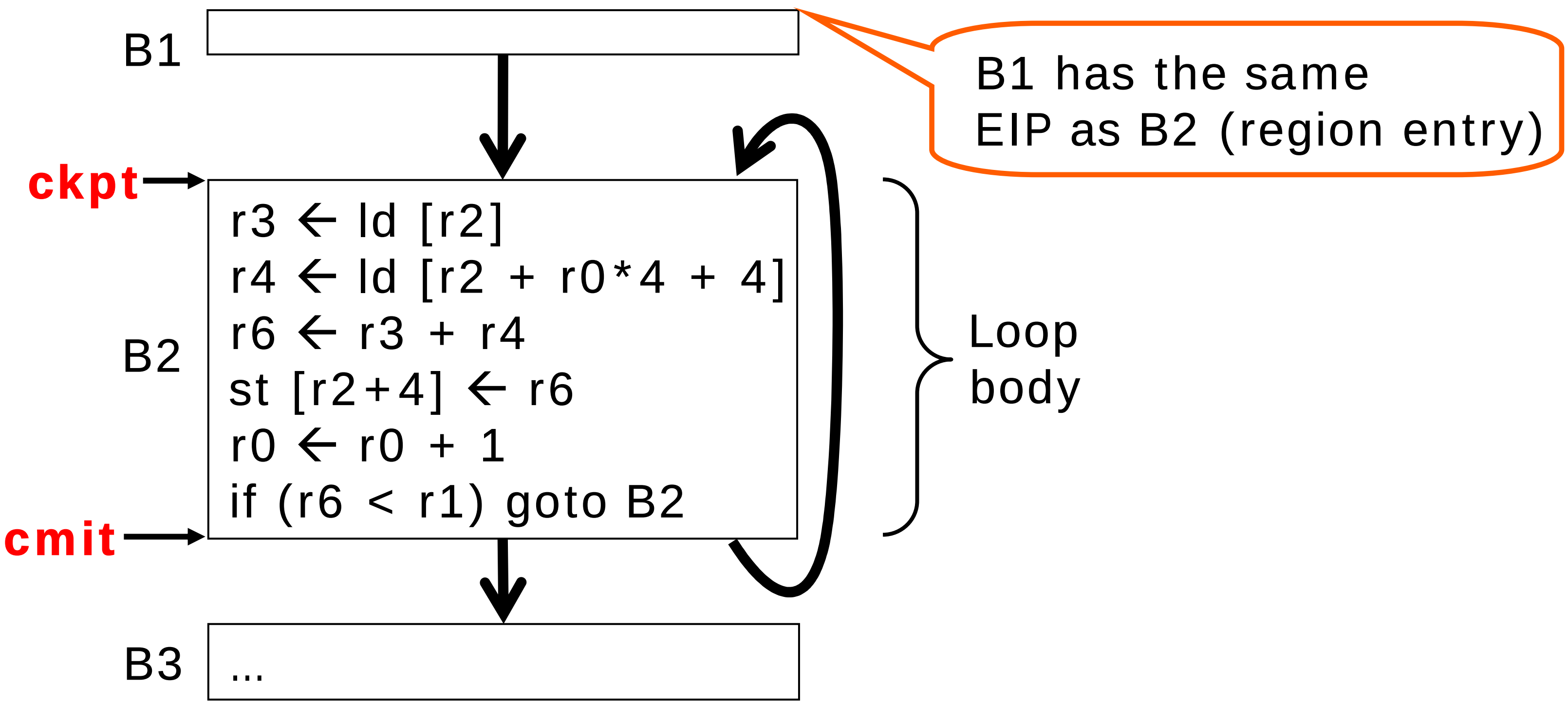
Atomicity and Binary Optimizations

- Binary optimizations are very limited without atomicity support
  - Many optimizations are not allowed: cannot reorder load/load, store/store, early load/later store
  - Many hard issues: memory accesses across cache line boundary, non-cacheable memory operation, precise exception, alias speculation, etc
- Atomic regions
  - Increase the optimization opportunities
  - Address many tough issues
  - Simplify the optimizations design

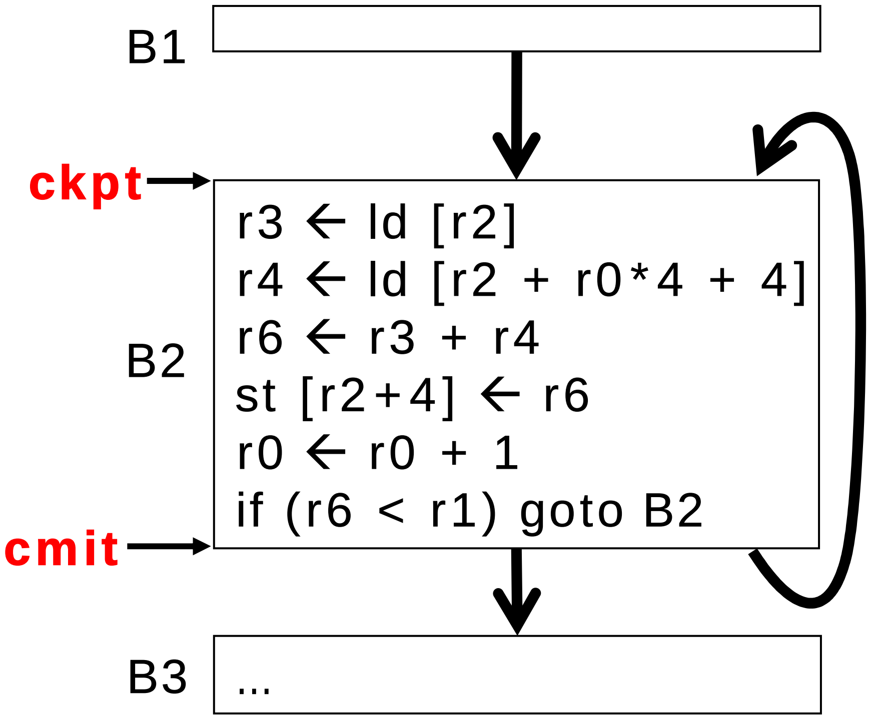
Atomic Region Scope

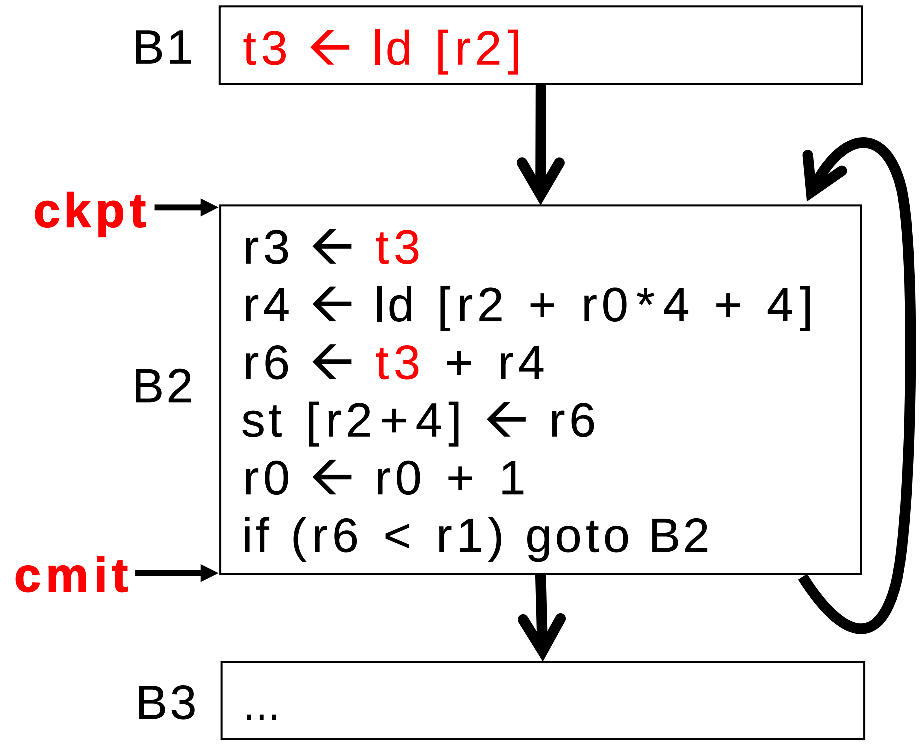
- **Small** atomicity scope
  - Traces, super-blocks, basic blocks, etc
  - Fast local optimizations, but small opportunities
- **Large** atomicity scope
  - Loops or large DAGs
  - Global optimizations, loop invariant hoisting, software pipelining, long range prefetch, etc
  - May suffer from resource overflow and large amount of work being discarded when rollback.

Small Atomicity Scope: Loop body



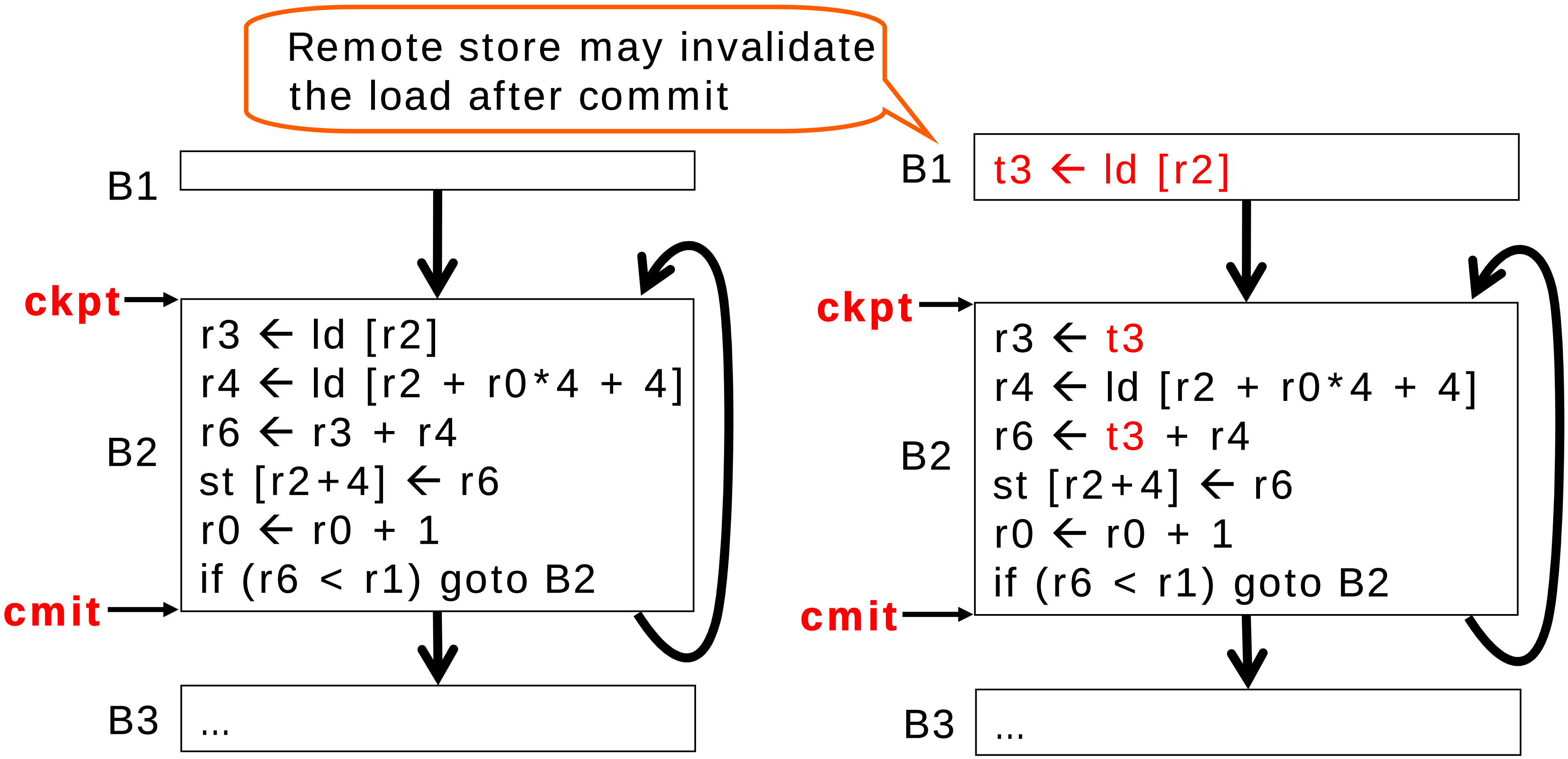
Small Atomicity Scope: Loop body





Optimizations: LICM +
copy propagation

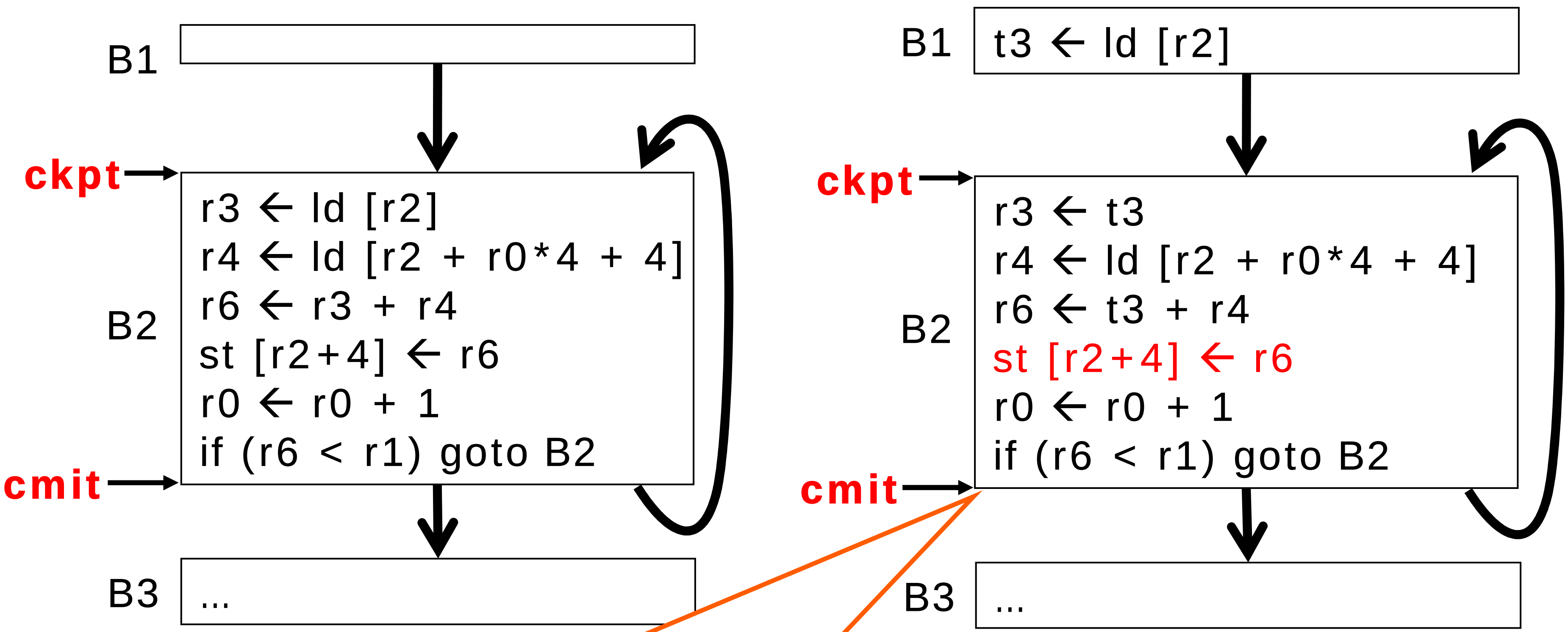
Small Atomicity Scope: Loop body



Optimizations: LICM + copy propagation

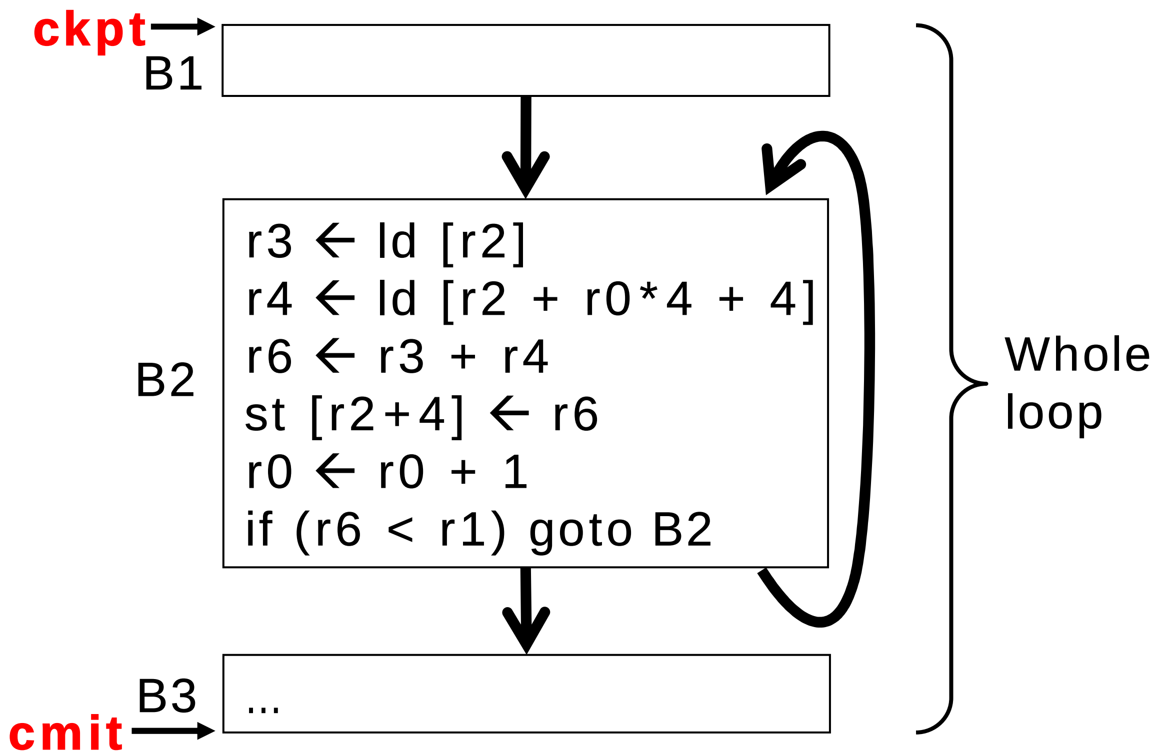
Binary Compilation Technology

Small Atomicity Scope: Loop body

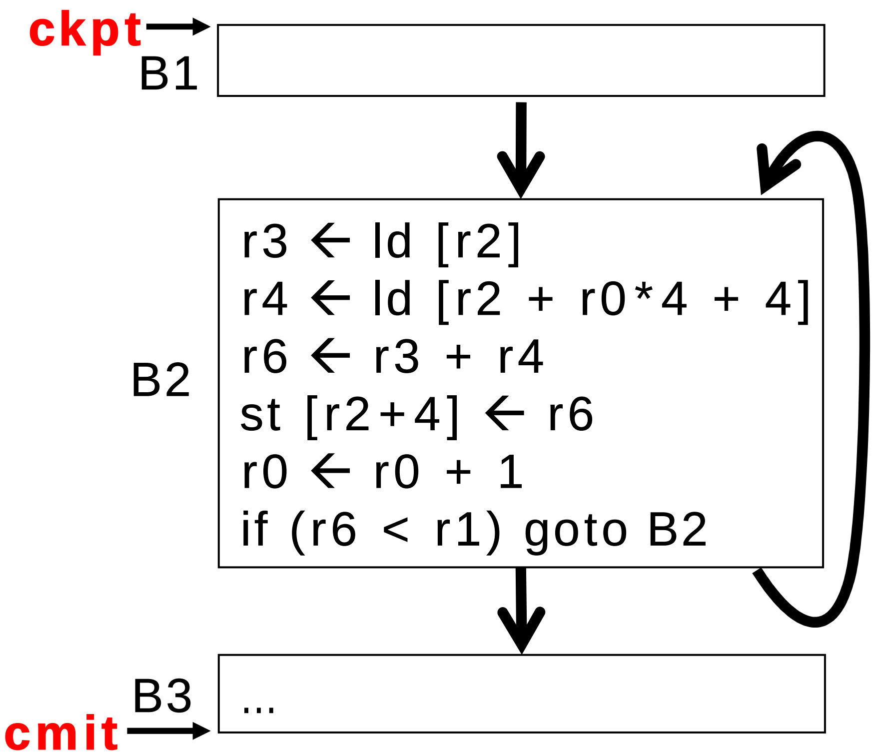


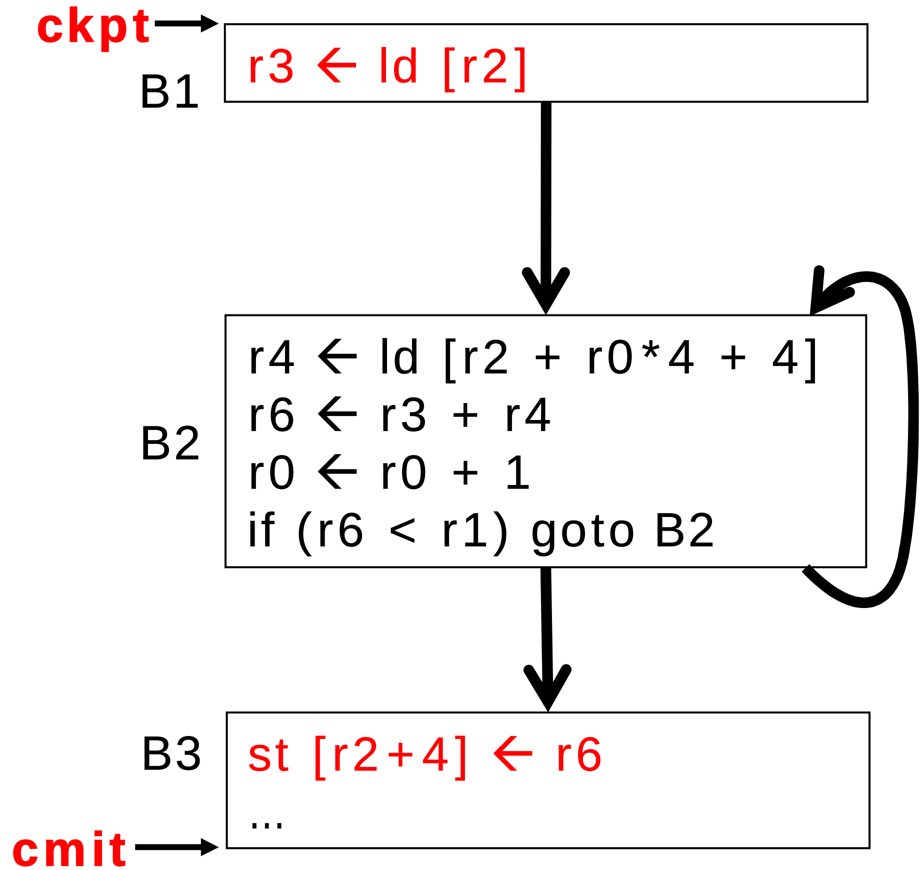
Commit requires precise architectural state:
Partially dead store elimination is invalid

Large Atomicity Scope: Whole loop



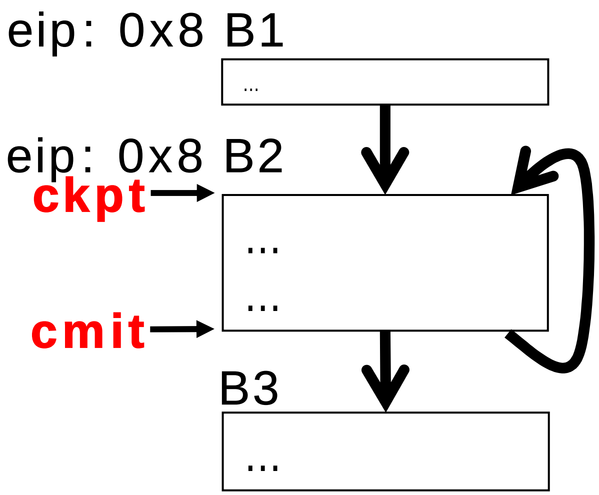
Large Atomicity Scope: Whole loop

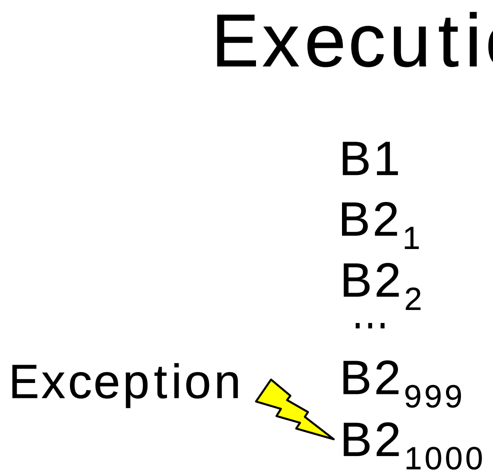




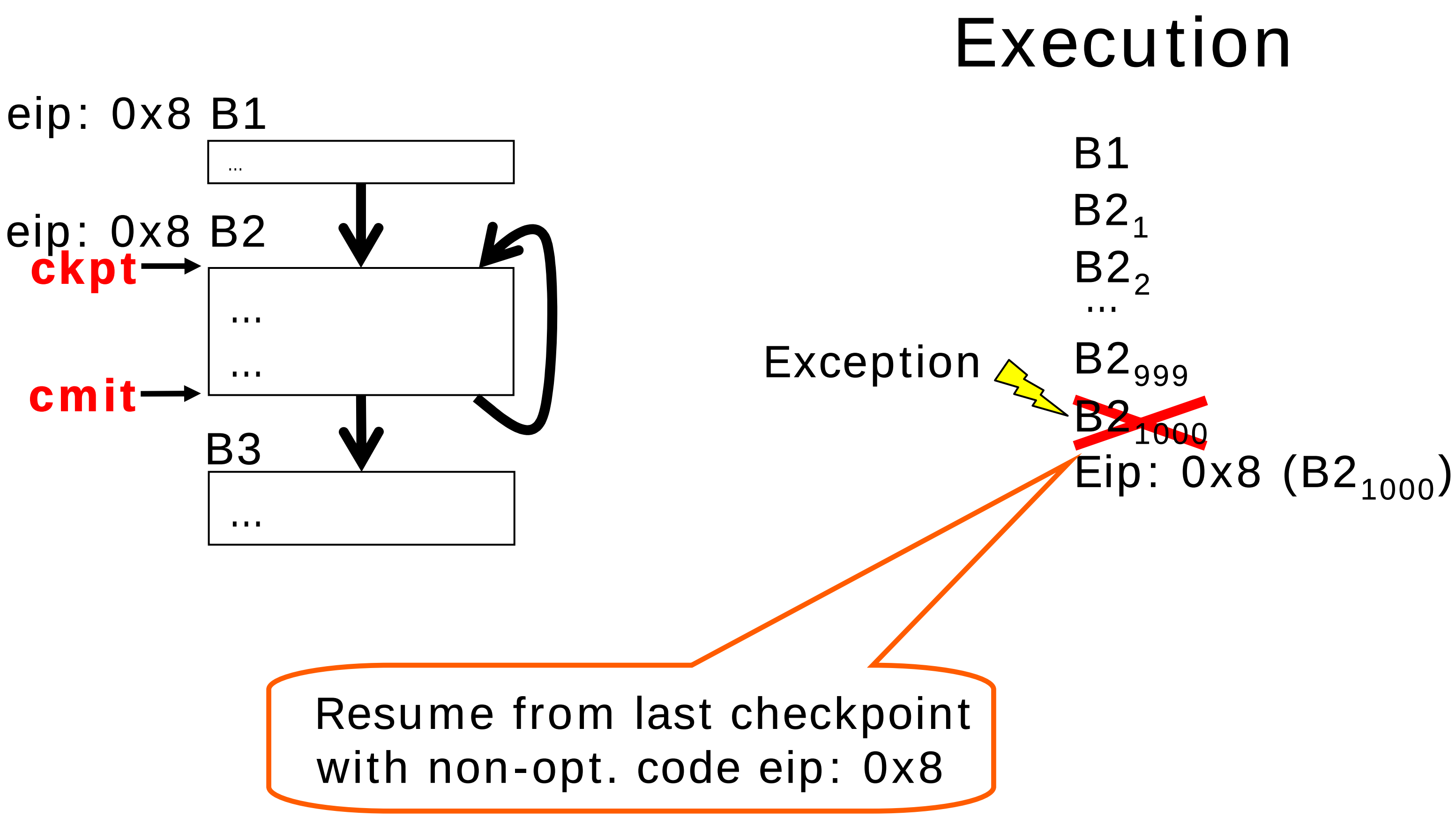
Optimizations: LICM + PDSE + copy propagation

Rollback Penalty: Small Atomicity Scope

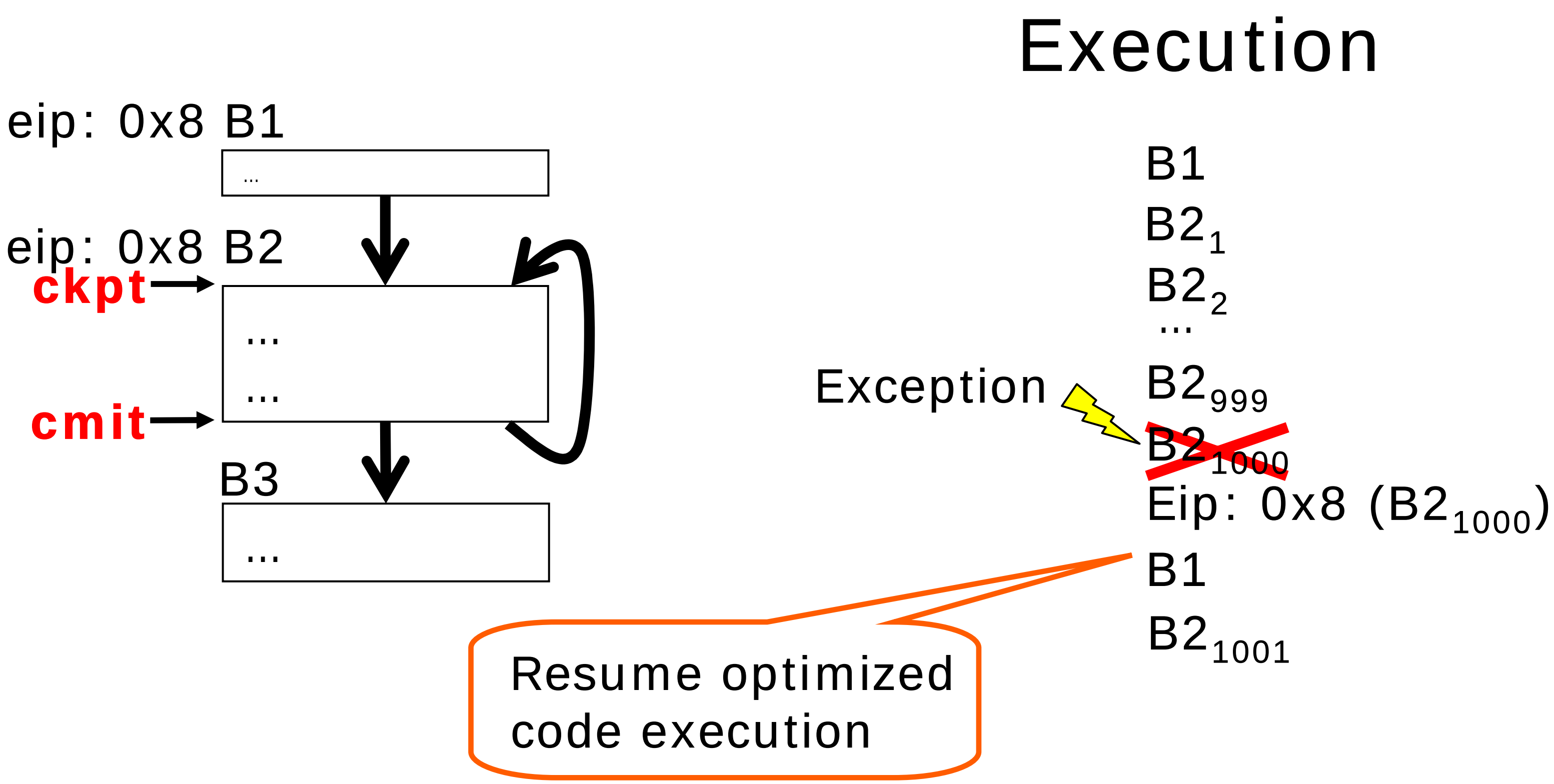




Rollback Penalty: Small Atomicity Scope

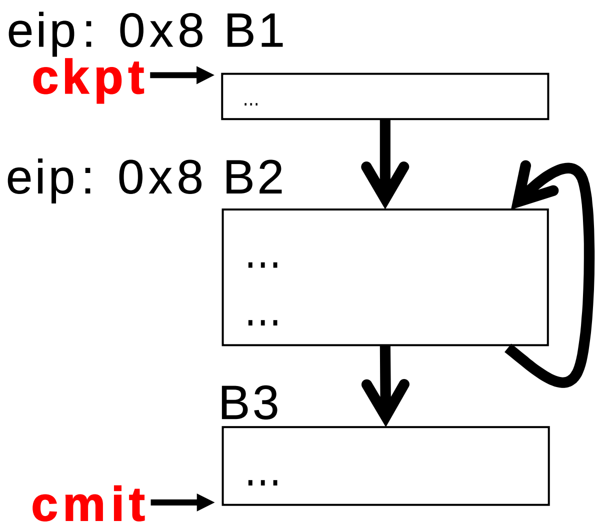


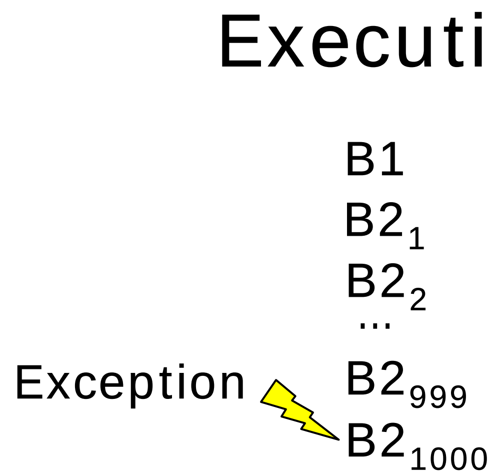
Rollback Penalty: Small Atomicity Scope



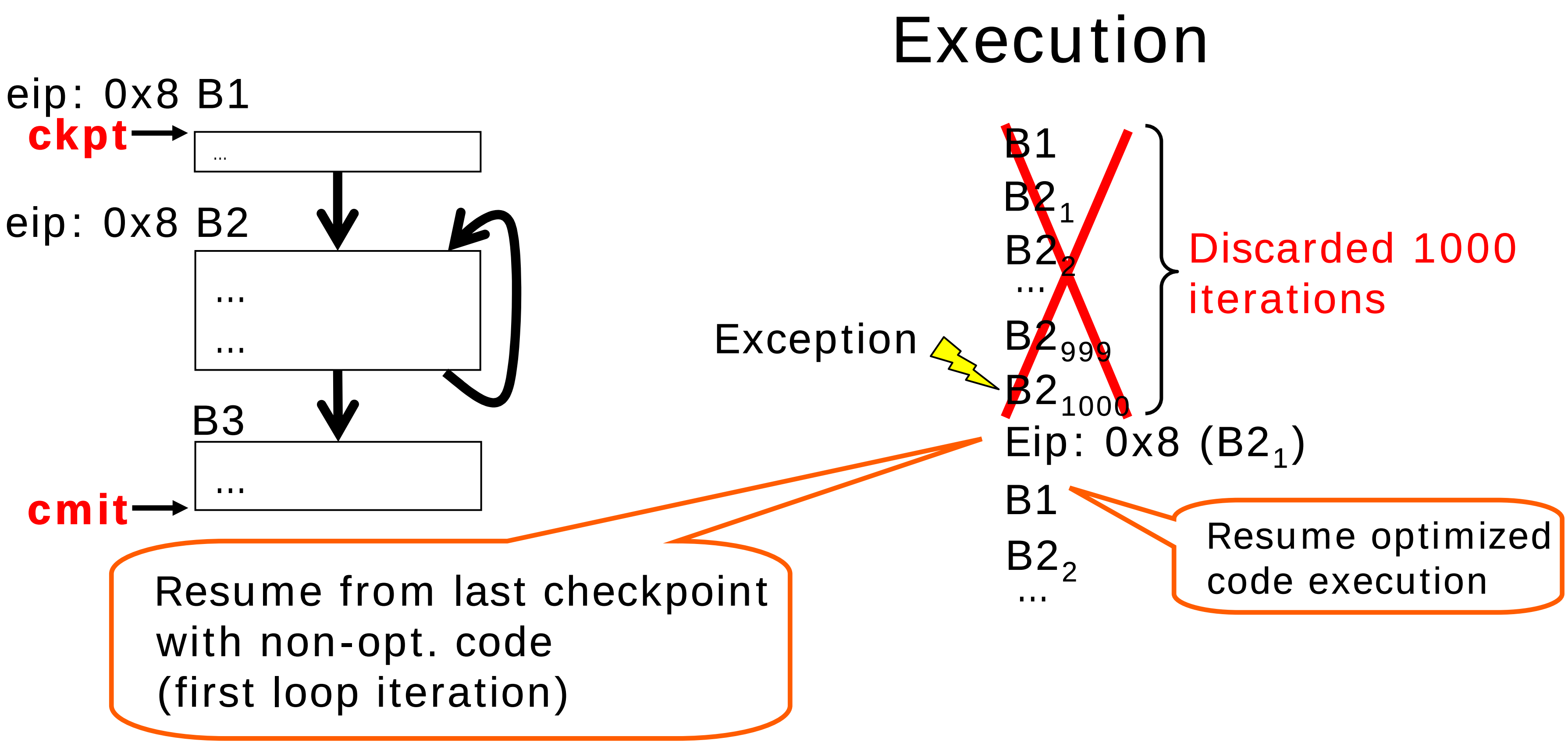
Amount of work discarded by rollback = 1 loop iteration

Rollback Penalty: Large Atomicity Scope

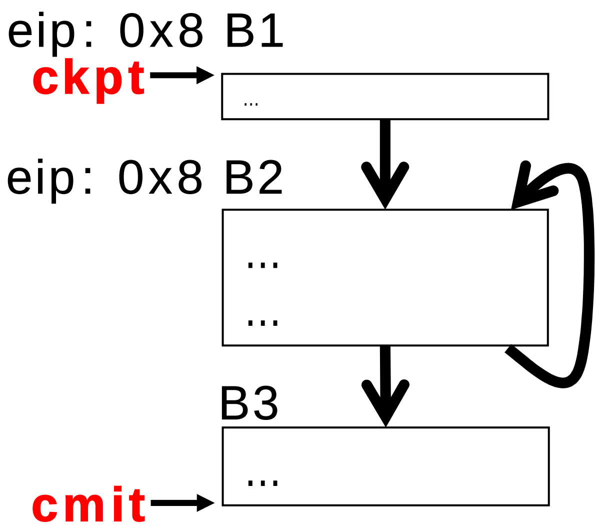


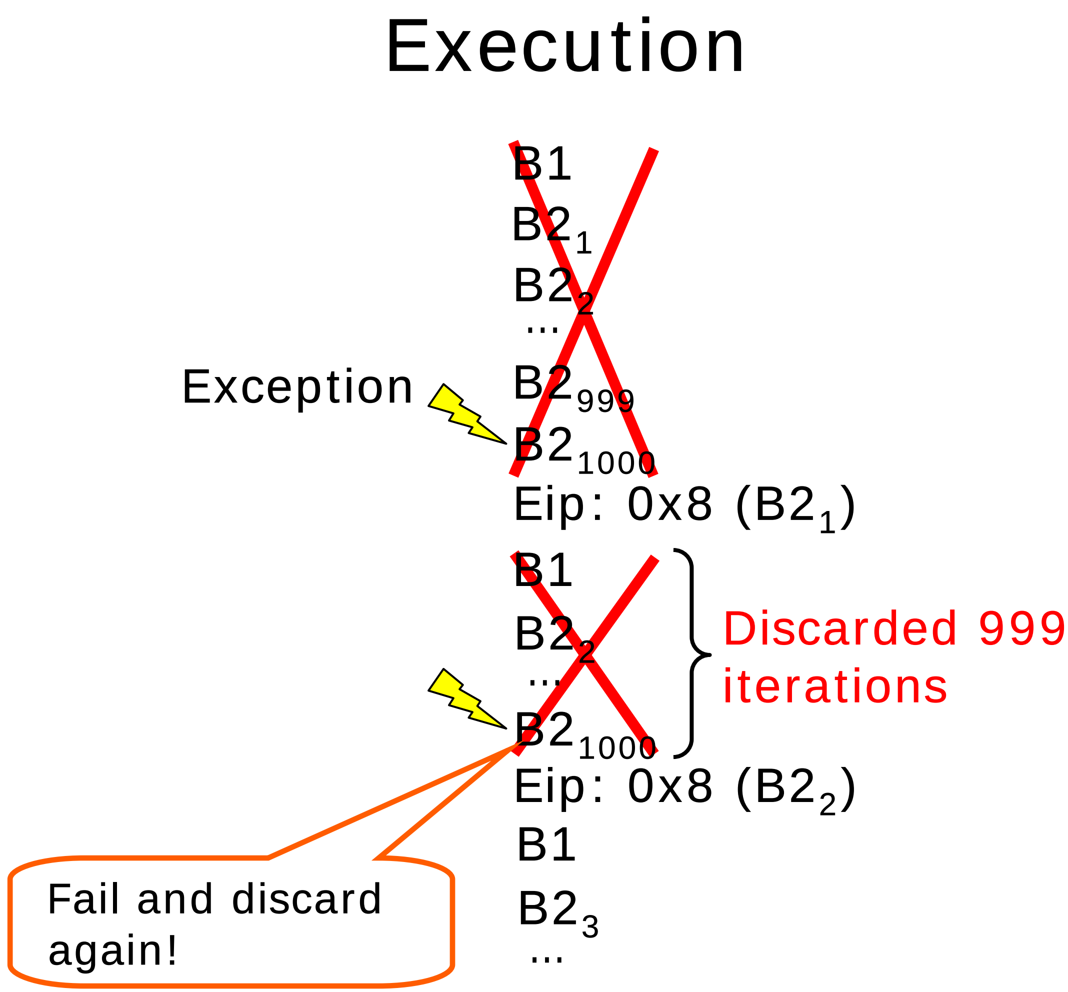


Rollback Penalty: Large Atomicity Scope

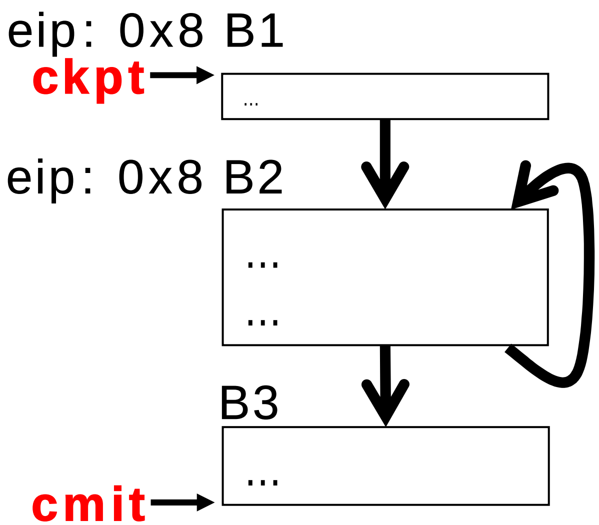


Rollback Penalty: Large Atomicity Scope





Rollback Penalty: Large Atomicity Scope

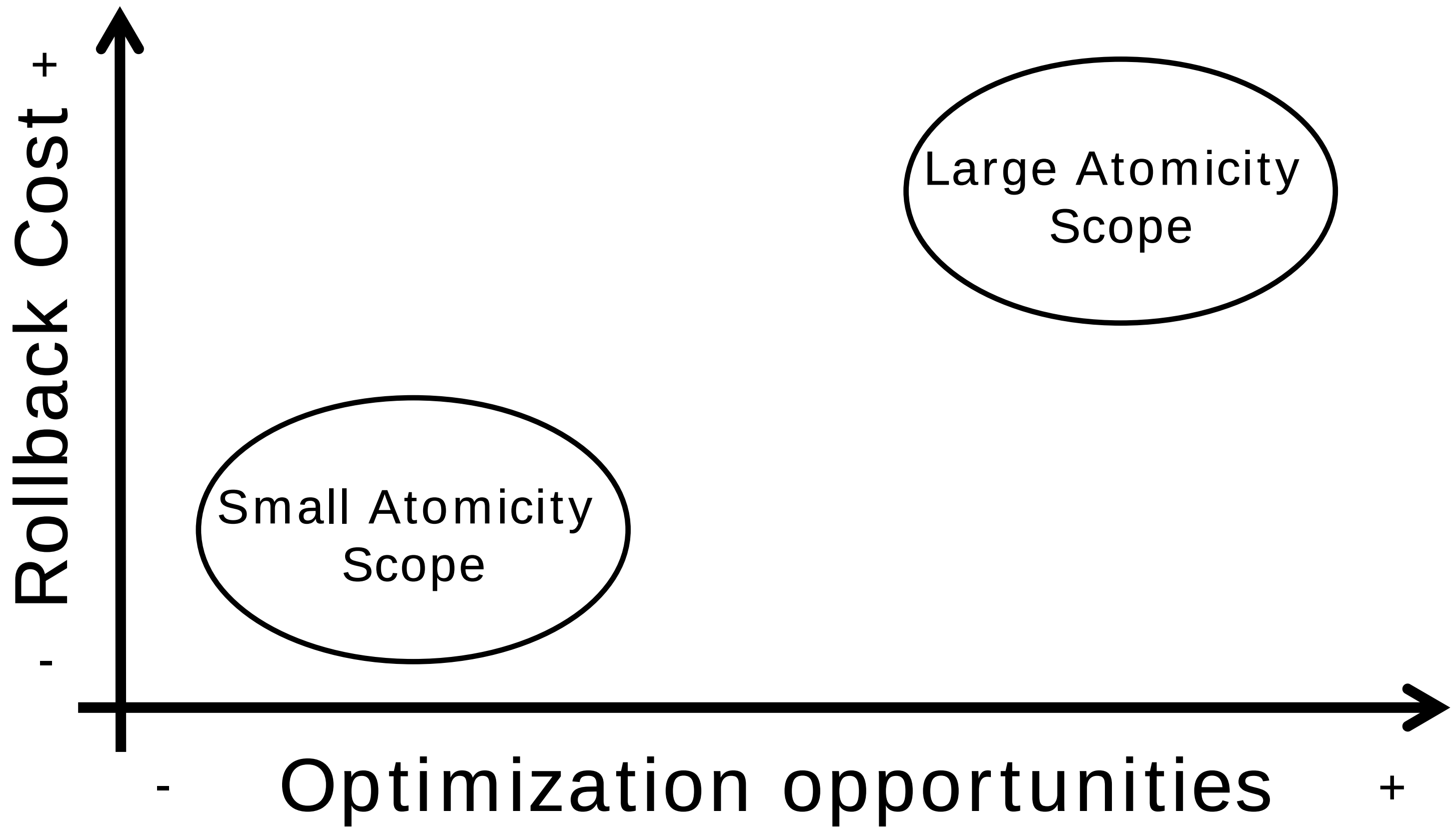


Execution

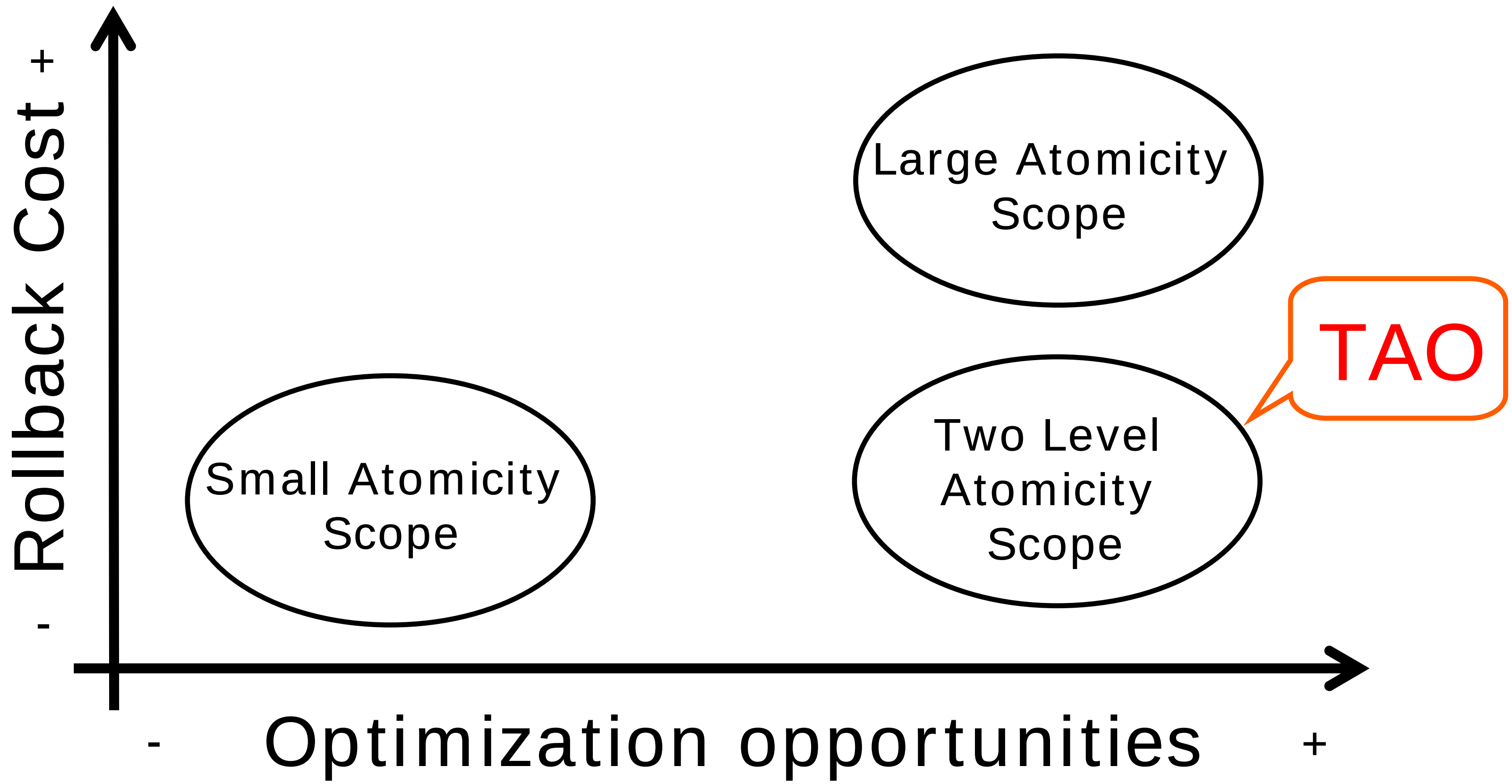
~~B1~~
~~$B2_1$~~
~~$B2_2$~~
~~...~~
Exception ~~$B2_{999}$~~
~~$B2_{1000}$~~
Eip: 0x8 ($B2_1$)
~~B1~~
~~$B2_2$~~
~~...~~
~~$B2_{1000}$~~
Eip: 0x8 ($B2_2$)
B1
$B2_3$
...

Large amount of work discarded at rollbacks

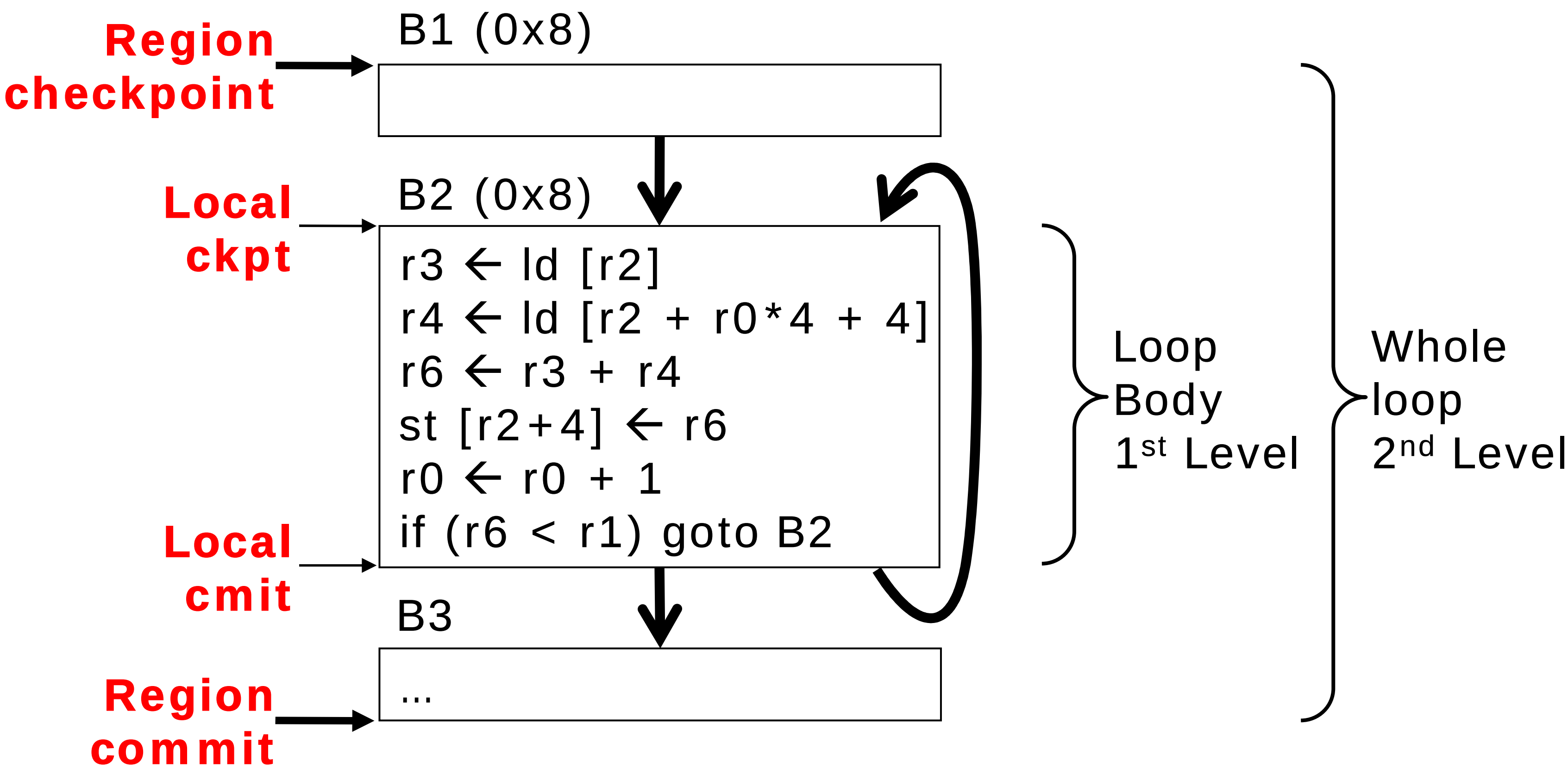
Small and Large Atomicity Scopes



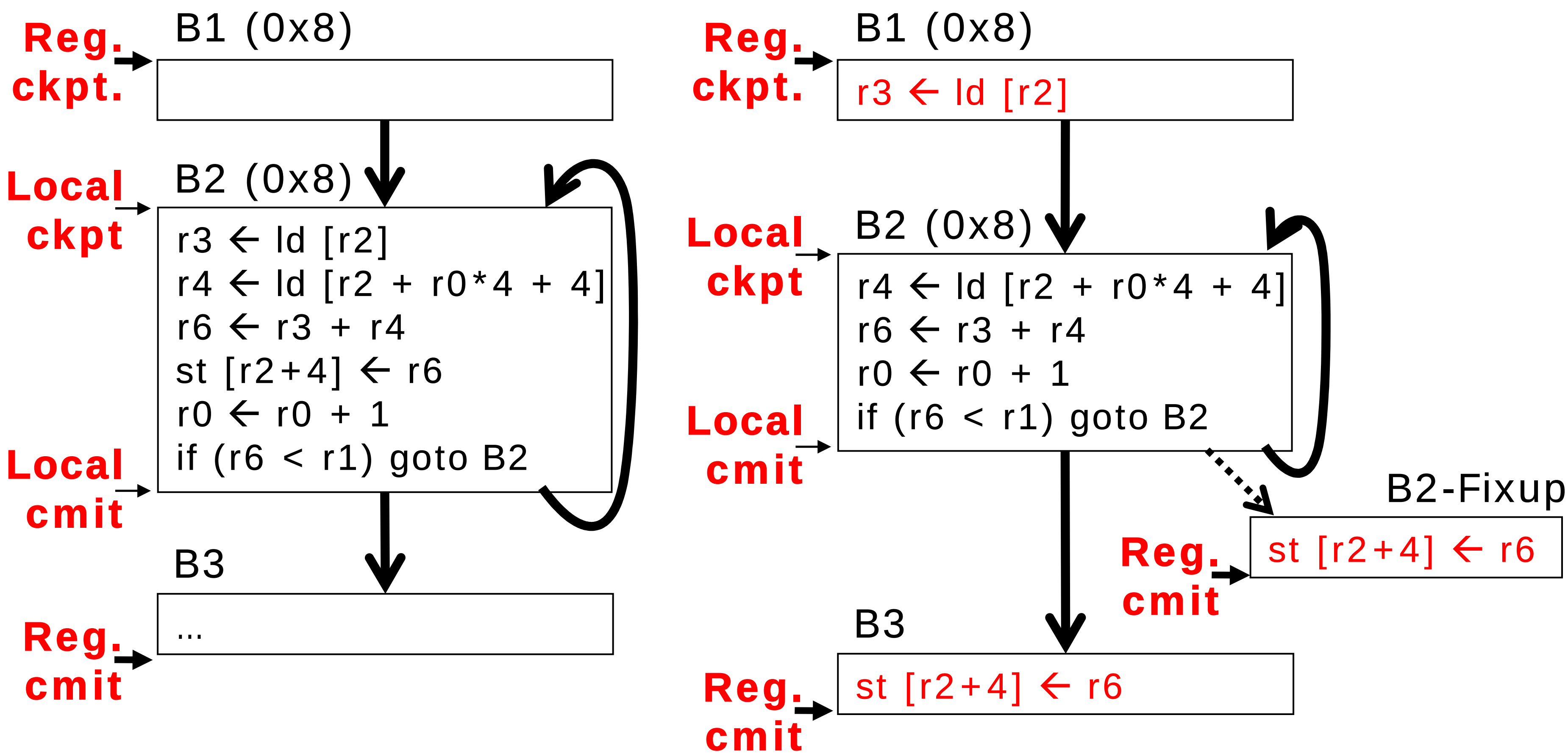
Small and Large Atomicity Scopes



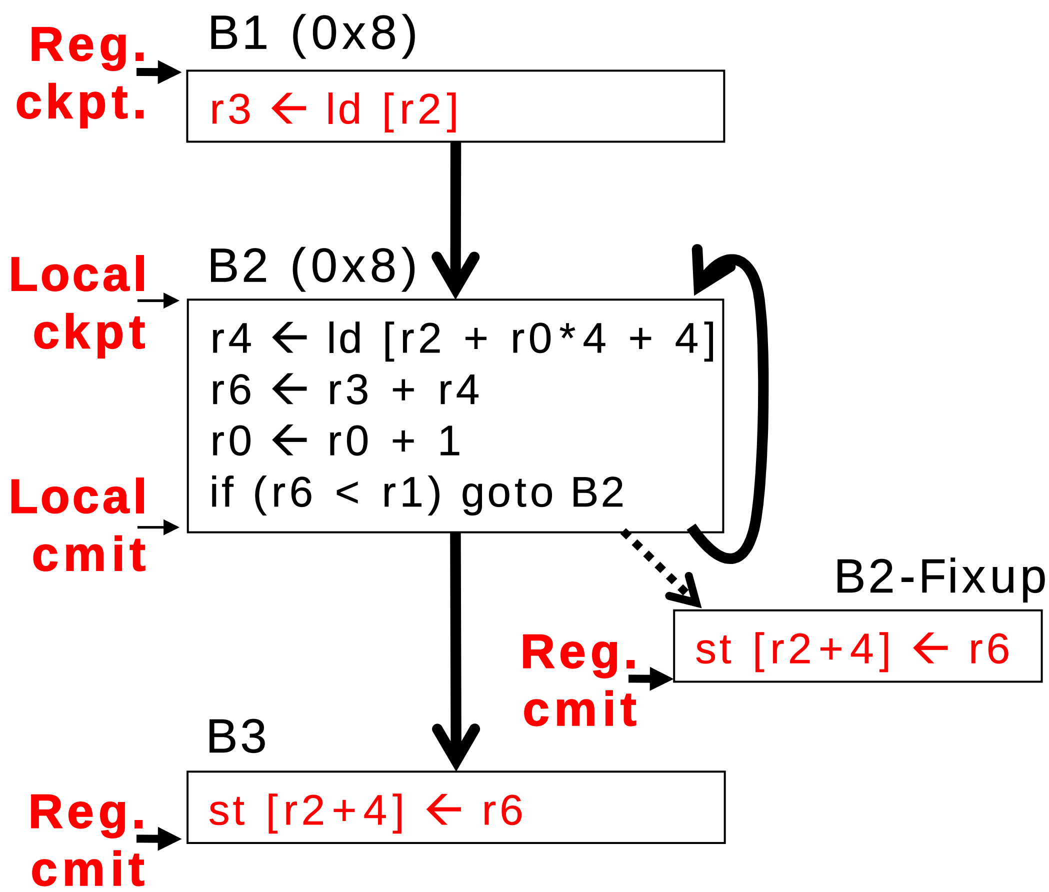
Two Level Atomicity



Two Level Atomicity

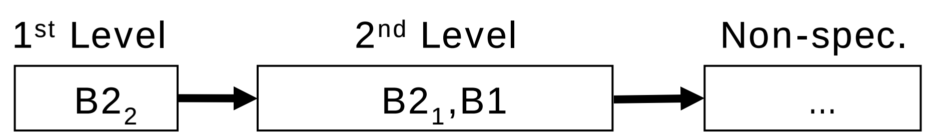


Two Level Atomicity

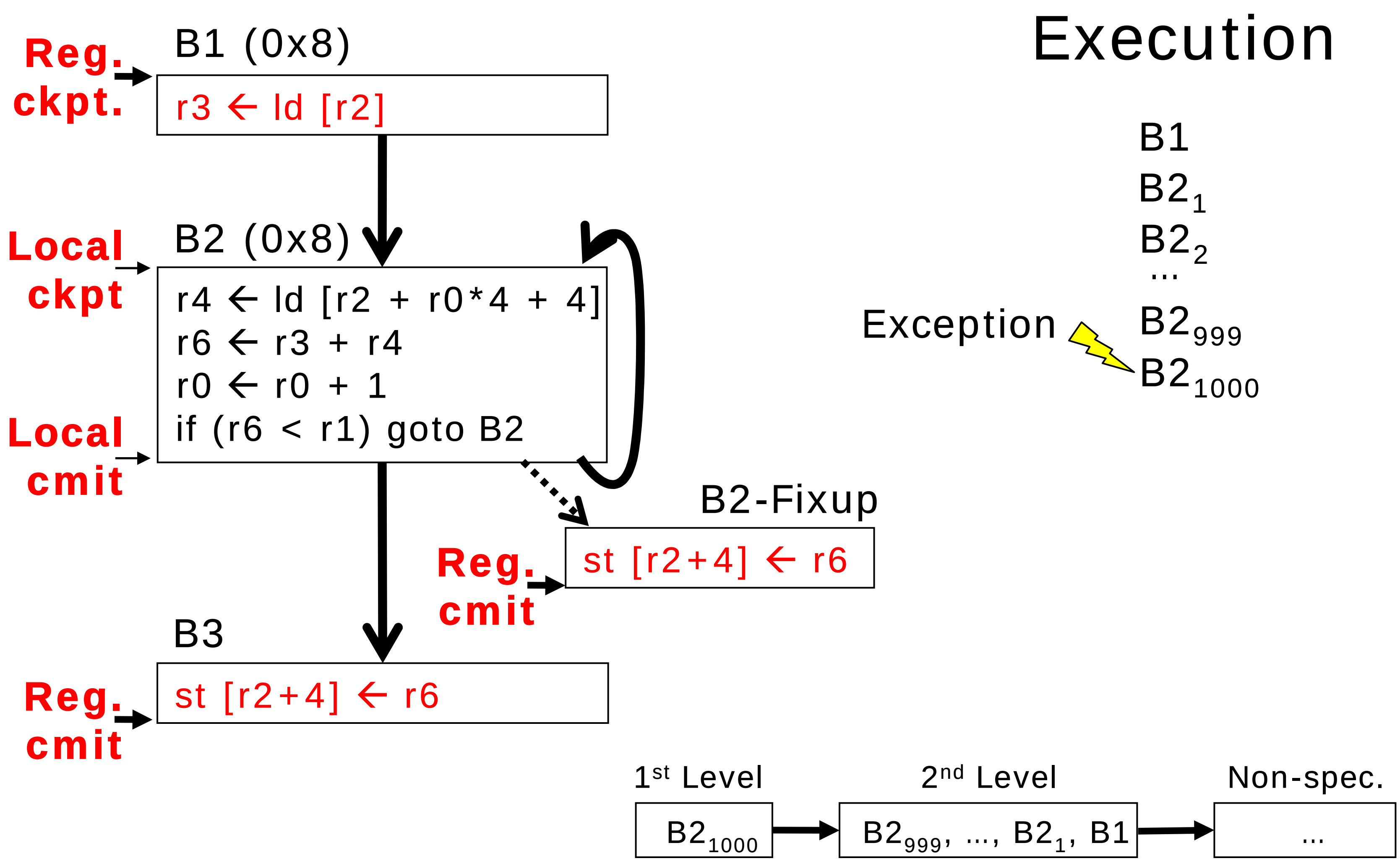


Execution

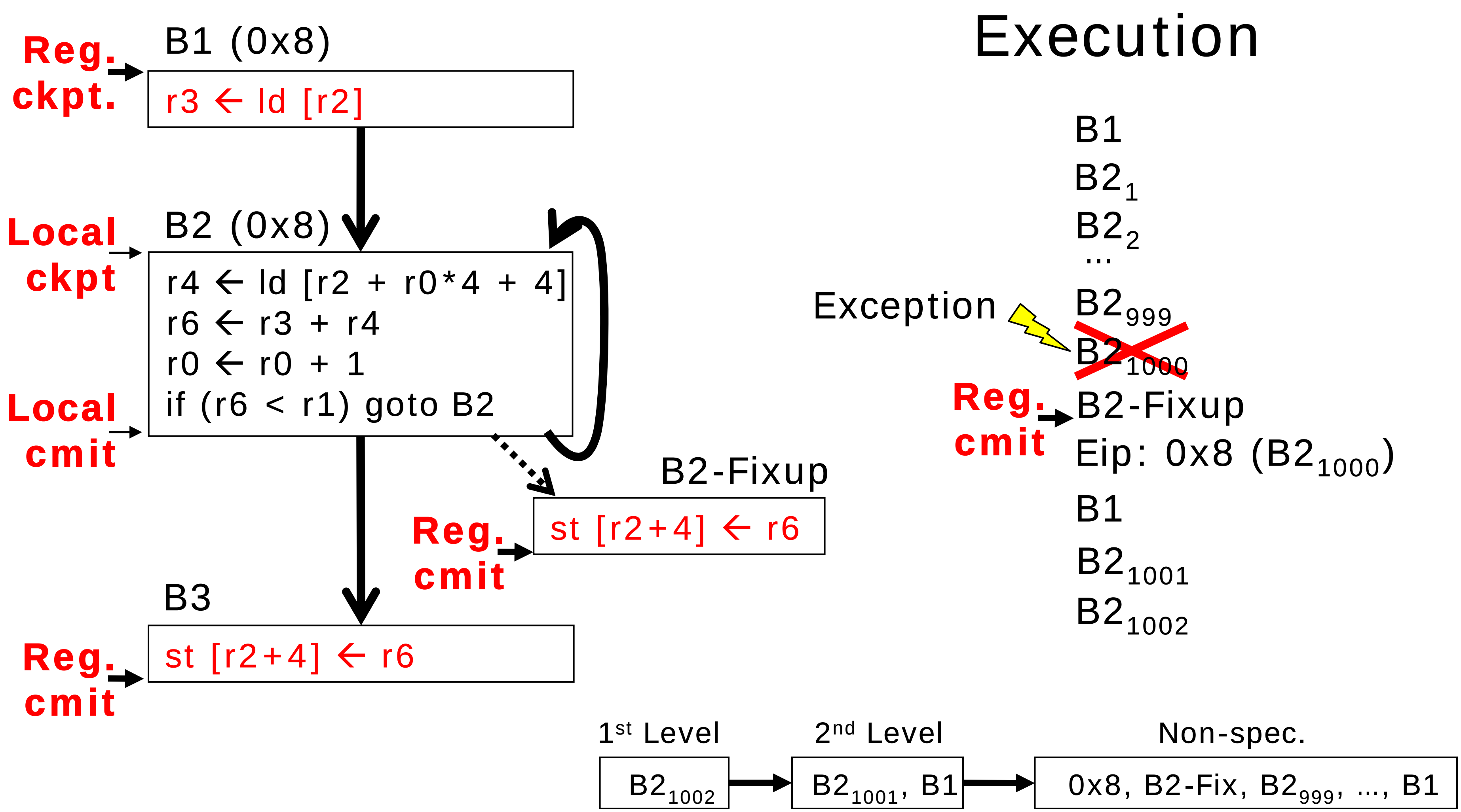
B1
$B2_1$
$B2_2$



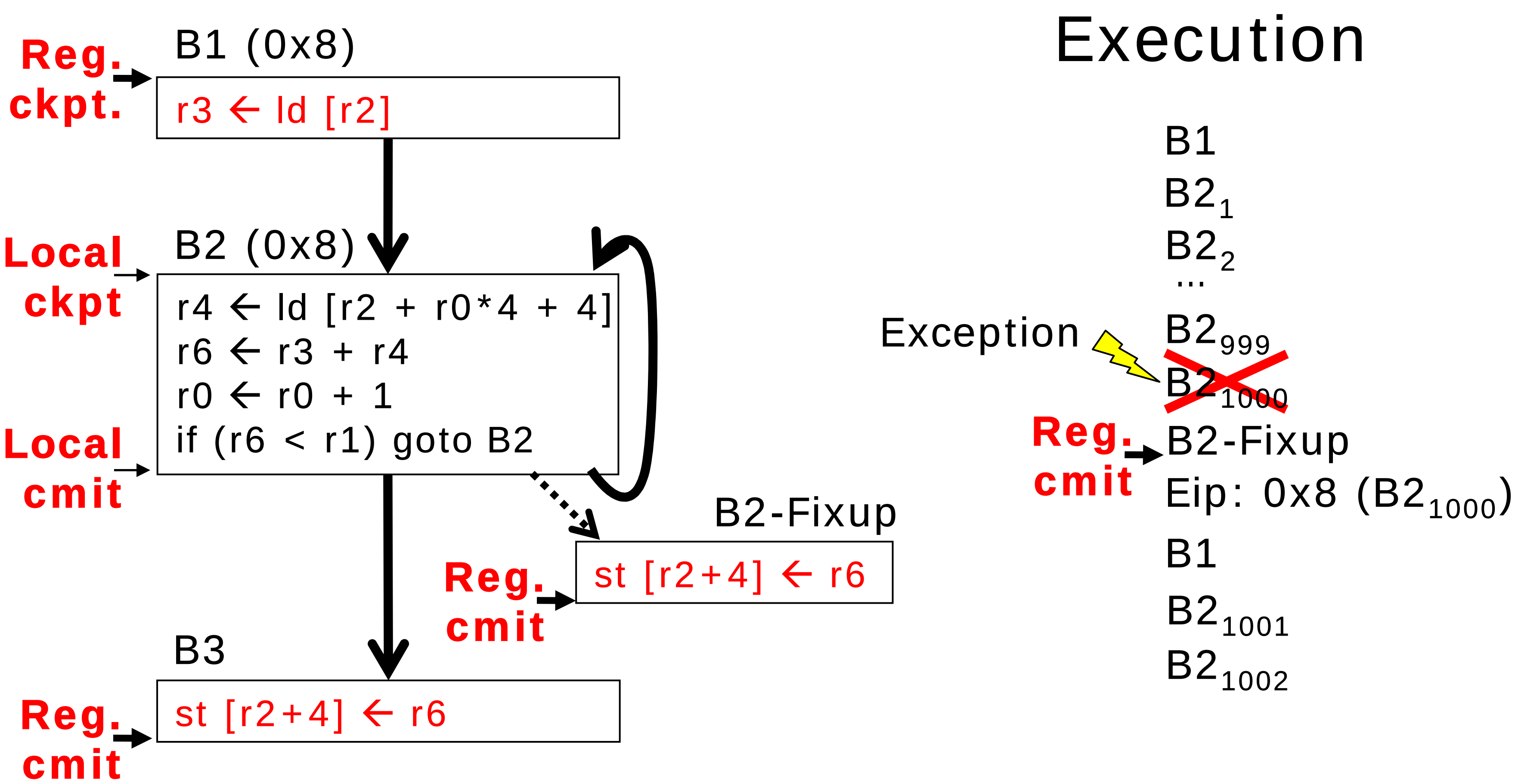
Two Level Atomicity



Two Level Atomicity



Two Level Atomicity

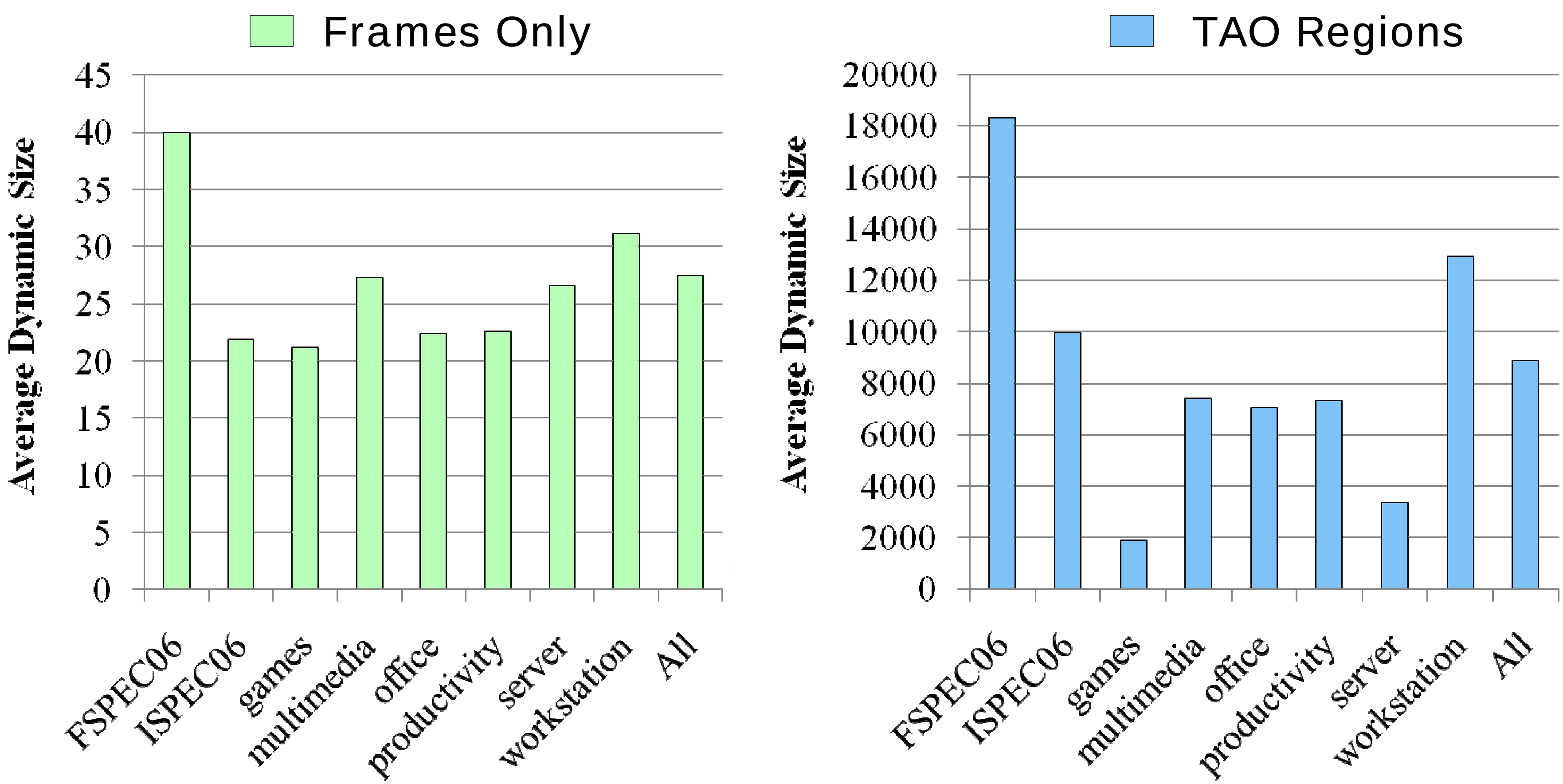


Amount of work discarded by rollback = 1 loop iterations

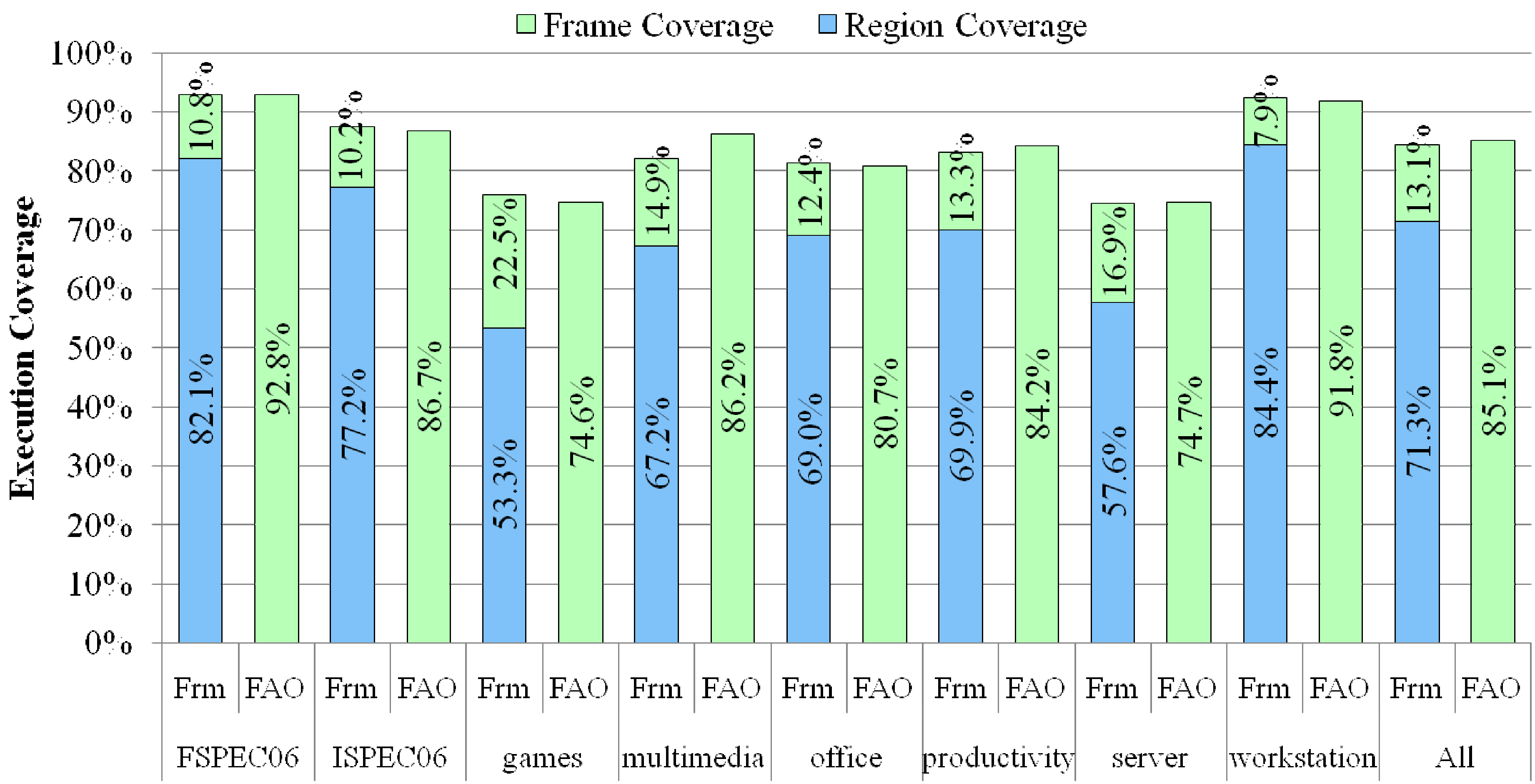
Preliminary Results

- Implemented TAO in a cycle accurate simulator
- 1st level atomicity is modeled with frame
- 2nd level atomicity is modeled assuming unlimited speculative cache
- Regions consist of frames
- Global partial redundancy elimination (PRE) and dead code elimination (PDE) are implemented
  - PDSE not measured due to simulator issue
  - Global optimization overhead is not measured, although frame level HW optimizations are modeled

Region and Frame Dynamic Sizes

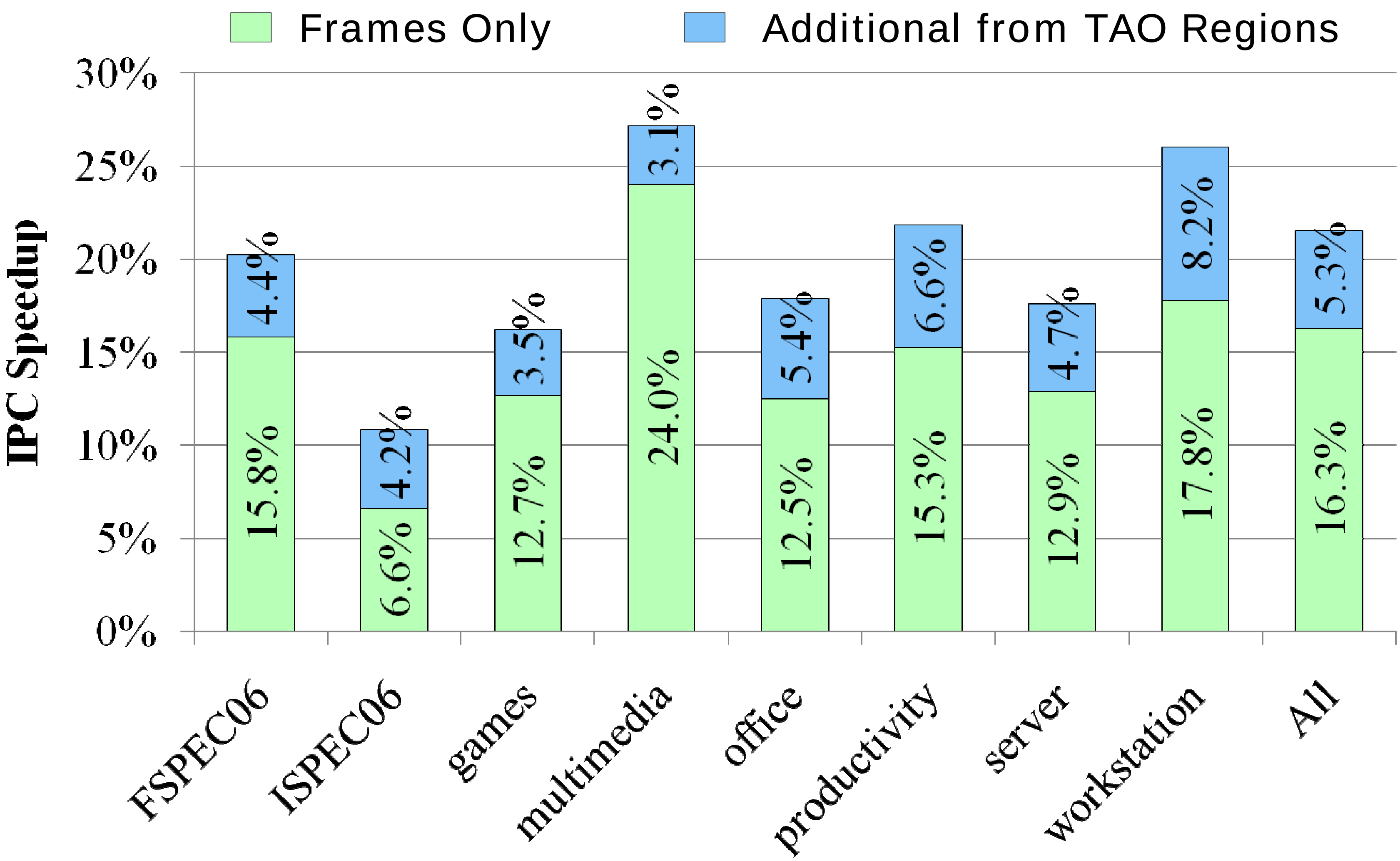


Region and Frame Coverage



Binary Compilation Technology

Performance Potential



Binary
Compilation
Technology

Conclusions & Future Work

- Two Level Atomicity enlarges the Optimization Scope with limited rollback costs
- Promising performance gains over frame level atomicity alone
- More optimizations can boost the performance gain
  - Global fusion, etc
- May improve in-order co-designed processor by enabling more global scheduling
- Hardware needs more investigation
  - Pipeline Buffers + Speculative Cache
  - 2-level speculative cache

Questions?