



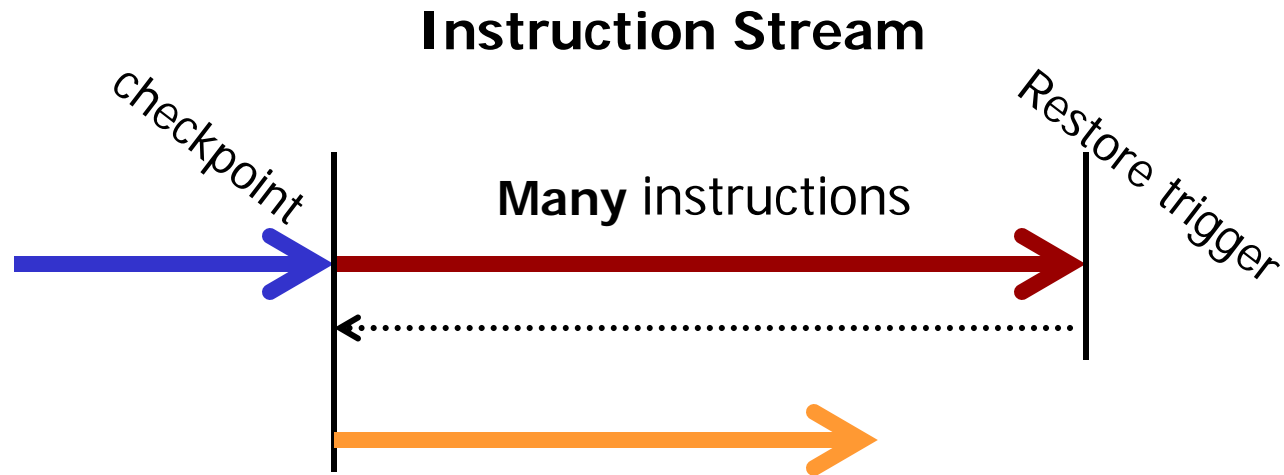
Memory State Compressors for Gigascale Checkpoint/Restore

www.eecg.toronto.edu/aenao

Andreas Moshovos
moshovos@eecg.toronto.edu



Gigascale Checkpoint/Restore



- Several Potential Uses:
 - Debugging
 - Runtime Checking
 - Reliability
 - Gigascale Speculation



Key Issues & This Study

- Track and Restore Memory State
- I/O?

- This Work: **Memory State Compression**
- Goals:
 - Minimize On-Chip Resources
 - Minimize Performance Impact

- Contributions:
 - Used Value Prediction to simplify compression hardware
 - Fast, Simple and Inexpensive
 - Benefits whether used alone or not

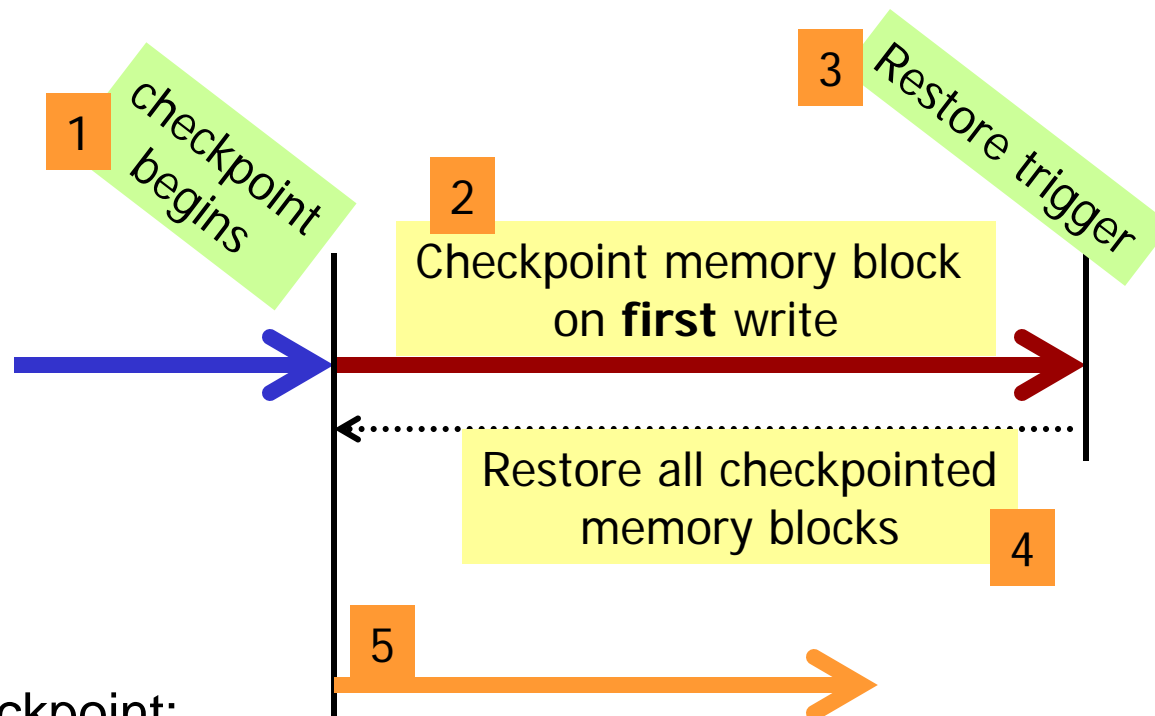


Outline

- Gigascale Checkpoint/Restore
- Compressor Architecture: Challenges
- Value-Prediction-Based Compressors
- Evaluation



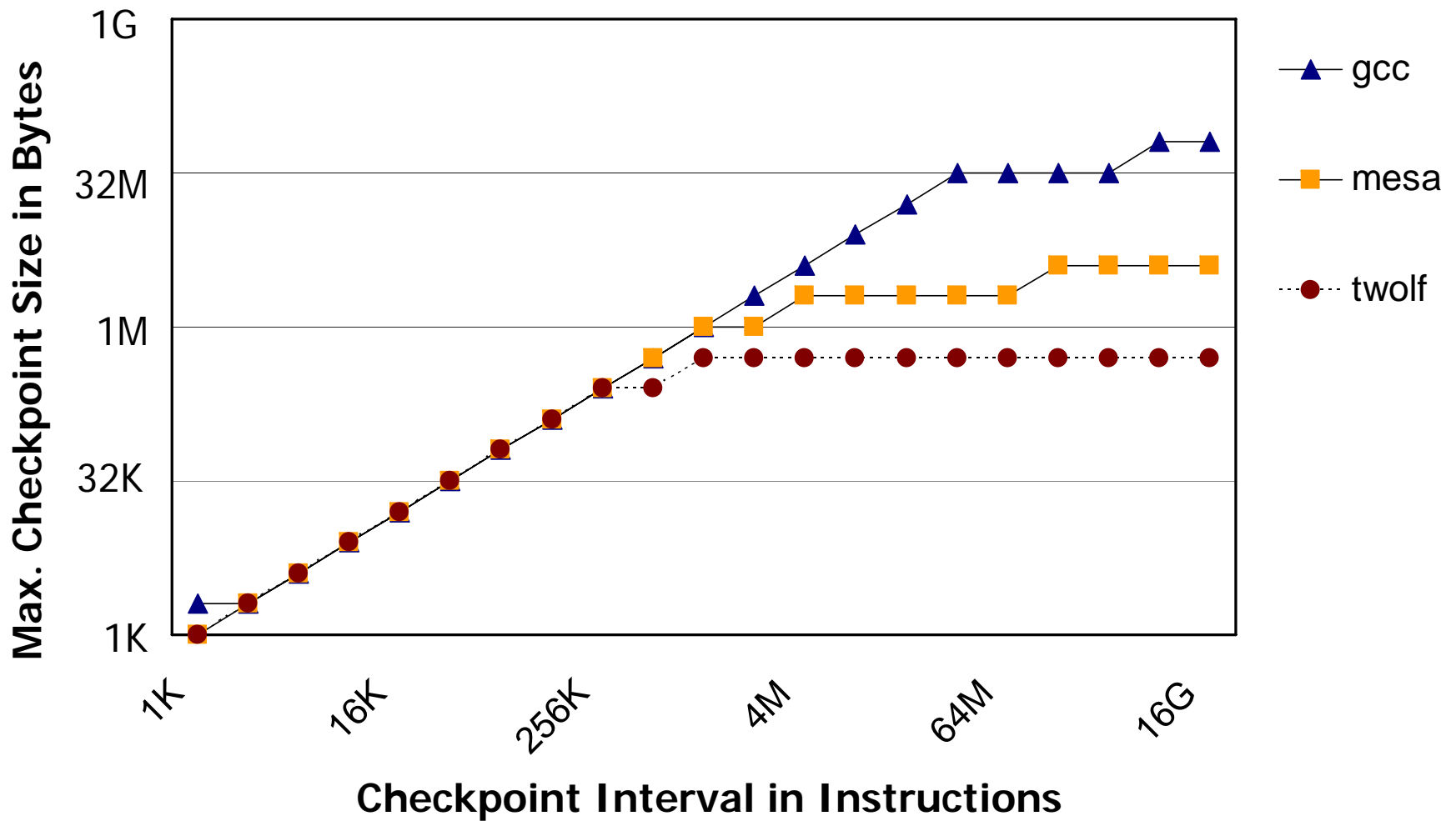
Our Approach to Gigascale CR (GCR)



- Checkpoint:
 - blocks that were written into
 - Current Memory State + Checkpoint = Previous Memory State
- Checkpoints: Can be large (Mbytes) and we may want many**

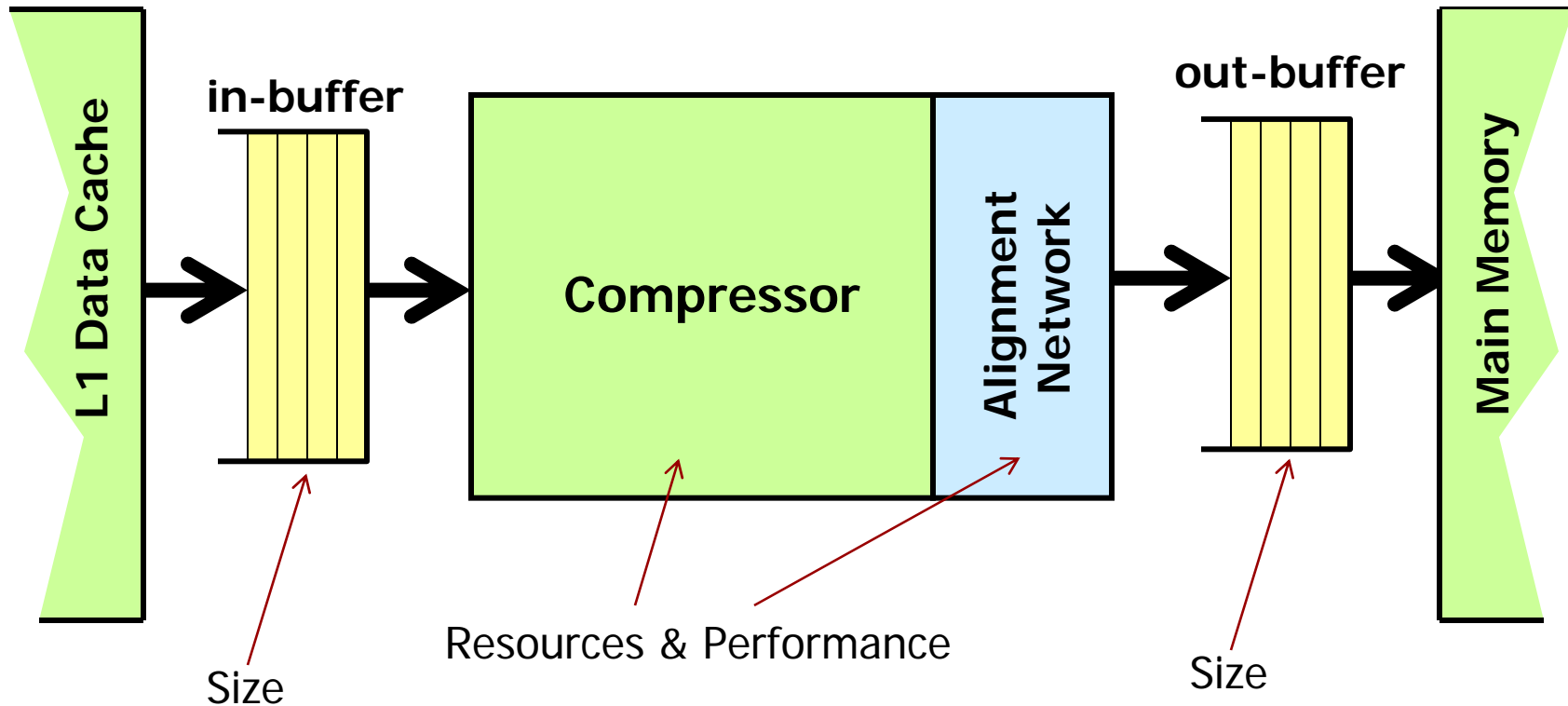


Checkpoint Storage Requirements





Architecture of a GCR Compressor



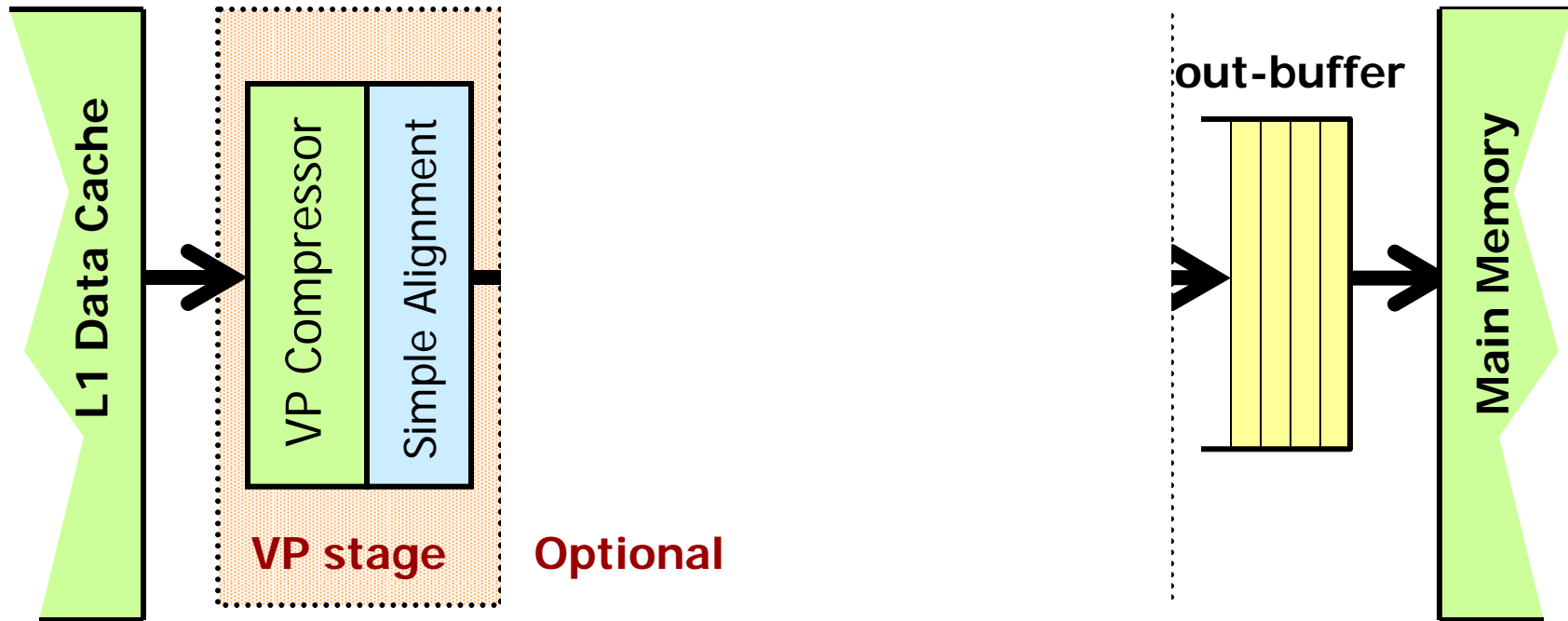
- **Previous work: Compressor = Dictionary-Based**
- **Relatively Slow, Complex Alignment, order 10K of Transistors**

▪ 64K In-Buffer → ~3.7% Avg. Slowdown

Moshovos ©



Our Compression Architecture



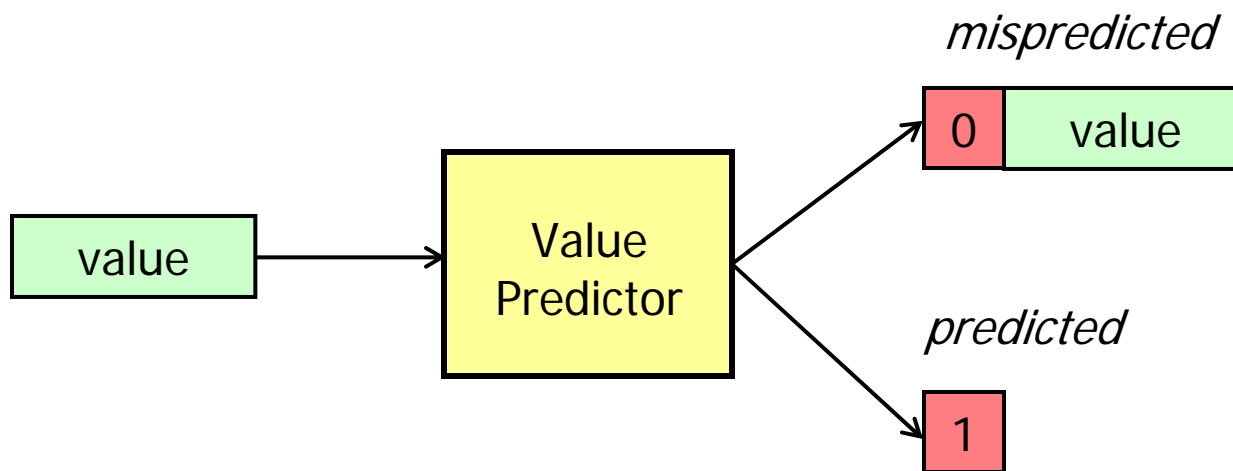
- Standalone:
 - ~Compression, - Resources
- In Combination:
 - -Resources (in-buffer), +Compression, +Performance



Value-Predictor-Based Compression

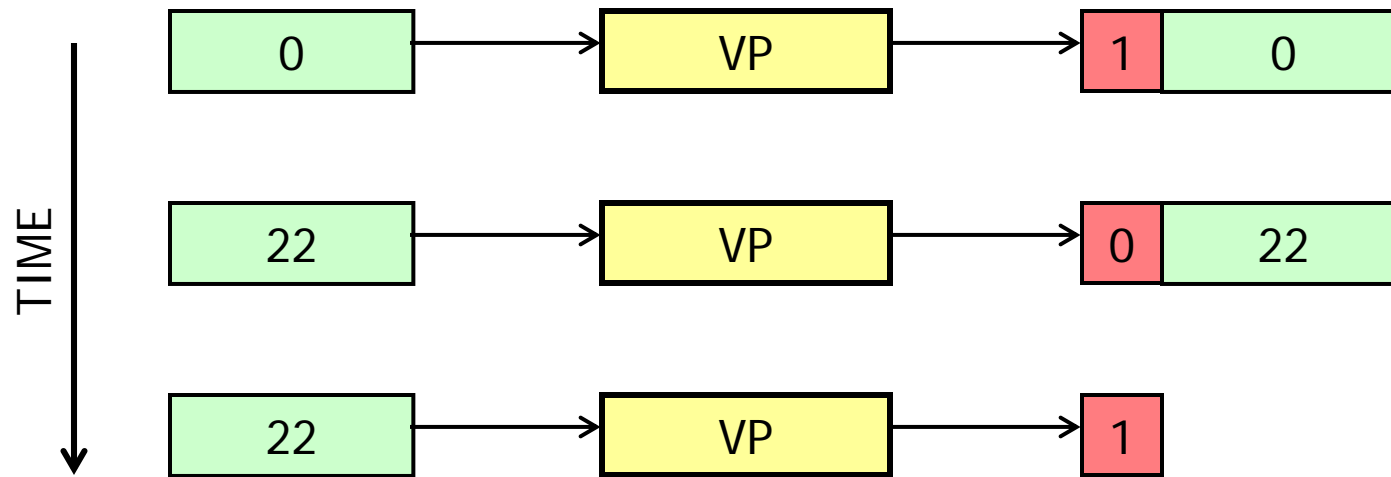
Input stream

Output stream



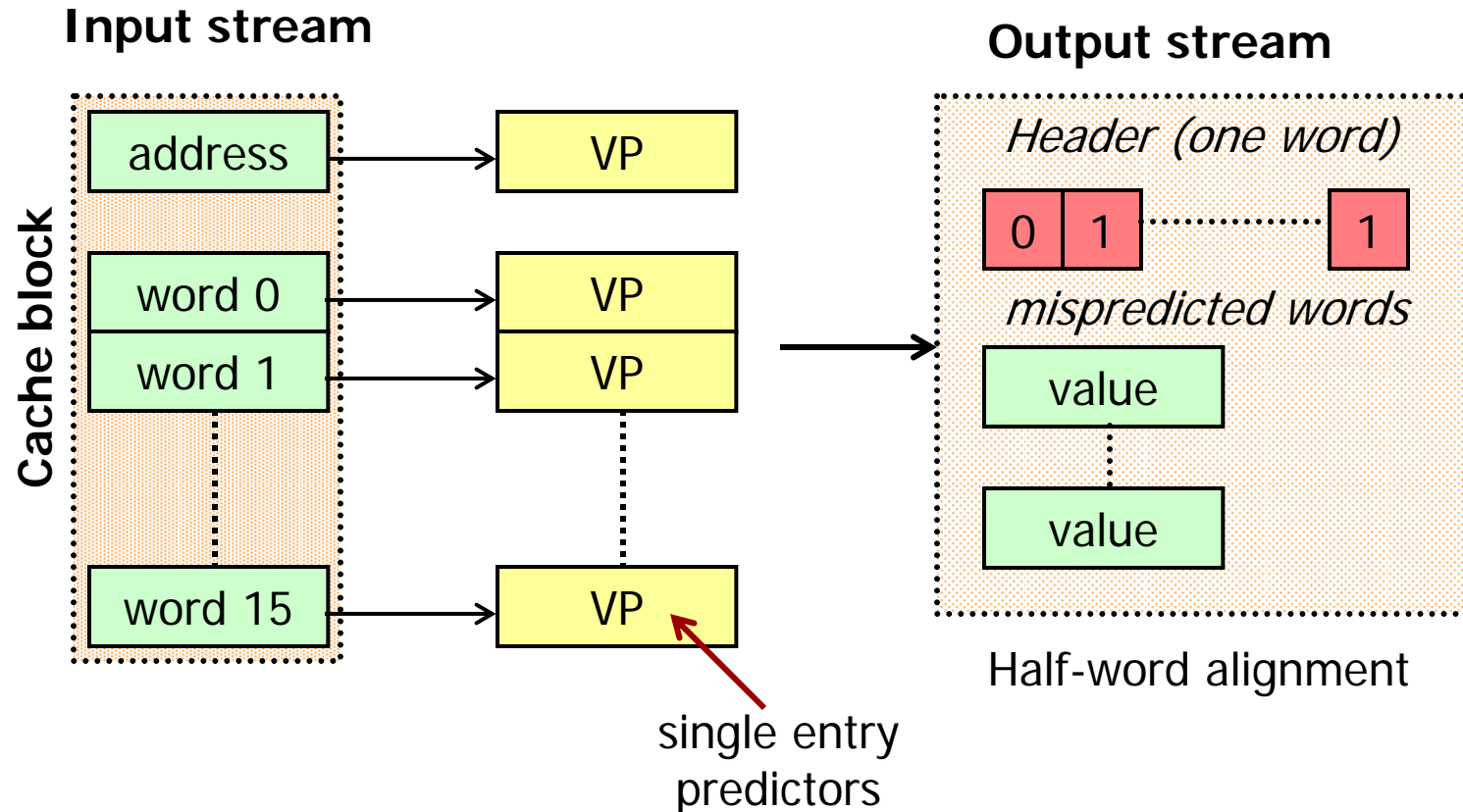


Example





Block VP-Based Compressor



- Shown is Last-Outcome Predictor
- Studied Others (four combinations per word)



Evaluation

- Compression Rates
 - Compared with LZW

- Performance
 - As a function of in-buffer size

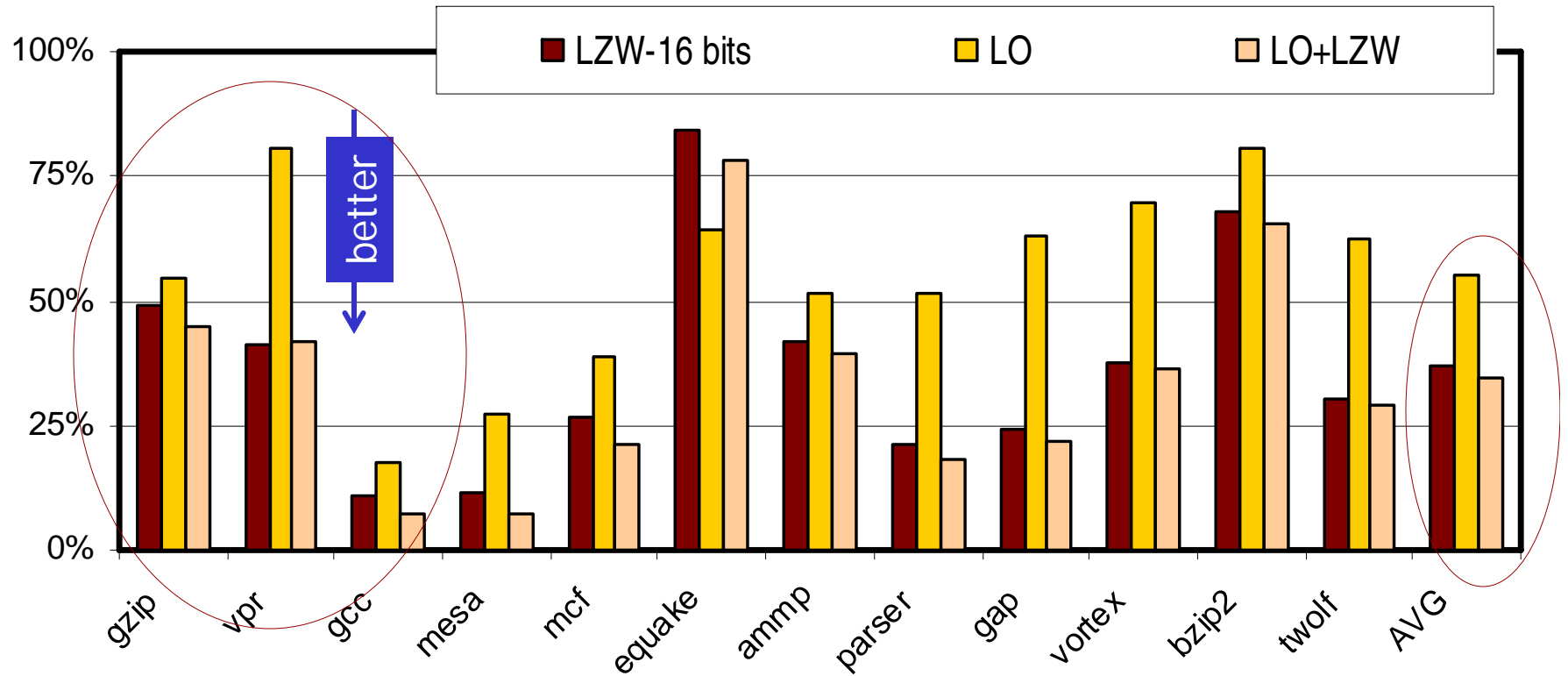


Methodology

- SimpleScalar v3
- SPEC CPU 2000 with reference inputs
- Ignore first checkpoint to avoid artificially skewing the results
- Simulated up to:
 - 80Billion instructions (compression rates)
 - 5Billion instructions (performance)
- 8-way OOO Superscalar
- 64K L1D, L1I, 1M UL2



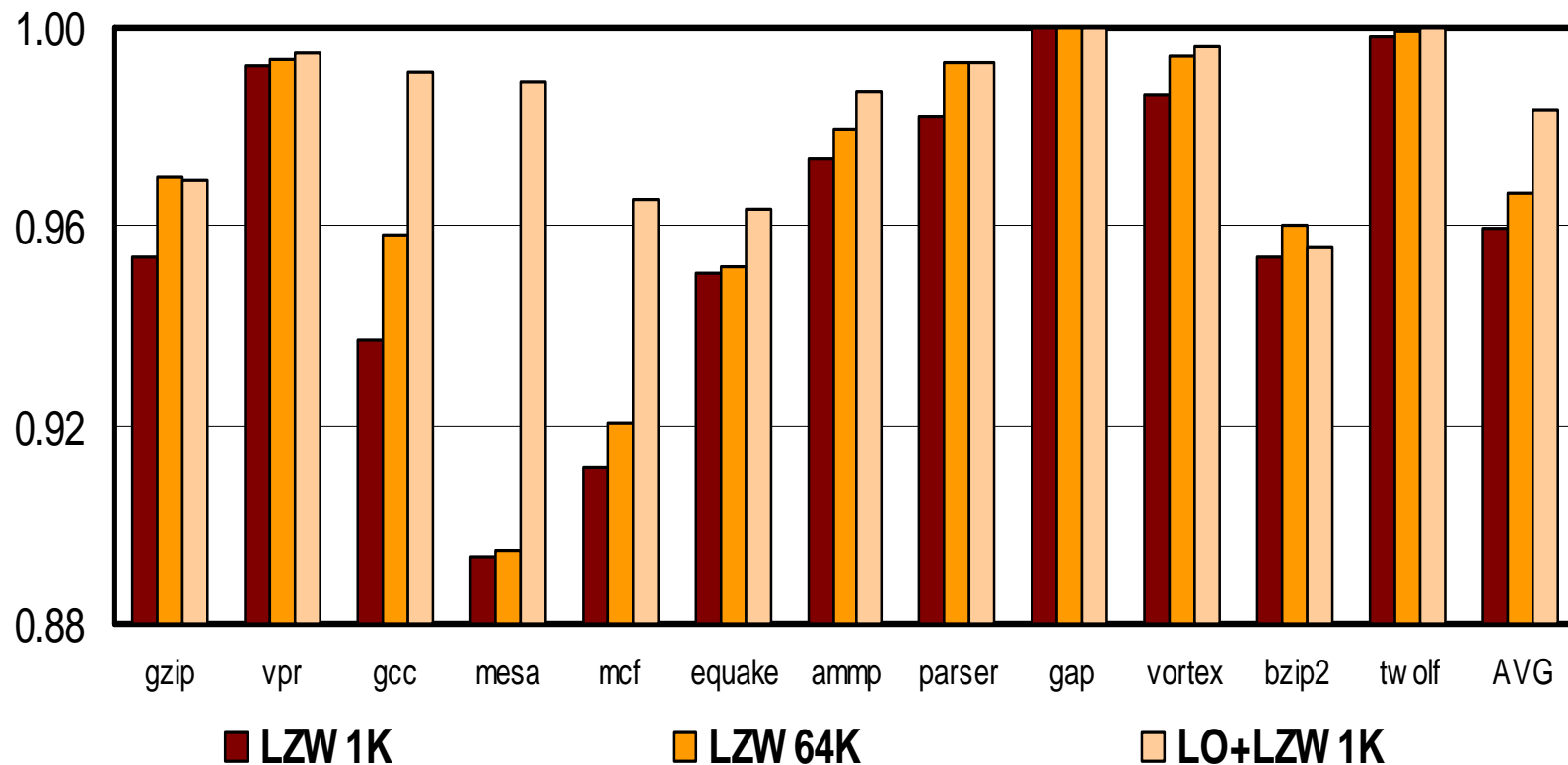
Compression Rate vs. LZW



256M Instructions Checkpoint Interval



Performance Degradation



- LZW + 64K buffer = ~3.7% slowdown
- LZW + LO + 1K buffer = 1.6% slowdown



Summary

- Memory State Compression for Gigascale CR
- Many Potential Applications
- Used Simple Value-Prediction Compressors
 - Few Resources
 - Low Complexity
 - Fast Performance
- Can be Used Alone
- Can be Combined with Dictionary-based Compressors
 - Reduced on-chip buffering
 - Better Performance
- Main memory compression?