Leader-Follower Processors

Smruti R. Sarangi

Department of Computer Science
Indian Institute of Technology
New Delhi, India

Outline

Leader Follower Architectures

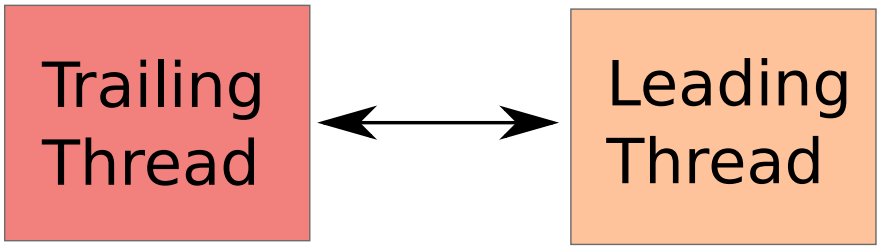


Figure 1: Leading-Trailing Configurations

Leader Follower Architectures

- The leading thread passes on memory values, branch prediction outcomes, and other hints to the trailing thread.
- The leading thread is partially correct.
- The trailing thread is guaranteed correct.
- The trailing thread is sped up by hints from the leading thread.

Leader Follower Architectures-II

What can leader follower architectures be used for:

- It can be used to increase the performance of the system
  - Slipstream processors
  - Future execution
  - Dual core execution
- It can be used to increase reliability
  - DIVA
  - Redundant Multi-threading
  - Paceline

Outline

Basic Idea

- The leader is called the *A-Stream*
- The trailing thread is called the *R-Stream*

Mechanism

- Hardware analyzes the R-stream to find sequences of instructions that are predictable.
- These instruction sequences are replaced with simpler substitutes in the A-Stream.
- The A-stream prefetches and passes hints to the R-Stream.
- Periodically, the R-stream repairs the architectural state of the A-Stream.

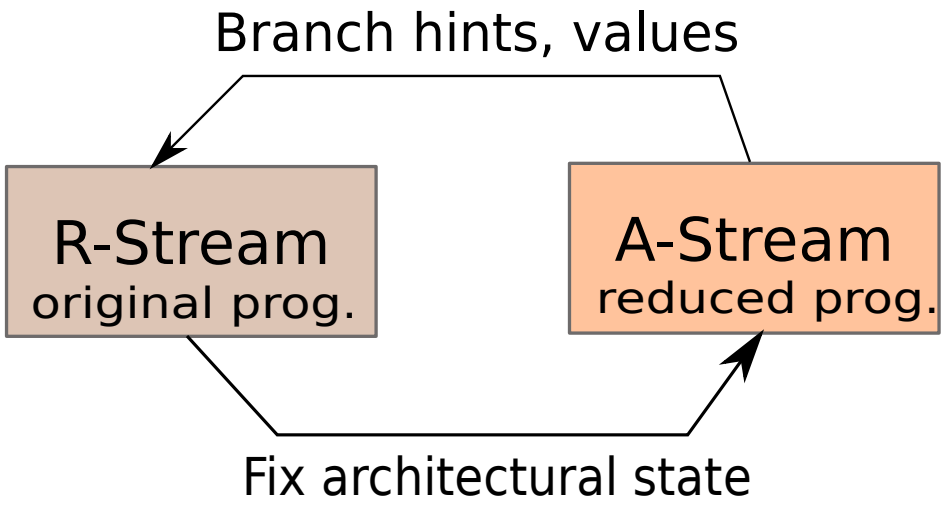


Figure 2: SlipStream Threads

- The A-Stream reduces the number of fetched and executed instructions.

Reducing Instructions in the A-Stream

Reducing Fetched Instructions

- Find all the branches that are very predictable.
- Remove all the instructions dependent on these branches.

Removing Executed Instructions

- Find all the data values that are predictable
- Dynamically remove all those instructions that generate these values and are also dependent on these values.

Main Components

- IR (Instruction Removal) Predictor
  - It predicts the high-confidence and low-confidence branches.
  - If a branch is high confidence, then it removes it and all its dependent instructions from the fetch stream.
- IR Detector
  - It looks at the instruction sequences in the R-stream and finds instructions that could have been removed by the program.
  - Example: Redundant stores, highly predictable branches
  - It conveys this information to the IR Predictor.

Main Components - II

- Delay Buffer : It is used to communicate values between the R-Stream and A-Stream.
- Recovery Controller
  - It monitors the memory accesses of the A and R-streams.
  - It detects erroneous execution in the A-Stream by detecting corrupted memory values.
  - It initiates recovery by copying the memory context from the R-Stream to the A-Stream.

Outline

Paceline Architecture

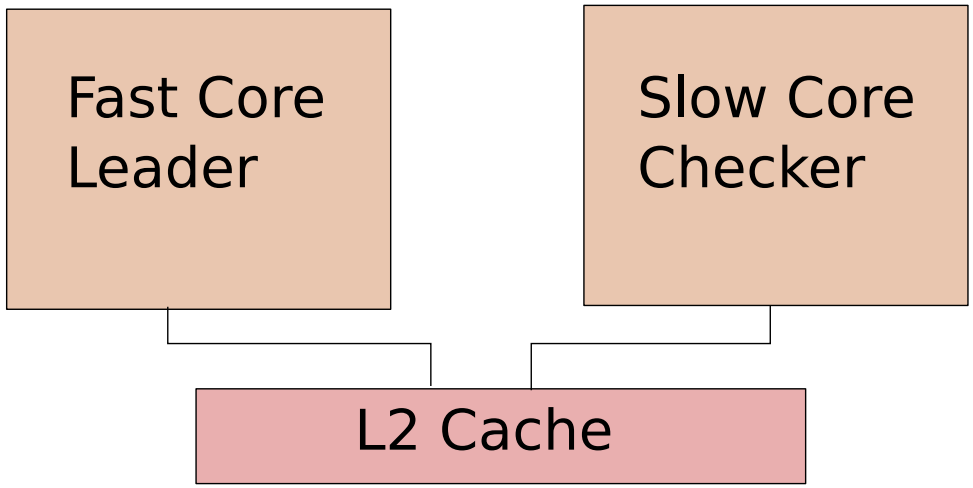
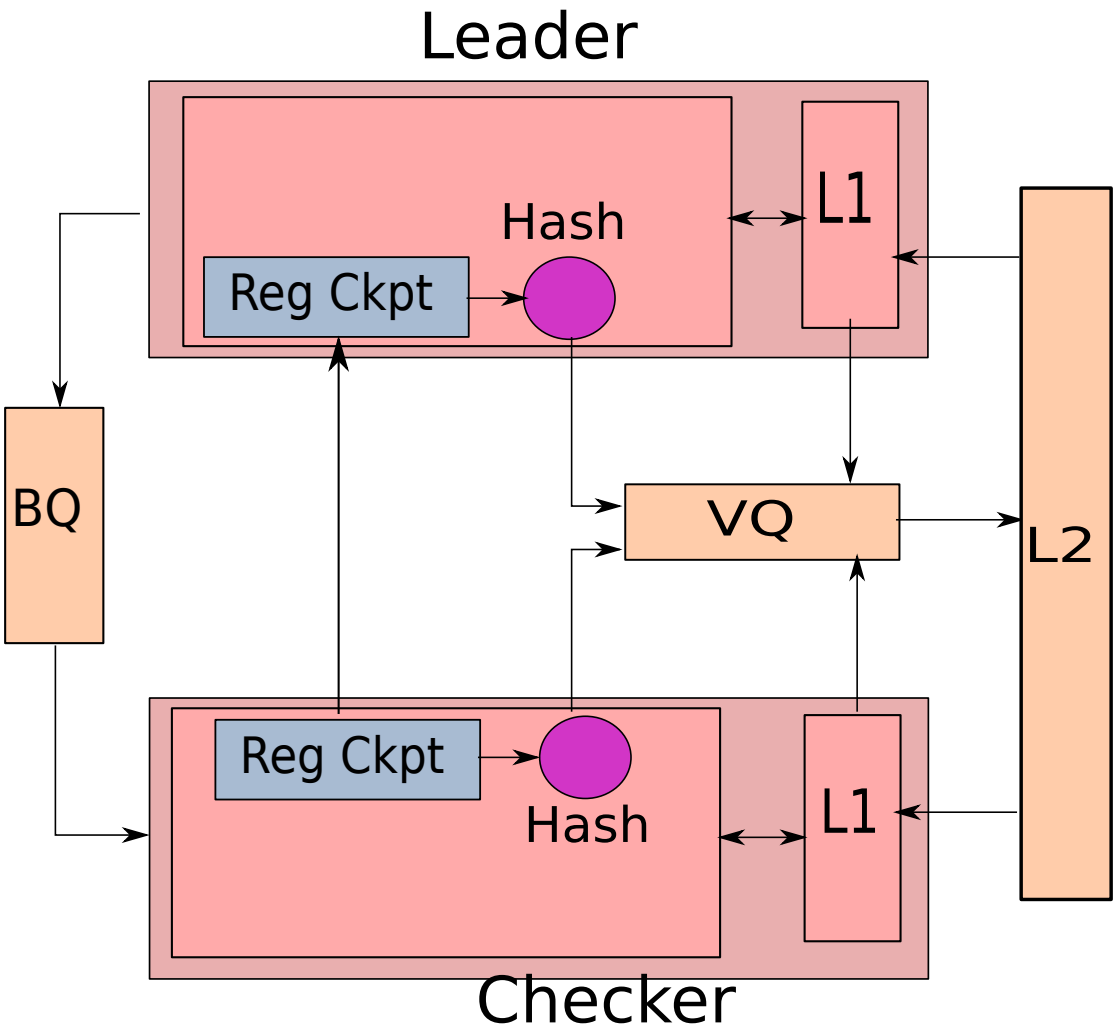


Figure 3: Paceline conceptual diagram

Leader-Checker

- The leader core runs at an unsafe frequency
- The checker core runs at a safe frequency
- Periodically, the leader and checker swap. This ensures that there is a uniform temperature distribution in the cores.
- Important Structures
  - BQ $\rightarrow$ Queue of branch outcomes
  - Reg Hash $\rightarrow$ Hash of a register checkpoint
  - VQ $\rightarrow$ Stores all the checkpoints

Detailed Architecture



Branch Queue & Validation Queue

Branch Queue

This queue contains the outcome of the leader's branches. The checker uses these hints to get a better branch prediction accuracy.

Validation Queue

- Both the processors takes a register checkpoint and hash it every *n* instructions.
- The VQ saves the checkpoint and compares them.
- The VQ stores the L1 writes.
- After it receives the write from both processors, it sends it to the L2 cache if the stored values are equal.
- Otherwise, the VQ flags an error.

Design Choices

Simple

- Assume that the checker is guaranteed correct.
- When there is an error cleanup the state of the leader.

Highly Reliable

- Assume that both the leader and the checker can have errors.
- Upon an error, clean up the state (reg. file + L1 cache) of both the processors
- Repopulate the register file from the last checkpoint stored in the VQ.


Improving Single-Thread Performance in Nanoscale CMPs through Core Overclocking, Brian Greskamp, and Josep Torrellas, PACT, 2007
http://iacoma.cs.uiuc.edu/iacoma-papers/pact07.pdf


A Study of Slipstream Processors, Zach Purser, Karthik Sundaramoorthy, and Eric Rotenberg. MICRO 2000.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.24.2465&rep=rep1&type=pdf


Aerodynamics of drafting
http://en.wikipedia.org/wiki/Drafting_(aerodynamics)