ACACES 2009 - Fifth International Summer School, July 12 to July 18, 2009, Terrassa (Barcelona), Spain

Exploring some multicore research opportunities. A first attempt.

LBUS
Lucian Blaga University of Sibiu

Ciprian Radu, Horia Calborean, Adrian Florea , Árpád Gellért, and Lucian Vinţan

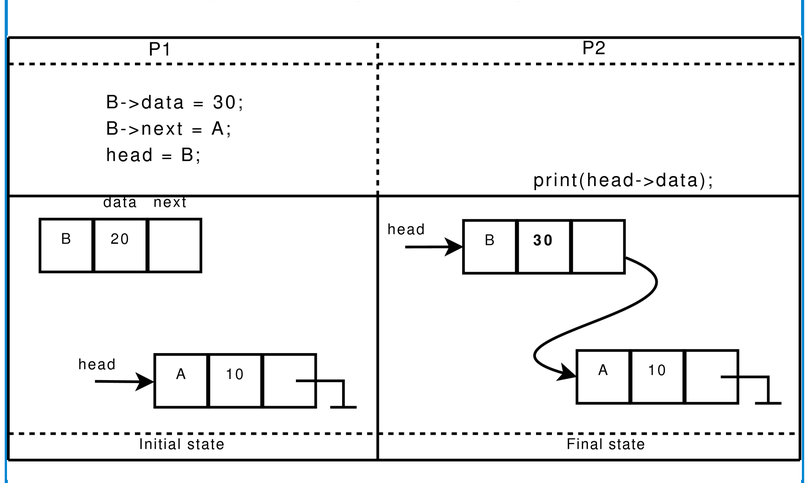
ACAPS (http://acaps.ulbsibiu.ro/research.php)
Lucian Blaga University of Sibiu, Romania

Introduction

- Multicores should still consider the benefits provided by ILP exploitation mechanisms.
- Multicore research should consider the synergistic exploitation of different kinds of parallelism (pipeline, ILP, TLP, tasks, data, etc.).
- In [GFV09] we have developed:
  - a mechanism of selective value anticipation dedicated to high latency instructions, which includes a reuse scheme for MULs and DIVs;
  - a value predictor for critical LOADs (miss in L2-cache).
- We consider integrating techniques like Value Prediction and Dynamic Instruction Reuse into Chip Multiprocessor architectures.
- We present the need for evaluating multicore architectures by Automatic Design Space Exploration.

Value Prediction in Multicores

- In a multicore architecture a simple implementation of value prediction is incorrect because it can violate the memory consistency model [MSC+01].



- Two detection mechanisms for memory consistency violation are proposed:
  - based on addresses - a processor must detect when another processor writes to an address that was speculatively read;
  - based on values - each speculative load will wait until its operands become non-speculative and then the load will be performed once again.

Dynamic Instruction Reuse in Multicores

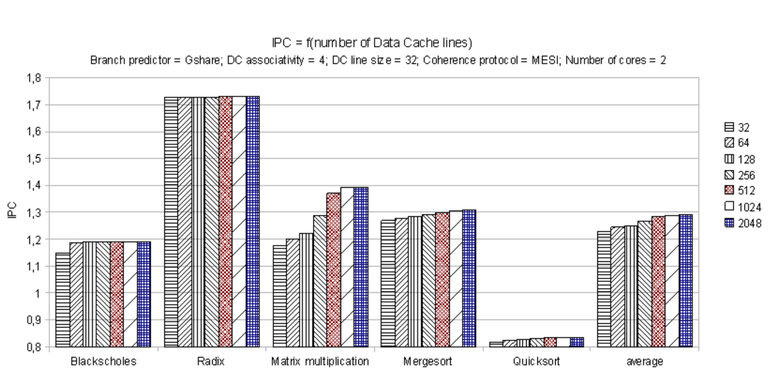
- Consider a multicore architecture with DIR used in each core. The following program is executed, considering a shared memory programming model.

```
// let k be a strictly positive integer number and let V be a shared variable
int i = 0;
while (i < k) {
      bool new_value_produced = false;
      if (pthread_self() == 0) {
            if (i > 0) {
                  // wait until the rest of the threads consume the previous value
                  while (!previous_value_consumed);
            }
            V = i; // thread 0 produces a new value for each loop iteration
            new_value_produced = true;
      } else {
            // wait until thread 0 produces the new value
            while (!new_value_produced);
            printf("%d\n", V); // consume the new value
            // this thread will have to notify that it consumed the value...
      }
      i++;
}
```

- The invalidation mechanism for the Reuse Buffer must be implemented globally because both local and remote stores have to be considered.
- The solution would be to benefit from the invalidation messages sent at the level of the processor's cache.
- Cache coherence can help at keeping the Reuse Buffer data correct.

Automatic Design Space Exploration

- Below it is presented a simulation result from our previous work with UNISIM: the evaluation of the evolution of the global IPC of a dual-core Symmetric Multiprocessor.



- Besides the parameters specific to a single processor architecture, we have to account the ones that are multicore specific.
- Simulating multicore architectures proves to be extremely time consuming.
- ADSE – developing optimized heuristic research methods for the huge space of the parameters of applications, compiler and architectures.

Conclusions

- We believe that the already developed mechanisms for exploiting Instruction Level Parallelism must be used in multicore architectures.
- Value Prediction and Dynamic Instruction Reuse can help in increasing the performance over the sequential parts of the parallel programs but, they have to be adapted to the more general context provided by parallel programming.
- Developing a multi-criteria ADSE would help in optimizing the process of determining the best architectural configuration.

References

[GFV09] Arpad Gellert, Adrian Florea, and Lucian Vintan. Exploiting selective instruction reuse and value prediction in a superscalar architecture. Journal of Systems Architecture, 55(3):188–195, 2009.

[MSC+01] Milo M. K. Martin, Daniel J. Sorin, Harold W. Cain, Mark D. Hill, and Mikko H. Lipasti. Correctly implementing value prediction in microprocessors that support multithreading or multiprocessing. In MICRO 34: Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture, pages 328–337, Washington, DC, USA, 2001. IEEE Computer Society.

ciprian.radu@ulbsibiu.ro http://webspace.ulbsibiu.ro/ciprian.radu/