

Paper #3

Author: Michael Chu et al.

Title: Region-based Hierarchical Operation Partitioning for Multicore Processors

Reviewer: Ilie Savga

Summary:

While the clustered architectures improve the performance of superscalar and VLIW processors by increasing instruction-level parallelism and eliminating the bottleneck of centralized register files, they may suffer from the inefficient partitioning of operations across the available resources. The scope of the paper is to introduce an approach to alleviate this problem by introducing a technique for clustering operations. The two key issues of the presented non-scheduler-centric algorithm are: a) multilevel graph partitioning and slack distribution to perform efficient operation partitioning at a global, region-based scope b) newly introduced region-based hierarchical operation partitioning (RHOP) to determine the quality of a partitioning. Step-by-step examples and the results of experimental evaluation support the content.

The main contributions

The paper presents a technique that aims for efficient resource partitioning in multicore architectures. It introduces new methods of assigning weights to nodes (to reflect their intracluster usage) and edges (to estimate the cost intercluster moves) in the dataflow graph. This, in turn, allows for intelligent division of dataflow graph using a multilevel graph-partitioning algorithm. The experimental evaluation results show the improvement of clustering over the classic Bottom-Up Greedy algorithm, which becomes drastic in cases of rich-resource multicore homogeneous machines.

Merits and weaknesses

The paper is well written and organized. It contains a solid background of clustering that is supported by references to previous work. The thorough description of the presented algorithm is guided by comprehensible examples. The fair analysis of results reveals advantages and disadvantages of the methodology and discusses its possible improvements.

The only weakness is that the example presented in the description partitioning phase is a bit too simplistic and does not demonstrate some possible implications of the algorithm (see comments).

Numerical rating

Significance	8	
Originality	5	
Interest to a journal on programming languages and compiler technology		2
Quality of experimental evaluation	7	
Overall organization	9	
Presentation (style and language)	10	
Length appropriate	10	
References appropriate	10	
Overall evaluation	9	
Recommendation	Weak accept	
Confidence	3	

Comments and suggestions for improvement:

Your well-organized paper describes an interesting technique described in clear technical language. The possible minor improvements are:

- In section 3.2.2 when calculating the cluster weight, you do not consider the total node weight, as it has no effect on the result (for the chosen example). It is clear, of course, that this has been done for the simplicity of the example, but, still, the efficiency of the algorithm in the cases of high parallelism and necessary even distribution of processes among the clusters remains unsupported. It would be nice if you introduce another example or, perhaps, extend the existing one, to show the effect of total node weight in calculation of cluster weight;

- In the conclusion, you promise to improve some heuristic that decreases the performance of the presented algorithm. Could you mention (in general words) which possible changes could be applied to the existing technique?