

Design and Analysis of Parallel N-Queens on Reconfigurable Hardware with Handel-C and MPI

Vikas Aggarwal, Ian A. Troxel¹, and Alan D. George
High-performance Computing and Simulation (HCS) Research Laboratory
Department of Electrical and Computer Engineering, University of Florida
Gainesville, FL 32611-6200

N-Queens is a classical problem that for many years has been popular for the benchmarking of processing, memory, and communications architectures on high-performance computing systems. The problem is combinatorially hard and involves placing N queens on an $N \times N$ chessboard so that no queen can attack any other. The N-Queens problem is well suited for the development and benchmarking of search algorithms and the architectures on which to map them. Traditional methods to solve this problem use backtracking, whereby a queen is iteratively placed at a safe location in each column until no safe location exists and then adjustments are made to the positions of previously placed queens until a full solution for the board is found. The backtracking approach has exponential computational complexity, in that the complexity of the algorithm and the number of solutions grow exponentially with the size of the problem. The problem can be partitioned for parallel processing in several fashions, such as embarrassingly parallel methods for finding separate and distinct solutions for a given board size, as well as domain-decomposition methods where a single solution is processed in parallel by cooperating resources each dedicated to components of the data structure (e.g. columns), exchanging data and synchronizing with one another on queen placement. Although this benchmark has been primarily used for the study of search algorithms and the analysis of conventional computer system architectures, it offers interesting challenges for implementation and benchmarking on reconfigurable computing systems featuring programmable logic devices such as FPGAs.

One of the key challenges in the benchmarking and exploitation of new and emerging forms of reconfigurable computing systems is the hardware design strategy. Solutions are often created in a custom fashion using hardware description languages and other tools from integrated circuit design. However, this design flow can be cumbersome for application developers attempting to solve problems that require high-performance computing. The structure of the underlying algorithm can be significantly different when posed for a reconfigurable system versus a conventional one, and the porting and adaptation of code from high-level language implementations (e.g. in C or Fortran) to hardware designs can be extremely challenging. In order to realize the performance gains these new systems offer, algorithms designed for general-purpose processor architectures must typically undergo a redesign with the new programming paradigm in mind. Several tools are available to help ease the burden of this migration from processor code to reconfigurable hardware logic, including tools based on extensions to traditional programming languages such as Handel-C and Streams-C. However, these tools are still relatively new and thus much room remains for assessing their strengths and weaknesses in mapping key algorithms to various target architectures.

This presentation showcases the development of a parallel backtracking approach to the N-Queens problem designed using the Handel-C tool from Celoxica with emphasis on design strategy, performance analysis, tradeoffs, and lessons learned. Our solutions to the N-Queens problem exploit parallel hardware structures within and between multiple FPGAs, the latter by means of interprocessor communication with the Message Passing Interface (MPI), the dominant programming model in cluster computing. Experiments are conducted using a reconfigurable computing cluster of four nodes in our lab. Each node in the cluster is a dual-Xeon server that houses a Celoxica RC1000 reconfigurable computing board containing a Xilinx Virtex 2000E device, and the nodes are interconnected with high-speed networks including Gigabit Ethernet, InfiniBand, and SCI. The results demonstrate performance tradeoffs of parallel search problems such as N-Queens on reconfigurable architectures in commodity-based clusters, in terms of problem size, device size, and decomposition strategy. Included are lessons learned in the design and analysis of solutions to this problem, as well as comparisons with implementations on conventional computing systems.

¹ Corresponding author, email: troxel@hcs.ufl.edu, telephone: 352-392-9046.