Research Paper Review – Deep Blue

By Ashvin Someshwar

A brief summary of the paper's goals or techniques introduced

This paper describes the Deep Blue machine built in 1997 that defeated the Grandmaster Gary Kasporov. Deep Thought 1 and Chip Test were the first iterations of a game playing AI that was developed by CMU in the 1980s.

The Deep Blue system is massively parallel designed to carry out chess game tree searches. Deep Blue search was in 3 phases. The master node searched the upper levels, the intermediate levels were searched by the workers and final levels were searched by chess chips. Deep blue uses quiescence search, iterative deepening, transposition tables and NegaScout just to name a few ideas.

Deep Blue uses a chess chip which models a silicon chess board through an 8*8 array of combinatorial logic. A finite state machine controls move generation. Moves are computed simultaneously ensuring low latency. Evaluation function was split into 2 types "fast" and "slow" evaluation. The fast evaluation function computes score for positions that can be evaluated quickly such as the sum of weights of pieces on the board. The slow function evaluates one column at a time for complex positions. Recognized features have programmable weights which can be adjusted easily. Search control used a null-window alpha beta search which was optimized to detect cycles and repeated moves. Hardware search is simple and fast in deep blue. It does not have the complexity and efficiency of software search. For this reason chess chips carry out only shallow searches. This is 4-5 ply in middle games and deeper searches in end games.

A new type of search was introduced in Deep Blue 2 which was named "*Dual credit with delayed extensions*". This is to do with searching when there are forced/forcing pair (ffp) of moves. These are essentially moves where one or both players are forced to make a particular move to delay/extend the game for example through repeatedly checking the king. This results in a search "explosion" or a non-terminating search. To solve this problem first "fail low" or moves which are below the current expectation level are not searched. Fractional extensions were introduced where for a less forcing ffp a smaller search was done. A credit mechanism was created which allows ffps to accumulate credit and search was extended only when sufficient credit was accumulated. Sometimes both sides are in a ffp move sequence and moves are not "fail low" (Principal Variation). In that case a dual credit system was created where both sides accumulate credit and search is extended only for the side which accumulates credit. A large set of mechanisms were introduced to determine how nodes generate credit. The algorithm used in Deep Blue was a combination of Negamax and Credit Algorithm. If the credit score increases above the credit limit then search is performed recursively with the number of plies = credit – CREDIT-LIMIT. If the new score after search is greater than the best score then a new credit is generated.

One main technique introduced in Deep blue is parallel search which enabled the system perform much better than its predecessors. Parallel search in Deep Blue was done using Message-Passing Interface (MPI) via a high speed switch. It has a processor hierarchy where one SP node controls 29 other SP nodes and each of those 29 control 16 chess-chips each. A mechanism was created to determine where parallelism should be allowed. The load balancing in parallel search was implemented by terminating long running searches and shifting the search to software. To prevent the master node from being overloaded, search was optimized in a such a way so that the workers always had a task. Sharing between slave nodes was prevented and communication was only to master.

The evaluation function is a sum of feature values. Over 8000 features were recognized by the system and each was assigned a value. Features were also static and dynamic where the values for dynamic features were initialized at the beginning of a search. Initialization of feature values was done at the master node.

Deep blue had an opening book consisting of 4000 moves prepared by hand. Also interestingly it had a grandmaster game database which influenced deep blue's moves by rewarding or penalizing certain positions. It also had endgame databases which is a set of positions with 5 or less pieces on board. Deep blue also had a time control mechanism which ensured that it did not exceed time limits in the game.

A brief summary of the paper's results

Deep blue defeated grandmaster Gary Kasporov in the 6 match game by 3.5-2.5. This paper describes the system and the design considerations that took place in creating this system.

Deep blue had a number of enhancements to it when compared to Deep Thought 1. It had 24 chess engines when compared to Deep Thought 1's 2 (later 4 and 6) engines. Larger RAM when compared to its predecessor. The search function was completely rewritten and a new opening book was incorporated. The redesigned chess chip was able to evaluate 8000 features. And search speed was 2-2.5 million positions per chip when compared to Deep Thought 1's 1.6-2 million positions.

Deep Blue's parallel search comprised of 24 chess chips was compared to the single chip version. For complex position speed increased by a factor of 7, with improved efficiency of 30% and for simple moves the speed improved by a factor of 18 and efficiency improved by 75%.

In conclusion the team determined that the parallel search efficiency could have been increased. Hardware search and evaluation functions could be improved through the external FPGA. Although the FPGA was present in the current system it was not used due to time constraints. This search in Deep Blue did not have a pruning mechanism and research suggests that it could have been improved significantly using pruning.