Research Review: Deep Blue

Deep Blue is a famous machine that beat the World Chess Champion Garry Kasparov. Being a multiple state game with long and deep paths to follow Deep Blue used a combination of specialize hardware and software to evaluate the multiple options presented on a chess game. The famous game was on 1997 but a lot of the concepts and algorithms used to implement Deep Blue are still relevant today.

The researched article explains briefly some of the technical details that allowed Deep Blue to accomplish its task. It is important to recognize that Deep Blue was not the first of its kind nor the first iteration, it has ancestors which were implemented 17 years before it success. The first ones were called ChipTest and Deep Thought starting on 1980, the former was the first chess machine to defeat a Grandmaster in tournament play. They already used a dedicated chess move generator chip. Afterwards Deep Thought 2 implemented several improvements in 1989-90, Deep Blue I improved the chess chip design and performance and Deep Blue II, the one that beat Kasparov, introduced significantly hardware and software enhancements.

Deep Blue's architecture was impressive for the time being and even now it is still a remarkable machine. It has 30 nodes, one processor each, and 480 single chip chess search engines, with 16 chess chips per processor. Each chess chip can search between 2 to 2.5 million chess position per second. This architecture allowed a mixed software, hardware and parallel search.

The hardware search takes place on the chess chip which divides into three parts: the move generator, the evaluation function and the search control. The move generator is based on the Deep Thought move generator chip, it is implemented as an 8 x 8 array of combinatorial logic controlled by a hardware finite state machine. The evaluation function uses a combination of a fast evaluation and a slow evaluation triggered by programmable weight allowing to adjust their importance very easy. The search control uses several states machines to implement null window and alpha-beta search.

Deep Blue's software search uses a selective search they called "Dual Credit with Delayed Extensions" that basically assigns credit to the possible moves and cashes in when a threshold is surpassed. The credit is assigned by using several heuristics. For example: a singular move, which is one that is significantly better than any other move, will have a high amount of credit while if there are more alternatives moves with similar results it will have lower credit.

To control all the nodes and chess chips, Deep Blue has a parallel search algorithm that determines which node should be used and when a node should stop searching a specific search path. It uses a centralized and static distribution control handled by a single node that controlled the other 29. The early iterations are carried out on the master node but on further and deeper iterations the main node starts balancing the load between the rest of the nodes to achieve a better performance and deeper search. Per the article the parallelism performance on Deep Blue is hard to measure and have a lot of room for improvements.

The evaluation function implemented on the chess chip has more than 8000 different patterns, each is assigned a value a produced a different result. Due to the complexity of the function the article explains a single case called Rooks on files which basically evaluates the rooks state depending if they are blocked by a pawn, unopposed or other features that determines how good the state is.

Finally, the article concludes that the success of Deep Blue was not due one factor but due to the sum of all the factors mentioned above. Also, it states that the parallelism is a wide area of improvement on the system and that time restrictions make them take some choices without exploring the alternatives which suggests that the architecture may accomplish greater tasks and better performance than the achieved.