Deep Blue

Introduction

Deep Blue was a computer system (s/w + specialized h/w) developed by IBM in the early 90's that decisively beat Garry Kasparov in a chess tournament played under normal international chess tournament rules. It was the culmination of a long stretch of earlier efforts and was one of the most seminal events in the world during that era.

Overall system architecture

At its core, Deep Blue was a program to perform adversarial search with several commonly used techniques like alpha-beta pruning and quiescence search. But all elements of the sytem were built with the purpose of playing chess. (and as noted below, in some ways it was even specialized further by being built to play against Garry Kasparov).

It had a massively parallel architecture. There were 16 CPU nodes, and each of these nodes had 30 dedicated h/w coprocessors (hence forth referred to as chess chips) that were purpose built to evaluate chess positions. There was also an ability to extend the coprocessors even more by using FPGAs but this ability was not used in practice.

The search algorithm was a combination of software search that executed on the CPUs and h/w search that ran on the chess chips. One main distinguishing design of the search algorithm was that it was *non-uniform* search .i.e. some subtrees were explored to a greater depth. This decision was taken because it was known that good human chess players think *non-uniformly* when evaluating a chess position.

The evaluation function was entirely in h/w and ran on the chess chips. There were 8000 chess specific features that the evaluation function understood. The overall function worked by assigning weights to these features and computing a score for the position. The weights would further be adjusted dynamically as the game progressed.

There were predefined game books (i.e. databases) for opening and end game phases. These game books were hand-curated by other international chess grand masters and the opening books in particular were tuned keeping in mind that the opponent would be Garry Kasparov.

The following sections will provide some details about some of the above aspects of the systems

Search strategy

One of the 16 CPUs would act as the master node and start the search. It would evaluate the tree near the root and then hand subtrees to the other CPUs . Each of the slave CPUs would evaluate the tree near the middle levels and hand off further evalutions to their local chess chips which would in turn be responsible for evaluating the tree near the leaves.

Thus the search ran on CPUs (i.e. general purpose s/w programmable) and also on the chess chips (hardcoded h/w search). The s/w search was flexible in the sense that it could be reprogrammed easily while the h/w search was relatively inflexible and the only programmability it had was the weight assigned to various features.

The above can be thought of as an operational description of the search. From a functional viewpoint, the s/w search can be thought of as the *high level* strategy that used the h/w search as a subroutine to evaluate positions. One of interesting design decisions regarding the search was to make it *non-uniform* where the idea was to exploit chess knowledge and explore promising subtrees to a greater depth. One such example is identifying a streak of forcing/forced moves and explore such a streak to a greater depth in the hopes of finding a decisive advantage. To achieve such non-uniformness, the search used a credit system where a position accumulated credit based on several factors and if its accumulated credit was larger than a threshold then the position would be immediately evaluated further (i.e. non-uniform expansion of the tree)

Position evaluation

The board evaluation happened in the chess chips. The chips performed a *fast* evaluation of the board in one clock cycle and a *slow* evaluation that scanned the board one column at a time to evaluate it for chess specific notions like pins, positional advantage and so on.

Move generator

The move generator was also implemented in the chess chips. The move generator could calculate all possible moves from a position and then would order the moves according to some chess specific notions (like low valued piece attacks a high value piece, positional advantage and so on). This ordering would determine which moves are then further expanded as per the high level search strategy. It also had some higher abstractions and , for e.g. , could be asked to *generate all attacking moves in this position* and so on.

Opening and end game books

These were databases that were hand-curated by human chess grandmasters and made available to Deep Blue. They contained summarized versions of games and positions from actual grandmaster level chess games. Deep Blue could consult these databases to determine what the consensus move was for a given position.

Summary

Deep Blue was a remarkable achievement and it played a huge part in showcasing AI based programming techniques to the world at large. Its success was largely due to being purposefully built from scratch, leveraging s/w and custom built h/w to do one thing and do it well - play chess. It showed how adversarial game playing ideas from AI research could achieve seemingly impossible results like defeating the best human chess player in the history of the game.