Deep Blue
Research Review
By Param Shah

Deep Blue was a Chess Playing Computer, developed by IBM in the 1990s. This paper gives us a captivating look into the development and progressive improvement of the first-of-its-kind Artificially Intelligent system that beat the world chess champion Garry Kasparov in a 6 game match during 1997.

The major factors that ensured Deep Blue's success are:

- Software Search
- Hardware Search
- Hardware based evaluation function
- Massively Parallel Search

Software search:
Software search, named as "dual credit with delayed extension", works on basic principle of extending forcing/forced pair (ffp) of moves. Ideally, to make sure that it is a ffp, the search is extended 2-ply for each ffp. But this leads to a search explosion. Therefore, techniques like:

- Eliminating the move as an ffp if it is 'fail low'
- Fractional extensions (e.g. 1.75-ply for each fpp)
- Delaying extension by providing a threshold credit (a series of fpp's accumulate more credit, and when the threshold is reached it is cashed in for an extension).
- Dual Credit
- Preserving the search envelop to avoid oscillating search

Hardware Search:
Deep Blue make use of a chess chip that carries out a fixed-depth null window search which includes a quiescence search. This searching can be made a lot faster by making use of multiple chips simultaneously. Hardware search makes use of many chess tactics. Some of the interesting ones are:

- Number of mating checks allowed for each side in quiescence search
- Number of singular checking moves allowed in quiescence search
- Flag to ignore stalemate at one ply above quiescence

Hardware based evaluation function:
Hardware evaluation function, which is evaluated in the Deep Blue chip, is composed of "fast evaluation" and "slow evaluation".
Fast evaluation contains all the easily computed major evaluation terms with high values. E.g. Piece Placement value.
Slow evaluation scans the chess board one column at a time, computing values for chess concepts such as square control, pins, king safety, etc.
Both the fast and the slow evaluation functions have programmable weights, allowing their relative importance to be easily adjusted.

Massively Parallel Search:
To utilize all the available resources at once, Deep blue implements Parallel search by first balancing the load to either hardware search or software search, then ensuring that the worker node always has a job in hand when it completes the active job, and then allowing communication between worker nodes only via the master node.

Database of Moves:

Deep blue makes use of 3 databases of moves: Opening Book, Extended Book and Endgame databases.

The *Opening Book* includes tactically complex openings and positional openings.
In absence of information in the Opening Book, the *Extended Book* was used, which was created by analysing past games played by the grandmasters. It also included analysis of the reruns of the earlier games to guess the best possible move for a situation.
The *Endgame Database* consists of all the chess positions with less than 5 pieces on the board and selected positions with 6 pieces that included a pair of blocked pawns.

Conclusion:
Success of Deep Blue was due to many features shown above. However, there were many more possible improvements such as a faster parallel search, better pruning mechanism, flexible hardware evaluation function, etc.
But, it was very interesting to read that Deep Blue (that beat the WORLD CHAMPION in chess!) used many of the concepts that we have implemented in this project. This is a huge motivation.