Background

Chess game tree on an average has a branching factor (b) of 35 and number of moves (n) of both players is ~100; it has game complexity of O(b^n) ~10^154. Searching the tree for optimal solution thus becomes impractical. Besides aggressive pruning, it becomes very important to apply heuristics in the game the way humans do.

Goal

The paper describes the system architecture of Deep Blue chess machine which defeated then world chess champion Gary Kasparov in 1997. Deep Blue from IBM is massively parallel system optimized for chess games searches. It is composed of 30 IBM RS/6000 SP computer nodes (1 Master and 29 slaves in a static configuration) with 16 chess-chips attached to each of the nodes. Chess chips can carry fixed-depth searches on hardware and can communicate with host nodes via microchannel bus. The master-slave architecture with chess chips enabled Deep Blue

1. to perform massively parallel searches starting the root search from master node (and its chess chips) and distributing it down to other nodes
2. perform a combination of software and hardware search by keeping the complex recursive extensions with transposition tables in software and complex quiescence searches with local extensions (mostly leaves) in hardware. This provides both speed and flexibility

Software Search

Software Search also called “dual credit with delayed extensions”. It uses a notion of credit on both sides and a CREDIT_LIMIT to control the depth-limited version of alpha beta algorithm (using nagamax function). It uses principles like *extend forcing/ forced pair of moves* by delaying the search extensions or doing fractional extension or applying dual credits.

Nodes can receive credits based on established chess heuristics like *threat, mate threat, Absolute Singular, Singular, Binary, Domain dependent (Check Evasion, Pawn push etc).* The intent is to delay / prune the search if the choices are obvious or deepen the search if confidence is not high by controlling the credit scores.

Parallel Search

Deep Blue parallelize search based on the types of nodes viz.

1. Type 1 PV nodes - After the first move has been examined at a PV node, all the alternatives may be examined in parallel
2. Good type 2 nodes - nodes where the first move “fails high”, or exceeds expectations
3. Bad type 2 nodes - nodes where the fail high move is not searched first
4. Type 3 nodes - nodes where all the moves fail low

It uses global synchronization points along with Type 1 and 2 nodes. The parallel search in Deep Blue is non-deterministic and is influenced by timing and processor job assignments etc. It has an observed efficiency of 8% in tactical positions and 12% in quieter positions.

Evaluation Function

Evaluation Function is essentially a sum of over 8000 feature values which can be recognized on chess chip. They represent patterns including simple ones like values of particular pieces to very complex ones. Deep Blue has a built in time control with two distinct targets - normal timing and panic timing.

Open and extended book

It uses opening book of about 4000 positions (prepared by hand primarily by Grandmaster Joel Benjamin) and an extended book which is a database of 700,000 games. Besides it uses endgame databases (primarily from the Ken Thompson CDROMs) for all five/ most six pieces games. This provided Deep Blue an option to use human heuristics to favour certains moves over others and reducing search space.

Conclusion

The paper concludes that the success of Deep Blue in the 1997 match was not the result of any one factor. The large searching capability, non-uniform search, and complex evaluation function were all critical. However other factors also played a role, e.g., endgame databases, the extended book, and evaluation function tuning.

References

1. https://pdfs.semanticscholar.org/ad2c/1efffcd7c3b7106e507396bdaa5fe00fa597.pdf
2. https://www.cse.iitb.ac.in/~cs621-2011/2011-seminars/foiltex-example.pdf