A better understanding of the game of chess

The situation

It was a Saturday night, during a ski weekend, that Giles and I decided to have a game of chess right by the fireplace in the cozy little flat, right on the foot of the Italian Alps, where we were invited to stay by our coworker Silvia.

This was an ideal setup for chess playing as you can see below:

Picture 1 - This picture should make chess aficionados salivate

The problem

After a game that started well for me but ended in defeat, Giles and I got into a discussion of how possible it would be for a super computer to calculate the full game tree for a given chessboard state.

Giles argued that it would be impossible for a massively parallel computer to calculate such a tree in a time period that would not be astronomical in size. In fact he mentioned that it would probably take something like the age of the Universe for such a calculation to be made.

I, on the other hand, thought that the time required for such a calculation would probably be much less than Giles' pessimistic forecast. However, I could see that in order to get a better appreciation of the numbers involved I needed to think about the problem in a less relaxed manner and employ the good old paper and pen formalization artifacts. In fact, I admitted right away that I may be talking nonsense as I had never thought about this problem before.

The calculations

So, next morning, while the others were gone for their second ski day (I stayed home rubbing and stretching my sore limbs) I decided to take some time and do some math. Turns out blank paper is a quite rare commodity during ski trips. So here is my calculation scratch paper (a Pirelli newspaper ad):

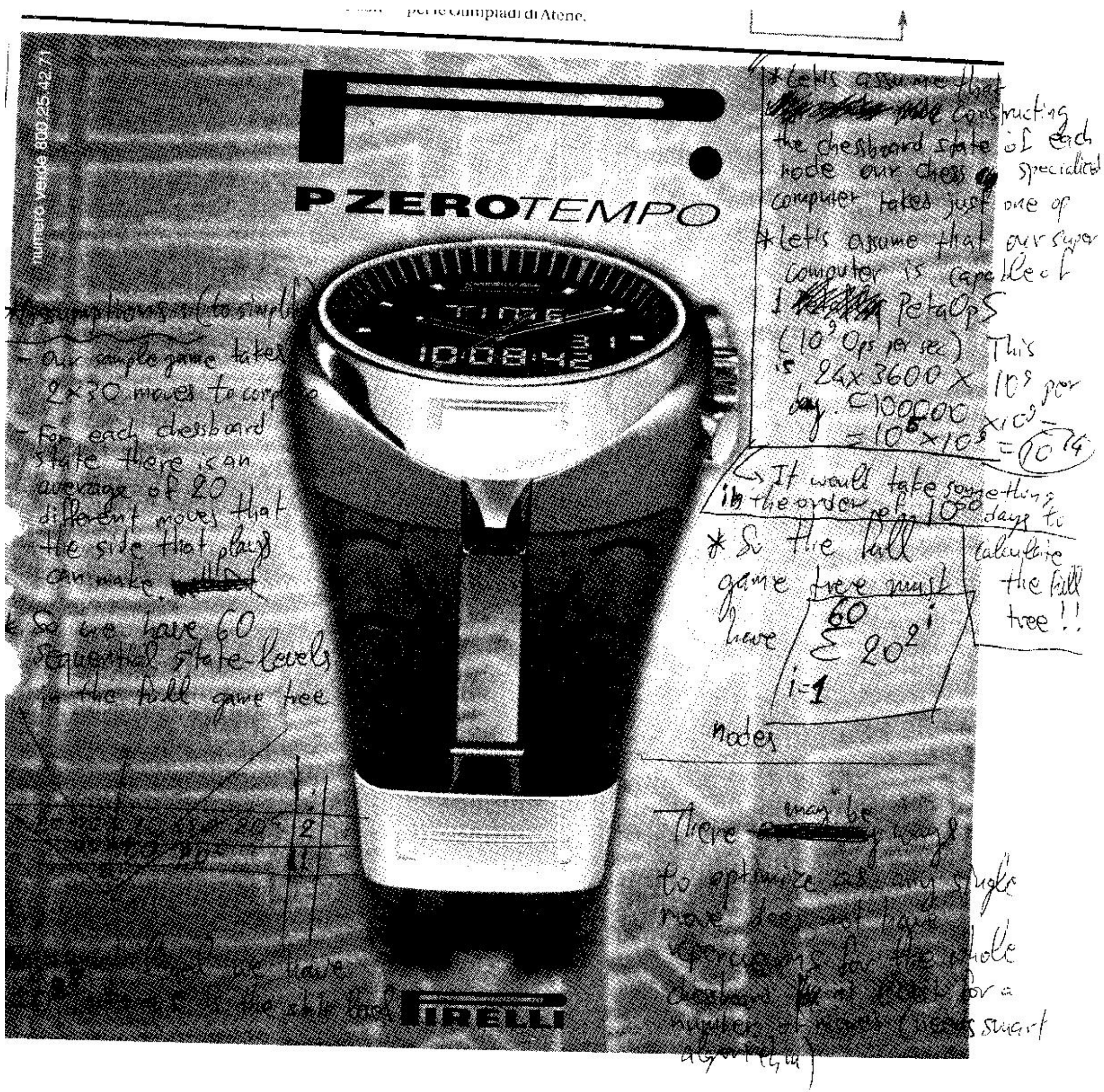

Figure 1 - My quick calculations, on Sunday morning, were not quite correct but succeeded in giving me a better appreciation of the numbers involved. The error was more due to absent-mindedness than mathematical thinking ineptitude which helps make me feel better. See below for correct calculations.

The calculations as seen in the scan above are based on a set of assumptions and culminate in the extraction of a formula that can be used to calculate the number of nodes in a full game tree given the number of average possible moves per chessboard state.

The error lies in my absent-minded assumption that each state level has 20^2 x (previous level's number of nodes) instead of just 20 x (previous level's number of nodes). It was probably the alluring resulting symmetry of the tree-like structure (seen depicted above) that you get when you go the 20^2 way that made me not realize the mistake.

So let's try the calculations once more hoping that we don't mess up this time:

The correct calculations

Assumptions (to simplify the problem)

- Our sample game takes 2x30 moves to complete
- For each chessboard state there is an average of 20 different moves that the playing side can make.
- We ignore the possibility of game paths that end the game before 2x30 moves

Discussion

So, our full game tree is 60 (2x30) levels deep.

……………………………….20^{60}
…………
20x20x20…………….20^3
20x20…………….20^2
20…………….20^1
(start of game) …………….

Thus the full game tree must have a total number of nodes given by the following formula:

$$\sum_{i=1}^{60} 20^i$$

$$\text{or}: \quad 20 + 20^2 + 20^3 + \ldots + 20^{60}$$

For the sake of simplicity let's assume that the total number of nodes in our sample game's full game tree is $\mathbf{10^{60}}$ (much less than what it actually is)

The super-computer

- Our chess optimized computer is capable of calculating each chessboard state in just a single operation. Each node in our game tree corresponds to a chessboard state.

- Our massively parallel computer is capable of 1 TeraOpS (10^{12} operations per second). Note here that at the start of the 21st century very fast supercomputers go well above 1 TeraFLOP. Also, note another mistake in my Sunday morning calculations. I am talking about a PetaOp supercomputer but I am marking it down as 10^9 (instead of 10^{15} which is what *peta* stands for). There are no petaOp supercomputers yet.

How long would it take?

Thus, in one day our supercomputer can calculate this many chessboard states (or game tree nodes):

$24 \times 60 \times 60 \times 10^{12} = 86400 \times 10^{12}$

or roughly

$100{,}000 \times 10^{12} = 10^5 \times 10^{12} = \mathbf{10^{17}}$

So, to calculate the full tree of 10^{60} nodes, our supercomputer would need:

$10^{60} / 10^{17}$ days or **<u>10^{43} earth days</u>**

Notice here, that perhaps intuitively (besides the fairly gross error in my calculations), I arrive to roughly the same result in my Sunday morning calculations. Ok, I was pretty damn lucky.

The age of the universe is not more than 20 billion years or roughly $10 \times 10^9 \times 2 \times 365$ or (to help calculations) $10^{10} \times 10^3 = 10^{13}$ days

So, it turns out, that it would not only take something like the age of the Universe, for our super-duper computer to calculate the full game tree for a game of chess, but much longer.

Note however that our calculations assume no optimizations and shortcuts are being used in calculating the tree.