

chess_db: A framework for working with large chess game datasets

Nicos Angelopoulos

Jan Wielemaker

University College London

SWI-Prolog solutions

<https://stoics.org.uk/~nicos>
nicos@email.stoics.org.uk

26.07.21

chess_db a suite of logic programming tools that can effectively manipulate chess games both in memory and via creating back end databases.

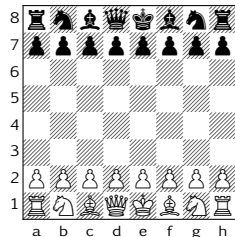
- ▶ representation and manipulation of board as a dictionary
- ▶ games as Prolog terms
- ▶ reading/writing Prolog terms to PGN files
- ▶ game terms to/from databases

relational: *proSQLite*

key-value: *Berkeley, RocksDB*

board as dictionary

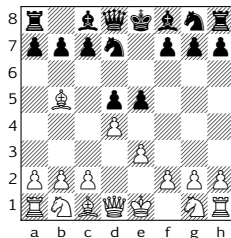
```
chess_dict_start_board(Board) :-  
    Board = board{  
        % castling, eg cwk: castle white king-side  
        % 1: available, 0: not-available  
        cwk:1,cwq:1,cbk:1,cbq:1,  
        % en passant square  
        eps:0,  
        hmv:0, % half moves clock:  
        % since last take or pawn move  
        fmv:0, % full moves: played so far  
        8:10,16:8,24:9,32:11,40:12,48:9,56:8,64:10,  
        7:7 ,15:7,23:7,31:7 ,39:7 ,47:7,55:7,63:7 ,  
        6:0 ,14:0,22:0,30:0 ,38:0 ,46:0,54:0,62:0 ,  
        5:0 ,13:0,21:0,29:0 ,37:0 ,45:0,53:0,61:0 ,  
        4:0 ,12:0,20:0,28:0 ,36:0 ,44:0,52:0,60:0 ,  
        3:0 ,11:0,19:0,27:0 ,35:0 ,43:0,51:0,59:0 ,  
        2:1 ,10:1,18:1,26:1 ,34:1 ,42:1,50:1,58:1 ,  
        1:4, 9:2,17:3,25:5 ,33:6 ,41:3,49:2,57:4 ,  
        0:0 % white move = 0; black = 1  
    }.
```



moves on dictionaries

Taken a board dictionary and the move **Nf6**, code

- ▶ disambiguates
- ▶ translates to a number of operations



disambiguation of move **Nf6**, means that only knight at **g8** can be moved as the knight at **d7** is "pinned".

Operations include:

- ▶ removing piece from a square
- ▶ adding a piece to a square
- ▶ updating castling and en passant rights
- ▶ clocks

PGNs Portable Game Notation

```
[Event "IBM Man-Machine"]  
[Site "New York, NY USA"]  
[Date "1997.05.11"]  
[EventDate "?"]  
[Round "6"]  
[Result "1-0"]  
[White "Deep Blue (Computer)"]  
[Black "Garry Kasparov"]  
[ECO "B17"]  
[WhiteElo "?"]  
[BlackElo "?"]  
[PlyCount "37"]
```

```
1.e4 c6 2.d4 d5 3.Nc3 dxe4 4.Nxe4 Nd7 5.Ng5 Ngf6 6.Bd3 e6 7.N1f3 h6 8.Nxe6  
Qe7 9.0-0 fxe6 10.Bg6+ Kd8 11.Bf4 b5 12.a4 Bb7 13.Re1 Nd5 14.Bg3 Kc8 15.axb5 cxb5  
16.Qd3 Bc6 17.Bf5 exf5 18.Rxe7 Bxe7 19.c4 1-0
```

PGNs to Prolog terms

chess_db includes parser for reading and writing from/to PGN files and Prolog terms.

It also incorporates incremental reading, when N games are read and processed (usually in populating DB tables) without being kept in memory.

This allows for very large PGN files to be processed into databases although they would n't fit in memory as Prolog terms.

databases

Four tables:

- ▶ game info
- ▶ game moves
- ▶ game original text
- ▶ position

Two types of database backends

- ▶ relational (*proSQLite*)
- ▶ key-value (*Berkeley, RocksDB*)

Games to databases

Relational databases are better match for game info and moves.

Original text is fine with either relational or key-value DBs.

Positions database is more naturally and efficiently mapped as a key-value entry, where key is a unique position identifier.

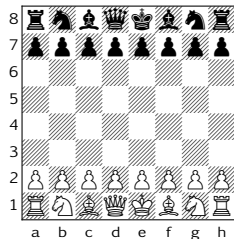
Position tables are crucial in training.

unique position identifiers

Following lichess NN training representation, we compact each board to an as-short-as-possible unique hex number. First part is the occupancy and the second part is the specific pieces in each ordered position.

C3C3C3C3C3C3C3C3 D17E21783179517B617C31792178D17E

```
chess_dec_hex(0, 0'0). % en passant pawn
chess_dec_hex(1, 0'1). % white pawn
chess_dec_hex(2, 0'2). % white knight
chess_dec_hex(3, 0'3). % white bishop
chess_dec_hex(4, 0'4). % white rook
chess_dec_hex(5, 0'5). % white queen
chess_dec_hex(6, 0'6). % white king
chess_dec_hex(7, 0'7). % black pawn
chess_dec_hex(8, 0'8). % black knight
chess_dec_hex(9, 0'9). % black bishop
chess_dec_hex(10, 0'A). % black rook
chess_dec_hex(11, 0'B). % black queen
chess_dec_hex(12, 0'C). % black king
chess_dec_hex(13, 0'D). % white castling rook
chess_dec_hex(14, 0'E). % black castling rook
chess_dec_hex(15, 0'F). % black moving king
```



position database experiments

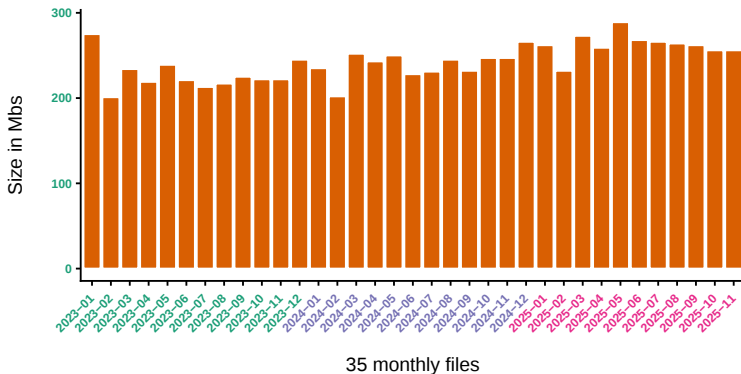
proSQLite is a very convenient database however, it does not scale to millions of games.

To test the performance of key-value databases and particularly *RocksDB* we downloaded 35 monthly games from elite players played on the *lichess* open source, free to access website.

These total 11 million games.

lichess.org elite datafile sizes

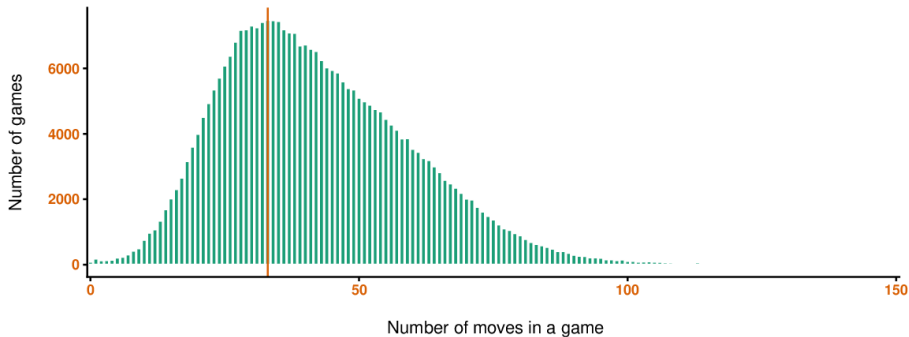
Elite Lichess PGN file sizes. January 2023 – November 2025



Quite stable between 200 and 300Mbs uncompressed.

Game length distribution

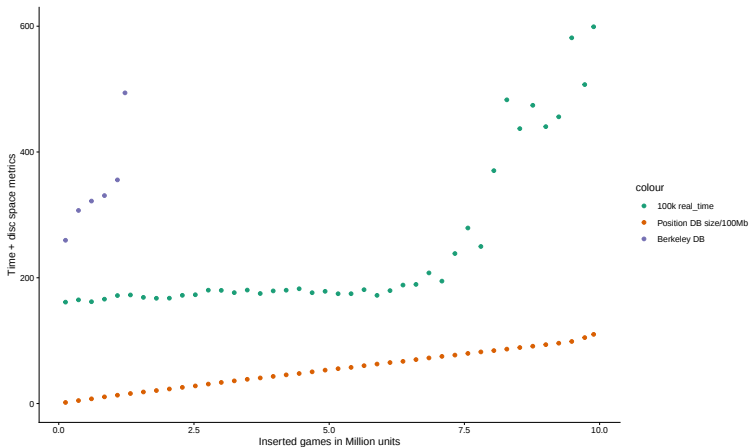
Elite Lichesss PGN, January 2023. Game lengths distribution.



...for the *Elite lichess* PGN file for Jan. 2023

Database performance

Averages over 10: for metrics per 10k insertions



chess_db

Can be easily installed as it is an *SWI-Prolog* pack.

Thank you