Chess on Golem

Summary

This project was created as an entry to Golem Hackathon 12/2020.

It's purpose is to show that any state based game / problem can be run in Golem Network and solved interactively by provider nodes.

This particular example shows classical chess game played by two AI players that facilitate golem network for computing.

Whole game is managed by Node.js server which distributes chess computing tasks across Golem Network providers.

Each move request is put into Golem Market and calculated by provider that puts best offer. To show how computing power could affect outcome of the game each player is allowed to calculate next move with particular depth. (With enough provider nodes in network that could be achieved without forcing one of players to ask for computations with lower depth than opponent)

Outcome:

Node Chess App

Node.js Server that can be run on linux or windows machines that is responsible for handling game and requesting Golem Network for next moves for each AI player.

Is used also as backend server for GUI App that displays chess game in real time with some statistics regarding provider nodes work.

Currently runs at http:// 20.52.154.16/3970 on Linux Ubuntu VM.

It creates a request to golem network for each move that is being computed by AI players. For demo purposes one player asks for best move with depth precision of 20 and the other one uses depth of 1.

```
cd chess

yarn js:chess
```

```
PS D:\js\chess_on_golem\node_chess_app\chess> yarn js:chess
yarn run v1.22.5
$ node ./chess/index.js
secp256k1 unavailable, reverting to browser version
extract : [object Object]
Using subnet: community.3
starting pos:
   +------------------------+
 8 | r  n  b  q  k  b  n  r |
 7 | p  p  p  p  p  p  p  p |
 6 | .  .  .  .  .  .  .  . |
 5 | .  .  .  .  .  .  .  . |
 4 | .  .  .  .  .  .  .  . |
 3 | .  .  .  .  .  .  .  . |
 2 | P  P  P  P  P  P  P  P |
 1 | R  N  B  Q  K  B  N  R |
   +------------------------+
     a  b  c  d  e  f  g  h

Listening on port 3970...
input path: D:\js\chess_on_golem\node_chess_app\chess\tmp\game_132\input\step_0001.txt
D:\js\chess_on_golem\node_chess_app\chess\tmp\game_132\output
2021-01-06 00:32:50 [yajsapi] info: GFTP Version:0.1.2
@@@@@@@@@@@@@@@@@@@@@@@ TASK hash_00000132_0001 /  computation started
2021-01-06 00:32:52 [yajsapi] info: Demand published on the market
```

Script runs until game is finished, when some calculation fails or timeouts golem is being asked to perform it again.

Multiple clients can connect to socket.io websocket server and listen for events that describe current game state

Events:

```
• currentTurnEvent
• providerFailed
• computationStarted
• movesRefreshed
• gameFinished
• offersReceived
• agreementCreated
• agreementConfirmed
• computationFinished
• invoiceReceived
• moveEvent
• positionEvent
```

When client reconnects server sends him automatically current state of the game.

Chess on Golem Viewer

React application that serves as GUI for displaying chess game progress for Chess on Golem.

It displays game progress and some interesting stats regarding provider nodes that took part in calculations.

There is live demo available at:

http://chess-on-golem-viewer.herokuapp.com/

If It's not running you can request start at pawel.burgchardt [A-T] gmail.com

DEMOs

- Demo v0.3

https://www.youtube.com/watch?v=Wp_lJEeN7UA&feature=youtu.be&ab_channel=Pawe%C5%82Burgchardt
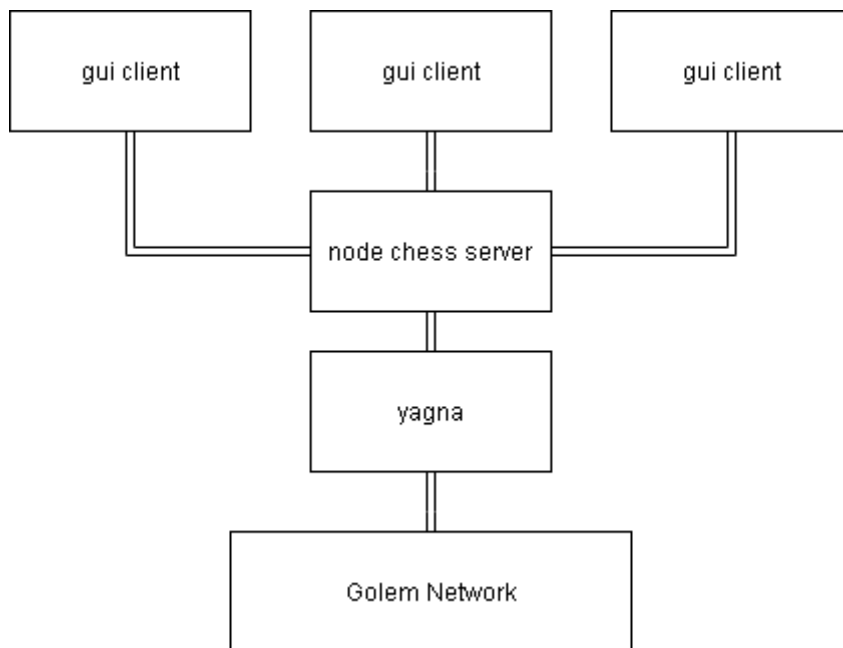
- Demo v0.2

https://www.youtube.com/watch?v=C65uTAZAsRA&list=UUxg1Vq50vwy7Pm3kFwb0ZQg&index=2&ab_channel=Pawe%C5%82Burgchardt

- Demo v0.1 (problem with some providers' payments)

    https://www.youtube.com/watch?v=cTD0zq7jURM&list=UUxg1Vq50vwy7Pm3kFwb0ZQg&index=3&ab_channel=Pawe%C5%82Burgchardt

System architecture:

Author:

Paweł Burgchardt 12.2020 / 01.2021

License: GPL v3