Computação Distribuída
Licenciatura em Engenharia Informática
Ano Letivo 2020/2021 2ºAno 2º Semestre

Relatório do Trabalho Prático Final

Password cracker

Professores:
Diogo Gomes
Nuno Lau

Trabalho realizado por:
Alexandre Serras 97505
Gonçalo Leal 98008

Arquitetura

Para fazer o nosso sistema Peer-to-Peer, decidimos criar uma classe Server e outra classe Client, com isto, vamos de forma aleatória escolher 1 dos processos para ser Servidor, quando 1 deles conseguir tornar-se Servidor, os outros passam a ser Client's.

Após ser feita a decisão, vamos começar a trocar mensagens entre os processos para se começar a descobrir a palavra passe.

Para tal, no nosso SD, vamos ter tanto mensagens enviadas para o grupo multicast com mensagens enviadas via unicast.

À medida que se vai entrando um peer no sistema distribuído, vai juntar-se na procura da palavra password.

Para fazer a distribuição, o servidor fica encarregue das palavras com os índices k+peers+1, o primeiro que se ligar fica com k+peers+1 e o último vai ficar com k+peers+1.

Sendo que é como se fosse o k do servidor no 0, primeiro fica com k=1 e o último k=2

Exemplo: No índice 2 liga-se 1 cliente, no instante 6 liga-se o C2

Índices										
0	1	2	3	4	5	6	7	8	9	10	11
S	S	S	C1	S	C1	S	C1	C2	S	C1	C2

Protocolo

Decidimos que no nosso protocolo as mensagens vão ser trocadas em JSON

Mensagens multicast:

"command":"keep_alive","slave": self.peers,"ip":self.ipslave} -> Nestas mensagens servem como keep alive, onde enviamos para o grupo multicast, o nosso conjunto de peers conhecidos, conjunto slave, e indicamos no campo ip o nosso ip

Mensagens unicast:

"command":"sync", "slave": self.peers, "ip": self.ipslave, "wrongs": self.cracker.get_wrong_passwords() -> Nestas mensagens sync são para sincronizar , indicamos no campo slave o nosso grupo de peers , no campo ip o nosso IP e no wrongs, indicamos as passwords já tentadas e que foram erradas

"command":"connecting_ack", "slave":self.peers, "ip":self.ipslave,"wrongs": self.cracker.get_wrong_passwords() -> Esta mensagem serve de resposta ao connect da socket ao servidor e tem exatamente os mesmos campos que a anterior.

{"command":"connecting","ip":self.ipslave} -> Mensagem enviada apenas pelos clientes, para se conectarem ao servidor

"command":"found_it_ack" -> Mensagem que serve para confirmar que recebeu a mensagem de que o resultado foi encontrado

"command":"found_it", "ip": self.ipslave}-> Mensagem para informar que encontrou a palavra pass e que assim vão poder finalizar as tarefas de procura

Mais algumas mensagens

Resultados obtidos

O resultado é encontrado rapidamente sendo que quanto mais perto da letra ‘a’ estiver mais rápido são os resultados

Para palavras pass de tamanho 1 funciona, e é bastante rápido.

Para os restantes tamanhos não foram feitos testes