Quebra de senhas

Abilio
Vanderson

20 de agosto de 2013

Sumário

Introdução

- O trabalho consistia em quebrar as 270 senha encripto grafadas em HASHs MD5.
- Essas estavam divididas em três grupos de 90 senhas, com senhas de 4,5 e 6 caracteres respectivamente.
- Para tal foi criado um programa em Java que usada uma estratégia de força bruta, dividindo o alfabeto pelo numero de Threads do processador usado.

Algorítimo 1

```
for (int i = indexInicial; i < indexFinal; i++)
   for (int j = 0; j < tamanhoAlfabeto; j++)
      for (int k = 0; k < tamanhoAlfabeto; k++)
         for (int l = 0; l < tamanhoAlfabeto; l++)
            for (int m = 0; m < tamanhoAlfabeto; m++)
               for (int n = 0; n < tamanhoAlfabeto; n++)
                  p = (String.valueOf(alfabeto.charAt(n)));
                  p = p.concat(String.valueOf(alfabeto.charAt(m)));
                  p = p.concat(String.valueOf(alfabeto.charAt(l)));
                  p = p.concat(String.valueOf(alfabeto.charAt(k)));
                  p = p.concat(String.valueOf(alfabeto.charAt(j)));
                  p = p.concat(String.valueOf(alfabeto.charAt(i)));
                  findHash(p);
```

Algoritmo 1 (n^{15})

- 4 laços "for" aninhados para achar as senhas de 4 dígitos.
- 5 laços "for" aninhados para achar as senhas 5 dígitos.
- 6 laços "for" aninhados para achar as senhas de 6 dígitos.
- No total são $(85^4) + (85^5) + (85^6) = 85^6 + 5 + 4$ operações .
- $85^{15} = 8,73542191 \times 10^2$ operações.

Otimização (n^6)

```
for (int i = indexInicial; i < indexFinal; i++)
   for (int j = 0; j < tamanhoAlfabeto; j++)
      for (int k = 0; k < tamanhoAlfabeto; k++)
         for (int l = 0; l < tamanhoAlfabeto; l++)
            for (int m = 0; m < tamanhoAlfabeto; m++)
               for (int n = 0; n < tamanhoAlfabeto; n++)
                  p = (String.valueOf(alfabeto.charAt(n)));
                  p = p.concat(String.valueOf(alfabeto.charAt(m)));
                  p = p.concat(String.valueOf(alfabeto.charAt(l)));
                  p = p.concat(String.valueOf(alfabeto.charAt(k)));
                  p = p.concat(String.valueOf(alfabeto.charAt(j)));
                  p = p.concat(String.valueOf(alfabeto.charAt(i)));
                  findHash(p);
                  for (int m = 0; m < tamanhoAlfabeto; m++)
                     p = p.concat(String.valueOf(alfabeto.charAt(m)));
                     findHash(p);
                        for (int n = 0; n < tamanhoAlfabeto; n++)
                           p = p.concat(String.valueOf(alfabeto.charAt(n)));
                           findHash(p);
```

Otimização (n^6)

- Apenas 6 laços "for" aninhados para achar as senhas de 4,5 e 6 dígitos.
- No total são $(85^6) = 377149515625$ operações.

Hardware

- Notebook 1
  - Intel Core i5-3317U CPU @ 1.70GHz $\times$ 4
  - Número de Threads: 4
  - Memória: 6 GB DDR3-1333/PC3-10600
- Notebook 2
  - Intel Core i7-3630QM CPU @ 2.40GHz $\times$ 8
  - Número de Threads: 8
  - Memória: 6 GB DDR3-1333/PC3-10600

Resultado

- Para quebrar as 230 senhas o programa começou a rodar sexta às 20:00 h e ficou rodando interruptamente até terça 12:45 h
- Tempo total aproximado 80 horas