

Avaliação de Desempenho - Trabalho Prático 1

- Cícero
- Fernando Costa
- Marcos Rebelo
- Micael Levi
- Ramon
- Yan Matheus

OBS: Iremos começar pela `Parte 2` pois não será possível fazer a `Parte 1` sem criar o arquivo modificado do *trace*.

base de dados usada: CRAWDADE dataset copelabs/usense

Esse conjunto de dados configura três experimentos com quatro dispositivos Samsung Galaxy S3. Para cada um, foram coletados informações de localização, proximidade, movimento e som a cada 1 minuto, que estão dispostas em quatro arquivos textos.

Os arquivos que reúnem os dados de cada arquivo foram nomeados de "USenseX" (onde X vai de 2 a 5).

Para esse trabalho utilizaremos apenas os arquivos de nome "SocialStrength.dat", que estão nas "Source" de cada "USense" em cada experimento.

Árvore de diretórios do *dataset* (o mesmo se repete pra cada um dos 3 experimentos)

```
social-interaction
├── Experiment-1
│   ├── USense2
│   │   └── Source
│   ├── USense3
│   │   └── Source
│   ├── USense4
│   │   └── Source
│   └── USense5
│       └── Source
└── ...
```

Parte 2 - Ler o arquivo de trace original e criar um arquivo de trace modificado, escrito no formato para o simulador The ONE

```
[time][action=CONN][first_node_ID][second_node_ID][type={up,down}]
```

- **timestamp**: tempo (em segundos) que acontece o tipo de ação - **action**: sempre é CONN, ou seja uma conexão entre dois nós - **first_node_ID** e **second_node_ID**: id numérico dos nós (precisa sempre começar com 0 os id dos nós até $\text{MAX_ID} = \text{NUM_NODES} - 1$) - **type**: é o tipo de conexão - **up**: ligação bidirecional entre 2 nós - **down**: desconexão entre 2 nós

Métodos para a definição da coluna `type`

Visando os arquivos "SocialStrength.csv" (na pasta "Source") de todos os `USense` unidos em um único ordenado pelo `timestamp`

Dados: - **A**: instante/time anterior do anterior - **B**: instante/time anterior - **C**: instante/time corrente - **E_A**: valor de "encounter_duration" no instante **A** - **E_B**: valor de "encounter_duration" no instante **B** - **E_C**: valor de "encounter_duration" no instante **C**

No primeiro registro de conexão entre os nós x_i e y_i , o tempo do "up" será dado pela diferença entre o B_i e E_{B_i} . Isso foi assumido pois o registro indica a duração do encontro em um dado tempo, então, acreditamos que esse *dado* tempo (*timestamp*) não é o tempo do encontro, mas sim o tempo em que o registro foi realizado. Assim, o tempo do encontro é calculado pela diferença entre o tempo da observação e a duração do encontro (em segundos).

Caso 01: `encounter_duration` constante

Com o passar do tempo a duração do encontro (entre uma conexão) deve sempre aumentar. Quando o valor do `encounter_duration` de um próximo registro/observação não sofre alteração, então assumimos que houve uma quebra na conexão ("down") no minuto anterior.

Exemplo:

De

```
Ai;0;1;Eai
Bi;0;1;Ebi  (> Eai)
Ci;0;1;Eci  (= Ebi)

17/11-18:33:22.022;0;1;1665.80477517;1665.80477517
17/11-18:34:22.022;0;1;1711.580405317;1711.580405317
17/11-18:35:22.022;0;1;1711.580405317;1711.580405317
```

Para

```
Ai - Eai;0;1;up // se for a "primeira" observação de <0,1>
Bi;0;1;down

17/11-18:05:36.217225;0;1;up
```

```
17/11-18:34:22.022000;0;1;down
```

Caso 02: `encounter_duration` igual a zero após uma conexão realizada

Onde dois nós se conectam, i.e., houve um "up", mas em algum momento antes de um "down" (citado no caso anterior) o `encounter_duration` é 0. Nessa situação, assumimos que houve uma quebra na conexão ("down") no minuto anterior à essa observação. E a próxima observação dessa relação será tratada como primeiro "up".

Exemplo:

De

```
Ai;1;0;Eai
Bi;x;y;Ebi // [opcional] outras observações que não sejam <1,0>
Ci;1;0;0

16/11-18:59:55.055;CONN;1;0;1644.929152577;1644.929152577
...
16/11-19:00:56.056;CONN;1;0;0;0
```

Para

```
Ai - Eai;1;0;up
...
Ci - 60s;1;0;down

16/11-19:00:55.056;CONN;1;0;down
```

Caso 03: `encounter_duration` menor que a duração da observação anterior

Quando há uma conexão entre dois nós mas ocorre uma "quebra rápida" na conexão. Assim, o `encounter_duration` que antes estava se acumulando, reinicia, continuando a partir de um novo "up". Nesse caso, o "down" ocorre no *timestamp* da observação anterior.

Exemplo:

De

```
Ai;1;0;Eai // não sendo a primeira ocorrência da observação de <1,0>
Bi;x;y;Ebi // [opcional] outras observações que não sejam <1,0>
Ci;1;0;Eci (< Eai)

17/11-09:59:48.048;CONN;1;0;3570.730112787;1785.3650563935
17/11-10:00:48.048;CONN;1;0;29.87849218;14.93924609
```

Para

```
Ai;1;0;down
Ci - Eci;1;0;up

17/11-09:59:48.048;CONN;1;0;down
17/11-10:00:18.169508;CONN;1;0;up
```

Gerar arquivo formatado

Com o script em Python 3, de nome `parser.py` localizado na pasta `social-interaction`. Basta executar:

```
python3 "Experiment-1/" "."
|                               |
|                               +--- diretório de saída para o arquivo tratado
+--- diretório do experimento
```

Um arquivo de nome "base_final.csv" será gerado no diretório definido, contendo o *dataset* (do experimento definido) formatado para o simulador The ONE. O *timestamp* foi normalizado para segundos da seguinte forma: o tempo da primeira observação foi considerado como tempo 1, assim, os demais foram se adequando de acordo com ele.

Parte 1 - Caracterizar o trace da sua equipe, respondendo aquelas questões contidas no último slide do tópico 4

As respostas abaixo foram calculadas visando apenas o **Experimento 1** do *dataset*

- 1ª questão: tempo médio entre contatos = 82,84545295770144 segundos
- 2ª questão: duração média dos contatos = 646,820781443506 segundos
- 3ª questão: número médio de reencontros = 16 reencontros
- 4ª questão: tempo médio entre reencontros = 7734,174152032969 segundos