

Programação Distribuída - Relógios vetoriais

Autor: Ariel Rossetto Ril

Matrícula: 15105050

Email: ariel.ril@edu.pucrs.br

Repositório: <https://github.com/arielril/distr-prog-vectorial-clocks>

Introdução

Neste trabalho é apresentado uma implementação de um programa distribuído que utiliza relógios vetoriais para sincronização entre processos. Cada processo do programa distribuído possui uma lista de referência para identificação de onde esta cada processo (IP e porta) para realizar o envio de pacotes UDP.

Desenvolvimento

Para realizar o desenvolvimento deste trabalho foi utilizado a linguagem de programação **Python** e os pacotes:

- *socket*: utilizado para comunicação UDP
- *threading*: para execução dos procesos de recebimento e envio de mensagens de forma independente

Organização

O programa desenvolvido esta organizado em:

- Index
- Starter
- Node

Index

Este é o arquivo principal para organizar a inicialização dos processos e coordenação de encerramento. Neste arquivo é realizado o parsing do arquivo de configuração (*data/config*) na função *parse_config* e é realizado a inicialização da espera do inicio de execução executando a função *await_start*.

```

if __name__ == "__main__":
    try:
        parser = argparse.ArgumentParser(description="Something")
        parser.add_argument(
            "-c", dest="config", help="config file path starting at cwd"
        )
        parser.add_argument("--id", dest="process_id", help="the id of the process")
        args = parser.parse_args()

        has_required_config = (args.config == None or len(args.config) < 1) or (
            args.process_id == None
            or len(args.process_id) < 1
            or int(args.process_id, 10) < 1 # process IDs must be greater than 0
        )

        if has_required_config:
            parser.print_help()
            sys.exit(0)

        configs = parse_config(args.config)
        node_config = get_process_config(configs, args.process_id)
        actual_node = create_node(*node_config)

        node_locations = get_node_locations(configs)
        node_locations.pop(args.process_id)
        actual_node.set_node_locations(node_locations)

        listen_thread = threading.Thread(target=node_listen, args=(actual_node,))

        # interact_thread = threading.Thread(target=node_interact, args=(actual_node,))

        actual_node.await_start()
        listen_thread.start()

        node_interact(actual_node)
        sys.exit(0)
        # listen_thread.join()

        # actual_node.show_results()

```

Starter

Ester arquivo foi criado apenas para inicializar a execução do programa, realizando o envio de uma mensagem de inicialização para os processos usando um grupo multicast.

```

import struct
import socket

mcast = ("224.1.1.1", 8888)

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, socket.IPPROTO_UDP)
s.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, 32)
s.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_LOOP, 1)

s.setsockopt(
    socket.IPPROTO_IP,
    socket.IP_ADD_MEMBERSHIP,
    struct.pack("4sl", socket.inet_aton(mcast[0]), socket.INADDR_ANY),
)

# tested on ipsec connection on wireshark
s.sendto(b"start_bitches", mcast)

s.close()

```

Node

Este é o arquivo principal do trabalho, pois possui toda a lógica para comunicação entre processos, sincronização de processos utilizando relógios vetoriais e controle do relógio local de cada processo. Cada instância possui um dicionário informando onde cada processo está localizado para envio de mensagens de sincronização. Neste arquivo existe os métodos:

- `__init__`: para realizar a inicialização de uma instância de um processo
- `interact`: o qual realiza a geração de eventos locais (incremento do relógio local) e realiza a geração de eventos de mensagem para entre processos (incremento do relógio local e envio de mensagem para outro processo)
- `listen`: para receber as mensagens de outros processos e realizar a sincronização do relógio vetorial
- `await_start`: para esperar a mensagem de inicialização, escutando o grupo multicast UDP

```

class Node:
    local_events = []
    # sent messages
    message_events = []
    received_messages = []

```

```
clock = 0
vclock = []

multicast_group = ("224.1.1.1", 8888)
start_message = "start_bitches"

node_locations = {}
die = False

def __init__(...
): ...

def set_node_locations(self, locations: dict): ...

def init_vec_clock(self): ...

def increment_clock(self): ...

def retrieve_clock(self): ...

def update_clock(self, remote_clock, remote_id): ...

def interact(self): ...

def send_message(self): ...

def add_message_event(self, clock, dst_node_id): ...

def send_local_event(self): ...

def listen(self): ...

def get_sleep_time(self): ...

def show_results(self): ...
```

```
def await_start(self): ...
```

Demonstração

Máquinas virtuais

Para inicias as máquinas virtuais para a demonstração é necessário ter a ferramenta **vagrant**(<https://www.vagrantup.com>) instalada além de possuir VirtualBox instalado. Após instalar as ferramentas necessárias é preciso estar na pasta **machine**, a qual contém o arquivo **Vagrantfile** com as configurações necessárias para inciar 3 máquinas virtuais, para executar o comando **vagrant up**.

host-1	host-2	host-3
<pre>XX ^C1 machine: vagrant ssh host-1 — vagrant (ssh) assign3.nosync/machine on ^ master [!?] via ^ v2.2.16 took 4m37s ^ vagrant ssh host-1 Linux host1 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Last login: Sun Jun 27 14:24:15 2021 from 10.0.2.2 -bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8) vagrant@host1:~\$</pre>	<pre>XX ^C2 machine: vagrant ssh host-2 — vagrant (ssh) Last login: Sun Jun 27 11:43:05 on tty000 assign3.nosync/machine on ^ master [!?] via ^ v2.2.16 ^ vagrant ssh host-2 Linux host2 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Last login: Sun Jun 27 14:24:18 2021 from 10.0.2.2 -bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8) vagrant@host2:~\$</pre>	<pre>XX ^C3 machine: vagrant ssh host-3 — vagrant (ssh) assign3.nosync/machine on ^ master [!?] via ^ v2.2.16 ^ vagrant ssh host-3 Linux host3 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. -bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8) vagrant@host3:~\$</pre>
<pre>XX ^C4 machine: vagrant ssh host-1 — vagrant (ssh) assign3.nosync/machine on ^ master [!?] via ^ v2.2.16 ^ vagrant ssh host-1 ^[[ALinux host1 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64]^ ^[[The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Last login: Sun Jun 27 14:43:19 2021 from 10.0.2.2 -bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8) vagrant@host1:~\$ exit</pre>	<pre>XX ^C5 machine: vagrant ssh host-2 — vagrant (ssh) Last login: Sun Jun 27 11:43:08 on tty003 assign3.nosync/machine on ^ master [!?] via ^ v2.2.16 ^ vagrant ssh host-2 Linux host2 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Last login: Sun Jun 27 14:43:27 2021 from 10.0.2.2 -bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8) vagrant@host2:~\$</pre>	<pre>XX ^C6 machine: vagrant ssh host-3 — vagrant (ssh) Last login: Sun Jun 27 11:43:10 on tty004 assign3.nosync/machine on ^ master [!?] via ^ v2.2.16 ^ vagrant ssh host-3 Linux host3 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Last login: Sun Jun 27 14:43:36 2021 from 10.0.2.2 -bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8) vagrant@host3:~\$</pre>

Simulação

Para executar a simulação é possível iniciar a quantidade que quiser de conexões com as máquinas virtuais, para este trabalho foi utilizado 3 máquinas virtuais. Para este trabalho foi utilizado 6 terminais, 2 em cada máquina virtual. Em 5 terminais foram inicializados os processos para execução da simulação e um terminal foi utilizado para executar o arquivo de inicialização.

