

Instruções de Execução do Algoritmo de Agrawala com PYRO (Versão Corrigida)

Este documento detalha os passos necessários para instalar as dependências, executar o servidor de nomes PYRO, os processos do algoritmo de Agrawala e a simulação, com foco na compatibilidade com sistemas Windows e nas correções implementadas.

1. Instalação de Dependências

Certifique-se de ter o Python 3 instalado em seu sistema. Em seguida, instale a biblioteca Pyro5 usando pip :

Bash

```
pip install Pyro5
```

2. Execução do Servidor de Nomes PYRO

O servidor de nomes PYRO é essencial para que os processos se descubram. Ele deve ser iniciado **antes** de qualquer processo do algoritmo de Agrawala.

Abra um terminal e execute o seguinte comando:

Bash

```
python -m Pyro5.nameserver
```

Deixe este terminal aberto, pois o servidor de nomes precisa estar em execução contínua.

3. Execução dos Processos do Algoritmo de Agrawala Individualmente

Você pode iniciar múltiplos processos do algoritmo de Agrawala manualmente. Cada processo deve ser executado em um terminal separado. Os nomes dos processos devem seguir o padrão `ricart.PeerX` (ex: `ricart.PeerA` , `ricart.PeerB` , etc.).

Para cada processo, abra um novo terminal e execute, utilizando o arquivo `agrawala_corrigido.py` :

Bash

```
python agrawala_corrigido.py PeerA
```

Substitua `PeerA` por `PeerB`, `PeerC`, `PeerD`, etc., para iniciar os outros processos. Por exemplo, para quatro processos:

- **Terminal 1 (Name Server):**
- **Terminal 2 (PeerA):**
- **Terminal 3 (PeerB):**
- **Terminal 4 (PeerC):**
- **Terminal 5 (PeerD):**

Após iniciar todos os processos, eles se registrarão no servidor de nomes e começarão a enviar heartbeats e a atualizar suas listas de peers.

4. Interagindo com os Processos (Exclusão Mútua)

Em cada terminal de processo (`PeerA`, `PeerB`, etc.), você verá o estado atual do processo. Para solicitar o recurso (entrar na seção crítica), digite `p` e pressione Enter.

Plain Text

```
[ricart.PeerA | RELEASED] Digite [p] para pedir recurso, ou [q] para sair: p
```

O processo tentará adquirir o recurso. Uma vez adquirido, ele o manterá pelo tempo definido (`resource_access_time`, padrão 5 segundos) e depois o liberará automaticamente. Você verá mensagens de log indicando o estado do recurso.

5. Testando Detecção de Falhas e Eleição de Coordenador (Opcional)

Para testar a detecção de falhas e o algoritmo de eleição Bully:

1. Inicie o Name Server (conforme Seção 2) e todos os processos (conforme Seção 3).
2. Identifique o processo com o maior PID (por exemplo, `ricart.PeerD` se os PIDs forem lexicograficamente ordenados). Este será o coordenador inicial.
3. Mate o processo que é o coordenador atual (por exemplo, feche o terminal de `PeerD` ou use `Ctrl+C`).
4. Observe os logs dos outros processos. Eles deverão detectar a falha do coordenador e iniciar uma nova eleição. O processo com o maior PID ativo restante se tornará o novo coordenador.

6. Execução do Script de Simulação (`simulate_agrawala.py`)

O script `simulate_agrawala.py` automatiza a inicialização dos processos do algoritmo de Agrawala, direcionando a saída de cada processo para arquivos de log separados (`log_ricart.PeerA.txt` , `log_ricart.PeerB.txt` , etc.).

Importante: O servidor de nomes PYRO (`python -m Pyro5.nameserver`) **deve ser iniciado manualmente em um terminal separado** antes de executar este script.

Abra um terminal (além do terminal do Name Server) e execute:

```
Bash
```

```
python simulate_agrawala.py
```

Este script iniciará os quatro processos (`PeerA` , `PeerB` , `PeerC` , `PeerD`) em segundo plano. Ele imprimirá instruções sobre como interagir com os processos e como verificar os logs.

Para parar a simulação, pressione `Ctrl+C` no terminal onde `simulate_agrawala.py` está sendo executado. Isso encerrará todos os processos iniciados pelo script.

Observações:

- Os PIDs dos processos são formados por `ricart.` seguido do nome que você passa como argumento (ex: `ricart.PeerA`).
- A detecção de falhas e a eleição de coordenador são automáticas e baseadas em timeouts de heartbeat e ping.
- Os logs de cada processo são úteis para depuração e para observar o comportamento do algoritmo.