Trabalho Prático de Sistemas Operativos

Stream Processing

Diogo Fernandes A78867
Jorge Cruz A78895
Luís Ferreira A78607
Grupo 23

May 27, 2018

Contents

1 Introdução

Neste trabalho pretende-se construir um sistema de stream processing. Estes sistemas utilizam uma rede de componentes para filtrar, modificar e processar um fluxo de eventos. Tipicamente, permitem tratar uma grande quantidade de dados explorando a concorrencia tanto entre etapas sucessivas (pipelining) como entre instâncias do mesmo componente.

Este projeto foi desenvolvido em linguagem C, no âmbito da disciplina de Sistemas Operativos, com implementação para sistemas **Unix** e está dividido em duas partes: a implementação do **conjunto de componentes**, cujas tarefas elementares serão realizadas pelos nodos, e do **controlador** que compõe e controla a rede de processamento .

2 Esquematização do Controlador

2.1 Representação Gráfica do Controlador

O esquema da figura 1 permite observar o fluxo de eventos que ocorre quando se executa o comando **inject** no nodo 1. O controlador envia para a entrada do nodo 1 a saída produzida pelo comando presente no **inject**, sendo os eventos filtrados pelo componente do nodo e enviados para os conetados, neste caso o nodo 2 e 3. Este envio de informação entre nodos é realizado através de um processo (distribuidor na figura) que lê constantemente da saída do nodo 1 e envia para a entrada dos nodos 2 e 3.

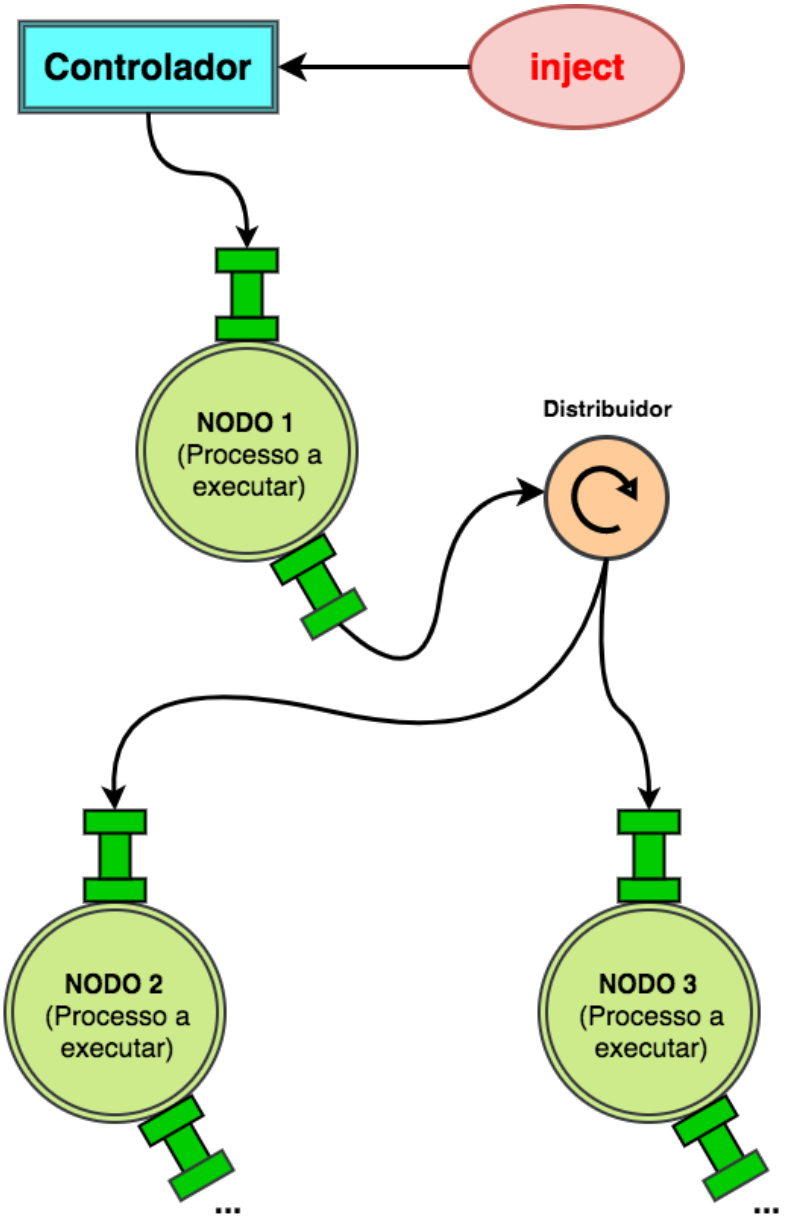

Figure 1: Esquema do controlador.

2.2 Estrutura do Controlador

O módulo do **Controlador** é o responsável pela composição e controlo da rede de processamento, sendo por isso o mais importante. Este módulo é composto por uma lista ligada de nodos que possui:

- Variável com número de nodos;
- Estrutura **nodo** ;

A estrutura nodo é composta por:

- Variável com o ID do nodo;
- Variável com o PID do nodo;
- Variável com o PID do distribuidor do nodo;
- Array com os nodos de saída;
- Variável com o pipeIn do nodo;
- Variável com o pipeOut do nodo;
- Apontador para o próximo nodo;

3 Desenvolvimento

3.1 Consola

A consola está ligada ao controlador através de um named pipe. A sua única função é a de fornecer comandos ao controlador, como por exemplo, inject, connect, disconnect, etc. O seu uso não é obrigatório e serve para fornecer comandos ao controlador enquanto que este está a executar. Para além deste, existe também um ficheiro config.txt onde se pode definir uma rede antes de se executar o controlador.

A consola deve ser executada numa instância do terminal à parte.

3.2 Componentes

Os componentes executam comandos elementares sobre os eventos que são fornecidos aos nodos. Por esta razão, cada nó tem associado um componente em execução num processo distinto do processo do controlador.

Fizemos a implementação de todos os componentes indicados no enunciado:

- Const;
- Filter;
- Window;
- Spawn;

3.3 Controlador

O controlador aceita os comandos Node, Connect, Disconnect e Inject (funcionalidades básicas). Este constrói e gere uma rede composta por nodos e pelas ligações entre esses.

Os nodos e as suas ligações são armazenados numa lista ligada. Cada nodo tem informações relativas ao id do processo onde está a correr o componente que lhe é adjacente, o id do processo do seu distribuidor(ver figura 1), a lista de ligações de saida deste nó e a identificação dos 2 pipes do nodo(um que entra no processo e outro que sai do processo).

Um **nodo** funciona da seguinte maneira. Tem associado um processo onde está a ser executado o componente que lhe é correspondente. Para além disso, tem um PipeIn ligado ao STDIN do componente do nó onde se introduz os eventos que irão ser processados pelo componente e um PipeOut, onde serão recebidos os eventos já processados pelo componente.

Cada nodo tem associado um **distribuidor**. Este é executado num processo distinto e a sua única função é ler do PipeOut do componente e escrever nos PipeIn's dos vários pipes de saída.

Connects e disconnects simplesmente retiram ou adicionam nodos de saída a um nodo, enviam um SIGKILL ao distribuidor atual e criam um processo com um novo distribuidor atualizado.

O **inject** executa um comando sobre um conjunto de eventos e o resultado desse comando é enviado para o nodo especificado, exercitando assim a rede que foi construída. De notar que o STDIN para o comando inject será o próprio terminal onde estará a correr o controlador.

Todos os comandos para o controlador para além dos especificados no ficheiro config.txt deverão ser enviados através da consola. **(3.1)**

4 Dificuldades

O primeiro obstáculo com que nos deparamos foi o de conseguir produzir uma estrutura capaz de resolver o problema em causa, tendo em conta as condições impostas. O facto de o trabalho envolver vários componentes como pipes anónimos, sinais, descritores, pipes com nome, escrita/leitura de ficheiros, entre outros, dificultou bastante esta etapa.

Para além disso, notamos também bastante dificuldade na gestão de situações imprevistas, como comandos inválidos, parâmetros de comandos ilegais, etc(controlo de erros).

Por último, achamos também difícil a gestão dos pipes, sobretudo no redirecionamente dos seus descritores e no fecho de descritores que já não eram necessários.

5 Conclusão

Consideramos que este projeto trata um assunto importante, já que tem aplicabilidade na área da "internet das coisas", área da tecnologia que atualmente está em grande expansão.

Mais importante ainda, ajudou-nos a perceber como é feita a gestão, criação e comunicação entre processos, a escrita e leitura de ficheiros(e não só), e a utilização de sinais.