

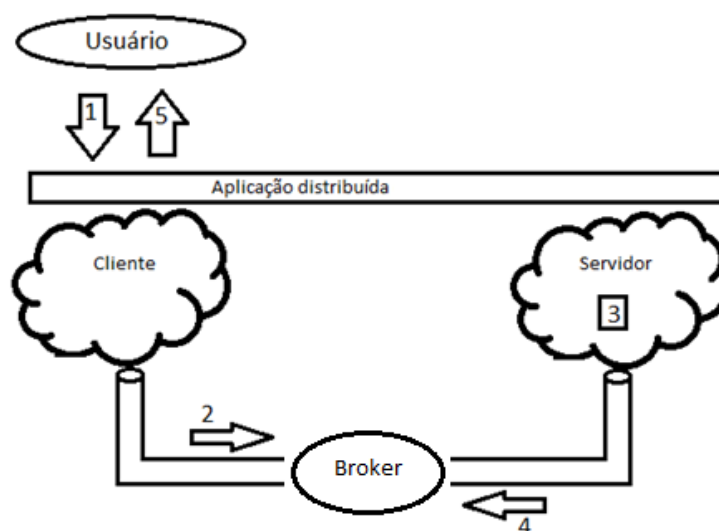
PSPD - Programação para Sistemas Paralelos e Distribuídos  
Prof. Fernando W Cruz

### Laboratório – Construindo aplicações distribuídas usando o paradigma *publish-subscriber*

**A) Objetivo:** O objetivo desse experimento é que o aluno compreenda algumas funcionalidades dos *brokers*, que funcionam como intermediários em conexões envolvendo módulos cliente e servidor.

#### B) Detalhes do laboratório

Esse experimento deve ser feito usando um *broker* RabbitMQ(<https://www.rabbitmq.com/getstarted.html>) ou Kafka (<https://kafka.apache.org/>) e a aplicação deve ser construída em ambiente Linux/gcc. Os alunos devem construir uma pequena aplicação distribuída (linguagem C), que contabilize o número de palavras de um arquivo de entrada, com a utilização de um serviço de *broker* intermediário, como ilustrado na Figura 1.



**Figura 01** – Arquitetura de uma aplicação distribuída usando *broker publish-subscriber*

Nesse caso, o processo master deve interagir com pelo menos um processo *worker*, via intermediação de um *broker*. As contabilizações feitas pelo *worker* devem ser feitas da seguinte forma: (i) contabilizar todas as palavras com menos de 6 caracteres, (ii) contabilizar as palavras que tenham entre 6 e 10 caracteres e, (iii) contabilizar todas as palavras do arquivo de entrada. Observações:

- Gerar testes iniciais com arquivos de tamanhos distintos, em modo convencional (sem uso de *broker*), a fim de identificar o tamanho máximo do arquivo de entrada que essa aplicação não distribuída consegue processar.
- O arquivo de entrada deve ter um tamanho superior ao que foi identificado no item anterior.

- c) Testar a aplicação com 1, 2, 4 e 6 *workers*, identificando se o desempenho da aplicação fica melhor, em função do aumento da quantidade de *workers*. Gerar uma tabela com essas informações a fim de permitir comparar desempenho.
- d) Procurar fazer a distribuição do trabalho, dividindo o arquivo proporcionalmente ao número de *workers* definido no item anterior.
- e) Ao final, os *workers* devem devolver os valores contabilizados para que o *master* possa apresentá-los na tela.

### C) Questões de ordem

- A atividade pode ser feita por grupos de até 2 alunos.
- Caso solicitados, os alunos devem estar preparados (slides e apresentação do código) para demonstração da solução em sala de aula, conforme sorteio definido pelo professor em data oportuna.
- A entrega será feita pelo envio de um arquivo zipado no ambiente Moodle da disciplina disponível em <http://aprender3.unb.br>. O arquivo zipado deve conter: (i) os arquivos .c das aplicações criadas, (ii) instruções de instalação do *broker* e dos processos que compõem a aplicação, e (iii) um relatório cujo conteúdo está descrito mais adiante.
- Os códigos, devidamente comentados e identados, devem ser entregues em conjunto com um relatório sobre a atividade realizada, o qual deve conter os seguintes pontos:
  - a) Título do experimento, dados da disciplina e do(s) aluno(s)
  - b) Introdução – pequena descrição do problema e do *broker* adotado (Kafka ou RabbitMQ)
  - c) Descrição da solução convencional e com uso do *broker* para arquivos de tamanhos variados, procurando apontar problemas e soluções encontradas até chegar à versão final do problema proposto. Incluir aqui eventuais limitações dos códigos entregues.
  - d) Documentar como as informações do arquivo de entrada foram organizadas no *broker* para permitir uma distribuição equilibrada entre os *workers*.
  - e) Incluir instruções de de instalação e uso dos processos comunicantes (e do *broker*, se for o caso). Caso seja gerado código diferente para esse vetor alterado, incluí-lo no mesmo arquivo zipado que será postado no Moodle.
  - f) Opinião geral sobre o experimento, apontando pontos onde contribuiu mais no laboratório, aprendizados e possíveis limitações percebidas no experimento, além de nota de auto-avaliação.