

MAC0352 - Redes de Computadores e Sistemas Distribuídos EP2

Data de Entrega: 16/10/2017

Prof. Daniel Macêdo Batista

1 Problema

Neste EP você deverá implementar um sistema distribuído que funcione na rede local para realizar um teste de primalidade de um número. O sistema deve ser composto por diversas máquinas em uma rede local (a correção será feita com 3 máquinas) e deve funcionar seguindo uma arquitetura P2P. A invocação do primeiro código deve ser feita recebendo como parâmetro apenas o número a ser testado. As outras invocações do código nos outros computadores não devem receber nenhum parâmetro na linha de comando. Todos os computadores conectados no sistema devem dividir o trabalho entre si seguindo qualquer regra definida por você. Além disso, essa regra deve se adaptar automaticamente à quantidade de computadores na rede. Ou seja, assim que você executar o primeiro código ele já deve começar, sozinho, a verificar se o número é primo. Quando o segundo computador entrar na rede, ele deve dividir a tarefa com o primeiro. O mesmo vale para os computadores seguintes. Se algum computador sair da rede, também é necessário redefinir a divisão de tarefas. Todos os códigos devem “se encontrar” na rede automaticamente sem definição explícita de endereços IP de nenhum deles. A comunicação entre as máquinas pode usar UDP ou TCP. Você decide.

Assim que o teste de primalidade terminar, todos os códigos devem finalizar as suas execuções nos vários computadores. Ao terminar, na saída padrão de todos os códigos, deve ser exibido o resultado do teste de primalidade. Computações desnecessárias devem ser evitadas. Ou seja, assim que uma máquina na rede descobrir que um número é ou não primo, ela deve avisar todas as outras máquinas para que finalizem as suas execuções.

Uma observação importante é que o código **não deve** fazer nenhuma verificação usando tabelas pré-computadas de números primos. As contas devem ser realizadas sem nenhum pré-conhecimento sobre se o número é primo ou não. Isso é necessário para garantir que o código vai rodar por tempo suficiente para haver uma quantidade suficiente de mensagens trocadas via rede.¹

2 Requisitos

O seu código deve implementar um protocolo de rede que atenda aos seguintes requisitos:

¹Esse EP é sobre redes e não sobre análise ou complexidade de algoritmos. Portanto, não se preocupe em implementar o melhor algoritmo para o teste de primalidade. Até porque, se o algoritmo tiver baixa complexidade, pode ser que ele rode rápido demais e não dê tempo de nenhuma mensagem ser trocada via rede

- descoberta automática das demais máquinas que já estão rodando na rede local (A porta pode estar pré-definida no código mas os endereços IP não. Eles devem ser descobertos automaticamente usando qualquer método);
- verificação periódica de que as máquinas continuam conectadas (Esse mecanismo existe em diversos sistemas e é chamado de *heartbeat*);
- eleição de um líder que será responsável por dividir as tarefas entre as demais máquinas (Essa eleição deve ser realizada periodicamente para evitar que uma mesma máquina na rede seja um ponto único de falha);
- comunicação contínua informando o estado das computações de modo que todas as máquinas saibam como estão as computações nas demais (Essa manutenção não precisa ser realizada nos mínimos detalhes. Por exemplo, pensando no algoritmo mais simples que faria divisões sucessivas, informar às demais máquinas o resultado de cada divisão tornaria o código muito lento. Bastaria informar por exemplo a cada 100 ou 1000 divisões).

O protocolo criado por você deve usar comandos em ASCII para todas as ações.

O seu código deve rodar também em modo DEBUG (a definição do modo DEBUG pode ser realizada com alguma flag de compilação) e, nesse modo, todas as máquinas da rede deverão manter um arquivo de log informando tudo que aconteceu durante o tempo em que o código ficou rodando. Esse arquivo de log deve informar o momento do evento e qual foi o evento. Alguns eventos que não podem deixar de serem registrados são:

- Entrada de uma nova máquina no sistema;
- Saída de uma máquina do sistema;
- Início da eleição do líder;
- Fim da eleição do líder, com o resultado da eleição;
- Término da computação, com o resultado do teste de primalidade e qual máquina que descobriu se o número é primo ou não.

2.1 Linguagem

Os programas podem ser escritos em qualquer linguagem de programação, desde que exista compilador/interpretador gratuito para GNU/Linux, e devem funcionar no shell, sem interface gráfica. Certifique-se de que seu programa funciona no GNU/Linux pois ele será compilado e avaliado apenas neste sistema operacional.

Você não pode utilizar bibliotecas, classes ou similares que já implementem um sistema distribuído similar ao pedido. Códigos que não respeitem esse requisito terão nota ZERO.

2.2 Slides

Diferente do EP1, neste EP você deverá entregar, além dos códigos, um .pdf com slides que você usaria caso você fosse apresentar o seu trabalho. Os slides deverão apresentar o seu protocolo, informando se ele foi baseado em algum protocolo existente ou se você criou ele do zero. Outras decisões de projeto

que você julgar que merecem ser apresentadas também podem ser incluídas. Além das informações sobre o protocolo, os slides também devem apresentar gráficos de análise de desempenho comparando o desempenho do sistema distribuído com 1, 2 e 3 máquinas a fim de verificar se houve ganhos em distribuir as tarefas (o tempo necessário para terminar o teste de primalidade seria a melhor métrica nesse caso). Inclua também gráficos que mostrem uso de CPU e uso da rede com as 3 quantidades diferentes de máquinas. Nos slides inclua informações sobre como você avaliou a carga na rede de modo a garantir que tudo que você mediu foi de fato decorrente da comunicação do seu código e não de outros programas que estavam usando a rede ao mesmo tempo.

Os seus slides devem ser feitos supondo que eles seriam apresentados em um tempo máximo de 10 minutos, portanto cuidado para não preparar slides muito extensos.

Entregas sem o .pdf da apresentação não serão corrigidas e receberão nota ZERO.

3 Entrega

Você deverá entregar um arquivo .tar.gz contendo os seguintes itens:

- fonte;
- Makefile (ou similar);
- arquivo LEIAME;
- .pdf dos slides.

O desempacotamento do arquivo .tar.gz deve produzir um diretório contendo os itens. O nome do diretório deve ser ep2-membros_da_equipe. Por exemplo: ep2-joao-maria.

A entrega do .tar.gz deve ser feita através do PACA.

O EP pode ser feito individualmente ou em dupla.

Obs.: Será descontado 1,0 ponto de EPs que não estejam nomeados como solicitado, que não criem o diretório com o nome correto após serem descompactados ou que não contenham todos os arquivos necessários.

Obs.: O prazo de entrega expira às 8:00:00 do dia 16/10/2017. Não serão aceitos EPs atrasados.

4 Avaliação

60% da nota será dada pela implementação, 10% pelo LEIAME e 30% pelos slides. Os critérios detalhados da correção serão disponibilizados apenas quando as notas forem liberadas.