**CES-33**

**Laboratório 5 – Balanceamento de Carga**

*O objetivo deste laboratório é o estudo de algoritmos de alocação de processador, ou seja, como processos podem ser atribuídos à CPUs em um sistema de multiprocessadores de forma a manter o sistema o mais balanceada possível. Algumas métricas serão consideradas.*

Um processo é criado em um determinado nó do sistema de multiprocessadores. Seria esperado que pudesse ser executado no próprio nó onde foi gerado, todavia, se o nó estiver sobrecarregado, este processo deveria ser escalonado para executar em outro nó. Considere que em cada processador, apenas um processo é executado por vez enquanto os outros estariam esperando em uma fila.

Algumas métricas de ocupação do processador que devem ser consideradas:

* quantidade de processos;
* taxa média de utilização de CPU;
* quantidade de processos relativa ao total de processos do sistema;

No caso de sobrecarga, uma heurística deve ser desenvolvida para decidir quem poderia abrigar este processo.Heurísticas possíveis:

* **Algoritmo heurístico distribuído iniciado pelo emissor**

Uma heurística frequentemente empregada consiste em selecionar aleatoriamente outro nó do sistema e questionar a sua carga segundo a métrica escolhida. Se a carga do nó investigado está abaixo de certo limiar, o processo é enviado a ele, caso contrário, outro nó é escolhido para a investigação.

Se nenhum nó adequado foi encontrado em *RETRY* tentativas, o algoritmo termina e o processo executa na máquina onde foi originalmente gerado. Desta maneira, os nós sobrecarregados tentam se livrar do excesso de trabalho.

Observa-se que em condições de muito trabalho, todas as máquinas vão sondar outras em vão, pois nenhuma vai aceitar o trabalho e haverá uma sobrecarga considerável na tentativa de encontrar alguma (excesso de mensagens de sondagem).

* **Algoritmo heurístico distribuído iniciado pelo receptor**

Um algoritmo complementar é um algoritmo que é iniciado por um receptor com pouca carga. Neste algoritmo quando um processo finaliza, o sistema verifica se ele dispõe de trabalho suficiente. Em caso negativo, ele escolhe alguma máquina aleatoriamente e solicita trabalho para ela. Se a máquina escolhida não tem nada a oferecer, investiga uma próxima máquina. Se nenhum trabalho é encontrado dentro de *RETRY* investigações, o nó para temporariamente de procurar, e tenta novamente quando o próximo processo acabar. Se nenhum trabalho está disponível, a máquina fica ociosa. Após algum intervalo fixo de tempo, o nó disponível retoma a sondagem novamente.

Uma vantagem deste esquema é que não sobrecarrega o sistema nos momentos críticos. Quando há pouco trabalho no sistema, o número de mensagens de sondagem é maior (todos estão caçando emprego), contudo a sobrecarga extra nesta situação é menos grave.

* **Algoritmo híbrido**

As máquinas tentam se livrar de trabalho quando este for excessivo e tentem adquirir trabalho quando não tiverem o suficiente.

A **tarefa** deste laboratório:

Escreva um programa que implemente os tres algoritmos de balanceamento. A entrada do programa é uma lista de novas tarefas criadas, especificadas como (NúmeroCPUqueCriou, HoraCriaçãoTarefa, TempodeCPUNecessário).

* NúmeroCPUqueCriou: identificação da CPU que criou a tarefa;
* HoraCriaçãoTarefa: momento da criação da tarefa;
* TempodeCPUNecessário: tempo de CPU que a tarefa precisa para ser executada (em segundos).

Imagine uma situação onde um nó está sobrecarregado quando uma nova tarefa é criada.

Considere que outro nó não tem tarefa alguma e, portanto, está ocioso. Exiba a quantidade de mensagens de sondagem enviadas pelos dois algoritmos quando a carga de trabalho está **intensa** e quando ela está **leve**. Exiba também os números máximo e mínimo de sondagens enviadas e recebidas por quaisquer dos nós.

Para criar a carga de trabalho, escreva dois geradores. O primeiro deve simular uma carga **intensa** e gerar, em média *Amount* tarefas a cada *TMT* segundos, onde:

* *TMT* é o tamanho médio da tarefa em segundos;
* *Amount* é a quantidade de tarefas;
* *N* é o número de processadores;

O tamanho das tarefas pode variar de longo a curto, mas o tamanho médio deve ser *TMT*. As tarefas devem ser criadas de forma aleatória (em cada nó) e distribuídas por todos os processadores. O segundo gerador deve simular uma carga de trabalho **leve** e criar *Amount/3* tarefas a cada *TMT* segundos.

Exiba o seguinte resultado para cada heurística:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Carga** **Leve** | | **Carga Intensa** | |
| **Nó** | **Nmsg Sondagem transmitidas** | **Nmsg Sondagem recebidas** | **Nmsg Sondagem transmitidas** | **Nmsg Sondagem recebidas** |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| ... |  |  |  |  |
| N |  |  |  |  |

Modifique os parâmetros dos geradores de carga (*N, TMT, Amount*) e observe de que forma as mudanças afetam as mensagens de sondagem.

O lab valerá 13 para quem implementar a seguinte melhoria:

As mensagens de sondagem são necessárias pois não se sabe a priori a carga de cada nó, ou seja, o sistema de alocação de tarefas é distribuído. Implemente um nó gerente que conhece a carga de cada nó, uma vez que ele determina quem executa cada tarefa, ou seja, abordagem centralizada. Discuta o que se ganha e o que se perde neste caso.

O relatório deve conter 4 seções:

1. **Introdução**: Comente qual é o problema e o objetivo do experimento. Breve descrição das heurísticas empregadas e decisões de projeto;
2. **Balanceador de carga**: Descrição de seu experimento. Detalhes necessários para a compreensão do que foi feito. Pseudo-código pode ser empregado. Aponte para o anexo onde inclua o código fonte comentado.
3. **Resultados**: Apresente as tabelas com o volume das mensagens trocadas em cada cenário.
   1. Algoritmo heurístico distribuído iniciado pelo emissor
   2. Algoritmo heurístico distribuído iniciado pelo receptor
   3. Algoritmo híbrido

Comente os resultados e os porquês. Comente a influência dos parâmetros (*N, TMT, Amount*)

1. **Conclusão**. Considerando a discussão do item anterior, chegue a sua conclusão (ex: Em que caso é bom usar qual heurística). Se você não soubesse se a carga será leve ou intensa que algoritmo escolheria? Justifique.

A tarefa deve ser desenvolvida em grupo de 2 alunos. Recomendo que cada aluno desenvolva uma heurística e haja uma competição entre os alunos do grupo na busca do melhor resultado.

O programa será apresentado no dia 06/06 na aula.