



Aula Prática 4

Os oito divisores de 24 são 1, 2, 3, 4, 6, 8, 12 e 24. Os dez números não excedendo 100 tendo exatamente oito divisores são 24, 30, 40, 42, 54, 56, 66, 70, 78 e 88. Seja $f(n)$ uma função que representa a quantidade dos números menores ou iguais a n com exatamente oito divisores. Como referência:

$$f(100) = 10$$

$$f(1000) = 180$$

$$f(10000) = 2114$$

Tarefa: Seu objetivo é fazer um código sequencial que encontre $f(1000000)$ no menor tempo possível.

Mesmo sendo um problema simples, o tempo gasto para resolvê-lo pode ser grande (dependendo do valor de n). Usando o Intel DevCloud, tanto por SSH quanto pelo JupyterHub, ficamos limitados ao tempo de sessão/login (que é insuficiente para resolvermos problemas grandes). Ao sermos desconectados, todos os processos que estão executando são encerrados. Essa política é adotada em praticamente todos os clusters/supercomputadores, por questões de segurança e melhor utilização dos recursos computacionais (um dos nós do cluster é responsável pelo login dos usuários, imagine um usuário executando um programa que “consome” todo o processador).

Para tirarmos todo proveito que o Intel DevCloud nos oferece, precisamos submeter um *job* (gerenciado pelo cluster), assim o tempo limite de execução pode ser de até 24h (em outros cluster de processamento, não é comum se falar em tempo limite de execução de um *job*).

Submetendo um *job* no Intel DevCloud¹

Ao acessar o Intel DevCloud por meio de SSH, você será conectado a um nó de login (*hostname login-1*). Neste nó, você pode editar o código e compilar aplicativos. No entanto, para executar aplicativos computacionais, você deve enviá-los a uma fila de trabalho (chamada de *job*) para ele seja executado em um dos nós do cluster. Cada nó é composto por processadores e memória que podem ou não ser compartilhados. Ao submeter um *job* você indica quantos nós (consequentemente quantos processadores ou núcleos de processamento) seu programa precisará. O gerenciador de *jobs* do cluster se encarregará de iniciar a execução do seu *job* assim que o cluster tiver os recursos solicitados por você disponíveis.

Ao acessar o Intel DevCloud por meio do JupyterHub, você será conectado a um dos nós. Em comparação com o nó de login, você terá mais recursos locais à sua disposição. No entanto, alguns recursos, como tempo de execução maior (você é desconectado do JupyterHub automaticamente após um certo tempo e

¹Fonte: <https://devcloud.intel.com/oneapi/learn/advanced-queue/>

todos os processos que estão executando são encerrados) e computação de vários nós, só estão disponíveis na com a submissão de *jobs*.

Considerando que a compilação do seu código gerou um executável com nome “programa”, o arquivo `job.sh` é responsável por executar o seu programa (ele é muito parecido com o arquivo `run.sh`, mas com apenas uma execução do seu programa). Se n for pequeno (menor que 10000), você pode testar o arquivo `job.sh` no terminal do JupyterHub ou do SSH (no terminal basta fazer `./job.sh`).

Caso você receba a mensagem “bash: ./job.sh: Permission denied” ao executar o comando “./job.sh”, basta executar o comando “`chmod +x job.sh`” (esse comando dará permissão de execução do `job.sh`).

Se tudo estiver certo, você verá a saída do seu programa no terminal, mas como dito, devemos submeter esse programa a um outro nó (não o nó de login) para fazer a computação. Para isso, podemos usar o comando abaixo:

```
qsub -l nodes=1:gpu:ppn=2 -d . job.sh
```

Para mais detalhes acesse esse link: <https://devcloud.intel.com/oneapi/learn/job-submission/>

Ao ser submetido um *job*, serão criados dois arquivos: `job.sh.e<jobID>` e `job.sh.o<jobID>`. O arquivo `job.sh.e<jobID>` irá conter as informações de erros do *job*. Já o arquivo `job.sh.o<jobID>` irá conter toda as saídas do seu programa e do `job.sh`. Provavelmente, o arquivo `job.sh.o<jobID>` que conterá a informação que você precisa.

Para obter informações sobre os seus *jobs* submetidos, basta usar o comando `qstat`, que listará todos os seus *jobs* e o *status* de cada um (se está na fila aguardando ou se está em execução). Se ele estiver em execução, na penúltima coluna aparecerá um R (*Running*), se ele estiver na fila de execução você verá um Q (*Queue*).

Caso seja necessário cancelar a execução de um *job*, use o comando `qdel <jobID>`.

Para saber mais:

<https://devcloud.intel.com/oneapi/learn/advanced-queue/managing-submitted-jobs>

O que entregar?

Nessa tarefa você deve enviar o arquivo com o código do seu programa e um pequeno comentário (2 linhas) ou um arquivo `.txt` com as informações abaixo (obtidas pelo seu programa):

```
1 f(1000000) =
2 Tempo gasto (em seg.):
```

Data de entrega: até às 23:59h do dia 12/10/2020.