Projeto de Implementação 2

Publicado em **12/12/2022 - Entrega em 19/12/2022**

Executar em Duplas

Para este e para os demais PIs, todos os alunos vão precisar de contas no e-mail institucional do **IComp/UFAM (ou seja, @icomp.ufam.edu.br)**. Todos devem criar uma conta para ter acesso ao Google Collaboratory pois é através dele que que os Projetos de Implementação (PIs) serão entregues.

Neste Projeto de Implementação os alunos deverão implementar em PySpark o algoritmo PageRank estudado durante a aula e executar em um cluster na nuvem. Abaixo apresento instruções para criação de um cluster em um serviço de computação em nuvem de forma gratuita. Também vou definir o que deve ser entregue e como.

Parte 1. Criação um Cluster em Nuvem

Existem diversos provedores de serviços de computação em nuvem que fornecem serviços de criação e operações de clusters computacionais. Na maioria dos casos, pode-se configurar um cluster onde versões atuais do Hadoop e do Spark já são automaticamente instalados e configurados, incluindo HDFS e outros pacotes e sistemas adicionais e complementares. Esse é o caso do Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), Databricks, Cloudera, etc. Vários deles oferecem, de forma gratuita, um valor de crédito para utilização de seus serviços por um período determinado de tempo. Eu testei alguns deles, e dentre estes, o que eu achei mais simples de usar e que oferece um valor de créditos bastante razoável para os nossos propósitos na disciplina é o Google Cloud Platform (GCP). Então é este serviço que eu vou utilizar abaixo.

Passo 1. Criação de um Conta Gratuita no GCP

Para a criação de uma conta gratuita no GCP, vocês podem olhar este tutorial [1], onde está tudo explicado em detalhes. Para criação da conta, vai precisar de um cartão de crédito, mas não vai haver cobrança sem autorização. A conta “Free Trial” inclui US$ 300 de crédito e dura 12 meses. Isso deve ser mais que suficiente para o nosso cluster e para as nossas atividades.

Existem outros materiais que podem ser consultados e vocês podem usar qualquer um destes, mas espera-se que ao final do processo cada aluno tenha uma **conta** e um **projeto** criados no GCP. Um projeto é como um espaço de trabalho dentro do GCP. O GCP gerencia todos os recursos, credenciais, permissões e informações de cobrança através de projeto.

Passo 2. Criação de um Cluster no GCP

Uma vez que a conta e o projeto foram criados no GCP, é muito fácil criar um cluster. Este tutorial [2], em vídeo, do mesmo autor do tutorial [1], tem tudo bem explicado. Pena que não

tem em formato textual 😢. Sugiro que assistam até o final tomando nota para depois executar os passos. Vocês vão ver que a interface do GCP atual é ligeiramente diferente, mas dá pra seguir tranquilamente.

Uma observação muito importante: como o vídeo explica, o GCP inclui um serviço chamado Cloud Dataproc que oferece três métodos alternativos de criar um cluster. (A) Através de um Console Gráfico do GCP; (B) através de linhas de comandos usando o GCP Shell; e (C) usando uma interface programática da API Cloud Dataproc. Destes três, no dia-a-dia vamos evitar o método (A), embora ele seja bem didático da primeira vez que se cria. A razão para isso é simples. Embora seja meio inconveniente, devemos **apagar o cluster completamente** depois de usar. Se precisar, **basta simplesmente criar novamente** quantas vezes for necessário. Se o cluster não for removido, mesmo que não esteja sendo utilizado, vai consumir os nossos créditos gratuitos bem rapidamente. Por isso, os métodos (B) e (C) são bem mais rápidos e simples que o método (A).

Particularmente, eu estou usando o Método (B), pois isso, eu achei conveniente usar o GCP Shell no meu notebook no Colab. Para isso, foi necessário instalar o GCP SDK. Vocês podem também usar o GCP SDK através de um browser. O GCP SDK também vai ser útil em outros momentos, portanto, melhor instalar.

O tutorial [2] acima explica o método (A) e eu recomendo que usem ele da primeira vez. Depois, podem usar os passos deste tutorial e outro tutorial [3] que explica razoavelmente o método (B). Como o tutorial [3] cria um cluster de um nó somente, podemos usá-lo para criar um cluster de seis nós, como o do tutorial [2], bem mais interessante para nossas atividades. Desde já, ***aceito voluntários*** para juntar os dois tutoriais e fazer um só em formato de texto 😅.

Para facilitar, segue a linha de de comando do GCP shell que eu estou usando para montar meus clusters, com alguns comentários:

```
1     gcloud dataproc clusters create ufam-111 \
2     --project ufam-bgd-2022-02 \
3     --region us-central1 \
4     --subnet default \
5     --zone us-central1-c \
6     --master-machine-type n1-standard-1 \
7     --master-boot-disk-size 32 \
8     --num-workers 5 \
9     --worker-machine-type n1-standard-1 \
10    --worker-boot-disk-size 32 \
11    --image-version 1.4-ubuntu18 \
12    --optional-components ANACONDA,JUPYTER \
13    --bucket ufam-bucket-1
```

Esse comando vai criar um cluster chamado **ufam-111** no GCP Dataproc. O cluster deve ser criado no projeto de ID **ufam-bgd-2022-02**, que já foi criado antes. O cluster vai rodar na região **us-central**, que é mais barata. Valor **default** de subnet mantido. Zona do cluster é **us-central1-c**, correspondente à região. Tipo de máquina do nó master é **n1-standard-1**, a mais em conta. O nó master vai usar um disco local HD comum de **32** GB, que é suficiente para nós. Nosso cluster terá **5** nós *workers*, além do master. O tipo de máquina dos nós worker também é **n1-standard-1** e também vão usar um disco local HD comum de **32** GB. O

cluster rodará uma imagem **1.4-ubuntu18**, baseada no Ubuntu Linux. Os detalhes desta configuração, incluindo versões do Hadoop, Spark, Linux, etc. estão aqui: **1.4-ubuntu18** . Existem várias opções. Além do SO, Hadoop, Spark, etc., estou incluindo Python e Jupyter Notebook. Um bucket é um disco virtual distribuído usado pelo cluster. Deve ser criado antes (ver abaixo).

Algumas observações:

- Para nomear seu cluster use a seguinte convenção: **NOME1-NOME2**
    - **NOME1** : nome do(a) primeiro(a) aluno(a) da dupla
    - **NOME2** : nome do(a) segundo(a) aluno(a) da dupla
- Os detalhes sobre o comando e os parâmetros estão disponíveis aqui, mas estes devem funcionar bem para as nossas atividades.
- Os parâmetros relacionados ao hardware não são os default do GCP, mas visam economizar créditos sem comprometer o que precisamos nas nossas atividades.
- Os pacotes **ANACONDA** e **JUPYTER**, correspondem ao interpretador Python e ao servidor de Jupyter Notebook, respectivamente.
- O bucket (neste caso, **ufam-bucket-1)** deve ser criado antes. Pode-se usar o usar o GCP Console ou a linha de ccomando
    - gsutil mb -c **standard** -l **us-central1** gs://**ufam-bucket-1** (detalhes aqui)
- Não esquecer de remover o cluster depois que de usar para evitar consumo de créditos desnecessariamente. Pode se usar o comando gcloud dataproc clusters delete

Passo 3. Testar o Cluster Criado

Com o cluster criado, podemos rodar um script Python para testar seu funcionamento. Estas instruções estão em [3].

```
1    gcloud dataproc jobs submit pyspark wordcount.py \
2    --cluster=ufam-111 -- \
3    gs://la-gcp-labs-resources/data-engineer/dataproc/romeoandjuliet.txt \
4    gs://ufam-bucket-1/output/
```

O script **wordcount.py** pode ser obtido com o seguinte comando:

```
1    gsutil cp gs://la-gcp-labs-resources/data-engineer/dataproc/wordcount.py
```

Ele deve estar acessível no diretório da máquina local. O script está sendo submetido para o cluster **ufam-111**, criado acima. O arquivo txt usado como entrada já está disponível no GCP para testes A saída será escrita no diretório **output**, no **ufam-bucket-1** definido acima

Depois de rodar o script, podemos usar o seguinte comando para ver o resultado da execução.

```
1     gsutil cat gs://ufam-bucket-1/output/*
```

Passo 4. Acessar Interfaces Web do Cluster

O cluster criado possui uma série de Interfaces Web que dão informações sobre o cluster e sobre os jobs executados. No entanto, para evitar que estas interfaces sofram ataques é necessário criar um túnel de acesso seguro ao servidor web do cluster. O tutorial [2] explica em detalhes esse passo. No entanto, em resumo, os seguintes comandos devem ser executados em um terminal. Nos dois comandos abaixo, devem ser ajustados no nome do cluster (no exemplo, **ufam-111**) e do projeto (no exemplo, **ufam-bgd-2022**).

- Criação do túnel de acesso seguro ao Servidor Web do cluster. Deve ser executado em um terminal e deve ficar ativo para manter o túnel aberto enquanto for necessário acessar o Servidor Web do cluster.

```
1    gcloud compute ssh ufam-111-m --project=ufam-bgd-2020 \
2    --zone=us-central1-c -- -D 1080 -N
```

- Executar o browser Chrome através do túnel para acessar o Servidor Web. Assumimos que o nós do cluster rodam Linux Ubuntu.

```
1    /usr/bin/google-chrome   --proxy-server="socks5://localhost:1080"
2    \
      --user-data-dir="/tmp/ufam-111-m" http://ufam-111-m:8088
```

Passo 5. Jupiter Notebook [opcional]

Uma vez que o acesso às Interfaces Web estiver configurado no Passo 5, é possível acessar o servidor Jupyter Notebook acessando a seguinte URL no Browser (Chrome), assumindo que o nome do cluster é **ufam-111**.: http://**ufam-111-m**:8123/,

Passo 6. Acessando o cluster a partir do Google Colab

Esse PI3 é uma excelente oportunidade para desenvolver o entendimento sobre criação dinâmica de um cluster virtual sob demanda. Neste PI3, isso deve ser feito dinamicamente a partir do Notebook colab e durante o processamento das consultas, as requisições devem ser feitas ao cluster criado. Não esqueçam de remover o cluster ao final do processo, senão serão feitas cobranças além dos US$ 300 de crédito. Algumas de dicas sobre como conectar o Notebook colab com o cluster no CGP estão disponíveis aqui [4]

Parte 2. Atividade a ser realizada

A atividade a ser realizada consiste na implementação em PySpark do algoritmo **PageRank** visto durante e sua execução experimental com um dataset indicado. Os alunos devem tomar como base o trecho de código do PageRank apresentado durante as aulas.

Para a execução experimental, os alunos devem usar o *Berkeley-Stanford web graph,* onde os nós representam páginas dos domínios berkely.edu e stanford.edu e as arestas direcionadas representam hiperlinks entre estas páginas. Os dados foram coletados em 2002.

3. O Que Entregar

- Jupyter Notebook com código completo da implementação do PageRank, incluindo os passos de criação do cluster em nuvem.
  - O notebook deve ser chamar
    - **`PI2-BGD-2022-02-NOME1-NOME2`**
      - `NOME1` : nome do(a) primeiro(a) aluno(a) da dupla
      - `NOME2` : nome do(a) segundo(a) aluno(a) da dupla
  - O notebook deverá estar pronto para execução sem erros.
- Lista ordenada pelo valor do pagerank das 100 páginas do dataset com maior valor de pagerank. Essa lista deve ser gerada como o último resultado da execução do Notebook.
- Conteúdo da página

  http://NOME1-NOME2-m:18080/history/application_NNNN_YYYY/executors/

  que é gerada a partir da execução do job no cluster Spark. Esse link é gerado como a última linha resultante da execução do comando de submissão do script no terminal. Para acessá-la, é necessário usar o browser através do túnel criado, como explicado no **Passo 4**.

  Esse conteúdo deverá ser copiado em uma célula "Markdown" do Notebook

4. Como Entregar

- **P**ara a entrega, os alunos devem compartilhar o Notebook no Google Colab compartilhado com *alti@icomp.ufam.edu.br*. O professor não solicitará compartilhamento. O notebook compartilhado deve ser "pinado" (Save and Pin Revision) e não deve ser modificado depois disso. O professor verificará a "Revision History" do notebook e não corrigirá o projeto se houverem versões posteriores ao prazo de entrega.
- **Correção:** A correção será feita da seguinte forma: eu vou acessar o notebook compartilhado, executá-lo e verificar se a saída está de acordo. Também vou examinar e avaliar o código do notebook. Os alunos devem preparar o notebook de forma que ele inclua a criação do cluster Spark virtual.

4. Referências

[1] Free virtual machine in Google Cloud

[2] Google Cloud Tutorial – Hadoop | Spark Multinode Cluster | DataProc

[3] Running a Pyspark Job on Cloud Dataproc Using Google Cloud Storage

[4] Colab+GCP Compute — how to link them together